User's Manual

S5D9 Microcontroller Group

User's Manual

Renesas Synergy™ Platform

Synergy Microcontrollers
S5 Series

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.30 Aug 2019

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

General Precautions

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Preface

1. About this Document

This manual describes the functions and electrical characteristics of the Renesas Synergy™ Microcontroller.

This manual is generally organized into an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes. This manual describes the product specification of the microcontroller (MCU) superset. Depending on your product, some pins, registers, or functions might not exist. Address space that store unavailable registers are reserved.

2. Audience

This manual is written for system designers who are designing and programming applications using the Renesas Synergy Microcontroller. The user is expected to have basic knowledge of electrical circuits, logic circuits, and the MCU.

3. Renesas Publications

Renesas provides the following documents for the Renesas Synergy Microcontroller. Before using any of these documents, visit renesassynergy.com/docs for the most up-to-date version of the document.

Component	Document type	Description
Microcontrollers	Datasheet	Features, overview, and electrical characteristics of the MCU
	User's Manual: Microcontrollers	MCU specifications such as pin assignments, memory maps, peripheral functions, electrical characteristics, timing diagrams, and operation descriptions
	Application Notes	Technical notes, board design guidelines, and software migration information
	Technical Update (TU)	Preliminary reports on product specifications such as restriction and errata
Software	Datasheet	Functional descriptions and specific performance data for software modules that are included in Renesas Synergy Software Package (SSP)
	User's Manual: Software	API reference including SSP architecture and programming information
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications
Tools & Kits, Solutions	User's Manual: Development Tools	User's manual and quick start guide for developing embedded software applications with Development Kit (DK), Starter Kit (SK), Promotion Kit (PK), Target Board Kit (TB), Product Examples (PE), and Application Examples (AE)
	User's Manual: Software	
	Quick Start Guide	
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications

4. Numbering Notation

The following numbering notation is used throughout this manual:

Example	Description
011b	Binary number. For example, the binary equivalent of the number 3 is 011b.
1Fh	Hexadecimal number. For example, the hexadecimal equivalent of the number 31 is described 1Fh. In some cases, a hexadecimal number is shown with the prefix 0x, based on C/C++ formatting.
1234	Decimal number. Decimal numbers are generally shown without a suffix.

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
ICU.NMICR.NMIMD	Periods separate a function module symbol (ICU), register symbol (NMICR), and bit field symbol (NMIMD)
ICU.NMICR	A period separates a function module symbol (ICU) and register symbol (NMICR)
NMICR.NMIMD	A period separates a register symbol (NMICR) and bit field symbol (NMIMD)
NFCLKSEL[1:0]	In a register bit name, the bit range enclosed in square brackets indicates the number of bits in the field at this location. In this example, NFCLKSEL[1:0] represents a 2-bit field at the specified location in the NMI Pin Interrupt Control Register (NMICR).

6. Unit Prefix

The following unit prefixes are sometimes misleading. Those unit prefixes are described throughout this manual with the following meaning:

Prefix	Description
b	Bit
B	Byte. This unit prefix is generally used for memory specification of the MCU and address space.
k	$1000 = 10^3$. k is also used to denote 1024 (2^{10}) but this unit prefix is used to denote 1000 (10^3) throughout this manual.
K	$1024 = 2^{10}$. This unit prefix is used to denote 1024 (2^{10}) not 1000 (10^3) throughout this manual.

7. Special Terms

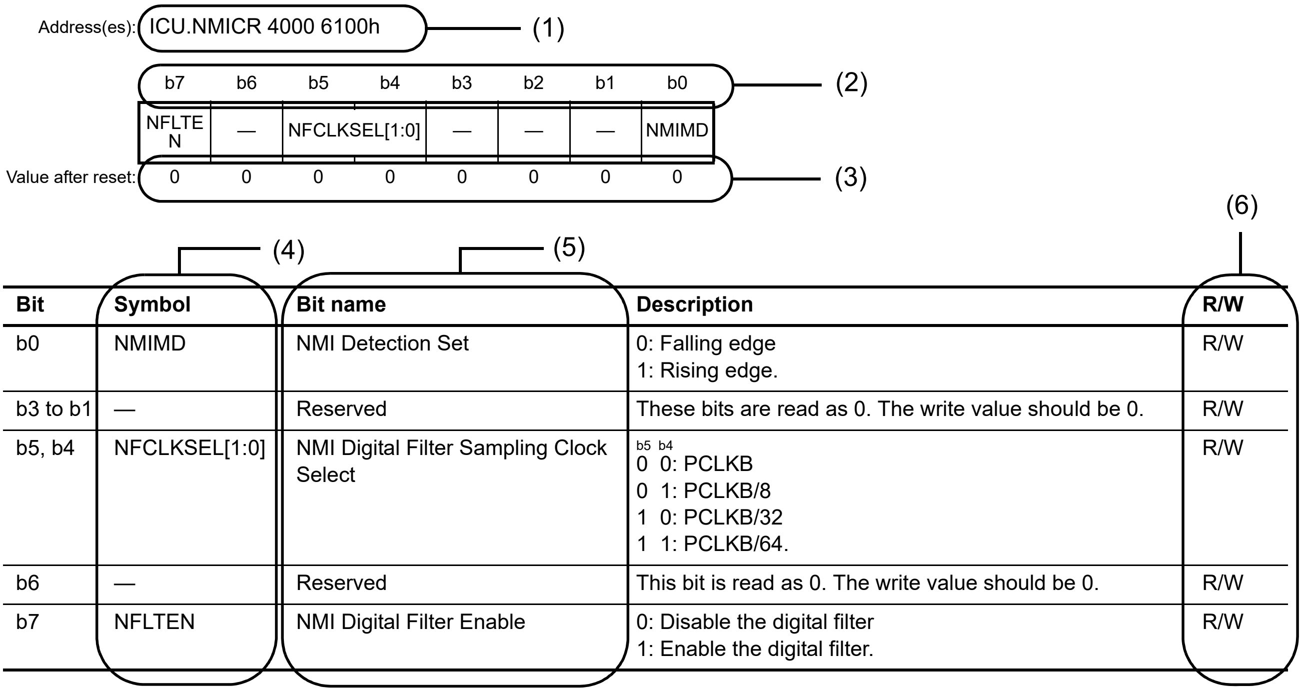
The following terms have special meanings:

Term	Description
NC	Not connected pin. NC means the pin is not connected to the MCU.
Hi-Z	High impedance

8. Register Description

Each register description includes both a register diagram that shows the bit assignments and a register bit table that describes the content of each bit. The example of symbols used in these tables are described in the sections that follow. The following is an example of a register description and associated bit field definition.

X.X.X NMI Pin Interrupt Control Register (NMICR)



Bit	Symbol	Bit name	Description	R/W
b0	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select	b5 b4 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	NFLTEN	NMI Digital Filter Enable	0: Disable the digital filter 1: Enable the digital filter.	R/W

(1) Function module symbol, register symbol, and address assignment

Function module symbol, register symbol, and address assignment of this register are generally expressed. ICU.NMICR 4000 6100h means NMI Pin Interrupt Control Register (NMICR) of Interrupt Controller Unit (ICU) is assigned to address 4000 6100h.

(2) Bit number

This number indicates the bit number. These bits are shown in order from b31 to b0 for a 32-bit register, from b15 to b0 for a 16-bit register, and from b7 to b0 for an 8-bit register.

(3) Value after reset

This symbol or number indicates the value of each bit after a reset. The value is shown in binary unless specified otherwise.
0: Indicates that the value is 0 after a reset.
1: Indicates that the value is 1 after a reset.
x: Indicates that the value is undefined after a reset.

(4) Bit symbol

Bit symbol indicates the short name of the bit field. Reserved bit is expressed with a —.

(5) Bit name

Bit name indicates the full name of the bit field.

(6) R/W

The R/W column indicates access type: whether the bit field is read or write.
R/W: The bit field is read and write.
R/(W): The bit field is read and write. But writing to this bit field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit field is read-only. Writing to this bit field has no effect.
W: The bit field is write-only. The read value is undefined.

9. Abbreviations

Abbreviations used in this manual are shown in the following table:

Abbreviation	Description
AES	Advanced Encryption Standard
AHB	Advanced High-Performance Bus
AHB-AP	AHB Access Port
APB	Advanced Peripheral Bus
ARC	Alleged RC
ATB	Advanced Trace Bus
BCD	Binary Coded Decimal
BSDL	Boundary Scan Description Language
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
FLL	Frequency Locked Loop
FPU	Floating-Point Unit
GSM	Global System for Mobile communications
HMI	Human Machine Interface
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NVIC	Nested Vector Interrupt Controller
PC	Program Counter
PFS	Port Function Select
PLL	Phase Locked Loop
POR	Power-On Reset
PWM	Pulse Width Modulation
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
S/H	Sample and Hold
SP	Stack Pointer
SWD	Serial Wire Debug
SW-DP	Serial Wire-Debug Port
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter

10. Proprietary Notice

All text, graphics, photographs, trademarks, logos, artwork and computer code, collectively known as content, contained in this document is owned, controlled or licensed by or to Renesas, and is protected by trade dress, copyright, patent and trademark laws, and other intellectual property rights and unfair competition laws. Except as expressly provided herein, no part of this document or content may be copied, reproduced, republished, posted, publicly displayed, encoded, translated, transmitted or distributed in any other medium for publication or distribution or for any commercial enterprise, without prior written consent from Renesas.

Arm® and Cortex® are registered trademarks of Arm Limited. CoreSight™ is a trademark of Arm Limited.

CoreMark® is a registered trademark of the Embedded Microprocessor Benchmark Consortium.

Magic Packet™ is a trademark of Advanced Micro Devices, Inc.

SuperFlash® is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Other brands and names mentioned in this document may be the trademarks or registered trademarks of their respective holders.

11. Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

12. Feedback on the Product

If you have any comments or suggestions about this product, go to renesassynergy.com/support.

13. Feedback on Content

If you have any comments on the document such as general suggestions for improvements, go to renesassynergy.com/support, and provide:

- The title of the Renesas Synergy document
- The document number
- If applicable, the page number(s) to which your comments refer
- A detailed explanation of your comments.

Contents

S5D9 Microcontroller Group

User’s Manual

Leading performance 120-MHz Arm® Cortex®-M4 core, up to 2-MB code flash memory, 640-KB SRAM, Graphics LCD Controller, 2D Drawing Engine, Capacitive Touch Sensing Unit, Ethernet MAC Controller with IEEE 1588 PTP, USB 2.0 High-Speed, USB 2.0 Full-Speed, SDHI, Quad SPI, security and safety features, and advanced analog.

Features

■ Arm Cortex-M4 Core with Floating Point Unit (FPU)

- Armv7E-M architecture with DSP instruction set
- Maximum operating frequency: 120 MHz
- Support for 4-GB address space
- On-chip debugging system: JTAG, SWD, and ETM
- Boundary scan and Arm Memory Protection Unit (Arm MPU)

■ Memory

- Up to 2-MB code flash memory (40 MHz zero wait states)
- 64-KB data flash memory (125,000 erase/write cycles)
- Up to 640-KB SRAM
- Flash Cache (FCACHE)
- Memory Protection Units (MPU)
- Memory Mirror Function (MMF)
- 128-bit unique ID

■ Connectivity

- Ethernet MAC Controller (ETHERC)
- Ethernet DMA Controller (EDMAC)
- Ethernet PTP Controller (EPTPC)
- USB 2.0 High-Speed (USBHS) module
  - On-chip transceiver with voltage regulator
  - Compliant with USB Battery Charging Specification 1.2
- USB 2.0 Full-Speed (USBFS) module
  - On-chip transceiver with voltage regulator
- Serial Communications Interface (SCI) with FIFO × 10
- Serial Peripheral Interface (SPI) × 2
- I^{2}C bus interface (IIC) × 3
- Controller Area Network (CAN) × 2
- Serial Sound Interface Enhanced (SSIE) × 2
- SD/MMC Host Interface (SDHI) × 2
- Quad Serial Peripheral Interface (QSPI)
- IrDA interface
- Sampling Rate Converter (SRC)
- External address space
  - 8-bit or 16-bit bus space is selectable per area
  - SDRAM support

■ Analog

- 12-bit A/D Converter (ADC12) with 3 sample-and-hold circuits each × 2
- 12-bit D/A Converter (DAC12) × 2
- High-Speed Analog Comparator (ACMPHS) × 6
- Programmable Gain Amplifier (PGA) × 6
- Temperature Sensor (TSN)

■ Timers

- General PWM Timer 32-bit Enhanced High Resolution (GPT32EH) × 4
- General PWM Timer 32-bit Enhanced (GPT32E) × 4
- General PWM Timer 32-bit (GPT32) × 6
- Asynchronous General-Purpose Timer (AGT) × 2
- Watchdog Timer (WDT)

■ Safety

- Error Correction Code (ECC) in SRAM
- SRAM parity error check
- Flash area protection
- ADC self-diagnosis function
- Clock Frequency Accuracy Measurement Circuit (CAC)
- Cyclic Redundancy Check (CRC) calculator
- Data Operation Circuit (DOC)
- Port Output Enable for GPT (POEG)
- Independent Watchdog Timer (IWDT)
- GPIO readback level detection
- Register write protection
- Main oscillator stop detection
- Illegal memory access

■ System and Power Management

- Low power modes
- Realtime Clock (RTC) with calendar and VBATT support
- Event Link Controller (ELC)
- DMA Controller (DMAC) × 8
- Data Transfer Controller (DTC)
- Key Interrupt Function (KINT)
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings

■ Security and Encryption

- AES128/192/256
- 3DES/ARC4
- SHA1/SHA224/SHA256/MD5
- GHASH
- RSA/DSA/ECC
- True Random Number Generator (TRNG)

■ Human Machine Interface (HMI)

- Graphics LCD Controller (GLCDC)
- JPEG codec
- 2D Drawing Engine (DRW)
- Capacitive Touch Sensing Unit (CTSU)
- Parallel Data Capture Unit (PDC)

■ Multiple Clock Sources

- Main clock oscillator (MOSC) (8 to 24 MHz)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO) (16/18/20 MHz)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- IWDT-dedicated on-chip oscillator (15 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- Clock out support

■ General-Purpose I/O Ports

- Up to 133 input/output pins
  - Up to 9 CMOS input
  - Up to 124 CMOS input/output
    - Up to 21 input/output 5 V tolerant
    - Up to 18 high current (20 mA)

■ Operating Voltage

- VCC: 2.7 to 3.6 V

■ Operating Temperature and Packages

- Ta = -40°C to +85°C
  - 176-pin BGA (13 mm × 13 mm, 0.8 mm pitch)
  - 145-pin LGA (7 mm × 7 mm, 0.5 mm pitch)
- Ta = -40°C to +105°C
  - 176-pin LQFP (24 mm × 24 mm, 0.5 mm pitch)
  - 144-pin LQFP (20 mm × 20 mm, 0.5 mm pitch)
  - 100-pin LQFP (14 mm × 14 mm, 0.5 mm pitch)

1. Overview

The MCU integrates multiple series of software- and pin-compatible Arm®-based 32-bit cores that share the same set of Renesas peripherals to facilitate design scalability and efficient platform-based product development.

The MCU in this series incorporates a high-performance Arm Cortex®-M4 core running up to 120 MHz, with the following features:

- Up to 2-MB code flash memory
- 640-KB SRAM
- Graphics LCD Controller (GLCDC)
- 2D Drawing Engine (DRW)
- Capacitive Touch Sensing Unit (CTSU)
- Ethernet MAC Controller (ETHERC) with IEEE 1588 PTP, USBFS, USBHS, SD/MMC Host Interface
- Quad Serial Peripheral Interface (QSPI)
- Security and safety features
- Analog peripherals.

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm Cortex-M4 core	• Maximum operating frequency: up to 120 MHz • Arm Cortex-M4 core: - Revision: r0p1-01rel0 - ARMv7E-M architecture profile - Single precision floating-point unit compliant with the ANSI/IEEE Std 754-2008. • Arm Memory Protection Unit (Arm MPU): - ARMv7 Protected Memory System Architecture - 8 protect regions. • SysTick timer: - Driven by SYSTICCLK (LOCO) or ICLK.

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 2-MB code flash memory. See section 55, Flash Memory.
Data flash memory	64-KB data flash memory. See section 55, Flash Memory.
Memory Mirror Function (MMF)	The Memory Mirror Function (MMF) can be configured to mirror the target application image load address in code flash memory to the application image link address in the 23-bit unused memory space (memory mirror space addresses). Your application code is developed and linked to run from this MMF destination address. The application code does not need to know the load location where it is stored in code flash memory. See section 5, Memory Mirror Function (MMF).
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 7, Option-Setting Memory.
SRAM	On-chip high-speed SRAM with either parity-bit or Error Correction Code (ECC). The first 32 KB in SRAM0 provides error correction capability using ECC. Parity check is performed for other areas. See section 53, SRAM.
Standby SRAM	On-chip SRAM that can retain data in Deep Software Standby mode. See section 54, Standby SRAM.

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Two operating modes: - Single-chip mode - SCI or USB boot mode. See section 3, Operating Modes.
Resets	14 resets: • RES pin reset • Power-on reset • Voltage monitor 0 reset • Voltage monitor 1 reset • Voltage monitor 2 reset • Independent watchdog timer reset • Watchdog timer reset • Deep software standby reset • SRAM parity error reset • SRAM ECC error reset • Bus master MPU error reset • Bus slave MPU error reset • Stack pointer error reset • Software reset. See section 6, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) function monitors the voltage level input to the VCC pin, and the detection level can be selected using a software program. See section 8, Low Voltage Detection (LVD).
Clocks	• Main clock oscillator (MOSC) • Sub-clock oscillator (SOSC) • High-speed on-chip oscillator (HOCO) • Middle-speed on-chip oscillator (MOCO) • Low-speed on-chip oscillator (LOCO) • PLL frequency synthesizer • IWDT-dedicated on-chip oscillator • Clock out support. See section 9, Clock Generation Circuit.
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See section 10, Clock Frequency Accuracy Measurement Circuit (CAC).
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the NVIC/DTC module and DMAC module. The ICU also controls NMI interrupts. See section 14, Interrupt Controller Unit (ICU).
Key Interrupt Function (KINT)	A key interrupt can be generated by setting the Key Return Mode Register (KRM) and inputting a rising or falling edge to the key interrupt input pins. See section 21, Key Interrupt Function (KINT).
Low power modes	Power consumption can be reduced in multiple ways, such as by setting clock dividers, controlling EBCLK output, controlling SDCLK output, stopping modules, selecting power control mode in normal operation, and transitioning to low power modes. See section 11, Low Power Modes.
Battery backup function	A battery backup function is provided for partial powering by a battery. The battery-powered area includes the RTC, SOSC, backup memory, and switch between VCC and VBATT. See section 12, Battery Backup Function.
Register write protection	The register write protection function protects important registers from being overwritten because of software errors. See section 13, Register Write Protection.
Memory Protection Unit (MPU)	Four Memory Protection Units (MPUs) and a CPU stack pointer monitor function are provided for memory protection. See section 16, Memory Protection Unit (MPU).

Table 1.3 System (2 of 2)

Feature	Functional description
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down-counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, a non-maskable interrupt or interrupt can be generated by an underflow. A refresh-permitted period can be set to refresh the counter and be used as the condition for detecting when the system runs out of control. See section 27, Watchdog Timer (WDT).
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) consists of a 14-bit down-counter that must be serviced periodically to prevent counter underflow. It can be used to reset the MCU or to generate a non-maskable interrupt or interrupt for a timer underflow. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail safe mechanism when the system runs out of control. The IWDT can be triggered automatically on a reset, underflow, refresh error, or by a refresh of the count value in the registers. See section 28, Independent Watchdog Timer (IWDT).

Table 1.4 Event link

Feature	Functional description
Event Link Controller (ELC)	The Event Link Controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals to connect them to different modules, enabling direct interaction between the modules without CPU intervention. See section 19, Event Link Controller (ELC).

Table 1.5 Direct memory access

Feature	Functional description
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request. See section 18, Data Transfer Controller (DTC).
DMA Controller (DMAC)	An 8-channel DMA Controller (DMAC) module is provided for transferring data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. See section 17, DMA Controller (DMAC).

Table 1.6 External bus interface

Feature	Functional description
External buses	• CS area (EXBIU): Connected to the external devices (external memory interface) • SDRAM area (EXBIU): Connected to the SDRAM (external memory interface) • QSPI area (EXBIUT2): Connected to the QSPI (external device interface).

Table 1.7 Timers

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with 14 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer. See section 23, General PWM Timer (GPT).
Port Output Enable for GPT (POEG)	Use the Port Output Enable for GPT (POEG) function to place the General PWM Timer (GPT) output pins in the output disable state. See section 22, Port Output Enable for GPT (POEG).
Asynchronous General-Purpose Timer (AGT)	The Asynchronous General-Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting of external events. This 16-bit timer consists of a reload register and a down-counter. The reload register and the down-counter are allocated to the same address, and can be accessed with the AGT register. See section 25, Asynchronous General-Purpose Timer (AGT).
Realtime Clock (RTC)	The Realtime Clock (RTC) has two counting modes, calendar count mode and binary count mode, that are controlled by the register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar. See section 26, Realtime Clock (RTC).

Table 1.8 Communication interfaces (1 of 2)

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communications Interface (SCI) is configurable to five asynchronous and synchronous serial interfaces: • Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA)) • 8-bit clock synchronous interface • Simple IIC (master-only) • Simple SPI • Smart card interface. The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. Each SCI has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator. See section 34, Serial Communications Interface (SCI).
IrDA interface	The IrDA interface sends and receives IrDA data communication waveforms in cooperation with the SCI1 based on the IrDA (Infrared Data Association) standard 1.0. See section 35, IrDA Interface.
I^{2}C bus interface (IIC)	The 3-channel I^{2}C bus interface (IIC) conforms with and provides a subset of the NXP I^{2}C (Inter-Integrated Circuit) bus interface functions. See section 36, I^{2}C Bus Interface (IIC).
Serial Peripheral Interface (SPI)	Two independent Serial Peripheral Interface (SPI) channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices. See section 38, Serial Peripheral Interface (SPI).
Serial Sound Interface Enhanced (SSIE)	The Serial Sound Interface Enhanced (SSIE) peripheral provides functionality to interface with digital audio devices for transmitting I^{2}S 2ch, 4ch, 6ch, 8ch, WS Continue/Monaural/TDM audio data over a serial bus. The SSIE supports an audio clock frequency of up to 50 MHz, and can be operated as a slave or master receiver, transmitter, or transceiver to suit various applications. The SSIE includes 32-stage FIFO buffers in the receiver and transmitter, and supports interrupts and DMA-driven data reception and transmission. See section 41, Serial Sound Interface Enhanced (SSIE).
Quad Serial Peripheral Interface (QSPI)	The Quad Serial Peripheral Interface (QSPI) is a memory controller for connecting a serial ROM (nonvolatile memory such as a serial flash memory, serial EEPROM, or serial FeRAM) that has an SPI-compatible interface. See section 39, Quad Serial Peripheral Interface (QSPI).
Controller Area Network (CAN) module	The Controller Area Network (CAN) module provides functionality to receive and transmit data using a message-based protocol between multiple slaves and masters in electromagnetically-noisy applications. The CAN module complies with the ISO 11898-1 (CAN 2.0A/CAN 2.0B) standard and supports up to 32 mailboxes, which can be configured for transmission or reception in normal mailbox and FIFO modes. Both standard (11-bit) and extended (29-bit) messaging formats are supported. See section 37, Controller Area Network (CAN) Module.
USB 2.0 Full-Speed (USBFS) module	The USB 2.0 Full-Speed (USBFS) module can operate as a host controller or device controller. The module supports full-speed and low-speed (host controller only) transfer as defined in Universal Serial Bus Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in the Universal Serial Bus Specification 2.0. The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on the peripheral devices used for communication or based on your system. See section 32, USB 2.0 Full-Speed Module (USBFS).
USB 2.0 High-Speed (USBHS) module	The USB 2.0 High-Speed (USBHS) module can operate as a host controller or a device controller. As a host controller, the USBHS supports high-speed transfer, full-speed transfer, and low-speed transfer as defined in the Universal Serial Bus Specification 2.0. As a device controller, the USBHS supports high-speed transfer and full-speed transfer as defined in the Universal Serial Bus Specification 2.0. The USBHS has an internal USB transceiver and supports all of the transfer types defined in the Universal Serial Bus Specification 2.0. The USBHS has FIFO buffers for data transfer, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or your system for communication. See section 33, USB 2.0 High-Speed Module (USBHS).

Table 1.8 Communication interfaces (2 of 2)

Feature	Functional description
Ethernet MAC with IEEE 1588 PTP (ETHERC)	One-channel Ethernet MAC Controller (ETHERC) compliant with the Ethernet/IEEE802.3 Media Access Control (MAC) layer protocol. An ETHERC channel provides one channel of the MAC layer interface, connecting the MCU to the physical layer LSI (PHY-LSI) that allows transmission and reception of frames compliant with the Ethernet and IEEE802.3 standards. The ETHERC is connected to the Ethernet DMA Controller (EDMAC) so data can be transferred without using the CPU. To handle timing and synchronization between devices, an on-chip Precision Time Protocol (PTP) module for the Ethernet PTP Controller (EPTPC) applies the PTP defined in the IEEE 1588-2008 version 2.0 standard. The EPTPC is composed of: • Synchronization Frame Processing unit (SYNFP0) • A Statistical Time Correction Algorithm unit (STCA). Use the EPTPC in combination with the on-chip Ethernet MAC Controller (ETHERC) and the DMA Controller for the PTP Ethernet Controller (PTPEDMAC). See section 29, Ethernet MAC Controller (ETHERC).
SD/MMC Host Interface (SDHI)	The SDHI and MultiMediaCard (MMC) interface module provides the functionality required to connect a variety of external memory cards to the MCU. The SDHI supports both 1-bit and 4-bit buses for connecting memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA). The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84-B451) device access. This interface also provides backward compatibility and supports high-speed SDR transfer modes. See section 43, SD/MMC Host Interface (SDHI).

Table 1.9 Analog

Feature	Functional description
12-bit A/D Converter (ADC12)	Up to two successive approximation 12-bit A/D Converters (ADC12) are provided. In unit 0, up to 13 analog input channels are selectable. In unit 1, up to 11 analog input channels, the temperature sensor output, and an internal reference voltage are selectable for conversion. The A/D conversion accuracy is selectable from 12-bit, 10-bit, and 8-bit conversion, making it possible to optimize the tradeoff between speed and resolution in generating a digital value. See section 47, 12-Bit A/D Converter (ADC12).
12-bit D/A Converter (DAC12)	The 12-bit D/A Converter (DAC12) converts data and includes an output amplifier. See section 48, 12-Bit D/A Converter (DAC12).
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is linear. The output voltage is provided to the ADC12 for conversion and can also be used by the end application. See section 49, Temperature Sensor (TSN).
High-Speed Analog Comparator (ACMPHS)	The High-Speed Analog Comparator (ACMPHS) compares a test voltage with a reference voltage and provides a digital output based on the conversion result. Both the test and reference voltages can be provided to the comparator from internal sources such as the DAC12 output and internal reference voltage, and an external source with or without an internal PGA. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion. See section 50, High-Speed Analog Comparator (ACMPHS).

Table 1.10 Human machine interfaces

Feature	Functional description
Capacitive Touch Sensing Unit (CTSU)	The Capacitive Touch Sensing Unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with an electrical insulator so that fingers do not come into direct contact with the electrodes. See section 51, Capacitive Touch Sensing Unit (CTSU).

Table 1.11 Graphics

Feature	Functional description
Graphics LCD Controller (GLCDC)	The Graphics LCD Controller (GLCDC) provides multiple functions and supports various data formats and panels. Key GLCDC features include: • GPX bus master function for accessing graphics data • Superimposition of three planes (single-color background plane, graphic 1-plane, and graphic 2-plane) • Support for many types of 32-bit or 16-bit per pixel graphics data and 8-bit, 4-bit, or 1-bit LUT data format • Digital interface signal output supporting a video image size of WVGA or greater. See section 58, Graphics LCD Controller (GLCDC).
2D Drawing Engine (DRW)	The 2D Drawing Engine (DRW) provides flexible functions that can support almost any object geometry rather than being bound to only a few specific geometries such as lines, triangles, or circles. The edges of every object can be independently blurred or antialiased. Rasterization is executed at one pixel per clock on the bounding box of the object from left to right and top to bottom. The DRW can also raster from bottom to top to optimize the performance in certain cases. In addition, optimization methods are available to avoid rasterization of many empty pixels of the bounding box. The distances to the edges of the object are calculated by a set of edge equations for every pixel of the bounding box. These edge equations can be combined to describe the entire object. If a pixel is inside the object, it is selected for rendering. If it is outside, it is discarded. If it is on the edge, an alpha value can be chosen proportional to the distance of the pixel to the nearest edge for antialiasing. Every pixel that is selected for rendering can be textured. The resulting aRGB quadruple can be modified by a general raster operation approach independently for each of the four channels. The aRGB quadruples can then be blended with one of the multiple blend modes of the DRW. The DRW provides two inputs (texture read and framebuffer read), and one output (framebuffer write). The internal color format is always aRGB (8888). The color formats from the inputs are converted to the internal format on read and a conversion back is made on write. See section 56, 2D Drawing Engine (DRW).
JPEG codec	The JPEG incorporates a JPEG codec that conforms to the JPEG baseline compression and decompression standard. This provides high-speed compression of image data and high-speed decoding of JPEG data. See section 57, JPEG Codec (JPEG).
Parallel Data Capture (PDC) unit	One Parallel Data Capture (PDC) unit is provided for communicating with external I/O devices, including image sensors, and transferring parallel data, such as an image output from the external I/O device through the DTC or DMAC to the on-chip SRAM and external address spaces (the CS and SDRAM areas). See section 44, Parallel Data Capture Unit (PDC).

Table 1.12 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generating polynomials are available. The snoop function allows monitoring reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer. See section 40, Cyclic Redundancy Check (CRC) Calculator.
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. See section 52, Data Operation Circuit (DOC).
Sampling Rate Converter (SRC)	The Sampling Rate Converter (SRC) converts the sampling rate of data produced by various audio decoders, such as the WMA, MP3, and AAC. Both 16-bit stereo and monaural data are supported. See section 42, Sampling Rate Converter (SRC).

Table 1.13 Security

Feature	Functional description
Secure Crypto Engine 7 (SCE7)	• Security algorithms: - Symmetric algorithms: AES, 3DES, and ARC4 - Asymmetric algorithms: RSA, DSA, and ECC. • Other support features: - TRNG (True Random Number Generator) - Hash-value generation: SHA1, SHA224, SHA256, GHASH, and MD5 - 128-bit unique ID. See section 46, Secure Cryptographic Engine (SCE7).

1.2 Block Diagram

Figure 1.1 shows a block diagram of the MCU superset, some individual devices within the group have a subset of the features.

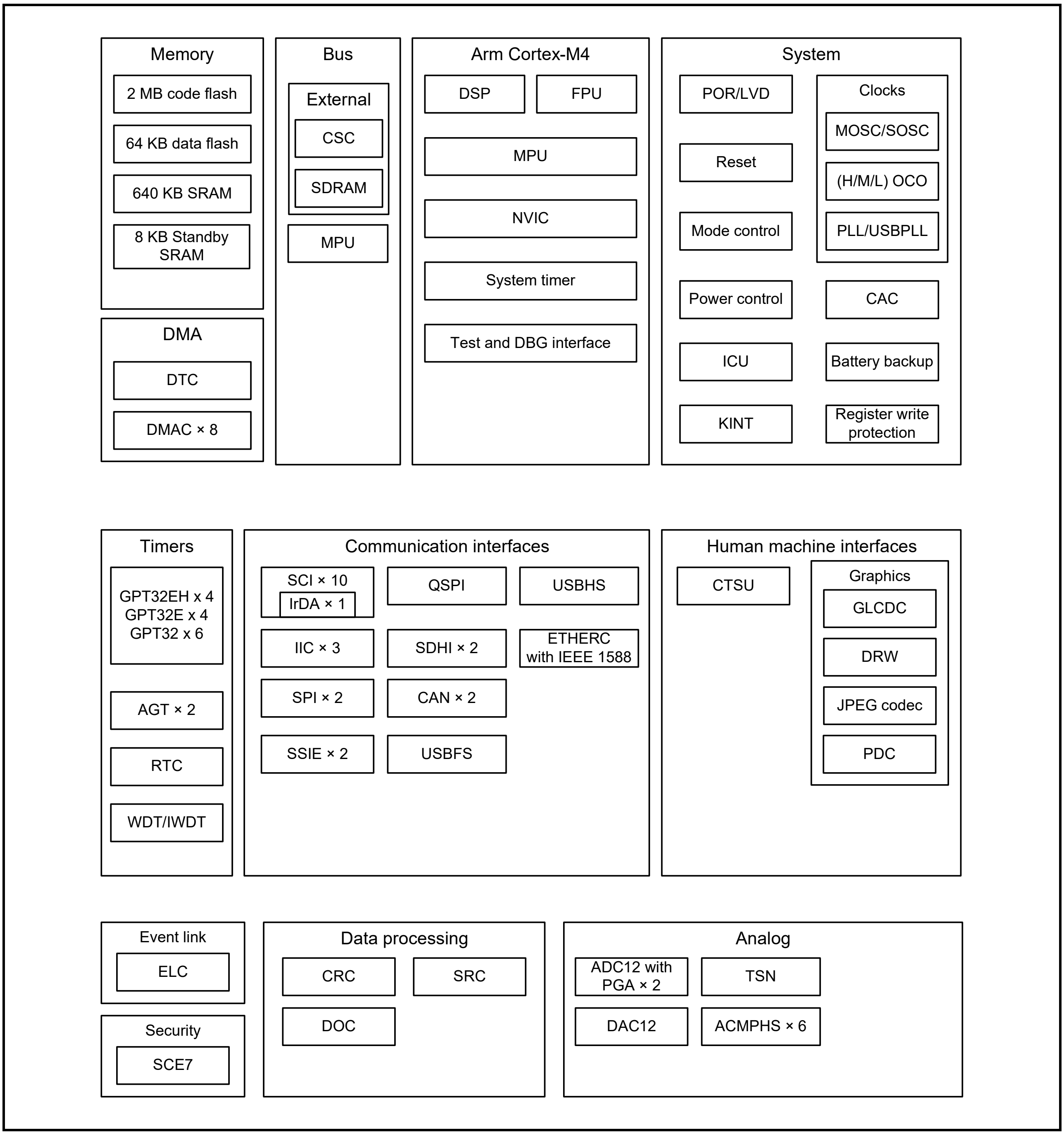


Figure 1.1 Block diagram

1.3 Part Numbering

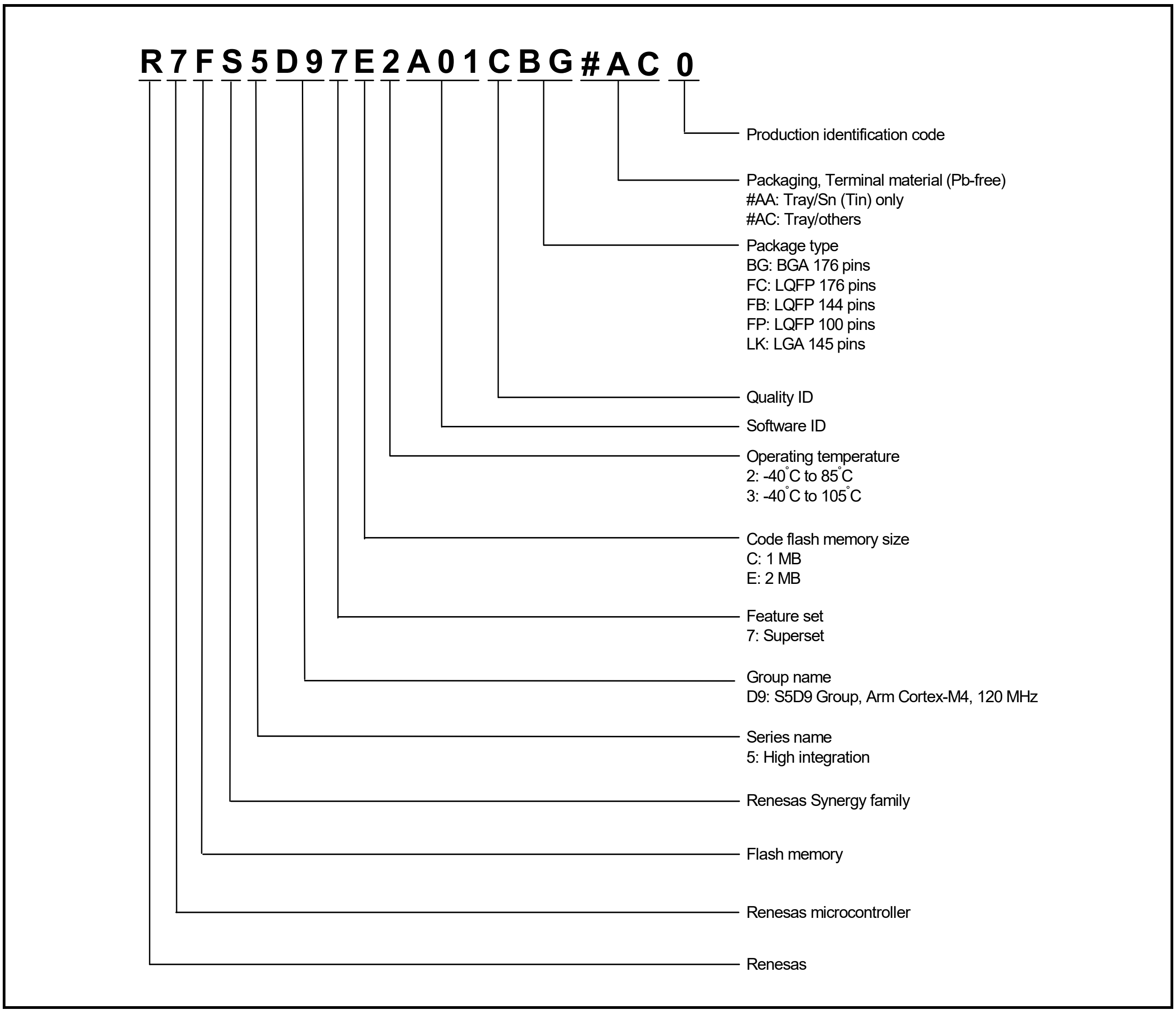


Figure 1.2 Part numbering scheme

Table 1.14 Product list

Product part number	Orderable part number	Package code	Code flash	Data flash	SRAM	Operating temperature
R7FS5D97E2A01CBG	R7FS5D97E2A01CBG#AC0	PLBG0176GE-A	2 MB	64 KB	640 KB	-40 to +85°C
R7FS5D97E3A01CFC	R7FS5D97E3A01CFC#AA0	PLQP0176KB-A				-40 to +105°C
R7FS5D97E2A01CLK	R7FS5D97E2A01CLK#AC0	PTLG0145KA-A				-40 to +85°C
R7FS5D97E3A01CFB	R7FS5D97E3A01CFB#AA0	PLQP0144KA-B				-40 to +105°C
R7FS5D97E3A01CFP	R7FS5D97E3A01CFP#AA0	PLQP0100KB-B				-40 to +105°C
R7FS5D97C2A01CBG	R7FS5D97C2A01CBG#AC0	PLBG0176GE-A	1 MB			-40 to +85°C
R7FS5D97C3A01CFC	R7FS5D97C3A01CFC#AA0	PLQP0176KB-A				-40 to +105°C
R7FS5D97C2A01CLK	R7FS5D97C2A01CLK#AC0	PTLG0145KA-A				-40 to +85°C
R7FS5D97C3A01CFB	R7FS5D97C3A01CFB#AA0	PLQP0144KA-B				-40 to +105°C
R7FS5D97C3A01CFP	R7FS5D97C3A01CFP#AA0	PLQP0100KB-B				-40 to +105°C

1.4 Function Comparison

Table 1.15 Functional comparison (Graphics)

Function			Part numbers				
			R7FS5D97E2XXXCBG/ R7FS5D97C2XXXCBG	R7FS5D97E3XXXCFC/ R7FS5D97C3XXXCFC	R7FS5D97E2XXXCLK/ R7FS5D97C2XXXCLK	R7FS5D97E3XXXCFB/ R7FS5D97C3XXXCFB	R7FS5D97E3XXXCFP/ R7FS5D97C3XXXCFP
Pin count			176	176	145	144	100
Package			BGA	LQFP	LGA	LQFP	LQFP
Code flash memory			2/1 MB				
Data flash memory			64 KB				
SRAM			640 KB				
		Parity	608 KB				
		ECC	32 KB				
Standby SRAM			8 KB				
System		CPU clock	120 MHz				
		Backup registers	512 B				
		ICU	Yes				
		KINT	8				
Event link		ELC	Yes				
DMA		DTC	Yes				
		DMAC	8				
BUS		External bus	16-bit bus				8-bit bus
		SDRAM	Yes				No
Timers		GPT32EH	4	4	4	4	4
		GPT32E	4	4	4	4	4
		GPT32	6	6	6	6	5
		AGT	2	2	2	2	2
		RTC	Yes				
		WDT/IWDT	Yes				
Communication		SCI	10				
		IIC	3				2
		SPI	2				
		SSIE	2				1
		QSPI	1				
		SDHI	2				
		CAN	2				
		USBFS	Yes				
		USBHS	Yes		No		
		ETHERC	1				
Analog		ADC12	24		22		19
		DAC12	2				
		ACMPHS	6				
		TSN	Yes				
HMI		CTSU	13		18		12
	Graphics	GLCDC	RGB888				
		DRW	Yes				
		JPEG	Yes				
		PDC	Yes				
Data processing		CRC	Yes				
		DOC	Yes				
		SRC	Yes				
Security			SCE7				

1.5 Pin Functions

Table 1.16 Pin functions (1 of 5)

Function	Signal	I/O	Description
Power supply	VCC	Input	Digital voltage supply pin. This is used as the digital power supply for the respective modules and internal voltage regulator, and used to monitor the voltage of the POR/LVD. Connect to the system power supply. Connect to VSS through a 0.1-μF smoothing capacitor close to each VCC pin.
	VCL0	-	Connect to VSS through a 0.1-μF smoothing capacitor close to each VCL pin. Stabilize the internal power supply.
	VCL	-	
	VSS	Input	Ground pin. Connect to the system power supply (0 V).
	VBATT	Input	Backup power pin
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCOUT	Output	
	EBCLK	Output	Outputs the external bus clock for external devices
	SDCLK	Output	Outputs the SDRAM-dedicated clock
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Measurement reference clock input pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ15	Input	Maskable interrupt request pins
KINT	KR00 to KR07	Input	A key interrupt can be generated by inputting a falling edge to the key interrupt input pins
On-chip emulator	TMS	I/O	On-chip emulator or boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TCLK	Output	This pin outputs the clock for synchronization with the trace data
	TDATA0 to TDATA3	Output	Trace data output
	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
	SWO	Output	Serial wire trace output pin
External bus interface	RD	Output	Strobe signal indicating that reading from the external bus interface space is in progress, active low
	WR	Output	Strobe signal indicating that writing to the external bus interface space is in progress, in 1-write strobe mode, active low
	WR0 to WR1	Output	Strobe signals indicating that either group of data bus pins (D07 to D00 or D15 to D08) is valid in writing to the external bus interface space, in byte strobe mode, active low
	BC0 to BC1	Output	Strobe signals indicating that either group of data bus pins (D07 to D00 or D15 to D08) is valid in access to the external bus interface space, in 1-write strobe mode, active low
	ALE	Output	Address latch signal when address/data multiplexed bus is selected
	WAIT	Input	Input pin for wait request signals in access to the external space, active low
	CS0 to CS7	Output	Select signals for CS areas, active low
	A00 to A23	Output	Address bus
	D00 to D15	I/O	Data bus
	A00/D00 to A15/D15	I/O	Address/data multiplexed bus

Table 1.16 Pin functions (2 of 5)

Function	Signal	I/O	Description
SDRAM interface	CKE	Output	SDRAM clock enable signal
	SDCS	Output	SDRAM chip select signal, active low
	RAS	Output	SDRAM low address strobe signal, active low
	CAS	Output	SDRAM column address strobe signal, active low
	WE	Output	SDRAM write enable signal, active low
	DQM0	Output	SDRAM I/O data mask enable signal for DQ07 to DQ00
	DQM1	Output	SDRAM I/O data mask enable signal for DQ15 to DQ08
	A00 to A15	Output	Address bus
	DQ00 to DQ15	I/O	Data bus
GPT	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOC0A to GTIOC13A, GTIOC0B to GTIOC13B	I/O	Input capture, output compare, or PWM output pins
	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W phase)
AGT	AGTEE0, AGTEE1	Input	External event input enable signals
	AGTIO0, AGTIO1	I/O	External event input and pulse output pins
	AGTO0, AGTO1	Output	Pulse output pins
	AGTOA0, AGTOA1	Output	Output compare match A output pins
	AGTOB0, AGTOB1	Output	Output compare match B output pins
RTC	RTCOUT	Output	Output pin for 1-Hz or 64-Hz clock
	RTCIC0 to RTCIC2	Input	Time capture event input pins
SCI	SCK0 to SCK9	I/O	Input/output pins for the clock (clock synchronous mode)
	RXD0 to RXD9	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXD0 to TXD9	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTS0_RTS0 to CTS9_RTS9	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active low
	SCL0 to SCL9	I/O	Input/output pins for the I^2C clock (simple IIC mode)
	SDA0 to SDA9	I/O	Input/output pins for the I^2C data (simple IIC mode)
	SCK0 to SCK9	I/O	Input/output pins for the clock (simple SPI mode)
	MISO0 to MISO9	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSI0 to MOSI9	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SS0 to SS9	Input	Chip-select input pins (simple SPI mode), active low
IIC	SCL0 to SCL2	I/O	Input/output pins for the clock
	SDA0 to SDA2	I/O	Input/output pins for data
SSIE	SSIBCK0	I/O	SSIE serial bit clock pins
	SSIBCK1		
	SSILRCK0/SSIFS0	I/O	LR clock/frame synchronization pins
	SSILRCK1/SSIFS1		
	SSITXD0	Output	Serial data output pins
	SSIRXD0	Input	Serial data input pins
	SSIDATA1	I/O	Serial data input/output pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)

Table 1.16 Pin functions (3 of 5)

Function	Signal	I/O	Description
SPI	RSPCKA, RSPCKB	I/O	Clock input/output pin
	MOSIA, MOSIB	I/O	Input or output pins for data output from the master
	MISOA, MISOB	I/O	Input or output pins for data output from the slave
	SSLA0, SSLB0	I/O	Input or output pin for slave selection
	SSLA1 to SSLA3, SSLB1 to SSLB3	Output	Output pins for slave selection
QSPI	QSPCLK	Output	QSPI clock output pin
	QSSL	Output	QSPI slave output pin
	QIO0 to QIO3	I/O	Data0 to Data3
CAN	CRX0, CRX1	Input	Receive data
	CTX0, CTX1	Output	Transmit data
USBFS	VCC_USB	Input	Power supply pins
	VSS_USB	Input	Ground pins
	USB_DP	I/O	D+ I/O pin of the USB on-chip transceiver. Connect this pin to the D+ pin of the USB bus
	USB_DM	I/O	D- I/O pin of the USB on-chip transceiver. Connect this pin to the D- pin of the USB bus
	USB_VBUS	Input	USB cable connection monitor pin. Connect this pin to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB_OVRCURA, USB_OVRCURB	Input	Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_ID	Input	Connect the MicroAB connector ID input signal to this pin during operation in OTG mode
USBHS	VCC_USBHS	Input	Power supply pin
	VSS1_USBHS	Input	Ground pin
	VSS2_USBHS	Input	Ground pin
	AVCC_USBHS	Input	Analog power supply pin for the USBHS
	AVSS_USBHS	Input	Analog ground pin for the USBHS. Must be shorted to the PVSS_USBHS pin
	PVSS_USBHS	Input	PLL circuit ground pin for the USBHS. Must be shorted to the AVSS_USBHS pin
	USBHS_RREF	I/O	USBHS reference current source pin. Connect this pin to the AVSS_USBHS pin through a 2.2-kΩ resistor (±1%)
	USBHS_DP	I/O	USB bus D+ data pin
	USBHS_DM	I/O	USB bus D- data pin
	USBHS_EXICEN	Output	Connect this pin to the OTG power supply IC
	USBHS_ID	Input	Connect this pin to the OTG power supply IC
	USBHS_VBUSEN	Output	VBUS power enable signal for USB
	USBHS_OVRCURA, USBHS_OVRCURB	Input	Overcurrent pin for USB
	USBHS_VBUS	Input	USB cable connection monitor input pin

Table 1.16 Pin functions (4 of 5)

Function	Signal	I/O	Description
ETHERC	REF50CK0	Input	50-MHz reference clock. This pin inputs reference signal for transmission/reception timing in RMII mode.
	RMII0_CRS_DV	Input	Indicates carrier detection signals and valid receive data on RMII0_RXD1 and RMII0_RXD0 in RMII mode
	RMII0_TXD0, RMII0_TXD1	Output	2-bit transmit data in RMII mode
	RMII0_RXD0, RMII0_RXD1	Input	2-bit receive data in RMII mode
	RMII0_TXD_EN	Output	Output pin for data transmit enable signal in RMII mode
	RMII0_RX_ER	Input	Indicates an error occurred during reception of data in RMII mode
	ET0_CRS	Input	Carrier detection/data reception enable signal
	ET0_RX_DV	Input	Indicates valid receive data on ET0_ERXD3 to ET0_ERXD0
	ET0_EXOUT	Output	General-purpose external output pin
	ET0_LINKSTA	Input	Input link status from the PHY-LSI
	ET0_ETXD0 to ET0_ETXD3	Output	4 bits of MII transmit data
	ET0_ERXD0 to ET0_ERXD3	Input	4 bits of MII receive data
	ET0_TX_EN	Output	Transmit enable signal. Functions as signal indicating that transmit data is ready on ET0_ETXD3 to ET0_ETXD0
	ET0_TX_ER	Output	Transmit error pin. Functions as signal notifying the PHY_LSI of an error during transmission
	ET0_RX_ER	Input	Receive error pin. Functions as signal to recognize an error during reception
	ET0_TX_CLK	Input	Transmit clock pin. This pin inputs reference signal for output timing from ET0_TX_EN, ET0_ETXD3 to ET0_ETXD0, and ET0_TX_ER
	ET0_RX_CLK	Input	Receive clock pin. This pin inputs reference signal for input timing to ET0_RX_DV, ET0_ERXD3 to ET0_ERXD0, and ET0_RX_ER
	ET0_COL	Input	Input collision detection signal
	ET0_WOL	Output	Receive Magic packets
	ET0_MDC	Output	Output reference clock signal for information transfer through ET0_MDIO.
	ET0_MDIO	I/O	Input or output bidirectional signal for exchange of management data with PHY-LSI
SDHI	SD0CLK, SD1CLK	Output	SD clock output pins
	SD0CMD, SD1CMD	I/O	Command output pin and response input signal pins
	SD0DAT0 to SD0DAT7, SD1DAT0 to SD1DAT7	I/O	SD and MMC data bus pins
	SD0CD, SD1CD	Input	SD card detection pins
	SD0WP, SD1WP	Input	SD write-protect signals
Analog power supply	AVCC0	Input	Analog voltage supply pin. This is used as the analog power supply for the respective modules. Supply this pin with the same voltage as the VCC pin.
	AVSS0	Input	Analog ground pin. This is used as the analog ground for the respective modules. Supply this pin with the same voltage as the VSS pin.
	VREFH0	Input	Analog reference voltage supply pin for the ADC12 (unit 0). Connect this pin to VCC when not using the ADC12 (unit 0) and sample-and-hold circuit for AN000 to AN002.
	VREFL0	Input	Analog reference ground pin for the ADC12. Connect this pin to VSS when not using the ADC12 (unit 0) and sample-and-hold circuit for AN000 to AN002
	VREFH	Input	Analog reference voltage supply pin for the ADC12 (unit 1) and D/A Converter. Connect this pin to VCC when not using the ADC12 (unit 1), sample-and-hold circuit for AN100 to AN102, and D/A Converter.
	VREFL	Input	Analog reference ground pin for the ADC12 and D/A Converter. Connect this pin to VSS when not using the ADC12 (unit 1), sample-and-hold circuit for AN100 to AN102, and D/A Converter.

Table 1.16 Pin functions (5 of 5)

Function	Signal	I/O	Description
ADC12	AN000 to AN007, AN016 to AN020	Input	Input pins for the analog signals to be processed by the ADC12
	AN100 to AN103, AN105 to AN107, AN116 to AN119	Input	
	ADTRG0	Input	Input pins for the external trigger signals that start the A/D conversion
	ADTRG1	Input	
	PGAVSS000/PGAVSS100	Input	Differential input pins
DAC12	DA0, DA1	Output	Output pins for the analog signals processed by the D/A converter
ACMPHS	VCOUT	Output	Comparator output pin
	IVREF0 to IVREF3	Input	Reference voltage input pins for comparator
	IVCMP0 to IVCMP2	Input	Analog voltage input pins for comparator
CTSU	TS00 to TS17	Input	Capacitive touch detection pins (touch pins)
	TSCAP	-	Secondary power supply pin for the touch driver
I/O ports	P000 to P007	Input	General-purpose input pins
	P008 to P010, P014, P015	I/O	General-purpose input/output pins
	P100 to P115	I/O	General-purpose input/output pins
	P200	Input	General-purpose input pin
	P201 to P214	I/O	General-purpose input/output pins
	P300 to P315	I/O	General-purpose input/output pins
	P400 to P415	I/O	General-purpose input/output pins
	P500 to P508, P511 to P513	I/O	General-purpose input/output pins
	P600 to P615	I/O	General-purpose input/output pins
	P700 to P713	I/O	General-purpose input/output pins
	P800 to P806	I/O	General-purpose input/output pins
	P900, P901, P905 to P908	I/O	General-purpose input/output pins
	PA00, PA01, PA08 to PA10	I/O	General-purpose input/output pins
	PB00, PB01	I/O	General-purpose input/output pins
GLCDC	LCD_DATA23 to LCD_DATA00	Output	Data output pins for panel
	LCD_TCON3 to LCD_TCON0	Output	Output pins for panel timing adjustment
	LCD_CLK	Output	Panel clock output pin
	LCD_EXTCLK	Input	Panel clock source input pin
PDC	PIXCLK	Input	Image transfer clock pin
	VSYNC	Input	Vertical synchronization signal pin
	HSYNC	Input	Horizontal synchronization signal pin
	PIXD0 to PIXD7	Input	8-bit image data pins
	PCKO	Output	Output pin for dot clock

1.6 Pin Assignments

Figure 1.3 to Figure 1.7 show the pin assignments.

R7FS5D9XX2XXXCBG

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
15	P407	P409	P411	P414	P708	USBHS_DM	PVSS_USBHS	P212/EXTAL	XCIN	VCL0	P707	P703	P700	P405	P401	15
14	USB_DP	USB_DM	P410	P412	P415	USBHS_DP	AVSS_USBHS	P213/XTAL	XCOUT	VBATT	P706	P701	P406	P402	P512	14
13	P204	VCC_USB	VSS_USB	P408	P413	VCC_USBHS	USBHS_RREF	AVCC_USBHS	VSS	PB01	P704	P404	P400	P511	P805	13
12	P313	P202	P207	P206	P205	VSS1_USBHS	VSS2_USBHS	VCC	PB00	P705	P702	P403	P513	P806	P000	12
11	P900	P315	P314	P203								VCC	P001	P004	P002	11
10	P214	P211	P901	VSS								VSS	P006	P008	P005	10
9	P210	P209	RES	VCC								P009	AVSS0	VREFL0	VREFH0	9
8	P208	P201/MD	P200	P908								P010	AVCC0	VREFL	VREFH	8
7	P906	P905	P312	P907								VCC	VSS	P015	P014	7
6	P310	P309	P307	P311								P007	P507	P505	P508	6
5	P308	P305	VSS	VCC								P003	P503	P504	P506	5
4	P306	P304	P300/TCK/SWCLK	P111	VSS	P613	PA09	PA00	P607	VCC	VSS	VSS	VCC	P501	P502	4
3	P303	P302	P108/TMS SWDIO	P110/TDI	VCC	P610	VCC	VSS	P604	P603	P105	P102	P800	P804	P500	3
2	P301	P112	P114	P608	P611	P614	PA10	PA01	P605	P601	P107	P104	P101	P802	P803	2
1	P109/TDO	P113	P115	P609	P612	P615	PA08	VCL	P606	P602	P600	P106	P103	P100	P801	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	

Figure 1.3 Pin assignment for 176-pin BGA (top view)

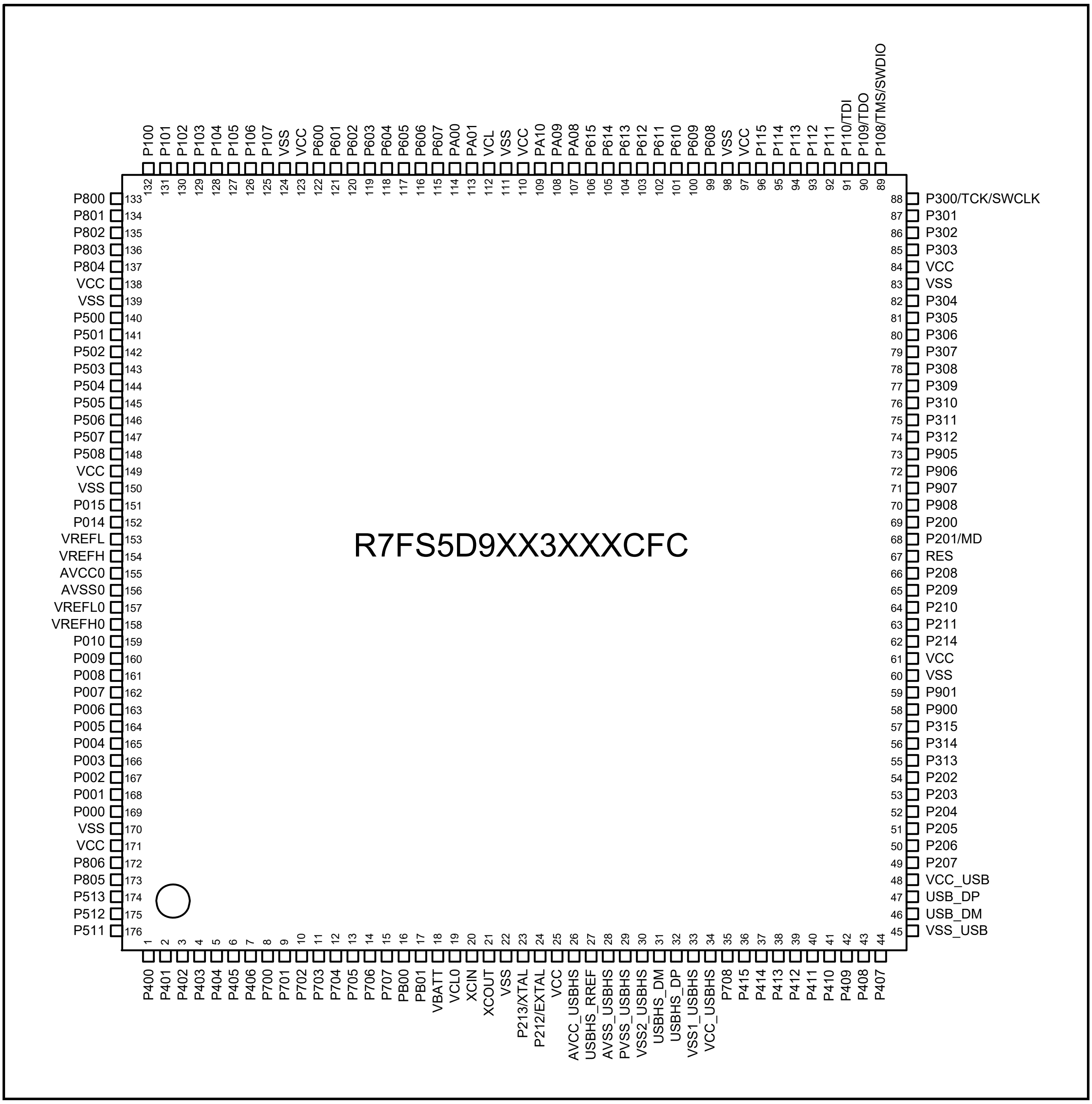
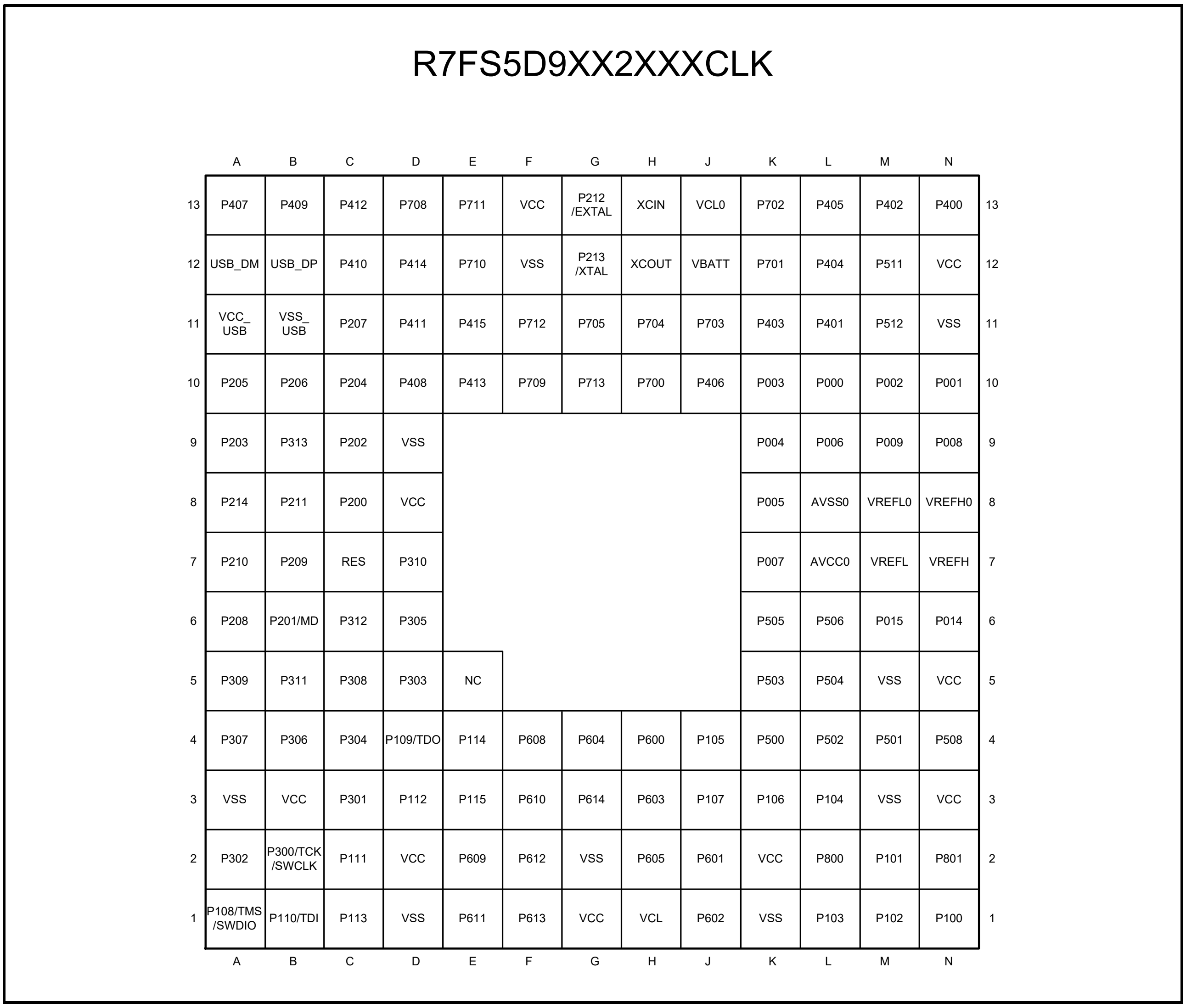


Figure 1.4 **Pin assignment for 176-pin LQFP (top view)**



R7FS5D9XX2XXXCLK

	A	B	C	D	E	F	G	H	J	K	L	M	N	
13	P407	P409	P412	P708	P711	VCC	P212 /EXTAL	XCIN	VCL0	P702	P405	P402	P400	13
12	USB_DM	USB_DP	P410	P414	P710	VSS	P213 /XTAL	XCOUT	VBATT	P701	P404	P511	VCC	12
11	VCC_ USB	VSS_ USB	P207	P411	P415	P712	P705	P704	P703	P403	P401	P512	VSS	11
10	P205	P206	P204	P408	P413	P709	P713	P700	P406	P003	P000	P002	P001	10
9	P203	P313	P202	VSS						P004	P006	P009	P008	9
8	P214	P211	P200	VCC						P005	AVSS0	VREFL0	VREFH0	8
7	P210	P209	RES	P310						P007	AVCC0	VREFL	VREFH	7
6	P208	P201/MD	P312	P305						P505	P506	P015	P014	6
5	P309	P311	P308	P303	NC					P503	P504	VSS	VCC	5
4	P307	P306	P304	P109/TDO	P114	P608	P604	P600	P105	P500	P502	P501	P508	4
3	VSS	VCC	P301	P112	P115	P610	P614	P603	P107	P106	P104	VSS	VCC	3
2	P302	P300/TCK /SWCLK	P111	VCC	P609	P612	VSS	P605	P601	VCC	P800	P101	P801	2
1	P108/TMS /SWDIO	P110/TDI	P113	VSS	P611	P613	VCC	VCL	P602	VSS	P103	P102	P100	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	

Figure 1.5 Pin assignment for 145-pin LGA (top view)

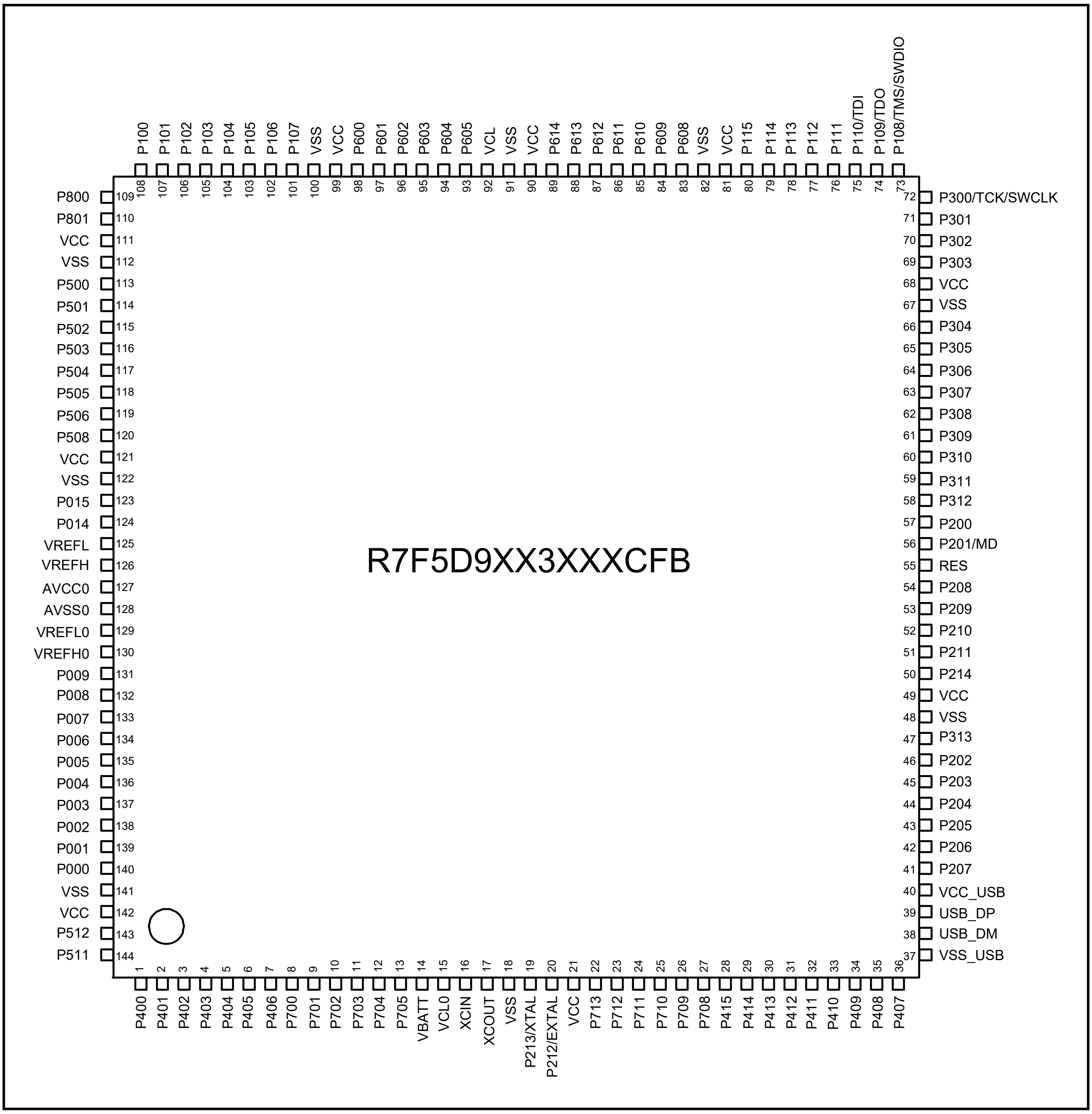


Figure 1.6 Pin assignment for 144-pin LQFP (top view)

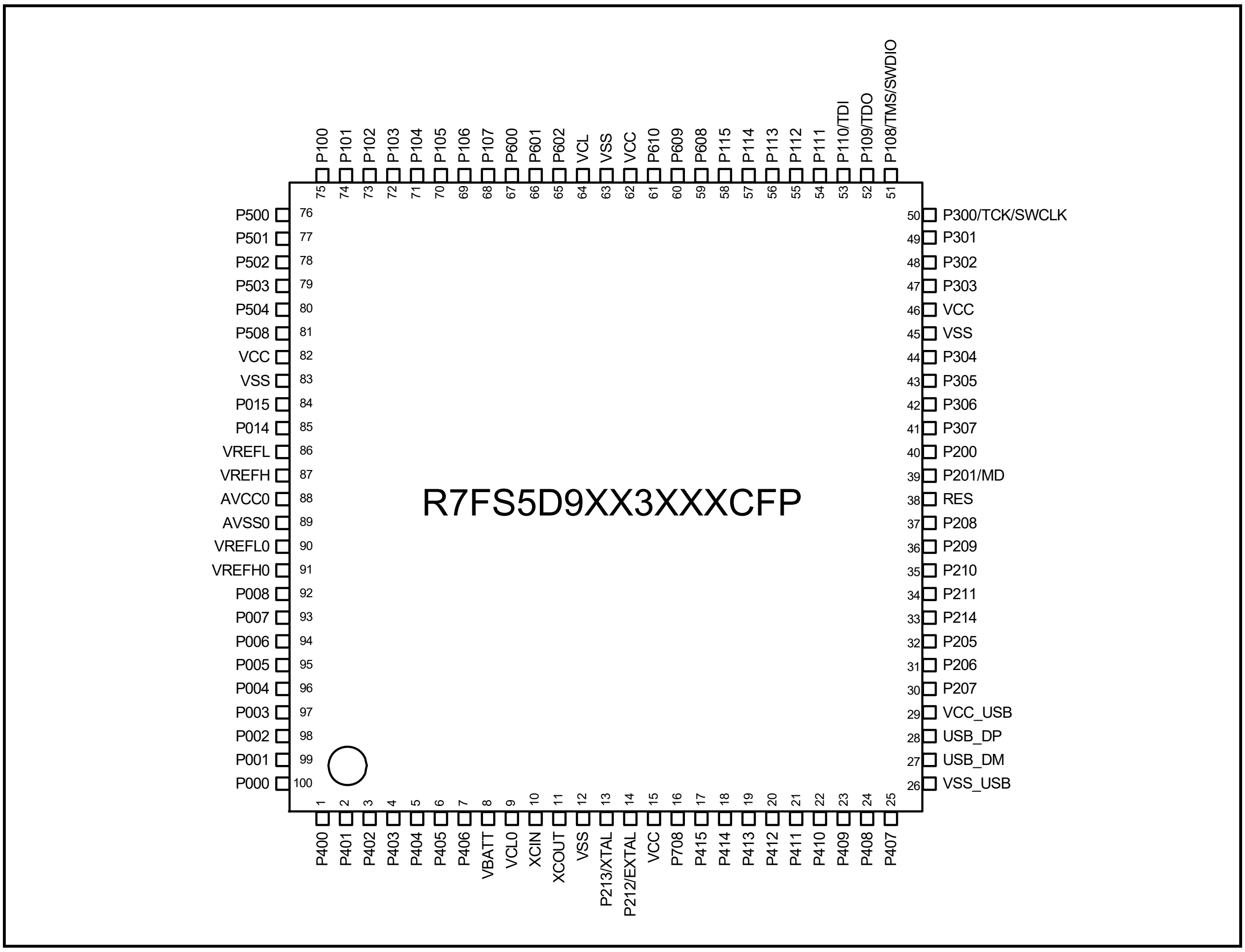


Figure 1.7 **Pin assignment for 100-pin LQFP (top view)**

1.7 Pin Lists

Pin number								Extbus		Timers				Communication interfaces										Analog		HMI	
BGA176	LQFP176	LGA145	LQFP144	LQFP100	Power, System, Clock, Debug, CAC	Interrupt	I/O port	External bus	SDRAM	AGT	GPT	GPT	RTC	USBFS, CAN	SCI0,2,4,6,8 (30 MHz)	SCI1,3,5,7,9 (30 MHz)	IIC	SPI, QSPI	SSIE	ETHERC (MII) (25 MHz)	ETHERC (RMII) (50 MHz)	USBHS	SDHI	ADC12	DAC12, ACMPHS	CTSU	GLCDC, PDC
N13	1	N13	1	1	-	IRQ0	P400	-	-	AGTIO1	-	GTIOC6A	-	-	SCK4	SCK7	SCL0_A	-	AUDIO_CLK	ET0_WOL	ET0_WOL	-	-	ADTRG1	-	-	-
R15	2	L11	2	2	-	IRQ5-DS	P401	-	-	-	GTETRGA	GTIOC6B	-	CTX0	CTS4_RTS4/SS4	TXD7/MOSI7/SDA7	SDA0_A	-	-	ET0_MDC	ET0_MDC	-	-	-	-	-	-
P14	3	M13	3	3	CACREF	IRQ4-DS	P402	-	-	AGTIO0/AGTIO1	-	-	RTCIC0	CRX0	-	RXD7/MISO7/SCL7	-	-	AUDIO_CLK	ET0_MDIO	ET0_MDIO	-	-	-	-	-	VSYNC
M12	4	K11	4	4	-	-	P403	-	-	AGTIO0/AGTIO1	-	GTIOC3A	RTCIC1	-	-	CTS7_RTS7/SS7	-	-	SSIBCK0_A	ET0_LINKSTA	ET0_LINKSTA	-	SD1DAT7_B	-	-	-	PIXD7
M13	5	L12	5	5	-	-	P404	-	-	-	-	GTIOC3B	RTCIC2	-	-	-	-	-	SSILRCK0/SSIFS0_A	ET0_EXOUT	ET0_EXOUT	-	SD1DAT6_B	-	-	-	PIXD6
P15	6	L13	6	6	-	-	P405	-	-	-	-	GTIOC1A	-	-	-	-	-	-	SSITXD0_A	ET0_TX_EN	RMII0_TXD_EN_B	-	SD1DAT5_B	-	-	-	PIXD5
N14	7	J10	7	7	-	-	P406	-	-	-	-	GTIOC1B	-	-	-	-	-	SSLB3_C	SSIRXD0_A	ET0_RX_ER	RMII0_TXD1_B	-	SD1DAT4_B	-	-	-	PIXD4
N15	8	H10	8	-	-	-	P700	-	-	-	-	GTIOC5A	-	-	-	-	-	MISOB_C	-	ET0_ETXD1	RMII0_TXD0_B	-	SD1DAT3_B	-	-	-	PIXD3
M14	9	K12	9	-	-	-	P701	-	-	-	-	GTIOC5B	-	-	-	-	-	MOSIB_C	-	ET0_ETXD0	REF50CK0_B	-	SD1DAT2_B	-	-	-	PIXD2
L12	10	K13	10	-	-	-	P702	-	-	-	-	GTIOC6A	-	-	-	-	-	RSPCKB_C	-	ET0_ERXD1	RMII0_RXD0_B	-	SD1DAT1_B	-	-	-	PIXD1
M15	11	J11	11	-	-	-	P703	-	-	-	-	GTIOC6B	-	-	-	-	-	SSLB0_C	-	ET0_ERXD0	RMII0_RXD1_B	-	SD1DAT0_B	-	VCOUT	-	PIXD0
L13	12	H11	12	-	-	-	P704	-	-	AGTO0	-	-	-	CTX0	-	-	-	SSLB1_C	-	ET0_RX_CLK	RMII0_RX_ER_B	-	SD1CLK_B	-	-	-	HSYNC
K12	13	G11	13	-	-	-	P705	-	-	AGTIO0	-	-	-	CRX0	-	-	-	SSLB2_C	-	ET0_CRS	RMII0_CRS_DV_B	-	SD1CMD_B	-	-	-	PIXCLK
L14	14	-	-	-	-	IRQ7	P706	-	-	-	-	-	-	-	-	RXD3/MISO3/SCL3	-	-	-	-	-	USBHS_OVRCURB	SD1CD_B	-	-	-	-
L15	15	-	-	-	-	IRQ8	P707	-	-	-	-	-	-	-	-	TXD3/MOSI3/SDA3	-	-	-	-	-	USBHS_OVRCURA	SD1WP_B	-	-	-	-
J12	16	-	-	-	-	-	PB00	-	-	-	-	-	-	-	-	SCK3	-	-	-	-	-	USBHS_VBUSEN	-	-	-	-	-
K13	17	-	-	-	-	-	PB01	-	-	-	-	-	-	-	-	CTS3_RTS3/SS3	-	-	-	-	-	USBHS_VBUS	-	-	-	-	-
K14	18	J12	14	8	VBATT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
K15	19	J13	15	9	VCL0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J15	20	H13	16	10	XCIN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J14	21	H12	17	11	XCOUT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J13	22	F12	18	12	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H14	23	G12	19	13	XTAL	IRQ2	P213	-	-	-	GTETRGC	GTIOC0A	-	-	-	TXD1/MOSI1/SDA1	-	-	-	-	-	-	-	ADTRG1	-	-	-
H15	24	G13	20	14	EXTAL	IRQ3	P212	-	-	AGTEE1	GTETRGD	GTIOC0B	-	-	-	RXD1/MISO1/SCL1	-	-	-	-	-	-	-	-	-	-	-
H12	25	F13	21	15	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H13	26	-	-	-	AVCC_USBHS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G13	27	-	-	-	USBHS_RREF	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G14	28	-	-	-	AVSS_USBHS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G15	29	-	-	-	PVSS_USBHS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G12	30	-	-	-	VSS2_USBHS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F15	31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	USBHS_DM	-	-	-	-	-
F14	32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	USBHS_DP	-	-	-	-	-
F12	33	-	-	-	VSS1_USBHS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F13	34	-	-	-	VCC_USBHS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	G10	22	-	-	-	P713	-	-	AGTOA0	-	GTIOC2A	-	-	-	-	-	-	-	-	-	-	-	-	-	TS17	-

Pin number								Extbus		Timers				Communication interfaces										Analog		HMI	
BGA176	**LQFP176**	**LGA145**	**LQFP144**	**LQFP100**	**Power, System, Clock, Debug, CAC**	**Interrupt**	**I/O port**	**External bus**	**SDRAM**	**AGT**	**GPT**	**GPT**	**RTC**	**USBFS, CAN**	**SCI0,2,4,6,8 (30 MHz)**	**SCI1,3,5,7,9 (30 MHz)**	**IIC**	**SPI, QSPI**	**SSIE**	**ETHERC (MII) (25 MHz)**	**ETHERC (RMII) (50 MHz)**	**USBHS**	**SDHI**	**ADC12**	**DAC12, ACMPHS**	**CTSU**	**GLCDC, PDC**
-	-	F11	23	-	-	-	P712	-	-	AGTOB0	-	GTIOC2B	-	-	-	-	-	-	-	-	-	-	-	-	-	TS16	-
-	-	E13	24	-	-	-	P711	-	-	AGTEE0	-	-	-	-	-	CTS1_RTS1/SS1	-	-	-	ET0_TX_CLK	-	-	-	-	-	TS15	-
-	-	E12	25	-	-	-	P710	-	-	-	-	-	-	-	-	SCK1	-	-	-	ET0_TX_ER	-	-	-	-	-	TS14	-
-	-	F10	26	-	-	IRQ10	P709	-	-	-	-	-	-	-	-	TXD1/MOSI1/SDA1	-	-	-	ET0_ETXD2	-	-	-	-	-	TS13	-
E15	35	D13	27	16	CACREF	IRQ11	P708	-	-	-	-	-	-	-	-	RXD1/MISO1/SCL1	-	SSLA3_B	AUDIO_CLK	ET0_ETXD3	-	-	-	-	-	TS12	PCKO
E14	36	E11	28	17	-	IRQ8	P415	-	-	-	-	GTIOC0A	-	USB_VBUSEN	-	-	-	SSLA2_B	-	ET0_TX_EN	RMII0_TXD_EN_A	-	SD0CD_A	-	-	TS11	PIXD5
D15	37	D12	29	18	-	IRQ9	P414	-	-	-	-	GTIOC0B	-	-	-	-	-	SSLA1_B	-	ET0_RX_ER	RMII0_TXD1_A	-	SD0WP_A	-	-	TS10	PIXD4
E13	38	E10	30	19	-	-	P413	-	-	-	GTOUUP	-	-	-	CTS0_RTS0/SS0	-	-	SSLA0_B	-	ET0_ETXD1	RMII0_TXD0_A	-	SD0CLK_A	-	-	TS09	PIXD3
D14	39	C13	31	20	-	-	P412	-	-	AGTEE1	GTOULO	-	-	-	SCK0	-	-	RSPCKA_B	-	ET0_ETXD0	REF50CK0_A	-	SD0CMD_A	-	-	TS08	PIX02
C15	40	D11	32	21	-	IRQ4	P411	-	-	AGTOA1	GTOVUP	GTIOC9A	-	-	TXD0/MOSI0/SDA0	CTS3_RTS3/SS3	-	MOSIA_B	-	ET0_ERXD1	RMII0_RXD0_A	-	SD0DAT0_A	-	-	TS07	PIX01
C14	41	C12	33	22	-	IRQ5	P410	-	-	AGTOB1	GTOVLO	GTIOC9B	-	-	RXD0/MISO0/SCL0	SCK3	-	MISOA_B	-	ET0_ERXD0	RMII0_RXD1_A	-	SD0DAT1_A	-	-	TS06	PIXD0
B15	42	B13	34	23	-	IRQ6	P409	-	-	-	GTOWUP	GTIOC10A	-	USB_EXICEN	-	TXD3/MOSI3/SDA3	-	-	-	ET0_RX_CLK	RMII0_RX_ER_A	USBHS_EXICEN	-	-	-	TS05	HSYNC
D13	43	D10	35	24	-	IRQ7	P408	-	-	-	GTOWLO	GTIOC10B	-	USB_ID	-	RXD3/MISO3/SCL3	SCL0_B	-	-	ET0_CRS	RMII0_CRS_DV_A	USBHS_ID	-	-	-	TS04	PIXCLK
A15	44	A13	36	25	-	-	P407	-	-	AGTIO0	-	-	RTCOUT	USB_VBUS	CTS4_RTS4/SS4	-	SDA0_B	SSLB3_A	-	ET0_EXOUT	ET0_EXOUT	-	-	ADTRG0	-	TS03	-
C13	45	B11	37	26	VSS_USB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B14	46	A12	38	27	-	-	-	-	-	-	-	-	-	USB_DM	-	-	-	-	-	-	-	-	-	-	-	-	-
A14	47	B12	39	28	-	-	-	-	-	-	-	-	-	USB_DP	-	-	-	-	-	-	-	-	-	-	-	-	-
B13	48	A11	40	29	VCC_USB	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C12	49	C11	41	30	-	-	P207	A17	-	-	-	-	-	-	-	-	-	SSLB2_A/QSSL	-	-	-	-	-	-	-	TS02	LCD_DATA23_B
D12	50	B10	42	31	-	IRQ0-DS	P206	WAIT	-	-	GTIU	-	-	USB_VBUSEN	RXD4/MISO4/SCL4	-	SDA1_A	SSLB1_A	SSIDATA1_A	ET0_LINKSTA	ET0_LINKSTA	-	SD0DAT2_A	-	-	TS01	-
E12	51	A10	43	32	CLKOUT	IRQ1-DS	P205	A16	-	AGTO1	GTIV	GTIOC4A	-	USB_OVRCURA-DS	TXD4/MOSI4/SDA4	CTS9_RTS9/SS9	SCL1_A	SSLB0_A	SSILRCK1/SSIFS1_A	ET0_WOL	ET0_WOL	-	SD0DAT3_A	-	-	TSCAP	-
A13	52	C10	44	-	CACREF	-	P204	A18	-	AGTIO1	GTIW	GTIOC4B	-	USB_OVRCURB-DS	SCK4	SCK9	SCL0_B	RSPCKB_A	SSIBCK1_A	ET0_RX_DV	-	-	SD0DAT4_A	-	-	TS00	-
D11	53	A9	45	-	-	IRQ2-DS	P203	A19	-	-	-	GTIOC5A	-	CTX0	CTS2_RTS2/SS2	TXD9/MOSI9/SDA9	-	MOSIB_A	-	ET0_COL	-	-	SD0DAT5_A	-	-	TSCAP	-
B12	54	C9	46	-	-	IRQ3-DS	P202	WR1/BC1	-	-	-	GTIOC5B	-	CRX0	SCK2	RXD9/MISO9/SCL9	-	MISOB_A		ET0_ERXD2	-	-	SD0DAT6_A	-	-	-	LCD_TCON3_B
A12	55	B9	47	-	-	-	P313	A20	-	-	-	-	-	-	-	-	-	-	-	ET0_ERXD3	-	-	SD0DAT7_A	-	-	-	LCD_TCON2_B
C11	56	-	-	-	-	-	P314	A21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	ADTRG0	-	-	LCD_TCON1_B
B11	57	-	-	-	-	-	P315	A22	-	-	-	-	-	-	RXD4	-	-	-	-	-	-	-	-	-	-	-	LCD_TCON0_B
A11	58	-	-	-	-	-	P900	A23	-	-	-	-	-	-	TXD4	-	-	-	-	-	-	-	-	-	-	-	LCD_CLK_B
C10	59	-	-	-	-	-	P901	-	-	AGTIO1	-	-	-	-	SCK4	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA15_B
D10	60	D9	48	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D9	61	D8	49	-	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A10	62	A8	50	33	TRCLK	-	P214	-	-	-	GTIU	-	-	-	-	-	-	QSPCLK	-	ET0_MDC	ET0_MDC	-	SD0CLK_B	-	-	-	LCD_DATA22_B
B10	63	B8	51	34	TRDATA0	-	P211	-	-	-	GTIV	-	-	-	-	-	-	QIO0	-	ET0_MDIO	ET0_MDIO	-	SD0CMD_B	-	-	-	LCD_DATA21_B
A9	64	A7	52	35	TRDATA1	-	P210	-	-	-	GTIW	-	-	-	-	-	-	QIO1	-	ET0_WOL	ET0_WOL	-	SD0CD_B	-	-	-	LCD_DATA20_B
B9	65	B7	53	36	TRDATA2	-	P209	-	-	-	GTOVUP	-	-	-	-	-	-	QIO2	-	ET0_EXOUT	ET0_EXOUT	-	SD0WP_B	-	-	-	LCD_DATA19_B

Pin number								Extbus		Timers				Communication interfaces										Analog		HMI	
BGA176	LQFP176	LGA145	LQFP144	LQFP100	Power, System, Clock, Debug, CAC	Interrupt	I/O port	External bus	SDRAM	AGT	GPT	GPT	RTC	USBFS, CAN	SCI0,2,4,6,8 (30 MHz)	SCI1,3,5,7,9 (30 MHz)	IIC	SPI, QSPI	SSIE	ETHERC (MII) (25 MHz)	ETHERC (RMII) (50 MHz)	USBHS	SDHI	ADC12	DAC12, ACMPHS	CTSU	GLCDC, PDC
A8	66	A6	54	37	TRDATA3	-	P208	-	-	-	GTOVLO	-	-	-	-	-	-	QIO3	-	ET0_LINKSTA	ET0_LINKSTA	-	SD0DAT0_B	-	-	-	LCD_DATA18_B
C9	67	C7	55	38	RES	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B8	68	B6	56	39	MD	-	P201	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C8	69	C8	57	40	-	NMI	P200	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D8	70	-	-	-	-	-	P908	CS7	-	-	-	GTIOC2A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA14_B
D7	71	-	-	-	-	-	P907	CS6	-	-	-	GTIOC2B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA13_B
A7	72	-	-	-	-	-	P906	CS5	-	-	-	GTIOC3A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA12_B
B7	73	-	-	-	-	-	P905	CS4	-	-	-	GTIOC3B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA11_B
C7	74	C6	58	-	-	-	P312	CS3	CAS	AGTOA1	-	-	-	-	-	CTS3_RTS3/SS3	-	-	-	-	-	-	-	-	-	-	-
D6	75	B5	59	-	-	-	P311	CS2	RAS	AGTOB1	-	-	-	-	-	SCK3	-	-	-	-	-	-	-	-	-	-	LCD_DATA23_A
A6	76	D7	60	-	-	-	P310	A15	A15	AGTEE1	-	-	-	-	-	TXD3	-	QIO3	-	-	-	-	-	-	-	-	LCD_DATA22_A
B6	77	A5	61	-	-	-	P309	A14	A14	-	-	-	-	-	-	RXD3	-	QIO2	-	-	-	-	-	-	-	-	LCD_DATA21_A
A5	78	C5	62	-	-	-	P308	A13	A13	-	-	-	-	-	-	-	-	QIO1	-	-	-	-	-	-	-	-	LCD_DATA20_A
C6	79	A4	63	41	-	-	P307	A12	A12	-	GTOUUP	-	-	-	CTS6	-	-	QIO0	-	-	-	-	-	-	-	-	LCD_DATA19_A
A4	80	B4	64	42	-	-	P306	A11	A11	-	GTOULO	-	-	-	SCK6	-	-	QSSL	-	-	-	-	-	-	-	-	LCD_DATA18_A
B5	81	D6	65	43	-	IRQ8	P305	A10	A10	-	GTOWUP	-	-	-	TXD6/MOSI6/SDA6	-	-	QSPCLK	-	-	-	-	-	-	-	-	LCD_DATA17_A
B4	82	C4	66	44	-	IRQ9	P304	A09	A09	-	GTOWLO	GTIOC7A	-	-	RXD6/MISO6/SCL6	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA16_A
C5	83	A3	67	45	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D5	84	B3	68	46	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A3	85	D5	69	47	-	-	P303	A08	A08	-	-	GTIOC7B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA15_A
B3	86	A2	70	48	-	IRQ5	P302	A07	A07	-	GTOUUP	GTIOC4A	-	-	TXD2/MOSI2/SDA2	-	-	SSLB3_B	-	-	-	-	-	-	-	-	LCD_DATA14_A
A2	87	C3	71	49	-	IRQ6	P301	A06	A06	AGTIO0	GTOULO	GTIOC4B	-	-	RXD2/MISO2/SCL2	CTS9_RTS9/SS9	-	SSLB2_B	-	-	-	-	-	-	-	-	LCD_DATA13_A
C4	88	B2	72	50	TCK/SWCLK	-	P300	-	-	-	GTOUUP	GTIOC0A_A	-	-	-	-	-	SSLB1_B	-	-	-	-	-	-	-	-	-
C3	89	A1	73	51	TMS/SWDIO	-	P108	-	-	-	GTOULO	GTIOC0B_A	-	-	-	CTS9_RTS9/SS9	-	SSLB0_B	-	-	-	-	-	-	-	-	-
A1	90	D4	74	52	CLKOUT/TDO/SWO	-	P109	-	-	-	GTOVUP	GTIOC1A_A	-	CTX1	-	TXD9/MOSI9/SDA9	-	MOSIB_B	-	-	-	-	-	-	-	-	-
D3	91	B1	75	53	TDI	IRQ3	P110	-	-	-	GTOVLO	GTIOC1B_A	-	CRX1	CTS2_RTS2/SS2	RXD9/MISO9/SCL9	-	MISOB_B	-	-	-	-	-	-	VCOUT	-	-
D4	92	C2	76	54	-	IRQ4	P111	A05	A05	-	-	GTIOC3A_A	-	-	SCK2	SCK9	-	RSPCKB_B	-	-	-	-	-	-	-	-	LCD_DATA12_A
B2	93	D3	77	55	-	-	P112	A04	A04	-	-	GTIOC3B_A	-	-	TXD2/MOSI2/SDA2	SCK1	-	SSLB0_B	SSIBCK0_B	-	-	-	-	-	-	-	LCD_DATA11_A
B1	94	C1	78	56	-	-	P113	A03	A03	-	-	GTIOC2A	-	-	RXD2/MISO2/SCL2	-	-	-	SSILRCK0/SSIFS0_B	-	-	-	-	-	-	-	LCD_DATA10_A
C2	95	E4	79	57	-	-	P114	A02	A02	-	-	GTIOC2B	-	-	-	-	-	-	SSIRXD0_B	-	-	-	-	-	-	-	LCD_DATA09_A
C1	96	E3	80	58	-	-	P115	A01	A01	-	-	GTIOC4A	-	-	-	-	-	-	SSITXD0_B	-	-	-	-	-	-	-	LCD_DATA08_A
E3	97	D2	81	-	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E4	98	D1	82	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D2	99	F4	83	59	-	-	P608	A00/BC0	A00/DQM1	-	-	GTIOC4B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA07_A
D1	100	E2	84	60	-	-	P609	CS1	CKE	-	-	GTIOC5A	-	CTX1	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA06_A
F3	101	F3	85	61	-	-	P610	CS0	WE	-	-	GTIOC5B	-	CRX1	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA05_A
E2	102	E1	86	-	CLKOUT/CACREF	-	P611	-	SDCS	-	-	-	-	-	-	CTS7_RTS7/SS7	-	-	-	-	-	-	-	-	-	-	-
E1	103	F2	87	-	-	-	P612	D08[A08/D08]	DQ08	-	-	-	-	-	-	SCK7	-	-	-	-	-	-	-	-	-	-	-
F4	104	F1	88	-	-	-	P613	D09[A09/D09]	DQ09	-	-	-	-	-	-	TXD7	-	-	-	-	-	-	-	-	-	-	-
F2	105	G3	89	-	-	-	P614	D10[A10/D10]	DQ10	-	-	-	-	-	-	RXD7	-	-	-	-	-	-	-	-	-	-	-
F1	106	-	-	-	-	-	P615	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA10_B
G1	107	-	-	-	-	-	PA08	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA09_B

Pin number								Extbus		Timers				Communication interfaces										Analog		HMI	
BGA176	LQFP176	LGA145	LQFP144	LQFP100	Power, System, Clock, Debug, CAC	Interrupt	I/O port	External bus	SDRAM	AGT	GPT	GPT	RTC	USBFS, CAN	SCI0,2,4,6,8 (30 MHz)	SCI1,3,5,7,9 (30 MHz)	IIC	SPI, QSPI	SSIE	ETHERC (MII) (25 MHz)	ETHERC (RMII) (50 MHz)	USBHS	SDHI	ADC12	DAC12, ACMPHS	CTSU	GLCDC, PDC
G4	108	-	-	-	-	-	PA09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA08_B
G2	109	-	-	-	-	-	PA10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA07_B
G3	110	G1	90	62	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H3	111	G2	91	63	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H1	112	H1	92	64	VCL	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H2	113	-	-	-	-	-	PA01	-	-	-	-	-	-	-	SCK8	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA06_B
H4	114	-	-	-	-	-	PA00	-	-	-	-	-	-	-	TXD8	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA05_B
J4	115	-	-	-	-	-	P607	-	-	-	-	-	-	-	RXD8	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA04_B
J1	116	-	-	-	-	-	P606	-	-	-	-	-	RTCOUT	-	CTS8_RTS8/SS8	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA03_B
J2	117	H2	93	-	-	-	P605	D11[A11/D11]	DQ11	-	-	GTIOC8A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
J3	118	G4	94	-	-	-	P604	D12[A12/D12]	DQ12	-	-	GTIOC8B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
K3	119	H3	95	-	-	-	P603	D13[A13/D13]	DQ13	-	-	GTIOC7A	-	-	-	CTS9_RTS9/SS9	-	-	-	-	-	-	-	-	-	-	-
K1	120	J1	96	65	-	-	P602	EBCLK	SDCLK	-	-	GTIOC7B	-	-	-	TXD9	-	-	-	-	-	-	-	-	-	-	LCD_DATA04_A
K2	121	J2	97	66	-	-	P601	WR/WR0	DQM0	-	-	GTIOC6A	-	-	-	RXD9	-	-	-	-	-	-	-	-	-	-	LCD_DATA03_A
L1	122	H4	98	67	CLKOUT/CACREF	-	P600	RD	-	-	-	GTIOC6B	-	-	-	SCK9	-	-	-	-	-	-	-	-	-	-	LCD_DATA02_A
K4	123	K2	99	-	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
L4	124	K1	100	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
L2	125	J3	101	68	-	KR07	P107	D07[A07/D07]	DQ07	AGTOA0	-	GTIOC8A	-	-	CTS8_RTS8/SS8	-	-	-	-	-	-	-	-	-	-	-	LCD_DATA01_A
M1	126	K3	102	69	-	KR06	P106	D06[A06/D06]	DQ06	AGTOB0	-	GTIOC8B	-	-	SCK8	-	-	SSLA3_A	-	-	-	-	-	-	-	-	LCD_DATA00_A
L3	127	J4	103	70	-	IRQ0/KR05	P105	D05[A05/D05]	DQ05	-	GTETRGA	GTIOC1A	-	-	TXD8/MOSI8/SDA8	-	-	SSLA2_A	-	-	-	-	-	-	-	-	LCD_TCON3_A
M2	128	L3	104	71	-	IRQ1/KR04	P104	D04[A04/D04]	DQ04	-	GTETRGB	GTIOC1B	-	-	RXD8/MISO8/SCL8	-	-	SSLA1_A	-	-	-	-	-	-	-	-	LCD_TCON2_A
N1	129	L1	105	72	-	KR03	P103	D03[A03/D03]	DQ03	-	GTOWUP	GTIOC2A_A	-	CTX0	CTS0_RTS0/SS0	-	-	SSLA0_A	-	-	-	-	-	-	-	-	LCD_TCON1_A
M3	130	M1	106	73	-	KR02	P102	D02[A02/D02]	DQ02	AGTO0	GTOWLO	GTIOC2B_A	-	CRX0	SCK0	-	-	RSPCKA_A	-	-	-	-	-	ADTRG0	-	-	LCD_TCON0_A
N2	131	M2	107	74	-	IRQ1/KR01	P101	D01[A01/D01]	DQ01	AGTEE0	GTETRGB	GTIOC5A	-	-	TXD0/MOSI0/SDA0	CTS1_RTS1/SS1	SDA1_B	MOSIA_A	-	-	-	-	-	-	-	-	LCD_CLK_A
P1	132	N1	108	75	-	IRQ2/KR00	P100	D00[A00/D00]	DQ00	AGTIO0	GTETRGA	GTIOC5B	-	-	RXD0/MISO0/SCL0	SCK1	SCL1_B	MISOA_A	-	-	-	-	-	-	-	-	LCD_EXTCLK_A
N3	133	L2	109	-	-	-	P800	D14[A14/D14]	DQ14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R1	134	N2	110	-	-	-	P801	D15[A15/D15]	DQ15	-	-	-	-	-	-	-	-	-	-	-	-	-	SD1DAT4_A	-	-	-	-
P2	135	-	-	-	-	-	P802	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SD1DAT5_A	-	-	-	LCD_DATA02_B
R2	136	-	-	-	-	-	P803	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SD1DAT6_A	-	-	-	LCD_DATA01_B
P3	137	-	-	-		-	P804	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SD1DAT7_A	-	-	-	LCD_DATA00_B
N4	138	N3	111	-	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
M4	139	M3	112	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R3	140	K4	113	76	-	-	P500	-	-	AGTOA0	GTIU	GTIOC11A	-	USB_VBUSEN	-	-	-	QSPCLK	-	-	-	-	SD1CLK_A	AN016	IVREF0	-	-
P4	141	M4	114	77	-	IRQ11	P501	-	-	AGTOB0	GTIV	GTIOC11B	-	USB_OVRCURA	-	TXD5/MOSI5/SDA5	-	QSSL	-	-	-	-	SD1CMD_A	AN116	IVREF1	-	-
R4	142	L4	115	78	-	IRQ12	P502	-	-	-	GTIW	GTIOC12A	-	USB_OVRCURB	-	RXD5/MISO5/SCL5	-	QIO0	-	-	-	-	SD1DAT0_A	AN017	IVCMP0	-	-
N5	143	K5	116	79	-	-	P503	-	-	-	GTETRGC	GTIOC12B	-	USB_EXICEN	CTS6_RTS6/SS6	SCK5	-	QIO1	-	-	-	-	SD1DAT1_A	AN117	-	-	-
P5	144	L5	117	80	-	-	P504	ALE	-	-	GTETRGD	GTIOC13A	-	USB_ID	SCK6	CTS5_RTS5/SS5	-	QIO2	-	-	-	-	SD1DAT2_A	AN018	-	-	-
P6	145	K6	118	-	-	IRQ14	P505	-	-	-	-	GTIOC13B	-	-	RXD6/MISO6/SCL6	-	-	QIO3	-	-	-	-	SD1DAT3_A	AN118	-	-	-

Pin number								Extbus		Timers				Communication interfaces										Analog		HMI	
BGA176	LQFP176	LGA145	LQFP144	LQFP100	Power, System, Clock, Debug, CAC	Interrupt	I/O port	External bus	SDRAM	AGT	GPT	GPT	RTC	USBFS, CAN	SCI0,2,4,6,8 (30 MHz)	SCI1,3,5,7,9 (30 MHz)	IIC	SPI, QSPI	SSIE	ETHERC (MII) (25 MHz)	ETHERC (RMII) (50 MHz)	USBHS	SDHI	ADC12	DAC12, ACMPHS	CTSU	GLCDC, PDC
R5	146	L6	119	-	-	IRQ15	P506	-	-	-	-	-	-	-	TXD6/MOSI6/SDA6	-	-	-	-	-	-	-	SD1CD_A	AN019	-	-	-
N6	147	-	-	-	-	-	P507	-	-	-	-	-	-	-	-	CTS5_RTS5/SS5	-	-	-	-	-	-	SD1WP_A	AN119	-	-	-
R6	148	N4	120	81	-	-	P508	-	-	-	-	-	-	-	SCK6	SCK5	-	-	-	-	-	-	-	AN020	-	-	-
M7	149	N5	121	82	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
N7	150	M5	122	83	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P7	151	M6	123	84	-	IRQ13	P015	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN006/AN106	DA1/IVCMP1	-	-
R7	152	N6	124	85	-	-	P014	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN005/AN105	DA0/IVREF3	-	-
P8	153	M7	125	86	VREFL	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R8	154	N7	126	87	VREFH	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
N8	155	L7	127	88	AVCC0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
N9	156	L8	128	89	AVSS0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P9	157	M8	129	90	VREFL0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R9	158	N8	130	91	VREFH0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
M8	159	-	-	-	-	IRQ14-DS	P010	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN103	-	-	-
M9	160	M9	131	-	-	IRQ13-DS	P009	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN004	-	-	-
P10	161	N9	132	92	-	IRQ12-DS	P008	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN003	-	-	-
M6	162	K7	133	93	-	-	P007	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PGAVSS100/AN107	-	-	-
N10	163	L9	134	94	-	IRQ11-DS	P006	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN102	IVCMP2	-	-
R10	164	K8	135	95	-	IRQ10-DS	P005	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN101	IVCMP2	-	-
P11	165	K9	136	96	-	IRQ9-DS	P004	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN100	IVCMP2	-	-
M5	166	K10	137	97	-	-	P003	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PGAVSS000/AN007	-	-	-
R11	167	M10	138	98	-	IRQ8-DS	P002	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN002	IVCMP2	-	-
N11	168	N10	139	99	-	IRQ7-DS	P001	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN001	IVCMP2	-	-
R12	169	L10	140	100	-	IRQ6-DS	P000	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	AN000	IVCMP2	-	-
M10	170	N11	141	-	VSS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
M11	171	N12	142	-	VCC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
P12	172	-	-	-	-	-	P806	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LCD_EXTCLK_B
R13	173	-	-	-	-	-	P805	-	-	-	-	-	-	-	-	TXD5	-	-	-	-	-	-	-	-	-	-	LCD_DATA17_B
N12	174	-	-	-	-	-	P513	-	-	-	-	-	-	-	-	RXD5	-	-	-	-	-	-	-	-	-	-	LCD_DATA16_B
R14	175	M11	143	-	-	IRQ14	P512	-	-	-	-	GTIOC0A	-	CTX1	TXD4/MOSI4/SDA4	-	SCL2	-	-	-	-	-	-	-	-	-	VSYNC
P13	176	M12	144	-	-	IRQ15	P511	-	-	-	-	GTIOC0B	-	CRX1	RXD4/MISO4/SCL4	-	SDA2	-	-	-	-	-	-	-	-	-	PCKO

Note: Some pin names have the added suffix of _A, _B, and _C. When assigning the GPT, IIC, SPI, SSIE, ETHERC (RMII), SDHI, and GLCDC functionality, select the functional pins with the same suffix.

2. CPU

2.1 Overview

The MCU is based on the Arm® Cortex®-M4 core.

2.1.1 CPU

- Arm Cortex-M4
    - Revision: r0p1-01rel0
    - Armv7E-M architecture profile
    - Single Precision Floating Point Unit compliant with the ANSI/IEEE Std 754-2008.
- Memory Protection Unit (MPU)
    - Armv7 Protected Memory System Architecture
    - 8 protected regions.
- SysTick timer
    - Driven by SYSTICCLK (LOCO) or ICLK.

See reference 1. and reference 2. for details.

2.1.2 Debug

- Arm CoreSight™ ETM-M4
    - Revision: r0p1-00rel0
    - Arm ETM architecture version 3.5.
- CoreSight Instrumentation Trace Macrocell (ITM)
- Data Watchpoint and Trace Unit (DWT)
    - 4 comparators for watchpoints and triggers.
- Flash Patch and Breakpoint Unit (FPB)
    - Flash Patch (remap) function is unavailable, only breakpoint function is available
    - 6 instruction comparators
    - 2 literal comparators.
- CoreSight Time Stamp Generator (TSG)
    - Time stamp for ETM and ITM
    - Driven by CPU clock.
- Debug Register Module (DBGREG)
    - Reset control
    - Halt control.
- CoreSight Debug Access Port (DAP)
    - JTAG Debug Port (JTAG-DP)
    - Serial Wire Debug Port (SW-DP).
- Cortex-M4 Trace Port Interface Unit (TPIU)
    - 4-bit TPIU formatter output
    - Serial Wire Output.

- CoreSight Embedded Trace Buffer (ETB)
  - CoreSight Trace Memory Controller with ETB configuration
  - Buffer size: 2 KB.

See reference 1. and reference 2. for details.

2.1.3 Operating Frequency

The operating frequencies for the MCU are as follows:

- CPU core: maximum 120 MHz
- Trace (4-bit TPIU): maximum 60 MHz
- Trace (SWO): maximum 60 MHz
- JTAG interface: maximum 25 MHz
- SWD interface: maximum 25 MHz.

Figure 2.1 shows a block diagram of the Cortex-M4 CPU.

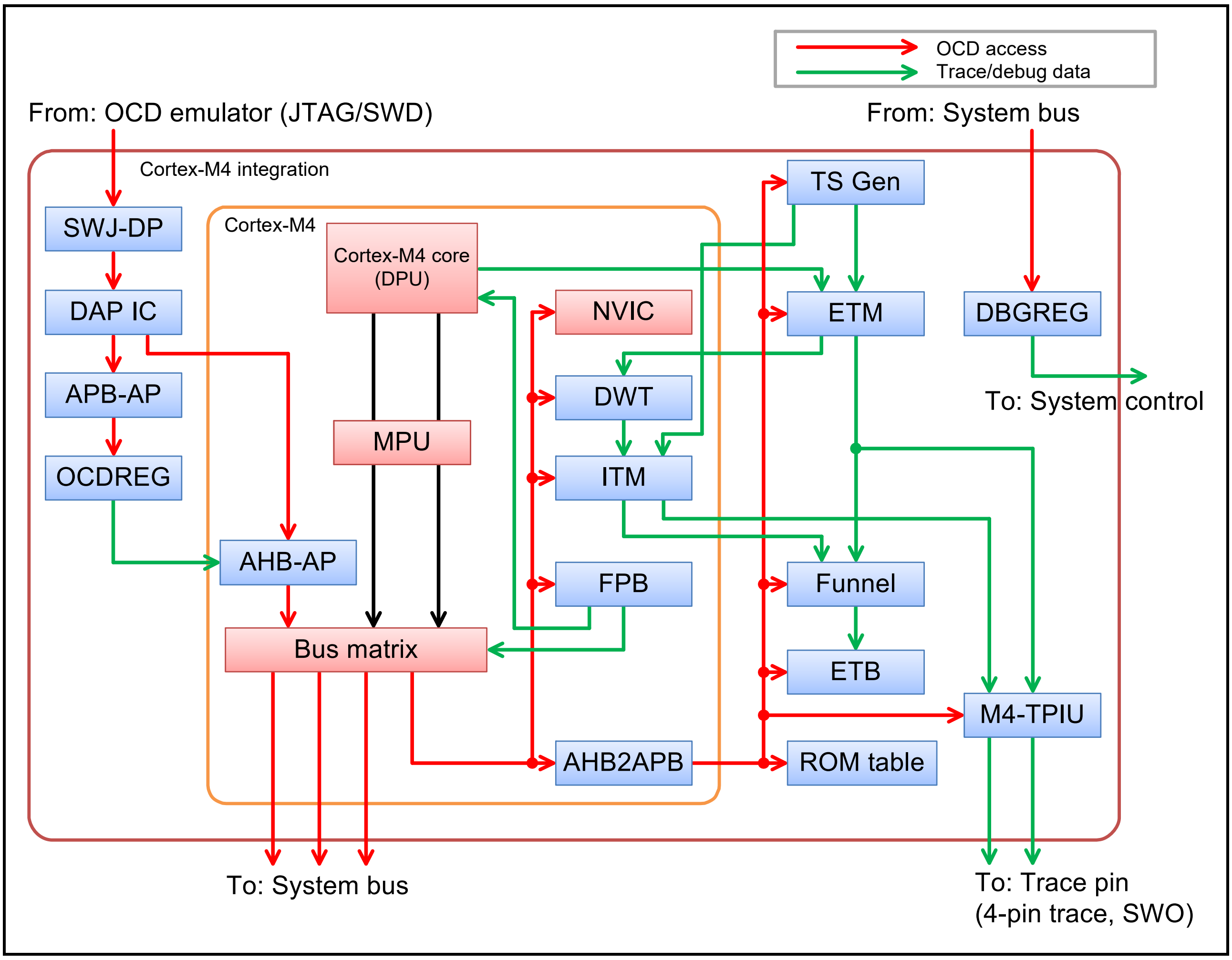


Figure 2.1 Cortex-M4 CPU block diagram

2.2 MCU Implementation Options

Table 2.1 Implementation options

Option	Implementation
MPU	Included, 8 protect regions
FPB	Flash Patch (remap) function is unavailable, only breakpoint function is available
FPU	Included
Number of interrupts	96
Number of priority bits	4
Number of Wakeup Interrupt Controllers (WIC*1)	Not included
Sleep mode power saving	Sleep mode and other low power modes are supported. For more details, see section 11, Low Power Modes. Note: SCB.SCR.SLEEPDEEP is ignored.
Endianness	Little-endian
SysTick SYST_CALIB register	SYST_CALIB = 4000 0147h Bit [31] = 0 Reference clock provided Bit [30] = 1 TERMS value is inexact Bits [29:24] = 00h Reserved Bits [23:0] = 000147h TERM: (32768 × 10 ms) - 1 / 32.768 kHz = 326.66 decimal = 327 with skew = 000147h
Event input/output	Not implemented
System reset request output	The SYSRESETREQ bit in the Application Interrupt and Reset Control Register causes a CPU reset
Auxiliary fault inputs (AUXFAULT)	Not implemented

Note 1. The ICU can wake up the CPU instead of the Wakeup Interrupt Controller (WIC). For details, see section 14, Interrupt Controller Unit (ICU).

2.3 Trace Interface

A Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO) provide trace output. Table 2.2 shows the MCU pins for the function. These pins are multiplexed with other functions.

Table 2.2 Trace function pins

Name	I/O	Width	Function
TCLK	Output	1 bit	Trace clock
TDATA0	Output	1 bit	Trace data output 0
TDATA1	Output	1 bit	Trace data output 1
TDATA2	Output	1 bit	Trace data output 2
TDATA3	Output	1 bit	Trace data output 3
TDO/SWO	Output	1 bit	Serial wire output Multiplexed with JTAG TDO pin

2.4 JTAG/SWD Interface

Table 2.3 shows the JTAG/SWD pins.

Table 2.3 JTAG/SWD pins

Name	I/O	P/N	Width	Function	When not in use
TCK/SWCLK	Input	Pos.	1 bit	JTAG clock pin	Pull-up
TMS/SWDIO	I/O	Neg.	1 bit	JTAG TMS pin SWD I/O pin	Pull-up
TDI	Input	Pos.	1 bit	JTAG TDI pin	Pull-up
TDO/SWO	Output	Neg.	1 bit	JTAG TDO pin Multiplexed with serial wire output	Open

2.5 Debug Mode

2.5.1 Debug Mode Definition

In single chip mode, the debugger state of the connection is defined as OCD mode, the debugger state of the unconnected is defined as User mode.

Table 2.4 shows the CPU debug modes and conditions.

Table 2.4 CPU debug mode and conditions

Conditions		Mode	
OCD connect	**JTAG/SWD authentication**	**Debug mode**	**Debug authentication**
Not connected	—	User mode	Disabled
Connected	Failed	User mode	Disabled
Connected	Passed	OCD mode	Enabled

Note 1. OCD connect is determined by the CDBGPWRUPREQ bit output in the SWJ-DP register. The bit can only be written by the OCD. However, the level of the bit can be confirmed by reading the DBGSTR.CDBGPWRUPREQ bit.

Note 2. Debug Authentication is defined by the ARMv7-M Architecture. Enabled means that both invasive and non-invasive CPU debugging are permitted. Disabled means that both are not permitted.

2.5.2 Debug Mode Effects

This section describes the effects of debug mode, which occur both internally and externally to the CPU.

2.5.2.1 Low power mode

All CoreSight debug components can store the register settings even when the CPU enters Software Standby, Snooze, or Deep Software Standby mode. However, AHB-AP cannot respond to On-Chip Debug (OCD) access in these low power modes. The OCD must wait for cancellation of the low power mode to access the CoreSight debug components. To request low power mode cancellation, the OCD can set the DBIRQ bit in the MCUCTRL register. For details, see section 2.6.5.3, MCU Control Register (MCUCTRL).

2.5.2.2 Reset

In OCD mode, some resets depend on the CPU status and the DBGSTOPCR setting.

Table 2.5 Reset or interrupt and mode setting (1 of 2)

	Control in On-Chip Debug (OCD) mode	
Reset or interrupt name	**OCD break mode**	**OCD run mode**
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2
Watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2
Voltage monitor 0 reset	Depends on DBGSTOPCR setting*3	

Table 2.5 Reset or interrupt and mode setting (2 of 2)

Reset or interrupt name	Control in On-Chip Debug (OCD) mode	
	OCD break mode	**OCD run mode**
Voltage monitor 1 reset/interrupt	Depends on DBGSTOPCR setting*3	
Voltage monitor 2 reset/interrupt	Depends on DBGSTOPCR setting*3	
SRAM parity error reset/interrupt	Depends on DBGSTOPCR setting*3	
SRAM ECC error reset/interrupt	Depends on DBGSTOPCR setting*3	
MPU bus master reset/interrupt	Same as user mode	
MPU bus slave reset/interrupt	Same as user mode	
Stack pointer error reset/interrupt	Same as user mode	
Deep software standby reset	Same as user mode	
Software reset	Same as user mode	

Note: In OCD break mode, the CPU is halted. In OCD run mode, the CPU is in OCD mode and the CPU is not halted.
Note 1. The IWDT and WDT always stop in this mode.
Note 2. IWDT and WDT operation depends on the DBGSTOPCR setting.
Note 3. Reset or interrupt masking depends on the DBGSTOPCR setting.

2.6 Programmers Model

2.6.1 Address Spaces

The MCU debug system includes two CoreSight Access Ports (AP):

- AHB-AP, which is connected to the CPU bus matrix and has the same access to the system address space as the CPU
- APB-AP, which has a dedicated address space (OCD address space) and is connected to the OCD register.

Figure 2.2 shows a block diagram of the AP connection and address spaces.

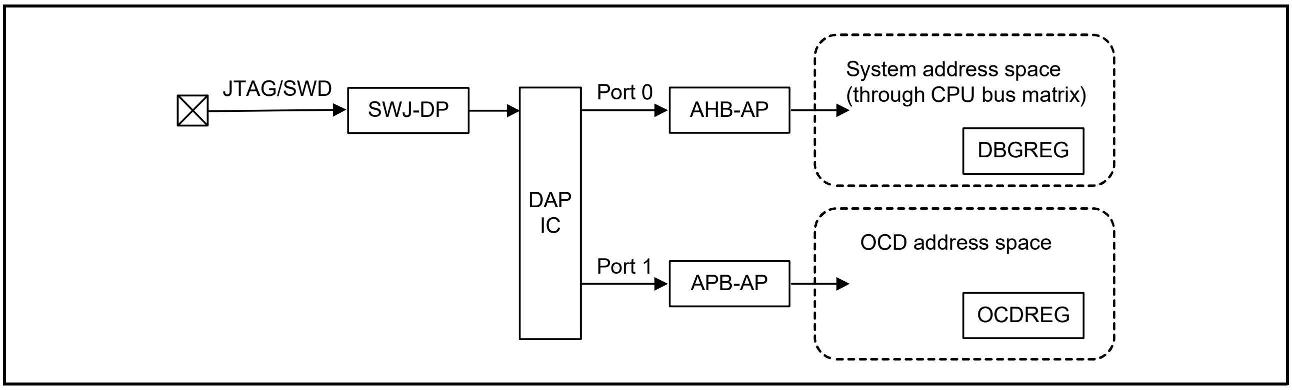


Figure 2.2 JTAG/SWD authentication block diagram

For debugging purposes, there are two register modules, DBGREG and OCDREG. DBGREG is located in the system address space can be accessed from the OCD emulator, CPU, and other bus masters in the MCU. OCDREG is located in the OCD address space and can only be accessed from the OCD tool. The CPU and other bus masters cannot access the OCD registers.

2.6.2 Cortex-M4 Peripheral Address Map

In the system address space, the Cortex-M4 core has a Private Peripheral Bus (PPB), which can be accessed only from the CPU and OCD emulator. The PPB is expanded from the Cortex-M4 original implementation for this MCU. Table 2.6 shows the address map of the MCU.

Table 2.6 Cortex-M4 peripheral address map

Component name	Start address	End address	Note
ITM	E000 0000h	E000 0FFFh	See reference 2.
DWT	E000 1000h	E000 1FFFh	See reference 2.
FPB	E000 2000h	E000 2FFFh	See reference 2.
SCS	E000 E000h	E000 EFFFh	See reference 2.
TPIU	E004 0000h	E004 0FFFh	See reference 2.
ETM	E004 1000h	E004 1FFFh	See reference 5.
ATB funnel	E004 2000h	E004 2FFFh	See section 2.7 and reference 4.
ETB	E004 3000h	E004 3FFFh	See reference 6.
Time Stamp Generator	E004 4000h	E004 4FFFh	See section 2.10 and reference 4.
ROM Table	E00F F000h	E00F FFFFh	See section 2.6.3 and reference 7.

2.6.3 CoreSight ROM Table

The MCU contains one CoreSight ROM Table, which lists the Arm components.

2.6.3.1 ROM entries

Table 2.7 shows the ROM entries in the CoreSight ROM Table. The OCD emulator can use the ROM entries to determine which components are implemented in a system. See reference 7. for details.

Table 2.7 CoreSight ROM Table

#	Address	Access size	R/W	Value	Component
0	E00F F000h	32 bits	R	FFF0F003	NVIC
1	E00F F004h	32 bits	R	FFF02003	SWT
2	E00F F008h	32 bits	R	FFF03003	FPB
3	E00F F00Ch	32 bits	R	FFF01003	ITM
4	E00F F010h	32 bits	R	FFF41003	TPIU
5	E00F F014h	32 bits	R	FFF42003	ETM
6	E00F F018h	32 bits	R	FFF43003	Funnel
7	E00F F01Ch	32 bits	R	FFF44003	ETB
8	E00F F020h	32 bits	R	FFF45003	TSG
9	E00F F024h	32 bits	R	00000000	(End of entries)

2.6.3.2 CoreSight component registers

The CoreSight ROM Table lists the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.8 shows the registers. See reference 7. for details of each register.

Table 2.8 CoreSight component registers in the CoreSight ROM Table (1 of 2)

Name	Address	Access size	R/W	Initial value
DEVTYPE	E00F FFCCh	32 bits	R	00000001h
PID4	E00F FFD0h	32 bits	R	00000004h
PID5	E00F FFD4h	32 bits	R	00000000h
PID6	E00F FFD8h	32 bits	R	00000000h

Table 2.8 CoreSight component registers in the CoreSight ROM Table (2 of 2)

Name	Address	Access size	R/W	Initial value
PID7	E00F FFDCh	32 bits	R	00000000h
PID0	E00F FFE0h	32 bits	R	00000010h
PID1	E00F FFE4h	32 bits	R	00000030h
PID2	E00F FFE8h	32 bits	R	0000000Ah
PID3	E00F FFECh	32 bits	R	00000000h
CID0	E00F FFF0h	32 bits	R	0000000Dh
CID1	E00F FFF4h	32 bits	R	00000010h
CID2	E00F FFF8h	32 bits	R	00000005h
CID3	E00F FFFCh	32 bits	R	000000B1h

2.6.4 DBGREG Module

The DBGREG register module controls the debug functionalities and is implemented as a CoreSight-compliant component.

Table 2.9 shows the DBGREG registers other than the CoreSight component registers.

Table 2.9 Non-CoreSight DBGREG registers

Name		DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0	4001 B000h	32 bits	R
Debug Stop Control Register	DBGSTOPCR	Port 0	4001 B010h	32 bits	R/W
Trace Control Register	TRACECTR	Port 0	4001 B020h	32 bits	R/W

2.6.4.1 Debug Status Register (DBGSTR)

Address(es): DBG.DBGSTR 4001 B000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	CDBGPWRUPACK	CDBGPWRUPREQ	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b27 to b0	—	Reserved	These bits are read as 0	R
b28	CDBGPWRUPREQ	Debug power-up request	0: OCD is not requesting debug power-up 1: OCD is requesting debug power-up.	R
b29	CDBGPWRUPACK	Debug power-up acknowledge	0: Debug power-up request is not acknowledged 1: Debug power-up request is acknowledged.	R
b31, b30	—	Reserved	These bits are read as 0	R

2.6.4.2 Debug Stop Control Register (DBGSTOPCR)

Address(es): DBG.DBGSTOPCR 4001 B010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	DBGSTOP_RECCR	DBGSTOP_RPER	—	—	—	—	—	DBGSTOP_LVD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGSTOP_WDT	DBGSTOP_IWDT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b0	DBGSTOP_IWDT	Mask bit for IWDT reset or interrupt	0: Enable IWDT reset or interrupt 1: Mask IWDT reset or interrupt and stop WDT count when CPU is in OCD break mode.	R/W
b1	DBGSTOP_WDT	Mask bit for WDT reset or interrupt	0: Enable WDT reset or interrupt 1: Mask WDT reset or interrupt and stop WDT count when CPU is in OCD break mode.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DBGSTOP_LVD[2:0]	Mask bit for LVD0 reset	0: Enable LVD0 reset 1: Mask LVD0 reset.	R/W
b17		Mask bit for LVD1 reset or interrupt	0: Enable LVD1 reset or interrupt 1: Mask LVD1 reset or interrupt.	R/W
b18		Mask bit for LVD2 reset or interrupt	0: Enable LVD2 reset or interrupt 1: Mask LVD2 reset or interrupt.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DBGSTOP_RPER	Mask bit for SRAM parity error reset or interrupt	0: Enable SRAM parity error reset or interrupt 1: Mask SRAM parity error reset or interrupt.	R/W
b25	DBGSTOP_RECCR	Mask bit for SRAM ECC error reset or interrupt	0: Enable SRAM ECC error reset or interrupt 1: Mask SRAM ECC error reset or interrupt.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The Debug Stop Control Register (DBGSTOPCR) specifies the functional stop in OCD mode. All bits in the register are regarded as 0 when the MCU is not in OCD mode.

2.6.4.3 Trace Control Register (TRACECTR)

Address(es): DBG.TRACECTR 4001 B020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ENETBFULL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b30 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b31	ENETBFULL	Enable bit for halt request on ETB full	0: ETB full does not cause a CPU halt 1: ETB full causes a CPU halt.	R/W

2.6.4.4 DBGREG CoreSight component registers

The DBGREG module provides the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.10 shows these registers. See reference 7. for details of each register.

Table 2.10 DBGREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PID4	4001 BFD0h	32 bits	R	00000004h
PID5	4001 BFD4h	32 bits	R	00000000h
PID6	4001 BFD8h	32 bits	R	00000000h
PID7	4001 BFDCh	32 bits	R	00000000h
PID0	4001 BFE0h	32 bits	R	00000005h
PID1	4001 BFE4h	32 bits	R	00000030h
PID2	4001 BFE8h	32 bits	R	0000001Ah
PID3	4001 BFECh	32 bits	R	00000000h
CID0	4001 BFF0h	32 bits	R	0000000Dh
CID1	4001 BFF4h	32 bits	R	000000F0h
CID2	4001 BFF8h	32 bits	R	00000005h
CID3	4001 BFFCh	32 bits	R	000000B1h

2.6.5 OCDREG Module

The OCDREG register module controls the On-Chip Debug (OCD) emulator functionalities and is implemented as a CoreSight-compliant component.

Table 2.11 shows the OCDREG registers other than the CoreSight component registers.

Table 2.11 Non-CoreSight OCDREG registers

Name		DAP port	Address	Access size	R/W
ID Authentication Code Register 0	IAUTH0	Port 1	8000_0000	32 bits	W
ID Authentication Code Register 1	IAUTH1	Port 1	8000_0100	32 bits	W
ID Authentication Code Register 2	IAUTH2	Port 1	8000_0200	32 bits	W
ID Authentication Code Register 3	IAUTH3	Port 1	8000_0300	32 bits	W
MCU Status Register	MCUSTAT	Port 1	8000 0400h	32 bits	R
MCU Control Register	MCUCTRL	Port 1	8000 0410h	32 bits	R/W

Note: OCDREG is located in the dedicated OCD address space. This address map is independent from the system address map. See section 2.6.2, Cortex-M4 Peripheral Address Map.

2.6.5.1 ID Authentication Code Register (IAUTH0 to 3)

Four authentication registers are provided for writing the 128-bit key. The registers must be written in sequential order from IAUTH0 to IAUTH3. If the set of register writes is not compliant with this order, the result is unpredictable.

Only 32-bit writes are permitted. The initial value of the registers is all 1s. This means that JTAG/SWD access is initially permitted when the ID code in the OSIS register has the initial value. See section 2.11.2, Unlock ID Code.

Address(es): IAUTH0 8000 0000h

	b31 … b0
	IAUTH0: AID 31-0 bits
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Address(es): IAUTH1 8000 0100h

	b31 … b0
	IAUTH1: AID 63-32 bits
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Address(es): IAUTH2 8000 0200h

	b31 … b0
	IAUTH2: AID 95-64 bits
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Address(es): IAUTH3 8000 0300h

	b31 … b0
	IAUTH3: AID 127-96 bits
Value after reset:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

2.6.5.2 MCU Status Register (MCUSTAT)

Address(es): MCUSTAT 8000 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUSTOPCLK	CPUSLEEP	AUTH
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1/0*1	1/0*1	0

Bit	Symbol	Bit name	Description	R/W
b0	AUTH	Authentication status	0: Authentication failed 1: Authentication succeeded.	R
b1	CPUSLEEP		0: CPU is not in Sleep mode 1: CPU is in Sleep mode.	R
b2	CPUSTOPCLK		0: CPU clock is not stopped. This indicates that the MCU is in Normal or Sleep mode 1: CPU clock is stopped. This indicates that the MCU is in Snooze or Software Standby mode.	R
b31 to b3	—	Reserved	These bits are read as 0.	R

Note 1. Depends on the MCU status.

2.6.5.3 MCU Control Register (MCUCTRL)

Address(es): MCUCTRL 8000 0410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DBIRQ	—	—	—	—	—	—	—	EDBGRQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EDBGRQ	External Debug Request	Writing 1 to the bit causes a CPU halt or debug monitor exception: 0: Debug event not requested 1: Debug event requested. When the EDBGRQ bit is set to 0 or the CPU is halted, the EDBCRQ bit is cleared.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DBIRQ	Debug Interrupt Request	Writing 1 to the bit wakes the MCU from low power mode: 0: Debug interrupt not requested 1: Debug interrupt requested. The condition can be cleared by writing 0 to the DBIRQ bit.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set DBIRQ and EDBGRQ to the same value.

2.6.5.4 CoreSight component registers

The DBGREG module provides the CoreSight component registers defined in the Arm CoreSight architecture. Table 2.12 lists these registers. See reference 7. for details.

Table 2.12 DBGREG registers

Name	Address	Access size	R/W	Initial value
PID4	8000 0FD0h	32 bits	R	00000004h
PID5	8000 0FD4h	32 bits	R	00000000h
PID6	8000 0FD8h	32 bits	R	00000000h
PID7	8000 0FDCh	32 bits	R	00000000h
PID0	8000 0FE0h	32 bits	R	00000004h
PID1	8000 0FE4h	32 bits	R	00000030h
PID2	8000 0FE8h	32 bits	R	0000000Ah
PID3	8000 0FECh	32 bits	R	00000000h
CID0	8000 0FF0h	32 bits	R	0000000Dh
CID1	8000 0FF4h	32 bits	R	000000F0h
CID2	8000 0FF8h	32 bits	R	00000005h
CID3	8000 0FFCh	32 bits	R	000000B1h

2.7 CoreSight ATB Funnel

There is one CoreSight ATB funnel in the MCU. The funnel has two ATB slaves and one ATB master, and it is used to select the debug trace source from ETM and ITM to ETB. Figure 2.3 shows the CoreSight ATB connection in the MCU.

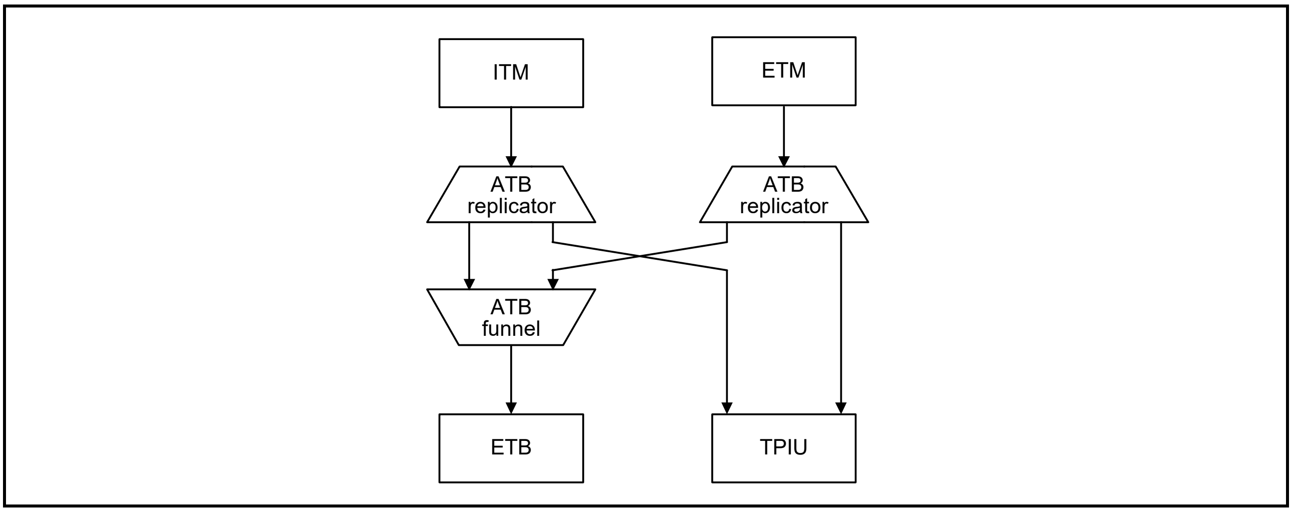


Figure 2.3 CoreSight ATB connection

Table 2.13 shows the ATB slave connection for the funnel.

Table 2.13 ATB slave connection

ATB slave number	Connected trace source
#0	ITM
#1	ETM

For details on the ATB and funnel, see reference 4.

2.8 Flash Patch and Break Unit

The MCU has a Flash Patch and Break Unit. Breakpoint function is available, but flash patch (remap) function is unavailable. Therefore, do not set 00b as the REPLACE bit (bit[31:30]) of the FP_COMPn register. Bit 28 of FP_REMAP register is fixed at 1b. When writing in this register, write 1b in bit 28. When reading this register, bit 28 always is read as 1b.

For details, see "Flash Patch and Breakpoint unit" chapter of reference 1.

2.9 SysTick System Timer

The SysTick system timer provides a simple 24-bit down counter. The reference clock for the timer can be selected as the CPU clock (ICLK) or SysTick Timer clock (SYSTICCLK). See reference 1.*1 for details.

Note 1. In the reference, the IMPLEMENTATION DEFINED external reference clock is SYSTICCLK (LOCO), and the processor clock is ICLK.

2.10 CoreSight Time Stamp Generator

A CoreSight Time Stamp Generator provides a CPU clock-based timestamp to ITM and ETM. The 48 LSB bits of the 64-bit counter are used for the two components. See reference 4. for details.

2.11 OCD Emulator Connection

A JTAG/SWD authentication mechanism checks access permission for debug and MCU resources. To obtain full debug functionality, a pass result of the authentication mechanism is required.

Figure 2.4 shows a block diagram of the authentication mechanism.

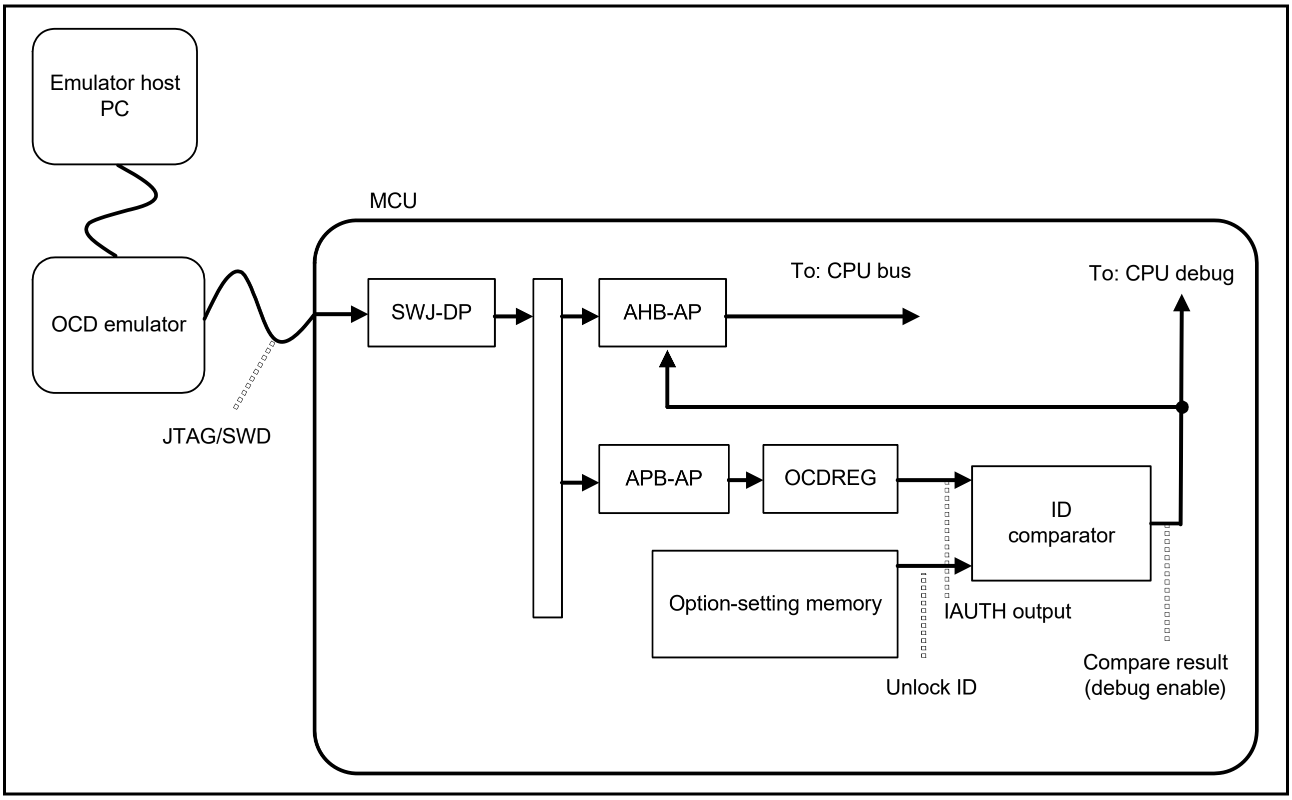


Figure 2.4 Authentication mechanism block diagram

An ID comparator is available in the MCU for authentication. The comparator compares the 128-bit IAUTH output from OCDREG and the 128-bit unlock ID code from the option-setting memory. When the two outputs are identical, the CPU debug functions and system bus access from the OCD emulator are permitted. After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting.

2.11.1 DBGEN

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting it. See section 11, Low Power Modes for details.

2.11.2 Unlock ID Code

The unlock ID code is used for checking permission for debug and access to on-chip resources. If the unlock ID code matches the 128-bit data written in ID Authentication Registers 0 to 3, the JTAG/SWD debugger obtains access permission. The unlock ID code is written in the OCD/Serial Programmer ID Setting Register (OSIS) in the option-setting memory. The initial value of the unlock ID code is all 1s (FFFFFFFF_FFFFFFFF_FFFFFFFFh). See section 7, Option-Setting Memory.

2.11.3 Restrictions on Connecting an OCD Emulator

This section describes the restrictions on emulator access.

2.11.3.1 Starting connection while in a low power mode

When starting a JTAG/SWD connection from an OCD emulator, the MCU must be in Normal or Sleep mode. If the MCU is in Software Standby, Snooze, or Deep Software Standby mode, the OCD emulator can cause the MCU to hang.

2.11.3.2 Changing low power mode while in OCD mode

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-AP is prohibited in Software Standby, Snooze, or Deep Software Standby mode. Only SWJ-DP, APB-AP, and OCDREG can be accessed from the OCD emulator in these modes. Table 2.14 shows the restrictions.

Table 2.14 Constraints by mode

Active mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Normal	Yes	Yes	Yes	Yes
Sleep	Yes	Yes	Yes	Yes
Software Standby	No	Yes	No	Yes
Snooze	No	Yes	No	Yes
Deep Software Standby	No	Yes	No	Yes

If system bus access is required in Software Standby, Snooze, or Deep Software Standby mode, set the OCUCTRL.DBIRG bit in OCDREG to wake the MCU from the low power modes. Simultaneously, using the OCUCTRL.EDBGRQ bit in OCDREG, the OCD emulator can wake the MCU without starting CPU execution by using a CPU break.

2.11.3.3 Modifying the unlock ID code in OSIS

After changing the Unlock ID code in the OSIS, the OCD emulator must reset the MCU by asserting the RES pin or setting the SYSRESETREQ bit of the Application Interrupt and Reset Control Register in the system control block to 1. The changed Unlock ID code is reflected after reset.

2.11.3.4 Connecting Sequence and JTAG/SWD Authentication

Because the OCD emulator is protected by the JTAG/SWD authentication mechanism, the OCD might be required to input the ID code to the authentication registers. The OSIS value in the option-setting memory determines whether the code is required.

After the negation of reset, a 5 µs wait time is required before comparing the OSIS value at the time of cold start.

(1) When MSB of OSIS is 0 (bit [127] = 0)

The ID code is always mismatching and connection to the OCD is prohibited.

(2) When OSIS is all 1s (default)

OCD authentication is not required and the OCD can use the AHB-AP without authentication.

1. Connect the OCD emulator to the MCU through the JTAG or SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJ-DP Control Status Register, then wait until CSDBGPWRUPACK in the same register is asserted.
3. Set up the AHB-AP to access the system address space. The AHB-AP is connected to DAP bus port 0.
4. Start accessing the CPU debug resources using the AHB-AP.

(3) When OSIS[127:126] = 2’b10

OCD authentication is required and the OCD must write the unlock ID code to IAUTH registers 0 to 3 in the OCDREG before using the AHB-AP.

1. Connect the OCD debugger to the MCU through the JTAG or SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJ-DP Control Status Register, and then wait until CSDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. The APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in the OCDREG using the APB-AP.
5. If the 128-bit ID code matches the OSIS value, the AHB-AP is authorized to issue an AHB transaction. The authorization result can be confirmed in the AUTH bit in the MCUSTAT Register or the DbgStatus bit in the AHB-AP Control Status Word Register.

  - When the DbgStatus bit is 1, the 128-bit ID code is a match with the OSIS value. AHB transfers are permitted.
  - When the DbgStatus bit is 0, the 128-bit ID code is not a match with the OSIS value. AHB transfers are not permitted.
6. Set up the AHB-AP to access the system address space. The AHB-AP is connected to DAP bus port 0.
7. Start accessing the CPU debug resources using the AHB-AP.

(4) When OSIS[127:126] = 2’b11

OCD authentication is required and the OCD must write the unlock ID code to IAUTH registers 0 to 3 in OCDREG. The connection sequence is the same when OSIS[127:126] is 2’b10 except for “ALeRASE” capability. When IATUH0-3 are “ALeRASE” in ASCII code, the contents of the code flash, data flash, and configuration area are erased at once. See section 55, Flash Memory for details.

ALeRASE sequence:

1. Connect the OCD debugger to the MCU through the JTAG or SWD interface.
2. Set up SWJ-DP to access DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJDP Control Status Register, and then wait until CSDBGPWRUPACK in the same register is asserted.
3. Set the APB-AP to access OCDREG. This APB-AP is connected to DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in the OCDREG using the APB-AP.
5. If the 128-bit ID code is “ALeRASE” in ASCII code (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFFh), the contents of code flash, data flash, and configuration area are erased. After that, the MCU transitions to Sleep mode.

2.12 References

1. *ARM®v7-M Architecture Reference Manual* (ARM DDI 0403D)
2. *ARM® Cortex®-M4 Processor Technical Reference Manual* (ARM DDI 0439D)
3. *ARM® Cortex®-M4 Devices Generic User Guide* (ARM DUI 0553A)
4. *ARM® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480F)
5. *ARM® CoreSight™ ETM-M4 Technical Reference Manual* (ARM DDI 0440C)
6. *ARM® CoreSight™ Trace Memory Controller Technical Reference Manual* (ARM DDI 0461B)
7. *ARM® CoreSight™ Architecture Specification* (ARM IHI 0029D).

3. Operating Modes

3.1 Overview

Table 3.1 shows the selection of operating modes by the mode-setting pin. For details, see section 3.2, Details of Operating Modes. Operation starts with the on-chip flash memory enabled, regardless of the mode in which operation started.

Table 3.1 Selection of operating modes by the mode-setting pin

Mode-setting pin			
MD	**Operating mode**	**On-chip flash memory**	**External bus**
1	Single-chip mode	Enable	Disable
0	SCI/USB boot mode	Enable	Disable

3.2 Details of Operating Modes

3.2.1 Single-Chip Mode

In single-chip mode, all I/O pins are available for use as input or output port, inputs or outputs for peripheral functions, or as interrupt inputs. When a reset is released while the MD pin is high, the MCU starts in single-chip mode and the on-chip flash is enabled.

3.2.2 SCI Boot Mode

In this mode, the on-chip flash memory programming routine (SCI boot program), stored in a dedicated area within the MCU, is used. The on-chip flash, including code flash memory and data flash memory, can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see section 55, Flash Memory. The MCU starts in SCI boot mode if the MD pin is held low on release from the reset state.

3.2.3 USB Boot Mode

In this mode, the on-chip flash memory programming routine (USB boot program), stored in the boot area within the MCU, is used. The on-chip flash, including code flash memory and data flash memory, can be modified from outside the MCU by using USB. For details, see section 55, Flash Memory. The chip starts in USB boot mode if the MD pin is held low on release from the reset state.

3.3 Operating Mode Transitions

3.3.1 Operating Mode Transitions as Determined by the Mode-Setting Pin

Figure 3.1 shows operating mode transitions determined by the settings of the MD pin.

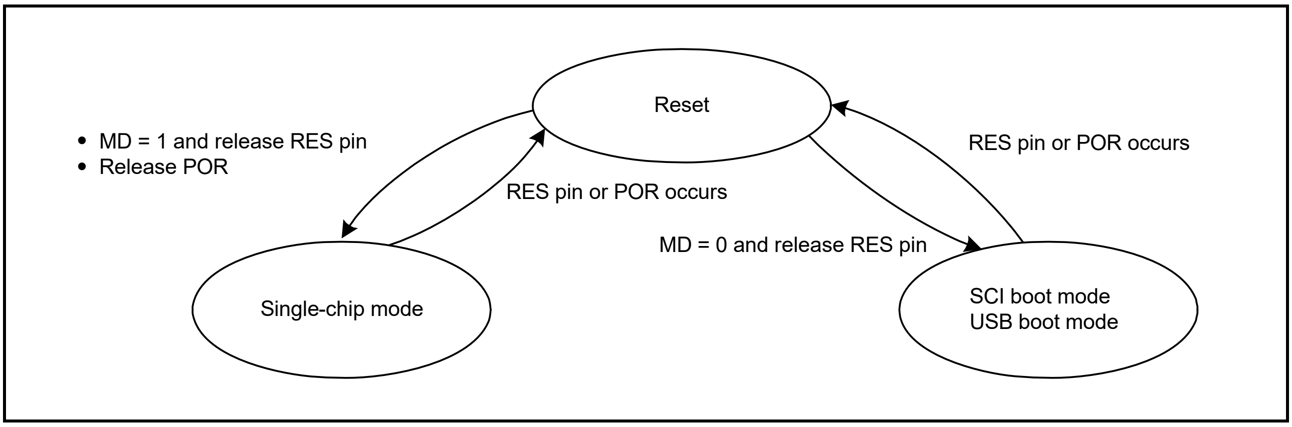
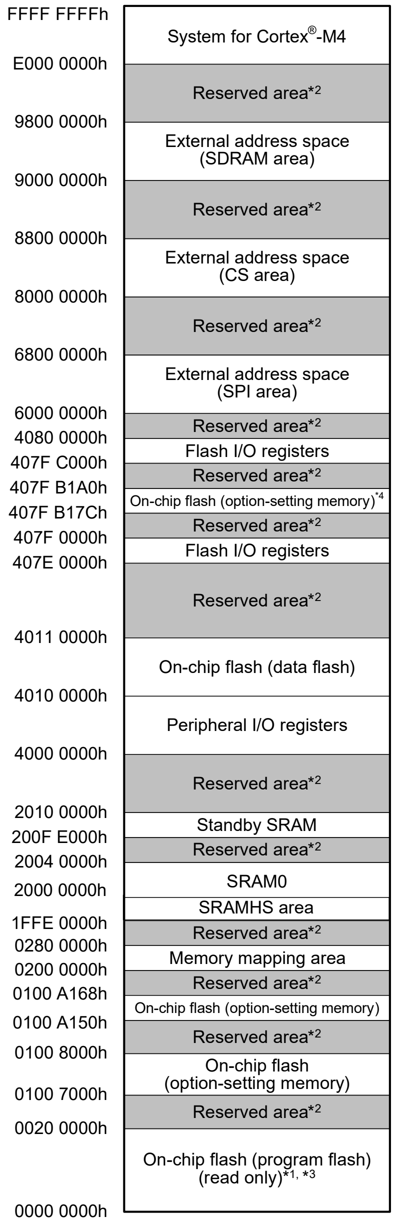


Figure 3.1 Mode-setting pin level and operating mode

4. Address Space

4.1 Address Space

The MCU supports a 4-GB linear address space ranging from 0000 0000h to FFFF FFFFh that can contain both program and data. Figure 4.1 shows the memory map.



Note 1. The capacity of the flash differs depending on the product.

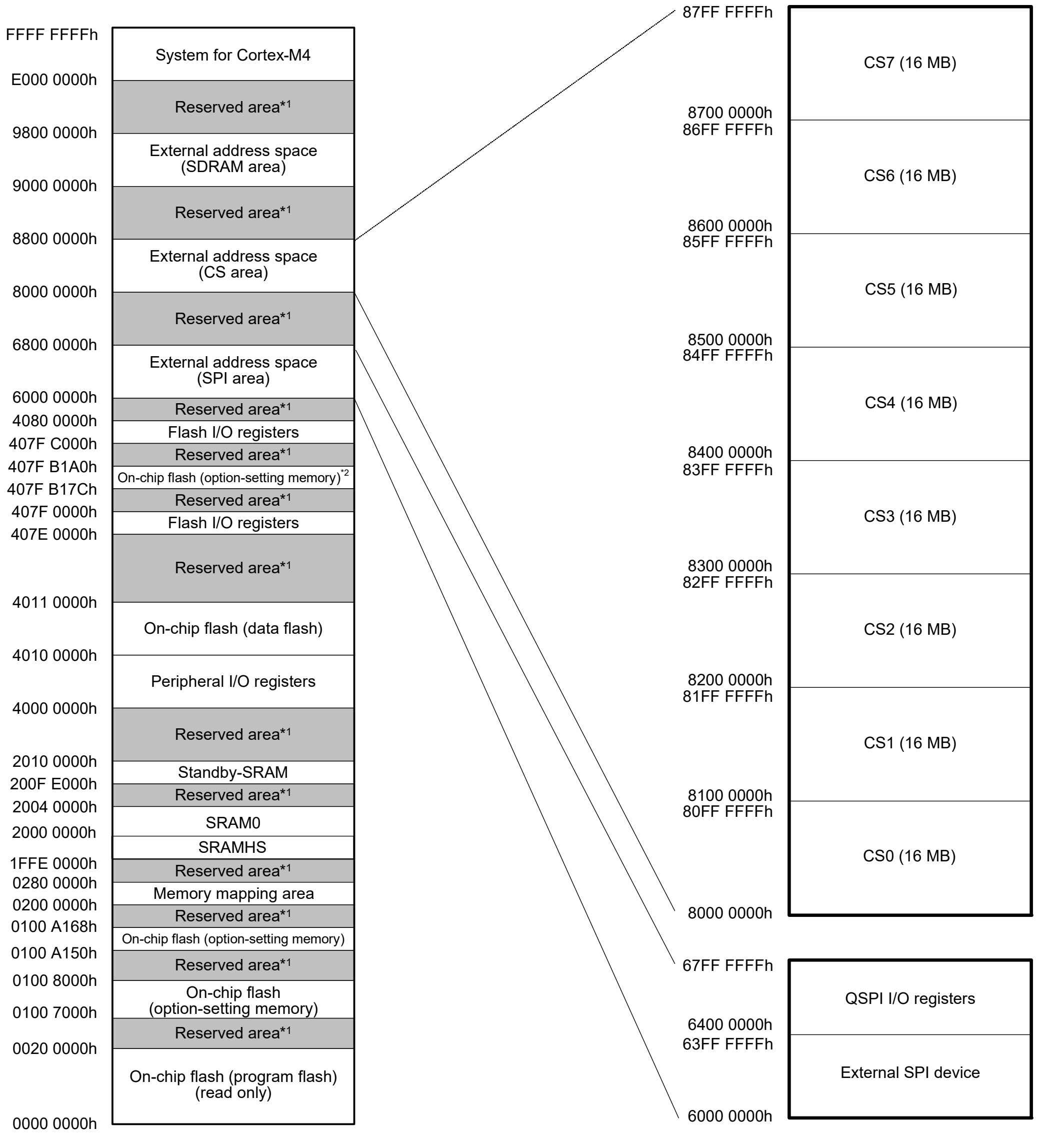
Code flash memory capacity	Address	Data flash memory capacity	Address	RAM capacity	Address
2 Mbytes	0000 0000h - 001F FFFFh	64 Kbytes	4010 0000h - 4010 FFFFh	640 Kbyte	1FFE 0000h - 2007 FFFFh
1 Mbytes	0000 0000h - 000F FFFFh				

Note 2. Do not access reserved areas.
Note 3. Some regions are reserved for the option-setting memory. For details, see section 7, Option-Setting Memory.
Note 4. Do not access from 407F B180h to 407F B19Bh.

Figure 4.1 Memory map

4.2 External Address Space

The external address space is divided into CS areas (CS0 to CS7), SDRAM area (SDCS), and SPI area. The eight CS areas (CS0 to CS7) each correspond to the CSn signal output from a CSn (n = 0 to 7) pin. The SPI area is divided into two areas, QSPI I/O registers and external SPI device space. Figure 4.2 shows the address ranges corresponding to the individual CS areas (CS0 to CS7), SDRAM area (SDCS) and SPI area.



Note 1. Do not access reserved areas.
Note 2. Do not access from 407F B180h to 407F B19Bh.

Figure 4.2 Association between external address spaces and CS areas

5. Memory Mirror Function (MMF)

5.1 Overview

The MCU provides a Memory Mirror Function (MMF). You can configure the MMF to map an application image load address in the code flash memory to the application image link address in the unused 23-bit memory mirror space addresses. Your application code must be developed and linked to run from this MMF destination address. The code is not required to know the load location where it is stored in the code flash memory.

Table 5.1 lists the MMF specifications.

Table 5.1 MMF specifications

Parameter	Specifications
Memory mirror space	8 MB (0200 0000h to 027F FFFFh)
Memory mirror boundary	128 bytes

5.2 Register Descriptions

5.2.1 MemMirror Special Function Register (MMSFR)

Address(es): MMF.MMSFR 4000 1000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	KEY[7:0]								—	MEMMIRADDR[15:9]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MEMMIRADDR[8:0]									—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b22 to b7	MEMMIRADDR[15:0]	Memory Mirror Address	0000h to FFFFh (8 MB)	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b24	KEY[7:0]	MMSFR Key Code	These bits enable or disable rewriting of the MEMMIRADDR bits	R/W

MEMMIRADDR[15:0] bits (Memory Mirror Address)

The MEMMIRADDR bits specify bits [22:7] of the memory mirror address. They define where the start address of the memory mirror space addresses (0200 0000h) is linked to. Writing to these bits is enabled only when this register is accessed in 32-bit words and DBh is written to the KEY[7:0] bits.

KEY[7:0] bits (MMSFR Key Code)

The KEY[7:0] bits enable or disable rewriting of the MEMMIRADDR bits. Data written to the KEY bits is not saved. These bits are read as 0. The KEY code and MEMMIRADDR must be written to in the same cycle.

5.2.2 MemMirror Enable Register (MMEN)

Address(es): MMF.MMEN 4000 1004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	KEY[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b0	EN	Memory Mirror Function Enable	0: Disable MMF 1: Enable MMF.	R/W
b23 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	KEY[7:0]	MMEN Key Code	These bits enable or disable rewriting of the EN bit	R/W

EN bit (Memory Mirror Function Enable)

Writing to the EN bit is enabled only when the MemMirror Enable Register is accessed in 32-bit words and DBh is written to the KEY[7:0] bits.

KEY[7:0] bits (MMEN Key Code)

The KEY[7:0] bits enable or disable rewriting of the EN bit. Data written to the KEY[7:0] bits is not saved. These bits are read as 0. The KEY code and EN must be written to in the same cycle.

5.3 Operation

5.3.1 MMF Operation

The MMF links the memory mirror space (0200 0000h to 027F FFFFh) to the code flash area. If MMEN.EN = 1, the CPU can access code flash using both normal addresses (starting at 0000 0000h) and memory mirror space addresses (starting at 0200 0000h). Figure 5.1 shows an overview of the MMF. MMSFR.MEMMIRADDR specifies where the start address of the memory mirror space addresses (0200 0000h) is linked to. Figure 5.2, Figure 5.3, and Figure 5.4 show the MMF operation. Figure 5.5 shows the setting procedure of the MMF.

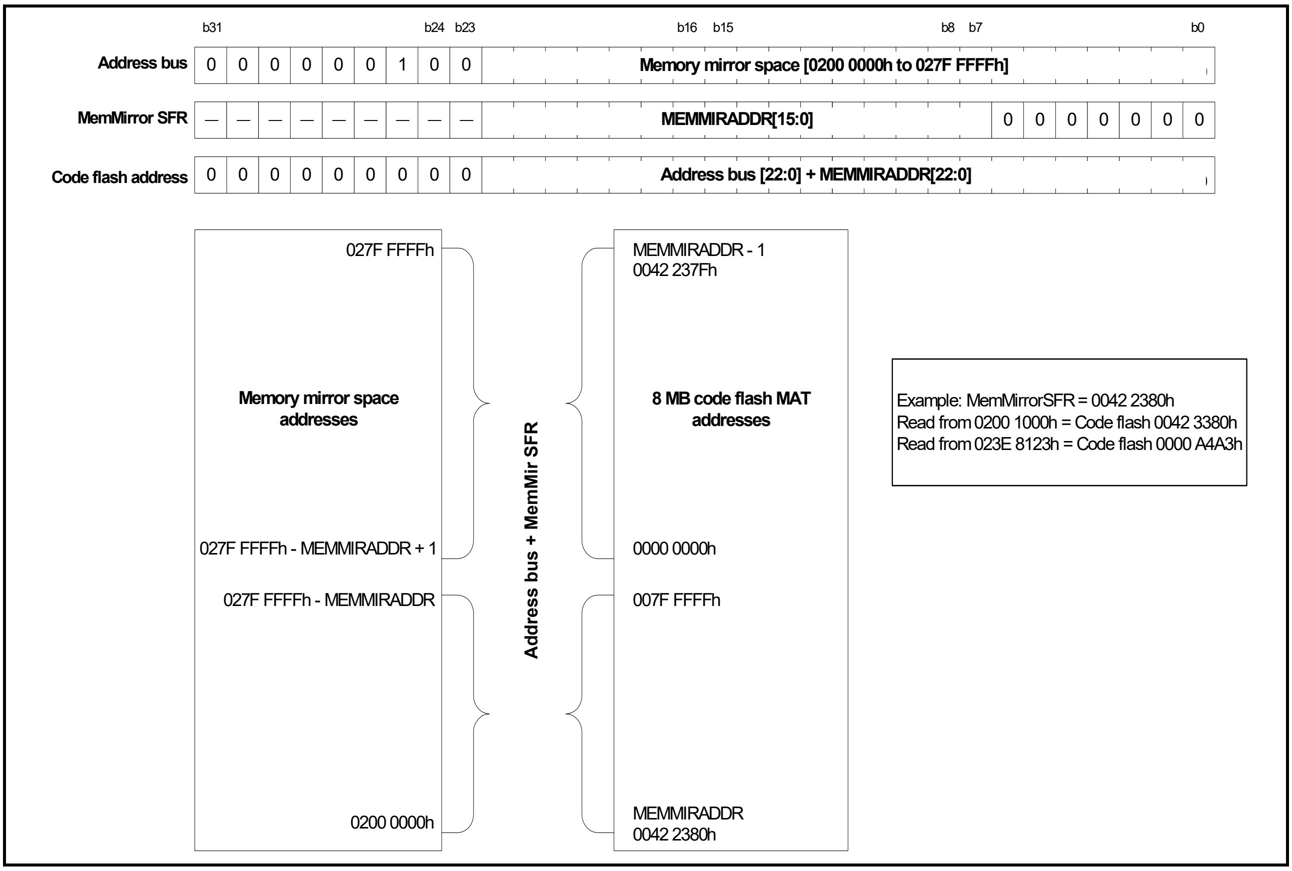
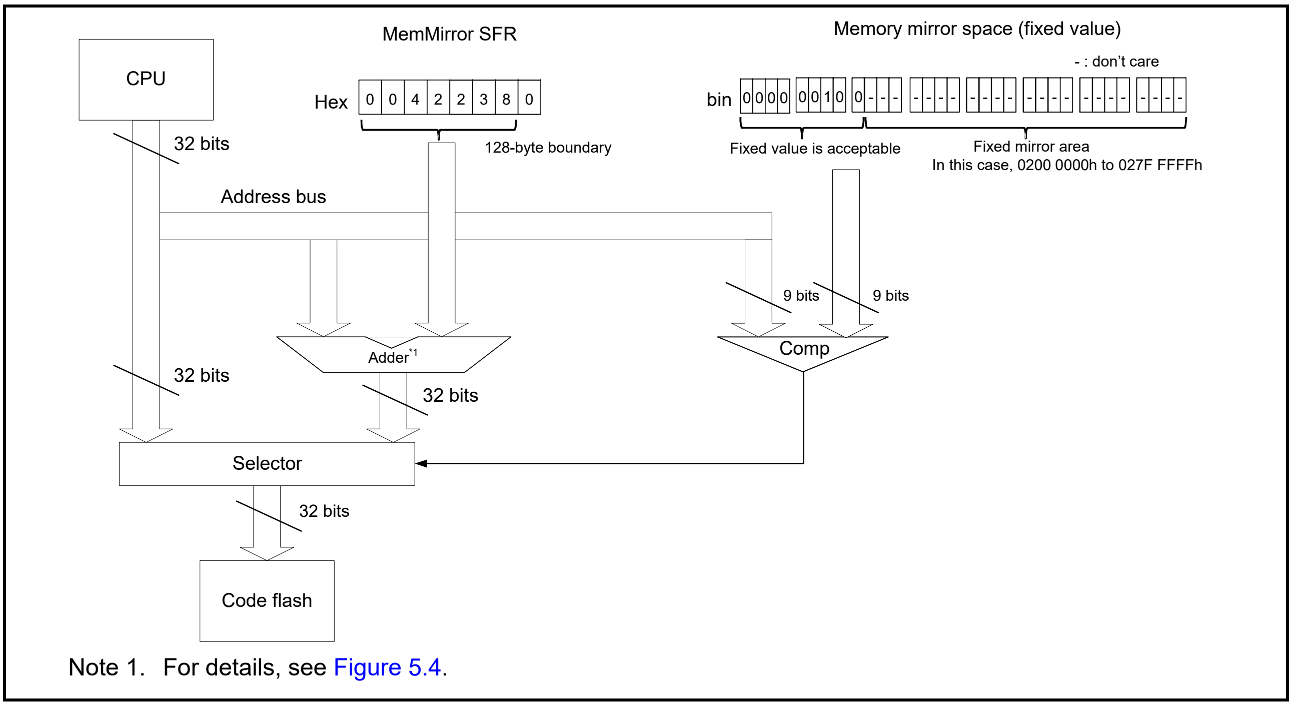


Figure 5.1 MMF operation



Note 1. For details, see Figure 5.4.

Figure 5.2 MMF block diagram

Figure 5.3 shows addresses handled by each module. The Arm® MPU uses the original address of the CPU.

The Security MPU and code flash memory each use an address after conversion through the memory mirror function.

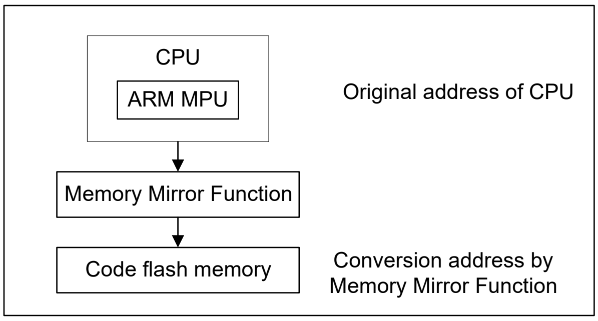


Figure 5.3 MMF address handling

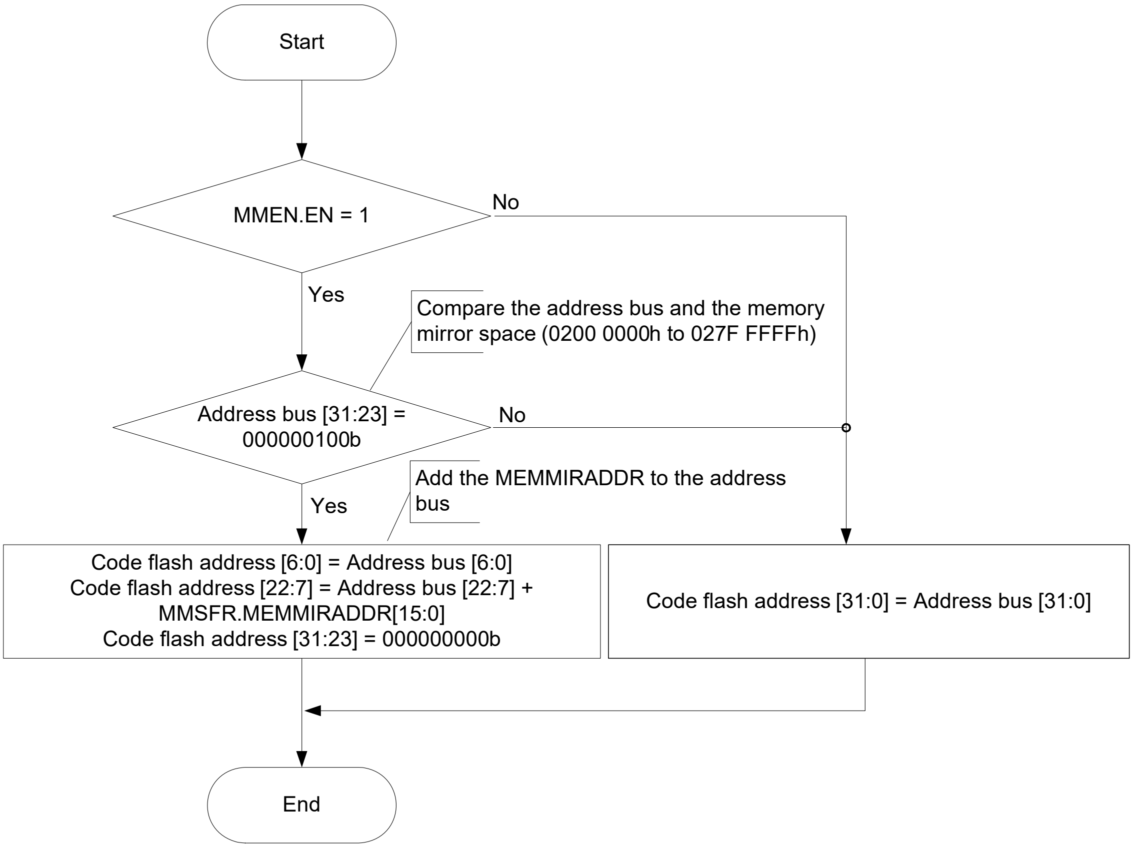


Figure 5.4 MMF operation flow

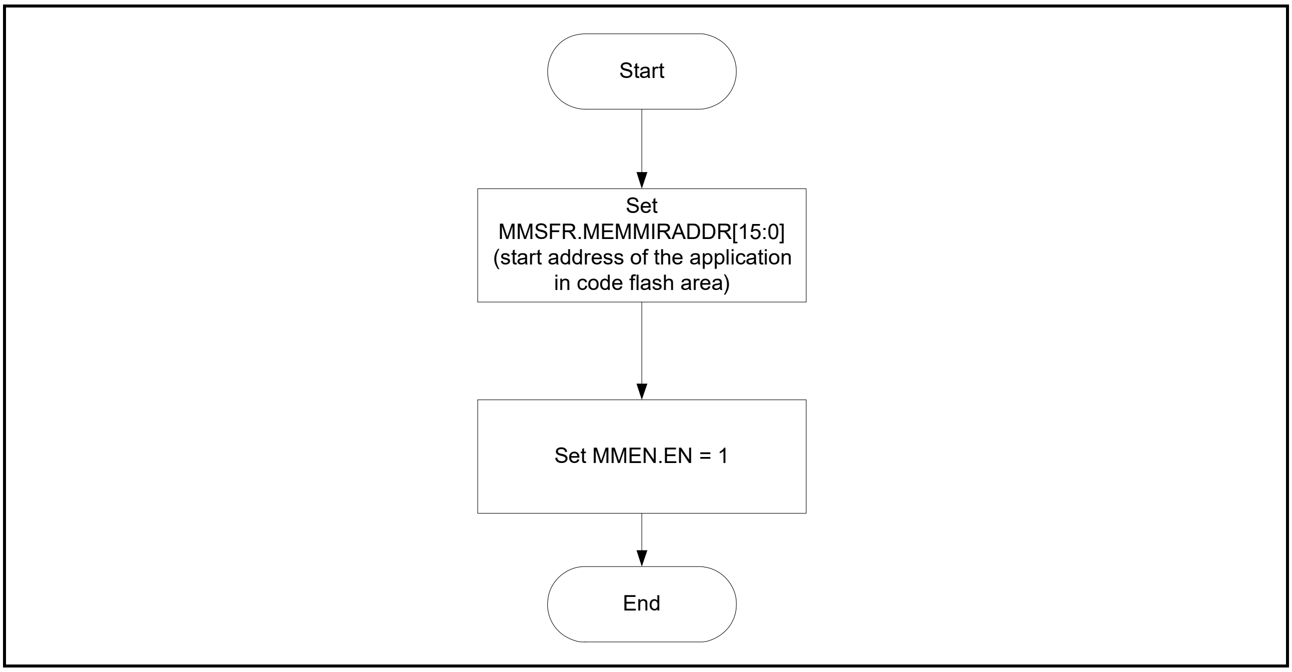


Figure 5.5 MMF setup flow

5.3.2 Setting Example

The target application code on the code flash can be accessed from address 0200 0000h on the memory mirror space by setting up the code flash start address in MMSFR.MEMMIRADDR and setting MMEN.EN = 1. Figure 5.6 shows an example of how to use the MMF.

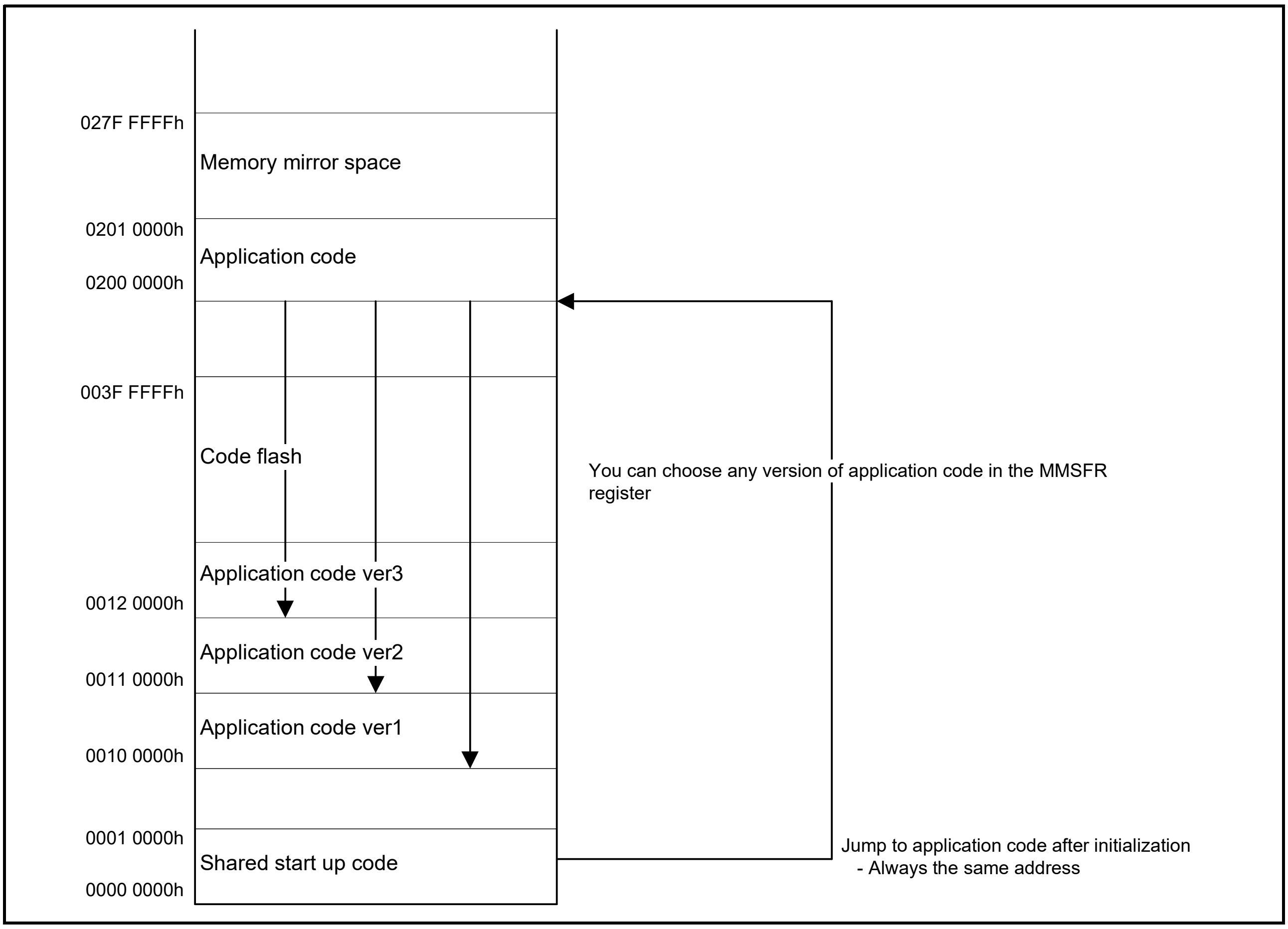


Figure 5.6 MMF setting example

- Set the MMSFR register to DB10 0000h to use the application code ver1
- Set the MMSFR register to DB11 0000h to use the application code ver2
- Set the MMSFR register to DB12 0000h to use the application code ver3.

6. Resets

6.1 Overview

The MCU provides 14 resets:

- RES pin reset
- Power-on reset
- Independent watchdog timer reset
- Watchdog timer reset
- Voltage monitor 0 reset
- Voltage monitor 1 reset
- Voltage monitor 2 reset
- SRAM parity error reset
- SRAM ECC error reset
- Bus master MPU error reset
- Bus slave MPU error reset
- Stack pointer error reset
- Deep software standby reset
- Software reset.

Table 6.1 lists the reset names and sources.

Table 6.1 Reset names and sources

Reset name	Source
RES pin reset	Voltage input to the RES pin is driven low
Power-on reset	VCC rise (voltage detection V_{POR})[*1]
Independent watchdog timer reset	IWDT underflow or refresh error
Watchdog timer reset	WDT underflow or refresh error
Voltage monitor 0 reset	VCC fall (voltage detection V_{det0})[*1]
Voltage monitor 1 reset	VCC fall (voltage detection V_{det1})[*1]
Voltage monitor 2 reset	VCC fall (voltage detection V_{det2})[*1]
SRAM parity error reset	SRAM parity error detection
SRAM ECC error reset	SRAM ECC error detection
Bus master MPU error reset	Bus master MPU error detection
Bus slave MPU error reset	Bus slave MPU error detection
Stack pointer error reset	Stack pointer error detection
Deep software standby reset	Canceling of Deep Software Standby mode by an interrupt
Software reset	Register setting (use the Arm® software reset bit AIRCR.SYSRESETREQ)

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, V_{det1}, and V_{det2}), see section 8, Low Voltage Detection (LVD) and section 60, Electrical Characteristics.

The internal state and pins are initialized by a reset. Table 6.2 and Table 6.3 list the targets initialized by resets.

Table 6.2 Reset detect flags initialized by each reset source

Flag to be initialized	Reset source							
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	×	×	×	×	×	×	×
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	✓	✓	×	×	×	×	×	×
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	✓	✓	✓	×	×	×	×	×
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	✓	✓	✓	×	×	×	×	×
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	✓	✓	✓	×	×	×	×	×
Voltage Monitor 2 Reset Detect Flag (RSTSR0.LVD2RF)	✓	✓	✓	×	×	×	×	×
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	×	×	×	×	×
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	✓	✓	✓	×	×	×	×	×
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	✓	✓	✓	×	×	×	×	×
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	✓	✓	✓	×	×	×	×	×
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	✓	✓	✓	×	×	×	×	×
Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	✓	✓	✓	×	×	×	×	×
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	✓	✓	✓	×	×	×	×	×
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	×	✓	×	×	×	×	×	×

Flag to be initialized	Reset source						
	SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	Deep Software Standby reset	
						DEEPCUT[0] = 0	DEEPCUT[0] = 1
Power-On Reset Detect Flag (RSTSR0.PORF)	×	×	×	×	×	×	×
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	×	×	×	×	×	×	×
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	×	×	×	×	×	✓	✓
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	×	×	×	×	×	✓	✓
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	×	×	×	×	×	×	×
Voltage Monitor 2 Reset Detect Flag (RSTSR0.LVD2RF)	×	×	×	×	×	×	×
Software Reset Detect Flag (RSTSR1.SWRF)	×	×	×	×	×	✓	✓
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	×	×	×	×	×	✓	✓
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	×	×	×	×	×	✓	✓
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	×	×	×	×	×	✓	✓
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	×	×	×	×	×	✓	✓
Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	×	×	×	×	×	✓	✓
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	×	×	×	×	×	×	×
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	×	×	×	×	×	×	×

✓: Initialized to 0, ×: Not initialized

Table 6.3 Module-related registers initialized by each reset source

Registers to be initialized		Reset source							
		RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓	✓	✓
Voltage monitor function 1 registers	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	✓	✓	✓	✓	✓	×	×	×
	LVD1CR1/LVD1SR	✓	✓	✓	✓	✓	×	×	×
Voltage monitor function 2 registers	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	✓	✓	✓	✓	✓	×	×	×
	LVD2CR1/LVD2SR	✓	✓	✓	✓	✓	×	×	×
SOSC register	SOSCCR	×	✓*1	×	×	×	×	×	×
	SOMCR	×	×	×	×	×	×	×	×
LOCO registers	LOCOCR	✓	✓	✓	✓	✓	✓	✓	✓
	LOCOUTCR	×	✓	✓	×	×	✓	✓	×
MOSC register	MOMCR	✓	✓	✓	✓	✓	✓	✓	✓
Realtime Clock (RTC) register*2		×	×	×	×	×	×	×	×
AGT registers		×	✓	✓	×	×	✓	✓	×
USBFS registers	Except DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓	✓
	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓	✓
USBHS registers	Except DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	✓	✓
	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	✓	✓
MPU register		✓	✓	✓	✓	✓	✓	✓	✓
Pin states (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	✓	✓	✓
Pin states (XCIN/XCOUT pin)		×	×	×	×	×	×	×	×
Low-power function registers	DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR2	✓	✓	✓	✓	✓	✓	✓	✓
Battery backup register		×	×	×	×	×	×	×	×
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	✓	✓	✓

Registers to be initialized		Reset source						
		SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	Deep Software Standby reset	
							DEEPCUT[0] = 0	DEEPCUT[0] = 1
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓	✓
Voltage monitor function 1 registers	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	×	×	×	×	×	×	×
	LVD1CR1/LVD1SR	×	×	×	×	×	✓	✓
Voltage monitor function 2 registers	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	×	×	×	×	×	×	×
	LVD2CR1/LVD2SR	×	×	×	×	×	✓	✓
SOSC register	SOSCCR	×	×	×	×	×	×	×
	SOMCR	×	×	×	×	×	×	×
LOCO register	LOCOCR	✓	✓	✓	✓	✓	✓	✓
	LOCOUTCR	×	×	×	×	×	×	✓
MOSC registers	MOMCR	✓	✓	✓	✓	✓	×	×
Realtime Clock (RTC) register*2		×	×	×	×	×	×	×
AGT register		×	×	×	×	×	×	✓
USBFS registers	Except DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓
	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	×	✓
USBHS registers	Except DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	✓	✓
	DPUSR0R, DPUSR1R, DPUSR2R, DPUSRCR	✓	✓	✓	✓	✓	×	✓
MPU register		✓	✓	×	×	×	✓	✓
Pin states (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	*3	*3
Pin states (XCIN/XCOUT pin)		×	×	×	×	×	×	×

Registers to be initialized		Reset source						
		SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	Deep Software Standby reset	
							DEEPCUT[0] = 0	DEEPCUT[0] = 1
Low-power function registers	DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR2	✓	✓	✓	✓	✓	×	×
Battery backup register		×	×	×	×	×	×	×
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	✓	✓

✓: Initialized, ×: Not initialized

Note 1. For the initial value of each register, see section 9, Clock Generation Circuit.

Note 2. The RTC has a software reset. RCR1.RTCOS, CIE and RCR2.RTCOE, ADJ30, and RESET are initialized by all types of resets. For details on the target bits, see section 26, Realtime Clock (RTC).

Note 3. Depends on the setting of DPSBYCR.IOKEEP.

The RTC is not initialized by any reset source. SOSC and LOCO can be selected as the clock source of RTC. The following are the states of SOSC and LOCO when a reset occurs.

Table 6.4 States of SOSC when a reset occurs

		Reset source	
		POR	Other
SOSC	Enable or disable	Initialized to enable	Continue with the state that was selected before the reset occurred
	Drive capability	Continue with the state that was selected before the reset occurred	

Table 6.5 States of LOCO when a reset occurs

		Reset source	
		POR, LVD0, LVD1, LVD2/ Deep Software Standby (DEEPCUT[0] = 1)	Other
LOCO	Enable or disable	Initialized to enable	
	Oscillation accuracy*1	Initialized to accuracy before trimming by LOCOUTCR (accuracy: ± 15%)	Continue with the accuracy that was trimmed by LOCOUTCR

Note 1. The LOCO User Trimming Control Register (LOCOUTCR) is reset by POR, LVD0, LVD1, LVD2, and Deep Software Standby (DEEPCUT[0] = 1) resets, returning the LOCO to the default oscillation accuracy. This can affect RTC accuracy if the RTC uses the LOCO (with a user trimming value in LOCOUTCR) as the RTC source clock. To restore the pre-reset LOCO oscillation accuracy, reload the required trimming value into LOCOUTCR after any of these resets.

Table 6.6 lists the pin related to the reset function.

Table 6.6 Reset I/O pin

Pin name	I/O	Function
RES	Input	Reset pin

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): SYSTEM.RSTSR0 4001 E410h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSRSTF	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	x*1	0	0	0	x*1	x*1	x*1	x*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	PORF	Power-On Reset Detect Flag	0: Power-on reset not detected 1: Power-on reset detected.	R(/W)*2
b1	LVD0RF	Voltage Monitor 0 Reset Detect Flag	0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected.	R(/W)*2
b2	LVD1RF	Voltage Monitor 1 Reset Detect Flag	0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected.	R(/W)*2
b3	LVD2RF	Voltage Monitor 2 Reset Detect Flag	0: Voltage monitor 2 reset not detected 1: Voltage monitor 2 reset detected.	R(/W)*2
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSRSTF	Deep Software Standby Reset Flag	0: Deep Software Standby mode cancellation not requested by an interrupt 1: Deep Software Standby mode cancellation requested by an interrupt.	R(/W)*2

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written, to clear the flag. The flag must be cleared by writing 0 after 1 is read.

PORF flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset occurred.

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to PORF.

LVD0RF flag (Voltage Monitor 0 Reset Detect Flag)

The LVD0RF flag indicates that the VCC voltage fell below V_{det0}.

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to LVD0RF.

LVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

The LVD1RF flag indicates that the VCC voltage fell below V_{det1}.

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to LVD1RF.

LVD2RF flag (Voltage Monitor 2 Reset Detect Flag)

The LVD2RF flag indicates that the VCC voltage fell below V_{det2}.

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to LVD2RF.

DPSRSTF flag (Deep Software Standby Reset Flag)

The DPSRSTF flag indicates that Deep Software Standby mode was canceled by an external or internal interrupt and that an internal reset (Deep Software Standby reset) occurred when an exception from Deep Software Standby mode occurred.

[Setting condition]

- When Deep Software Standby mode is canceled by an external or internal interrupt. For details, see section 11, Low Power Modes.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to DPSRSTF.

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): SYSTEM.RSTSR1 4001 E0C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPERF	BUSMRF	BUSSRF	REERF	RPERF	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	0	0	0	x*1	x*1	x*1	x*1	x*1	0	0	0	0	0	x*1	x*1	x*1

Bit	Symbol	Bit name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R(/W) *2
b1	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog Timer reset not detected 1: Watchdog Timer reset detected.	R(/W) *2
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected 1: Software reset detected.	R(/W) *2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	RPERF	SRAM Parity Error Reset Detect Flag	0: SRAM parity error reset not detected 1: SRAM parity error reset detected.	R(/W) *2
b9	REERF	SRAM ECC Error Reset Detect Flag	0: SRAM ECC error reset not detected 1: SRAM ECC error reset detected.	R(/W) *2
b10	BUSSRF	Bus Slave MPU Error Reset Detect Flag	0: Bus slave MPU error reset not detected 1: Bus slave MPU error reset detected.	R(/W) *2
b11	BUSMRF	Bus Master MPU Error Reset Detect Flag	0: Bus master MPU error reset not detected 1: Bus master MPU error reset detected.	R(/W) *2

Bit	Symbol	Bit name	Description	R/W
b12	SPERF	SP Error Reset Detect Flag	0: SP error reset not detected 1: SP error reset detected.	R(/W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

IWDTRF flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset occurred.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to IWDTRF.

WDTRF flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset occurred.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to WDTRF.

SWRF flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset occurred.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to SWRF.

RPERF flag (SRAM Parity Error Reset Detect Flag)

The RPERF flag indicates that a SRAM parity error reset occurred.

[Setting condition]

- When an SRAM parity error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to RPERF.

REERF flag (SRAM ECC Error Reset Detect Flag)

The REERF flag indicates that an SRAM ECC error reset occurred.

[Setting condition]

- When an SRAM ECC error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to REERF.

BUSSRF flag (Bus Slave MPU Error Reset Detect Flag)

The BUSSRF flag indicates that a bus slave MPU error reset occurred.

[Setting condition]

- When a bus slave MPU error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to BUSSRF.

BUSMRF flag (Bus Master MPU Error Reset Detect Flag)

The BUSMRF flag indicates that a bus master MPU error reset occurred.

[Setting condition]

- When a bus master MPU error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to BUSMRF.

SPERF flag (SP Error Reset Detect Flag)

The SPERF flag indicates that a stack pointer error reset occurred.

[Setting condition]

- When a stack pointer error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to SPERF.

6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): SYSTEM.RSTSR2 4001 E411h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	x*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start.	R(/W) *2
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written, to set the flag.

RSTSR2 determines whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

CWSF flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing, either cold start or warm start. The CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES pin.

[Setting condition]

- When 1 is written through the software. Writing 0 to CWSF does not set it to 0.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

6.3 Operation

6.3.1 RES Pin Reset

The RES pin generates this reset. When the RES pin is driven low, all the processing in progress is aborted and the MCU enters a reset state. To successfully reset the MCU, the RES pin must be held low for the power supply stabilization time specified at power-on.

When the RES pin is driven high from low, the internal reset is canceled after the post-RES cancellation wait time (t_{RESWT}) elapses. The CPU then starts the reset exception handling.

For details, see section 60, Electrical Characteristics.

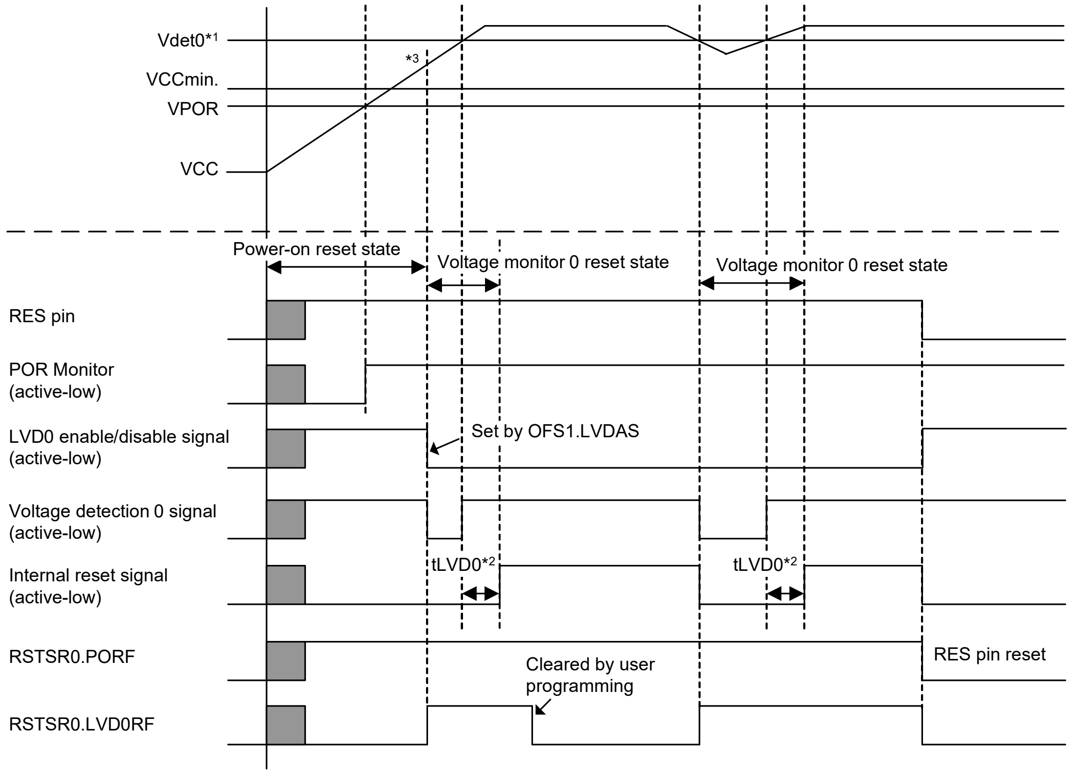
6.3.2 Power-On Reset

The power-on reset circuit generates this internal reset. If the RES pin is in a high-level state when power is supplied, a power-on reset is generated. After VCC exceeds VPOR and the specified period (power-on reset time) elapses, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is the stabilization period for the external power supply and the MCU circuit. After a power-on reset is generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES pin reset.

The voltage monitor 0 reset is an internal reset generated by the voltage monitor circuit. If the Voltage Detection 0 Circuit Start bit (LVDAS) in Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.LVD0RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitor 0 reset is to be used.

After VCC exceeds V_{det0} and the voltage monitor 0 reset time (tLVD0) elapses, the internal reset is canceled and the CPU starts the reset exception handling. The V_{det0} voltage detection level can be changed by the setting in the VDSEL[1:0] bits in Option Function Select Register 1 (OFS1).

Figure 6.1 shows an example of operations during a power-on reset and voltage monitor 0 reset.



Note: For details on the electrical characteristics, see section 60, Electrical Characteristics.

Note 1. V_{det0} shows a voltage monitor 0 reset detection level, VPOR shows a power-on reset detection level, and VCCmin shows minimum guaranteed voltage of MCU.

Note 2. tLVD0 shows a time for voltage monitor 0 reset.

Note 3. At power-on, VCC should have risen to the minimum guaranteed voltage before the POR reset is released.

Figure 6.1 Example of operations during power-on and voltage monitor 0 resets

6.3.3 Voltage Monitor Reset

The voltage monitor circuit generates this internal reset. If the Voltage Detection 0 Circuit Start bit (LVDAS) in Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.LVD0RF flag is set to 1 and the voltage detection circuit generates voltage monitor 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitor 0 reset is to be used. After VCC exceeds V_{det0} and the voltage monitor 0 reset time (tLVD0) elapses, the internal reset is canceled and the CPU starts the reset exception handling.

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitor 1 circuit mode select bit (LVD1CR0.RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 1 reset if VCC falls to or below V_{det1}.

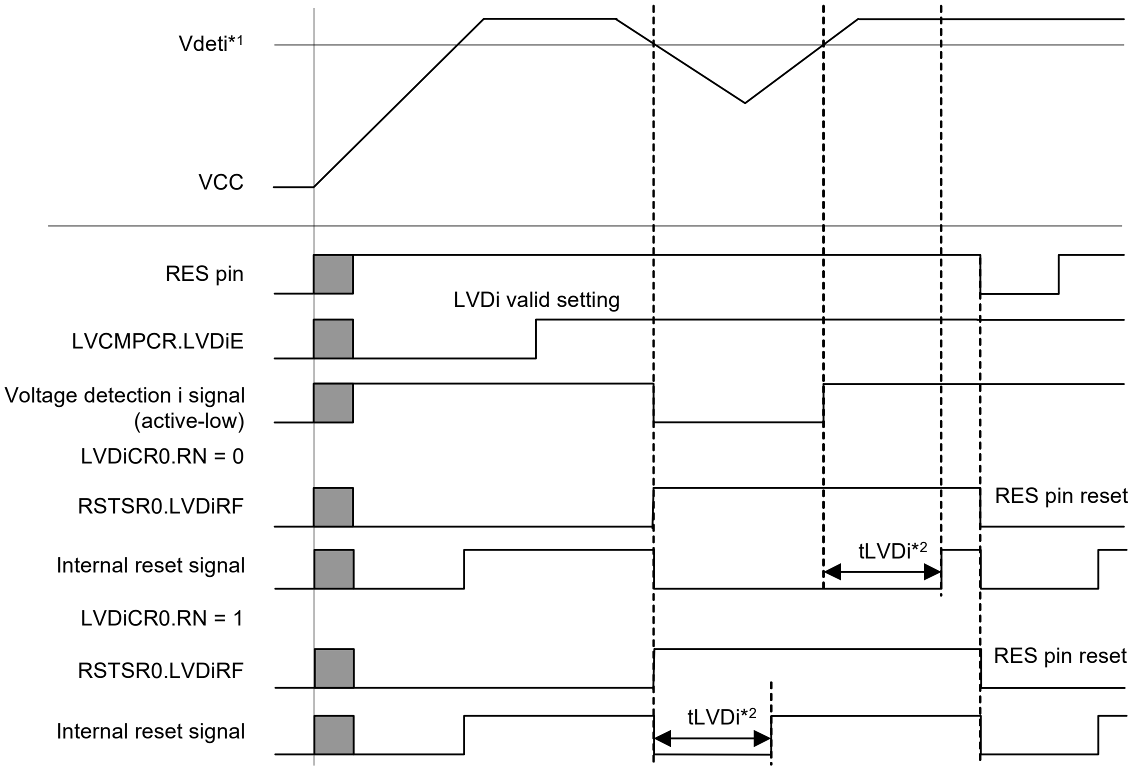
Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0), the RSTSR0.LVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 2 reset if VCC falls to or below V_{det2}.

Similarly, timing for release from the voltage monitor 1 reset state is selectable in the Voltage Monitor 1 Reset Negate Select bit (RN) in LVD1CR0. When the RN bit is 0 and VCC falls to or below V_{det1}, the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (tLVD1) elapses after VCC rises above V_{det1}. When the LVD1CR0.RN bit is 1 and VCC falls to or below V_{det1}, the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (tLVD1) elapses.

Likewise, timing for release from the voltage monitor 2 reset state is selectable in the Voltage Monitor 2 Reset Negate Select bit (RN) in the LDV2CR0 register.

Detection levels V_{det1} and V_{det2} can be changed in the Voltage Detection Level Select Register (LDVLVLR).

Figure 6.2 shows examples of operations during voltage monitor 1 and 2 resets. For details on the voltage monitor 1 reset and voltage monitor 2 reset, see section 8, Low Voltage Detection (LVD).



Note: For details on the electrical characteristics, see section 60, Electrical Characteristics.
Note 1. Vdeti indicates the detection level of a voltage monitor 1 reset and voltage monitor 2 reset. (i = 1, 2)
Note 2. tLVDi indicates the time for a voltage monitor 1 reset and voltage monitor 2 reset. (i = 1, 2)

Figure 6.2 Example operations during voltage monitor 1 and voltage monitor 2 resets

6.3.4 Deep Software Standby Reset

This internal reset is generated when Deep Software Standby mode is canceled by an associated interrupt. The Deep Software Standby reset is canceled after tDSBY (return time after Deep Software Standby mode cancellation) elapses. At the same time, Deep Software Standby mode is also canceled.

When tDSBYWT (wait time after Deep Software Standby mode cancellation) elapses after Deep Software Standby mode is canceled, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the Deep Software Standby reset, see section 11, Low Power Modes.

6.3.5 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated from the Independent Watchdog Timer (IWDT). Output of the reset from the IWDT can be selected in the Option Function Select Register 0 (OFS0).

When output of the independent watchdog timer reset is selected, the reset is generated if the IWDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) elapses after the independent watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the independent watchdog timer reset, see section 28, Independent Watchdog Timer (IWDT).

6.3.6 Watchdog Timer Reset

The watchdog timer generates this internal reset. Output of the reset from the WDT can be selected in the WDT Reset Control Register (WDTRCR) or Option Function Select Register 0 (OFS0).

When output of the watchdog timer reset is selected, the reset is generated if the WDT underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) elapses after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the watchdog timer reset, see section 27, Watchdog Timer (WDT).

6.3.7 Software Reset

This internal reset is generated by a software setting of the SYSRESETREQ bit in the AIRCR register in the Arm core. When the SYSRESETREQ bit is set to 1, a software reset is generated. When the internal reset time (tRESW2) elapses after the software reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the SYSRESETREQ bit, see the *ARM® Cortex®-M4 Technical Reference Manual*.

6.3.8 Determination of Cold/Warm Start

Read the CWSF flag in RSTSR2 to determine the cause of reset processing. The flag indicates whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

The flag is set to 0 when a power-on reset occurs (cold start). Otherwise, the flag is not set to 0. The flag is set to 1 when 1 is written to it through software. It is not set to 0 even on writing 0 to it.

Figure 6.3 shows an example of a cold/warm start determination operation.

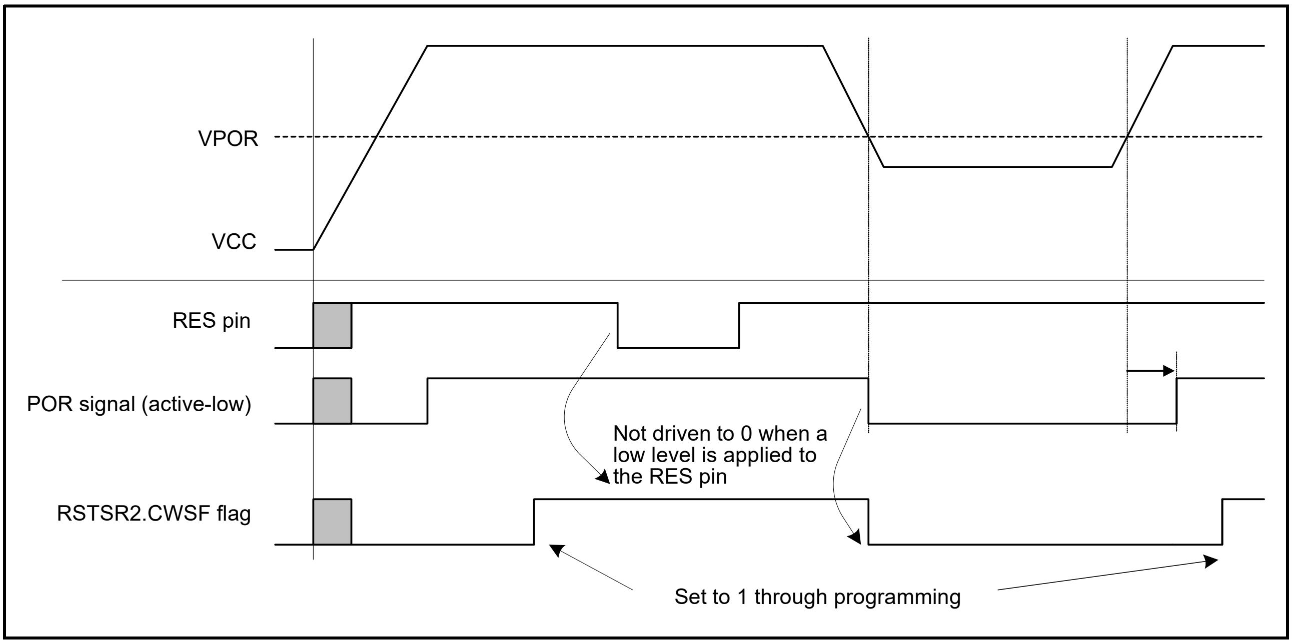
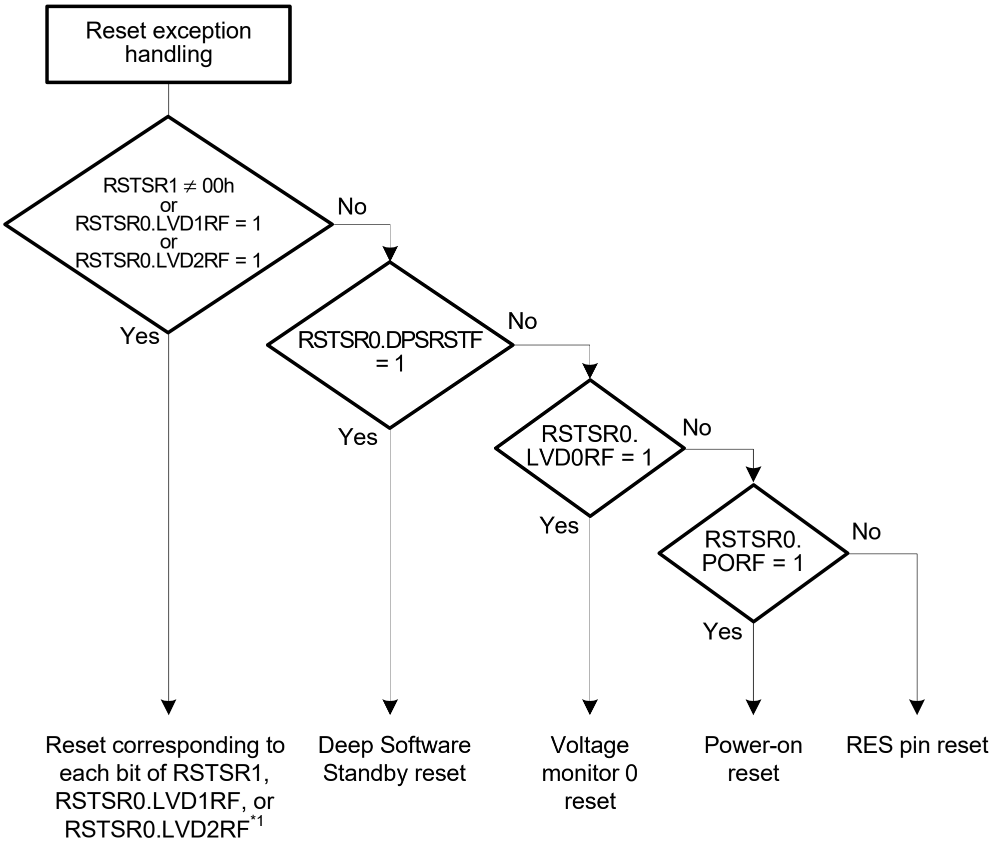


Figure 6.3 Example of cold/warm start determination operation

6.3.9 Determination of Reset Generation Source

Read RSTSR0 and RSTSR1 to determine which reset is used to execute the reset exception handling. Figure 6.4 shows an example of the flow for identifying a reset generation source. The reset flag must be written with 0 after it is read as 1.



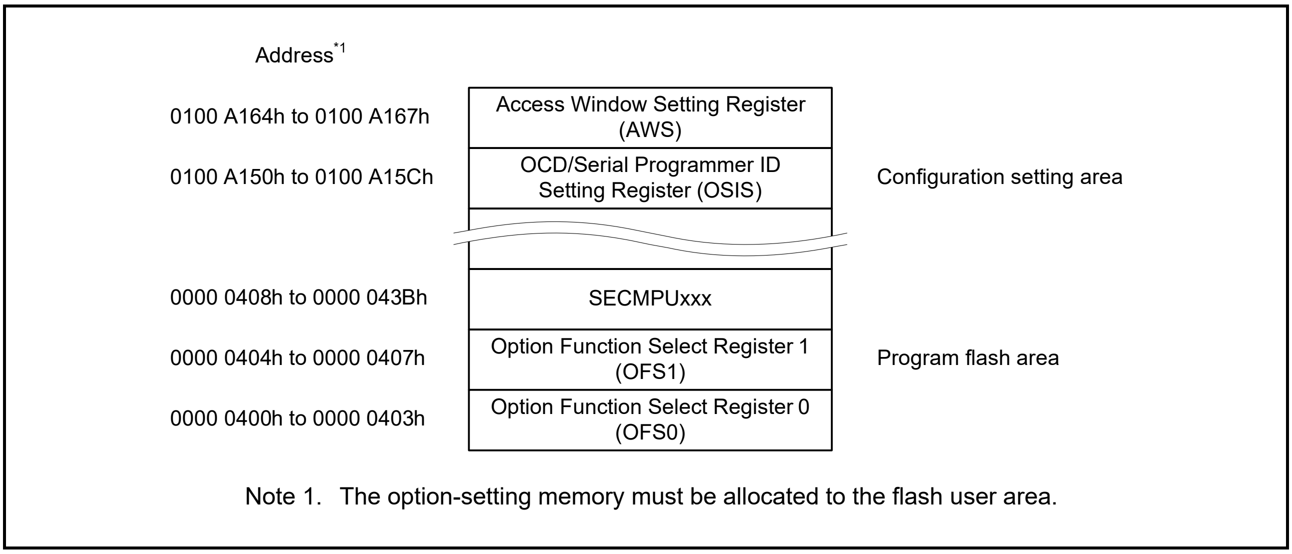
Note 1. If two or more resets associated with the RSTSR1, RSTSR0.LVD1RF, or RSTSR0.LVD2RF bits occur at the same time, all reset flags are set to 1.

Figure 6.4 Example of reset generation source determination flow

7. Option-Setting Memory

7.1 Overview

The option-setting memory determines the state of the MCU after a reset. It is allocated to the configuration setting area and the program flash area of the flash memory, and the available methods of setting are different for the two areas. Figure 7.1 shows the option-setting memory area.



Note 1. The option-setting memory must be allocated to the flash user area.

Figure 7.1 Option-setting memory area

7.2 Register Descriptions

7.2.1 Option Function Select Register 0 (OFS0)

Address(es): OFS0 0000 0400h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	WDTSTPCTL	—	WDTRSTIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTSTRT	—

Value after reset: User setting*1

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	IWDTSTPCTL	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—

Value after reset: User setting*1

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	When read, this bit returns the written value. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: Automatically activate IWDT after a reset (auto start mode) 1: Disable IWDT.	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh).	R

Bit	Symbol	Bit name	Description	R/W
b7 to b4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: × 1 0 0 1 0: × 1/16 0 0 1 1: × 1/32 0 1 0 0: × 1/64 1 1 1 1: × 1/128 0 1 0 1: × 1/256. Other settings are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting).	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting).	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Enable non-maskable interrupt requests or interrupt requests 1: Enable resets.	R
b13	—	Reserved	When read, this bit returns the written value. The write value should be 1.	R
b14	IWDTSTPCTL	IWDT Stop Control	0: Continue counting 1: Stop counting when in Sleep, Snooze mode, Software Standby, or Deep Software Standby mode.	R
b16, b15	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b17	WDTSTRT	WDT Start Mode Select	0: Automatically activate WDT after a reset (auto start mode) 1: Stop WDT after a reset (register start mode).	R
b19, b18	WDTTOPS[1:0]	WDT Timeout Period Select	b19 b18 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh).	R
b23 to b20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: PCLKB divided by 4 0 1 0 0: PCLKB divided by 64 1 1 1 1: PCLKB divided by 128 0 1 1 0: PCLKB divided by 512 0 1 1 1: PCLKB divided by 2048 1 0 0 0: PCLKB divided by 8192. Other settings are prohibited.	R
b25, b24	WDTRPES[1:0]	WDT Window End Position Select	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting).	R
b27, b26	WDTRPSS[1:0]	WDT Window Start Position Select	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting).	R
b28	WDTRSTIRQS	WDT Reset Interrupt Request Select	WDT Behavior Select 0: NMI 1: Reset.	R
b29	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b30	WDTSTPCTL	WDT Stop Control	0: Continue counting 1: Stop counting when entering Sleep mode.	R
b31	—	Reserved	When read, these bits return the written value. The write value should be 1.	R

Note 1. The value in a blank product is FFFF FFFFh. It is set to the value written by your application.

IWDTSTRT bit (IWDT Start Mode Select)

The IWDTSTRT bit selects the mode in which the IWDT is activated after a reset (stopped state or activated state).

IWDTTOPS[1:0] bits (IWDT Timeout Period Select)

The IWDTTOPS[1:0] bits specify the timeout period, the time it takes for the down counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set in the IWDTCKS[3:0] bits. The number of clock cycles that the counter takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTCKS[3:0] bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

The IWDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of the clock for the IWDT as 1/1, 1/16, 1/32, 1/64, 1/128, or 1/256. Using this setting combined with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 128 to 524,288 IWDT clock cycles.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTRPES[1:0] bits (IWDT Window End Position Select)

The IWDTRPES[1:0] bits specify the position where the window for the down counter ends as 0%, 25%, 50%, or 75% of the count value. The value of the window end position must be smaller than the value of the window start position. Otherwise, only the value for the window start position is valid.

The counter values corresponding to the settings for the start and end positions of the window, in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits, vary with the setting in the IWDTTOPS[1:0] bits.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTRPSS[1:0] bits (IWDT Window Start Position Select)

The IWDTRPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the window start and end positions becomes the period in which a refresh is possible. Refresh is not possible outside this period.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit selects the operation on an underflow of the down counter or generation of a refresh error. The selected operation can be an independent watchdog timer reset, non-maskable interrupt request, or interrupt request.

For details, see section 28, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit specifies whether to stop counting when entering Sleep, Snooze, or Software Standby mode.

For details, see section 28, Independent Watchdog Timer (IWDT).

WDTSTRT bit (WDT Start Mode Select)

The WDTSTRT bit selects the mode in which the WDT is activated after a reset (stopped state or activated state). When the WDT is activated in auto start mode, the OFS0 register setting for the WDT is effective.

WDTTOPS[1:0] bits (WDT Timeout Period Select)

The WDTTOPS[1:0] bits specify the timeout period, the time it takes for the down counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDTCKS[3:0] bits. The number of PCLKB cycles that the counter takes to underflow after a refresh operation is determined by the combination of the WDTCKS[3:0] and WDTTOPS[1:0] bits.

For details, see section 27, Watchdog Timer (WDT).

WDTCKS[3:0] bits (WDT Clock Frequency Division Ratio Select)

The WDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the PCLKB frequency as 1/4, 1/64, 1/128, 1/512, 1/2048, or 1/8192. Using this setting combined with the WDTTOPS[1:0] bit setting, the WDT counting period can be set from 4,096 to 134,217,728 PCLKB cycles.

For details, see section 27, Watchdog Timer (WDT).

WDTRPES[1:0] bits (WDT Window End Position Select)

The WDTRPES[1:0] bits specify the position where the window for the down counter ends as 0%, 25%, 50%, or 75% of the counted value. The value of the window end position must be smaller than the value of the window start position. Otherwise, only the value for the window start position is valid.

The counter values corresponding to the settings for the start and end positions of the window, in the WDTRPSS[1:0] and WDTRPES[1:0] bits, vary with the setting of the WDTTOPS[1:0] bits.

For details, see section 27, Watchdog Timer (WDT).

WDTRPSS[1:0] bits (WDT Window Start Position Select)

The WDTRPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the window start and end positions becomes the period in which a refresh is possible. Refresh is not possible outside this period.

For details, see section 27, Watchdog Timer (WDT).

WDTRSTIRQS bit (WDT Reset Interrupt Request Select)

The WDTRSTIRQS bit selects the operation on an underflow of the down counter or generation of a refresh error. The selected operation can be a watchdog timer reset, non-maskable interrupt request, or interrupt request.

For details, see section 27, Watchdog Timer (WDT).

WDTSTPCTL bit (WDT Stop Control)

The WDTSTPCTL bit specifies whether to stop counting when entering Sleep mode. For details, see section 27, Watchdog Timer (WDT).

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): OFS1 0000 0404h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Value after reset: User setting*1

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	HOCOFRQ0[1:0]		HOCOEN	—	—	—	—	—	LVDAS	VDSEL0[1:0]	

Value after reset: User setting*1

Bit	Symbol	Bit name	Description	R/W
b1, b0	VDSEL0[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: Setting prohibited 0 1: Select 2.94 V 1 0: Select 2.87 V 1 1: Select 2.80 V.	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset.	R
b7 to b3	—	Reserved	When read, these bits return the written value. The write value should be 1.	R

Bit	Symbol	Bit name	Description	R/W
b8	HOCOEN	HOCO Oscillation Enable	0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset.	R
b10, b9	HOCOFRQ0[1:0]	HOCO Frequency Setting 0	b10 b9 0 0: 16 MHz 0 1: 18 MHz 1 0: 20 MHz 1 1: Setting prohibited.	R
b31 to b11	—	Reserved	When read, these bits return the written value. The write value should be 1.	R

Note 1. The value in a blank product is FFFF FFFFh. It is set to the value written by your application.

VDSEL0[1:0] bits (Voltage Detection 0 Level Select)

The VDSEL0[1:0] bits select the voltage detection level of the voltage detection 0 circuit.

LVDAS bit (Voltage Detection 0 Circuit Start)

The LVDAS bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

HOCOEN bit (HOCO Oscillation Enable)

The HOCOEN bit selects whether the HOCO oscillation is enable or disable after a reset. Setting this bit to 0 allows the HOCO oscillation to start before the CPU starts operation, which reduces the wait time for oscillation stabilization.

Note: When the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is only switched to HOCO by setting the clock source select bits (SCKSCR.CKSEL[2:0]). If you use the HOCO clock, you must set the OFS1.HOCOFRQ0 bit to an optimum value.

HOCOFRQ0[1:0] bits (HOCO Frequency Setting 0)

The HOCOFRQ0[1:0] bits specify the HOCO frequency after a reset as 16, 18, or 20 MHz.

7.2.3 Access Window Setting Register (AWS)

Address(es): AWS 0100 A164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTFLG	—	—	—	—	FAWE[10:0]										
Value after reset:	User setting															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSPR	—	—	—	—	FAWS[10:0]										
Value after reset:	User setting															

Bit	Symbol	Bit name	Description	R/W
b10 to b0	FAWS[10:0]	Access Window Start Block Address	These bits specify the start block address for the access window. They do not represent the block number of the access window. The access window is only valid in the program flash area. The block address specifies the first address of the block and consists of address bits [23:13].	R
b14 to b11	—	Reserved	When read, these bits return the written value. The write value should be 1.	R

Bit	Symbol	Bit name	Description	R/W
b15	FSPR	Protection of Access Window and Startup Area Select Function	This bit controls programming of the program/erase protection for the access window, the Startup Area Select flag (AWS.BTFLG), and the temporary boot swap. Once this bit is set to 0, it is impossible to change this bit to 1. 0: Executing the configuration setting command for programming the access window (FAWE[10:0], FAWS[10:0]) and the Startup Area Select flag (AWS.BTFLG) is invalid 1: Executing the configuration setting command for programming the access window (FAWE[10:0], FAWS[10:0]) and the Startup Area Select flag (AWS.BTFLG) is valid.	R
b26 to b16	FAWE[10:0]	Access Window End Block Address	These bits specify the end block address for the access window. They do not represent the block number of the access window. The access window is only valid in the program flash area. The end block address for the access window is the next block to the region acceptable for programming and erasure defined by the access window. The block address specifies the first address of the block and consists of the address bits [23:13].	R
b30 to b27	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b31	BTFLG	Startup Area Select Flag	This bit specifies whether the address of the startup area is exchanged for the boot swap function or not: 0: Exchange the first 8-KB area (0000 0000h to 0000 1FFFh) and second 8-KB area (0000 2000h to 0000 3FFFh) 1: Do not exchange the first 8-KB area (0000 0000h to 0000 1FFFh) and second 8-KB area (0000 2000h to 0000 3FFFh).	R

Issuing the program or erase (P/E) command to the area outside the access window causes a command-locked state. The access window is only valid in the program flash area. The access window provides protection in self-programming mode, serial programming mode, and on-chip debug mode. The access window can be locked by the FSPR bit.

The access window is specified in FAWS[10:0] and FAWE[10:0]:

- FAWE[10:0] = FAWS[10:0]: The P/E command is allowed to execute in the full program flash area
- FAWE[10:0] > FAWS[10:0]: The P/E command is only allowed to execute into the window from the block pointed to by the FAWS bits to the block one lower than the block pointed to by the FAWE bits
- FAWE[10:0] < FAWS[10:0]: The P/E command is not allowed to execute in the program flash area.

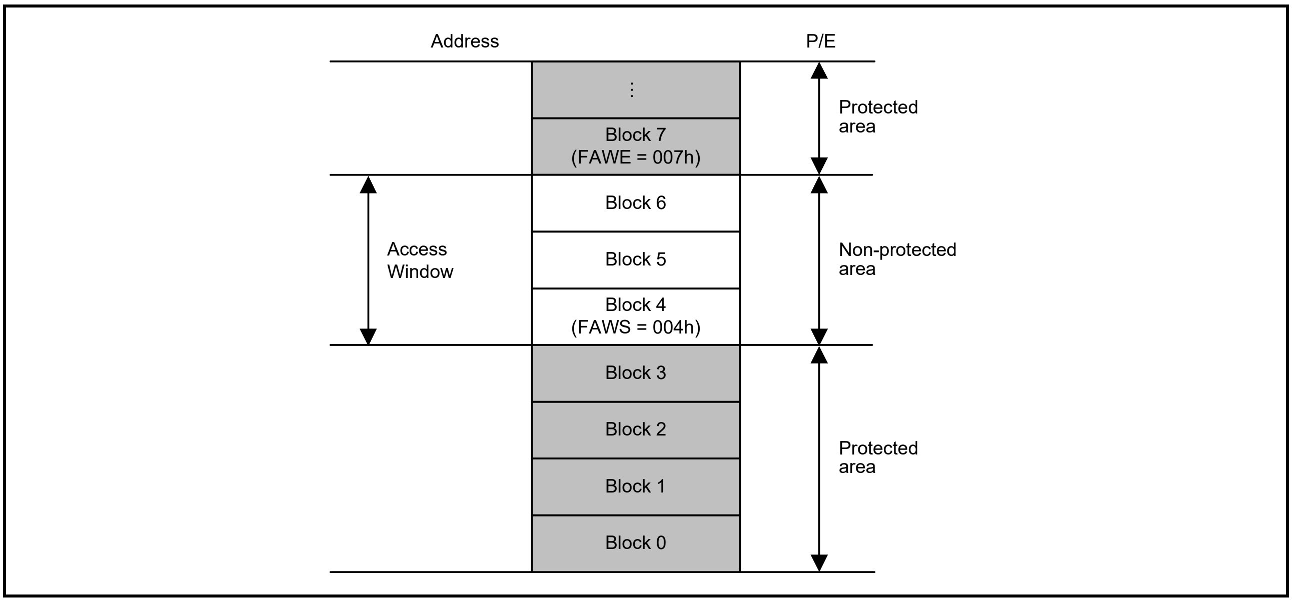


Figure 7.2 **Access window overview**

7.2.4 OCD/Serial Programmer ID Setting Register (OSIS)

The OSIS register stores the ID for ID code protection of the OCD/serial programmer. When connecting the OCD/serial programmer, write values so that the MCU can determine whether to permit the connection. Use this register to check whether a code transmitted from the OCD/serial programmer matches the ID code in the option-setting memory.

When the ID codes match, connection of the OCD/serial programmer is permitted. If the ID codes do not match, connection with the OCD/serial programmer is not possible. The OSIS register must be set in 32-bit units.

Address(es): OSIS 0100 A150h, OSIS 0100 A154h, OSIS 0100 A158h, OSIS 0100 A15Ch

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16

Value after reset: User setting

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

Value after reset: User setting

These fields hold the ID for use in ID authentication for the OCD/serial programmer.

ID code bits [127] and [126] determine whether ID code protection is enabled and the method of authentication to use with the host. Table 7.1 shows how ID code determines the method of authentication.

Table 7.1 Specifications for ID code protection

Operating mode on boot up	ID code	State of protection	Operations on connection to programmer or on-chip debugger
Serial programming mode (SCI/USB boot mode) On-chip debug mode (JTAG/SWD boot mode)	FFh, …, FFh (all bytes are FFh)	Protection disabled	The ID code is not checked, ID code always matches, and connection to programmer or on-chip debugger is permitted
	Bit [127] = 1 and [126] = 1, and at least one of the 16 bytes is not FFh.	Protection enabled	Matching ID code = Authentication is complete and connection with the programmer or the on-chip debugger is permitted. Non-matching ID code = Transition to the ID code protection wait state. When the ID code sent from the programmer or the on-chip debugger is ALeRASE in ASCII code (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFFh), the contents of the user flash (code and data) area, and the configuration area are erased. However, forced erasure is not executed when the FSPR bit is 0.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code = Authentication is complete and connection with the programmer or the on-chip debugger is permitted. Non-matching ID code = Transition to the ID code protection wait state.
	Bit [127] = 0	Protection enabled	The ID code is not checked, the ID code is always non-matching, and connection to the programmer or the on-chip debugger is prohibited

7.3 Setting the Option-Setting Memory

7.3.1 Allocation of Data in the Option-Setting Memory

Data for programming in the option-setting memory should be allocated to the addresses shown in Figure 7.1. The allocation of data is for use by tools such as a flash programming software or an on-chip debugger.

Note: Programming formats vary depending on the compiler. See the compiler manual for details.

7.3.2 Setting Data for Programming the Option-Setting Memory

Allocating data as described in section 7.3.1, Allocation of Data in the Option-Setting Memory does not alone result in the data being written to the option-setting memory. You must also follow one of the actions described in this section.

(1) Changing the option-setting memory by self-programming

To write data to the program flash area, use the programming command. To write data to the option-setting memory in the configuration setting area, use the configuration setting command. In addition, use startup area select function to safely update the boot program that includes the option-setting memory.

For details on the programming command, the configuration setting command, and the startup area select function, see section 55, Flash Memory.

Note: While programming the configuration setting area, the following restrictions apply:

- The code must not access addresses that satisfy the ranges described by the expression defined in Expression 1 from all bus masters
- The code must not execute on addresses that satisfy the ranges described by the expression defined in Expression 1.

Expression 1

If (((address & 0x0101F800) = = 0x01010000) || ((address & 0x0101FC00) = = 0x01012000))

For example, the ranges of addresses 0x1FFF0000 to 0x1FFF07FF or 0x1FFF2000 to 0x1FFF23FF are associated with the SRAMHS area that is tagged as restricted. Also, interrupts are allowed, however, the ISR has these specified restrictions. Therefore, it is highly recommended that you disable all interrupts, and bus masters except the CPU while programming the configuration setting area because the interrupts and these modules might access prohibited area in Expression 1.

(2) Debugging through an OCD or programming by a flash writer

This procedure depends on the tool in use, so refer to the tool manual for details. There are two setting procedures:

- Read the data, allocated as described in section 7.3.1, Allocation of Data in the Option-Setting Memory, from an object file or Motorola S-format file generated by the compiler, and write the data to the MCU
- Use the GUI interface of the tool to program the same data, allocated as described in section 7.3.1, Allocation of Data in the Option-Setting Memory.

Note: While programming the OSIS or AWS registers, the following restrictions apply:

- The code must not access addresses that satisfy the ranges described by the expression defined in Expression 1 from all bus masters
- The code must not execute on addresses that satisfy the ranges described by the expression defined in Expression 1.

7.4 Usage Notes

7.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory

When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 to all bits in reserved areas and all reserved bits. Operation is not guaranteed if 0 is written to these bits.

8. Low Voltage Detection (LVD)

8.1 Overview

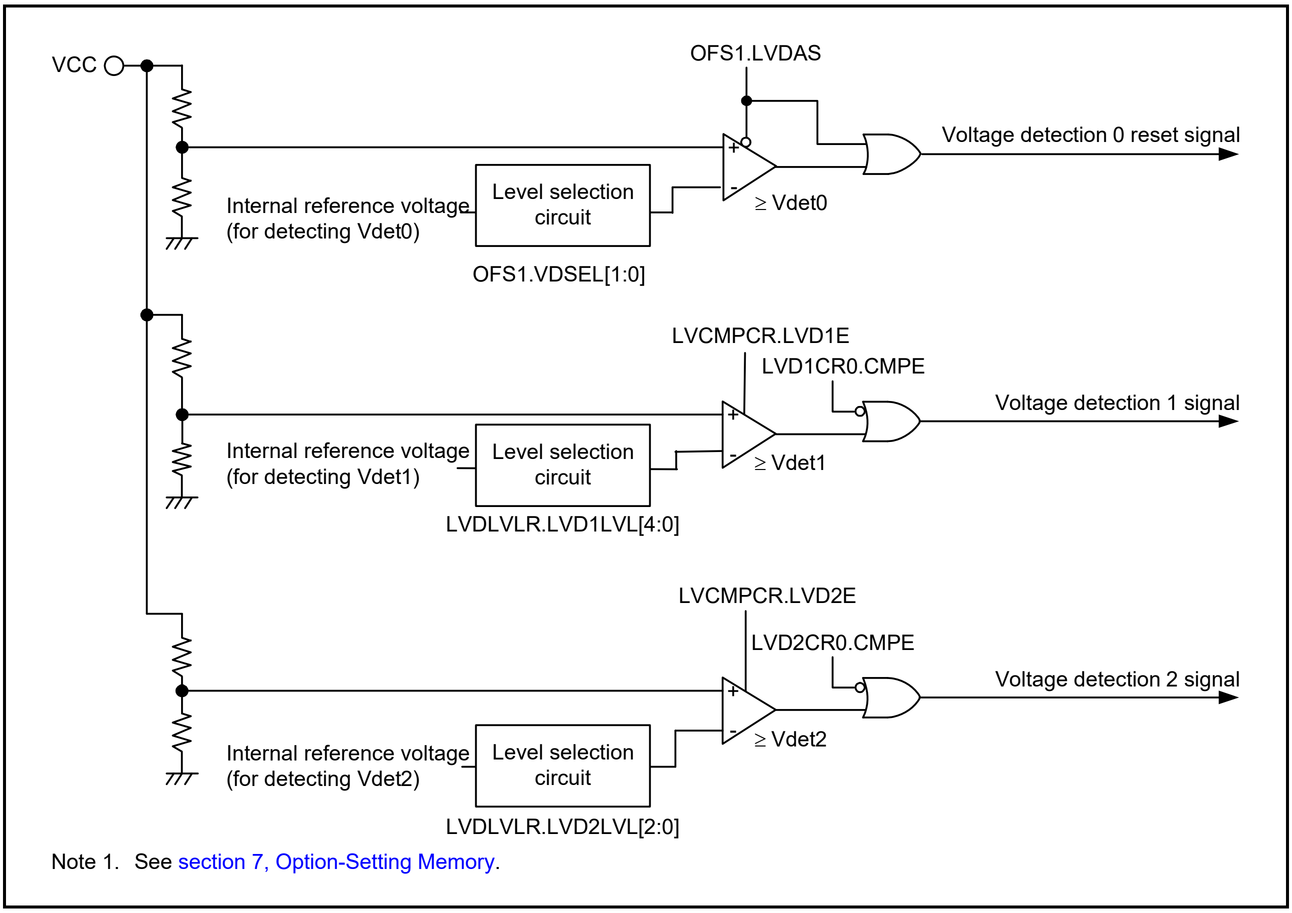
The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin, and the detection level can be selected using a software program. The LVD module consists of three separate voltage level detectors, 0, 1, and 2, which measure the voltage level input to the VCC pin. LVD voltage detection registers allow your application to configure detection of VCC changes at various voltage thresholds.

Each voltage level detector has a voltage monitor associated with it, for example voltage monitors 0, 1, and 2. Voltage monitor registers are used to configure the LVD to trigger an interrupt, event link output, or reset when the thresholds are crossed.

Table 8.1 lists the LVD specifications. Figure 8.1 shows a block diagram of voltage detectors 0, 1, and 2, Figure 8.2 shows a block diagram of the voltage monitor 1 interrupt and reset circuit, and Figure 8.3 shows a block diagram of the voltage monitor 2 interrupt and reset circuit.

Table 8.1 LVD specifications

Parameter		Voltage monitor 0	Voltage monitor 1	Voltage monitor 2
VCC monitoring	Monitored voltage	V_{det0}	V_{det1}	V_{det2}
	Detected event	Voltage falls past V_{det0}	Voltage rises or falls past V_{det1}	Voltage rises or falls past V_{det2}
	Detection voltage	Selectable from three different levels in the OFS1.VDSEL0[1:0] bits	Selectable from three different levels in the LVDLVLR.LVD1LVL[4:0] bits	Selectable from three different levels in the LVDLVLR.LVD2LVL[2:0] bits
	Monitor flag	None	LVD1SR.MON flag: Monitors whether voltage is higher or lower than V_{det1}	LVD2SR.MON flag: Monitors whether voltage is higher or lower than V_{det2}
			LVD1SR.DET flag: V_{det1} passage detection	LVD2SR.DET flag: V_{det2} passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor 1 reset	Voltage monitor 2 reset
		Reset when V_{det0} > VCC CPU restart after specified time with VCC > V_{det0}	Reset when V_{det1} > VCC CPU restart timing selectable: after specified time with VCC > V_{det1} or V_{det1} > VCC	Reset when V_{det2} > VCC CPU restart timing selectable: after specified time with VCC > V_{det2} or V_{det2} > VCC
	Interrupt	No interrupt	Voltage monitor 1 interrupt	Voltage monitor 2 interrupt
			Non-maskable or maskable interrupt selectable	Non-maskable or maskable interrupt selectable
			Interrupt request issued when V_{det1} > VCC or VCC > V_{det1}	Interrupt request issued when V_{det2} > VCC or VCC > V_{det2}
Digital filter	Enable/ Disable switching	Digital filter function not available	Available	Available
	Sampling time	—	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)
Event linking		None	Available Output of event signals on detection of V_{det1} crossings	Available Output of event signals on detection of V_{det2} crossings



Note 1. See section 7, Option-Setting Memory.

Figure 8.1 Voltage detection 0, 1, and 2 block diagram

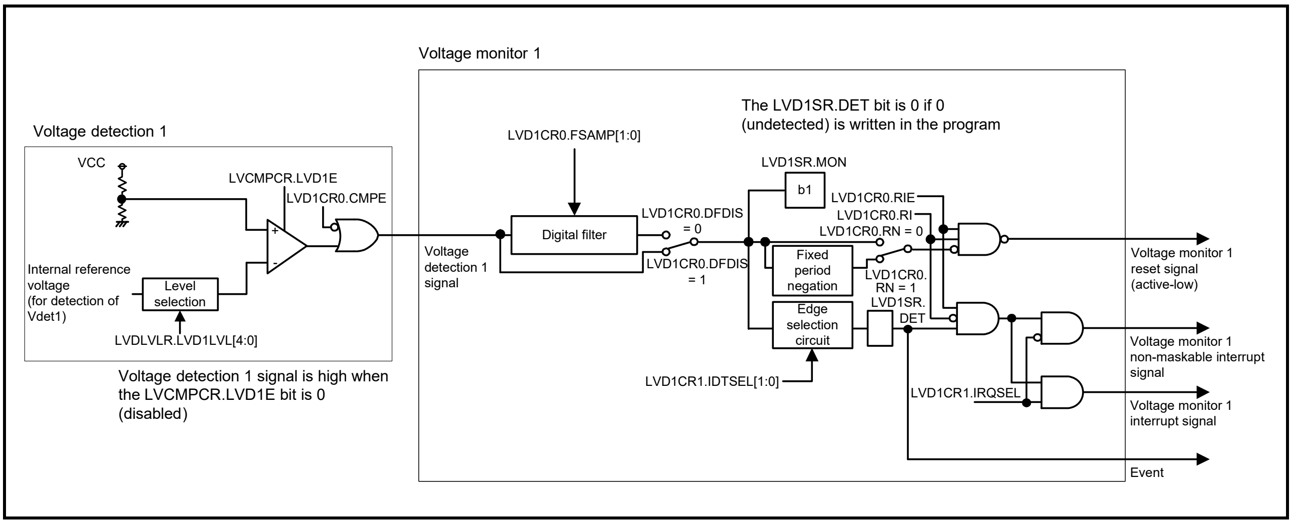


Figure 8.2 Voltage monitor 1 interrupt/reset circuit block diagram

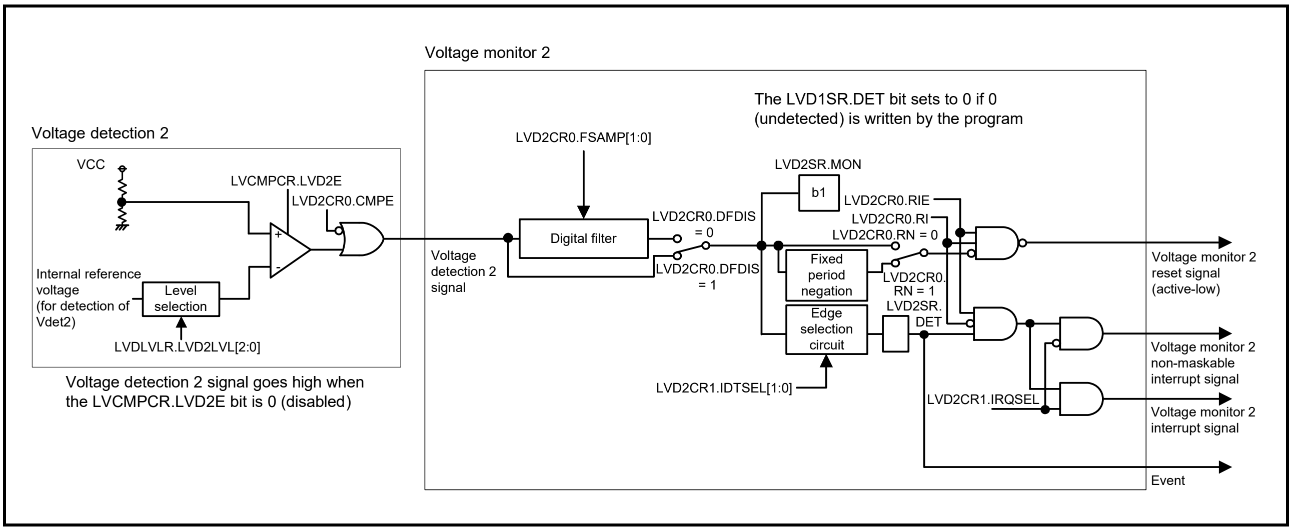


Figure 8.3 Voltage monitor 2 interrupt/reset circuit block diagram

8.2 Register Descriptions

8.2.1 Voltage Monitor 1 Circuit Control Register 1 (LVD1CR1)

Address(es): SYSTEM.LVD1CR1 4001 E0E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	IDTSEL[1:0]	Voltage Monitor 1 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ V_{det1} (rise) is detected 0 1: When VCC < V_{det1} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited.	R/W
b2	IRQSEL	Voltage Monitor 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt.*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.LVD1EN bit in the ICU from the reset state.

8.2.2 Voltage Monitor 1 Circuit Status Register (LVD1SR)

Address(es): SYSTEM.LVD1SR 4001 E0E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MON	DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	DET	Voltage Monitor 1 Voltage Change Detection Flag	0: Not detected 1: V_{det1} passage detected.	R(/W) *1
b1	MON	Voltage Monitor 1 Signal Monitor Flag	0: VCC < V_{det1} 1: VCC ≥ V_{det1} or MON is disabled.	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 1 Voltage Change Detection Flag)

The DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.CMPE bit is 1 (voltage monitor 1 circuit comparison result output enabled).

Set the DET flag to 0 after LVD1CR0.RIE is set to 0 (disabled). LVD1CR0.RIE can be set to 1 (enabled) after 2 or more PCLKB cycles have elapsed.

MON flag (Voltage Monitor 1 Signal Monitor Flag)

The MON flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.CMPE bit is 1 (voltage monitor 1 circuit comparison result output enabled).

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.3 Voltage Monitor 2 Circuit Control Register 1 (LVD2CR1)

Address(es): SYSTEM.LVD2CR1 4001 E0E2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	IDTSEL[1:0]	Voltage Monitor 2 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ V_{det2} (rise) is detected 0 1: When VCC < V_{det2} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited.	R/W
b2	IRQSEL	Voltage Monitor 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt.*1	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.LVD1EN bit in the ICU from the reset state.

8.2.4 Voltage Monitor 2 Circuit Status Register (LVD2SR)

Address(es): SYSTEM.LVD2SR 4001 E0E3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MON	DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	DET	Voltage Monitor 2 Voltage Change Detection Flag	0: Not detected 1: V_{det2} passage detection.	R(/W)*1
b1	MON	Voltage Monitor 2 Signal Monitor Flag	0: VCC < V_{det2} 1: VCC ≥ V_{det2} or MON is disabled.	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 2 Voltage Change Detection Flag)

The DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.CMPE bit is 1 (voltage monitor 2 circuit comparison result output enabled).

The DET flag must be set to 0 after LVD2CR0.RIE is set to 0 (disabled). LVD2CR0.RIE can be set to 1 (enabled) after 2 or more PCLKB cycles have elapsed.

MON flag (Voltage Monitor 2 Signal Monitor Flag)

The MON flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.CMPE bit is 1 (voltage monitor 2 circuit comparison result output enabled).

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.5 Voltage Monitor Circuit Control Register (LVCMPCR)

Address(es): SYSTEM.LVCMPCR 4001 E417h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1 Enable	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled.	R/W
b6	LVD2E	Voltage Detection 2 Enable	0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

LVD1E bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.MON bit, set the LVD1E bit to 1. The voltage detection 1 circuit starts when td(E-A) elapses after the LVD1E bit value is changed from 0 to 1. When using the voltage detection 1 circuit in Deep Software Standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

LVD2E bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.MON bit, set the LVD2E bit to 1. The voltage detection 2 circuit starts when td(E-A) elapses after the LVD2E bit value is changed from 0 to 1. When using the voltage detection 2 circuit in Deep Software Standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): SYSTEM.LVDLVLR 4001 E418h

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2LVL[2:0]			LVD1LVL[4:0]				
Value after reset	1	1	1	1	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b4 to b0	LVD1LVL[4:0]	Voltage Detection 1 Level Select (Standard voltage during fall in voltage)	b4 b0 1 0 0 0 1: 2.99 V (V_{det1_1}) 1 0 0 1 0: 2.92 V (V_{det1_2}) 1 0 0 1 1: 2.85 V (V_{det1_3}). Other settings are prohibited.	R/W
b7 to b5	LVD2LVL[2:0]	Voltage Detection 2 Level Select (Standard voltage during fall in voltage)	b7 b5 1 0 1: 2.99 V (V_{det2_1}) 1 1 0: 2.92 V (V_{det2_2}) 1 1 1: 2.85 V V_{det2_3}). Other settings are prohibited.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

The contents of the LVDLVLR register can only be changed if the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits (voltage detection n circuit disable, n = 1, 2) are both 0. Do not set LVD detectors 1 and 2 to the same voltage detection level.

8.2.7 Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0)

Address(es): SYSTEM.LVD1CR0 4001 E41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	RN	RI	FSAMP[1:0]		—	CMPE	DFDIS	RIE
Value after reset:	1	0	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RIE	Voltage Monitor 1 Interrupt/Reset Enable	0: Disable 1: Enable.	R/W
b1	DFDIS	Voltage Monitor 1 Digital Filter Disable Mode Select	0: Enable digital filter 1: Disable digital filter.	R/W
b2	CMPE	Voltage Monitor 1 Circuit Comparison Result Output Enable	0: Disable voltage monitor 1 circuit comparison result output 1: Enable voltage monitor 1 circuit comparison result output.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Bit	Symbol	Bit name	Description	R/W
b5, b4	FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency.	R/W
b6	RI	Voltage Monitor 1 Circuit Mode Select	0: Generate voltage monitor 1 interrupt on V_{det1} passage 1: Enable voltage monitor 1 reset when the voltage falls to and below V_{det1}.	R/W
b7	RN	Voltage Monitor 1 Reset Negate Select	0: Negate after a stabilization time (tLVD1) when VCC > V_{det1} is detected 1: Negate after a stabilization time (tLVD1) on assertion of the LVD1 reset.	R/W

RIE bit (Voltage Monitor 1 Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor 1 interrupt/reset. Set this bit to 1 to ensure that neither a voltage monitor 1 interrupt nor a voltage monitor 1 reset is generated during programming or erasure of the flash memory.

DFDIS bit (Voltage Monitor 1 Digital Filter Disable Mode Select)

The DFDIS bit enables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (LOCO operating) if this bit is 0 (enabled). Set the bit to 1 (disabled) when using the voltage monitor 1 circuit in Software Standby or Deep Software Standby mode.

FSAMP[1:0] bits (Sampling Clock Select)

Only change the FSAMP[1:0] bits when the LVD1CR0.DFDIS bit is 1 (digital filter circuit disabled), but not when LVD1CR0.DFDIS is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor 1 Circuit Mode Select)

When the RI bit is 1 (voltage monitor 1 reset selected) or when the LVD2CR0.RI bit is 1 (voltage monitor 2 reset selected), transition to Deep Software Standby mode cannot be made, and operation transitions to Software Standby mode instead. To enter Deep Software Standby mode, set the RI bit to 0 (voltage monitor 1 interrupt selected) and the LVD2CR0.RI bit to 0 (voltage monitor 2 interrupt selected).

RN bit (Voltage Monitor 1 Reset Negate Select)

If the RN bit is to be set to 1 (negation follows a stabilization time on assertion of the LVD1 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Additionally, for a transition to Software Standby or Deep Software Standby, the only possible value for the RN bit is 0 (negation follows a stabilization time when VCC > V_{det1} is detected). Do not set the RN bit to 1 when this is the case.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.8 Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0)

Address(es): SYSTEM.LVD2CR0 4001 E41Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	RN	RI	FSAMP[1:0]		—	CMPE	DFDIS	RIE
Value after reset:	1	0	0	0	x	0	1	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RIE	Voltage Monitor 2 Interrupt/Reset Enable	0: Disable 1: Enable.	R/W
b1	DFDIS	Voltage Monitor 2 Digital Filter Disable Mode Select	0: Enable digital filter 1: Disable digital filter.	R/W

Bit	Symbol	Bit name	Description	R/W
b2	CMPE	Voltage Monitor 2 Circuit Comparison Result Output Enable	0: Disable voltage monitor 2 circuit comparison result output 1: Enable voltage monitor 2 circuit comparison result output.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b5, b4	FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency.	R/W
b6	RI	Voltage Monitor 2 Circuit Mode Select	0: Generate voltage monitor 2 interrupt on V_{det2} passage 1: Enable voltage monitor 2 reset when the voltage falls to and below V_{det2}.	R/W
b7	RN	Voltage Monitor 2 Reset Negate Select	0: Negate after a stabilization time (tLVD2) when VCC > V_{det2} is detected 1: Negate after a stabilization time (tLVD2) on assertion of the LVD2 reset.	R/W

RIE bit (Voltage Monitor 2 Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor 2 interrupt/reset. Set this bit to 1 to ensure that neither a voltage monitor 2 interrupt nor a voltage monitor 2 reset is generated during programming or erasure of the flash memory.

DFDIS bit (Voltage Monitor 2 Digital Filter Disable Mode Select)

The DFDIS bit enables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if this bit is 0 (enabled). Set the bit to 1 (disabled) when using the voltage monitor 2 circuit in Software Standby or Deep Software Standby mode.

FSAMP[1:0] bits (Sampling Clock Select)

Only change the FSAMP[1:0] bits when the LVD2CR0.DFDIS bit is 1 (digital filter circuit disabled), but not when LVD2CR0.DFDIS is 0 (digital filter circuit enabled).

RI bit (Voltage Monitor 2 Circuit Mode Select)

When the RI bit is 1 (voltage monitor 2 reset selected) or when the LVD1CR0.RI bit is 1 (voltage monitor 1 reset selected), transition to Deep Software Standby mode cannot be made, and operation transitions to Software Standby mode instead. To enter Deep Software Standby mode, set the RI bit to 0 (voltage monitor 2 interrupt selected) and the LVD1CR0.RI bit to 0 (voltage monitor 1 interrupt selected).

RN bit (Voltage Monitor 2 Reset Negate Select)

If the RN bit is to be set to 1 (negation follows a stabilization time on assertion of the LVD2 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). In addition, for a transition to Software Standby or Deep Software Standby, the only possible value for the RN bit is 0 (negation follows a stabilization time when VCC > V_{det2} is detected). Do not set the RN bit to 1 when this is the case.

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring V_{det0}

The comparison results from voltage monitor 0 are not available for reading.

8.3.2 Monitoring V_{det1}

Table 8.2 shows the procedure to set up monitoring against V_{det1}. After the settings are complete, the comparison results from voltage monitor 1 can be monitored with the LVD1SR.MON flag.

Table 8.2 Procedure to set up monitoring against V_{det1}

Step		Monitoring the results of comparison by voltage monitor 1
Setting up the voltage detection 1 circuit	1	Set LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to LVDLVLR register.
	2	Select the detection voltage in the LVDLVLR.LVD1LVL[4:0] bits.
	3	Set LVCMPCR.LVD1E = 1 to enable voltage detection 1.
	4	Wait for at least td(E-A) for the LVD operation stabilization time after LVD is enabled.*1
Setting up the digital filter*2	5	Select the sampling clock for the digital filter in the LVD1CR0.FSAMP[1:0] bits.
	6	Set LVD1CR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, 16 and the sampling clock for the digital filter is the LOCO frequency-divided by n.
Enabling output	8	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.

Note 1. Steps 5 to 7 can be performed during the wait time in step 4. For details on td(E-A), see section 60, Electrical Characteristics.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

8.3.3 Monitoring V_{det2}

Table 8.3 shows the procedure to set up monitoring against V_{det2}. After the settings are complete, the comparison results from voltage monitor 2 can be monitored with the LVD2SR.MON flag.

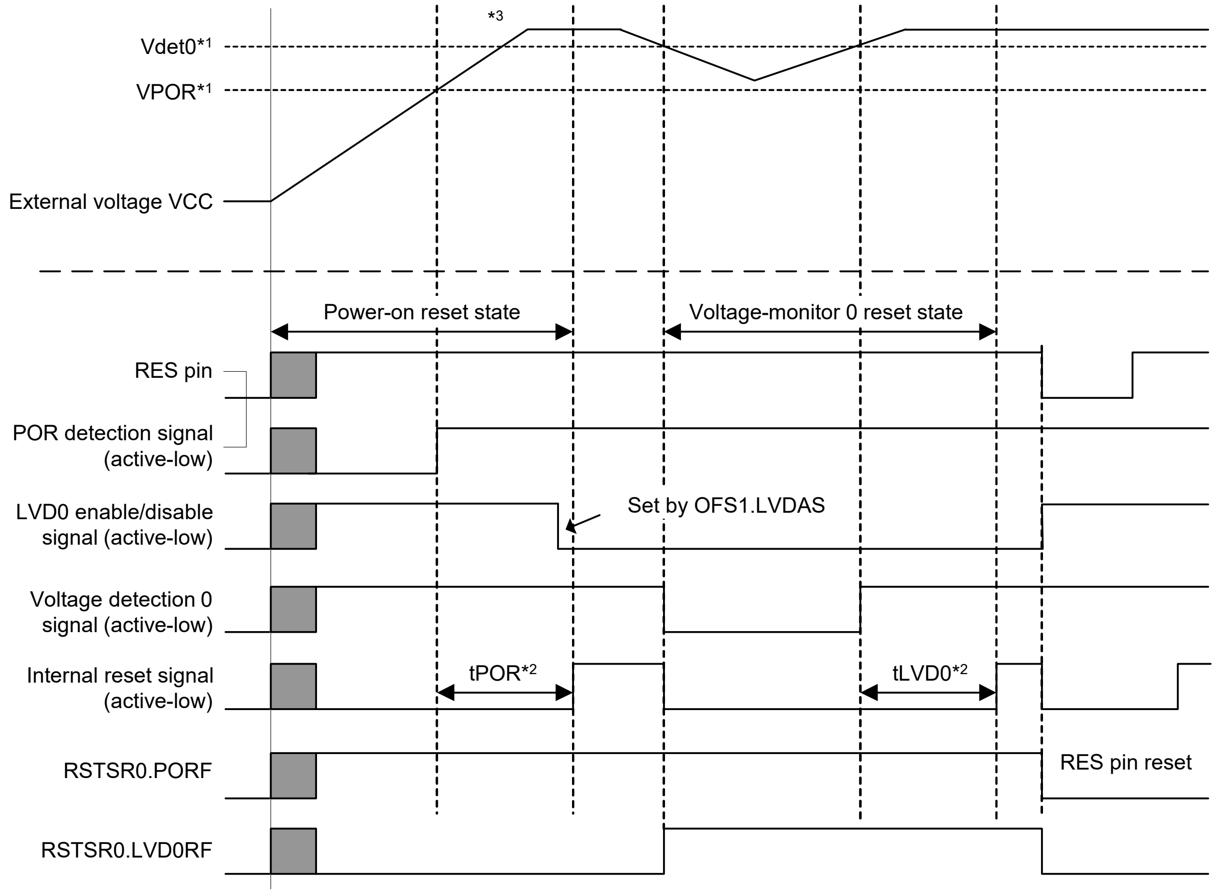
Table 8.3 Procedure to set up monitoring against V_{det2}

Step		Monitoring the results of comparison by voltage monitor 2
Setting up the voltage detection 2 circuit	1	Set LVCMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVDLVLR register.
	2	Select the detection voltage in the LVDLVLR.LVD2LVL[2:0] bits.
	3	Set LVCMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.
	4	Wait for at least td(E-A) for the LVD operation stabilization time after LVD is enabled.*1
Setting up the digital filter*2	5	Select the sampling clock for the digital filter in the LVD2CR0.FSAMP[1:0] bits.
	6	Set LVD2CR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, 16 and the sampling clock for the digital filter is the LOCO frequency-divided by n.
Enabling output	8	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.

Note 1. Steps 5 to 7 can be performed during the wait time in step 4. For details on td(E-A), see section 60, Electrical Characteristics.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

8.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1.LVDAS bit to 0 to enable the voltage monitor 0 reset after a reset. However, at boot mode, the reset from voltage monitor 0 is disabled regardless of the value of OFS1.LVDAS bit. Figure 8.4 shows an example of operations for a voltage monitor 0 reset.



Note: For details on the electrical characteristics, see section 60, Electrical Characteristics.

Note 1. VPOR indicates the detection level for a power-on reset and V_{det0} indicates the detection level for a voltage monitor 0 reset.

Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitor 0 reset.

Note 3. At power-on, raise VCC to the minimum guaranteed voltage before releasing the POR reset.

Figure 8.4 Example of voltage monitor 0 reset operation

8.5 Interrupt and Reset from Voltage Monitor 1

An interrupt or reset can be generated in response to the results of comparison from the voltage monitor 1 circuit.

Table 8.4 shows the procedure for setting bits related to the voltage monitor 1 interrupt and reset so that voltage monitoring operates. Table 8.5 shows the procedure for setting bits related to the voltage monitor 1 interrupt and reset so that voltage monitoring stops. Figure 8.5 shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see Figure 6.2 in section 6, Resets.

When using the voltage monitor 1 circuit in Software Standby or Deep Software Standby, set up the circuit with the following procedures.

(1) Settings in Software Standby mode

- Disable the digital filter (LVD1CR0.DFDIS = 1)
- When VCC > V_{det1} is detected, negate the voltage monitor 1 reset signal (LVD1CR0.RN = 0) following a stabilization time.

(2) Settings in Deep Software Standby mode

- Disable the digital filter (LVD1CR0.DFDIS = 1)

- Enable voltage monitor 1 interrupts (LVD1CR0.RI = 0). If the voltage monitor 1 reset is enabled (LVD1CR0.RI = 1), a transition to Deep Software Standby mode is not possible, and the operation transitions to Software Standby mode instead
- When the DPSBYCR.DEEPCUT[1:0] bits are 11b, the voltage monitor 1 circuit stops. To use the voltage monitor 1 circuit in Deep Software Standby mode, set the DPSBYCR.DEEPCUT[1:0] bits to a value other than 11b.

Table 8.4 Procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring operates

<table>
<tr><th colspan="2">Step</th><th>Voltage monitor 1 interrupt (voltage monitor 1 ELC event output)</th><th>Voltage monitor 1 reset</th></tr>
<tr><td rowspan="4">Setting up the voltage detection 1 circuit</td><td>1</td><td colspan="2">Set LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVDLVLR register.</td></tr>
<tr><td>2</td><td colspan="2">Select the detection voltage in the LVDLVLR.LVD1LVL[3:0] bits.</td></tr>
<tr><td>3</td><td colspan="2">Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.</td></tr>
<tr><td>4</td><td colspan="2">Wait for at least td(E-A) for the LVD operation stabilization time after LVD is enabled.*1</td></tr>
<tr><td rowspan="3">Setting up the digital filter *2</td><td>5</td><td colspan="2">Select the sampling clock for the digital filter in the LVD1CR0.FSAMP[1:0] bits.</td></tr>
<tr><td>6</td><td colspan="2">Set LVD1CR0.DFDIS = 0 to enable the digital filter.</td></tr>
<tr><td>7</td><td colspan="2">Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, 16 and the sampling clock for the digital filter is the LOCO frequency-divided by n.*4</td></tr>
<tr><td rowspan="2">Setting up the voltage monitor 1 interrupt or reset</td><td>8</td><td>Set LVD1CR0.RI = 0 to select the voltage monitor 1 interrupt.</td><td>• Set LVD1CR0.RI = 1 to select the voltage monitor 1 reset
• Select the type of reset negation in the LVD1CR0.RN bit.</td></tr>
<tr><td>9</td><td>• Select the interrupt request timing in the LVD1CR1.IDTSEL[1:0] bits
• Select the interrupt type in the LVD1CR1.IRQSEL bit.</td><td>—</td></tr>
<tr><td rowspan="3">Enabling output</td><td>10</td><td colspan="2">Set LVD1SR.DET = 0.</td></tr>
<tr><td>11</td><td colspan="2">Set LVD1CR0.RIE = 1 to enable the voltage monitor 1 interrupt or reset.*3</td></tr>
<tr><td>12</td><td colspan="2">Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.</td></tr>
</table>

Note 1. Steps 5 to 11 can be performed during the wait time in step 4. For details on td(E-A), see section 60, Electrical Characteristics.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.
Note 3. Step 11 is not required if only the ELC event signal is to be output.
Note 4. Steps 8 to 11 can be performed during the wait time in step 7.

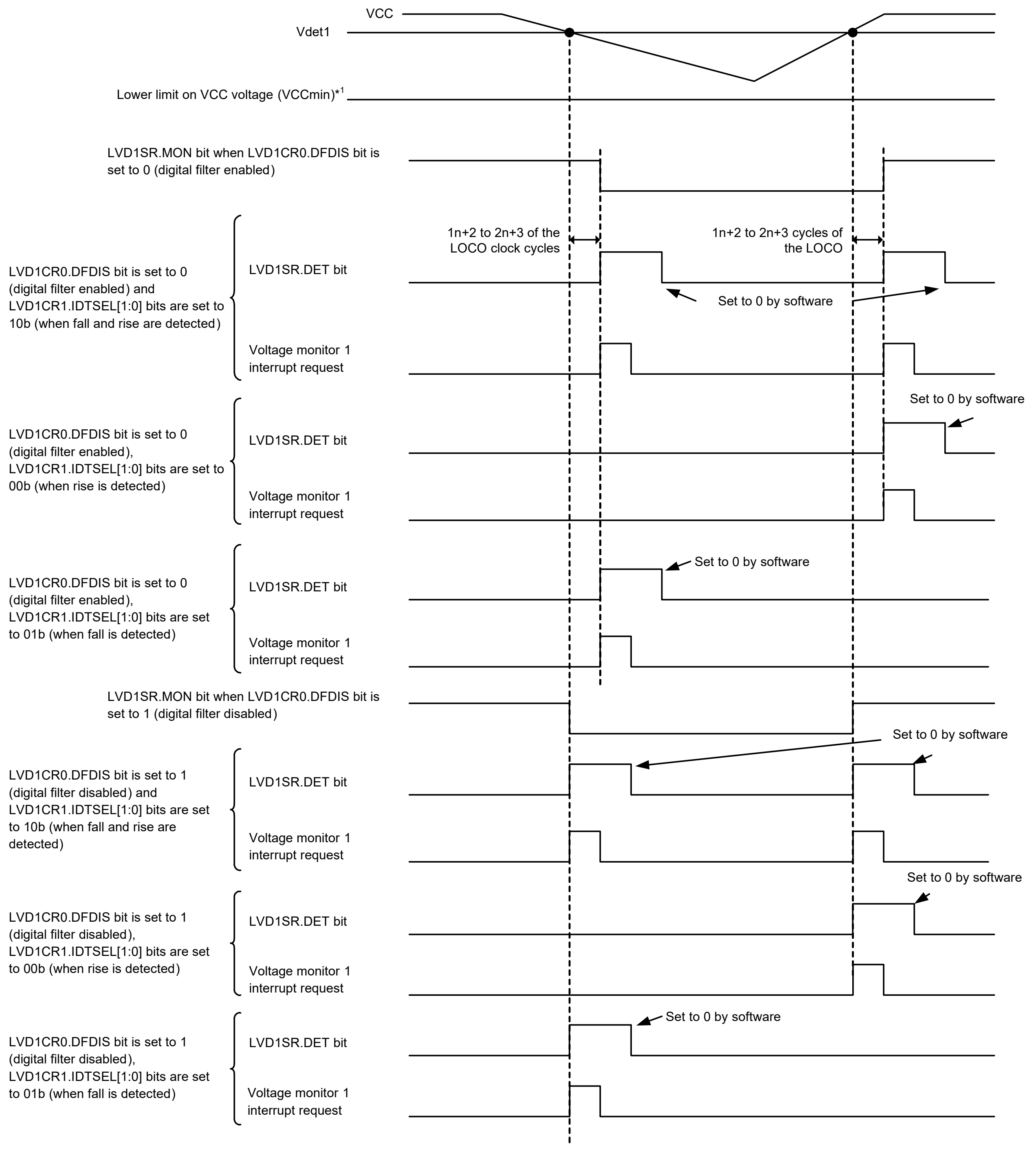
Table 8.5 Procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring stops

Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset
Settings to stop enabling output	1	Set LVD1CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 1.
	2	Wait for at least 2n + 3 cycles of the LOCO, where n = 2, 4, 8, 16 and the sampling clock for the digital filter is the LOCO frequency-divided by n.*1
	3	Set LVD1CR0.RIE = 0 to disable the voltage monitor 1 interrupt or reset.*2
Stopping the digital filter	4	Set LVD1CR0.DFDIS = 1 to disable the digital filter.*1, *3
Stopping the voltage detection 1 circuit	5	Set LVCMPCR.LVD1E = 0 to disable the voltage detection 1 circuit.

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. Step 3 is not required if only the ELC event signal is to be output.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least 2 LOCO clock cycles before re-enabling it.

If the voltage monitor 1 interrupt or reset setting is to be made again after it is used and stopped once, omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting or stopping the voltage detection 1 circuit is not required if the settings for the circuit do not change
- Setting or stopping the digital filter is not required if the settings for the digital filter do not change
- Setting the voltage monitor 1 interrupt or reset is not required if the settings for the voltage monitor 1 interrupt or reset do not change.



n: Frequency of the sampling clock for the digital filter is the LOCO frequency divided by n

Note 1. When the voltage monitor 0 reset is not in use, VCC ≥ VCCmin.

Figure 8.5 Voltage monitor 1 interrupt operation example

8.6 Interrupt and Reset from Voltage Monitor 2

An interrupt or reset can be generated in response to the comparison results from the voltage monitor 2 circuit.

Table 8.6 shows the procedure for setting bits related to the voltage monitor 2 interrupt and reset so that voltage monitoring operates. Table 8.7 shows the procedure for setting bits related to the voltage monitor 2 interrupt and reset so that voltage monitoring stops. Figure 8.6 shows an example of operations for a voltage monitor 2 interrupt. For the operation of the voltage monitor 2 reset, see Figure 6.2 in section 6, Resets.

When using the voltage monitor 2 circuit in Software Standby or Deep Software Standby, set up the circuit with the following procedures.

(1) Settings in Software Standby mode

- Disable the digital filter (LVD2CR0.DFDIS = 1)
- When VCC > V_{det2} is detected, negate the voltage monitor 2 reset signal (LVD2CR0.RN = 0) following a stabilization time.

(2) Settings in Deep Software Standby mode

- Disable the digital filter (LVD2CR0.DFDIS = 1)
- Enable voltage monitor 2 interrupts (LVD2CR0.RI = 0). If the voltage monitor 2 reset is enabled (LVD2CR0.RI = 1), a transition to Deep Software Standby mode is not possible, and the operation transitions to Software Standby mode instead.
- When the DPSBYCR.DEEPCUT[1:0] bits are 11b, the voltage monitor 2 circuit stops. To use the voltage monitor 2 circuit in Deep Software Standby mode, set the DPSBYCR.DEEPCUT[1:0] bits to a value other than 11b.

Table 8.6 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring occurs

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting up the voltage detection 2 circuit	1	Set LVCMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVDLVLR register.	
	2	Select the detection voltage in the LVDLVLR.LVD2LVL[2:0] bits.	
	3	Set LVCMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.	
	4	Wait for at least td(E-A) for the LVD operation stabilization time after LVD is enabled.*1	
Setting up the digital filter *2	5	Select the sampling clock for the digital filter in the LVD2CR0.FSAMP[1:0] bits.	
	6	Set LVD2CR0.DFDIS = 0 to enable the digital filter.	
	7	Wait for at least 2n + 3 LOCO clock cycles, where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.*4	
Setting up the voltage monitor 2 interrupt or reset	8	Set LVD2CR0.RI = 0 to select the voltage monitor 2 interrupt.	• Set LVD2CR0.RI = 1 to select the voltage monitor 2 reset • Select the type of reset negation in the LVD2CR0.RN bit.
	9	• Select the interrupt request timing in the LVD2CR1.IDTSEL[1:0] bits • Select the interrupt type in the LVD2CR1.IRQSEL bit.	—
Enabling output	10	Set LVD2SR.DET = 0.	
	11	Set LVD2CR0.RIE = 1 to enable the voltage monitor 2 interrupt or reset.*3	
	12	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.	

Note 1. Steps 5 to 11 can be performed during the wait time in step 4. For details on td(E-A), see section 60, Electrical Characteristics.
Note 2. Steps 5 to 7 are not required if the digital filter is not in use.
Note 3. Step 11 is not required if only the ELC event signal is to be output.
Note 4. Steps 8 to 10 can be performed during the wait time in step 7.

Table 8.7 Procedure for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring stops

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output), voltage monitor 2 reset
Settings to stop enabling output	1	Set LVD2CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 2.
	2	Wait for at least 2n + 3 LOCO clock cycles, where n = 2, 4, 8, 16, and the sampling clock for the digital filter is the LOCO frequency-divided by n.*1
	3	Set LVD2CR0.RIE = 0 to disable the voltage monitor 2 interrupt or reset.*2
Stopping the digital filter	4	Set LVD2CR0.DFDIS = 1 to disable the digital filter.*1, *3
Stopping the voltage detection 1 circuit	5	Set LVCMPCR.LVD2E = 0 to disable the voltage detection 2 circuit.

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.
Note 2. Step 3 is not required if only the ELC event signal is to be output.
Note 3. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least 2 LOCO clock cycles before re-enabling it.

If the voltage monitor 2 interrupt or reset setting is to be made again after it is used and stopped once, omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting or stopping the voltage detection 2 circuit is not required if the settings for the circuit do not change
- Setting or stopping the digital filter is not required if the settings for the digital filter do not change
- Setting the voltage monitor 2 interrupt or reset is not required if the settings for the voltage monitor 2 interrupt or voltage monitor 2 reset do not change.

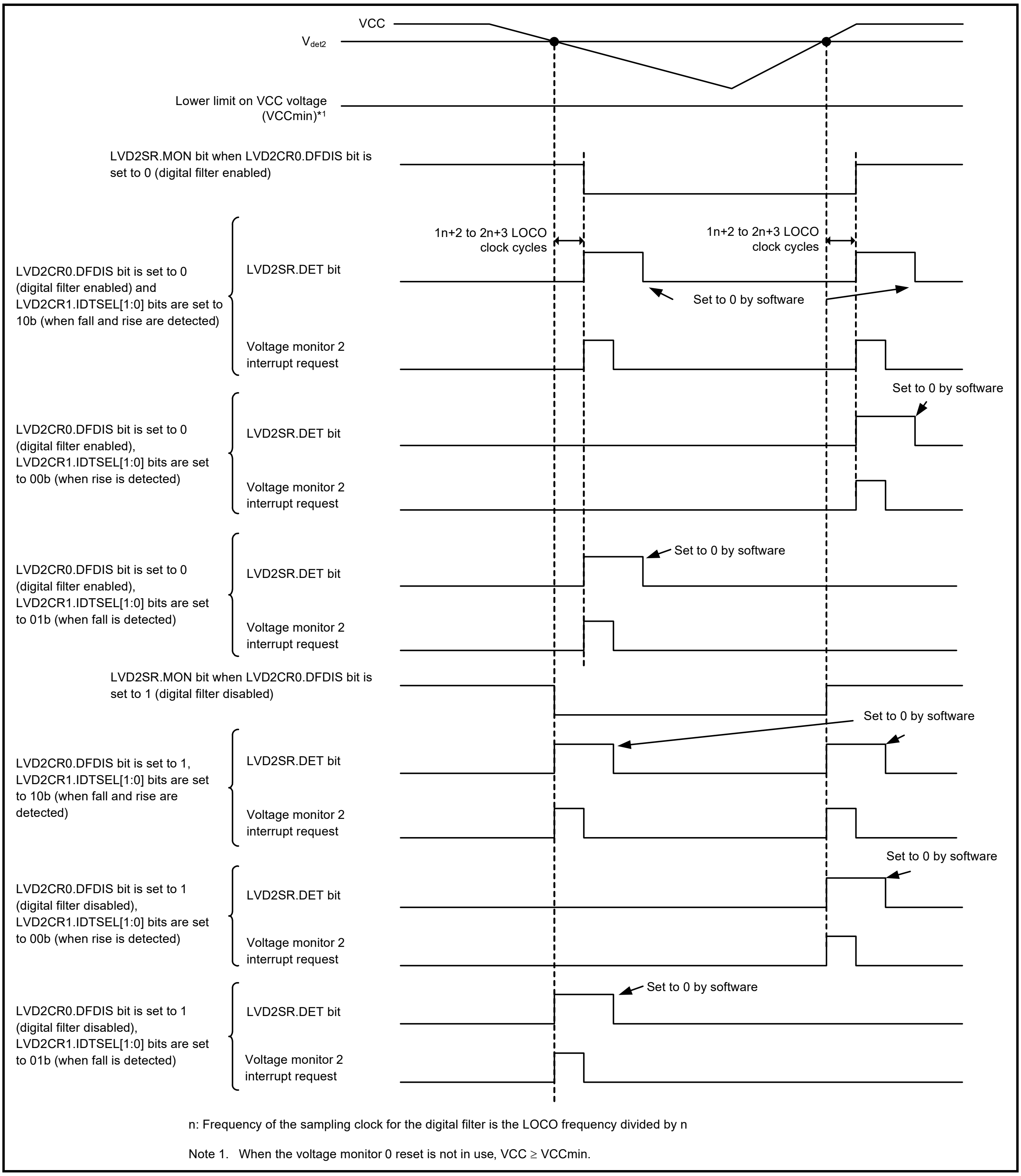


Figure 8.6 **Example of voltage monitor 2 interrupt operation**

8.7 Event Link Output

The LVD can output the event signals to the Event Link Controller (ELC).

(1) V_{det1} Crossing Detection Event

The LVD outputs the event signal when it detects that the voltage has passed the V_{det1} voltage while both the voltage detection 1 circuit and the voltage monitor 1 circuit comparison result output are enabled.

(2) V_{det2} Crossing Detection Event

The LVD outputs the event signal when it detects that the voltage has passed the V_{det2} voltage while both the voltage detection 2 circuit and the voltage monitor 2 circuit comparison result output are enabled.

When enabling the event link output function of the LVD, you must enable the LVD before enabling the LVD event link function of the ELC. To stop the event link output function of the LVD, you must stop the LVD before disabling the LVD event link function of the ELC.

8.7.1 Interrupt Handling and Event Linking

The LVD provides bits to individually enable or disable the voltage monitor 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal (LVD1CR0.RIE or LVD2CR0.RIE) is output to the CPU.

On the other hand, as soon as an interrupt source is generated, an event link signal is output as the event signal to the other module through the ELC, regardless of the state of the interrupt enable bit.

It is possible to output voltage monitor 1 and 2 interrupts in Software Standby and Deep Software Standby modes. The event signals for the ELC in Software Standby and Deep Software Standby modes are output as follows:

- When a V_{det1} or V_{det2} passage event is detected in Software Standby mode, event signals are not generated for the ELC because the clock is not supplied in Software Standby mode. Because the V_{det1} and V_{det2} passage detection flags are saved, when the clock supply resumes after returning from Software Standby mode, the event signals for the ELC are output based on the state of the V_{det1} and V_{det2} detection flags.
- When a V_{det1} or V_{det2} passage events is detected in Deep Software Standby mode, event signals are not generated for the ELC.

9. Clock Generation Circuit

9.1 Overview

The MCU provides a clock generation circuit.

Table 9.1 and Table 9.2 list the clock generation circuit specifications, Figure 9.1 shows a block diagram, and Table 9.3 lists the I/O pins.

Table 9.1 Specifications of the clock generation circuit for the clock sources

Clock source	Parameter	Specifications
Main clock oscillator (MOSC)	Resonator frequency	8 to 24 MHz USB boot mode: 8, 10, 12, 15, 16, 20, 24 MHz
	External clock input frequency	Up to 24 MHz
	External resonator or additional circuit: ceramic resonator, crystal	Available
	Connection pins EXTAL, XTAL	
	Drive capability switching	
	Oscillation stop detection function	
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	External resonator or additional circuit: crystal resonator	Available
	Connection pins: XCIN, XCOUT	
	Drive capability switching	
PLL circuit	Input clock source	MOSC, HOCO
	Input pulse frequency division ratio	Selectable from 1, 2, and 3
	Input frequency	8 to 24 MHz
	Frequency multiplication ratio	Selectable from 10 to 30 (0.5 steps) *1,*2
	PLL output frequency	120 to 240 MHz
High-speed on-chip oscillator (HOCO)	Oscillation frequency	16, 18, 20 MHz
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
IWDT-dedicated on-chip oscillator (IWDTLOCO)	Oscillation frequency	15 kHz
External clock input for JTAG (TCK)	Input clock frequency	Up to 25 MHz
External clock input for SWD (SWCLK)	Input clock frequency	Up to 25 MHz

Note 1. Selectable from 10 to 20 when oscillation stop detection function is enabled and input frequency less than 12 MHz is used.

Note 2. Except for the condition in note 1, oscillation stop detection function is available by CAC.

Table 9.2 Specifications of the clock generation circuit for the internal clocks (1 of 3)

Parameter	Clock sources	Clock supply	Specifications
System clock (ICLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	CPU, DTC, DMAC, Flash, SRAM	Up to 120 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64

Table 9.2 Specifications of the clock generation circuit for the internal clocks (2 of 3)

Parameter	Clock sources	Clock supply	Specifications
Peripheral module clock A (PCLKA)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral modules (ETHERC, EDMAC, USBHS, QSPI, SPI, SCI, SCE7, GLCDC, SDHI, CRC, JPEG engine, DRW, IrDA, GPT bus-clock)	Up to 120 MHz*2 Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock B (PCLKB)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral modules (IIC, SSIE, SRC, DOC, CAC, CAN, DAC12, POEG, CTSU, AGT, Standby SRAM, ELC, I/O ports, RTC, WDT, IWDT, ADC12, KINT, USBFS, ACMPHS, TSN, PDC)	Up to 60 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock C (PCLKC)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral module (ADC12 conversion clock)	Up to 60 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock D (PCLKD)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral module (GPT count-clock)	Up to 120 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Flash interface clock (FCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Flash interface	4 to 60 MHz (P/E) Up to 60 MHz (read) *1 Division ratios: 1, 2, 4, 8, 16, 32, 64
External bus clock (BCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	External bus	Up to 120 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
EBCLK pin output (EBCLK)	BCLK or 1/2 BCLK	EBCLK pin	Up to 60 MHz Division ratios: 1, 2
SDCLK pin output (SDCLK)	BCLK	SDCLK pin	Up to 120 MHz
USB clock (UCLK)	PLL	USB	48 MHz Division ratios: 3, 4, 5
USB-PHY clock (USBMCLK)	MOSC	USB-PHY	12, 20, 24 MHz
CAN clock (CANMCLK)	MOSC	CAN	8 to 24 MHz
LCD_CLK pin output (LCD_CLK) and graphic LCD pixel clock (PXCLK)	LCD_EXTCLK, PLL output	LCD_CLK pin, peripheral module (Graphics LCD Controller)	Up to 54 MHz (parallel RGB) Up to 60 MHz (serial RGB) LCD_CLK division ratios: 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 16, 24, 32 LCD_CLK : PXCLK = 1:1 (parallel RGB) LCD_CLK : PXCLK = 4:1 (serial RGB)
AGT clock (AGTSCLK, AGTLCLK)	SOSC, LOCO	AGT	32.768 MHz
CAC main clock (CACMCLK)	MOSC	CAC	Up to 24 MHz
CAC sub-clock (CACSCLK)	SOSC	CAC	32.768 kHz
CAC LOCO clock (CACLCLK)	LOCO	CAC	32.768 kHz
CAC MOCO clock (CACMOCLK)	MOCO	CAC	8 MHz
CAC HOCO clock (CACHCLK)	HOCO	CAC	16, 18, 20 MHz
CAC IWDTLOCO clock (CACILCLK)	IWDTLOCO	CAC	15 kHz
RTC clock (RTCSCLK, RTCLCLK)	SOSC, LOCO	RTC	32.768 kHz
IWDT clock (IWDTCLK)	IWDTLOCO	IWDT	15 kHz
SysTick timer clock (SYSTICCLK)	LOCO	SysTick timer	32.768 kHz
JTAG clock (JTAGTCK)	TCK pin	JTAG	Up to 25 MHz

Table 9.2 Specifications of the clock generation circuit for the internal clocks (3 of 3)

Parameter	Clock sources	Clock supply	Specifications
Clock and buzzer output (CLKOUT)	MOSC, SOSC, LOCO, MOCO, HOCO	CLKOUT pin	Up to 24 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64, 128
Serial wire clock (SWCLK)	SWCLK pin	OCD	Up to 25 MHz
Trace clock (TRCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	CPU-OCD	Up to 60 MHz Division ratios: 1, 2, 4
TCLK pin output (TCLK)	1/2 TRCLK	TCLK pin	Up to 30 MHz

Note: Constraints on clock frequency settings: ICLK ≥ PCLKA ≥ PCLKB, PCLKD ≥ PCLKA ≥ PCLKB
Constraints on clock frequency ratio (N: integer, and up to 64):
ICLK:FCLK = N:1, ICLK:BCLK = N:1, ICLK:PCLKA = N:1, ICLK:PCLKB = N:1
ICLK:PCLKC = N:1 or 1:N, ICLK:PCLKD = N:1 or 1:N
If the A/D converter is enabled, clock frequency ratio is constrained as below:
PCLKB:PCLKC = 1:1 or 1:2 or 1:4 or 2:1 or 4:1 or 8:1.

Note: Clocks have a permissible frequency range. See Table 9.2.
Flash memory and SRAM also have a permissible operating frequency range in each wait cycle setting. See section 53, SRAM, section 55, Flash Memory.
Those clock frequency ranges must be satisfied even if the HOCO has its maximum or minimum frequency. See section 60, Electrical Characteristics.

Note: If PLL reference clock source is HOCO, PLL multiplication setting must be set to 120 - 240 MHz in consideration of HOCO frequency (minimum/maximum).

Note 1. The minimum FCLK frequency is 4 MHz in Programming/Erasure (P/E) mode.

Note 2. When using ETHERC, the PCLKA frequency is in the range 12.5 MHz ≤ PCLKA ≤ 120 MHz.
When using ETHERC, GLCDC, DRW, and JPEG, PCLKA = ICLK.

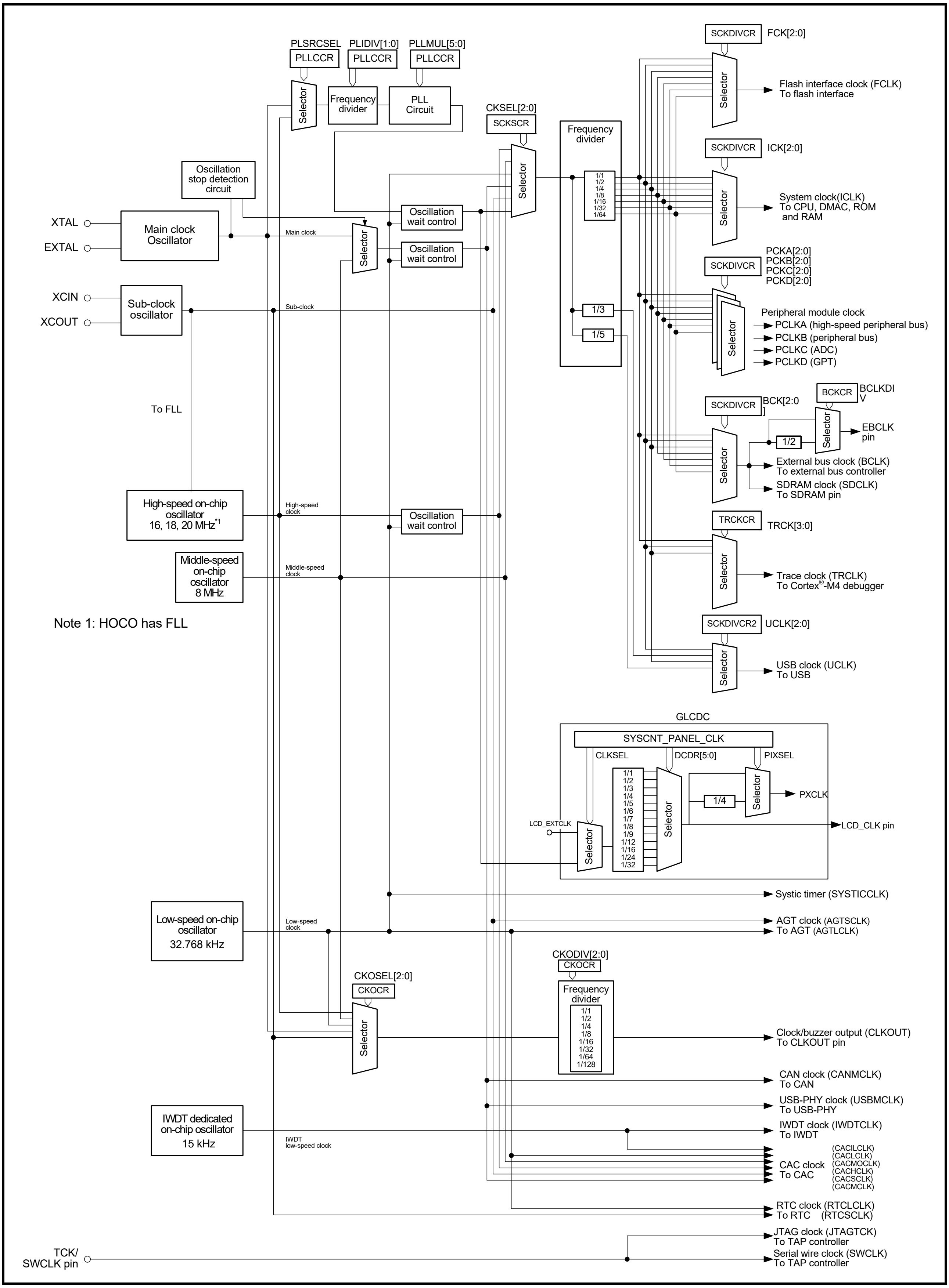


Figure 9.1 **Clock generation circuit block diagram**

Table 9.3 Clock Generation Circuit I/O pins

Pin name	I/O	Description
XTAL	Output	Crystal resonator connections The EXTAL pin can also be used to input an external clock. For details, section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	32.768-kHz crystal resonator connection
XCOUT	Output	
TCK/SWCLK	Input	JTAG clock input
EBCLK	Output	External bus clock (EBCLK) supply for external devices
SDCLK	Output	SDRAM clock (SDCLK) supply for external devices
CLKOUT	Output	CLKOUT and BUZZER clock output

9.2 Register Descriptions

9.2.1 System Clock Division Control Register (SCKDIVCR)

Address(es): SYSTEM.SCKDIVCR 4001 E020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	FCK[2:0]			—	ICK[2:0]			—	—	—	—	—	BCK[2:0]		
Value after reset:	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	PCKA[2:0]			—	PCKB[2:0]			—	PCKC[2:0]			—	PCKD[2:0]		
Value after reset:	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	PCKD[2:0]	Peripheral Module Clock D (PCLKD) Select*4	b2 b0 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64. Other settings are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	PCKC[2:0]	Peripheral Module Clock C (PCLKC) Select*4	b6 b4 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64. Other settings are prohibited.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b10 to b8	PCKB[2:0]	Peripheral Module Clock B (PCLKB) Select*3	b10 b8 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64. Other settings are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b12	PCKA[2:0]	Peripheral Module Clock A (PCLKA) Select*3	b14 b12 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64. Other settings are prohibited.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	BCK[2:0]	External Bus Clock (BCLK) Select*2	b18 b16 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64. Other settings are prohibited.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b24	ICK[2:0]	System Clock (ICLK) Select*1,*2,*3,*4,*5	b26 b24 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64. Other settings are prohibited.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	FCK[2:0]	Flash Interface Clock (FCLK) Select*1	b30 b28 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 1 0 1: ×1/32 1 1 0: ×1/64. Other settings are prohibited.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The following association is required between the frequencies of the system clock (ICLK) and the flash interface clock (FCLK): ICLK:FCLK = N:1 (N: integer)
If a setting where ICLK < FCLK is written, the write is ignored.

Note 2. The following association is required between the frequencies of the system clock (ICLK) and the external bus clock (BCLK): ICLK:BCLK = N:1 (N: integer)
If a setting where ICLK < BCLK is written, the write is ignored.

Note 3. The following association is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKA, PCLKB): ICLK:PCLKA = N:1, ICLK:PCLKB = N:1 (N: integer)
If a setting where ICLK < PCLKA or ICLK < PCLKB is written, the write is ignored.

Note 4. The following association is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKC, PCLKD): ICLK:PCLKC, PCLKD = N:1 or 1:N (N: integer).

Note 5. The frequency of the system clock (ICLK) is limited to the flash wait cycle register (FLWT). Refer to section 55, Flash Memory.

The SCKDIVCR register selects the frequencies of the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD), flash interface clock (FCLK), external bus clock (BCLK), and SDRAM clock (SDCLK).

When the PLL is selected as the clock source, set the following modules into the module-stop state before changing the value of SCKDIVCR: ETHERC, EPTPC, EDMAC, SCE7, DRW, JPEG, GLCDC, GPT32EH, and GPT32E.

In addition, when changing any value in SCKDIVCR from a lower division ratio to a higher division ratio, wait at least 750 ns before changing the value. When changing any value from a higher division ratio to a lower division ratio, wait at least 250 ns after changing the value before starting subsequent processing.

The recommended method to measure the wait time is to do so in software. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses.

Figure 9.2 shows an example flow for changing the value of SCKDIVCR.

PCKD[2:0] bits (Peripheral Module Clock D (PCLKD) Select)

The PCKD[2:0] bits select the frequency for peripheral module clock D (PCLKD).

PCKC[2:0] bits (Peripheral Module Clock C (PCLKC) Select)

The PCKC[2:0] bits select the frequency for peripheral module clock C (PCLKC).

PCKB[2:0] bits (Peripheral Module Clock B (PCLKB) Select)

The PCKB[2:0] bits select the frequency for peripheral module clock B (PCLKB).

PCKA[2:0] bits (Peripheral Module Clock A (PCLKA) Select)

The PCKA[2:0] bits select the frequency for peripheral module clock A (PCLKA).

BCK[2:0] bits (External Bus Clock (BCLK) Select)

The BCK[2:0] bits select the frequency for the external bus clock (BCLK) and the SDRAM clock (SDCLK).

ICK[2:0] bits (System Clock (ICLK) Select)

The ICK[2:0] bits select the frequency for the system clock for the CPU, DMAC, and DTC.

FCK[2:0] bits (Flash Interface Clock (FCLK) Select)

The FCK[2:0] bits select the frequency for the flash interface clock (FCLK).

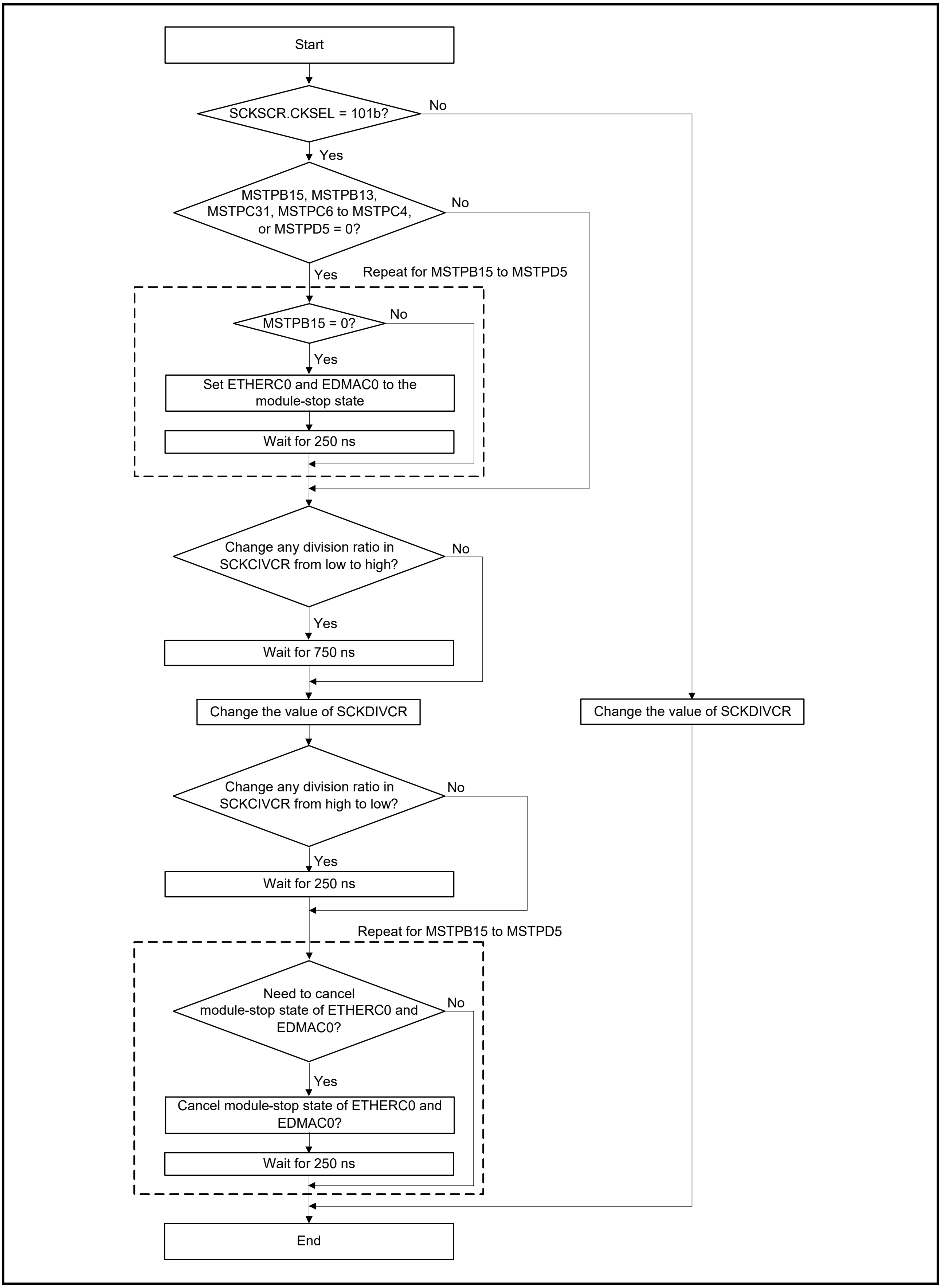


Figure 9.2 **Example flow for changing the value of SCKDIVCR**

9.2.2 System Clock Division Control Register 2 (SCKDIVCR2)

Address(es): SYSTEM.SCKDIVCR2 4001 E024h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	UCK[2:0]			—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6 to b4	UCK[2:0]	USB Clock (UCLK) Select	b6 b4 0 1 0: ×1/3 0 1 1: ×1/4 1 0 0: ×1/5. Other settings are prohibited.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Do not write to SCKDIVCR2 and SCKSCR at the same time by 32-bit access.

The SCKDIVCR2 register selects the frequency of the USB clock (UCLK).

UCK[2:0] bits (USB Clock (UCLK) Select)

The UCK[2:0] bits select the frequency of the USB clock (UCLK). The duty ratio is 2:1 when ×1/3 is selected or 3:2 when ×1/5 is selected.

9.2.3 System Clock Source Control Register (SCKSCR)

Address(es): SYSTEM.SCKSCR 4001 E026h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CKSEL[2:0]	Clock Source Select	b2 b0 0 0 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: Main clock oscillator 1 0 0: Sub-clock oscillator 1 0 1: PLL. Other settings are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Do not write to SCKDIVCR2 and SCKSCR at the same time by 32-bit access.

The SCKSCR register selects the clock source for the system clock.

When changing the value of SCKSCR to either select or deselect the PLL, set the following modules into the module-stop state before changing the SCKSCR value: ETHERC, EPTPC, EDMAC, SCE7, DRW, JPEG, GLCDC, GPT32EH, GPT32E.

In addition, when changing the value of SCKSCR from the PLL to a different clock source, wait at least 750 ns before changing the value. When changing the value from a non-PLL clock source to the PLL, wait at least 250 ns after changing the value before starting subsequent processing.

The recommended method to measure the wait time is to do so in software. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses.

Figure 9.3 shows an example flow for changing the value of SCKSCR.

CKSEL[2:0] bits (Clock Source Select)

The CKSEL[2:0] bits select the clock source for the following modules:

- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD)
- Flash interface clock (FCLK)
- External bus clock (BCLK)
- SDRAM clock (SDCLK)
- USBFS clock (UCLK).

The bits select from one of the following sources:

- Low-speed on-chip oscillator (LOCO)
- Middle-speed on-chip oscillator (MOCO)
- High-speed on-chip oscillator (HOCO)
- Main clock oscillator
- Sub-clock oscillator
- PLL circuit.

The clock sources should be switched when there are no occurring internal asynchronous interrupt.

Transitions to clock sources that are not in operation are prohibited.

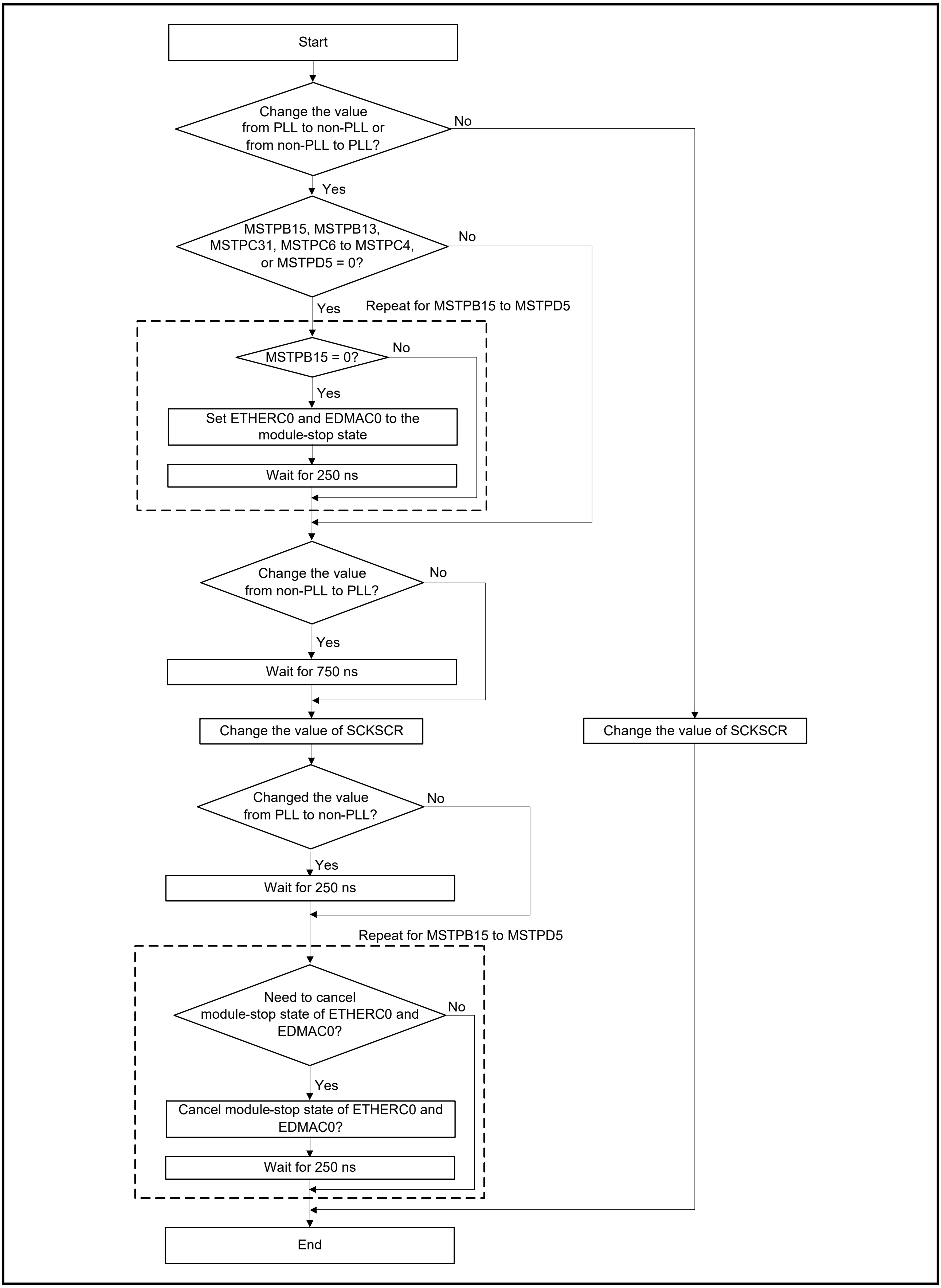


Figure 9.3 **Example flow for changing the value of SCKSCR**

9.2.4 PLL Clock Control Register (PLLCCR)

Address(es): SYSTEM.PLLCCR 4001 E028h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PLLMUL[5:0]						—	—	—	PLSRC SEL	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select*1	b1 b0 0 0: × 1 0 1: × 1/2 1 0: × 1/3 1 1: Setting prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PLSRCSEL	PLL Clock Source Select	0: Main clock oscillator*4 1: HOCO.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	PLLMUL[5:0]	PLL Frequency Multiplication Factor Select*2,*3	b13 b8 0 1 0 0 1 1: × 10.0 0 1 0 1 0 0: × 10.5 0 1 0 1 0 1: × 11.0 … 0 1 1 1 0 0: × 14.5 0 1 1 1 0 1: × 15.0 0 1 1 1 1 0: × 15.5 … 1 1 1 0 1 0: × 29.5 1 1 1 0 1 1: × 30.0. Other settings are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. PLIDIV[1:0] must be set so that the frequency of the PLL input signal is within the range listed in Table 9.1.
Note 2. PLLMUL[5:0] must be set so that the frequency of the PLL output signal is within the range listed in Table 9.1.
Note 3. PLLMUL[5:0] should be set up to 20 when oscillation stop detection function is enabled and input frequency less than 12 MHz is used.
Note 4. PLSRCSEL must be set to 0 when using UCLK.

The PLLCCR register sets up the operation of the PLL circuit. Writing to the PLLCCR is prohibited when the PLL is operating (when the PLLCR.PLLSTP bit is 0).

PLIDIV[1:0] bits (PLL Input Frequency Division Ratio Select*1)

The PLIDIV[1:0] bits select the frequency division ratio for the PLL clock source.

PLSRCSEL bit (PLL Clock Source Select)

The PLSRCSEL bit selects the clock source for the PLL.

PLLMUL[5:0] bits (PLL Frequency Multiplication Factor Select*2,*3)

The PLLMUL[5:0] bits select the frequency multiplication factor for the PLL circuit.

9.2.5 PLL Control Register (PLLCR)

Address(es): SYSTEM.PLLCR 4001 E02Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	PLLSTP	PLL Stop Control	0: Operate the PLL 1: Stop the PLL.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PLLCR register controls the operation of the PLL circuit.

PLLSTP bit (PLL Stop Control)

The PLLSTP bit starts or stops the PLL circuit. When selecting the main clock oscillator as the clock source for the PLL in the PLLCCR.PLSRCSEL bit, you must also set the Main Clock Oscillator Wait Control Register (MOSCWTCR).

After setting the PLLSTP bit to 0, confirm that the OSCSF.PLLSF bit is set to 1 before using the PLL clock. A fixed stabilization wait is required after setting the PLL to start operation. A fixed wait for oscillation to stop is also required.

The following constraints apply when starting and stopping operation:

- After stopping the PLL, confirm that the OSCSF.PLLSF bit is 0 before restarting the PLL
- Confirm that the PLL is operating and that the OSCSF.PLLSF bit is 1 before stopping the PLL
- Regardless of whether the PLL clock is selected as the system clock, after setting the PLL to start operation, confirm that the OSCSF.PLLSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow a setting to stop the PLL, confirm that the OSCSF.PLLSF bit is cleared to 0 before executing the WFI instruction.

Writing 1 to PLLSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL).

Make sure the following conditions apply before writing 0 to PLLSTP:

- When PLL source clock = MOSC, OSCSF.MOSCSF bit = 1
- When PLL source clock = HOCO, OSCSF.HOCOSF bit = 1.

9.2.6 External Bus Clock Control Register (BCKCR)

Address(es): SYSTEM.BCKCR 4001 E030h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BCLKDIV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BCLKDIV	EBCLK Pin Output Select	0: BCLK 1: BCLK/2.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BCKCR register controls the external bus clock pin.

BCLKDIV bit (EBCLK Pin Output Select)

The BCLKDIV bit selects the clock signal for output from the EBCLK pin. The signal can be selected from either the BCLK clock with the frequency selected in the BCK[2:0] bits in SCKDIVCR or the BCLK clock divided by 2.

9.2.7 Main Clock Oscillator Control Register (MOSCCR)

Address(es): SYSTEM.MOSCCR 4001 E032h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Operate the main clock oscillator*1 1: Stop the main clock oscillator.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The MOMCR register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator. To start the main clock oscillator, set this bit to 0. When changing the value of the bit, only execute subsequent instructions after reading the bit to check that the value was updated. When using the main clock oscillator, you must set the Main Clock Oscillator Mode Oscillation Control Register (MOMCR) and the Main Clock Oscillator Wait Control Register (MOSCWTCR) before setting MOSTP to 0.

After setting the MOSTP bit to 0, confirm that the OSCSF.MOSCSF bit is set to 1 before using the main clock oscillator. A fixed stabilization wait is required after setting the main clock oscillator to start operation. A fixed wait for oscillation to stop is also required.

The following constraints apply when starting and stopping operation:

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF bit is 0 before restarting the main clock oscillator
- Confirm that the main clock oscillator is operating and that the OSCSF.MOSCSF bit is 1 before stopping the main clock oscillator
- Regardless of whether the main clock oscillator is selected as the system clock, confirm that the OSCSF.MOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow a setting to stop the main clock oscillator, confirm that the OSCSF.MOSCSF bit is cleared to 0 before executing the WFI instruction.

Writing 1 to MOSTP is prohibited under the following conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL)
- PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and PLLCR.PLLSTP = 0 (PLL is operating).

9.2.8 Subclock Oscillator Control Register (SOSCCR)

Address(es): SYSTEM.SOSCCR 4001 E480h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SOSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SOSTP	Sub-Clock Oscillator Stop	0: Operate the sub-clock oscillator*1 1: Stop the sub-clock oscillator.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The SOMCR register must be set before setting SOSTP to 0.

The SOSCCR register controls the sub-clock oscillator.

SOSTP bit (Sub-Clock Oscillator Stop)

The SOSTP bit starts or stops the sub-clock oscillator. When changing the value of the bit, only execute subsequent instructions after reading the bit to check that the value was updated. Use the SOSTP bit when using the sub-clock oscillator as the source for a peripheral module, for example the RTC. When using the sub-clock oscillator, you must set the Sub-Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0.

After setting SOSTP to 0, only use the sub-clock oscillator after the sub-clock oscillation stabilization wait time (tSUBOSCOWT) elapses. A fixed stabilization wait is required after setting the sub-clock oscillator to start operation. A fixed wait for oscillation to stop is also required.

The following constraints apply when starting and stopping operation:

- After stopping the sub-clock oscillator, allow a stop interval of at least 5 SOSC cycles before restarting it
- Confirm that sub-clock oscillation is stable before stopping the sub-clock oscillator
- Regardless of whether the sub-clock oscillator is selected as the system clock, confirm that sub-clock oscillation is stable before executing a WFI instruction to place the MCU in Software Standby mode
- When a transition to Software Standby mode is to follow a setting to stop the sub-clock oscillator, wait for at least 3 SOSC cycles after the stop setting before executing the WFI instruction.

Writing 1 to SOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC).

9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): SYSTEM.LOCOCR 4001 E490h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	LCSTP	LOCO Stop	0: Operate the LOCO clock 1: Stop the LOCO clock.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The LOCOCR register controls the LOCO clock.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock. After setting LCSTP bit to 0 to start the LOCO clock, only use the clock after the LOCO clock oscillation stabilization wait time (tLOCOWT) elapses. A fixed stabilization wait is required after setting the LOCO clock to start operation. A fixed wait for oscillation to stop is also required.

The following constraints apply when starting and stopping operation:

- After stopping the LOCO clock, allow a stop interval of at least 5 LOCO cycles before restarting it
- Confirm that LOCO oscillation is stable before stopping the LOCO clock
- Regardless of whether the LOCO clock is selected as the system clock, confirm that LOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow a setting to stop the LOCO clock, wait for at least 3 LOCO cycles after the stop setting before executing the WFI instruction.

Writing 1 to LOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 010b (system clock source = LOCO).

Because the LOCO clock measures the wait time for other oscillators, it continues to oscillate while measuring this time, regardless of the setting in LOCOCR.LCSTP. As a result, the LOCO clock might be unintentionally supplied even when the LCSTP is set to stop.

9.2.10 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): SYSTEM.HOCOCR 4001 E036h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit name	Description	R/W
b0	HCSTP	HOCO Stop	0: Operate the HOCO clock*2 1: Stop the HOCO clock.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.
Note 2. If you are using the HOCO (HCSTP = 0), you must set the OFS1.HOCOFRQ1 bit to the optimum value.

The HOCOCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock. After setting the HCSTP bit to 0 to start the HOCO clock, confirm that the OSCSF.HOSCSF bit is set to 1 before using the clock. When OFS1.HOCOEN is set to 1, confirm that the OSCSF.HOCOSF is also set to 1 before using the HOCO clock. A fixed stabilization wait is required after setting the HOCO clock to start operation. A fixed wait for oscillation to stop is also required. For the HOCO to operate, the HOCO Wait Control Register (HOCOWTCR) must also be set.

The following constraints apply when starting and stopping operation:

- After stopping the HOCO, confirm that the OSCSF.HOCOSF bit is 0 before restarting the HOCO clock
- Confirm that the HOCO clock is operating and that the OSCSF.HOCOSF bit is 1 before stopping the HOCO clock
- Regardless of whether the HOCO clock is selected as the system clock, confirm that the OSCSF.HOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow a setting to stop the HOCO clock, confirm that the OSCSF.MOSCSF bit is cleared to 0 before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following conditions:

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO)
- PLLCCR.PLSRCSEL = 1 (PLL source clock = HOCO) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL)
- PLLCCR.PLSRCSEL = 1 (PLL source clock = HOCO) and PLLCR.PLLSTP = 0 (PLL is operating).

9.2.11 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR)

Address(es): SYSTEM.HOCOWTCR 4001 E0A5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HSTS[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	HSTS[2:0]	HOCO Wait Time Setting	Wait time (s) = (HSTS[2:0] setting +3) /fLOCO	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

HSTS[2:0] bit (HOCO Wait Time Setting)

The HOCOWTCR.HSTS[2:0] bits must be set to 110b, except when using SCI0 in Snooze mode. When using SCI0 in Snooze mode, HOCOWTCR.HSTS[2:0] must be set to 010b.

9.2.12 Middle-Speed On-Chip Oscillator Control Register (MOCOCR)

Address(es): SYSTEM.MOCOCR 4001 E038h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MCSTP	MOCO Stop	0: Operate the MOCO clock 1: Stop the MOCO clock.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MOCOCR register controls the MOCO clock.

MCSTP bit (MOCO Stop)

The MCSTP bit starts or stops the MOCO clock. After setting MCSTP to 0 to start the MOCO clock, only use the clock after the MOCO clock oscillation stabilization time (tMOCOWT) elapses. A fixed stabilization wait is required after setting the MOCO clock to start operation. A fixed wait for oscillation to stop is also required.

The following constraints apply when starting and stopping operation:

- After stopping the MOCO clock, allow a stop interval of at least 5 MOCO cycles before restarting it
- Confirm that MOCO oscillation is stable before stopping the MOCO clock
- Regardless of whether the MOCO clock is selected as the system clock, confirm that MOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow a setting to stop the MOCO clock, wait for at least 3 MOCO clock cycles after the stop setting before executing the WFI instruction.

Writing 1 to MCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 001b (system clock source = MOCO).

Writing 1 to the MCSTP bit (stopping the MOCO) is prohibited if oscillation stop detection is enabled in the Oscillation Stop Detection Control Register (OSTDCR.OSTDE).

9.2.13 FLL Control Register 1 (FLLCR1)

Address(es): SYSTEM.FLLCR1 4001 E039h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FLLEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FLLEN	FLL Enable	0: FLL function is disabled 1: FLL function is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: HOCO must be stopped (HOCOCR.HCSTP = 1) before FLLCR1.FLLEN is changed.
Note: SOSC must be operating with stabilization while FLL is enabled (FLLCR1.FLLEN = 1).

The FLLCR1 register controls the FLL function of the HOCO. The purpose of FLL is to utilize SOSC when available for better accuracy in HOCO.

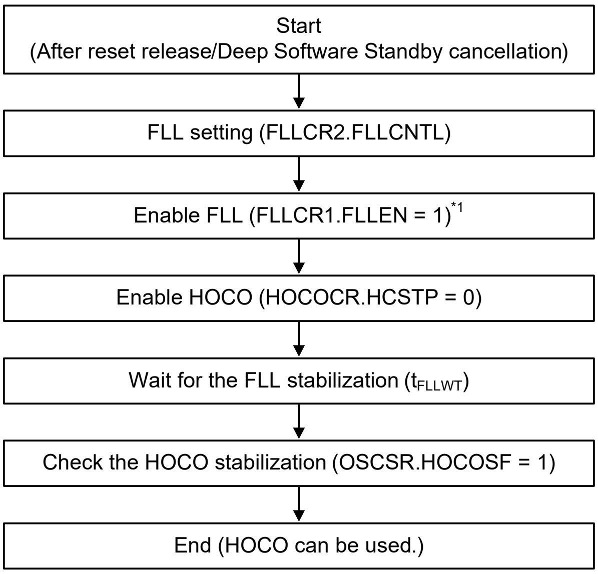
FLLEN bit (FLL Enable)

This bit enables or disables the FLL function of the HOCO.

If FLL is enabled, the frequency accuracy is guaranteed after FLL is stabilized. The FLL stabilization can be checked by the frequency measurement of the Clock Frequency Accuracy Measurement Circuit (CAC) after the HOCO is stabilized.

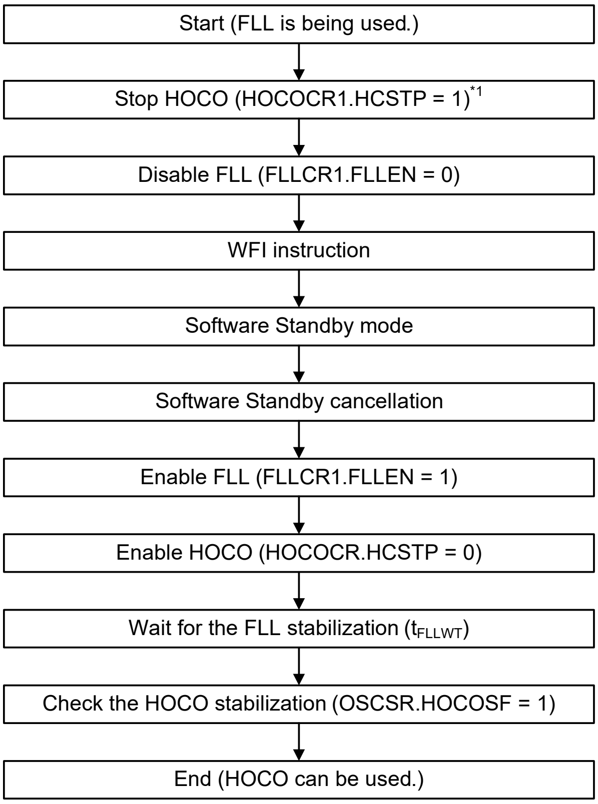
The FLL must be disabled before the transition to Software Standby mode. Therefore, this bit must be set to 0 before the transition to Software Standby mode.

Figure 9.4 and Figure 9.5 show an example flow of the FLL setting in each case.



Note 1. SOSC must be running with the oscillation stabilized.

Figure 9.4 **FLL setting flow (after reset release / Deep Software Standby cancellation)**



Note 1. If HOCO is used as the system clock or the PLL reference clock, these clock sources must be changed to another clock before HOCO is stopped.

Figure 9.5 **Software Standby transition / cancellation flow**

9.2.14 FLL Control Register 2 (FLLCR2)

Address(es): SYSTEM.FLLCR2.4001 E03Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FLLCNTL[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	FLLCNTL[10:0]	FLL Multiplication Control	• When OFS1.HOCOFRQ0[1:0] is 00b (16 MHz), these bits must be set to 1E9h • When OFS1.HOCOFRQ0[1:0] is 01b (18 MHz), these bits must be set to 226h • When OFS1.HOCOFRQ0[1:0] is 10b (20 MHz), these bits must be set to 263h. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The FLLCR2 register controls the FLL function of the HOCO.

FLLCNTL[10:0] bits (FLL Multiplication Control)

These bits select the multiplication ratio of the FLL reference clock.

These bits must be set before FLL is enabled (FLLCR1.FLLEN = 1).

9.2.15 Oscillation Stabilization Flag Register (OSCSF)

Address(es): SYSTEM.OSCSF 4001 E03Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PLLSF	—	MOSCSF	—	—	HOCOSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit name	Description	R/W
b0	HOCOSF	HOCO Clock Oscillation Stabilization Flag	0: HOCO clock is stopped or is not yet stable 1: HOCO clock is stable, so is available for use as the system clock.	R
b2, b1	—	Reserved	These bits are read as 0.	R
b3	MOSCSF	Main Clock Oscillation Stabilization Flag	0: Main clock oscillator is stopped (MOSTP = 1) or is not yet stable*2 1: Main clock oscillator is stable, so is available for use as the system clock.	R
b4	—	Reserved	This bit is read as 0.	R
b5	PLLSF	PLL Clock Oscillation Stabilization Flag	0: PLL clock is stopped or is not yet stable 1: PLL clock is stable, so is available for use as the system clock.	R
b7, b6	—	Reserved	These bits are read as 0.	R

Note 1. The value after reset depends on the OFS1.HOCOEN setting.
When OFS1.HOCOEN = 1, the value after reset of HOCOSF is 0.
When OFS1.HOCOEN = 0, the HOCOSF value is set to 0 immediately after reset is released, and HOCOSF is set to 1 after the HOCO oscillation stabilization wait time elapses.

Note 2. This is true when an appropriate value is set in the wait control register for the given oscillator. If the wait time value is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable.

The OSCSF register flags indicate the operating status of the counters in the oscillation stabilization wait circuits for the individual oscillators. After oscillation starts, these counters measure the wait time until their associated oscillator output clocks are supplied to the internal circuits. An overflow of a counter indicates that the clock supply is stable and available for the associated circuit.

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

The HOCOSF flag indicates the operating status of the counter that measures the wait time for the high-speed clock oscillator (HOCO). When OFS1.HOCOEN is set to 1, confirm that the OSCSF.HOCOSF is also set to 1 before using the HOCO clock.

[Setting condition]

- After the HOCO clock stops and the HOCOCR.HCSTP bit is set to 0, supply of the MCU clock starts after the number of LOCO cycles associated with the setting of the HOCOWTCR register elapse.

[Clearing condition]

- When the HOCO clock is operating and then is deactivated because the HOCOCR.HCSTP bit is set to 1.

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

The MOSCSF flag indicates the operating status of the counter that measures the wait time for the main clock oscillator.

[Setting condition]

- After the main clock oscillator stops and the MOSCCR.MOSTP bit is set to 0, supply of the MCU clock starts after the number of LOCO cycles associated with the setting of the MOSCWTCR register elapse.

[Clearing condition]

- When the main clock oscillator is operating and then is deactivated because the MOSCCR.MOSTP bit is set to 1.

PLLSF flag (PLL Clock Oscillation Stabilization Flag)

The PLLSF flag indicates the operating status of the counter that measures the wait time for the PLL.

[Setting condition]

- After the PLL stops and the PLLCR.PLLSTP bit is set to 0, supply of the MCU clock starts after 31 LOCO cycles. If oscillation by the PLL clock source selected in the PLLCCR.PLSRCSEL bit is not stable when the PLLSTP bit is set to 0, counting of the LOCO cycles continues after the PLL clock source oscillation is stabilized. Wait time is calculated as:

  1 cycle = LOCO (32.768 kHz) x 8 (3.81 µs typical).

[Clearing condition]

- When the PLL is operating and then is deactivated because the PLLCCR.PLLSTP bit is set to 1.

9.2.16 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): SYSTEM.OSTDCR 4001 E040h

	b7	b6	b5	b4	b3	b2	b1	b0
	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: Disable oscillation stop detection interrupt (do not notify the POEG) 1: Enable oscillation stop detection interrupt (notify the POEG).	R/W

Bit	Symbol	Bit name	Description	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Disable oscillation stop detection function 1: Enable oscillation stop detection function.	R/W

The OSTDCR register controls the oscillation stop detection function.

OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

The OSTDIE bit enables the oscillation stop detection function interrupt. It also controls whether oscillation stop detection is reported to the POEG.

If the oscillation stop detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, clear the OSTDIE bit to 0 before clearing OSTDF. Wait for at least 2 cycles of PCLKB before setting OSTDIE to 1. A longer PCLKB wait time might be required, depending on the number of cycles required to read a given I/O register.

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function. When OSTDE is 1 (enable), the MOCO stop bit (MOCOCR.MCSTP) is cleared to 0 and MOCO operation starts. The MOCO clock cannot be stopped while the oscillation stop detection function is enabled. Writing 1 to the MOCOCR.MCSTP bit (MOCO stopped) is invalid.

When the oscillation stop detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

OSTDE must be cleared before invoking Software Standby or Deep Software Standby mode. To transition to either of these modes, first clear OSTDE to 0 and then execute the WFI instruction.

The following constraints apply when using the oscillation stop detection function:

- In low-speed mode, selecting division by 1, 2, 4, 8 for ICLK, FCLK, BCLK, PCLKA, PCLKB, PCLKC, PCLKD is prohibited.

9.2.17 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): SYSTEM.OSTDSR 4001 E041h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: Main clock oscillation stop not detected 1: Main clock oscillation stop detected.	R(/W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit can only be set to 0.

The OSTDSR register indicates the stop detection status of the main clock oscillator.

OSTDF flag (Oscillation Stop Detection Flag)

The OSTDF flag indicates the main clock oscillator status. When OSTDF is 1, it indicates that a main clock oscillation stop was detected. After this stop is detected, the OSTDF bit is not cleared to 0 even when oscillation is restarted. The OSTDF bit is cleared to 0 by writing 0 after reading it as 1.

At least 3 ICLK cycles of wait time are required between writing 0 to OSTDF and reading OSTDF as 0. If the OSTDF bit is cleared to 0 when the main clock oscillation is stopped, the OSTDF bit becomes 0 and then returns to 1.

OSTDSR.OSTDF cannot be cleared to 0 under the following conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL).

The OSTDF bit must be set to 0 after switching the clock source to sources other than the main clock oscillator and PLL.

[Setting condition]

- The main clock oscillator is stopped while OSTDCR.OSTDE = 1 (oscillation stop detection enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are not 011b (system clock = MOSC) or 101b (system clock = PLL) and the PLLCCR.PLSRCSEL bit is not 0 (PLL source clock = MOSC).

9.2.18 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): SYSTEM.MOSCWTCR 4001 E0A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MSTS[3:0]			
Value after reset:	0	0	0	0	0	1	0	1

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting	When drive capability automatic switching function is disabled (MOMCR.AUTODRVEN = 0 [default]): b3 b0 0 0 0 1: Wait time = 35 cycles (133.5 µs) 0 0 1 0: Wait time = 67 cycles (255.6 µs) 0 0 1 1: Wait time = 131 cycles (499.7 µs) 0 1 0 0: Wait time = 259 cycles (988.0 µs) 0 1 0 1: Wait time = 547 cycles (2086.6 µs) (value after reset) 0 1 1 0: Wait time = 1059 cycles (4039.8 µs) 0 1 1 1: Wait time = 2147 cycles (8190.2 µs) 1 0 0 0: Wait time = 4291 cycles (16368.9 µs) 1 0 0 1: Wait time = 8163 cycles (31139.4 µs). When drive capability automatic switching function is enabled (MOMCR.AUTODRVEN = 1): b3 b0 0 0 0 1: Wait time = 36 cycles (137.3 µs) 0 0 1 0: Wait time = 68 cycles (259.4 µs) 0 0 1 1: Wait time = 132 cycles (503.5 µs) 0 1 0 0: Wait time = 260 cycles (991.8 µs) 0 1 0 1: Wait time = 548 cycles (2090.5 µs) (value after reset) 0 1 1 0: Wait time = 1060 cycles (4043.6 µs) 0 1 1 1: Wait time = 2148 cycles (8194.0 µs) 1 0 0 0: Wait time = 4292 cycles (16372.7 µs) 1 0 0 1: Wait time = 8164 cycles (31143.2 µs). Other settings are prohibited. Wait time is calculated as: 1 cycle (µs) = 1 / (f_LOCO [MHz] × 8) = 1 / (0.032768 × 8) = 3.81 µs	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

MSTS[3:0] bits (Main Clock Oscillator Wait Time Setting)

Set the MSTS[3:0] bits to select the oscillation stabilization wait time for the main clock oscillator. Specify a time period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 0001b, because the oscillation stabilization time is not required.

The wait time set in these bits is counted using:

1 cycle (μs) = 1 / (f_ LOCO [MHz] × 8) = 1 / (0.032768 x 8) = 3.81 (μs).

The LOCO clock automatically oscillates when necessary, regardless of the value of the LOCOCR.LOSTP bit. After the specified wait time elapses, supply of the main clock oscillator starts internally in the MCU, and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock oscillator starts before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCSF.MOSCSF flag is 0. Do not rewrite this register under any other conditions.

9.2.19 Main Clock Oscillator Mode Oscillation Control Register (MOMCR)

Address(es): SYSTEM.MOMCR 4001 E413h

	b7	b6	b5	b4	b3	b2	b1	b0
	AUTODRVEN	MOSEL	MODRV0[1:0]		—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	MODRV0[1:0]	Main Clock Oscillator Drive Capability 0 Switching	b5 b4 0 0: 20 to 24 MHz 0 1: 16 to 20 MHz 1 0: 8 to 16 MHz 1 1: 8 MHz.	R/W
b6	MOSEL	Main Clock Oscillator Switching	0: Resonator 1: External clock input.	R/W
b7	AUTODRVEN	Main Clock Oscillator Drive Capability Auto Switching Enable	0: Disable 1: Enable.	R/W

Note: The EXTAL/XTAL pins are also used as ports. In the initial state, the port function is selected.
Note: The MOSTP bit must be 1 (MOSC = stopped) before changing this register.

MODRV0[1:0] bits (Main Clock Oscillator Drive Capability 0 Switching)

The MODRV0[1:0] bits switch the drive capability of the main clock oscillator.

MOSEL bit (Main Clock Oscillator Switching)

The MOSEL bit switches the source for the main clock oscillator.

AUTODRVEN bit (Main Clock Oscillator Drive Capability Auto Switching Enable)

The AUTODRVEN bit controls the drive capability auto switching of the main clock oscillator.

When AUTODRVEN = 1, after the time set in the MSTS bits in the Main Clock Oscillator Wait Control Register elapses, the effective main clock oscillator drive capability is automatically set to the lowest, regardless of the MOMCR.MODRV0[1:0] setting. The main clock oscillator restarts oscillation with MOMCR.MODRV0 specified drive capability after oscillation stops by MOSCCR.MOSTP setting or by entering Software Standby mode.

9.2.20 Subclock Oscillator Mode Control Register (SOMCR)

Address(es): SYSTEM.SOMCR 4001 E481h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SODRV1	—
Value after reset:	0	0	0	0	0	0	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SODRV1	Sub-Clock Oscillator Drive Capability Switching	0: Standard 1: Low.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: The SOSCCR.SOSTP bit must be 1 (SOSC = stopped) before changing this register.

SODRV1 bit (Sub-Clock Oscillator Drive Capability Switching)

The SODRV1 bit switches the drive capability of the sub-clock oscillator. This bit is undefined at the first power on, but the value after reset of SOSCCR.SOSTP is 0 (SOSC = operating). Set up the SOSC as follows at the first power on:

1. Set the SOSCCR.SOSTP bit to 1 (SOSC = stopped).
2. Set this bit to the correct value for the current capacitor.
3. Clear the SOSCCR.SOSTP to 0 (SOSC = operating).

9.2.21 Clock Out Control Register (CKOCR)

Address(es): SYSTEM.CKOCR 4001 E03Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKOEN	CKODIV[2:0]			—	CKOSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CKOSEL[2:0]	Clock Out Source Select	b2 b0 0 0 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: MOSC 1 0 0: SOSC. Other settings are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CKODIV[2:0]	Clock Out Input Frequency Division Select	b6 b4 0 0 0: ×1 0 0 1: /2 0 1 0: /4 0 1 1: /8 1 0 0: /16 1 0 1: /32 1 1 0: /64 1 1 1: /128.	R/W
b7	CKOEN	Clock Out Enable	0: Disable clock out 1: Enable clock out.	R/W

CKOSEL[2:0] bits (Clock Out Source Select)

The CKOSEL[2:0] bits specify the HOCO, MOCO, LOCO, MOSC, or SOSC clock as the source of the clock to be output from the CLKOUT pin. When changing the CLKOUT source clock, clear the CKOEN bit to 0.

CKODIV[2:0] bits (Clock Out Input Frequency Division Select)

The CKODIV[2:0] bits specify the clock division ratio. Clear the CKOEN bit to 0 when changing the division ratio. The division ratio of the output clock frequency must be set to a value no higher than the characteristics of the CLKOUT pin output frequency. For details on the characteristics of the CLKOUT pin, see section 60, Electrical Characteristics.

CKOEN bit (Clock Out Enable)

The CKOEN bit enables output from the CLKOUT pin. When CKOEN is set to 1, the selected clock is output. When CKOEN is set to 0, low is output. When changing this bit, confirm that the clock out source clock selected in the CKOSEL[2:0] bits is stable. Otherwise, a glitch might be generated in the output.

The CKOEN bit must be cleared before entering Software Standby or Deep Software Standby mode if the selected clock out source clock is stopped in that mode.

9.2.22 External Bus Clock Output Control Register (EBCKOCR)

Address(es): SYSTEM.EBCKOCR 4001 E052h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EBCKOEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EBCKOEN	EBCLK Pin Output Control	0: Disable EBCLK pin output (fixed high) 1: Enable EBCLK pin output.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

9.2.23 SDRAM Clock Output Control Register (SDCKOCR)

Address(es): SYSTEM.SDCKOCR 4001 E053h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDCKOEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SDCKOEN	SDCLK Pin Output Control	0: Disable SDCLK pin output (fixed high) 1: Enable SDCLK pin output.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

9.2.24 LOCO User Trimming Control Register (LOCOUTCR)

Address(es): SYSTEM.LOCOUTCR 4001 E492h

	b7	b6	b5	b4	b3	b2	b1	b0
	LOCOUTRM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	LOCOUTRM[7:0]	LOCO User Trimming	b7 b0 1 0 0 0 0 0 0 0: -128 1 0 0 0 0 0 0 1: -127 1 0 0 0 0 0 1 0: -126 … 1 1 1 1 1 1 1 1: -1 0 0 0 0 0 0 0 0: Center Code 0 0 0 0 0 0 0 1: +1 … 0 1 1 1 1 1 0 1: +125 0 1 1 1 1 1 1 0: +126 0 1 1 1 1 1 1 1: +127. These bits are added to the original LOCO trimming bits.	R/W

Note: MCU operation is not guaranteed when LOCOUTCR is set to a value that causes the LOCO frequency to be outside of the specification range.

Note: When LOCOUTCR is changed, the frequency stabilization wait required corresponds to the frequency stabilization wait at the start of MCU operation.

9.2.25 MOCO User Trimming Control Register (MOCOUTCR)

Address(es): SYSTEM.MOCOUTCR 4001 E061h

	b7	b6	b5	b4	b3	b2	b1	b0
	MOCOUTRM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	MOCOUTRM[7:0]	MOCO User Trimming	b7 b0 1 0 0 0 0 0 0 0: -128 1 0 0 0 0 0 0 1: -127 1 0 0 0 0 0 1 0: -126 … 1 1 1 1 1 1 1 1: -1 0 0 0 0 0 0 0 0: Center Code 0 0 0 0 0 0 0 1: +1 … 0 1 1 1 1 1 0 1: +125 0 1 1 1 1 1 1 0: +126 0 1 1 1 1 1 1 1: +127. These bits are added to the original MOCO trimming bits.	R/W

Note: MCU operation is not guaranteed when MOCOUTCR is set to a value that causes the MOCO frequency to be outside of the specification range.

Note: When MOCOUTCR is changed, the frequency stabilization wait required corresponds to the frequency stabilization wait at the start of MCU operation.

9.2.26 HOCO User Trimming Control Register (HOCOUTCR)

Address(es): SYSTEM.HOCOUTCR 4001 E062h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOCOUTRM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	HOCOUTRM[7:0]	HOCO User Trimming	b7 b0 1 0 0 0 0 0 0 0: -128 1 0 0 0 0 0 0 1: -127 1 0 0 0 0 0 1 0: -126 … 1 1 1 1 1 1 1 1: -1 0 0 0 0 0 0 0 0: Center Code 0 0 0 0 0 0 0 1: +1 … 0 1 1 1 1 1 0 1: +125 0 1 1 1 1 1 1 0: +126 0 1 1 1 1 1 1 1: +127. These bits are added to the original HOCO trimming bits.	R/W

Note: MCU operation is not guaranteed when HOCOUTCR is set to a value that causes the HOCO frequency to be outside of the specification range.

Note: When HOCOUTCR is changed, the frequency stabilization wait required corresponds to the frequency stabilization wait at the start of MCU operation.

Note: These bits must be 00000000b when FLL is enabled (FLLCR1.FLLEN = 1).

9.2.27 Trace Clock Control Register (TRCKCR)

Address(es): SYSTEM.TRCKCR 4001 E03Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TRCKEN	—	—	—	TRCK[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b3 to b0	TRCK[3:0]	Trace Clock Operating Frequency Select	b3 b0 0 0 0 0: /1 0 0 0 1: /2 (value after reset) 0 0 1 0: /4. Other settings are prohibited.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TRCKEN	Trace Clock Operation Enable	0: Disable operation 1: Enable operation.	R/W

The Trace Clock Control Register controls the switching of the trace clock. Before changing the TRCLK frequency, set the TRCKEN bit to 0. The TRCKCR register is initialized by all reset sources.

9.3 Main Clock Oscillator

Use one of the following ways to supply the clock signal to the main clock oscillator:

- Connect an oscillator
- Connect the input of an external clock signal.

9.3.1 Connecting the Crystal Resonator

Figure 9.6 shows an example connection to a crystal resonator. A damping resistor (Rd) can be added, if required. Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between EXTAL and XTAL by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 9.1.

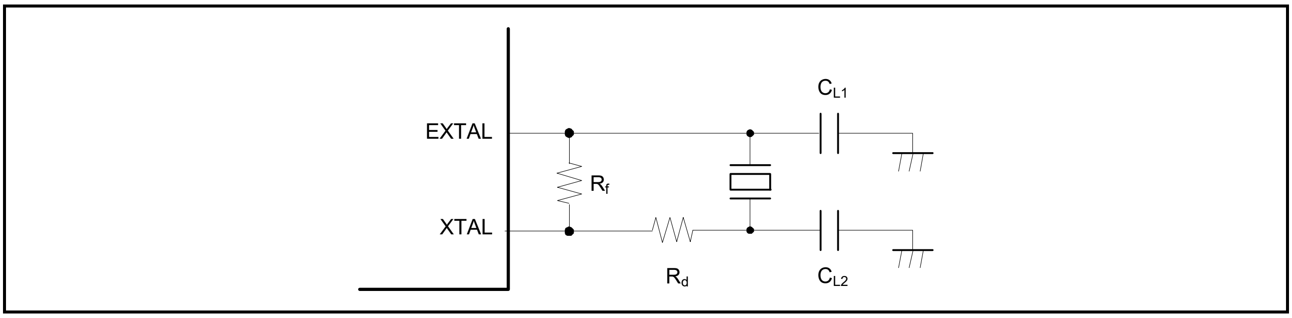


Figure 9.6 Example of crystal resonator connection

Figure 9.7 shows an equivalent circuit of the crystal resonator.

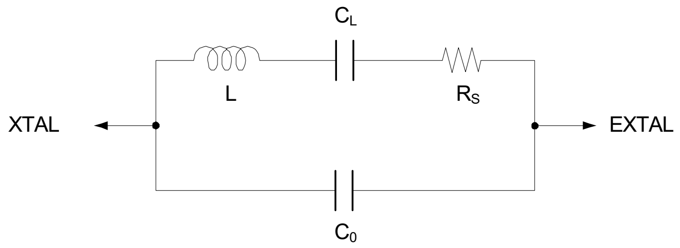


Figure 9.7 Equivalent circuit of the crystal resonator

9.3.2 External Clock Input

Figure 9.8 shows an example connection to an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin is the function that is set in PFS.P213PFS.

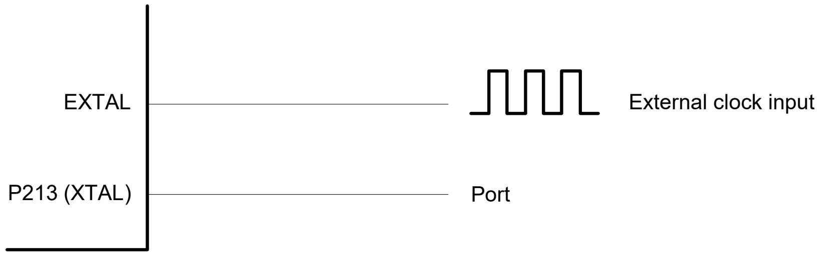


Figure 9.8 Equivalent circuit for external clock

9.3.3 Notes on External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input when the main clock oscillator stop bit (MOSCCR.MOSTP) is 0.

9.4 Sub-Clock Oscillator

The only way of supplying a clock signal to the sub-clock oscillator is by connecting a crystal oscillator.

9.4.1 Connecting a 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator as shown in Figure 9.9. A damping resistor (Rd) can be added, if required. Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between XCIN and XCOUT by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the sub-clock oscillator as described in Table 9.1.

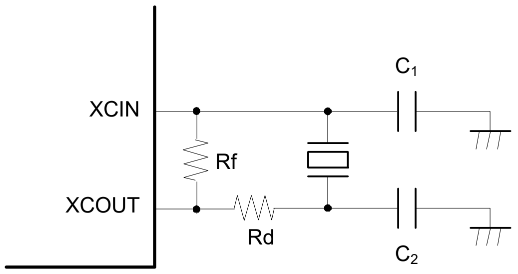


Figure 9.9 Connection example of 32.768-kHz crystal resonator

Figure 9.10 shows an equivalent circuit for the 32.768-kHz crystal resonator.

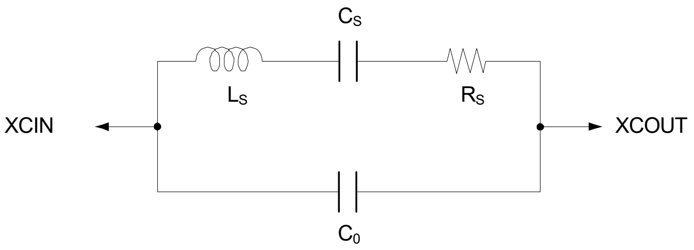


Figure 9.10 Equivalent circuit for the 32.768-kHz crystal resonator

9.4.2 Handling of Pins When the Sub-Clock Oscillator Is Not Used

When the sub-clock oscillator is not in use, connect the XCIN pin to VSS through a resistor (to pull VSS down) and leave the XCOUT pin open as shown in Figure 9.11. In addition, if an oscillator is not connected, set the sub-clock oscillator stop bit (SOSCCR.SOSTP) to 1 to stop the oscillator.

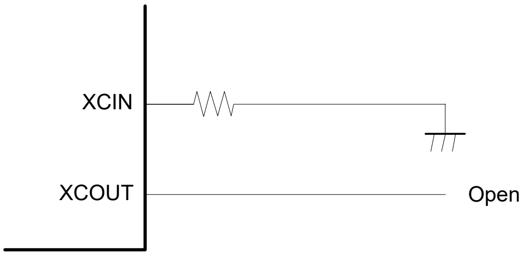


Figure 9.11 Pin handling when the sub-clock oscillator is not used

9.5 Oscillation Stop Detection Function

9.5.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function detects the main clock oscillator stop. When an oscillation stop is detected, the system clock switches as follows:

- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC), the system clock source switches to the MOCO clock.
- If an oscillation stop is detected with PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL), the PLL clock remains as the system clock source. The frequency becomes free-running, and the setting in the SCKSCR.CKSEL[2:0] bits does not change.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the General PWM Timer (GPT) output can be forced to a high-impedance state on detection.

The main clock oscillation stop is detected when the input clock remains at 0 or 1 for a certain period, for example, when a malfunction occurs in the main clock oscillator. See section 60, Electrical Characteristics.

Switching between the main clock oscillator and the MOCO clock or between the PLL clock and the PLL free-running clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF).

The OSTDF flag controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC):
  - When OSTDF changes from 0 to 1, the clock source switches to the MOCO clock.
  - When OSTDF changes from 1 to 0, the clock source switches to MOSC again.
- When PLLCCR.PLSRCSEL = 0 (PLL source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL)
  - When OSTDF changes from 0 to 1, the clock source switches to the PLL free-running oscillation clock.
  - When OSTDF changes from 1 to 0, the clock source switches to PLL again.

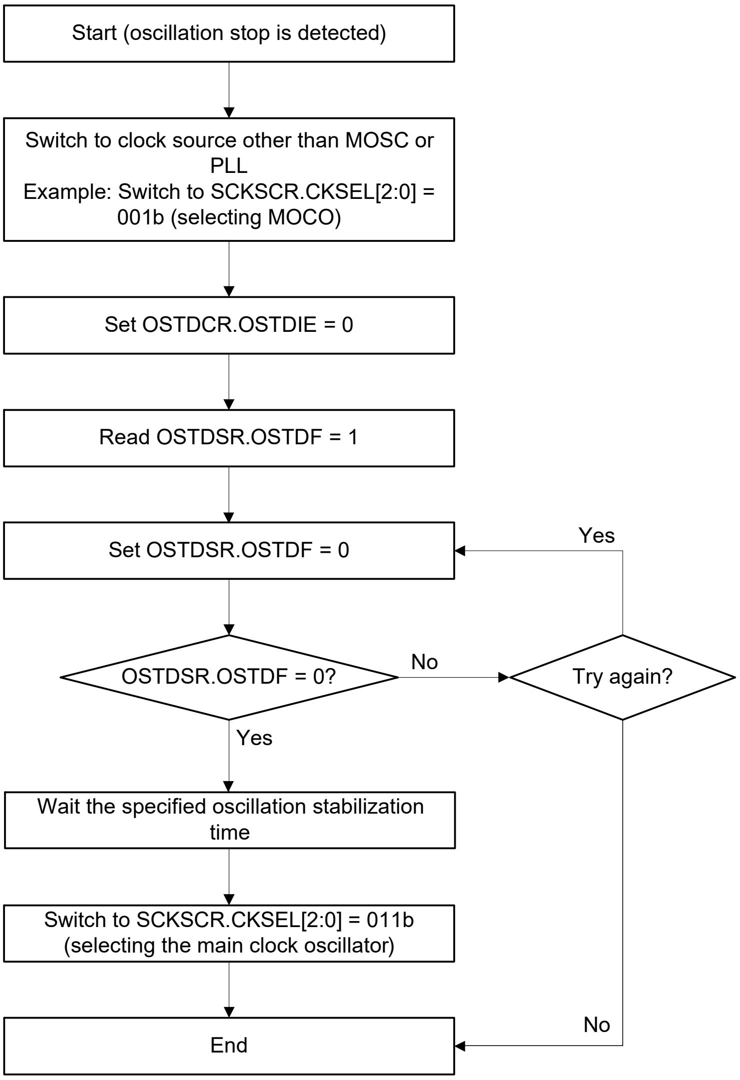
To switch the clock source to the main clock oscillator or PLL clock again after oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock oscillator or PLL clock, and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock oscillator or PLL clock after the specified oscillation stabilization time elapses.

After a reset release, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after the specified oscillation stabilization time elapses.

The oscillation stop detection function detects when the main clock oscillator is stopped by an external cause. This means that the oscillation stop detection function must be disabled before the main clock oscillator is stopped by software or before entering Software Standby or Deep Software Standby mode.

The oscillation stop detection function switches the following clocks to the MOCO clock (when the system clock is MOSC) or the PLL free-running clock (when the system clock is PLL):

- All clocks that can be selected as the MOSC clock or PLL except CLKOUT
- The system clock (ICLK) frequency during MOCO operation (when the system clock is MOSC) or PLL free-running operation (when the system clock is PLL) is specified in the MOCO oscillation frequency and the division ratio set in the system clock select bits (SCKDIVCR.ICK[2:0]).



Example of returning from oscillation stop detection when CKSEL[2:0] = 011b (selecting the main clock oscillator)

Note: On returning from the oscillation-stopped state, the factor responsible for stopping the main clock oscillator must be removed from the system to allow oscillation to resume.

Figure 9.12 Flow of recovery on detection of oscillator stop

9.5.2 Oscillation Stop Detection Interrupts

An oscillation stop detection interrupt (MOSC_STOP) is generated when the oscillation stop detection flag (OSTDSR.OSTDF) is 1 and the oscillation stop detection interrupt enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE) is 1 (enabled). The Port Output Enable for GPT (POEG) is notified of the main clock oscillator stop. On receiving the notification, the POEG sets the Oscillation Stop Detection Flag in the POEG Group n Setting Register (POEGGn.OSTPF) to 1 (n = A, B, C, D).

After the oscillation stop is detected, wait at least 10 cycles of PCLKB before writing to the POEGGn.OSTPF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the oscillation stop detection interrupt enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE). Wait for at least 2 cycles of the PCLKB clock before setting the OSTDCR.OSTDIE bit to 1 again. A longer PCLKB wait time might be required, depending on the number of cycles required to read a given I/O register.

The oscillation stop detection interrupt is a non-maskable interrupt. Because non-maskable interrupts are disabled in the initial state after a reset release, enable non-maskable interrupts through software before using oscillation stop detection interrupts. For details, see section 14, Interrupt Controller Unit (ICU).

9.6 PLL Circuit

The PLL circuit provides a function for multiplying the frequency from the oscillator.

9.7 Internal Clock

Clock sources for the internal clock signals include:

- Main clock oscillator
- Sub-clock oscillator
- HOCO clock
- MOCO clock
- LOCO clock
- PLL clock
- Dedicated clock for the IWDT
- External clock for JTAG.

The following internal clocks are produced from these sources:

- Operating clock for the CPU, DMAC, DTC, flash memory, and SRAM — system clock (ICLK)
- Operating clocks for peripheral modules — PCLKA, PCLKB, PCLKC, and PCLKD
- Operating clock for the flash interface — FCLK
- Clock for the external bus controller and external pin output — EBCLK
- Clock for the external bus controller and external pin output for the SDRAM — SDCLK
- Operating clock for the USBFS and USBHS — UCLK
- Operating clock for the USBHS — USBMCLK
- Operating clock for the CAN — CANMCLK
- Operating clock for the CAC — CACCLK
- Operating clock for the RTC LOCO clock — RTCLCLK
- Operating clock for the RTC sub-clock — RTCSCLK
- Operating clock for the IWDT — IWDTCLK
- Operating clock for the AGT LOCO clock — AGTLCLK
- Operating clock for the AGT sub-clock — AGTSCLK
- Operating clock for the SysTick timer — SYSTICCLK
- Clock for external pin output — CLKOUT
- Operating clock for the JTAG — JTAGTCK.

For details on the registers used to set the frequencies of the internal clocks, see section 9.7.1, System Clock (ICLK) to section 9.7.15, JTAG Clock (JTAGTCK). If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

9.7.1 System Clock (ICLK)

The system clock, ICLK, is the operating clock for the CPU, DMAC, DTC, flash memory, and SRAM. Specify the frequency in the following bits:

- ICK[2:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- HOCOFRQ[1:0] bits in OFS1.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See Figure 9.13 and Figure 9.14.

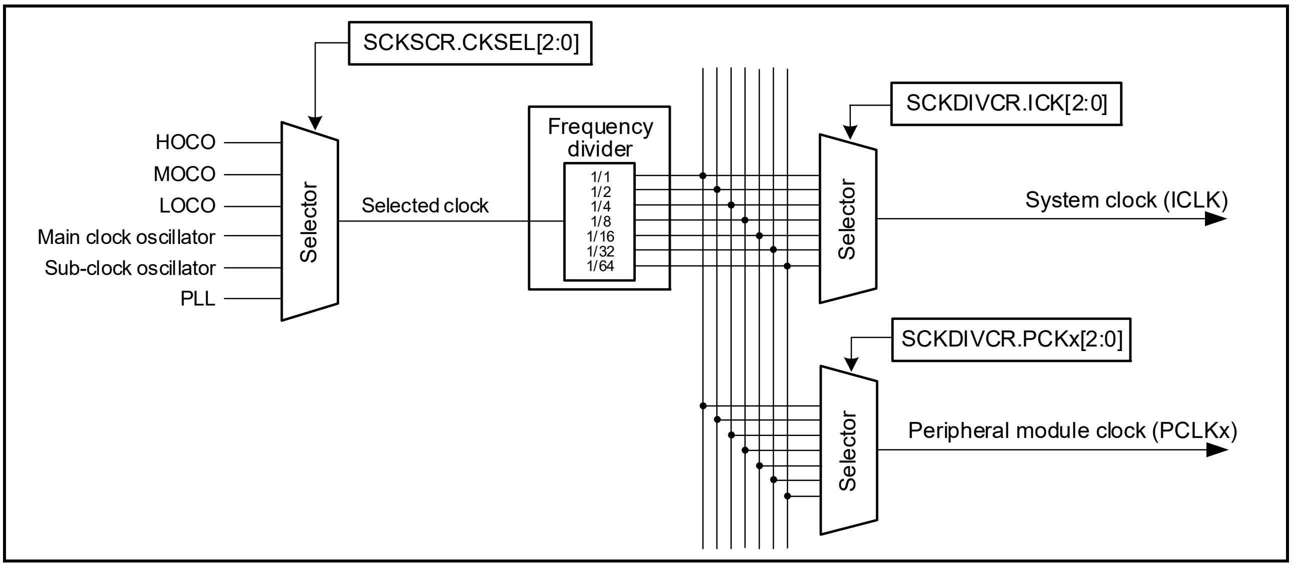


Figure 9.13 Clock source selector block diagram

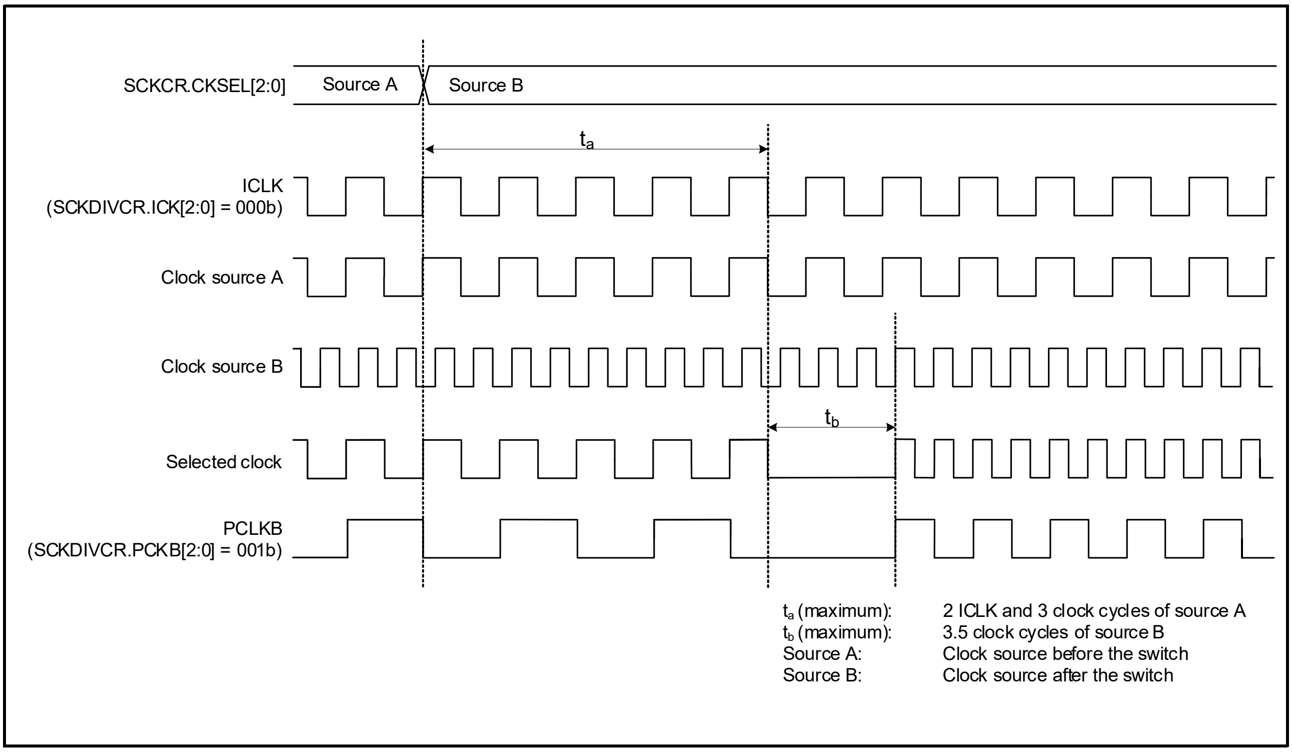


Figure 9.14 Clock source switching timing diagram

9.7.2 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD)

The peripheral module clocks, PCLKA, PCLKB, PCLKC, and PCLKD, are the operating clocks for the peripheral modules. Specify the frequency in the following bits:

- PCKA[2:0], PCKB[2:0], PCKC[2:0], and PCKD[2:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- HOCOFRQ[1:0] bits in OFS1.

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer during the clock source transition period. See Figure 9.13 and Figure 9.14.

9.7.3 Flash Interface Clock (FCLK)

The flash interface clock, FCLK, is the operating clock for the flash memory interface. In addition to reading from the data flash, it is used for the programming and erasure of the code flash and data flash. Specify the frequency in the following bits:

- FCK[2:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- HOCOFRQ[1:0] bits in OFS1.

9.7.4 External Bus Clock (BCLK)

The external bus clock, BCLK, is an operating clock for the external bus controller. It is also output externally from the EBCLK pin for the external connection bus. To output BCLK from the EBCLK pin, set the EBCKOCR.EBCKOEN bit to 1 and set the PmnPFS.PSEL[4:0] bits to 01011b. Only change the PmnPFS.PSEL[4:0] bits to 01011b when the EBCKOCR.EBCKOEN bit is 0. When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the EBCLK pin. Specify the frequency in the following bits:

- BCK[2:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR
- HOCOFRQ[1:0] bits in OFS1.

Do not set BCLK to a frequency higher than that of the system clock (ICLK).

9.7.5 SDRAM Clock (SDCLK)

The SDRAM clock, SDCLK, is an operating clock for the external bus controller. It is output externally from the SDCLK pin for the SDRAM that is connected to the external bus. To output SDCLK on the SDCLK pin, set the SDCKOCR.SDCKOEN bit to 1 and set the PmnPFS.PSEL[4:0] bits to 01011b (enabling SDCLK output). Only change the value in the PmnPFS.PSEL[4:0] bits when the SDCKOCR.SDCKOEN bit is 0. Specify the frequency in the following bits:

- SCKDIVCR.BCK[2:0], SCKSCR.CKSEL[2:0], the PLLMUL[5:0], and PLIDIV[1:0] bits in PLLCCR
- HOCOFRQ[1:0] bits in OFS1.

Do not set SDCLK to a frequency higher than that of the system clock (ICLK).

9.7.6 USB Clock (UCLK)

The USB clock, UCLK, is the operating clock for the USBFS module. A 48-MHz clock must be supplied to the USBFS module. When the module is used, the UCLK clock must be specified as 48 MHz. Specify the frequency in the following bits:

- UCK[2:0] bits in SCKDIVCR2
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[5:0] and PLIDIV[1:0] bits in PLLCCR.

9.7.7 USB-PHY Clock (USBMCLK)

The USB-PHY clock, USBMCLK, is the operating clock for the USBHS-PHY. The USBMCLK frequency is 12, 20, or 24 MHz supplied from the main clock oscillator.

9.7.8 CAN Clock (CANMCLK)

The CAN clock, CANMCLK, is the operating clock for the CAN module. CANMCLK is generated by the main clock oscillator.

9.7.9 CAC Clock (CACCLK)

The CAC clock, CACCLK, is the operating clock for the CAC. CACCLK is generated by the following oscillators:

- Main clock oscillator
- Sub-clock oscillator
- High-speed clock oscillator (HOCO)
- Middle-speed clock oscillator (MOCO)
- Low-speed on-chip oscillator (LOCO)
- IWDT-dedicated on-chip oscillator.

9.7.10 RTC-Dedicated Clock (RTCSCLK, RTCLCLK)

The RTC-dedicated clocks, RTCSCLK and RTCLCLK, are the operating clocks for the RTC. RTCSCLK is generated by the sub-clock oscillator and RTCLCLK by the LOCO clock.

9.7.11 IWDT-Dedicated Clock (IWDTCLK)

The IWDT-dedicated clock, IWDTCLK, is the operating clock for the IWDT. IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

9.7.12 AGT-Dedicated Clock (AGTSCLK, AGTLCLK)

The AGT-dedicated clocks, AGTSCLK and AGTLCLK, are the operating clocks for the AGT. AGTSCLK is generated by the sub-clock oscillator and AGTLCLK is generated by the LOCO clock.

9.7.13 SysTick Timer-Dedicated Clock (SYSTICCLK)

The SysTick timer-dedicated clock, SYSTICCLK, is the operating clock for the SYSTICCLK. SYSTICCLK is generated by the LOCO clock.

9.7.14 Clock/Buzzer Output Clock (CLKOUT)

The CLKOUT is output externally from the CLKOUT pin for the clock or buzzer output. CLKOUT is output to the CLKOUT pin when CKOCR.CKOEN is set to 1. Only change the value in the CKODIV[2:0] or CKOSEL[2:0] bits in CKOCR when the CKOCR.CKOEN bit is 0. Specify the frequency in the following bits:

- CKODIV[2:0] or CKOSEL[2:0] bits in CKOCR
- HOCOFRQ[1:0] bits in OFS1.

9.7.15 JTAG Clock (JTAGTCK)

The JTAG-dedicated clock, JTAGTCK, is the operating clock for the JTAG. JTAGTCK is generated by the external clock for JTAG (TCK).

9.8 Usage Notes

9.8.1 Constraints on Clock Generation Circuit

The frequencies of the system clock (ICLK), peripheral module clock (PCLKA to PCLKD), flash interface clock (FCLK), external bus clock (BCLK), and SDRAM clock (SDCLK) supplied to each module change according to the settings in SCKDIVCR. Each frequency must meet the following conditions:

- Each frequency must be selected within the operation-guaranteed range of the clock cycle time (tcyc) specified in the AC electrical characteristics. See section 60, Electrical Characteristics.
- The frequencies must not exceed the ranges listed in Table 9.2.
- The peripheral modules operate on PCLKB and PCLKA. As a result, the operating speed of modules such as the timer and SCI is different before and after the frequency is changed.
- The system clock (ICLK), peripheral module clock (PCLKA to PCLKD), flash interface clock (FCLK), and external bus clock (BCLK) must be set as shown in Table 9.2.

Do not change the clock frequency during external bus access. Additionally, when external bus access starts after a change to the clock frequency, always confirm that the frequency changes are complete before starting the access. To ensure correct processing after the clock frequency changes, first write to the relevant clock control register to change the frequency, then read the value from the register, and finally perform the subsequent processing.

9.8.2 Constraints on the Resonator

Because the resonator characteristics relate closely to your board design, adequate evaluation is required before use. See the resonator connection example in Figure 9.9. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Always consult the resonator manufacturer when determining the circuit constants. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

9.8.3 Constraints on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Route other signal lines away from the oscillation circuit, as shown in Figure 9.15, to prevent electromagnetic induction from interfering with correct oscillation.

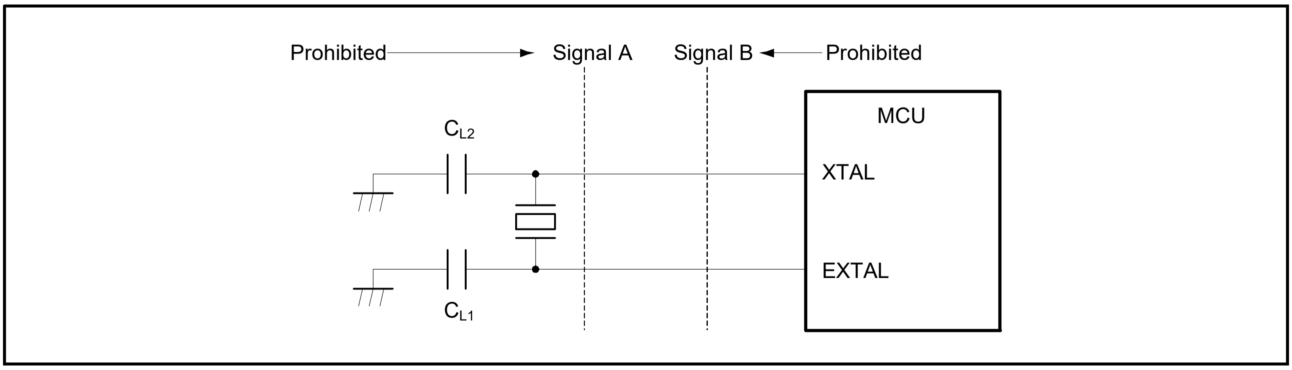


Figure 9.15 **Notes on board design for oscillation circuit (applies to the sub-clock oscillator for the main clock oscillator)**

9.8.4 Constraints on the Resonator Connect Pin

When the main clock oscillator is not used, the EXTAL and XTAL pins can be used as general ports P212 and P213. When these pins are used as general ports, the main clock oscillator must be stopped (MOSCCR.MOSTP must be set to 1).

9.8.5 Constraints on Using Sub-Clock Oscillator for BGA and LGA Packages

The output of the P212 (EXTAL) and P213 (XTAL) pins may affect the oscillation by the sub-clock oscillator.

If the sub-clock oscillator is used, implement the board design so as not to affect to the oscillation. Renesas strongly recommends setting the DSCR[1:0] bits to 00b or 01b when using P212 (EXTAL) and P213 (XTAL) as output pins and using the sub-clock oscillator.

In addition, when using the sub-clock oscillator in middle drive capability (SOMCR.SODRV1 = 1), Renesas recommends not to use P212 (EXTAL) and P213 (XTAL) simultaneously as output pins to avoid affecting the oscillation.

9.8.6 Constraints on the Main Clock Oscillator Drive Capability Auto Switching Function

The drive capability auto switching function lowers the main clock oscillator drive capability automatically after the main clock oscillator starts and suppresses the EMI associated with the main clock oscillator.

To enable drive capability auto switching, set MOMCR.AUTODRVEN to 1 while the main clock oscillator is stopped (MOSCCR.MOSTP = 1). Regardless of the MOMCR.AUTODRVEN setting, the drive capability switching register (MOMCR.MODRV0[1:0]) must be set properly according to the selected oscillator.

Then, enable the main clock oscillator (MOSCCR.MOSTP = 0). After the oscillation stabilization flag (OSCSF.MOSCSF) becomes 1, the main clock can be used.

EMI suppression is gained in return for an extension in the oscillation stabilization wait time. For more information, see section 9.2.18, Main Clock Oscillator Wait Control Register (MOSCWTCR).

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

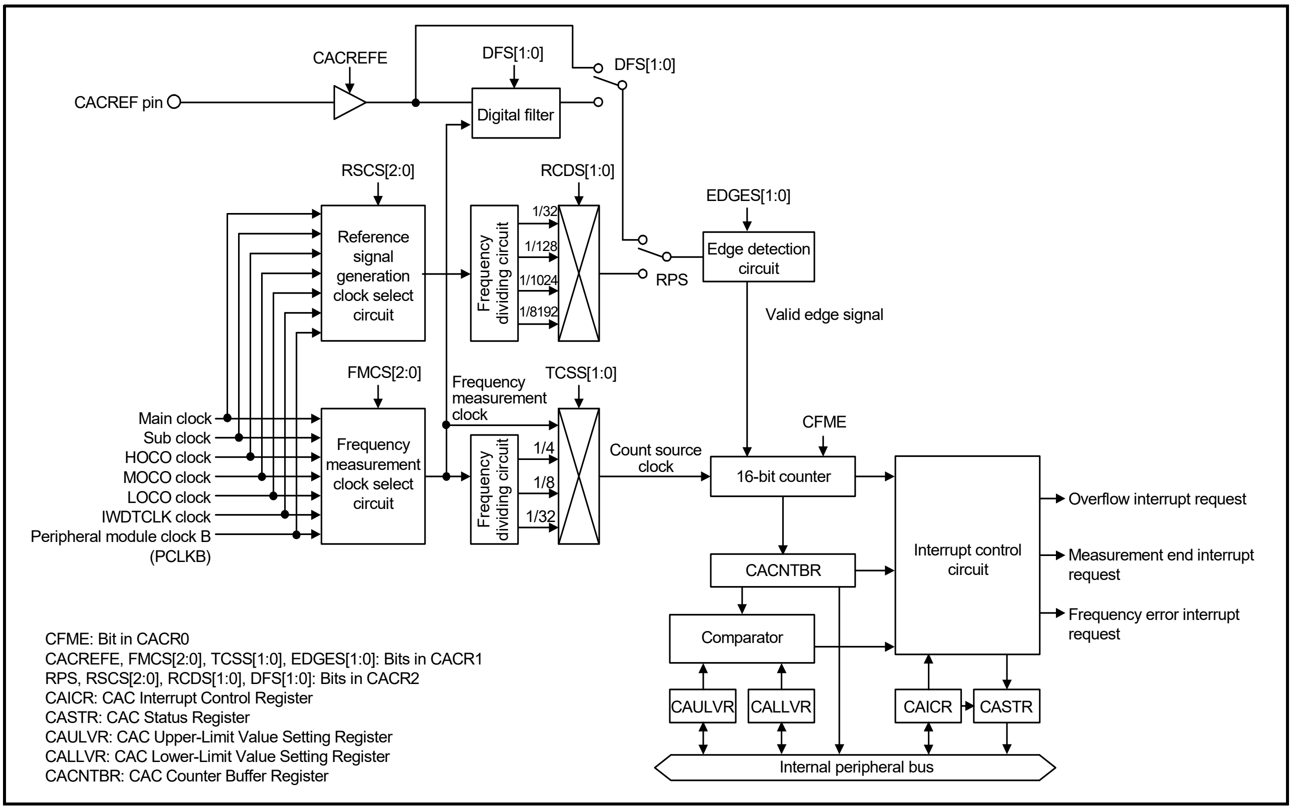
The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.

The reference clock can be provided externally through an I/O port pin or internally from various on-chip oscillators. Interrupt signals can be generated when the clock does not match or measurement ends. This feature is useful in implementing a fail-safe mechanism for home and industrial automation applications.

Table 10.1 lists the CAC specifications, Figure 10.1 shows a block diagram, and Table 10.2 describes the I/O pin.

Table 10.1 CAC specifications

Parameter	Specifications
Measurement target clocks	Frequency can be measured for: • Main clock oscillator • Sub-clock oscillator • HOCO clock • MOCO clock • LOCO clock • IWDTCLK clock • Peripheral module clock B (PCLKB)
Measurement reference clocks	Frequency can be referenced to: • External clock input to the CACREF pin • Main clock oscillator • Sub-clock oscillator • HOCO clock • MOCO clock • LOCO clock • IWDTCLK clock • Peripheral module clock B (PCLKB)
Selectable function	Digital filter
Interrupt sources	• Measurement end • Frequency error • Overflow
Module-stop function	Module-stop state can be set to reduce power consumption



CFME: Bit in CACR0
CACREFE, FMCS[2:0], TCSS[1:0], EDGES[1:0]: Bits in CACR1
RPS, RSCS[2:0], RCDS[1:0], DFS[1:0]: Bits in CACR2
CAICR: CAC Interrupt Control Register
CASTR: CAC Status Register
CAULVR: CAC Upper-Limit Value Setting Register
CALLVR: CAC Lower-Limit Value Setting Register
CACNTBR: CAC Counter Buffer Register

Figure 10.1 CAC block diagram

Table 10.2 CAC I/O pin

Pin name	I/O	Function
CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): CAC.CACR0 4004 4600h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Disable 1: Enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CFME bit (Clock Frequency Measurement Enable)

The CFME bit enables clock frequency measurement. Read the CFME bit to confirm that the bit value has changed. Additional write accesses are ignored before the change is complete.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): CAC.CACR1 4004 4601h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: Disable 1: Enable.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock oscillator 0 0 1: Sub-clock oscillator 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock (PCLKB) 1 1 0: IWDTCLK clock 1 1 1: Setting prohibited.	R/W
b5, b4	TCSS[1:0]	Measurement Target Clock Frequency Division Ratio Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock.	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited.	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE bit (CACREF Pin Input Enable)

The CACREFE bit enables the CACREF pin input.

FMCS[2:0] bits (Measurement Target Clock Select)

The FMCS[2:0] bits select the clock for which the frequency is to be measured.

TCSS[1:0] bits (Measurement Target Clock Frequency Division Ratio Select)

The TCSS[1:0] bits select the division ratio of the measurement target clock.

EDGES[1:0] bits (Valid Edge Select)

The EDGES[1:0] bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): CAC.CACR2 4004 4602h

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal).	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock oscillator 0 0 1: Sub-clock oscillator 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock (PCLKB) 1 1 0: IWDTCLK clock 1 1 1: Setting prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ratio Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock.	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Disable digital filtering 0 1: Use sampling clock for the digital filter as the frequency measuring clock 1 0: Use sampling clock for the digital filter as the frequency measuring clock divided by 4 1 1: Use sampling clock for the digital filter as the frequency measuring clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS bit (Reference Signal Select)

The RPS bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0] bits (Measurement Reference Clock Select)

The RSCS[2:0] bits select the clock source for generating the measurement reference clock.

RCDS[1:0] bits (Measurement Reference Clock Frequency Division Ratio Select)

The RCDS[1:0] bits select the frequency division ratio of the measurement reference clock, when an internal reference clock is selected (RPS = 1). When RPS = 0 (CACREF pin is used as the reference clock source), the reference clock is not divided.

DFS[1:0] bits (Digital Filter Select)

The setting of the DFS[1:0] bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Control Register (CAICR)

Address(es): CAC.CAICR 4004 4603h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Disable 1: Enable.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Disable 1: Enable.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Disable 1: Enable.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE bit (Frequency Error Interrupt Request Enable)

The FERRIE bit enables the frequency error interrupt request.

MENDIE bit (Measurement End Interrupt Request Enable)

The MENDIE bit enables the measurement end interrupt request.

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit enables the overflow interrupt request.

FERRFCL bit (FERRF Clear)

Writing 1 to the FERRFCL bit to 1 clears the CASTR.FERRF flag.

MENDFCL bit (MENDF Clear)

Writing 1 to the MENDFCL bit to 1 clears the CASTR.MENDF flag.

OVFFCL bit (OVFF Clear)

Writing 1 to the OVFFCL bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): CAC.CASTR 4004 4604h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FERRF	Frequency Error Flag	0: Clock frequency is within the allowable range 1: Clock frequency has deviated beyond the allowable range (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress 1: Measurement ended.	R
b2	OVFF	Overflow Flag	0: Counter has not overflowed 1: Counter overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

FERRF flag (Frequency Error Flag)

The FERRF flag indicates a deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside the allowable range defined in the CAULVR and CALLVR registers.

[Clearing condition]

- 1 is written to the FERRFCL bit.

MENDF flag (Measurement End Flag)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement completes.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflows.

[Clearing condition]

- 1 is written to the OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC.CAULVR 4004 4606h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAULVR is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value exceeds the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0.

The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): CAC.CALLVR 4004 4608h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CALLVR is a 16-bit read/write register that specifies the lower value of the allowable range. When the counter value falls below the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0.

The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): CAC.CACNTBR 4004 460Ah

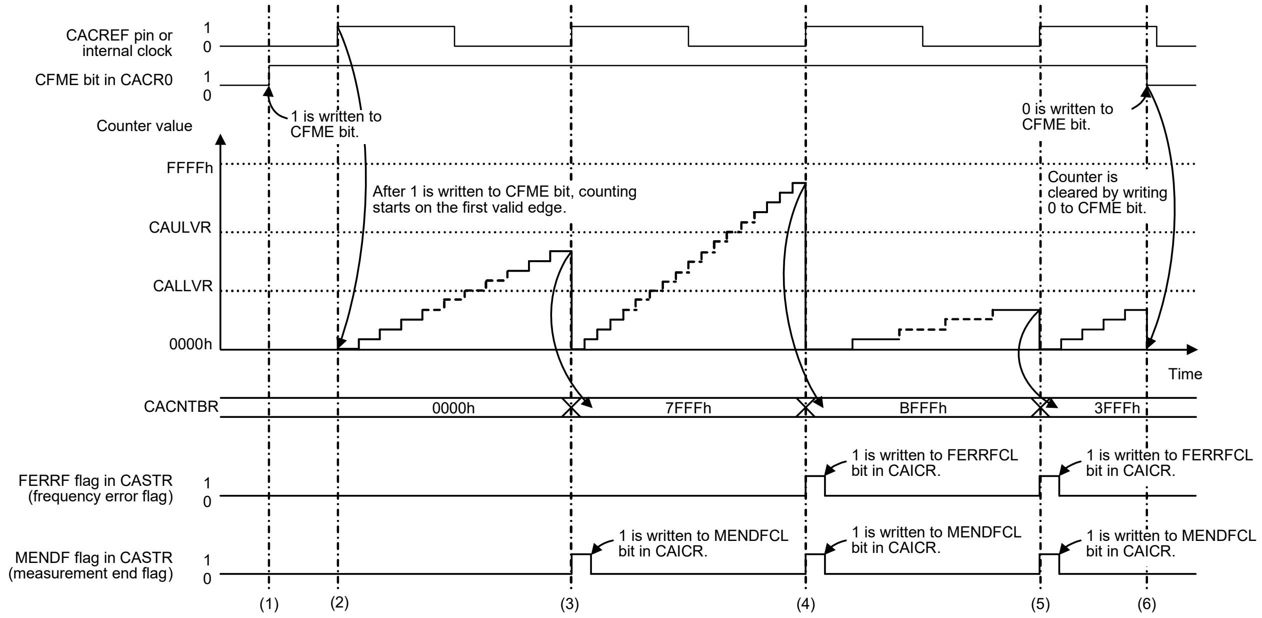
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CACNTBR is a 16-bit read-only register that stores the measurement result.

10.3 Operation

10.3.1 Measuring Clock Frequency

The CAC measures the clock frequency using the CACREF pin input or an internal clock as a reference. Figure 10.2 shows an operating example of the CAC.



When the CACREF pin input is used as a reference:
In CACR1: CACREFE bit = 1, EDGES[1:0] bits = 00b
CAULVR register = AAAAh, CALLVR register = 5555h
When the internal clock is used as a reference:
In CACR1: CACREFE bit = 0, EDGES[1:0] bits = 00b
CAULVR register = AAAAh, CALLVR register = 5555h

Figure 10.2 CAC operating example

The events in Figure 10.2 are:

1. Before writing 1 to CACR0.CFME, set CACR1 and CACR2 to define the measurement target clock and measurement reference clock. Writing 1 to the CACR0.CFME bit enables clock frequency measurement.
2. The timer starts counting up if the valid edge selected in the CACR1.EDGES[1:0] bits is input from the measurement reference clock. In Figure 10.2, the valid edge is a rising edge (CACR1.EDGES[1:0] = 00b).
3. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are true, only the MENDF flag in CASTR is set to 1, because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
5. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
6. When the CFME bit in CACR0 is 1, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on CACREF Pin

The CACREF pin has a digital filter, and levels on the CACREF pin are transmitted to the internal circuitry after three consecutive matches in the selected sampling interval. The same level continues to be transmitted internally until the level on the pin has three consecutive matches again. Enabling or disabling of the digital filter and its sampling clock are selectable.

The counter value transferred to CACNTBR might be in error by up to one cycle of the sampling clock because of the difference between the phases of the digital filter and the signal input to the CACREF pin. When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt.

When an interrupt source is generated, the associated status flag is set to 1. Table 10.3 provides information on the CAC interrupt requests.

Table 10.3 CAC interrupt requests

Interrupt request	Interrupt enable bit	Status flag	Interrupt sources
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	Result of comparing CACNTBR with CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	• Valid edge is input from the CACREF pin or internal clock • Measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	Counter overflows

10.5 Usage Notes

10.5.1 Settings for the Module-Stop Function

CAC operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CAC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

11. Low Power Modes

11.1 Overview

The MCU provides several functions for reducing power consumption, such as setting clock dividers, controlling EBCLK output, controlling SDCLK output, stopping modules, selecting power control mode in normal mode, and transitioning to low power modes.

Table 11.1 lists the specifications of the low power mode functions. Table 11.2 list the conditions to transition to low power modes, the states of the CPU and peripheral modules, and the method for canceling each mode. After a reset, the MCU enters the program execution state, but only the DMAC, DTC, and SRAM operate.

Table 11.1 Specifications of the low power mode functions

Parameter	Specifications
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD), external bus clock (BCLK), and flash interface clock (FCLK)*1
EBCLK output control	Selectable to BCLK output or high-level output
SDCLK output control	Selectable to SDCLK output or high-level output
Module-stop state	Peripheral module functions can be stopped independently
Low power modes	• Sleep mode • Software Standby mode • Snooze mode • Deep Software Standby mode.
Power control modes	Power consumption can be reduced in Normal, Sleep, and Snooze modes by selecting an appropriate operating power control mode according to the operating frequency and voltage. Three operating power control modes are available: • High-speed mode • Low-speed mode • Subosc-speed mode.

Note 1. For details, refer to section 9, Clock Generation Circuit.

Table 11.2 Operating conditions of each low power mode (1 of 3)

Parameter	Sleep mode	Software Standby mode	Snooze mode*1	Deep Software Standby mode
Transition condition	WFI instruction while SBYCR.SSBY = 0	WFI instruction while SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 0	Snooze request trigger in Software Standby mode. SNZCR.SNZE = 1	WFI instruction while SBYCR.SSBY = 1 and DPSBYCR.DPSBY = 1
Canceling method	All interrupts. Any reset available in the mode.	Interrupts shown in Table 11.3. Any reset available in the mode.	Interrupts shown in Table 11.3. Any reset available in the mode.	Interrupts shown in Table 11.3. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Reset state
State after cancellation by a reset	Reset state	Reset state	Reset state	Reset state
Main clock oscillator	Selectable	Stop	Selectable*2	Stop
Sub-clock oscillator	Selectable	Selectable	Selectable	Selectable
High-speed on-chip oscillator	Selectable	Stop	Selectable	Stop
Middle-speed on-chip oscillator	Selectable	Stop	Selectable	Stop
Low-speed on-chip oscillator	Selectable	Selectable	Selectable	Selectable*3

Table 11.2 Operating conditions of each low power mode (2 of 3)

Parameter	Sleep mode	Software Standby mode	Snooze mode*1	Deep Software Standby mode
IWDT-dedicated on-chip oscillator	Selectable*7	Selectable*7	Selectable*7	Stop
PLL	Selectable	Stop	Selectable*2	Stop
Oscillation stop detection function	Selectable	Operation prohibited	Operation prohibited	Operation prohibited
Clock/buzzer output function	Selectable	Selectable*4	Selectable	Stop (Undefined)
External bus (EBCLK)	Selectable	Stop (Retained)	Operation prohibited	Stop (Retained)
CPU	Stop (Retained)	Stop (Retained)	Stop (Retained)	Stop (Undefined)
SRAMn (n = 0, 1), SRAMHS, ECC SRAM	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Standby SRAM	Selectable	Stop (Retained)	Selectable	Stop (Retained/Undefined)*5
Flash memory	Operating	Stop (Retained)	Stop (Retained)	Stop (Retained)
DMA Controller (DMAC)	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
USB 2.0 Full-Speed Module (USBFS)	Selectable	Stop (Retained). Detection of USB resumption is possible.	Operation prohibited. Detection of USB resumption is possible.	Stop (Retained/Undefined) Detection of USB resumption is possible.*6
USB 2.0 High-Speed Module (USBHS)	Selectable	Stop (Retained). Detection of USB resumption is possible.	Operation prohibited. Detection of USB resumption is possible.	Stop (Retained/Undefined) Detection of USB resumption is possible.*6
Watchdog Timer (WDT)	Selectable*7	Stop (Retained)	Stop (Retained)	Stop (Undefined)
Independent Watchdog Timer (IWDT)	Selectable*7	Selectable*7	Selectable *7	Stop (Undefined)
Realtime clock (RTC)	Selectable	Selectable	Selectable	Selectable*8
Asynchronous General Purpose Timer (AGTn, n = 0, 1)	Selectable	Selectable*9	Selectable*9	Selectable*9
12-Bit A/D Converter (ADC12)	Selectable	Stop (Retained)	Selectable*19	Stop (Undefined)
Programmable Gain Amplifiers (PGAs)	Selectable*13	Selectable*13	Selectable*13	Stop (Undefined)
12-Bit D/A Converter (DAC12)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Capacitive Touch Sensing Unit (CTSU)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Data Operation Circuit (DOC)	Selectable	Stop (Retained)	Selectable	Stop (Undefined)
Serial Communications Interface (SCI0)	Selectable	Stop (Retained)	Selectable (RXD0 falling edge is available, to enter Snooze mode) (only in asynchronous mode).*15	Stop (Undefined)
Serial Communications Interface (SCIn, n = 1 to 9)	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
I^{2}C Bus Interface (IIC0)	Selectable	Selectable*14	Selectable*14	Stop (Undefined)
I^{2}C Bus Interface (IICn, n = 1, 2)	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)

Table 11.2 Operating conditions of each low power mode (3 of 3)

Parameter	Sleep mode	Software Standby mode	Snooze mode*1	Deep Software Standby mode
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*10	Stop (Undefined)
High-Speed Analog Comparator (ACMPHS0)	Selectable	Selectable*12	Selectable. VCOUT function only.*12	Stop (Undefined)
High-Speed Analog Comparator (ACMPHSn, n = 1 to 5)	Selectable	Selectable*11	Selectable. VCOUT function only.*11	Stop (Undefined)
IRQn (n = 0 to 15) pin interrupt	Selectable	Selectable	Selectable	Stop (Undefined)
NMI, IRQn-DS (n = 0 to 14) pin interrupt	Selectable	Selectable	Selectable	Selectable
Key Interrupt Function (KINT)	Selectable	Selectable	Selectable	Stop (Undefined)
Low Voltage Detection (LVD)	Selectable	Selectable	Selectable	Selectable*16
Power-on reset circuit	Operating	Operating	Operating	Operating*17
Other peripheral modules	Selectable	Stop (Retained)	Operation prohibited	Stop (Undefined)
I/O ports	Operating	Retained*18	Operating	Retained*18

Note: Selectable means that operating or not operating can be selected by the control registers.
Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Operation prohibited means that the function must be stopped before entering Software Standby mode.
Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.

Note 1. All modules whose module-stop bits are 0 start as soon as PCLKs are supplied after entering Snooze mode. To avoid an increase in ICC in Snooze mode, set the module-stop bit of modules that are not required in Snooze mode to 1 before entering Software Standby mode.

Note 2. When using SCI0 in Snooze mode, the MOSCCR.MOSTP and PLLCR.PLLSTP bits must be 1.

Note 3. If the DPSBYCR.DEEPCUT[1:0] bits are 00b, the oscillator status is the same as before entering Deep Software Standby mode. When the DPSBYCR.DEEPCUT[1:0] bits are not 00b, the oscillator stops when the MCU enters Deep Software Standby mode.

Note 4. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO) and 100b (SOSC).

Note 5. If the DPSBYCR.DEEPCUT[1:0] bits are 00b, data in the Standby SRAM is retained in Deep Software Standby mode. When the DPSBYCR.DEEPCUT[1:0] bits are not 00b, data in the Standby SRAM is retained is undefined in Deep Software Standby mode.

Note 6. If the DPSBYCR.DEEPCUT[1:0] bits are 00b, the values of the USB resume detection circuit registers are retained and detection of USB resumption is enabled, and the values of other registers are undefined in Deep Software Standby mode. When the DPSBYCR.DEEPCUT[1:0] bits are not 00b, the values of all registers are undefined in Deep Software Standby mode.

Note 7. In IWDT-dedicated on-chip oscillator and IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode. In WDT, operating or stopping is selected by setting the WDT Stop Control bit (WDTSTPCTL) in Option Function Select register 0 (OFS0) in WDT auto start mode.

Note 8. When the RCR4.RCKSEL bit set to 1 (LOCO), the DPSBYCR.DEEPCUT[1:0] bits must set to 00b before entering Deep Software Standby mode.

Note 9. AGT0 operation is possible when 100b (AGTLCLK) or 110b (AGTSCLK) is selected in the AGT0.AGTMR1.TCK[2:0] bits.
AGT1 operation is possible when 100b (AGTLCLK), 110b (AGTSCLK) or 101 (underflow event signal from AGT0) is selected in the AGT1.AGTMR1.TCK[2:0] bits.
When 100b (AGTLCLK) is selected in AGTn.AGTMR1.TCK[2:0] bits (n = 0, 1), the DPSBYCR.DEEPCUT[1:0] bits must set to 00b before entering Deep Software Standby mode.

Note 10. Event lists the restrictions described in section 11.10.13, ELC Events in Snooze Mode.

Note 11. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS uses no digital filter.
For details on digital filter, see section 50, High-Speed Analog Comparator (ACMPHS).

Note 12. When CMPCTL0.CSTEN bit is 1, canceling Software Standby Mode or entering Snooze mode by the comparator detection is available.

Note 13. When using the Programmable Gain Amplifiers, MSTPDn (n = 15, 16) must be set to 0. For details, see section 47.3.12, Programmable Gain Amplifiers.

Note 14. IIC0 wakeup interrupt is available.

Note 15. Serial communication mode of SCI0 is asynchronous mode.

Note 16. When using LVD in Deep Software Standby mode, DPSBYCR.DEEPCUT[1:0] bits must be 00b or 01b before entering Deep

Software Standby mode.

Note 17. When the MCU enters Deep Software Standby mode with the DPSBYCR.DEEPCUT[1:0] bits set to 11b, the LVD circuit stops and the low-power function of the power-on reset circuit is enabled.

Note 18. For the address bus and bus control signals (For SRAM: [CS0 to CS7, RD, WR0 to WR1, WR, BC0 to BC1, and ALE], and for SDRAM: [SDCS, RAS, CAS and WE]), keeping the output state or changing to the high-impedance state can be selected in the SBYCR.OPE bit.

Note 19. When using the 12-Bit A/D Converter in Snooze mode, the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits must be 1.

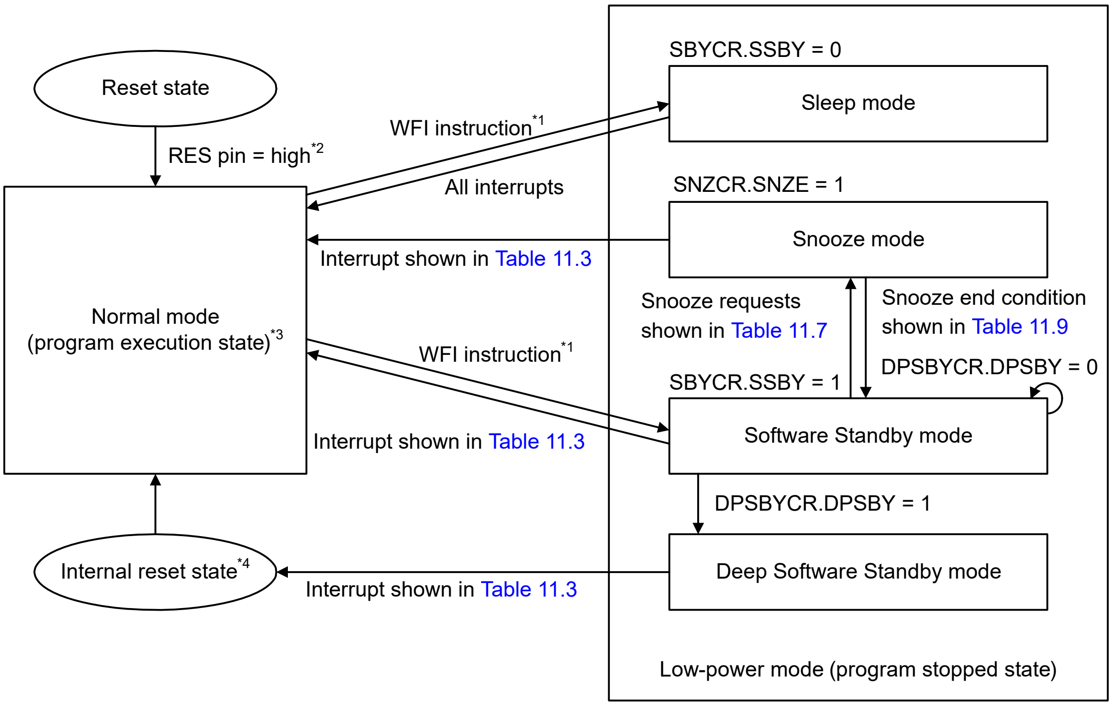
Table 11.3 Interrupt sources for canceling Snooze, Software Standby, and Deep Software Standby modes

Interrupt source	Name	Software Standby mode	Snooze mode	Deep Software Standby mode
NMI		Yes	Yes	Yes
Port	PORT_IRQn (n = 0 to 15)	Yes	Yes	No
	PORT_IRQn-DS (n = 0 to 14)	Yes	Yes	Yes
LVD	LVD_LVD1	Yes	Yes	Yes
	LVD_LVD2	Yes	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes	No
USBFS	USBFS_USBR	Yes	Yes	Yes
USBHS	USBHS_USBIR	Yes	Yes	Yes
RTC	RTC_ALM	Yes	Yes	Yes
	RTC_PRD	Yes	Yes	Yes
KINT	KEY_INTKR	Yes	Yes	No
AGT1	AGT1_AGTI	Yes	Yes*3	Yes
	AGT1_AGTCMAI	Yes	Yes	No
	AGT1_AGTCMBI	Yes	Yes	No
ACMPHS	ACMP_HS0	Yes	Yes	No
IIC0	IIC0_WUI	Yes	Yes	No
ADC12n (n = 0, 1)	ADC12n_WCMPM	No	Yes with SELSR0*1,*3	No
	ADC12n_WCMPUM	No	Yes with SELSR0*1,*3	No
SCI0	SCI0_AM	No	Yes with SELSR0*1,*2	No
	SCI0_RXI_OR_ERI	No	Yes with SELSR0*1,*2	No
DTC	DTC_COMPLETE	No	Yes with SELSR0*1,*3	No
DOC	DOC_DOPCI	No	Yes with SELSR0*1	No
CTSU	CTSU_CTSUFN	No	Yes with SELSR0*1	No

Note 1. To use the interrupt request as a trigger for exiting Snooze mode, the request must be selected in SELSR0. See section 14, Interrupt Controller Unit (ICU). When a trigger selected in SELSR0 occurs after executing a WFI instruction and during the transition from Normal to Software Standby mode, the request might or might not be accepted, depending on the timing of the occurrence.

Note 2. Only one of either SCI0_AM or SCI0_RXI_OR_ERI can be set.

Note 3. The event that is enabled by the SNZEDCR register must not be used.



Note 1. When an interrupt that acts as a trigger for cancel is received during a transition to the program stopped state after the execution of a WFI instruction, the MCU executes interrupt exception handling instead of transitioning to low power mode.

Note 2. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.

Note 3. The transition to Normal mode is made because of an interrupt from Sleep, Snooze, or Software Standby mode. The clock source is the same as before entering the low power mode.

Note 4. When an available interrupt request is generated, an internal reset (Deep Software Standby reset) is generated over a fixed period. Canceling of Deep Software Standby mode accompanies release from the internal reset state, and then the MCU transitions to Normal mode and execute a reset exception processing with the MOCO clock as the source of the operating clock.

Figure 11.1 Mode transitions

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): SYSTEM.SBYCR 4001 E00Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSBY	OPE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	OPE	Output Port Enable	0: In Software Standby or Deep Software Standby mode, set the address bus and bus control signals to the high-impedance state. In Snooze mode, the status of the address bus and bus control signals are the same as before entering Software Standby mode. 1: In Software Standby or Deep Software Standby mode, retain the output state of the address bus and bus control signals.	R/W
b15	SSBY	Software Standby	0: Sleep mode 1: Software Standby mode when DPSBYCR.DPSBY = 0 and Deep Software Standby mode when DPSBYCR.DPSBY = 1.	R/W

OPE bit (Output Port Enable)

The OPE bit specifies whether to set to the high-impedance state or to retain the output of the address bus and bus control signals (for SRAM: CS0 to CS7, RD, WR0 to WR1, WR, BC0 to BC1, and ALE, and for SDRAM: SDCS, RAS, CAS, and WE) in Software Standby mode or Deep Software Standby mode.

SSBY bit (Software Standby)

The SSBY bit specifies the target transition after a WFI instruction is executed. When the SSBY bit is set to 1, the MCU enters Software Standby mode after executing the WFI instruction. When the MCU cancels Software Standby mode by an interrupt, the SSBY bit remains set to 1. The SSBY bit can be cleared by writing 0 to it.

When the OSTDCR.OSTDE bit is 1, the SSBY bit is ignored. Even if the SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction.

When the FENTRYR.FENTRYi bit (i = 0 to 3) is 1 or the FENTRYR.FENTRYD bit is 1, the SSBY is ignored. Even if the SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction. See Table 11.6 for details.

When using the HOCO clock to enter Software Standby mode, STCONR.STCON[1:0] must be set to 00b and HOCOWTCR.HSTS[2:0] must be set to 110b. However, when using SCI0 in Snooze mode, HOCOWTCR.HSTS[2:0] must be set to 010b.

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): SYSTEM.MSTPCRA 4001 E01Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	MSTPA 22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	MSTPA 7	MSTPA 6	MSTPA 5	—	—	—	MSTPA 1	MSTPA 0
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MSTPA0	SRAM0 Module Stop*1	Target module: SRAM0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b1	MSTPA1	SRAM1 Module Stop	Target module: SRAM1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b4 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	MSTPA5	High-Speed SRAM Module Stop	Target module: high-speed SRAM 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6	MSTPA6	ECC SRAM Module Stop*1	Target module: ECC SRAM 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b7	MSTPA7	Standby SRAM Module Stop	Target module: Standby SRAM 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b21 to b8	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b22	MSTPA22	DMA Controller/Data Transfer Controller Module Stop*2	Target modules: DMAC, DTC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b31 to b23	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPA0 and MSTPA6 bit settings must be the same.
Note 2. When rewriting the MSTPA22 bit from 0 to 1, disable the DMAC and DTC before setting the MSTPA22 bit.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): MSTP.MSTPCRB 4004 7000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPB 31	MSTPB 30	MSTPB 29	MSTPB 28	MSTPB 27	MSTPB 26	MSTPB 25	MSTPB 24	MSTPB 23	MSTPB 22	—	—	MSTPB 19	MSTPB 18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPB 15	—	MSTPB 13	MSTPB 12	MSTPB 11	—	MSTPB 9	MSTPB 8	MSTPB 7	MSTPB 6	MSTPB 5	—	—	MSTPB 2	MSTPB 1	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b1	MSTPB1	Controller Area Network 1 Module Stop*1	Target module: CAN1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b2	MSTPB2	Controller Area Network 0 Module Stop*1	Target module: CAN0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b4, b3	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	MSTPB5	IrDA Module Stop	Target module: IrDA 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6	MSTPB6	Quad Serial Peripheral Interface Module Stop	Target module: QSPI 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b7	MSTPB7	I^{2}C Bus Interface 2 Module Stop	Target module: IIC2 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b8	MSTPB8	I^{2}C Bus Interface 1 Module Stop	Target module: IIC1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b9	MSTPB9	I^{2}C Bus Interface 0 Module Stop	Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11	MSTPB11	Universal Serial Bus 2.0 FS Interface Module Stop*2	Target module: USBFS 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b12	MSTPB12	Universal Serial Bus 2.0 HS Interface Module Stop	Target module: USBHS 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b13	MSTPB13	EPTPC and PTPEDMAC Module Stop*3	Target modules: EPTPC and PTPEDMAC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b14	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15	MSTPB15	ETHERC0 and EDMAC0 Controller Module Stop	Target modules: ETHERC0, EDMAC0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b17, b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b18	MSTPB18	Serial Peripheral Interface 1 Module Stop	Target module: SPI1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W

Bit	Symbol	Bit name	Description	R/W
b19	MSTPB19	Serial Peripheral Interface 0 Module Stop	Target module: SPI0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b21, b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b22	MSTPB22	Serial Communication Interface 9 Module Stop	Target module: SCI9 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b23	MSTPB23	Serial Communication Interface 8 Module Stop	Target module: SCI8 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b24	MSTPB24	Serial Communication Interface 7 Module Stop	Target module: SCI7 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop	Target module: SCI4 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W

Note 1. The MSTPBi bit must be written to while the oscillation of the clock controlled by this bit is stable. To enter Software Standby mode after writing to this bit, wait for two CAN clock (CANMCLK) cycles after writing, then execute a WFI instruction (i = 1, 2).

Note 2. To enter Software Standby mode after writing to the MSTPB11 bit, wait for two USB clock (UCLK) cycles after writing, then execute a WFI instruction.

Note 3. Even when EPTPC and PTPEDMAC operation is enabled (MSTPB13 = 0), some registers in the EPTPC module become inaccessible depending on the combination of the MSTPB15 bit and EPTPC bypass bit (EPTPC_CFG.BYPASS.BYPASS0) settings. For details, see section 30, Ethernet PTP Controller (EPTPC).

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): MSTP.MSTPCRC 4004 7004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPC 31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MSTPC 14	MSTPC 13	MSTPC 12	MSTPC 11	—	MSTPC 9	MSTPC 8	MSTPC 7	MSTPC 6	MSTPC 5	MSTPC 4	MSTPC 3	MSTPC 2	MSTPC 1	MSTPC 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	MSTPC0*1	Clock Frequency Accuracy Measurement Circuit Module Stop	Target module: CAC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b1	MSTPC1	Cyclic Redundancy Check Calculator Module Stop	Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b2	MSTPC2	Parallel Data Capture Module Stop	Target module: PDC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b3	MSTPC3	Capacitive Touch Sensing Unit Module Stop	Target module: CTSU 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b4	MSTPC4	Graphics LCD Controller Module Stop	Target module: GLCDC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b5	MSTPC5	JPEG Codec Engine Module Stop	Target module: JPEG 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6	MSTPC6	2D Drawing Engine Module Stop	Target module: DRW 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b7	MSTPC7	Serial Sound Interface Enhanced (channel 1) Module Stop	Target module: SSIE1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b8	MSTPC8	Serial Sound Interface Enhanced (channel 0) Module Stop	Target module: SSIE0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b9	MSTPC9	Sampling Rate Converter Module Stop	Target module: SRC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11	MSTPC11	Secure Digital Host IF/ MultiMediaCard 1 Module Stop	Target module: SDHI/MMC1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b12	MSTPC12	Secure Digital Host IF/ MultiMediaCard 0 Module Stop	Target module: SDHI/MMC0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b13	MSTPC13	Data Operation Circuit Module Stop	Target module: DOC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b14	MSTPC14	Event Link Controller Module Stop	Target module: ELC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W

Bit	Symbol	Bit name	Description	R/W
b30 to b15	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	MSTPC31	SCE7 Module Stop	Target module: SCE7 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W

Note 1. The MSTPC0 bit must be written to while the oscillation of the clock controlled by this bit is stable. To enter Software Standby mode after writing to this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators, then execute a WFI instruction.

11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): MSTP.MSTPCRD 4004 7008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	MSTPD 28	MSTPD 27	MSTPD 26	MSTPD 25	MSTPD 24	MSTPD 23	MSTPD 22	—	MSTPD 20	—	—	—	MSTPD 16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPD 15	MSTPD 14	—	—	—	—	—	—	—	MSTPD 6	MSTPD 5	—	MSTPD 3	MSTPD 2	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b2	MSTPD2	Asynchronous General Purpose Timer 1 Module Stop*1	Target module: AGT1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b3	MSTPD3	Asynchronous General Purpose Timer 0 Module Stop*2	Target module: AGT0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b5	MSTPD5	General PWM Timer 32EH0 to 32EH3 and 32E4 to 32E7 and PWM Delay Generation Circuit Module Stop	Target modules: GPT32EHx (x = 0 to 3), GPT32Ey (y = 4 to 7), and PWM Delay Generation Circuit 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6	MSTPD6	General PWM Timer 328 to 3213 Module Stop	Target modules: GPT32x (x = 8 to 13) 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b13 to b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b14	MSTPD14	Port Output Enable for GPT Module Stop	Target module: POEG 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b15	MSTPD15	12-Bit A/D Converter 1 Module Stop	Target module: ADC121 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b16	MSTPD16	12-Bit A/D Converter 0 Module Stop	Target module: ADC120 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b19 to b17	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b20	MSTPD20	12-Bit D/A Converter Module Stop	Target module: DAC12 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b21	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit name	Description	R/W
b22	MSTPD22	Temperature Sensor Module Stop	Target module: TSN 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b23	MSTPD23	High-Speed Analog Comparator 5 Module Stop	Target module: ACMPHS5 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b24	MSTPD24	High-Speed Analog Comparator 4 Module Stop	Target module: ACMPHS4 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b25	MSTPD25	High-Speed Analog Comparator 3 Module Stop	Target module: ACMPHS3 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b26	MSTPD26	High-Speed Analog Comparator 2 Module Stop	Target module: ACMPHS2 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b27	MSTPD27	High-Speed Analog Comparator 1 Module Stop	Target module: ACMPHS1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b28	MSTPD28	High-Speed Analog Comparator 0 Module Stop	Target module: ACMPHS0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b31 to b29	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. When the count source is sub-clock oscillator or LOCO, AGT1 counting does not stop even if MSTPD2 is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGT1 registers.

Note 2. When the count source is sub-clock oscillator or LOCO, AGT0 counting does not stop even if MSTPD3 is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGT0 registers.

11.2.6 Operating Power Control Register (OPCCR)

Address(es): SYSTEM.OPCCR 4001 E0A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	—	OPCM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	OPCM[1:0]	Operating Power Control Mode Select	b1 b0 0 0: High-speed mode 1 1: Low-speed mode. Other settings are prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	OPCMTSF	Operating Power Control Mode Transition Status Flag	• Read 0: Transition complete 1: Transition in progress. • Write The write value should be 0.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The OPCCR register is used to reduce power consumption in Normal and Sleep modes by specifying a lower operating frequency and operating voltage. For the procedure for changing the operating power control modes, see section 11.5, Function for Lower Operating Power Consumption.

When transitioning from Software Standby mode to Normal or Snooze mode, the settings in the OPCCR.OPCM[1:0] and SOPCCR.SOPCM bits are as follows, regardless of their settings before entering Software Standby mode:

- OPCCR.OPCM[1:0] = 00b (High-speed mode)

- SOPCCR.SOPCM = 0b (not Subosc-speed mode).

If Software Standby mode is canceled before the transition to Software Standby completes, the OPCCR.OPCM[1:0] and SOPCCR.SOPCM bits retain their settings from before the WFI instruction executed. If this causes any problem, set the MCU to High-speed mode during the exception handling procedure when canceling Software Standby mode.

OPCM[1:0] bits (Operating Power Control Mode Select)

The OPCM[1:0] bits select the operating power control mode in Normal and Sleep modes. Table 11.4 shows the relationship between the operating power control modes and the OPCM[1:0] and SOPCM settings.

OPCMTSF flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched. The flag is set to 1 on a write access to the OPCM[1:0] bits, and to 0 when the mode transition completes. Confirm that the flag is 0 before proceeding.

11.2.7 Sub Operating Power Control Register (SOPCCR)

Address(es): SYSTEM.SOPCCR 4001 E0AAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SOPC MTSF	—	—	—	SOPC M
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SOPCM	Sub Operating Power Control Mode Select	0: Not Subosc-speed mode 1: Subosc-speed mode.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SOPCMTSF	Sub Operating Power Control Mode Transition Status Flag	0: Transition complete 1: Transition in progress.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SOPCCR register is used to reduce power consumption in Normal and Sleep modes by initiating entry to and exit from Subosc-speed mode. Subosc-speed mode is only available when using the sub-clock oscillator or LOCO without dividing the frequency.

The flash cache function should be set to disabled by setting FCACHEE.FCACHEEN to 0 before switching the operating power control mode. For details, see section 55, Flash Memory.

For the procedure for changing operating power control modes, see section 11.5, Function for Lower Operating Power Consumption.

SOPCM bit (Sub Operating Power Control Mode Select)

The SOPCM bit selects the operating power control mode in Normal and Sleep modes. Setting this bit to 1 allows transition to Subosc-speed mode. Setting this bit to 0 allows a return to the operating mode (set in OPCCR.OPCM[1:0]) that was active prior to the transition to Subosc-speed mode.

When transitioning from Software Standby mode to Normal mode or Snooze mode, the OPCCR.OPCM[1:0] and SOPCCR.SOPCM settings are as follows, regardless of their settings before entering Software Standby mode:

- OPCCR.OPCM[1:0] = 00b (High-speed mode)
- SOPCCR.SOPCM = 0b (not Subosc-speed mode).

If Software Standby mode is canceled before the transition to Software Standby completes, the OPCCR.OPCM[1:0] and SOPCCR.SOPCM bits retain their settings from before the WFI instruction executed. If this causes any problem, set the MCU to High-speed mode during the exception handling procedure when canceling Software Standby mode.

Table 11.4 shows the relationship between the operating power control modes and the OPCM[1:0] and SOPCM settings.

SOPCMTSF flag (Sub Operating Power Control Mode Transition Status Flag)

The SOPCMTSF flag indicates the switching control state when the operating power control mode is switched to Subosc-speed mode or from Subosc-speed mode. The flag is set to 1 on a write access to the SOPCM bit, and to 0 when the mode transition completes. Confirm that the flag is 0 before proceeding.

Table 11.4 Relationship between the operating power control modes and the OPCM[1:0] and SOPCM settings

Operating power control mode	OPCM[1:0] bits	SOPCM bit	Power consumption
High-speed mode	00b	0	High
Low-speed mode	11b	0	↓
Subosc-speed mode	00b, 11b	1	Low

Note: See section 60, Electrical Characteristics, for the operating frequency range and voltage range.

High-speed operating mode

After a reset cancellation, the MCU is activated in this mode.

Low-speed mode

The following constraints apply in Low-speed mode:

- Programming and erasure operations for the flash memory are prohibited
- Using the PLL is prohibited. See section 11.10.1, Register Access.

In this mode, lower power consumption is possible than in High-speed mode when the same operation is performed under the same conditions, such as operating frequency and operating voltage.

Subosc-speed mode

The following constraints apply in Subosc-speed mode:

- Programming and erasure operations for the flash memory are prohibited
- Reading of the data flash is prohibited
- Using MOSC, PLL, MOCO, or HOCO is prohibited. See section 11.10.1, Register Access.
- Using the divided clock for ICK or FCK is prohibited. See section 11.10.1, Register Access.
- Using the oscillation stop detection function of the main clock oscillator is prohibited.

11.2.8 Snooze Control Register (SNZCR)

Address(es): SYSTEM.SNZCR 4001 E092h

	b7	b6	b5	b4	b3	b2	b1	b0
	SNZE	—	—	—	—	—	SNZDTCEN	RXDREQEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RXDREQEN	RXD0 Snooze Request Enable	0: Ignore the RXD0 falling edge in Software Standby mode 1: Detect the RXD0 falling edge in Software Standby mode.	R/W
b1	SNZDTCEN	DTC Enable in Snooze mode	0: Disable DTC operation in Snooze mode 1: Enable DTC operation in Snooze mode.	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SNZE	Snooze Mode Enable	0: Disable Snooze mode 1: Enable Snooze mode.	R/W

RXDREQEN bit (RXD0 Snooze Request Enable)

The RXDREQEN bit specifies whether to detect a falling edge of the RXD0 pin in Software Standby mode. This bit is only available when SCI0 is operating in asynchronous mode. To detect a falling edge of the RXD0 pin, set this bit before entering Software Standby mode. When this bit is set to 1, a falling edge of the RXD0 pin in Software Standby mode causes the MCU to enter Snooze mode.

SNZDTCEN bit (DTC Enable in Snooze mode)

The SNZDTCEN bit specifies whether to use the DTC and SRAM in Snooze mode. To use the DTC and SRAM in Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, the DTC can be activated by setting IELSRn (ICU Event Link setting Register n).

SNZE bit (Snooze Mode Enable)

The SNZE bit enables or disables a transition from Software Standby to Snooze mode. To use Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, one of the event triggers shown in Table 11.7 occurring in Software Standby mode causes the MCU to enter Snooze mode. After the MCU transitions from Software Standby or Snooze mode to Normal mode, clear the SNZE bit once and then set it before re-entering Software Standby mode. For details, see section 11.8, Snooze Mode.

11.2.9 Snooze End Control Register (SNZEDCR)

Address(es): SYSTEM.SNZEDCR 4001 E094h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCI0UMTED	AD1UMTED	AD1MATED	AD0UMTED	AD0MATED	DTCNZRED	DTCZRED	AGTUNFED
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AGTUNFED	AGT1 Underflow Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b1	DTCZRED	Last DTC Transmission Completion Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b2	DTCNZRED	Not Last DTC Transmission Completion Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b3	AD0MATED	AD Compare Match 0 Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b4	AD0UMTED	AD Compare Mismatch 0 Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b5	AD1MATED	AD Compare Match 1 Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b6	AD1UMTED	AD Compare Mismatch 1 Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b7	SCI0UMTED	SCI0 Address Mismatch Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W

To use one of the triggers shown in Table 11.8 as a condition for switching from Snooze to Software Standby mode, set the associated bit in the SNZEDCR register to 1.

The event that is used for returning to Normal mode from Snooze mode as listed in Table 11.3 must not be enabled in the SNZEDCR register.

AGTUNFED bit (AGT1 Underflow Snooze End Enable)

The AGTUNFED bit enables or disables a transition from Snooze to Software Standby mode on an AGT1 underflow. For details on the trigger conditions, see section 25, Asynchronous General-Purpose Timer (AGT).

DTCZRED bit (**Last DTC Transmission Completion Snooze End Enable**)

The DTCZRED bit enables or disables a transition from Snooze to Software Standby mode on completion of the last DTC transmission, signaled when the CRA or CRB register in the DTC is 0. For details on the trigger conditions, see section 18, Data Transfer Controller (DTC).

DTCNZRED bit (**Not Last DTC Transmission Completion Snooze End Enable**)

The DTCNZRED bit enables or disables a transition from Snooze to Software Standby mode on completion of each DTC transmission, signaled when the CRA or CRB register in the DTC is not 0. For details on the trigger conditions, see section 18, Data Transfer Controller (DTC).

AD0MATED bit (**AD Compare Match 0 Snooze End Enable**)

The AD0MATED bit enables or disables a transition from Snooze to Software Standby mode on an AD0 event when a conversion result matches the expected data. For details on the trigger conditions, see section 47, 12-Bit A/D Converter (ADC12).

AD0UMTED bit (**AD Compare Mismatch 0 Snooze End Enable**)

The AD0UMTED bit enables or disables a transition from Snooze to Software Standby mode on an AD0 event when the conversion result does not match the expected data. For details on the trigger conditions, see section 47, 12-Bit A/D Converter (ADC12).

AD1MATED bit (**AD Compare Match 1 Snooze End Enable**)

The AD1MATED bit enables or disables a transition from Snooze to Software Standby mode on an AD1 event when the conversion result matches the expected data. For details on the trigger conditions, see section 47, 12-Bit A/D Converter (ADC12).

AD1UMTED bit (**AD Compare Mismatch 1 Snooze End Enable**)

The AD1UMTED bit enables or disables a transition from Snooze to Software Standby mode on an AD1 event when the conversion result does not match the expected data. For details on the trigger conditions, see section 47, 12-Bit A/D Converter (ADC12).

SCI0UMTED bit (**SCI0 Address Mismatch Snooze End Enable**)

The SCI0UMTED bit enables or disables a transition from Snooze to Software Standby mode on an SCI0 event when an address received in Software Standby mode does not match the expected data. For details on the trigger conditions, see section 34, Serial Communications Interface (SCI). Only set this bit to 1 when SCI0 is operating in asynchronous mode.

11.2.10 Snooze Request Control Register (SNZREQCR)

Address(es): SYSTEM.SNZREQCR 4001 E098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	SNZREQEN30	SNZREQEN29	SNZREQEN28	—	—	SNZREQEN25	SNZREQEN24	—	SNZREQEN22	—	—	—	—	SNZREQEN17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SNZREQEN15	SNZREQEN14	SNZREQEN13	SNZREQEN12	SNZREQEN11	SNZREQEN10	SNZREQEN9	SNZREQEN8	SNZREQEN7	SNZREQEN6	SNZREQEN5	SNZREQEN4	SNZREQEN3	SNZREQEN2	SNZREQEN1	SNZREQEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SNZREQEN0	Snooze Request Enable 0	Enables the IRQ0 pin Snooze request: 0: Disable 1: Enable.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	SNZREQEN1	Snooze Request Enable 1	Enables the IRQ1 pin Snooze request: 0: Disable 1: Enable.	R/W
b2	SNZREQEN2	Snooze Request Enable 2	Enables the IRQ2 pin Snooze request: 0: Disable 1: Enable.	R/W
b3	SNZREQEN3	Snooze Request Enable 3	Enables the IRQ3 pin Snooze request: 0: Disable 1: Enable.	R/W
b4	SNZREQEN4	Snooze Request Enable 4	Enables the IRQ4 pin Snooze request: 0: Disable 1: Enable.	R/W
b5	SNZREQEN5	Snooze Request Enable 5	Enables the IRQ5 pin Snooze request: 0: Disable 1: Enable.	R/W
b6	SNZREQEN6	Snooze Request Enable 6	Enables the IRQ6 pin Snooze request: 0: Disable 1: Enable.	R/W
b7	SNZREQEN7	Snooze Request Enable 7	Enables the IRQ7 pin Snooze request: 0: Disable 1: Enable.	R/W
b8	SNZREQEN8	Snooze Request Enable 8	Enables the IRQ8 pin Snooze request: 0: Disable 1: Enable.	R/W
b9	SNZREQEN9	Snooze Request Enable 9	Enables the IRQ9 pin Snooze request: 0: Disable 1: Enable.	R/W
b10	SNZREQEN10	Snooze Request Enable 10	Enables the IRQ10 pin Snooze request: 0: Disable 1: Enable.	R/W
b11	SNZREQEN11	Snooze Request Enable 11	Enables the IRQ11 pin Snooze request: 0: Disable 1: Enable.	R/W
b12	SNZREQEN12	Snooze Request Enable 12	Enables the IRQ12 pin Snooze request: 0: Disable 1: Enable.	R/W
b13	SNZREQEN13	Snooze Request Enable 13	Enables the IRQ13 pin Snooze request: 0: Disable 1: Enable.	R/W
b14	SNZREQEN14	Snooze Request Enable 14	Enables the IRQ14 pin Snooze request: 0: Disable 1: Enable.	R/W
b15	SNZREQEN15	Snooze Request Enable 15	Enables the IRQ15 pin Snooze request: 0: Disable 1: Enable.	R/W
b16	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b17	SNZREQEN17	Snooze Request Enable 17	Enables the Key Interrupt Snooze request: 0: Disable 1: Enable.	R/W
b21 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b22	SNZREQEN22	Snooze Request Enable 22	Enables the ACMPHS0 Snooze request: 0: Disable 1: Enable.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b24	SNZREQEN24	Snooze Request Enable 24	Enables the RTC alarm Snooze request: 0: Disable 1: Enable.	R/W

Bit	Symbol	Bit name	Description	R/W
b25	SNZREQEN25	Snooze Request Enable 25	Enables the RTC period Snooze request: 0: Disable 1: Enable.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	SNZREQEN28	Snooze Request Enable 28	Enables the AGT1 underflow Snooze request: 0: Disable 1: Enable.	R/W
b29	SNZREQEN29	Snooze Request Enable 29	Enables the AGT1 compare match A Snooze request: 0: Disable 1: Enable.	R/W
b30	SNZREQEN30	Snooze Request Enable 30	Enables the AGT1 compare match B Snooze request: 0: Disable 1: Enable.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The SNZREQCR register controls which triggers cause the MCU to switch from Software Standby to Snooze mode. If a trigger is selected as a request to cancel Software Standby mode in the WUPEN register (see section 14, Interrupt Controller Unit (ICU)), the MCU enters Normal mode when the trigger is generated even when the associated bit of the SNZREQCR is 1. The WUPEN register setting always has higher priority than the SNZREQCR register setting. For details, see section 11.8, Snooze Mode, and section 14, Interrupt Controller Unit (ICU).

11.2.11 Deep Software Standby Control Register (DPSBYCR)

Address(es): SYSTEM.DPSBYCR 4001 E400h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSBY	IOKEEP	—	—	—	—	DEEPCUT[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	DEEPCUT[1:0]	Power-Supply Control	b1 b0 0 0: Supply power to the Standby SRAM, low-speed on-chip oscillator, AGTn, and USBFS/USBHS resume detecting unit in Deep Software Standby mode 0 1: Do not supply power to the Standby SRAM, low-speed on-chip oscillator, AGTn, and USBFS/USBHS resume detecting unit in Deep Software Standby mode 1 0: Setting prohibited 1 1: Do not supply power to the Standby SRAM, low-speed on-chip oscillator, AGTn, and USBFS/USBHS resume detecting unit in Deep Software Standby mode. In addition, disable the LVD and enable the low-power function of the power-on reset circuit.	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	IOKEEP	I/O Port Retention	0: When Deep Software Standby mode is canceled, clear the I/O ports to the reset state 1: When Deep Software Standby mode is canceled, keep the I/O ports in the same state as in Deep Software Standby mode.	R/W
b7	DPSBY	Deep Software Standby	0: Sleep mode (SBYCR.SSBY = 0) or Software Standby mode (SBYCR.SSBY = 1) 1: Sleep mode (SBYCR.SSBY = 0) or Deep Software Standby mode (SBYCR.SSBY = 1).	R/W

The DPSBYCR register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source.

DEEPCUT[1:0] bits (Power-Supply Control)

The DEEPCUT[1:0] bits control the internal power supply to the Standby SRAM, low-speed on-chip oscillator, AGTn, and USBFS/USBHS resume detecting unit in Deep Software Standby mode. In addition, these bits control the state of the LVD and power-on reset circuit in Deep Software Standby mode. When a USBFS/HS suspend/resume interrupt is used as a canceling source for Deep Software Standby mode, the DEEPCUT[1:0] bits must be set to 00b. When an LVD interrupt is used in Deep Software Standby mode, the DEEPCUT[1:0] bits must be set to 00b or 01b.

For lower power consumption, set the DEEPCUT[1:0] bits to 11b so that the LVD is stopped and the low power mode function of the power-on reset circuit is enabled. The internal power supply of the SRAM stops in Deep Software Standby mode regardless of the DEEPCUT[1:0] bit settings.

IOKEEP bit (I/O Port Retention)

In Deep Software Standby mode, the I/O ports keep the same states as in Software Standby mode. The IOKEEP bit specifies whether to reset the state of the I/O ports when Deep Software Standby mode is canceled.

DPSBY bit (Deep Software Standby)

The DPSBY bit controls transitions to Deep Software Standby mode. See Table 11.6 for details.

When the WFI instruction is executed while the SBYCR.SSBY and DPSBYCR.DPSBY bits are both 1, the MCU enters Deep Software Standby mode through Software Standby mode.

The DPSBY bit remains 1 when Deep Software Standby mode is canceled by certain pins that are the sources of external pin interrupts (NMI and IRQ0-DS to IRQ14-DS) or by a peripheral interrupt (RTC alarm, RTC interval, USB suspend/resume, voltage monitor 1, or voltage monitor 2). Write 0 to this bit to clear it.

The DPSBY setting is invalid when the OFS0.IWDTSTPCTL bit is 0 (counting continues), regardless of the setting in the OFS0.IWDTSTRT bit. When the SBYCR.SSBY and DPSBY bits are 1, the MCU transitions to Software Standby mode on execution of a WFI instruction.

The DPSBY setting is invalid when the voltage monitor 1 reset is enabled (LVD1CR0.RI = 1) or when the voltage monitor 2 reset is enabled (LVD2CR0.RI = 1). When the SBYCR.SSBY and the DPSBY bits are 1, the MCU transitions to Software Standby mode on execution of a WFI instruction.

11.2.12 Deep Software Standby Interrupt Enable Register 0 (DPSIER0)

Address(es): SYSTEM.DPSIER0 4001 E402h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DIRQ0E	IRQ0-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ0-DS pin: 0: Disable 1: Enable.	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ1-DS pin: 0: Disable 1: Enable.	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ2-DS pin: 0: Disable 1: Enable.	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ3-DS pin: 0: Disable 1: Enable.	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ4-DS pin: 0: Disable 1: Enable.	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ5-DS pin: 0: Disable 1: Enable.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ6-DS pin: 0: Disable 1: Enable.	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ7-DS pin: 0: Disable 1: Enable.	R/W

The DPSIER0 register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source. After a setting in DPSIER0 is changed, an edge can be internally generated depending on the associated pin state, resulting in the associated DPSIFR0 bit being set to 1. Clear DPSIFR0 to 0 before entering Deep Software Standby mode.

11.2.13 Deep Software Standby Interrupt Enable Register 1 (DPSIER1)

Address(es): SYSTEM.DPSIER1 4001 E403h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DIRQ8E	IRQ8-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ8-DS pin: 0: Disable 1: Enable.	R/W
b1	DIRQ9E	IRQ9-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ9-DS pin: 0: Disable 1: Enable.	R/W
b2	DIRQ10E	IRQ10-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ10-DS pin: 0: Disable 1: Enable.	R/W
b3	DIRQ11E	IRQ11-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ11-DS pin: 0: Disable 1: Enable.	R/W
b4	DIRQ12E	IRQ12-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ12-DS pin: 0: Disable 1: Enable.	R/W
b5	DIRQ13E	IRQ13-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ13-DS pin: 0: Disable 1: Enable.	R/W
b6	DIRQ14E	IRQ14-DS Pin Enable	Enables canceling of Deep Software Standby mode by the IRQ14-DS pin: 0: Disable 1: Enable.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DPSIER1 register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source. After a setting in DPSIER1 is changed, an edge can be internally generated depending on the associated pin state, resulting in the associated DPSIFR1 bit being set to 1. Clear DPSIFR1 to 0 before entering Deep Software Standby mode.

11.2.14 Deep Software Standby Interrupt Enable Register 2 (DPSIER2)

Address(es): SYSTEM.DPSIER2 4001 E404h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DNMIE	DRTCAIE	DRTCIIE	DLVD2IE	DLVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DLVD1IE	LVD1 Deep Software Standby Cancel Signal Enable	Enables canceling of Deep Software Standby mode by the voltage monitor 1 signal: 0: Disable 1: Enable.	R/W
b1	DLVD2IE	LVD2 Deep Software Standby Cancel Signal Enable	Enables canceling of Deep Software Standby mode by the voltage monitor 2 signal: 0: Disable 1: Enable.	R/W
b2	DRTCIIE	RTC Interval Interrupt Deep Software Standby Cancel Signal Enable	Enables canceling of Deep Software Standby mode by the RTC interval interrupt signal: 0: Disable 1: Enable.	R/W
b3	DRTCAIE	RTC Alarm Interrupt Deep Software Standby Cancel Signal Enable	Enables canceling of Deep Software Standby mode by the RTC alarm interrupt signal: 0: Disable 1: Enable.	R/W
b4	DNMIE	NMI Pin Enable	Enables canceling of Deep Software Standby mode by the NMI pin: 0: Disable 1: Enable.	R/W*1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. 1 can be written only once. After 1 is written to this bit, subsequent write accesses are disabled.

The DPSIER2 register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source. After a setting in DPSIER2 is changed, an edge can be internally generated depending on the associated pin state, resulting in the associated DPSIFR2 bit being set to 1. Clear DPSIFR2 to 0 before entering Deep Software Standby mode.

11.2.15 Deep Software Standby Interrupt Enable Register 3 (DPSIER3)

Address(es): SYSTEM.DPSIER3 4001 E405h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAGT1IE	DUSBHSIE	DUSBFSIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DUSBFSIE	USBFS Suspend/Resume Deep Software Standby Cancel Signal Enable	Enables canceling of Deep Software Standby mode by a USBFS suspend/resume: 0: Disable 1: Enable.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	DUSBHSIE	USBHS Suspend/Resume Deep Software Standby Cancel Signal Enable	Enables canceling of Deep Software Standby mode by a USBHS suspend/resume: 0: Disable 1: Enable.	R/W
b2	DAGT1IE	AGT1 Underflow Deep Software Standby Cancel Signal Enable	Enables canceling of Deep Software Standby mode by an AGT1 underflow: 0: Disable 1: Enable.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DPSIER3 register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source. After a setting in DPSIER3 is changed, an edge can be internally generated depending on the associated pin state, resulting in the associated DPSIFR3 bit setting to 1. Clear DPSIFR3 to 0 before entering Deep Software Standby mode.

11.2.16 Deep Software Standby Interrupt Flag Register 0 (DPSIFR0)

Address(es): SYSTEM.DPSIFR0 4001 E406h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DIRQ0F	IRQ0-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ0-DS pin: 0: No request generated 1: Request generated.	R(/W) *1
b1	DIRQ1F	IRQ1-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ1-DS pin: 0: No request generated 1: Request generated.	R(/W) *1
b2	DIRQ2F	IRQ2-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ2-DS pin: 0: No request generated 1: Request generated.	R(/W) *1
b3	DIRQ3F	IRQ3-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ3-DS pin: 0: No request generated 1: Request generated.	R(/W) *1
b4	DIRQ4F	IRQ4-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ4-DS pin: 0: No request generated 1: Request generated.	R(/W) *1
b5	DIRQ5F	IRQ5-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ5-DS pin: 0: No request generated 1: Request generated.	R(/W) *1
b6	DIRQ6F	IRQ6-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ6-DS pin: 0: No request generated 1: Request generated.	R(/W) *1
b7	DIRQ7F	IRQ7-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ7-DS pin: 0: No request generated 1: Request generated.	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

The flags in the DPSIFR0 register set to 1 when the associated cancel request specified in DPSIEGR0 is generated. Each flag can be set to 1 when a cancel request is generated in any mode, not only in Deep Software Standby mode, or when the setting in DPSIER0 is changed. Clear DPSIFR0 to 00h before entering Deep Software Standby mode.

To clear DPSIFR0 to 00h after modifying DPSIER0, wait for at least 6 PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. 6 or more PCLKB cycles can be secured, for example, by reading DPSIER0. DPSIFR0 is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source.

DIRQnF flags (IRQn-DS Deep Software Standby Cancel Flag) (n = 0 to 7)

The DIRQnF flag indicates that a cancel request was generated by the IRQn-DS pin.

[Setting condition]

- A cancel request generated by an IRQn-DS pin specified in DPSIEGR0.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

11.2.17 Deep Software Standby Interrupt Flag Register 1 (DPSIFR1)

Address(es): SYSTEM.DPSIFR1 4001 E407h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DIRQ8F	IRQ8-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ8-DS pin: 0: No request generated 1: Request generated.	R(/W)*1
b1	DIRQ9F	IRQ9-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ9-DS pin: 0: No request generated 1: Request generated.	R(/W)*1
b2	DIRQ10F	IRQ10-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ10-DS pin: 0: No request generated 1: Request generated.	R(/W)*1
b3	DIRQ11F	IRQ11-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ11-DS pin: 0: No request generated 1: Request generated.	R(/W)*1
b4	DIRQ12F	IRQ12-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ12-DS pin: 0: No request generated 1: Request generated.	R(/W)*1
b5	DIRQ13F	IRQ13-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ13-DS pin: 0: No request generated 1: Request generated.	R(/W)*1
b6	DIRQ14F	IRQ14-DS Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the IRQ14-DS pin: 0: No request generated 1: Request generated.	R(/W)*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R(/W)*1

Note 1. Only 0 can be written to clear the flag.

The flags in the DPSIFR1 register set to 1 when the associated cancel request specified in DPSIEGR1 is generated. Each flag can be set to 1 when a cancel request is generated in any mode, not only in Deep Software Standby mode, or when the setting in DPSIER1 is changed. Clear DPSIFR1 to 00h before entering Deep Software Standby mode.

To clear DPSIFR1 to 00h after modifying DPSIER1, wait for at least 6 PCLKB cycles, read DPSIFR1, and then write 0 to DPSIFR1. 6 or more PCLKB cycles can be secured, for example, by reading DPSIER1. DPSIFR1 is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source.

DIRQnF flags (IRQn-DS Deep Software Standby Cancel Flag) (n = 8 to 14)

The DIRQnF flag indicates that a cancel request was generated by the IRQn-DS pin.

[Setting condition]

- A cancel request generated by the IRQn-DS pin specified in DPSIEGR1.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

11.2.18 Deep Software Standby Interrupt Flag Register 2 (DPSIFR2)

Address(es): SYSTEM.DPSIFR2 4001 E408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DNMIF	DRTCAIF	DRTCIIF	DLVD2IF	DLVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DLVD1IF	LVD1 Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the voltage monitor 1 signal: 0: No request generated 1: Request generated.	R(/W)*1
b1	DLVD2IF	LVD2 Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the voltage monitor 2 signal: 0: No request generated 1: Request generated.	R(/W)*1
b2	DRTCIIF	RTC Interval Interrupt Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the RTC interval interrupt signal: 0: No request generated 1: Request generated.	R(/W)*1
b3	DRTCAIF	RTC Alarm Interrupt Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the RTC alarm interrupt signal: 0: No request generated 1: Request generated.	R(/W)*1
b4	DNMIF	NMI Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by the NMI pin: 0: No request generated 1: Request generated.	R(/W)*1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag.

The flags in the DPSIFR2 register set to 1 when the associated cancel request specified in DPSIEGR2 is generated. Each flag can be set to 1 when a cancel request is generated in any mode, not only in Deep Software Standby mode, or when the setting in DPSIER2 is changed. Clear DPSIFR2 to 00h before entering Deep Software Standby mode.

To clear DPSIFR2 to 00h after modifying DPSIER2, wait for at least 6 PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. 6 or more PCLKB cycles can be secured, for example, by reading DPSIER2. DPSIFR2 is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source.

DLVDmIF flag (LVDm Deep Software Standby Cancel Flag) (m = 1 or 2)

The DLVDmIF flag indicates that a cancel request was generated by the voltage monitor m signal.

[Setting condition]

- A cancel request generated by the voltage monitor m signal specified in DPSIEGR2.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

DRTCIIF flag (RTC Interval Interrupt Deep Software Standby Cancel Flag)

The DRTCIIF flag indicates that a cancel request was generated by the RTC interval interrupt signal.

[Setting condition]

- A cancel request generated by the RTC interval interrupt signal.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

DRTCAIF flag (RTC Alarm Interrupt Deep Software Standby Cancel Flag)

The DRTCAIF flag indicates that a cancel request was generated by the RTC alarm interrupt signal.

[Setting condition]

- A cancel request generated by the RTC alarm interrupt signal.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

DNMIF flag (NMI Deep Software Standby Cancel Flag)

The DNMIF flag indicates that a cancel request was generated by the NMI pin.

[Setting condition]

- A cancel request generated by the NMI pin specified in DPSIEGR2.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

11.2.19 Deep Software Standby Interrupt Flag Register 3 (DPSIFR3)

Address(es): SYSTEM.DPSIFR3 4001 E409h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAGT1 IF	DUSBH SIF	DUSBF SIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DUSBFSIF	USBFS Suspend/Resume Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by a USBFS suspend/resume: 0: No request generated 1: Request generated.	R(/W)*1
b1	DUSBHSIF	USBHS Suspend/Resume Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by a USBHS suspend/resume: 0: No request generated 1: Request generated.	R(/W)*1

Bit	Symbol	Bit name	Description	R/W
b2	DAGT1IF	AGT1 Underflow Deep Software Standby Cancel Flag	Indicates Deep Software Standby cancel request by an AGT1 underflow: 0: No request generated 1: Request generated.	R(/W)*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag.

The flags in the DPSIFR3 register set to 1 when the associated cancel request is generated. Each flag can be set to 1 when a cancel request is generated in any mode, not only in Deep Software Standby mode, or when the setting in DPSIER3 is changed. Clear DPSIFR3 to 00h before entering Deep Software Standby mode.

To clear DPSIFR3 to 00h after modifying DPSIER3, wait for at least 6 PCLKB cycles, read DPSIFR3, and then write 0 to DPSIFR3. 6 or more PCLKB cycles can be secured, for example, by reading DPSIER3. DPSIFR3 is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see section 6, Resets.

DUSBFSIF flag (USBFS Suspend/Resume Deep Software Standby Cancel Flag)

The DUSBFSIF flag indicates that a cancel request was generated by a USBFS suspend/resume.

[Setting condition]

- A cancel request generated by the USBFS suspend/resume.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

DUSBHSIF flag (USBHS Suspend/Resume Deep Software Standby Cancel Flag)

This DUSBHSIF flag indicates that a cancel request was generated by a USBHS suspend/resume.

[Setting condition]

- A cancel request generated by the USBHS suspend/resume.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

DAGT1IF flag (AGT1 Underflow Deep Software Standby Cancel Flag)

The DAGT1IF flag indicates that a cancel request was generated by an AGT1 underflow.

[Setting condition]

- A cancel request generated by the AGT1 underflow.

[Clearing condition]

- Writing 0 to the flag after reading it as 1.

11.2.20 Deep Software Standby Interrupt Edge Register 0 (DPSIEGR0)

Address(es): SYSTEM.DPSIEGR0 4001 E40Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DIRQ0EG	IRQ0-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	DIRQ1EG	IRQ1-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b2	DIRQ2EG	IRQ2-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b3	DIRQ3EG	IRQ3-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b4	DIRQ4EG	IRQ4-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b5	DIRQ5EG	IRQ5-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b6	DIRQ6EG	IRQ6-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b7	DIRQ7EG	IRQ7-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W

The DPSIEGR0 register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source.

11.2.21 Deep Software Standby Interrupt Edge Register 1 (DPSIEGR1)

Address(es): SYSTEM.DPSIEGR1 4001 E40Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DIRQ8EG	IRQ8-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b1	DIRQ9EG	IRQ9-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b2	DIRQ10EG	IRQ10-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b3	DIRQ11EG	IRQ11-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b4	DIRQ12EG	IRQ12-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b5	DIRQ13EG	IRQ13-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b6	DIRQ14EG	IRQ14-DS Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DPSIEGR1 register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source.

11.2.22 Deep Software Standby Interrupt Edge Register 2 (DPSIEGR2)

Address(es): SYSTEM.DPSIEGR2 4001 E40Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DNMIEG	—	—	DLVD2EG	DLVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DLVD1EG	LVD1 Edge Select	0: Generate cancel request when VCC < V_{det1} (fall) is detected 1: Generate cancel request when VCC ≥ V_{det1} (rise) is detected.	R/W
b1	DLVD2EG	LVD2 Edge Select	0: Generate cancel request when VCC < V_{det2} (fall) is detected 1: Generate cancel request when VCC ≥ V_{det2} (rise) is detected.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIEG	NMI Pin Edge Select	0: Generate cancel request on falling edge 1: Generate cancel request on rising edge.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DPSIEGR2 register is not initialized by the internal reset signal that cancels Deep Software Standby mode. For details, see Table 6.2, Reset detect flags initialized by each reset source.

11.2.23 System Control OCD Control Register (SYOCDCR)

Address(es): SYSTEM.SYOCDCR 4001 E40Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	DBGEN	—	—	—	—	—	—	DOCDF
Value after reset:	0	0	0	0	0	0	0	x

Bit	Symbol	Bit name	Description	R/W
b0	DOCDF	Deep Software Standby OCD Flag	Indicates cancel request by the DBIRQ: 0: DBIRQ is not generated 1: DBIRQ is generated.	R/(W)*1
b6 to b1	—	Reserved	These bits are read as 0. The write value must be 0.	R/W
b7	DBGEN	Debugger Enable Bit	0: Disable on-chip debugger 1: Enable on-chip debugger. Set to 1 first in on-chip debug mode.	R/W

Note 1. Writing 0 clears the flag. Writing 1 is ignored.

SYOCDCR is not initialized by the internal reset signal that cancels Deep Software Standby mode.

DOCDF flag (Deep Software Standby OCD Flag)

The DOCDF flag indicates that a Deep Software Standby cancel request was generated by the MCUCTRL.DBIRQ bit. The flag is set to 1 when the cancel request is generated. The flag can be set to 1 when a cancel request is generated in any mode, not only in Deep Software Standby mode. Clear the DOCDF flag to 0 before entering Deep Software Standby mode.

[Setting condition]

- A cancel request generated by the MCUCTRL.DBIRQ bit.

[Clearing condition]

- Writing 0 to the flag after reading it as 1

- When the DBGEN bit is 0.

DBGEN bit (Debugger Enable Bit)

The DBGEN bit enables the on-chip debugger mode. This bit must be set to 1 first in the on-chip debug mode.

[Setting condition]

- Writing 1 to the bit when the debugger is connected.

[Clearing condition]

- Power-on reset is generated
- Writing 0 to the bit.

11.2.24 Standby Condition Register (STCONR)

Address(es): SYSTEM.STCONR 4001 E40Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	STCON[1:0]	
Value after reset:	1	1	0	0	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b1 to b0	STCON[1:0]	SSTBY Condition Bit	b1 b0 0 0: Set this value to transition to Software Standby mode when using HOCO 1 1: Set this value to transition to Software Standby mode when not using HOCO.	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

STCON[1:0] bits (SSTBY Condition Bit)

The STCON[1:0] bits must always be set to 00b when using HOCO to enter Software Standby mode.

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKDIVCR.FCK[2:0], ICK[2:0], BCK[2:0], PCKA[2:0], PCKB[2:0], PCKC[2:0], and PCKD[2:0] bits are set, the clock frequency changes. The module and clock associations are as follows:

- The CPU, DMAC, DTC, flash, and SRAM use the operating clock specified in the ICK[2:0] bits
- Peripheral modules use the operating clocks specified in the PCKA[2:0], PCKB[2:0], PCKC[2:0], and PCKD[2:0] bits
- The flash memory interface uses the operating clock specified in the FCK[2:0] bits
- The external bus uses the operating clock specified in the BCK[2:0] bits.

For details, see section 9, Clock Generation Circuit.

11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module. When the MSTPmi bit (m = A to D; i = 31 to 0) in MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. Clearing the MSTPmi bit to 0 cancels the module-stop state, allowing the module to resume operation at the end of the bus cycle. The internal states of the modules are retained in the module-stop state.

After a reset is canceled, all modules other than the DMAC, DTC, and SRAM modules are placed in the module-stop state. Do not access the module while the associated MSTPmi bit is 1. Otherwise, the read/write data and the operation of the module is not guaranteed. Do not set the MSTPmi bit to 1 while the associated module is being accessed.

When the PLL is selected as the clock source, MSTPmi bits must be changed only one bit at a time. In this case, wait at least 250 ns after changing each MSTPmi bit before starting subsequent processing if you change any of the following bits: MSTPA22 (DMAC, DTC), MSTPB15 (ETHERC0, EDMAC0), MSTPB13 (EPTPC, PTPEDMAC), MSTPB12 (USBHS), MSTPC31 (SCE7), MSTPC6 (DRW), MSTPC5 (JPEG), MSTPC4 (GLCDC), or MSTPD5 (GPT32EH, GPT32E).

The recommended method to measure the wait time is to do so in software. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses

11.5 Function for Lower Operating Power Consumption

Power consumption can be reduced in Normal, Sleep, and Snooze modes by selecting an appropriate operating power mode for the given operating frequency and operating voltage.

11.5.1 Setting the Operating Power Control Mode

Make sure that the operating conditions, such as the voltage and frequency ranges, are always within the specified ranges before and after switching the operating power control modes. This section provides example procedures for switching operating power control modes.

Table 11.5 Available oscillators in each mode

	Oscillator						
Mode	**PLL**	**High-speed on-chip oscillator**	**Middle-speed on-chip oscillator**	**Low-speed on-chip oscillator**	**Main clock oscillator**	**Sub-clock oscillator**	**IWDT-dedicated on-chip oscillator**
High-speed	Available	Available	Available	Available	Available	Available	Available
Low-speed	N/A	Available	Available	Available	Available	Available	Available
Subosc-speed	N/A	N/A	N/A	Available	N/A	Available	Available

(1) Switching from a higher to a lower power mode

Example 1: To switch from High-speed mode to Low-speed mode:

Operation begins in High-speed mode.

1. Change the oscillator to that used in Low-speed mode. Set the frequency of each clock lower than the maximum operating frequency in Low-speed mode.
2. Turn off the oscillator that is not required in Low-speed mode.
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Set the OPCCR.OPCM[1:0] bits to 11b (Low-speed mode).
5. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).

Operation is now in Low-speed mode.

Example 2: To switch from High-speed mode to Subosc-speed mode:

Operation begins in High-speed mode.

1. Change the clock source to the sub-clock oscillator. Turn off HOCO, MOCO, LOCO, main oscillator, and PLL.
2. Confirm that all clock sources except the sub-clock oscillator are stopped.
3. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
4. Set the SOPCCR.SOPCM bit to 1 (Subosc-speed mode).
5. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).

Operation is now in Subosc-speed mode.

(2) Switching from a lower to a higher power mode

Example 1: To switch from Subosc-speed mode to High-speed mode:

Operation begins in Subosc-speed mode.

1. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
2. Set the SOPCCR.SOPCM bit to 0 (High-speed mode).
3. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
4. Turn on the oscillator wanted in High-speed mode.
5. Set the frequency of each clock lower than the maximum operating frequency for High-speed mode.

Operation is now in High-speed mode.

Example 2: To switch Low-speed mode to High-speed mode:

Operation begins in Low-speed mode.

1. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
2. Set the OPCCR.OPCM[1:0] bits to 00b (High-speed mode).
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Turn on any oscillator wanted in High-speed mode.
5. Set the frequency of each clock lower than the maximum operating frequency for High-speed mode.

Operation is now in High-speed mode.

11.6 Sleep Mode

11.6.1 Transition to Sleep Mode

When a WFI instruction is executed while the SBYCR.SSBY bit is 0, the MCU enters Sleep mode. In Sleep mode, the CPU stops operating, but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in Sleep mode cause the MCU to cancel Sleep mode. All interrupt sources are available. If using an interrupt to cancel Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 14, Interrupt Controller Unit (ICU).

Counting by the IWDT stops when the MCU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep, Software Standby, or Snooze mode).

Counting by the IWDT continues when the MCU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Sleep, Software Standby, or Snooze mode).

Counting by the WDT stops when the MCU enters Sleep mode while the WDT is in auto start mode and the OFS0.WDTSTPCTL bit is 1 (WDT stops in Sleep mode). In the same way, counting by the WDT stops when the MCU enters Sleep mode while the WDT is in register start mode and the WDTCSTPR.SLCSTP bit in is 1 (WDT stops in Sleep mode).

Counting by the WDT continues when the MCU enters Sleep mode while the WDT is in auto start mode and the OFS0. WDTSTPCTL bit is 0 (WDT does not stop in Sleep mode). In the same way, counting by the WDT continues when the MCU enters Sleep mode while the WDT is in register start mode and the WDTCSTPR.SLCSTP bit is 0 (WDT does not stop in Sleep mode).

11.6.2 Canceling Sleep Mode

Sleep mode is canceled by any interrupt, RES pin reset, power-on reset, voltage monitor reset, SRAM parity error reset, SRAM ECC error reset, Bus master MPU error reset, Bus slave MPU error reset, or reset caused by IWDT or WDT underflow. The operations are as follows:

- Canceling by an interrupt
  When an available interrupt request (see Table 11.3) is generated, Sleep mode is canceled and the MCU starts the

interrupt handling.

- Canceling by RES pin reset
  When the RES pin is driven low, the MCU enters the reset state. You must keep the RES pin low for the time period specified in section 60, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts reset exception handling.
- Canceling by IWDT reset
  Sleep mode is canceled by an internal reset generated by an IWDT underflow, and the MCU starts reset exception handling. Under the following conditions, the IWDT stops in Sleep mode and an internal reset for canceling Sleep mode is not generated:
  - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.
- Canceling by WDT reset
  Sleep mode is canceled by an internal reset generated by an WDT underflow and the MCU starts reset exception handling. Under the following conditions, the WDT stops in Sleep mode even when counting in Normal mode and an internal reset for canceling Sleep mode is not generated:
  - OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1
  - OFS0.WDTSTRT = 1 (register start mode) and WDTCSTPR.SLCSTP = 1.
- Canceling by other resets available in Sleep mode
  Sleep mode is canceled by the associated resets, and the MCU starts reset exception handling.

Note: For details on correct setting of the interrupts, see section 14, Interrupt Controller Unit (ICU).

11.7 Software Standby Mode

11.7.1 Transition to Software Standby Mode

When a WFI instruction is executed while the SBYCR.SSBY bit is 1 and the DPSBYCR.DPSBY bit is 0, the MCU enters Software Standby mode. In this mode, the CPU, most of the on-chip peripheral functions, and the oscillators stop. However, the contents of the CPU internal registers and the SRAM data, the states of the on-chip peripheral functions, and the I/O port states are retained. Software Standby mode allows a significant reduction in power consumption because most of the oscillators stop in this mode. Table 11.2 shows the status of the on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode cause the MCU to cancel Software Standby mode. See Table 11.3 for available interrupt sources and section 14.2.9, Wake Up Interrupt Enable Register (WUPEN) for information on waking up the MCU from Software Standby mode. If using an interrupt to cancel Software Standby mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 14, Interrupt Controller Unit (ICU).

The status of the address bus and bus control signals in Software Standby mode can be selected with the SBYCR.OPE bit.

Clear the DMAST.DMST and DTCST.DTCST bits to 0 before executing a WFI instruction, except when using the DTC in Snooze mode. If the DTC is required in Snooze mode, set the DTCST.DTCST bit to 1 before executing a WFI instruction.

Counting by the IWDT stops when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep, Software Standby, or Snooze mode).

Counting by the IWDT continues when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Sleep mode, Software Standby, or Snooze mode).

The WDT stops counting when the MCU enters Software Standby mode.

Do not enter Software Standby mode while OSTDCR.OSTDE is 1 (oscillation stop detection function enabled). To enter Software Standby mode, execute a WFI instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE is 0). If the software executes a WFI instruction while OSTDCR.OSTDE is 1, the MCU enters Sleep mode even when SBYCR.SSBY is 1. Do not enter Software Standby mode while the flash memory is performing a programming or erasing procedure. To enter Software Standby mode, execute a WFI instruction after the programming or erasing procedure completes.

When the PLL is selected as the clock source, set the following modules into the module-stop state before executing a

WFI instruction: ETHERC, EPTPC, EDMAC, SCE7, DRW, JPEG, GLCDC, GPT32EH, GPT32E. In this case, you must also insert wait time at least 750 ns before executing the WFI instruction. The recommended method to measure the wait time is to do so in software. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses.

Table 11.6 shows the setting of the related control bits and the modes entered on execution of a WFI instruction.

Figure 11.2 shows an example flow for transitioning to Software Standby or Deep Software Standby mode.

Table 11.6 Bit settings that affect modes on WFI instruction execution

		SBYCR.SSBY and DPSBYCR.DPSBY bit settings			
Other bit settings		**SSBY = 0, DPSBY = 0**	**SSBY = 0, DPSBY = 1**	**SSBY = 1, DPSBY = 0**	**SSBY = 1, DPSBY = 1**
OSTDCR.OSTDE	0	Sleep mode	Sleep mode	Software Standby mode	Deep Software Standby mode
	1			Sleep mode	Sleep mode
FENTRYR.FENTRYi	0	Sleep mode	Sleep mode	Software Standby mode	Deep Software Standby mode
	1			Sleep mode	Sleep mode
OFS0.IWDTSTPCTL	0	Sleep mode	Sleep mode	Software Standby mode	Software Standby mode
	1				Deep Software Standby mode
LVD1CR0.RI	0	Sleep mode	Sleep mode	Software Standby mode	Deep Software Standby mode
	1				Software Standby mode
LVD2CR0.RI	0	Sleep mode	Sleep mode	Software Standby mode	Deep Software Standby mode
	1				Software Standby mode

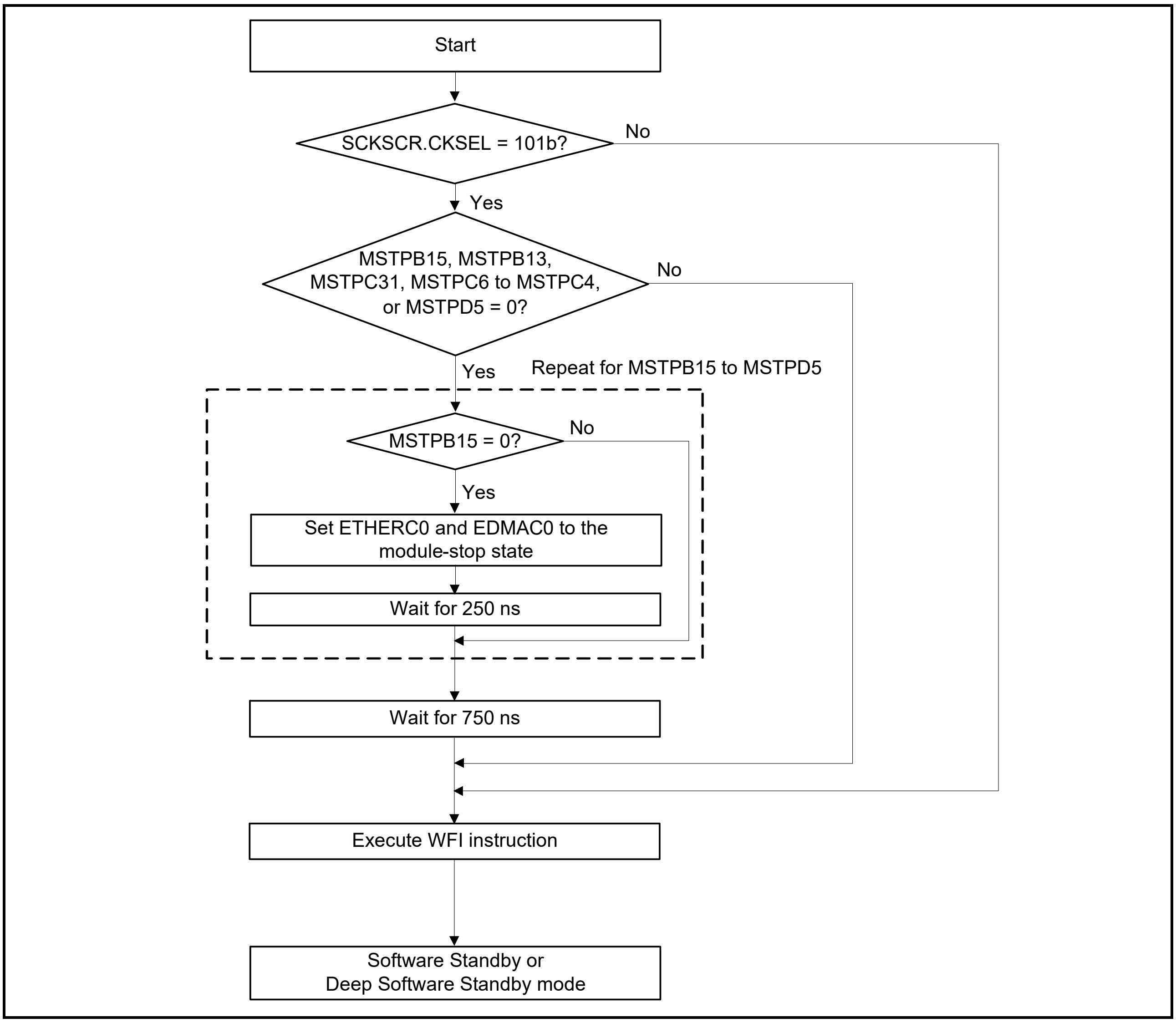


Figure 11.2 Example flow for transition to Software Standby or Deep Software Standby mode

11.7.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in Table 11.3
- A RES pin reset
- A power-on reset
- A voltage monitor reset
- A reset caused by an IWDT underflow.

On exiting Software Standby, the oscillators that operate before transitioning to the mode restart. After all of these oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode. See section 14.2.9, Wake Up Interrupt Enable Register (WUPEN), for information on waking up the MCU from Software Standby mode.

You can cancel Software Standby mode in any of the following ways:

- Canceling by an interrupt
  When an available interrupt request (see Table 11.3) is generated, all oscillators that were operating before the transition to Software Standby mode restart. After all of these oscillators are stabilized, the MCU cancels Software Standby mode and starts the interrupt handling. When the PLL is selected as the clock source, you must insert wait

time at least 250 ns at the beginning of the interrupt handling. The recommended method to measure the wait time is to do so in software. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses. Figure 11.3 shows an example flow for canceling Software Standby by an interrupt.

- Canceling by RES pin reset
  When the RES pin is driven low, the MCU enters the reset state and the oscillators start operating in their default status. Make sure to keep the RES pin low for the time period specified in section 60, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts reset exception handling.
- Canceling by a power-on reset
  Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.
- Canceling by a voltage monitor reset
  Software Standby mode is canceled by a voltage monitor reset from the voltage detection circuit and the MCU starts the reset exception handling.
- Canceling by IWDT reset
  Software Standby mode is canceled by an internal reset generated by an IWDT underflow, and the MCU starts reset exception handling. However, the IWDT stops in Software Standby mode and an internal reset for canceling Software Standby mode is not generated in the following conditions:
  - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.
- Canceling by other resets available in Software Standby mode
  Software Standby mode is canceled by the associated resets, and the MCU starts reset exception handling.

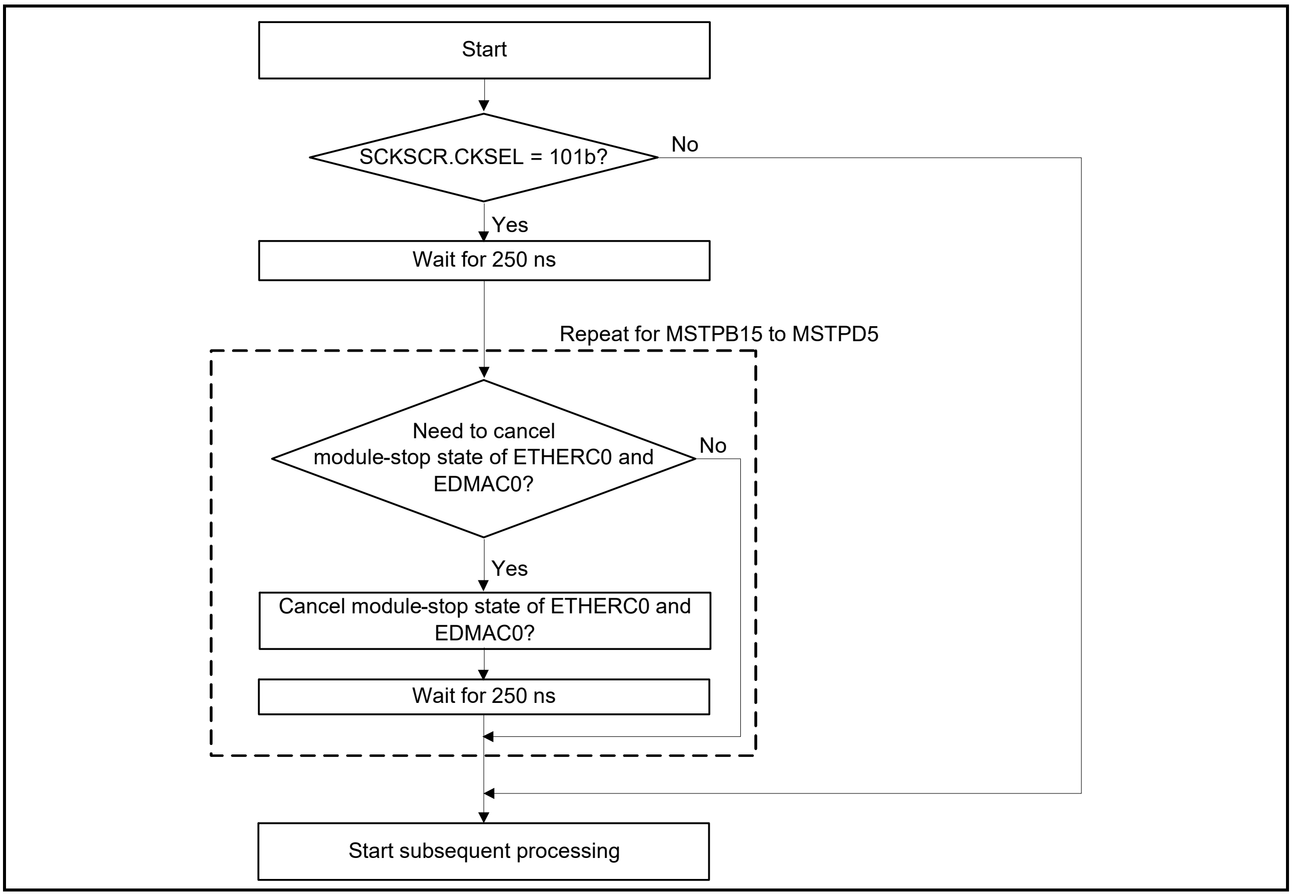


Figure 11.3 Example flow for canceling Software Standby mode

11.7.3 Example of Software Standby Mode Application

Figure 11.4 shows an example of entry to Software Standby mode on detection of a falling edge of the IRQn pin, and exit from Software Standby mode on a rising edge of the IRQn pin. In this example, an IRQn pin interrupt is accepted when the IRQCRi.IRQMD[1:0] bits of the ICU are set to 01b (falling edge) in Normal mode, and then set to 10b (rising edge).

Next, the SBYCR.SSBY bit is set to 1 and a WFI instruction is executed. As a result, entry to Software Standby mode completes, and exit from Software Standby mode is initiated by a rising edge of the IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see section 14, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 11.4 is specified in section 60, Electrical Characteristics.

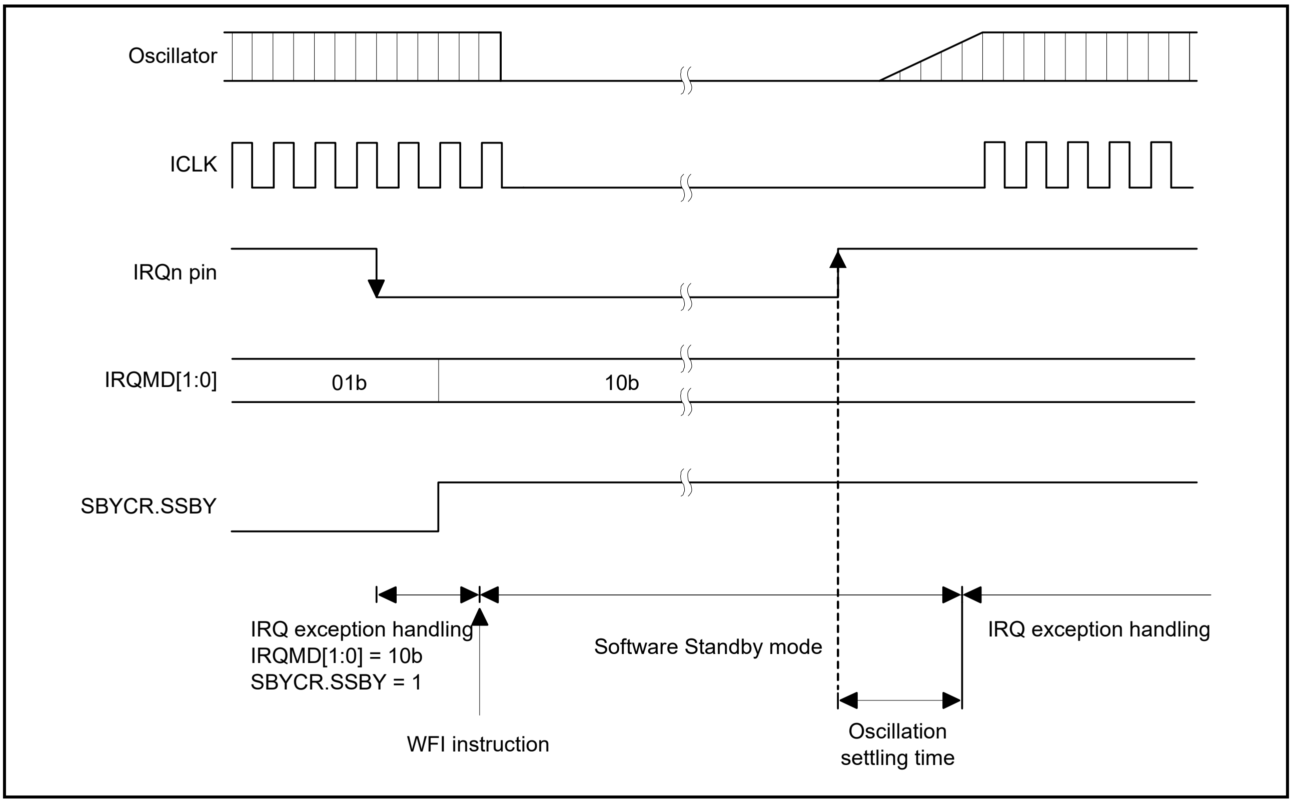


Figure 11.4 Example of Software Standby mode application

11.8 Snooze Mode

11.8.1 Transition to Snooze Mode

Figure 11.5 shows Snooze mode entry configuration. When the Snooze control circuit receives a Snooze request in Software Standby mode, the MCU transitions to Snooze mode. In this mode, some peripheral modules operate without waking up the CPU. Table 11.2 shows the peripheral modules that can operate in Snooze mode. Also, DTC operation can be selected in Snooze mode by setting the SNZCR.SNZDTCEN bit.

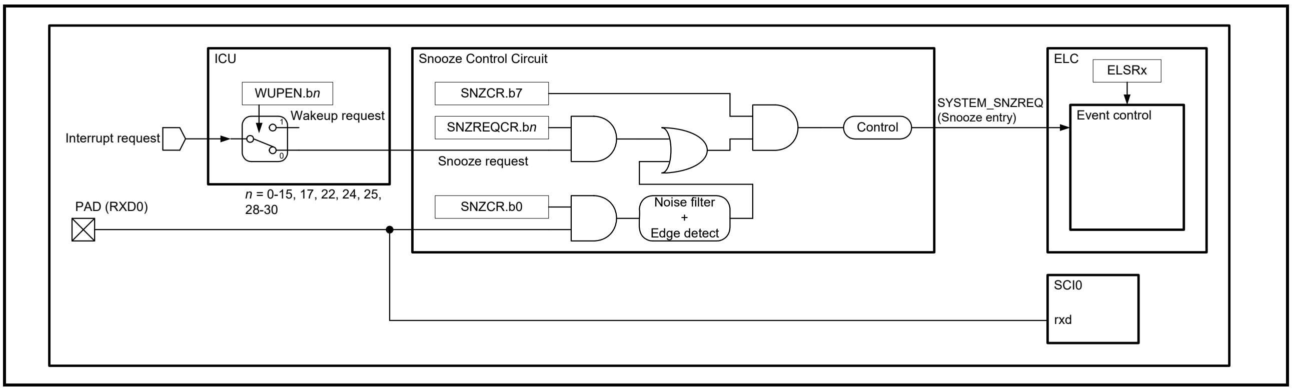


Figure 11.5 Snooze entry configuration

Table 11.7 shows the Snooze requests that switch the MCU from Software Standby to Snooze mode. To use a listed Snooze requests as a trigger to switch to Snooze mode, you must set the associated SNZREQENn bit of the SNZREQCR register or RXDREQEN bit of the SNZCR register before entering Software Standby mode.

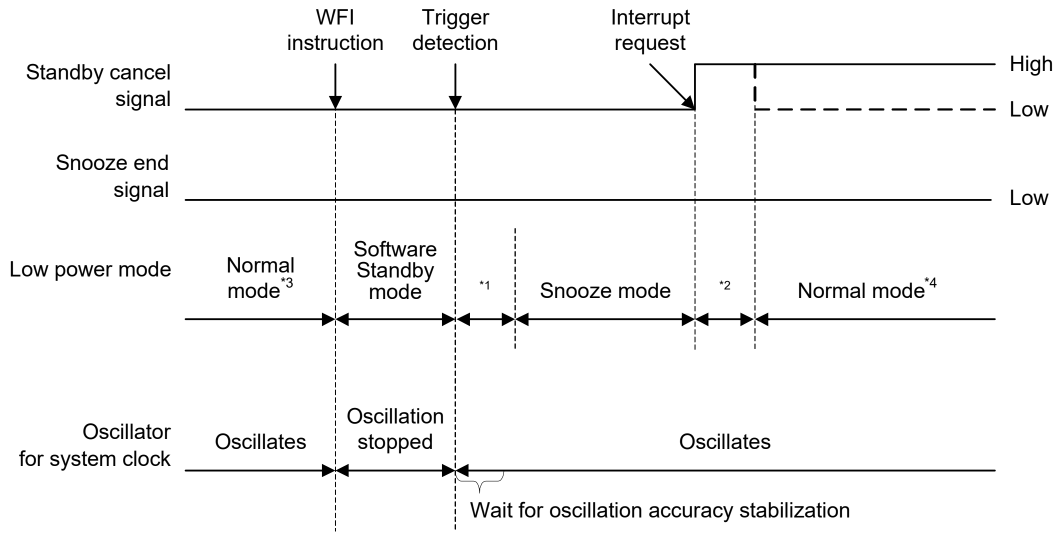
Table 11.7 Available events for invoking Snooze mode

	Control Register	
Snooze request	**Register**	**Bit*1**
PORT_IRQn (n = 0 to 15)	SNZREQCR	SNZREQENn (n = 0 to 15)
KEY_INTKR	SNZREQCR	SNZREQEN17
ACMP_HS0	SNZREQCR	SNZREQEN22
RTC_ALM	SNZREQCR	SNZREQEN24
RTC_PRD	SNZREQCR	SNZREQEN25
AGT1_AGTI	SNZREQCR	SNZREQEN28
AGT1_AGTCMAI	SNZREQCR	SNZREQEN29
AGT1_AGTCMBI	SNZREQCR	SNZREQEN30
RXD0 falling edge	SNZCR	RXDREQEN*2

Note 1. Do not enable multiple Snooze requests at the same time.
Note 2. Do not set the RXDREQEN bit to 1 except in asynchronous mode.

11.8.2 Canceling Snooze Mode

Snooze mode is canceled by any interrupt request that is available in Software Standby mode or any reset. Table 11.3 shows the requests that can be used to exit each mode. On exiting Snooze mode, the MCU transitions to Normal mode and proceeds with exception processing for the given interrupt or reset. An action triggered by the interrupt requests selected in SELSR0 cancels Snooze mode. The interrupt(s) canceling Snooze mode must be selected in IELSRn (n = 0 to 96) to link to the NVIC for the corresponding interrupt handling. See section 14, Interrupt Controller Unit (ICU), for information on setting SELSR0 and IELSRn.



Note 1. Transition time from Software Standby to Snooze mode.
Note 2. Transition time from Snooze mode to Normal mode.
Note 3. Enable Snooze mode (SNZCR.SNZE = 1) immediately before entering to Software Standby mode.
Note 4. You must disable Snooze mode (SNZCR.SNZE = 0) immediately after canceling Snooze mode.

Figure 11.6 Canceling of Snooze mode when an interrupt request signal is generated

11.8.3 Return to Software Standby Mode

Table 11.8 shows the Snooze end requests that can be used as triggers to return to Software Standby mode. The Snooze end requests are available only in Snooze mode. If the requests are generated when the MCU is not in Snooze mode, they are ignored. When multiple requests are selected, each one of the requests invokes transition to Software Standby mode from Snooze mode.

Table 11.9 shows the Snooze end conditions that consist of the Snooze end requests and the conditions of the peripheral modules. The CTSU, SCI0, ADC120, ADC121, and DTC modules can keep the MCU in Snooze mode until they complete their operation.

An AGT1 underflow as a trigger to return to Software Standby mode cancels Snooze mode without waiting for the completion of the SCI0 operation.

Figure 11.7 shows the timing diagram for the transition from Snooze mode to Software Standby mode. This mode transition occurs according to which Snooze end requests are set in the SNZEDCR register. A Snooze request is cleared automatically after it is returned to Software Standby mode.

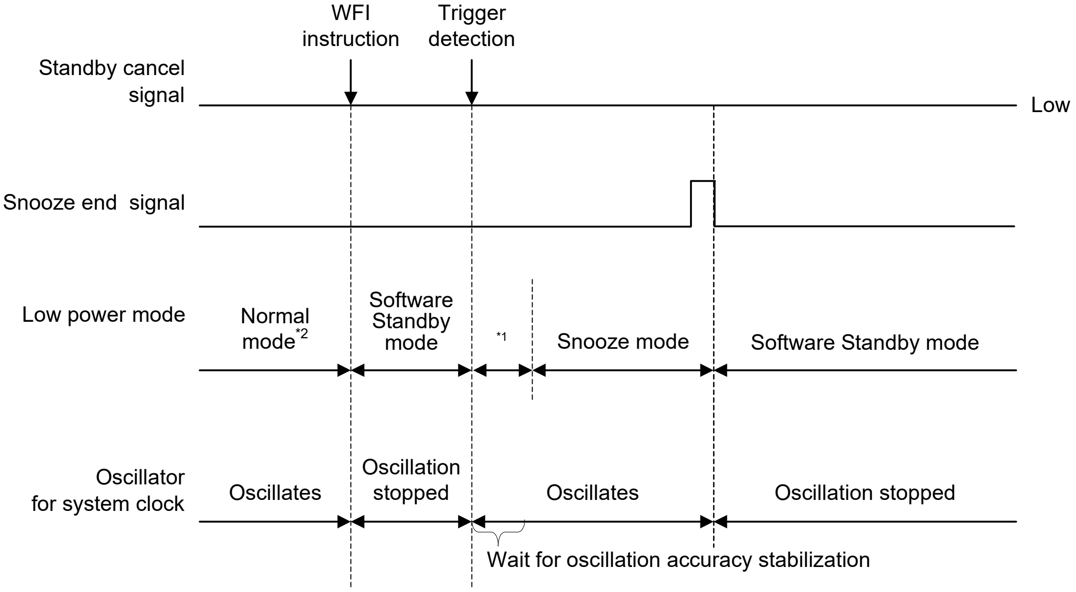
Table 11.8 Available Snooze end requests (triggers to return to Software Standby mode)

Snooze end request	Enable/disable control	
	Register	**Bit**
AGT1 Unterflow or measurement complete (AGT1_AGTI)	SNZEDCR	b0
DTC transfer completion (DTC_COMPLETE)	SNZEDCR	b1
Not DTC transfer completion (DTC_TRANSFER)	SNZEDCR	b2
ADC120 window A/B compare match (ADC120_WCMPM)	SNZEDCR	b3
ADC120 window A/B compare mismatch (ADC120_WCMPUM)	SNZEDCR	b4
ADC121 window A/B compare match (ADC121_WCMPM)	SNZEDCR	b5
ADC121 window A/B compare mismatch (ADC121_WCMPUM)	SNZEDCR	b6
SCI0 address mismatch (SCI0_DCUF)	SNZEDCR	b7

Table 11.9 Snooze end conditions

Module operating when a Snooze end request occurs	Snooze end request	
	AGT1 underflow	**All except AGT1 underflow**
DTC	The MCU transitions to Software Standby mode after all of these modules complete operation	The MCU transitions to Software Standby mode after all of these modules complete operation
ADC120		
ADC121		
CTSU		
SCI0	The MCU transitions to Software Standby mode immediately after the Snooze end request is generated	
All other modules	The MCU transitions to Software Standby mode immediately after the Snooze end request is generated	

Note: If the DTC is used to activate the ADC120, ADC121, CTSU, or SCI, the MCU transitions to Software Standby mode immediately after a Snooze end request is generated.



Note 1. Transition time from Software Standby to Snooze mode.
Note 2. Enable Snooze mode (SNZCR.SNZE = 1) immediately before entering Software Standby mode.

Figure 11.7 Canceling of Snooze mode when interrupt request signal is not generated

11.8.4 Snooze Operation Example

Figure 11.8 shows an example setting for using ELC in Snooze mode.

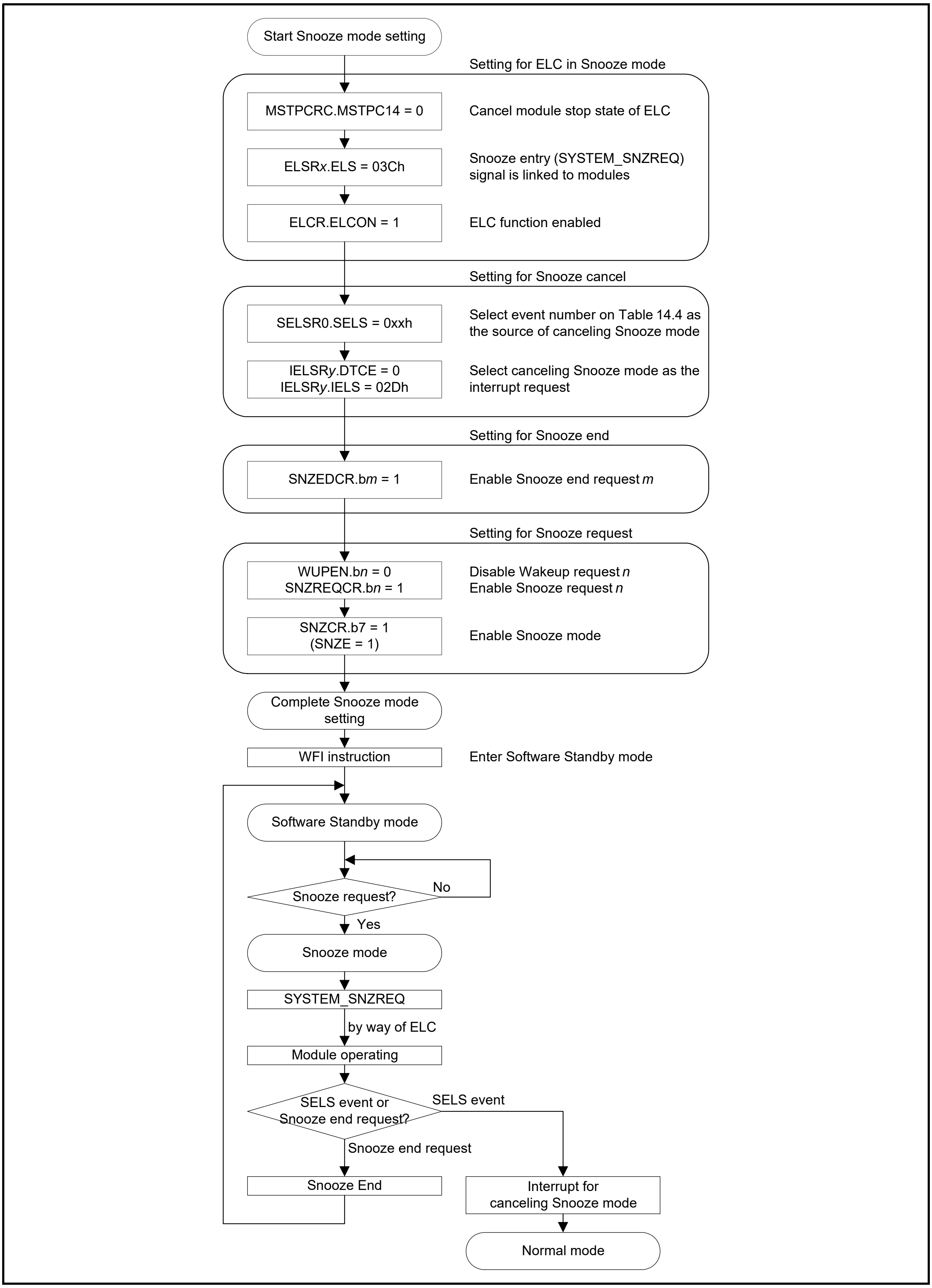


Figure 11.8 Setting example of using ELC in Snooze mode

The MCU is capable of data transmission/reception in SCI0 asynchronous mode without CPU intervention. Table 11.10 shows the maximum transfer rate of SCI0 in Snooze mode. When using the SCI0 in Snooze mode use one of the following operating modes: High-speed mode or Low-speed mode.

Do not use Subosc-speed mode.

Table 11.10 HOCO: ± 1.4% (Ta = -20 to 105°C) (Unit: bps)

Maximum division ratio of ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, BCLK, and TRCLK	HOCO frequency					
	LOCO is not operating			LOCO is operating		
	16 MHz	18 MHz	20 MHz	16 MHz	18 MHz	20 MHz
1	2400	2400	2400	4800	4800	4800
2						
4						
8						
16	1200			2400	2400	
32		1200	1200			2400
64						

Figure 11.9 shows an example setting for using SCI0 in Snooze mode entry.

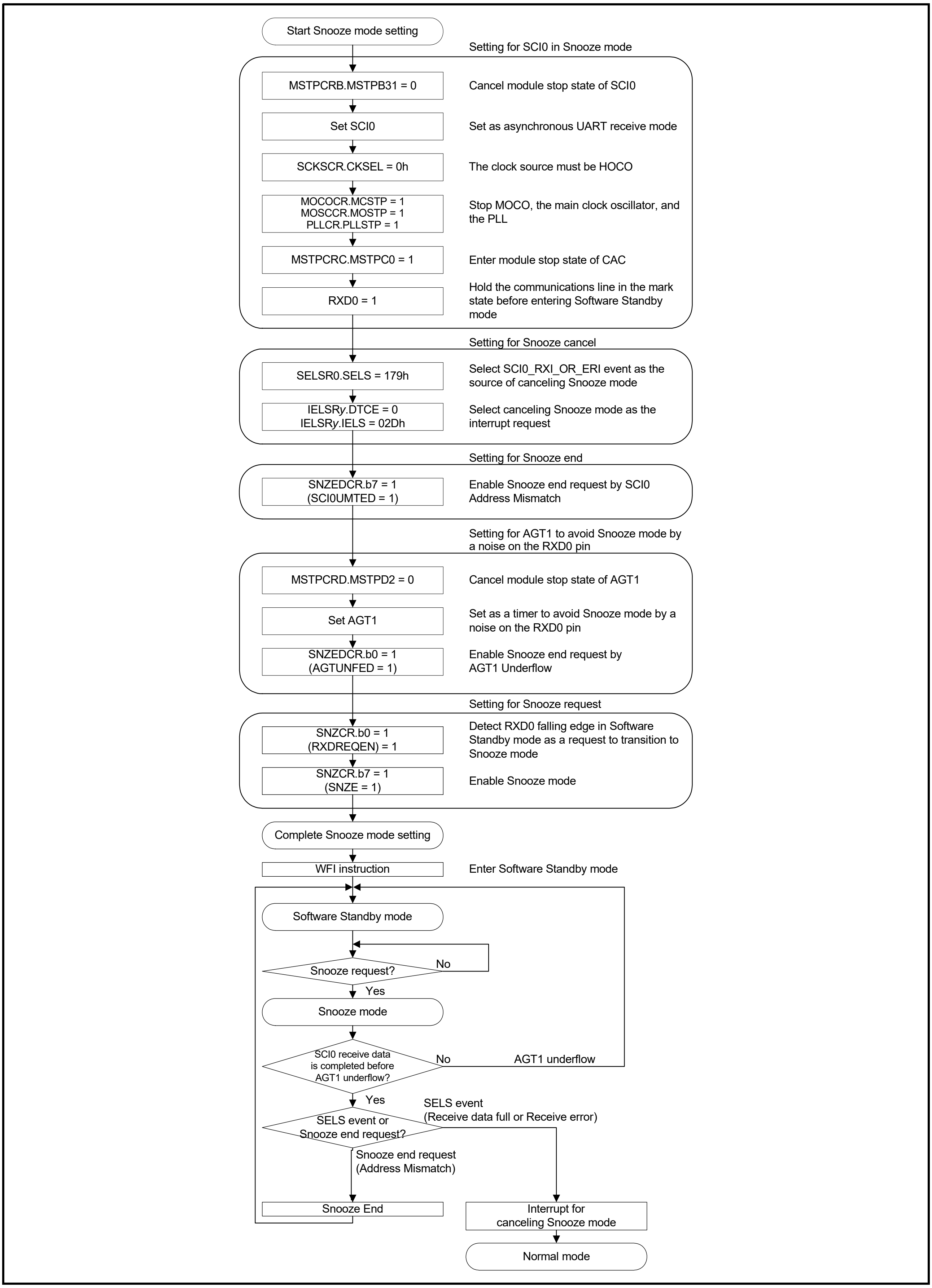


Figure 11.9 Setting example of using SCI0 in Snooze mode entry

11.9 Deep Software Standby Mode

11.9.1 Transition to Deep Software Standby Mode

The MCU enters Deep Software Standby mode when a WFI instruction is executed with the SBYCR.SSBY bit set to 1 and the DPSBYCR.DPSBY bit set to 1. See Table 11.6 for the settings of the related control bits.

In Deep Software Standby mode, the CPU, on-chip peripheral functions except for the RTC alarm, RTC interval, and USB suspend/resume detecting unit, SRAM (not the Standby SRAM), and all oscillators except for the sub-clock oscillator and low-speed on-chip oscillator are stopped. Power consumption is reduced because the internal power supply to these modules is stopped. The contents of all of the CPU registers and internal peripheral modules, except for the RTC alarm, RTC interval, and USB suspend/resume detecting unit, become undefined.

Data in the Standby SRAM is saved if the setting in the DEEPCUT[1:0] bits is 00b. If the setting in the DEEPCUT[1:0] bits is 01b, the internal power supply to the Standby SRAM and the USB resume detecting unit is cut off, and power consumption is reduced. Data in the Standby SRAM becomes undefined at this time.

If the setting in the DEEPCUT[1:0] bits is 11b, the internal power supply to the Standby SRAM and the USB resume detecting unit is cut off, the LVD is stopped, and the low power mode function of the power-on reset circuit is enabled. Therefore, power consumption is further reduced. For details, see section 60, Electrical Characteristics.

When the MCU enters Deep Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1, power supply to the IWDT-dedicated clock and the IWDT is cut off. Counting by the IWDT also stops.

When the OFS0.IWDTSTPCTL bit is 0, the MCU enters Software Standby mode instead of Deep Software Standby mode, regardless of the setting in the OFS0.IWDTSTRT or DPSBYCR.DPSBY bit. When the OFS0.IWDTSTPCTL bit is 0 while the OFS0.IWDTSTRT bit is 0 (auto start mode), the IWDT-dedicated clock and IWDT continue operation.

When the LVD1CR0.RI bit is 1 (selecting the voltage monitor 1 reset) or the LVD2CR0.RI bit is 1 (selecting the voltage monitor 2 reset), the MCU enters Software Standby mode instead of Deep Software Standby mode. The I/O port states are the same as in Software Standby mode.

When the PLL is selected as the clock source, set the following modules into the module-stop state before executing a WFI instruction: ETHERC, EPTPC, EDMAC, SCE7, DRW, JPEG, GLCDC, GPT32EH, GPT32E. In this case, you must also insert wait time at least 750 ns before executing the WFI instruction. The recommended method to measure the wait time is to do so in software. Be sure to consider the worst-case use conditions to ensure that the required wait time elapses.

Figure 11.2 shows an example flow for transitioning to Software Standby or Deep Software Standby mode.

Note 1. Conditions on the DTC, DMAC, and IWDT for transition to Software Standby mode must be met before the WFI instruction is executed. For details, see section 11.7, Software Standby Mode.

11.9.2 Canceling Deep Software Standby Mode

Deep Software Standby mode is canceled by the interrupts shown in Table 11.3, a RES pin reset, a power-on reset, or a voltage monitor 0 reset. The operations are as follows:

1. Canceling by an interrupt
   Canceling by interrupts is controlled by DPSIERn (n = 0 to 3) and DPSIFRn (n = 0 to 3). When an available interrupt request is generated, the associated flag in DPSIFRn is set to 1. If the interrupt is enabled in DPSIERn, Deep Software Standby mode is canceled. Rising or falling edge detection can be selected in DPSIEGRn (n = 0 to 2). The detection edge can be selected for the NMI, IRQ0-DS to IRQ14-DS, voltage monitor 1, and voltage monitor 2 interrupts. When a Deep Software Standby mode canceling request occurs, internal power is supplied, the MOCO clock starts to oscillate, and then an internal reset (Deep Software Standby reset) is generated for the entire MCU. The stable MOCO clock is supplied to the entire MCU and Deep Software Standby reset is canceled. The MCU starts reset exception handling.
   When Deep Software Standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

2. Canceling by RES pin reset
   When the RES pin is driven low, the MCU cancels Deep Software Standby mode and enters a reset state. Make sure to keep the RES pin low for the time period specified in section 60, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts reset exception handling.
3. Canceling by power-on reset
   Deep Software Standby mode is canceled by a power-on reset, and the MCU starts reset exception handling.
4. Canceling by voltage monitor 0 reset
   Deep Software Standby mode is canceled by a voltage monitor 0 reset from the voltage detection circuit, and the MCU starts reset exception handling.

11.9.3 Pin States when Deep Software Standby Mode is Canceled

In Deep Software Standby mode, the I/O ports retain the same states from Software Standby mode. The MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled, and reset exception handling starts immediately. The DPSBYCR.IOKEEP bit setting determines whether to initialize the I/O ports or to retain the I/O ports states for Software Standby mode. The following is the state of the I/O ports for each bit setting:

- When the DPSBYCR.IOKEEP bit = 0
  The I/O ports are initialized by an internal reset generated when Deep Software Standby mode is canceled.
- When the DPSBYCR.IOKEEP bit = 1
  Although the MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled, the I/O ports retain their states from Software Standby mode regardless of the MCU internal state. The I/O ports states remain unchanged from Software Standby mode even when settings are made to the I/O ports or peripheral modules. The retained I/O ports states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the MCU operates in accordance with the internal state. The DPSBYCR.IOKEEP bit is not initialized by any internal reset generated when Deep Software Standby mode is canceled.

11.9.4 Example of Deep Software Standby Mode Application

(1) Entering and exiting Deep Software Standby mode

Figure 11.10 shows an example transition to Deep Software Standby mode on the falling edge of the IRQn-DS pin, and an exit from Deep Software Standby mode on the rising edge of the IRQn-DS pin. In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge). After the DPSIEGRy.DIRQnEG bit (y = 0, 1 and n = 0 to 14) is set to 1 (rising edge) and the SBYCR.SSBY and DPSBYCR.DPSBY bits are both set to 1, the WFI instruction is executed. As a result, the MCU transitions to Deep Software Standby mode. Deep Software Standby mode is then canceled on the rising edge of the IRQn-DS pin.

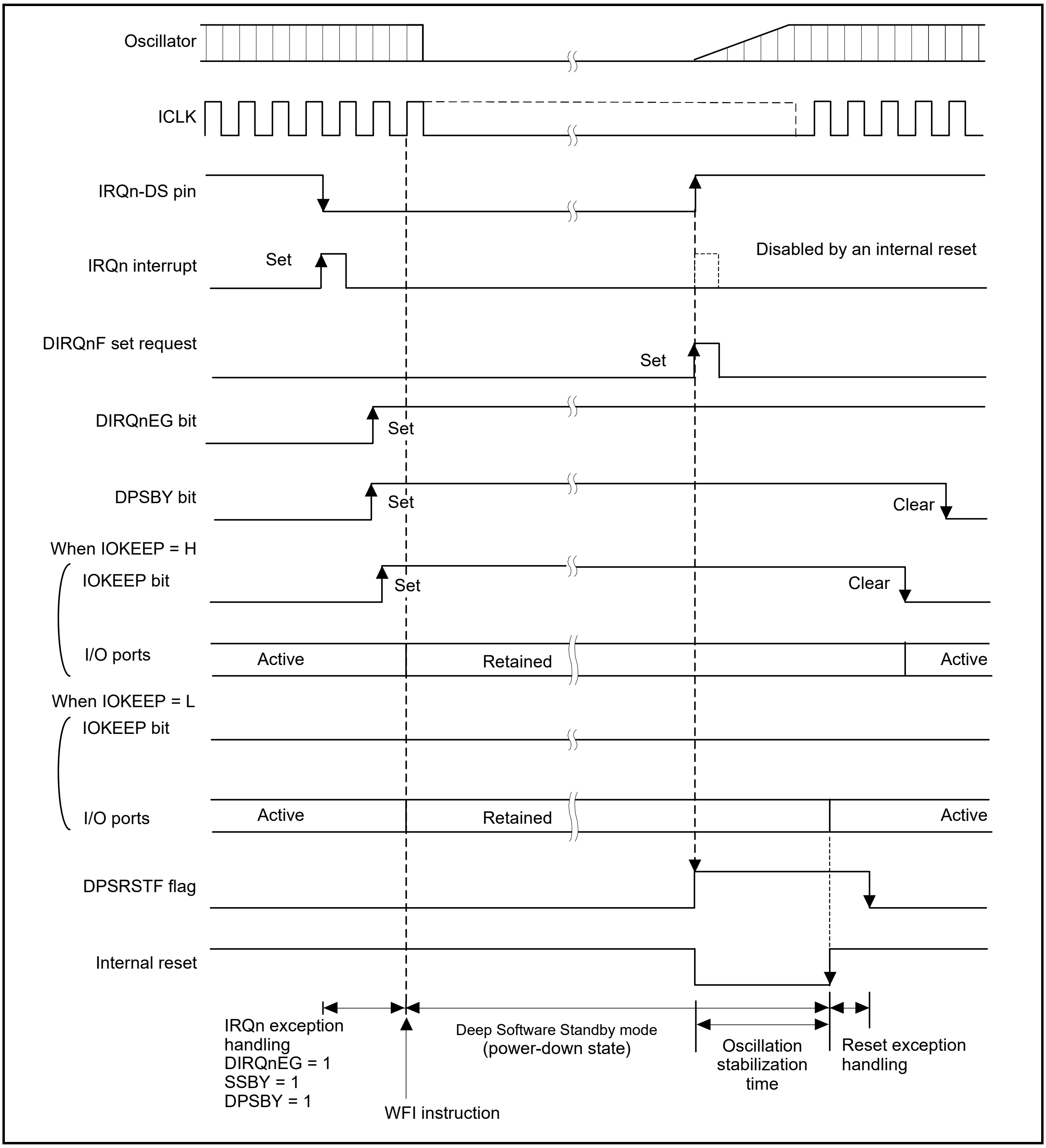


Figure 11.10 Example of Deep Software Standby mode application

11.9.5 Usage Flow for Deep Software Standby Mode

Figure 11.11 shows an example flow for using Deep Software Standby mode. In this example, the RSTSR0.DPSRSTF flag of the reset function is read after reset exception handling to determine whether the reset was generated by the RES pin or by the cancellation of Deep Software Standby mode. For a reset by the RES pin, the MCU transitions to Deep Software Standby mode after the required register settings are made. For a reset by the cancellation of Deep Software Standby mode, the DPSBYCR.IOKEEP bit clears to 0 after the I/O port settings are made.

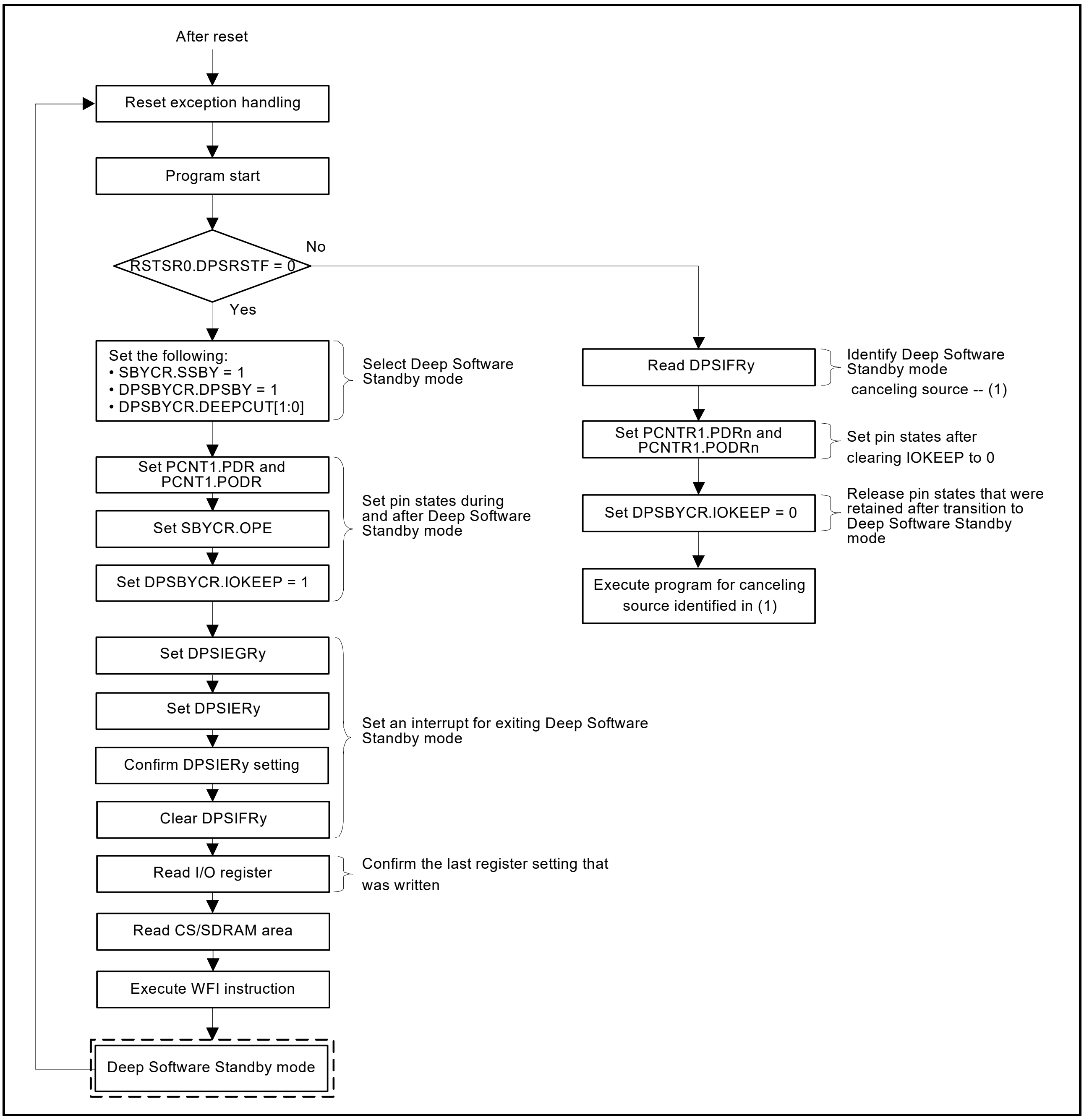


Figure 11.11 Example flow for use of Deep Software Standby mode

11.10 Usage Notes

11.10.1 Register Access

(1) Invalid register write accesses during specific modes or transitions

Do not write to registers listed in this section under any of the listed conditions.

[Registers]

- All registers with a peripheral name of “SYSTEM”.

[Conditions]

- OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of the operating power control mode)

- During time period from executing a WFI instruction to returning to Normal mode
- When FENTRYR.FENTRYi = 1 (i = 0 to 3) (flash P/E mode) or FENTRYR.FENTRYD = 1 (data flash P/E mode).

(2) Valid settings for the clock-related registers

Table 11.11 and Table 11.12 show the valid settings of the clock-related registers in each operating power control mode. Do not write any value other than the valid setting. Any other written value is ignored. Each register has certain prohibited settings under conditions other than those related to the operating power control modes. See section 9, Clock Generation Circuit, for these other conditions for each register.

Table 11.11 Valid settings for the clock-related registers (1)

Mode	Valid settings							
	SCKSCR.CKSEL[2:0], CKOCR.CKOSEL[2:0]	**SCKDIVCR.FCK[2:0], ICK[2:0]**	**PLLCR.PLLSTP**	**HOCOCR. HCSTP**	**MOCOCR. MCSTP**	**LOCOCR. LCSTP**	**MOSCCR. MOSTP**	**SOSCCR. SOSTP**
High-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (Main clock) 100b (Sub-clock) 101b (PLL)*Note:	000b (1/1) 001b (1/2) 010b (1/4) 011b (1/8) 100b (1/16) 101b (1/32) 110b (1/64)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)
Low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (Main clock) 100b (Sub-clock)		1 (stopped)					
Subosc-speed	010b (LOCO) 100b (Sub-clock)	000b (1/1)	1 (stopped)	1 (stopped)	1 (stopped)	0 (operating) 1 (stopped)	1 (stopped)	0 (operating) 1 (stopped)

Note: SCKSCR.CKSEL[2:0] only.

Table 11.12 Valid settings for the clock related registers (2)

Operating oscillator	Valid settings	
	OPCCR.OPCM[1:0]	**SOPCCR.SOPCM**
PLL	00b	0
High-speed on-chip oscillator	00b, 11b	0
Middle-speed on-chip oscillator		
Main clock oscillator		
Low-speed on-chip oscillator	00b, 11b	0, 1
Sub-clock oscillator		
IWDT-dedicated on-chip oscillator		

(3) Invalid register write accesses in subosc-speed mode

Do not write to registers listed in this section under the listed condition.

[Registers]

- SCKSCR, OPCCR.

[Condition]

- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(4) Invalid register write accesses by the DTC or DMAC

Do not write to registers listed in this section by the DTC or DMAC.

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD.

(5) Invalid register write accesses in Snooze mode

Do not write to registers listed in this section in Snooze mode. They must be set before entering Software Standby mode.

[Registers]

- SNZCR, SNZEDCR, SNZREQCR.

(6) Invalid write access to FLWT.FLWT[2:0]

Do not write any value other than 000b to the FLWT.FLWT[2:0] bits under the listed condition.

[Condition]

- SOPCCR.SOPCM = 1 (Subosc-speed mode)

(7) Invalid write access when PRCR.PRC1 is 0

Do not write to registers listed in this section when the PRCR.PRC1 bit is 0.

[Registers]

- SBYCR, SNZCR, SNZEDCR, SNZREQCR, OPCCR, SOPCCR, DPSBYCR, DPSIERn (n = 0 to 3), DPSIFRn (n = 0 to 3), DPSIEGRn (n = 0 to 2), and SYOCDCR.

11.10.2 I/O Port States

The I/O port states in Software Standby, Deep Software Standby, and Snooze modes (except when modifying in Snooze mode) are the same before entering the modes. Therefore, the power consumption is not reduced while the output signals are held high.

11.10.3 Module-Stop State of DMAC and DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0.

11.10.4 Internal Interrupt Sources

Interrupts do not operate in the module-stop state. If the module-stop bit is set when an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. Always disable the associated interrupts before setting the module-stop bits.

11.10.5 Input Buffer Control by the DIRQnE Bit (n = 0 to 14)

Setting the DPSIERy.DIRQnE bit (y = 0, 1 and n = 0 to 14) to 1 enables the associated input buffer of the IRQ0-DS to IRQ14-DS pins. Although inputs to these pins are sent to the DPSIFRy.DIRQnF bits (y = 0, 1 and n = 0 to 14), they are not sent to the ICU, peripheral modules, or I/O ports.

11.10.6 Transition to Low-Power Modes

Because the MCU does not support wakeup by events, do not enter the low power modes (Sleep, Software Standby, or Deep Software Standby mode) by executing a WFE instruction. Also, do not set the SLEEPDEEP bit of the System Control Register in the Cortex®-M4 core, because the MCU does not support low power modes by SLEEPDEEP.

11.10.7 Timing of WFI Instruction

It is possible for the WFI instruction to be executed before I/O register and CS/SDRAM area writes are complete, in which case operation might not proceed as intended. This can happen if the WFI is placed immediately after a write to an I/O register or CS/SDRAM area. To avoid this problem, read back the register or CS/SDRAM area that was written to confirm that the write completed. For example, reading MSTPCRB register before execution of WFI instruction can secure the period to complete writing to the I/O register.

11.10.8 Writing to the WDT and IWDT Registers by the DMAC or DTC in Sleep or Snooze Mode

Do not write to the WDT or IWDT registers by the DMAC or DTC while the WDT or IWDT is stopped after entering

Sleep or Snooze mode.

11.10.9 Oscillators in Snooze Mode

Oscillators that stop on entering Software Standby mode automatically restart when a trigger for switching to Snooze mode is generated. The MCU does not enter Snooze mode until all of the oscillators stabilize. In Snooze mode, you must disable oscillators that are not required in Snooze mode before entering Software Standby mode. Otherwise, the transition from Software Standby to Snooze mode takes longer.

11.10.10 Snooze Mode Entry by RXD0 Falling Edge

When the SNZCR.RXDREQEN bit is 1, noise on the RXD0 pin might cause the MCU transition from Software Standby to Snooze mode. Any subsequent RXD0 data can be received in Snooze mode by noise on the RXD0 pin. If the MCU does not receive any RXD0 data after the noise, interrupts such as SCI0_ERI or SCI0_RXI, and address mismatch events are not generated, and the MCU stays in Snooze mode. To avoid this, an AGT1 underflow interrupt must be used to return to Software Standby or Normal mode when using SCI0 in Snooze mode. However, do not use the AGT1 underflow as a source to return to Software Standby mode during an SCI communication. This makes the SCI0 stop the operation in a half-finished state.

11.10.11 Using SCI0 in Snooze Mode

When using SCI0 in Snooze mode, the AGT1 underflow must be used for the interrupt request or Snooze end request. Do not use any other trigger.

When using SCI0 in Snooze mode, the following conditions must be satisfied:

- The clock source must be HOCO
- MOCO, the main clock oscillator, and the PLL must stop before entering Software Standby mode
- The RXD0 pin must be kept at the high level before entering Software Standby mode
- A transition to Software Standby mode must not occur during an SCI communication
- The MSTPCRC.MSTPC0 bit must be 1 before entering Software Standby mode.

11.10.12 Conditions of A/D Conversion Start in Snooze Mode

The ADC12 can only be triggered by the ELC in Snooze mode. Do not use a software trigger or ADTRGn pin.

11.10.13 ELC Events in Snooze Mode

Only the ELC events listed in this section are available in Snooze mode. Do not use any other events. If starting peripheral modules for the first time after entering Snooze mode, the Event Link Setting Register (ELSRn) must set a Snooze mode entry event (SYSTEM_SNZREQ) as the trigger.

- Snooze mode entry (SYSTEM_SNZREQ)
- DTC transfer end (DTC_DTCEND)
- ADC12n Window A/B compare match (ADC12n_WCMPM) (n = 0, 1)
- ADC12n Window A/B compare mismatch (ADC12n_WCMPUM) (n = 0, 1)
- Data operation circuit interrupt (DOC_DOPCI).

11.10.14 Conditions of CTSU in Snooze Mode

The CTSU can only be started by the ELC in Snooze mode.

12. Battery Backup Function

12.1 Overview

The battery backup function maintains partial battery powering in the event of power loss. Switching between VCC and VBATT, it always maintains power to the RTC, SOSC, and backup memory. During normal operation, the battery powered area is powered by the main power supply, the VCC pin. When a VCC voltage drop is detected, the power source switches to the dedicated battery backup power pin, the VBATT pin. When the voltage rises again, the power source switches back from VBATT to VCC.

12.1.1 Features of Battery Backup Function

Battery backup features include:

- Battery power supply switch
- Backup registers
- Time capture pin detection.

12.1.2 Battery Power Supply Switch

When the voltage applied to the VCC pin drops, this feature switches the power supply from the VCC pin to the VBATT pin. When the voltage rises, it switches the power supply from the VBATT pin back to the VCC pin.

12.1.3 Backup Registers

The battery powered area provides 512 one-byte backup registers. These registers retain data only when VBATT is supplied and VCC is powered off.

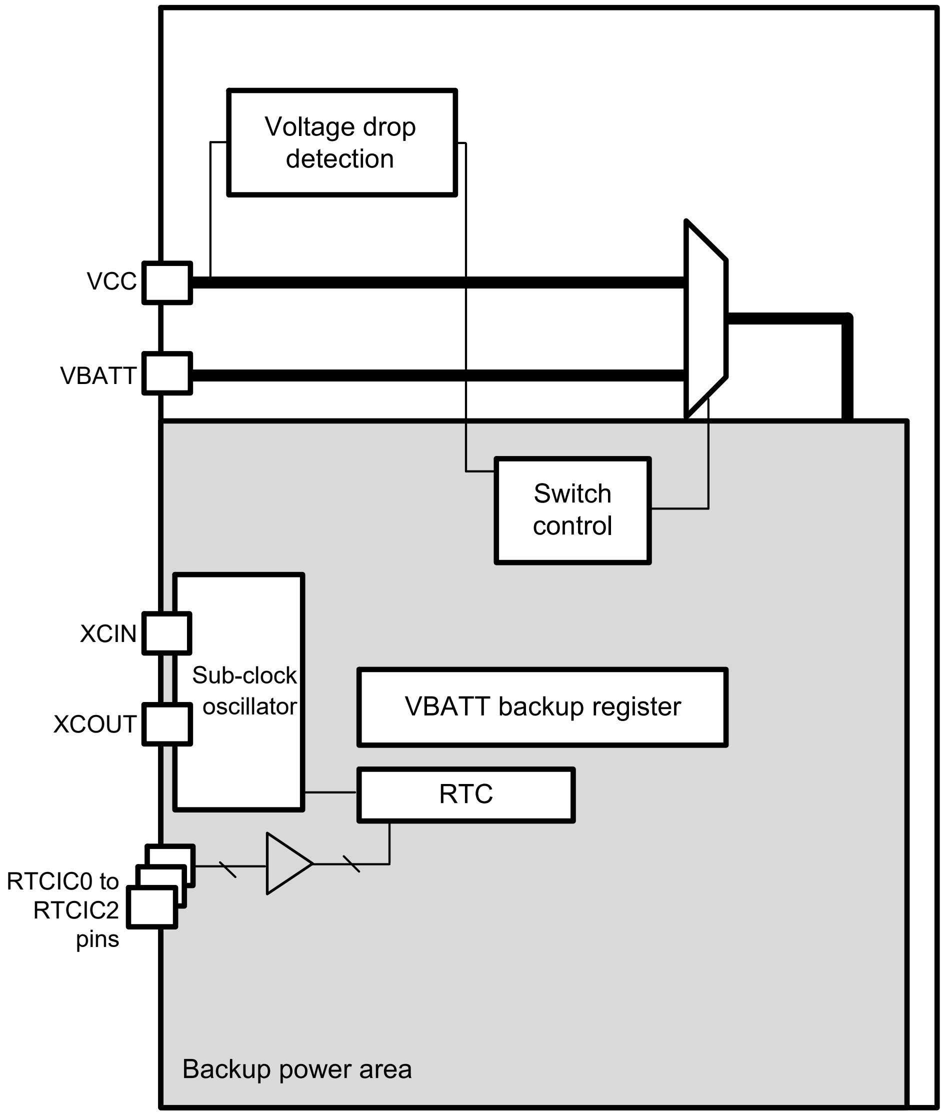
12.1.4 Time Capture Pin Detection

The RTC detects input level changes on the time capture pin. For more information, see section 26, Realtime Clock (RTC).

Note: When VCC is < $V_{DETBATT}$ and > (VBATT + 0.6 V), the injected current connects from the VCC to the VBATT pin through an internal diode. If the power supply battery connected to the VBATT pin cannot support this current injection, for example if the battery is not rechargeable, Renesas strongly recommends that you connect through a low-voltage threshold diode between the power supply battery and the VBATT pin.

Note: You must enable voltage monitor 0 resets to use the battery backup function. The voltage monitor 0 level must be higher than the VBATT switch level.

Figure 12.1 shows the configuration of the battery backup function.



VCC:	Main power supply pin
VBATT:	Battery backup power supply pin
XCIN/XCOUT:	SOSC input/output
RTCICn (n = 0 to 2):	Input port for battery backup function
LVD:	Low Voltage Detection

Figure 12.1 Configuration of the battery backup function

12.2 Register Descriptions

12.2.1 VBATT Backup Register (VBTBKRn) (n = 0 to 511)

Address(es): SYSTEM.VBTBKR[0] 4001 E500h to SYSTEM.VBTBKR[511] 4001 E6FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	VBTBKR[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

VBTBKRn is an 8-bit access read/write register for storing data powered by VBATT. The value of this register is retained when VCC is not powered and VBATT is powered. This register is not initialized by any reset.

12.2.2 VBATT Input Control Register (VBTICTLR)

Address(es): SYSTEM.VBTICTLR 4001 E4BBh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	VCH2INEN	VCH1INEN	VCH0INEN
Value after reset:	0	0	0	0	0	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	VCH0INEN	VBATT CH0 Input Enable	0: Disable 1: Enable.	R/W
b1	VCH1INEN	VBATT CH1 Input Enable	0: Disable 1: Enable.	R/W
b2	VCH2INEN	VBATT CH2 Input Enable	0: Disable 1: Enable.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTICTLR register selects the VBATT I/O direction as input.

VCHnINEN bit (VBATT CHn Input Enable Bit) (n = 0 to 2)

The VCHnINEN bit enables the input direction on the associated VBATT channel.

For more information on CH0 to CH2 corresponding function, see section 20.5.5, I/O Buffer Specification.

12.3 Operation

12.3.1 Battery Backup Function

When the voltage on the VCC pin drops, power can be supplied to the RTC and sub-clock oscillator from the VBATT pin. When a power supply drop from the VCC pin is detected, the power connection switches from the power supply to the VBATT pin. The power supply from the VCC pin is resumed when the voltage on the VCC pin exceeds $V_{DETBATT}$. This power supply change does not affect the RTC operation.

You must enable voltage monitor 0 resets to use the battery backup function. The RTC supports time capture detection, triggered by a change to the time capture pin input level.

The VBATT pin supplies power to the following modules:

- RTC
- Sub-clock oscillator (including XCIN and XCOUT pins)
- VBATT Backup Register.

Table 12.1 shows the operating states in VBATT mode.

Table 12.1 Operating states in VBATT mode (1 of 2)

Operating state	VBATT mode
Transition condition	Detection of VCC voltage drop
Canceling method other than reset	Detection of VCC voltage rise
State after cancellation by an interrupt	—
State after cancellation by a reset	—
Main clock oscillator	Stopped

Table 12.1 Operating states in VBATT mode (2 of 2)

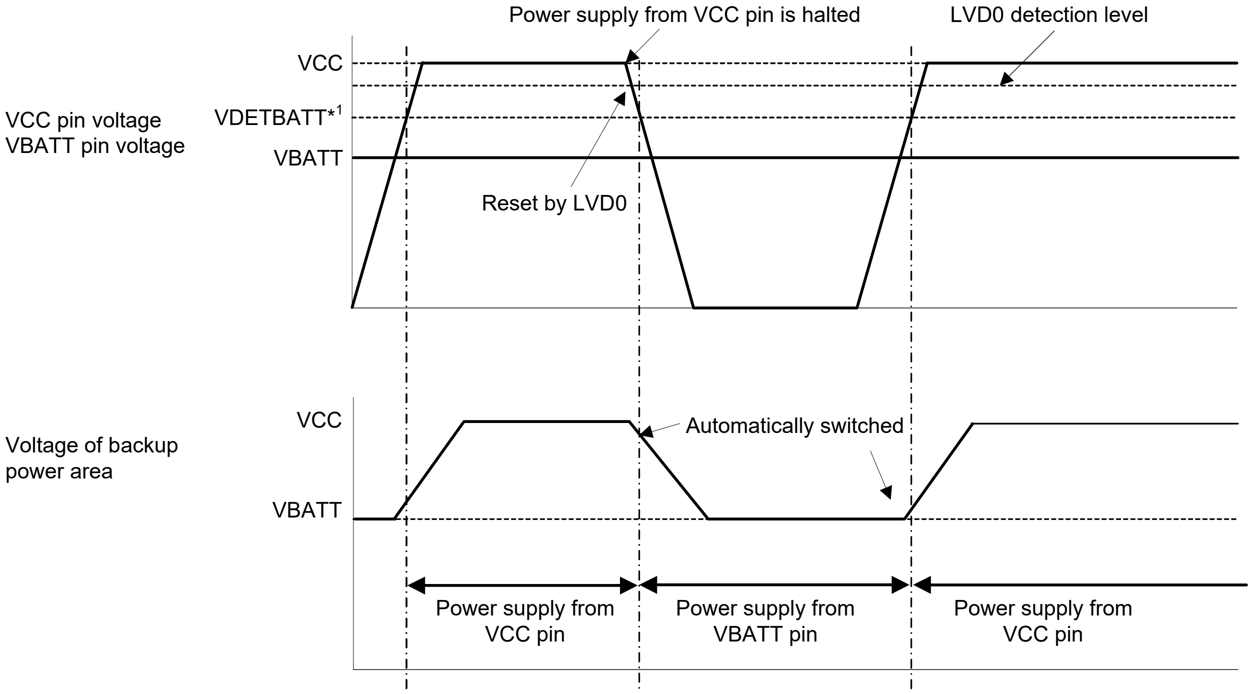
Operating state	VBATT mode
Sub-clock oscillator	Operating
High-speed on-chip oscillator	Stopped
Middle-speed on-chip oscillator	Stopped
Low-speed on-chip oscillator	Stopped
IWDT-dedicated on-chip oscillator	Stopped
PLL	Stopped
CPU	Stopped (undefined)
SRAM (ECC SRAM included)	Stopped (undefined)
Standby SRAM	Stopped (undefined)
VBATT Backup Register	Stopped (retained)
Flash memory	Stopped (retained)
Realtime Clock (RTC)	Selectable when selecting clock that serves as the count source
AGTn (n = 0, 1)	Stopped (undefined)
Low Voltage Detection (LVD)	Stopped
Power-on reset circuit	Stopped
Other peripheral modules	Stopped (undefined)
I/O ports	• RTCICn ports (n = 0 to 2): Operating • All ports not specified here: Undefined.

Note: *Selectable* means that operating or stopped is selectable in the control registers. Some modules are also controlled by the associated module-stop bit.

Note: *Stopped (retained)* means that the contents of the internal registers are retained but the operations are suspended.

Note: *Stop (undefined)* means that the contents of the internal registers are undefined and power to the internal circuit is cut off.

Figure 12.2 shows the switching sequence of the battery backup function.



Note: For details on electrical characteristics, see section 60, Electrical Characteristics.

Note 1. $V_{DETBATT}$ indicates the threshold level of the power supply change between the VCC pin and the VBATT pin.

Figure 12.2 Switching sequence for the battery backup function

12.3.2 VBATT Battery Power Supply Switch Usage

The battery power supply switch can switch the power supply from the VCC pin to the VBATT pin when the voltage being applied to the VCC pin drops. When the voltage rises, this switch changes the power supply from the VBATT pin to the VCC pin.

Note: You must enable voltage monitor 0 resets to use the battery backup function. Voltage monitor 0 level must be higher than the VBATT switch level.

12.3.3 VBATT Backup Register Usage

Use the VBATT backup registers VBTBKRn, where n = 0 to 511, to store or restore data with an 8-bit read or write operation.

12.4 Usage Notes

1. Operation of the sub-clock oscillator and RTC are not guaranteed when the voltage level on VBATT is lower than the guaranteed operation range. The RTC must be initialized to restart the power supply after the VBATT pin falls below the guaranteed operating voltage.
2. A reset generated while writing to registers described in this section might destroy the register value.
3. When VCC is higher than $V_{DETBATT}$, the VCC pin and VBATT pin are separated. When VCC is lower than $V_{DETBATT}$ and the switch is connected to the VBATT pin, and if the voltage on VBATT drops lower than (VCC - 0.6 V), current might flow into the VBATT pin through the parasitic diode between the VCC and VBATT pins.
4. During RTC operation using the voltage from the VBATT pin and the I/O ports (P402, P403 and P404) within the backup, the power supply area can only be used as time capture event input pins for the RTC.

13. Register Write Protection

13.1 Overview

The register write protection function protects important registers from being overwritten because of software errors. The registers to be protected are set with the Protect Register (PRCR). Table 13.1 lists the association between the PRCR bits and the registers to be protected.

Table 13.1 Association between PRCR bits and registers to be protected

PRCR bit	Registers to be protected
PRC0	• Registers related to the Clock Generation Circuit: SCKDIVCR, SCKDIVCR2, SCKSCR, PLLCCR, PLLCR, BCKCR, MOSCCR, HOCOCR, MOCOCR, CKOCR, TRCKCR, OSTDCR, OSTDSR, EBCKOCR, SDCKOCR, MOCOUTCR, HOCOUTCR, MOSCWTCR, MOMCR, SOSCCR, SOMCR, LOCOCR, LOCOUTCR, HOCOWTCR, FLLCR1, FLLCR2
PRC1	• Registers related to the low power modes: SBYCR, SNZCR, SNZEDCR, SNZREQCR, OPCCR, SOPCCR, DPSBYCR, DPSIER0-3, DPSIFR0-3, DPSIEGR0-2, SYOCDCR, STCONR • Registers related to the battery backup function: VBTBKRn (n = 0 to 511), VBTICTLR
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVCMPCR, LVDLVLR, LVD1CR0, LVD2CR0

13.2 Register Descriptions

13.2.1 Protect Register (PRCR)

Address(es): SYSTEM.PRCR 4001 E3FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Function	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the Clock Generation Circuit: 0: Disable writes 1: Enable writes.	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to the low power modes and the battery backup function: 0: Disable writes 1: Enable writes.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD: 0: Disable writes 1: Enable writes.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control write access to the PRCR register. To modify the PRCR register, write A5h to the eight higher-order bits and the wanted value to the eight lower-order bits as a 16-bit unit.	W*1

Note 1. Write data is not retained. Always reads 00h.

PRCn bits (Protect Bit n) (n = 0, 1, 3)

The PRCn bits enable or disable writing to the protected registers listed in Table 13.1. Setting PRCn to 1 or 0 enables or disables writing, respectively.

14. Interrupt Controller Unit (ICU)

14.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the NVIC, DTC, and DMAC modules. The ICU also controls non-maskable interrupts. Table 14.1 lists the ICU specifications, Figure 14.1 shows a block diagram, and Table 14.2 lists the I/O pins.

Table 14.1 ICU specifications

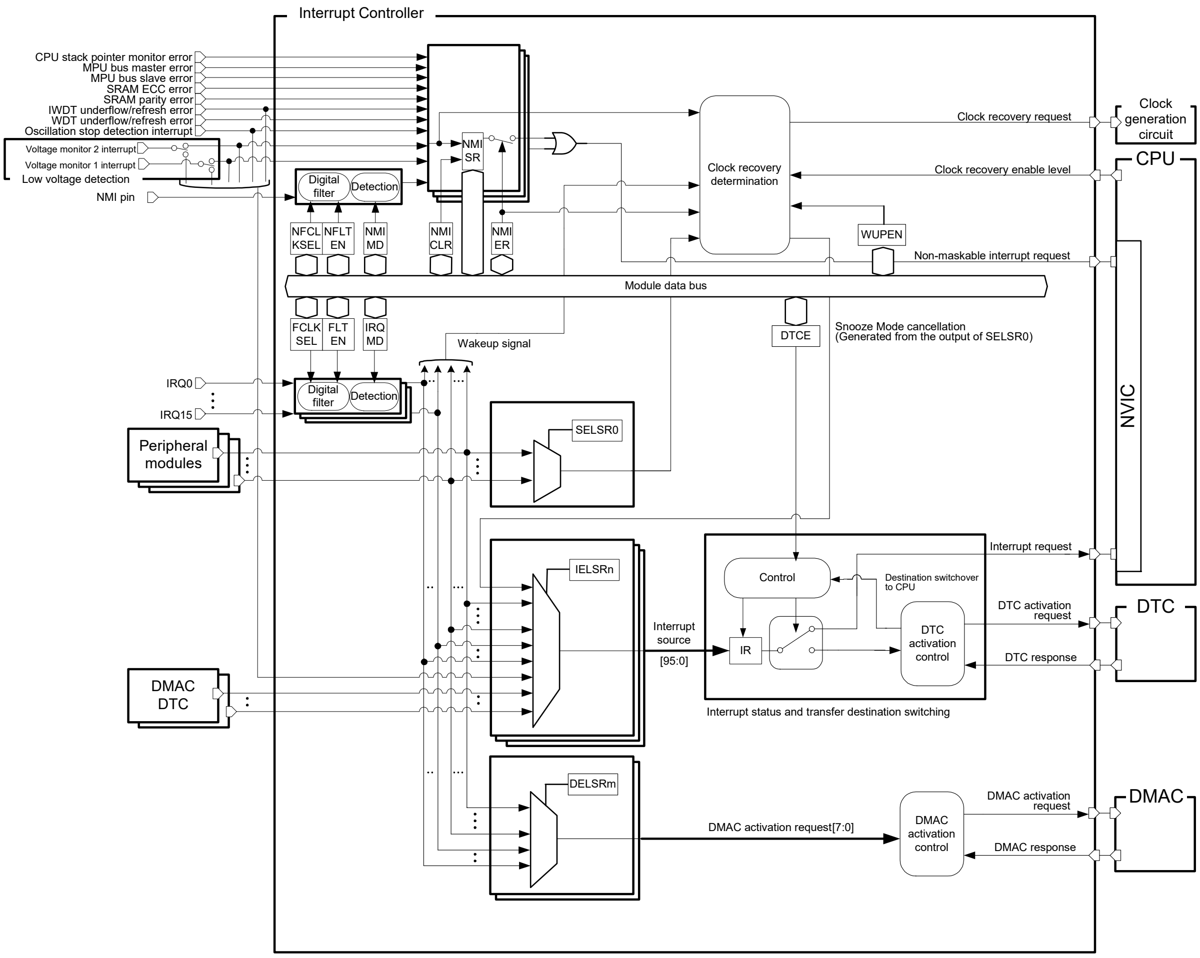
Parameter		Specifications
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 315 (select factors within event list numbers 64 to 511)
	External pin interrupts	• Interrupt detection on low level, falling edge, rising edge, rising and falling edges One of these detection methods can be set for each source. • Digital filter function supported • 16 sources, with interrupts from IRQ0 to IRQ15 pins.
	DTC and DMAC control	The DTC and DMAC can be activated using interrupt sources[*1]
	Interrupt sources for NVIC	96 sources
Non-maskable interrupts[*2]	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported.
	Oscillation stop detection interrupt[*3]	Interrupt on detecting that the main oscillator has stopped
	WDT underflow/refresh error[*3]	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error[*3]	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Voltage monitor 1 interrupt[*3]	Voltage monitor interrupt of Low Voltage Detection detector 1 (LVD1)
	Voltage monitor 2 interrupt[*3]	Voltage monitor interrupt of Low Voltage Detection detector 2 (LVD2)
	RPEST	Interrupt on SRAM parity error
	RECCST	Interrupt on SRAM ECC error
	BUSSST	Interrupt on MPU bus slave error
	BUSMST	Interrupt on MPU bus master error
	SPEST	Interrupt on CPU stack pointer monitor
Return from low power mode[*4]		• Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source • Software Standby mode: Return is initiated by non-maskable interrupts Interrupts can be selected in the WUPEN register. • Snooze mode: Return is initiated by non-maskable interrupts Interrupts can be selected in the SELSR0 and WUPEN registers. See section 14.2.8, SYS Event Link Setting Register (SELSR0), and section 14.2.9, Wake Up Interrupt Enable Register (WUPEN).

Note 1. For the DTC and DMAC activation sources, see Table 14.4.

Note 2. Non-maskable interrupts can be enabled only once after a reset release.

Note 3. These non-maskable interrupts can also be used as event signals. When used as interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 and 2 interrupts, set the LVD1CR1.IRQSEL and LVD2CR1.IRQSEL bits to 1.

Note 4. For return from Deep Software Standby mode, see section 11.9, Deep Software Standby Mode.



NMISR: Non-Maskable Interrupt Status Register
NMIER: Non-Maskable Interrupt Enable Register
NMICLR: Non-Maskable Interrupt Status Clear Register
NMIMD: NMI Detection Set (NMICR.NMIMD)
NFCLKSEL: NMI Digital Filter Sampling Clock Select (NMICR.NFCLKSEL)
NFLTEN: NMI Digital Filter Enable (NMICR.NFLTEN)
IRQMD: IRQi Detection Sense Select (IRQCRi.IRQMD (i = 0 to 15))
FCLKSEL: IRQi Digital Filter Sampling Clock Select (IRQCRi.FCLKSEL (i = 0 to 15))
FLTEN: IRQi Digital Filter Enable (IRQCRi.FLTEN (i = 0 to 15))
SELSR0: SYS Event Link Setting Register 0
WUPEN: Wake Up Interrupt Enable Register
IELSRn: ICU Event Link Setting Register (n = 0 to 95)
IR: Interrupt Status Flag (IELSRn.IR)
DTCE: DTC Activation Enable (IELSRn.DTCE)
DELSRm: DMAC Event Link Setting Register (m = 0 to 7)

Figure 14.1 ICU block diagram

Table 14.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ15	Input	External interrupt request pins

14.2 Register Descriptions

This chapter does not describe the Arm® NVIC internal registers. For information on these registers, see the *ARM® Cortex®-M4 Processor Technical Reference Manual* (ARM DDI 0439D).

14.2.1 IRQ Control Register i (IRQCRi) (i = 0 to 15)

Address(es): ICU.IRQCR0 4000 6000h, ICU.IRQCR1 4000 6001h, ICU.IRQCR2 4000 6002h, ICU.IRQCR3 4000 6003h, ICU.IRQCR4 4000 6004h, ICU.IRQCR5 4000 6005h, ICU.IRQCR6 4000 6006h, ICU.IRQCR7 4000 6007h ICU.IRQCR8 4000 6008h, ICU.IRQCR9 4000 6009h, ICU.IRQCR10 4000 600Ah, ICU.IRQCR11 4000 600Bh, ICU.IRQCR12 4000 600Ch, ICU.IRQCR13 4000 600Dh, ICU.IRQCR14 4000 600Eh, ICU.IRQCR15 4000 600Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN	—	FCLKSEL[1:0]		—	—	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	IRQMD[1:0]	IRQi Detection Sense Select	b1 b0 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	FCLKSEL[1:0]	IRQi Digital Filter Sampling Clock Select	b5 b4 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	FLTEN	IRQi Digital Filter Enable	0: Disable 1: Enable.	R/W

IRQCRi register changes must satisfy the following conditions:

- For a CPU interrupt or DTC trigger:
  Change the IRQCRi register setting before setting the target IELSRn register (n = 0 to 95).
  You can change the register values only when the IELSRn.IELS[8:0] bits are 000h.
- For a DMAC trigger:
  Change the IRQCRi register setting before setting the target DELSRn register (n = 0 to 7).
  You can change the register values only when the DELSRn.DELR[8:0] bits are 000h.
- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN.IRQWUPENn bit (n = 0 to 15).
  You can only change the register values when the target WUPEN.IRQWUPENn bit is 0.

IRQMD[1:0] bits (IRQi Detection Sense Select)

The IRQMD[1:0] bits set the detection sensing method for the IRQi external pin interrupt sources. For more information on the settings, see section 14.4.4, External Pin Interrupts.

FCLKSEL[1:0] bits (IRQi Digital Filter Sampling Clock Select)

The FCLKSEL[1:0] bits select the digital filter sampling clock for the IRQi external pin interrupt requests, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every eight cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles).

For details on the digital filter, see section 14.4.3, Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the IRQi external pin interrupt sources. The filter is enabled when FLTEN is 1 and disabled when FLTEN is 0. The IRQi pin level is sampled at the cycle specified in FCLKSEL[1:0].

When the sampled level matches three times, the output level from the digital filter changes. For details on the digital filter, see section 14.4.3, Digital Filter.

14.2.2 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 4000 6140h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPEST	BUSMST	BUSSST	RECCST	RPEST	NMIST	OSTST	—	—	LVD2ST	LVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b1	WDTST	WDT Underflow/Refresh Error Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b2	LVD1ST	Voltage Monitor 1 Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b3	LVD2ST	Voltage Monitor 2 Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b5, b4	—	Reserved	These bits are read as 0.	R
b6	OSTST	Main Oscillation Stop Detection Interrupt Status Flag	0: Interrupt not requested for main oscillation stop 1: Interrupt requested for main oscillation stop.	R
b7	NMIST	NMI Status Flag	0: NMI pin interrupt not requested 1: NMI pin interrupt requested.	R
b8	RPEST	SRAM Parity Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b9	RECCST	SRAM ECC Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b10	BUSSST	MPU Bus Slave Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b11	BUSMST	MPU Bus Master Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b12	SPEST	CPU Stack Pointer Monitor Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b15 to b13	—	Reserved	These bits are read as 0.	R

The NMISR register monitors the status of non-maskable interrupt sources. Writes to the NMISR register are ignored. The setting in the Non-Maskable Interrupt Enable Register (NMIER) does not affect the status flags in this register. Before the end of the non-maskable interrupt handler, check that all of the bits in this register are set to 0 to confirm that no other NMI requests have occurred during handler processing.

IWDTST flag (IWDT Underflow/Refresh Error Status Flag)

The IWDTST flag indicates an IWDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.IWDTCLR bit.

[Setting condition]

When an IWDT underflow/refresh error interrupt occurs and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.IWDTCLR bit.

WDTST flag (WDT Underflow/Refresh Error Status Flag)

The WDTST flag indicates a WDT underflow/refresh error interrupt request. It is read-only and cleared by the

NMICLR.WDTCLR bit.

[Setting condition]

When a WDT underflow/refresh error interrupt occurs.

[Clearing condition]

When 1 is written to the NMICLR.WDTCLR bit.

LVD1ST flag (Voltage Monitor 1 Interrupt Status Flag)

The LVD1ST flag indicates a request for voltage monitor 1 interrupt. It is read-only and cleared by the NMICLR.LVD1CLR bit.

[Setting condition]

When a voltage monitor 1 interrupt occurs and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.LVD1CLR bit.

LVD2ST flag (Voltage Monitor 2 Interrupt Status Flag)

The LVD2ST flag indicates a request for voltage monitor 2 interrupt. It is read-only and cleared by the NMICLR.LVD2CLR bit.

[Setting condition]

When a voltage monitor 2 interrupt occurs and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.LVD2CLR bit.

OSTST flag (Main Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates a main oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit.

[Setting condition]

When an oscillation stop detection interrupt occurs.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI Status Flag)

The NMIST flag indicates an NMI pin interrupt request. It is read-only and cleared by the NMICLR.NMICLR bit.

[Setting condition]

When an edge specified in the NMICR.NMIMD bit is input to the NMI pin.

[Clearing condition]

When 1 is written to the NMICLR.NMICLR bit.

RPEST flag (SRAM Parity Error Interrupt Status Flag)

The RPEST flag indicates an SRAM parity error interrupt request.

[Setting condition]

When an interrupt occurs in response to an SRAM parity error.

[Clearing condition]

When 1 is written to the NMICLR.RPECLR bit.

RECCST flag (SRAM ECC Error Interrupt Status Flag)

The RECCST flag indicates an SRAM ECC error interrupt request.

[Setting condition]

When an interrupt occurs in response to an SRAM ECC error.

[Clearing condition]

When 1 is written to the NMICLR.RECCCLR bit.

BUSSST flag (MPU Bus Slave Error Interrupt Status Flag)

The BUSSST flag indicates a bus slave error interrupt request.

[Setting condition]

When an interrupt occurs in response to a bus slave error.

[Clearing condition]

When 1 is written to the NMICLR.BUSSCLR bit.

BUSMST flag (MPU Bus Master Error Interrupt Status Flag)

The BUSMST flag indicates a bus master error interrupt request.

[Setting condition]

When an interrupt occurs in response to a bus master error.

[Clearing condition]

When 1 is written to the NMICLR.BUSMCLR bit.

SPEST flag (CPU Stack Pointer Monitor Interrupt Status Flag)

The SPEST flag indicates a CPU stack pointer monitor interrupt request.

[Setting condition]

When an interrupt occurs in response to a CPU stack pointer monitor error.

[Clearing condition]

When 1 is written to the NMICLR.SPECLR bit.

14.2.3 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 4000 6120h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPEEN	BUSMEN	BUSSEN	RECCEN	RPEEN	NMIEN	OSTEN	—	—	LVD2EN	LVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b2	LVD1EN	Voltage Monitor 1 Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b3	LVD2EN	Voltage Monitor 2 Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b5 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Disable for main oscillation 1: Enable for main oscillation.	R/(W) *1, *2
b7	NMIEN	NMI Pin Interrupt Enable	0: Disable 1: Enable.	R/(W) *1
b8	RPEEN	SRAM Parity Error Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b9	RECCEN	SRAM ECC Error Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b10	BUSSEN	MPU Bus Slave Error Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b11	BUSMEN	MPU Bus Master Error Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b12	SPEEN	CPU Stack Pointer Monitor Interrupt Enable	0: Disable 1: Enable.	R/(W) *1, *2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. A 1 can be written to this bit only once after reset, and subsequent write accesses are invalid. Writing 0 to this bit is invalid.
Note 2. Do not write 1 to this bit when the source is used as an event signal.

IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT underflow/refresh error interrupts as an NMI trigger.

WDTEN bit (WDT Underflow/Refresh Error Interrupt Enable)

The WDTEN bit enables WDT underflow/refresh error interrupts as an NMI trigger.

LVD1EN bit (Voltage Monitor 1 Interrupt Enable)

The LVD1EN bit enables voltage monitor 1 interrupts as an NMI trigger.

LVD2EN bit (Voltage Monitor 2 Interrupt Enable)

The LVD2EN bit enables voltage monitor 2 interrupts as an NMI trigger.

OSTEN bit (Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main oscillation stop detection interrupts as an NMI trigger.

NMIEN bit (NMI Pin Interrupt Enable)

The NMIEN bit enables NMI pin interrupts as an NMI trigger.

RPEEN bit (SRAM Parity Error Interrupt Enable)

The RPEEN bit enables SRAM parity error interrupts as an NMI trigger.

RECCEN bit (SRAM ECC Error Interrupt Enable)

The RECCEN bit enables SRAM ECC error interrupts as an NMI trigger.

BUSSEN bit (MPU Bus Slave Error Interrupt Enable)

The BUSSEN bit enables bus slave error interrupts as an NMI trigger.

BUSMEN bit (MPU Bus Master Error Interrupt Enable)

The BUSMEN bit enables bus master error interrupts as an NMI trigger.

SPEEN bit (CPU Stack Pointer Monitor Interrupt Enable)

The SPEEN bit enables CPU stack pointer monitor interrupts as an NMI trigger.

14.2.4 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): ICU.NMICLR 4000 6130h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPECLR	BUSMCLR	BUSSCLR	RECCCLR	RPECLR	NMICLR	OSTCLR	—	—	LVD2CLR	LVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IWDTCLR	IWDT Clear	0: No effect 1: Clear the NMISR.IWDTST flag.	R/(W)*1
b1	WDTCLR	WDT Clear	0: No effect 1: Clear the NMISR.WDTST flag.	R/(W)*1
b2	LVD1CLR	LVD1 Clear	0: No effect 1: Clear the NMISR.LVD1ST flag.	R/(W)*1
b3	LVD2CLR	LVD2 Clear	0: No effect 1: Clear the NMISR.LVD2ST flag.	R/(W)*1
b5 to b4	—	Reserved	The write value should be 0.	R/(W)*1
b6	OSTCLR	OST Clear	0: No effect 1: Clear the NMISR.OSTST flag.	R/(W)*1
b7	NMICLR	NMI Clear	0: No effect 1: Clear the NMISR.NMIST flag.	R/(W)*1
b8	RPECLR	SRAM Parity Error Clear	0: No effect 1: Clear the NMISR.RPEST flag.	R/(W)*1
b9	RECCCLR	SRAM ECC Error Clear	0: No effect 1: Clear the NMISR.RECCST flag.	R/(W)*1
b10	BUSSCLR	Bus Slave Error Clear	0: No effect 1: Clear the NMISR.BUSSST flag.	R/(W)*1
b11	BUSMCLR	Bus Master Error Clear	0: No effect 1: Clear the NMISR.BUSMST flag.	R/(W)*1
b12	SPECLR	CPU Stack Pointer Monitor Interrupt Clear	0: No effect 1: Clear the NMISR.SPEST flag.	R/(W)*1
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/(W)*1

Note 1. Only 1 can be written to this bit.

IWDTCLR bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

WDTCLR bit (WDT Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

LVD1CLR bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

LVD2CLR bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

OSTCLR bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

NMICLR bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

RPECLR bit (SRAM Parity Error Clear)

Writing 1 to the RPECLR bit clears the NMISR.RPEST flag. This bit is read as 0.

RECCCLR bit (SRAM ECC Error Clear)

Writing 1 to the RECCCLR bit clears the NMISR.RECCST flag. This bit is read as 0.

BUSSCLR bit (Bus Slave Error Clear)

Writing 1 to the BUSSCLR bit clears the NMISR.BUSSST flag. This bit is read as 0.

BUSMCLR bit (Bus Master Error Clear)

Writing 1 to the BUSMCLR bit clears the NMISR.BUSMSST flag. This bit is read as 0.

SPECLR bit (CPU Stack Pointer Monitor Interrupt Clear)

Writing 1 to the SPECLR bit clears the NMISR.SPEST flag. This bit is read as 0.

14.2.5 NMI Pin Interrupt Control Register (NMICR)

Address(es): ICU.NMICR 4000 6100h

	b7	b6	b5	b4	b3	b2	b1	b0
	NFLTEN	—	NFCLKSEL[1:0]		—	—	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select	b5 b4 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	NFLTEN	NMI Digital Filter Enable	0: Disable 1: Enable.	R/W

Change the NMICR register settings before enabling NMI pin interrupts (before setting NMIER.NMIEN to 1).

NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for NMI pin interrupts.

NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

The NFCLKSEL[1:0] bits select the digital filter sampling clock for NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every eight cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles).

For details on the digital filter, see section 14.4.3, Digital Filter.

NFLTEN bit (NMI Digital Filter Enable)

The NFLTEN bit enables the digital filter used for NMI pin interrupts. The filter is enabled when NFLTEN is 1 and disabled when NFLTEN is 0. The NMI pin level is sampled at the cycle specified in NMIFLTC.NFCLKSEL[1:0]. When the sampled level matches three times, the output level from the digital filter changes. For details on the digital filter, see section 14.4.3, Digital Filter.

14.2.6 ICU Event Link Setting Register n (IELSRn) (n = 0 to 95)

Address(es): ICU.IELSR0 4000 6300h, ICU.IELSR1 4000 6304h, ICU.IELSR2 4000 6308h, ICU.IELSR3 4000 630Ch……
……ICU.IELSR92 4000 6470h, ICU.IELSR93 4000 6474h, ICU.IELSR94 4000 6478h, ICU.IELSR95 4000 647Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	DTCE	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	IELS[8:0]	ICU Event Link Select	b8 b0 000000000:Disable interrupts to the associated NVIC or DTC module 000000001 to 111111111: Event signal number to be linked. For details, see Table 14.4.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	IR	Interrupt Status Flag	0: No interrupt request occurred 1: Interrupt request occurred.	R/(W)*2
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DTCE	DTC Activation Enable	0: Disable 1: Enable.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register requires halfword or word access.
Note 2. Writing 1 to the IR flag is prohibited.

The IELSRn register selects the IRQ source used by the NVIC. For details, see Table 14.4. IELSRn, where n = 0 to 95, corresponds to the NVIC IRQ input source numbers 0 to 95.

IELS[8:0] bits (ICU Event Link Select)

The IELS[8:0] bits link an event signal to the associated NVIC or DTC module. All IELS[8:0] bits must be written to simultaneously.

IR flag (Interrupt Status Flag)

The IR flag indicates an individual interrupt request from the event specified in IELS[8:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing conditions]

When 0 is written to the bit. DTCE must be set to 0 before writing 0 to the IR flag.

To clear the IR flag:

1. Negate the input interrupt signal.
2. Read access the peripheral once and wait for 2 clock cycles of the target module clock.
3. Clear the IR flag by writing 0.

DTCE bit (DTC Activation Enable)

When the DTCE bit is set to 1, the associated event is selected as the source for DTC activation.

[Setting condition]

When 1 is written to the DTCE bit.

[Clearing conditions]

- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.
- When 0 is written to the bit.

14.2.7 DMAC Event Link Setting Register n (DELSRn) (n = 0 to 7)

Address(es): ICU.DELSR0 4000 6280h, ICU.DELSR1 4000 6284h, ICU.DELSR2 4000 6288h, ICU.DELSR3 4000 628Ch, ICU.DELSR4 4000 6290h, ICU.DELSR5 4000 6294h, ICU.DELSR6 4000 6298h, ICU.DELSR7 4000 629Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	DELS[8:0]	DMAC Event Link Select	b8 b0 000000000:Disable DMA start requests to the associated DMAC module 000000001 to 111111111: Event signal number to be linked. For details, see Table 14.4.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	IR	Interrupt Status Flag for DMAC	0: No interrupt request is generated 1: An interrupt request is generated.	R/(W)*2
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register requires halfword or word access.
Note 2. Writing 1 to the IR flag is prohibited.

DELS[8:0] bits (DMAC Event Link Select)

The DELS[8:0] bits link an event signal to the DMAC module. All DELS[8:0] bits must be written to simultaneously.

IR flag (Interrupt Status Flag for DMAC)

This is the status flag of an individual DMA transfer request. This corresponds to DELS[8:0] bits of the same register.

[Setting condition]

The flag is set to 1 when a DMA transfer request is generated from the corresponding peripheral module or IRQi pin.

[Clearing conditions]

- When 0 is written to the flag.
- At the start of the DMA transfer after the DMA transfer request is issued.

14.2.8 SYS Event Link Setting Register (SELSR0)

Address(es): ICU.SELSR0 4000 6200h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	SELS[8:0]	SYS Event Link Select	b8 b0 000000000: Disable event output to the associated low power mode module 000000001 to 111111111: Event signal number to be linked. For details, see Table 14.4.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register requires halfword access.

The SELSR0 register selects the events that wake the CPU from Snooze mode. You can only use the events listed in Table 14.4 checked under "Canceling Snooze using SELSR0". Events specified in this register are defined as ICU_SNZCANCEL (02Dh) in Table 14.4. When 02Dh is set in IELSRn.IELS, the SELSR0 event interrupt occurs.

SELS[8:0] bits (SYS Event Link Select)

All SELS[8:0] bits must be written to simultaneously.

14.2.9 Wake Up Interrupt Enable Register (WUPEN)

Address(es): ICU.WUPEN 4000 61A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IIC0WUPEN	AGT1CBWUPEN	AGT1CAWUPEN	AGT1UDWUPEN	USBFSWUPEN	USBHSWUPEN	RTCPRDWUPEN	RTCALMWUPEN	—	ACMPHS0WUPEN	—	—	LVD2WUPEN	LVD1WUPEN	KEYWUPEN	IWDTWUPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IRQWUPEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	IRQWUPEN[15:0]	IRQ Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by IRQ interrupts 1: Enable Software Standby returns by IRQ interrupts.	R/W
b16	IWDTWUPEN	IWDT Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by IWDT interrupts 1: Enable Software Standby returns by IWDT interrupts.	R/W
b17	KEYWUPEN	Key Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by KEY interrupts 1: Enable Software Standby returns by KEY interrupts.	R/W
b18	LVD1WUPEN	LVD1 Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by LVD1 interrupts 1: Enable Software Standby returns by LVD1 interrupts.	R/W
b19	LVD2WUPEN	LVD2 Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by LVD2 interrupts 1: Enable Software Standby returns by LVD2 interrupts.	R/W
b21 to b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b22	ACMPHS0WUPEN	ACMPHS0 Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by ACMPHS0 interrupts 1: Enable Software Standby returns by ACMPHS0 interrupts.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b24	RTCALMWUPEN	RTC Alarm Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by RTC alarm interrupts 1: Enable Software Standby returns by RTC alarm interrupts.	R/W
b25	RTCPRDWUPEN	RTC Period Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by RTC period interrupts 1: Enable Software Standby returns by RTC period interrupts.	R/W
b26	USBHSWUPEN	USBHS Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by USBHS interrupts 1: Enable Software Standby returns by USBHS interrupts.	R/W
b27	USBFSWUPEN	USBFS Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by USBFS interrupts 1: Enable Software Standby returns by USBFS interrupts.	R/W
b28	AGT1UDWUPEN	AGT1 Underflow Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by AGT1 underflow interrupts 1: Enable Software Standby returns by AGT1 underflow interrupts.	R/W
b29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by AGT1 compare match A interrupts 1: Enable Software Standby returns by AGT1 compare match A interrupts.	R/W
b30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by AGT1 compare match B interrupts 1: Enable Software Standby returns by AGT1 compare match B interrupts.	R/W
b31	IIC0WUPEN	IIC0 Address Match Interrupt Software Standby Returns Enable	0: Disable Software Standby returns by IIC0 address match interrupts 1: Enable Software Standby returns by IIC0 address match interrupts.	R/W

The bits in this register control whether the associated interrupt can wake the CPU from Software Standby mode.

IRQWUPEN[15:0] bits (IRQ Interrupt Software Standby Returns Enable)

The IRQWUPEN[15:0] bits enable the use of IRQn interrupts to cancel Software Standby mode.

IWDTWUPEN bit (IWDT Interrupt Software Standby Returns Enable)

The IWDTWUPEN bit enables the use of IWDT interrupts to cancel Software Standby mode.

KEYWUPEN bit (Key Interrupt Software Standby Returns Enable)

The KEYWUPEN bit enables the use of key interrupts to cancel Software Standby mode.

LVD1WUPEN bit (LVD1 Interrupt Software Standby Returns Enable)

The LVD1WUPEN bit enables the use of LVD1 interrupts to cancel Software Standby mode.

LVD2WUPEN bit (LVD2 Interrupt Software Standby Returns Enable)

The LVD2WUPEN bit enables the use of LVD2 interrupts to cancel Software Standby mode.

ACMPHS0WUPEN bit (ACMPHS0 Interrupt Software Standby Returns Enable)

The ACMPHS0WUPEN bit enables the use of ACMPHS0 interrupts to cancel Software Standby mode.

RTCALMWUPEN bit (RTC Alarm Interrupt Software Standby Returns Enable)

The RTCALMWUPEN bit enables the use of RTC alarm interrupts to cancel Software Standby mode.

RTCPRDWUPEN bit (RTC Period Interrupt Software Standby Returns Enable)

The RTCPRDWUPEN bit enables the use of RTC period interrupts to cancel Software Standby mode.

USBHSWUPEN bit (USBHS Interrupt Software Standby Returns Enable)

The USBHSWUPEN bit enables the use of USBHS interrupts to cancel Software Standby mode.

USBFSWUPEN bit (USBFS Interrupt Software Standby Returns Enable)

The USBFSWUPEN bit enables the use of USBFS interrupts to cancel Software Standby mode.

AGT1UDWUPEN bit (AGT1 Underflow Interrupt Software Standby Returns Enable)

The AGT1UDWUPEN bit enables the use of AGT1 underflow interrupts to cancel Software Standby mode.

AGT1CAWUPEN bit (AGT1 Compare Match A Interrupt Software Standby Returns Enable)

The AGT1CAWUPEN bit enables the use of AGT1 compare match A interrupts to cancel Software Standby mode.

AGT1CBWUPEN bit (AGT1 Compare Match B Interrupt Software Standby Returns Enable)

The AGT1CBWUPEN bit enables the use of AGT1 compare match B interrupts to cancel Software Standby mode.

IIC0WUPEN bit (IIC0 Address Match Interrupt Software Standby Returns Enable)

The IIC0WUPEN bit enables the use of IIC0 interrupts to cancel Software Standby mode.

14.3 Vector Table

The ICU detects two types of interrupts, maskable and non-maskable. Interrupt priorities are set up in the Arm NVIC. See the NVIC chapter of the *ARM® Cortex®-M4 Processor Technical Reference Manual* (ARM DDI 0439D).

14.3.1 Interrupt Vector Table

Table 14.3 describes the interrupt vectors. The addresses conform to the NVIC specifications.

Table 14.3 Interrupt vector table (1 of 4)

Exception number	IRQ number	Vector offset	Source	Description
0	-	000h	Arm	Initial stack pointer
1	-	004h	Arm	Initial program counter (reset vector)
2	-	008h	Arm	Non-maskable interrupt (NMI)
3	-	00Ch	Arm	Hard fault
4	-	010h	Arm	MemManage fault
5	-	014h	Arm	Bus fault
6	-	018h	Arm	Usage fault
7	-	01Ch	Arm	Reserved
8	-	020h	Arm	Reserved
9	-	024h	Arm	Reserved
10	-	028h	Arm	Reserved
11	-	02Ch	Arm	Supervisor call (SVCall)
12	-	030h	Arm	Debug Monitor
13	-	034h	Arm	Reserved
14	-	038h	Arm	Pendable request for system service (PendableSrvReq)
15	-	03Ch	Arm	System tick timer (SysTick)

Table 14.3 Interrupt vector table (2 of 4)

Exception number	IRQ number	Vector offset	Source	Description
16	0	040h	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	044h	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	048h	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	04Ch	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	050h	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	054h	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	058h	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	05Ch	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	060h	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	064h	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	068h	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	06Ch	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	070h	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	074h	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	078h	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	07Ch	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	080h	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	084h	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	088h	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	08Ch	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	090h	ICU.IELSR20	Event selected in the ICU.IELSR20 register
37	21	094h	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	098h	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	09Ch	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	0A0h	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	0A4h	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	0A8h	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	0ACh	ICU.IELSR27	Event selected in the ICU.IELSR27 register
44	28	0B0h	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	0B4h	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	0B8h	ICU.IELSR30	Event selected in the ICU.IELSR30 register
47	31	0BCh	ICU.IELSR31	Event selected in the ICU.IELSR31 register
48	32	0C0h	ICU.IELSR32	Event selected in the ICU.IELSR32 register
49	33	0C4h	ICU.IELSR33	Event selected in the ICU.IELSR33 register
50	34	0C8h	ICU.IELSR34	Event selected in the ICU.IELSR34 register
51	35	0CCh	ICU.IELSR35	Event selected in the ICU.IELSR35 register
52	36	0D0h	ICU.IELSR36	Event selected in the ICU.IELSR36 register
53	37	0D4h	ICU.IELSR37	Event selected in the ICU.IELSR37 register
54	38	0D8h	ICU.IELSR38	Event selected in the ICU.IELSR38 register
55	39	0DCh	ICU.IELSR39	Event selected in the ICU.IELSR39 register
56	40	0E0h	ICU.IELSR40	Event selected in the ICU.IELSR40 register
57	41	0E4h	ICU.IELSR41	Event selected in the ICU.IELSR41 register
58	42	0E8h	ICU.IELSR42	Event selected in the ICU.IELSR42 register
59	43	0ECh	ICU.IELSR43	Event selected in the ICU.IELSR43 register
60	44	0F0h	ICU.IELSR44	Event selected in the ICU.IELSR44 register

Table 14.3 Interrupt vector table (3 of 4)

Exception number	IRQ number	Vector offset	Source	Description
61	45	0F4h	ICU.IELSR45	Event selected in the ICU.IELSR45 register
62	46	0F8h	ICU.IELSR46	Event selected in the ICU.IELSR46 register
63	47	0FCh	ICU.IELSR47	Event selected in the ICU.IELSR47 register
64	48	100h	ICU.IELSR48	Event selected in the ICU.IELSR48 register
65	49	104h	ICU.IELSR49	Event selected in the ICU.IELSR49 register
66	50	108h	ICU.IELSR50	Event selected in the ICU.IELSR50 register
67	51	10Ch	ICU.IELSR51	Event selected in the ICU.IELSR51 register
68	52	110h	ICU.IELSR52	Event selected in the ICU.IELSR52 register
69	53	114h	ICU.IELSR53	Event selected in the ICU.IELSR53 register
70	54	118h	ICU.IELSR54	Event selected in the ICU.IELSR54 register
71	55	11Ch	ICU.IELSR55	Event selected in the ICU.IELSR55 register
72	56	120h	ICU.IELSR56	Event selected in the ICU.IELSR56 register
73	57	124h	ICU.IELSR57	Event selected in the ICU.IELSR57 register
74	58	128h	ICU.IELSR58	Event selected in the ICU.IELSR58 register
75	59	12Ch	ICU.IELSR59	Event selected in the ICU.IELSR59 register
76	60	130h	ICU.IELSR60	Event selected in the ICU.IELSR60 register
77	61	134h	ICU.IELSR61	Event selected in the ICU.IELSR61 register
78	62	138h	ICU.IELSR62	Event selected in the ICU.IELSR62 register
79	63	13Ch	ICU.IELSR63	Event selected in the ICU.IELSR63 register
80	64	140h	ICU.IELSR64	Event selected in the ICU.IELSR64 register
81	65	144h	ICU.IELSR65	Event selected in the ICU.IELSR65 register
82	66	148h	ICU.IELSR66	Event selected in the ICU.IELSR66 register
83	67	14Ch	ICU.IELSR67	Event selected in the ICU.IELSR67 register
84	68	150h	ICU.IELSR68	Event selected in the ICU.IELSR68 register
85	69	154h	ICU.IELSR69	Event selected in the ICU.IELSR69 register
86	70	158h	ICU.IELSR70	Event selected in the ICU.IELSR70 register
87	71	15Ch	ICU.IELSR71	Event selected in the ICU.IELSR71 register
88	72	160h	ICU.IELSR72	Event selected in the ICU.IELSR72 register
89	73	164h	ICU.IELSR73	Event selected in the ICU.IELSR73 register
90	74	168h	ICU.IELSR74	Event selected in the ICU.IELSR74 register
91	75	16Ch	ICU.IELSR75	Event selected in the ICU.IELSR75 register
92	76	170h	ICU.IELSR76	Event selected in the ICU.IELSR76 register
93	77	174h	ICU.IELSR77	Event selected in the ICU.IELSR77 register
94	78	178h	ICU.IELSR78	Event selected in the ICU.IELSR78 register
95	79	17Ch	ICU.IELSR79	Event selected in the ICU.IELSR79 register
96	80	180h	ICU.IELSR80	Event selected in the ICU.IELSR80 register
97	81	184h	ICU.IELSR81	Event selected in the ICU.IELSR81 register
98	82	188h	ICU.IELSR82	Event selected in the ICU.IELSR82 register
99	83	18Ch	ICU.IELSR83	Event selected in the ICU.IELSR83 register
100	84	190h	ICU.IELSR84	Event selected in the ICU.IELSR84 register
101	85	194h	ICU.IELSR85	Event selected in the ICU.IELSR85 register
102	86	198h	ICU.IELSR86	Event selected in the ICU.IELSR86 register
103	87	19Ch	ICU.IELSR87	Event selected in the ICU.IELSR87 register
104	88	1A0h	ICU.IELSR88	Event selected in the ICU.IELSR88 register
105	89	1A4h	ICU.IELSR89	Event selected in the ICU.IELSR89 register

Table 14.3 Interrupt vector table (4 of 4)

Exception number	IRQ number	Vector offset	Source	Description
106	90	1A8h	ICU.IELSR90	Event selected in the ICU.IELSR90 register
107	91	1ACh	ICU.IELSR91	Event selected in the ICU.IELSR91 register
108	92	1B0h	ICU.IELSR92	Event selected in the ICU.IELSR92 register
109	93	1B4h	ICU.IELSR93	Event selected in the ICU.IELSR93 register
110	94	1B8h	ICU.IELSR94	Event selected in the ICU.IELSR94 register
111	95	1BCh	ICU.IELSR95	Event selected in the ICU.IELSR95 register

14.3.2 Event Numbers

The following table lists heading details for Table 14.4, which describes each event number.

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	"✓" indicates the interrupt can be used as a CPU interrupt (IELSRn setting)
Invoke DTC	"✓" indicates the interrupt can be used to request DTC activation (IELSRn setting)
Invoke DMAC	"✓" indicates the interrupt can be used to request DMAC activation (DELSRn setting)
Canceling Snooze mode	"✓" indicates the interrupt can be used to request a return from Snooze mode using SELSR0. Otherwise, "✓" indicates it can be used directly
Canceling Software Standby mode	"✓" indicates the interrupt can be used to request a return from Software Standby mode
Canceling Deep Software Standby mode	"✓" indicates the interrupt can be used to request a return from Deep Software Standby mode

Table 14.4 Event table (1 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
			Connect to NVIC	Invoke DTC	Invoke DMAC			
001h	Port	PORT_IRQ0	✓	✓	✓	✓	✓	✓
002h		PORT_IRQ1	✓	✓	✓	✓	✓	✓
003h		PORT_IRQ2	✓	✓	✓	✓	✓	✓
004h		PORT_IRQ3	✓	✓	✓	✓	✓	✓
005h		PORT_IRQ4	✓	✓	✓	✓	✓	✓
006h		PORT_IRQ5	✓	✓	✓	✓	✓	✓
007h		PORT_IRQ6	✓	✓	✓	✓	✓	✓
008h		PORT_IRQ7	✓	✓	✓	✓	✓	✓
009h		PORT_IRQ8	✓	✓	✓	✓	✓	✓
00Ah		PORT_IRQ9	✓	✓	✓	✓	✓	✓
00Bh		PORT_IRQ10	✓	✓	✓	✓	✓	✓
00Ch		PORT_IRQ11	✓	✓	✓	✓	✓	✓
00Dh		PORT_IRQ12	✓	✓	✓	✓	✓	✓
00Eh		PORT_IRQ13	✓	✓	✓	✓	✓	✓
00Fh		PORT_IRQ14	✓	✓	✓	✓	✓	✓
010h		PORT_IRQ15	✓	✓	✓	✓	✓	-
020h	DMAC0	DMAC0_INT	✓	✓	-	-	-	-
021h	DMAC1	DMAC1_INT	✓	✓	-	-	-	-
022h	DMAC2	DMAC2_INT	✓	✓	-	-	-	-

Table 14.4 Event table (2 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
			Connect to NVIC	Invoke DTC	Invoke DMAC			
023h	DMAC3	DMAC3_INT	✓	✓	-	-	-	-
024h	DMAC4	DMAC4_INT	✓	✓	-	-	-	-
025h	DMAC5	DMAC5_INT	✓	✓	-	-	-	-
026h	DMAC6	DMAC6_INT	✓	✓	-	-	-	-
027h	DMAC7	DMAC7_INT	✓	✓	-	-	-	-
029h	DTC	DTC_COMPLETE	✓	-	-	✓*5	-	-
02Dh	ICU	ICU_SNZCANCEL	✓	-	-	✓	-	-
030h	FCU	FCU_FIFERR	✓	-	-	-	-	-
031h		FCU_FRDYI	✓	-	-	-	-	-
038h	LVD	LVD_LVD1	✓	-	-	✓	✓	✓
039h		LVD_LVD2	✓	-	-	✓	✓	✓
03Bh	MOSC	MOSC_STOP	✓	-	-	-	-	-
03Ch	Low-power mode	SYSTEM_SNZREQ	-	✓	-	-	-	-
040h	AGT0	AGT0_AGTI	✓	✓	✓	-	-	-
041h		AGT0_AGTCMAI	✓	✓	✓	-	-	-
042h		AGT0_AGTCMBI	✓	✓	✓	-	-	-
043h	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓	✓
044h		AGT1_AGTCMAI	✓	✓	✓	✓	✓	-
045h		AGT1_AGTCMBI	✓	✓	✓	✓	✓	-
046h	IWDT	IWDT_NMIUNDF	✓	-	-	✓	✓	-
047h	WDT	WDT_NMIUNDF	✓	-	-	-	-	-
048h	RTC	RTC_ALM	✓	-	-	✓	✓	✓
049h		RTC_PRD	✓	-	-	✓	✓	✓
04Ah		RTC_CUP	✓	-	-	-	-	-
04Bh	ADC120	ADC120_ADI	✓	✓	✓	-	-	-
04Ch		ADC120_GBADI	✓	✓	✓	-	-	-
04Dh		ADC120_CMPAI	✓	-	-	-	-	-
04Eh		ADC120_CMPBI	✓	-	-	-	-	-
04Fh		ADC120_WCMPM	-	✓	✓	✓*5	-	-
050h		ADC120_WCMPUM	-	✓	✓	✓*5	-	-
051h	ADC121	ADC121_ADI	✓	✓	✓	-	-	-
052h		ADC121_GBADI	✓	✓	✓	-	-	-
053h		ADC121_CMPAI	✓	-	-	-	-	-
054h		ADC121_CMPBI	✓	-	-	-	-	-
055h		ADC121_WCMPM	-	✓	✓	✓*5	-	-
056h		ADC121_WCMPUM	-	✓	✓	✓*5	-	-
057h	ACMPHS	ACMP_HS0	✓	-	-	✓*1	✓*1	-
058h		ACMP_HS1	✓	-	-	-	-	-
059h		ACMP_HS2	✓	-	-	-	-	-
05Ah		ACMP_HS3	✓	-	-	-	-	-
05Bh		ACMP_HS4	✓	-	-	-	-	-
05Ch		ACMP_HS5	✓	-	-	-	-	-
05Fh	USBFS	USBFS_D0FIFO	✓	✓	✓	-	-	-
060h		USBFS_D1FIFO	✓	✓	✓	-	-	-
061h		USBFS_USBI	✓	-	-	-	-	-
062h		USBFS_USBR	✓	-	-	✓	✓	✓

Table 14.4 Event table (3 of 9)

Event number	Interrupt request source	Name	IELSRn Connect to NVIC	IELSRn Invoke DTC	DELSRn Invoke DMAC	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
063h	IIC0	IIC0_RXI	✓	✓	✓	-	-	-
064h		IIC0_TXI	✓	✓	✓	-	-	-
065h		IIC0_TEI	✓	-	-	-	-	-
066h		IIC0_EEI	✓	-	-	-	-	-
067h		IIC0_WUI	✓	-	-	✓	✓	-
068h	IIC1	IIC1_RXI	✓	✓	✓	-	-	-
069h		IIC1_TXI	✓	✓	✓	-	-	-
06Ah		IIC1_TEI	✓	-	-	-	-	-
06Bh		IIC1_EEI	✓	-	-	-	-	-
06Dh	IIC2	IIC2_RXI	✓	✓	✓	-	-	-
06Eh		IIC2_TXI	✓	✓	✓	-	-	-
06Fh		IIC2_TEI	✓	-	-	-	-	-
070h		IIC2_EEI	✓	-	-	-	-	-
072h	SSIE0	SSIE0_SSITXI	✓	✓	✓	-	-	-
073h		SSIE0_SSIRXI	✓	✓	✓	-	-	-
075h		SSIE0_SSIF	✓	-	-	-	-	-
078h	SSIE1	SSIE1_SSIRT	✓	✓	✓	-	-	-
079h		SSIE1_SSIF	✓	-	-	-	-	-
07Ah	SRC	SRC_IDEI	✓	✓	✓	-	-	-
07Bh		SRC_ODFI	✓	✓	✓	-	-	-
07Ch		SRC_OVFI	✓	-	-	-	-	-
07Dh		SRC_UDFI	✓	-	-	-	-	-
07Eh		SRC_CEFI	✓	-	-	-	-	-
07Fh	PDC	PDC_PCDFI	✓	✓	✓	-	-	-
080h		PDC_PCFEI	✓	-	-	-	-	-
081h		PDC_PCERI	✓	-	-	-	-	-
082h	CTSU	CTSU_CTSUWR	✓	✓	✓	-	-	-
083h		CTSU_CTSURD	✓	✓	✓	-	-	-
084h		CTSU_CTSUFN	✓	-	-	✓*5	-	-
085h	KINT	KEY_INTKR	✓	-	-	✓*2	✓*2	-
086h	DOC	DOC_DOPCI	✓	-	-	✓*5	-	-
087h	CAC	CAC_FERRI	✓	-	-	-	-	-
088h		CAC_MENDI	✓	-	-	-	-	-
089h		CAC_OVFI	✓	-	-	-	-	-
08Ah	CAN0	CAN0_ERS	✓	-	-	-	-	-
08Bh		CAN0_RXF	✓	-	-	-	-	-
08Ch		CAN0_TXF	✓	-	-	-	-	-
08Dh		CAN0_RXM	✓	-	-	-	-	-
08Eh		CAN0_TXM	✓	-	-	-	-	-
08Fh	CAN1	CAN1_ERS	✓	-	-	-	-	-
090h		CAN1_RXF	✓	-	-	-	-	-
091h		CAN1_TXF	✓	-	-	-	-	-
092h		CAN1_RXM	✓	-	-	-	-	-
093h		CAN1_TXM	✓	-	-	-	-	-

Table 14.4 Event table (4 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
			Connect to NVIC	Invoke DTC	Invoke DMAC			
094h	I/O port	IOPORT_GROUP1	✓	✓*3	✓*3	-	-	-
095h		IOPORT_GROUP2	✓	✓*3	✓*3	-	-	-
096h		IOPORT_GROUP3	✓	✓*3	✓*3	-	-	-
097h		IOPORT_GROUP4	✓	✓*3	✓*3	-	-	-
098h	ELC	ELC_SWEVT0	✓*4	✓	-	-	-	-
099h		ELC_SWEVT1	✓*4	✓	-	-	-	-
09Ah	POEG	POEG_GROUP0	✓	-	-	-	-	-
09Bh		POEG_GROUP1	✓	-	-	-	-	-
09Ch		POEG_GROUP2	✓	-	-	-	-	-
09Dh		POEG_GROUP3	✓	-	-	-	-	-
0B0h	GPT32EH0	GPT0_CCMPA	✓	✓	✓	-	-	-
0B1h		GPT0_CCMPB	✓	✓	✓	-	-	-
0B2h		GPT0_CMPC	✓	✓	✓	-	-	-
0B3h		GPT0_CMPD	✓	✓	✓	-	-	-
0B4h		GPT0_CMPE	✓	✓	✓	-	-	-
0B5h		GPT0_CMPF	✓	✓	✓	-	-	-
0B6h		GPT0_OVF	✓	✓	✓	-	-	-
0B7h		GPT0_UDF	✓	✓	✓	-	-	-
0B8h		GPT0_ADTRGA	✓	✓	✓	-	-	-
0B9h		GPT0_ADTRGB	✓	✓	✓	-	-	-
0BAh	GPT32EH1	GPT1_CCMPA	✓	✓	✓	-	-	-
0BBh		GPT1_CCMPB	✓	✓	✓	-	-	-
0BCh		GPT1_CMPC	✓	✓	✓	-	-	-
0BDh		GPT1_CMPD	✓	✓	✓	-	-	-
0BEh		GPT1_CMPE	✓	✓	✓	-	-	-
0BFh		GPT1_CMPF	✓	✓	✓	-	-	-
0C0h		GPT1_OVF	✓	✓	✓	-	-	-
0C1h		GPT1_UDF	✓	✓	✓	-	-	-
0C2h		GPT1_ADTRGA	✓	✓	✓	-	-	-
0C3h		GPT1_ADTRGB	✓	✓	✓	-	-	-
0C4h	GPT32EH2	GPT2_CCMPA	✓	✓	✓	-	-	-
0C5h		GPT2_CCMPB	✓	✓	✓	-	-	-
0C6h		GPT2_CMPC	✓	✓	✓	-	-	-
0C7h		GPT2_CMPD	✓	✓	✓	-	-	-
0C8h		GPT2_CMPE	✓	✓	✓	-	-	-
0C9h		GPT2_CMPF	✓	✓	✓	-	-	-
0CAh		GPT2_OVF	✓	✓	✓	-	-	-
0CBh		GPT2_UDF	✓	✓	✓	-	-	-
0CCh		GPT2_ADTRGA	✓	✓	✓	-	-	-
0CDh		GPT2_ADTRGB	✓	✓	✓	-	-	-

Table 14.4 Event table (5 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0CEh	GPT32EH3	GPT3_CCMPA	✓	✓	✓	-	-	-
0CFh		GPT3_CCMPB	✓	✓	✓	-	-	-
0D0h		GPT3_CMPC	✓	✓	✓	-	-	-
0D1h		GPT3_CMPD	✓	✓	✓	-	-	-
0D2h		GPT3_CMPE	✓	✓	✓	-	-	-
0D3h		GPT3_CMPF	✓	✓	✓	-	-	-
0D4h		GPT3_OVF	✓	✓	✓	-	-	-
0D5h		GPT3_UDF	✓	✓	✓	-	-	-
0D6h		GPT3_ADTRGA	✓	✓	✓	-	-	-
0D7h		GPT3_ADTRGB	✓	✓	✓	-	-	-
0D8h	GPT32E4	GPT4_CCMPA	✓	✓	✓	-	-	-
0D9h		GPT4_CCMPB	✓	✓	✓	-	-	-
0DAh		GPT4_CMPC	✓	✓	✓	-	-	-
0DBh		GPT4_CMPD	✓	✓	✓	-	-	-
0DCh		GPT4_CMPE	✓	✓	✓	-	-	-
0DDh		GPT4_CMPF	✓	✓	✓	-	-	-
0DEh		GPT4_OVF	✓	✓	✓	-	-	-
0DFh		GPT4_UDF	✓	✓	✓	-	-	-
0E0h		GPT4_ADTRGA	✓	✓	✓	-	-	-
0E1h		GPT4_ADTRGB	✓	✓	✓	-	-	-
0E2h	GPT32E5	GPT5_CCMPA	✓	✓	✓	-	-	-
0E3h		GPT5_CCMPB	✓	✓	✓	-	-	-
0E4h		GPT5_CMPC	✓	✓	✓	-	-	-
0E5h		GPT5_CMPD	✓	✓	✓	-	-	-
0E6h		GPT5_CMPE	✓	✓	✓	-	-	-
0E7h		GPT5_CMPF	✓	✓	✓	-	-	-
0E8h		GPT5_OVF	✓	✓	✓	-	-	-
0E9h		GPT5_UDF	✓	✓	✓	-	-	-
0EAh		GPT5_ADTRGA	✓	✓	✓	-	-	-
0EBh		GPT5_ADTRGB	✓	✓	✓	-	-	-
0ECh	GPT32E6	GPT6_CCMPA	✓	✓	✓	-	-	-
0EDh		GPT6_CCMPB	✓	✓	✓	-	-	-
0EEh		GPT6_CMPC	✓	✓	✓	-	-	-
0EFh		GPT6_CMPD	✓	✓	✓	-	-	-
0F0h		GPT6_CMPE	✓	✓	✓	-	-	-
0F1h		GPT6_CMPF	✓	✓	✓	-	-	-
0F2h		GPT6_OVF	✓	✓	✓	-	-	-
0F3h		GPT6_UDF	✓	✓	✓	-	-	-
0F4h		GPT6_ADTRGA	✓	✓	✓	-	-	-
0F5h		GPT6_ADTRGB	✓	✓	✓	-	-	-

Table 14.4 Event table (6 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0F6h	GPT32E7	GPT7_CCMPA	✓	✓	✓	-	-	-
0F7h		GPT7_CCMPB	✓	✓	✓	-	-	-
0F8h		GPT7_CMPC	✓	✓	✓	-	-	-
0F9h		GPT7_CMPD	✓	✓	✓	-	-	-
0FAh		GPT7_CMPE	✓	✓	✓	-	-	-
0FBh		GPT7_CMPF	✓	✓	✓	-	-	-
0FCh		GPT7_OVF	✓	✓	✓	-	-	-
0FDh		GPT7_UDF	✓	✓	✓	-	-	-
0FEh		GPT7_ADTRGA	✓	✓	✓	-	-	-
0FFh		GPT7_ADTRGB	✓	✓	✓	-	-	-
100h	GPT328	GPT8_CCMPA	✓	✓	✓	-	-	-
101h		GPT8_CCMPB	✓	✓	✓	-	-	-
102h		GPT8_CMPC	✓	✓	✓	-	-	-
103h		GPT8_CMPD	✓	✓	✓	-	-	-
104h		GPT8_CMPE	✓	✓	✓	-	-	-
105h		GPT8_CMPF	✓	✓	✓	-	-	-
106h		GPT8_OVF	✓	✓	✓	-	-	-
107h		GPT8_UDF	✓	✓	✓	-	-	-
10Ah	GPT329	GPT9_CCMPA	✓	✓	✓	-	-	-
10Bh		GPT9_CCMPB	✓	✓	✓	-	-	-
10Ch		GPT9_CMPC	✓	✓	✓	-	-	-
10Dh		GPT9_CMPD	✓	✓	✓	-	-	-
10Eh		GPT9_CMPE	✓	✓	✓	-	-	-
10Fh		GPT9_CMPF	✓	✓	✓	-	-	-
110h		GPT9_OVF	✓	✓	✓	-	-	-
111h		GPT9_UDF	✓	✓	✓	-	-	-
114h	GPT3210	GPT10_CCMPA	✓	✓	✓	-	-	-
115h		GPT10_CCMPB	✓	✓	✓	-	-	-
116h		GPT10_CMPC	✓	✓	✓	-	-	-
117h		GPT10_CMPD	✓	✓	✓	-	-	-
118h		GPT10_CMPE	✓	✓	✓	-	-	-
119h		GPT10_CMPF	✓	✓	✓	-	-	-
11Ah		GPT10_OVF	✓	✓	✓	-	-	-
11Bh		GPT10_UDF	✓	✓	✓	-	-	-
11Eh	GPT3211	GPT11_CCMPA	✓	✓	✓	-	-	-
11Fh		GPT11_CCMPB	✓	✓	✓	-	-	-
120h		GPT11_CMPC	✓	✓	✓	-	-	-
121h		GPT11_CMPD	✓	✓	✓	-	-	-
122h		GPT11_CMPE	✓	✓	✓	-	-	-
123h		GPT11_CMPF	✓	✓	✓	-	-	-
124h		GPT11_OVF	✓	✓	✓	-	-	-
125h		GPT11_UDF	✓	✓	✓	-	-	-

Table 14.4 Event table (7 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
			Connect to NVIC	Invoke DTC	Invoke DMAC			
128h	GPT3212	GPT12_CCMPA	✓	✓	✓	-	-	-
129h		GPT12_CCMPB	✓	✓	✓	-	-	-
12Ah		GPT12_CMPC	✓	✓	✓	-	-	-
12Bh		GPT12_CMPD	✓	✓	✓	-	-	-
12Ch		GPT12_CMPE	✓	✓	✓	-	-	-
12Dh		GPT12_CMPF	✓	✓	✓	-	-	-
12Eh		GPT12_OVF	✓	✓	✓	-	-	-
12Fh		GPT12_UDF	✓	✓	✓	-	-	-
132h	GPT3213	GPT13_CCMPA	✓	✓	✓	-	-	-
133h		GPT13_CCMPB	✓	✓	✓	-	-	-
134h		GPT13_CMPC	✓	✓	✓	-	-	-
135h		GPT13_CMPD	✓	✓	✓	-	-	-
136h		GPT13_CMPE	✓	✓	✓	-	-	-
137h		GPT13_CMPF	✓	✓	✓	-	-	-
138h		GPT13_OVF	✓	✓	✓	-	-	-
139h		GPT13_UDF	✓	✓	✓	-	-	-
150h	GPT	GPT_UVWEDGE	✓			-	-	-
160h	Ethernet Controller	ETHER_IPLS	✓	✓	✓	-	-	-
161h		ETHER_MINT	✓	-	-	-	-	-
162h		ETHER_PINT	✓	-	-	-	-	-
163h		ETHER_EINT0	✓	-	-	-	-	-
171h	USBHS	USBHS_D0FIFO	✓	✓	✓	-	-	-
172h		USBHS_D1FIFO	✓	✓	✓	-	-	-
173h		USBHS_USBIR	✓	-	-	✓	✓	✓
174h	SCI0	SCI0_RXI	✓	✓	✓	-	-	-
175h		SCI0_TXI	✓	✓	✓	-	-	-
176h		SCI0_TEI	✓	-	-	-	-	-
177h		SCI0_ERI	✓	-	-	-	-	-
178h		SCI0_AM	✓	-	-	✓*5	-	-
179h		SCI0_RXI_OR_ERI	-	-	-	✓*5	-	-
17Ah	SCI1	SCI1_RXI	✓	✓	✓	-	-	-
17Bh		SCI1_TXI	✓	✓	✓	-	-	-
17Ch		SCI1_TEI	✓	-	-	-	-	-
17Dh		SCI1_ERI	✓	-	-	-	-	-
17Eh		SCI1_AM	✓	-	-	-	-	-
180h	SCI2	SCI2_RXI	✓	✓	✓	-	-	-
181h		SCI2_TXI	✓	✓	✓	-	-	-
182h		SCI2_TEI	✓	-	-	-	-	-
183h		SCI2_ERI	✓	-	-	-	-	-
184h		SCI2_AM	✓	-	-	-	-	-
186h	SCI3	SCI3_RXI	✓	✓	✓	-	-	-
187h		SCI3_TXI	✓	✓	✓	-	-	-
188h		SCI3_TEI	✓	-	-	-	-	-
189h		SCI3_ERI	✓	-	-	-	-	-
18Ah		SCI3_AM	✓	-	-	-	-	-

Table 14.4 Event table (8 of 9)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
			Connect to NVIC	Invoke DTC	Invoke DMAC			
18Ch	SCI4	SCI4_RXI	✓	✓	✓	-	-	-
18Dh		SCI4_TXI	✓	✓	✓	-	-	-
18Eh		SCI4_TEI	✓	-	-	-	-	-
18Fh		SCI4_ERI	✓	-	-	-	-	-
190h		SCI4_AM	✓	-	-	-	-	-
192h	SCI5	SCI5_RXI	✓	✓	✓	-	-	-
193h		SCI5_TXI	✓	✓	✓	-	-	-
194h		SCI5_TEI	✓	-	-	-	-	-
195h		SCI5_ERI	✓	-	-	-	-	-
196h		SCI5_AM	✓	-	-	-	-	-
198h	SCI6	SCI6_RXI	✓	✓	✓	-	-	-
199h		SCI6_TXI	✓	✓	✓	-	-	-
19Ah		SCI6_TEI	✓	-	-	-	-	-
19Bh		SCI6_ERI	✓	-	-	-	-	-
19Ch		SCI6_AM	✓	-	-	-	-	-
19Eh	SCI7	SCI7_RXI	✓	✓	✓	-	-	-
19Fh		SCI7_TXI	✓	✓	✓	-	-	-
1A0h		SCI7_TEI	✓	-	-	-	-	-
1A1h		SCI7_ERI	✓	-	-	-	-	-
1A2h		SCI7_AM	✓	-	-	-	-	-
1A4h	SCI8	SCI8_RXI	✓	✓	✓	-	-	-
1A5h		SCI8_TXI	✓	✓	✓	-	-	-
1A6h		SCI8_TEI	✓	-	-	-	-	-
1A7h		SCI8_ERI	✓	-	-	-	-	-
1A8h		SCI8_AM	✓	-	-	-	-	-
1AAh	SCI9	SCI9_RXI	✓	✓	✓	-	-	-
1ABh		SCI9_TXI	✓	✓	✓	-	-	-
1ACh		SCI9_TEI	✓	-	-	-	-	-
1ADh		SCI9_ERI	✓	-	-	-	-	-
1AEh		SCI9_AM	✓	-	-	-	-	-
1BCh	SPI0	SPI0_SPRI	✓	✓	✓	-	-	-
1BDh		SPI0_SPTI	✓	✓	✓	-	-	-
1BEh		SPI0_SPII	✓	-	-	-	-	-
1BFh		SPI0_SPEI	✓	-	-	-	-	-
1C0h		SPI0_SPTEND	✓	-	-	-	-	-
1C1h	SPI1	SPI1_SPRI	✓	✓	✓	-	-	-
1C2h		SPI1_SPTI	✓	✓	✓	-	-	-
1C3h		SPI1_SPII	✓	-	-	-	-	-
1C4h		SPI1_SPEI	✓	-	-	-	-	-
1C5h		SPI1_SPTEND	✓	-	-	-	-	-
1C6h	QSPI	QSPI_INTR	✓	-	-	-	-	-
1C7h	SDHI0	SDHI_MMC0_ACCS	✓	-	-	-	-	-
1C8h		SDHI_MMC0_SDIO	✓	-	-	-	-	-
1C9h		SDHI_MMC0_CARD	✓	-	-	-	-	-
1CAh		SDHI_MMC0_ODMSDBREQ	-	✓	✓	-	-	-

Table 14.4 Event table (9 of 9)

Event number	Interrupt request source	Name	IELSRn Connect to NVIC	IELSRn Invoke DTC	DELSRn Invoke DMAC	Canceling Snooze mode	Canceling Software Standby mode	Canceling Deep Software Standby mode
1CBh	SDHI1	SDHI_MMC1_ACCS	✓	-	-	-	-	-
1CCh		SDHI_MMC1_SDIO	✓	-	-	-	-	-
1CDh		SDHI_MMC1_CARD	✓	-	-	-	-	-
1CEh		SDHI_MMC1_ODMSDBREQ	-	✓	✓	-	-	-
1FAh	GLCDC	GLCDC_VPOS	✓	-	-	-	-	-
1FBh		GLCDC_L1UNDF	✓	-	-	-	-	-
1FCh		GLCDC_L2UNDF	✓	-	-	-	-	-
1FDh	DRW	DRW_IRQ	✓	-	-	-	-	-
1FEh	JPEG	JPEG_JEDI	✓	-	-	-	-	-
1FFh		JPEG_JDTI	✓	-	-	-	-	-

Note 1. Only supported when CMPCTL0.CSTEN = 1.
Note 2. Only supported when KRCTL.KRMD = 1.
Note 3. Only the first edge detection is valid.
Note 4. Only interrupts after DTC transfer are supported.
Note 5. Using SELSR0.

14.4 Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation, or DMAC activation.

14.4.1 Detecting Interrupts

External pin interrupt requests are detected by either the edge or level (falling edge, rising edge, rising and falling edges, or low level) of the interrupt signal. Set the IRQMD[1:0] bits in the IRQCRi register to select the detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see section 14.3.2, Event Numbers. Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.

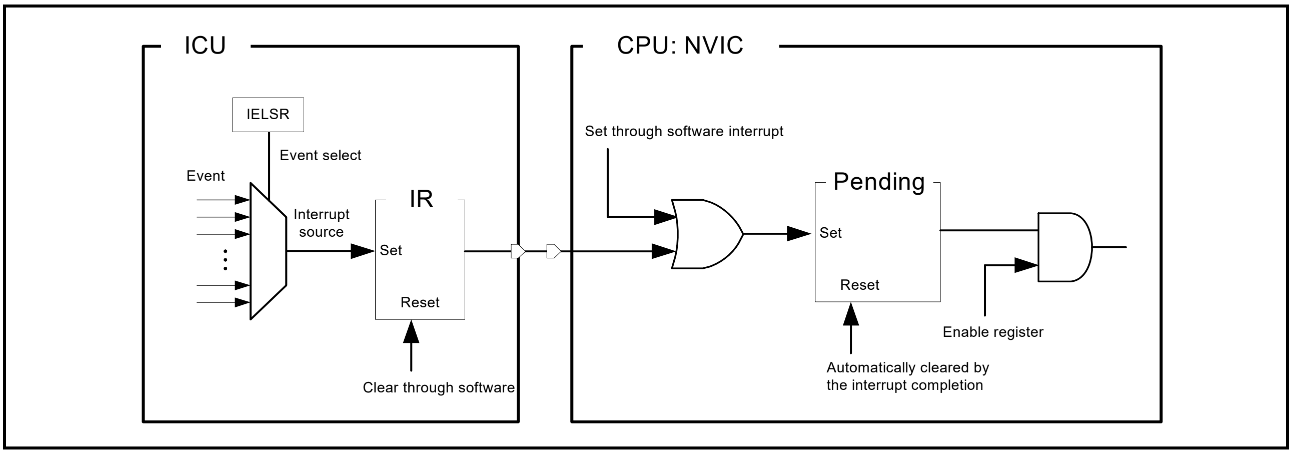


Figure 14.2 Interrupt path of the ICU and CPU: NVIC

Use the procedures in this section to detect interrupts:

- General operations during an interrupt
  - When a non-software interrupt occurs:
    The IELSRn.IR flag and Interrupt Set/Clear-Pending register (NVIC) are set.

  - When a software interrupt occurs:
    Set the Interrupt Set-Pending register.
  - When an interrupt is complete:
    Clear the IELSRn.IR flag in the software.
    The Interrupt Set/Clear-Pending register clears automatically.
- When interrupts are enabled
  a. Set the Interrupt Set-Enable register (NVIC).
  b. Set the IELSRn.IELS bits as the interrupt source.
  c. Specify the operation settings for the event source.
- When interrupts are disabled
  a. Disable the settings for the event source.
  b. Clear the IELSRn.IELS bits (IELSRn.IELS = 0000h). Clear the IELSRn.IR flag as required.
  c. Clear the Interrupt Clear-Enable register. Clear the Interrupt Clear-Pending register as required.
- When polling for interrupts
  a. Set the Interrupt Clear-Enable register (disabling interrupts).
  b. Set the IELSRn.IELS bits (selecting the source).
  c. Specify the operation settings for the event source.
  d. Poll the Interrupt Set-Pending register.
  e. When polling is no longer required, follow the procedure for clearing an interrupt when it is complete. Clear the IELSRn.IR flag in the software.

14.4.2 Selecting Interrupt Request Destinations

The interrupt output destination, CPU, DTC, or DMAC, can be independently selected for each interrupt source. The available destinations are fixed for each interrupt, as described in Table 14.4.

Note: Do not use an interrupt request destination setting that is not indicated by a "✓" in the event list (Table 14.4).

If you select the CPU or DTC in one IELSRn register, setting the same interrupt factor in any other IELSRn register is prohibited. Similarly, if you select the DMAC in one DELSRn register, setting the same interrupt factor in any other DELSRn register is prohibited.

Note: Setting the same interrupt factor for IELSRn and DELSRn is prohibited.

If the DMAC or DTC is selected as the destination for requests from an IRQi pin, you must set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.

14.4.2.1 CPU interrupt request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC. Use the following procedure:

Set the IELSRn.IELS bits to the target event and the IELSRn.DTCE bit to 0.

14.4.2.2 DTC activation

When IELSRn.DTCE = 1, the event specified in the IELSRn register is output to the DTC. After DTC transmission completes, the associated interrupt occurs. Use the following procedure:

1. Set the IELSRn.IELS bits to the target event and the IELSRn.DTCE bit to 1.
2. Set the DTC module start bit (DTCST.DTCST) to 1.

Table 14.5 shows operation when the DTC is the request destination.

Table 14.5 Operations when the DTC is activated

Interrupt request destination	DISEL[1]	Remaining transfer operations	Operations per request	IR[2]	Interrupt request destination after transfer
DTC[3]	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The IELSRn.DTCE bit is cleared and the CPU becomes the destination
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after DTC transfer data is read	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The IELSRn.DTCE bit is cleared and the CPU becomes the destination

Note 1. Set the interrupt request mode for the DTC in the DTC.MRB.DISEL bit.
Note 2. When the IELSRn.IR flag is 1, an interrupt request (DTC activation request) that occurs again is ignored.
Note 3. For chain transfers, DTC transfer continues until the last chain transfer ends. At this point, the DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the IELSRn.IR flag clear timing, and the interrupt request destination after transfer. See Table 18.3, Chain transfer conditions, in section 18, Data Transfer Controller (DTC).

14.4.2.3 Operations with the DMAC activated

Events specified in the DELSRn registers are output to the DMAC. When using interrupts, you must select the DMAC as the interrupt source in the IELSRn.IELS[8:0] bits and enable DMAC output by setting IELSRn.DTCE to 1. When IELSRn.DTCE is 0, the events specified in the IELSRn registers are output to the NVIC. Use the following procedure:

1. Set the DELSRn.DELS[8:0] bits to the target event.
2. When using interrupts, set the IELSRn.IELS bit to DMAC interrupts as the source, and set the IELSRn.DTCE bit to 1.
3. Set the activation source for the target DMAC channel (DMACm.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
4. Set the DMAC transfer enable bit for the target DMAC channel (DMACm.DMCNT.DTE) to 1.
5. Set the DMAC operation enable bit (DMACm.DMAST.DMST) to 1.

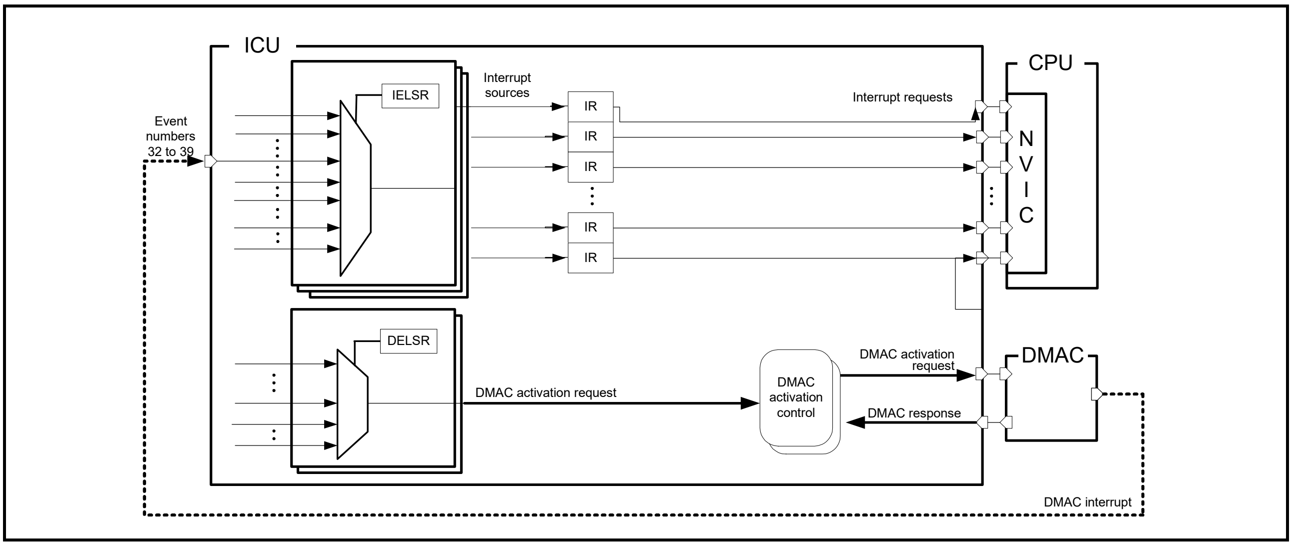


Figure 14.3 DMAC request trigger and interrupt path

14.4.3 Digital Filter

A digital filter function is provided for the external interrupt request pins (IRQi, i = 0 to 15) and NMI pin interrupt. It samples input signals on the filter sampling clock (PCLKB) and removes any signal with a pulse width less than three sampling cycles.

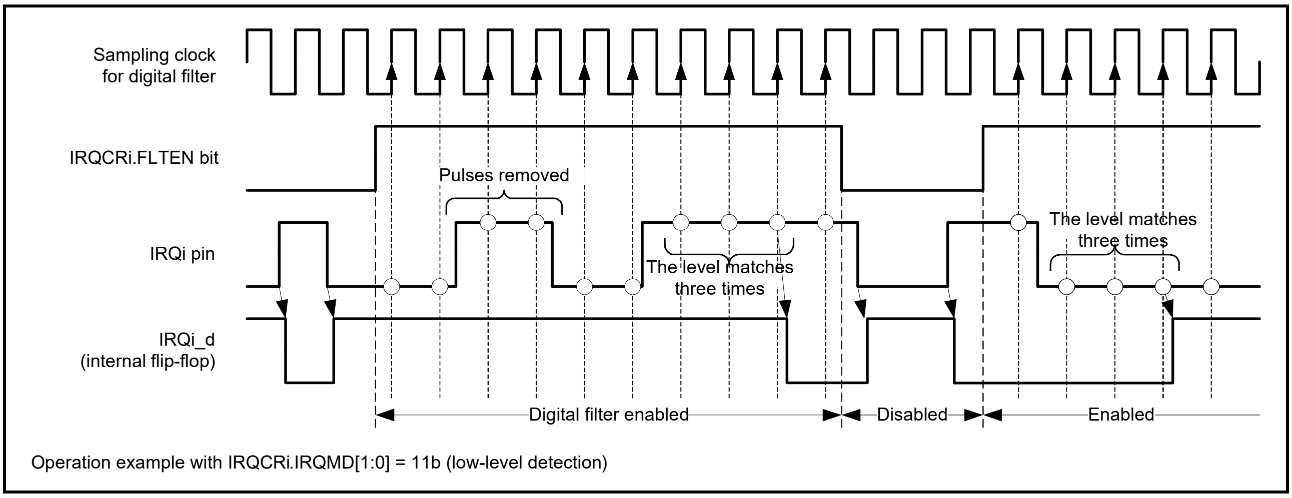
To use the digital filter for an IRQi pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the IRQCRi.FCLKSEL[1:0] bits.
2. Set the IRQCRi.FLTEN bit to 1 (digital filter enabled).

To use the digital filter for an NMI pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the NMICR.NFCLKSEL[1:0] bits.
2. Set the NMICR.NFLTEN bit to 1 (digital filter enabled).

Figure 14.4 shows an example of digital filter operation.



Operation example with IRQCRi.IRQMD[1:0] = 11b (low-level detection)

Figure 14.4 Digital filter operation example

Before entering Software Standby mode, disable the digital filters by clearing the IRQCRi.FLTEN and NMICR.NFLTEN bits. The clock for the ICU stops in Software Standby mode. On exiting Software Standby, the circuit detects the edge by comparing the state before standby to the state after standby release. If the input changes during Software Standby, an incorrect edge might be detected. You can enable the digital filters again after exiting Software Standby mode.

14.4.4 External Pin Interrupts

To use external pin interrupts:

1. Clear the IRQCRi.FLTEN bit (i = 0 to 15) to 0 (digital filter disabled).
2. Make or confirm the I/O port settings.
3. Set the IRQMD[1:0] bits, FCLKSEL[1:0] bits, and FLTEN bit of the IRQCRi register.
4. Select the IRQ pin as follows:
    - If the IRQ pin is to be used for CPU interrupt requests, set the IELSRn.IELS bits and set the IELSRn.DTCE bit to 0
    - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS bits and set the IELSRn.DTCE bit to 1
    - If the IRQ pin is to be used for DMAC activation, set the DELSRn.DELS bits.

14.5 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor 2 interrupt
- SRAM parity error interrupt
- SRAM ECC error interrupt
- MPU bus master error interrupt
- MPU bus slave error interrupt
- CPU stack pointer monitor interrupt.

Non-maskable interrupts can only be used with the CPU, not to activate the DTC or DMAC. Non-maskable interrupts take precedence over all other interrupts. The non-maskable interrupt states can be verified in the Non-Maskable Interrupt Status Register (NMISR). Confirm that all bits in the NMISR are 0 before returning from the NMI handler.

Non-maskable interrupts are disabled by default. To use non-maskable interrupts, you must:

1. To use the NMI pin, clear the NMICR.NFLTEN bit to 0 (digital filter disabled).
2. To use the NMI pin, set the NMIMD bit, NFCLKSEL[1:0] bits, and NFLTEN bit of NMICR register.
3. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
4. Enable the non-maskable interrupt by writing 1 to the associated bit in the Non-Maskable Interrupt Enable Register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. An NMI interrupt cannot be disabled when enabled, except by a reset.

14.6 Return from Low-Power Modes

Table 14.4 lists the interrupt sources you can use to exit Sleep or Software Standby mode. For more information, see section 11, Low Power Modes. Sections 14.6.1 to 14.6.3 describe how to use interrupts to return from Sleep, Software Standby, and Snooze modes. For Deep Software Standby, see section 11.9, Deep Software Standby Mode.

14.6.1 Return from Sleep Mode

To return from Sleep mode in response to an interrupt:

1. Select the CPU as the interrupt request destination.
2. Enable the interrupt in the NVIC.

To return from Sleep mode in response to a non-maskable interrupt, enable the wanted interrupt request in the NMIER register.

14.6.2 Return from Software Standby Mode

The ICU can return from Software Standby mode using a non-maskable interrupt or an interrupt selected in the WUPEN register. See section 14.2.9, Wake Up Interrupt Enable Register (WUPEN).

To return from Software Standby mode, you must:

1. Select the interrupt source that enables return from Software Standby.

- For non-maskable interrupts, use the NMIER register to enable the wanted interrupt request

- For maskable interrupts, use the WUPEN register to enable the wanted interrupt request.

2. Select the CPU as the interrupt request destination.
3. Enable the interrupt in the NVIC.

Interrupt requests through IRQ pins that do not satisfy these conditions are not detected while the clock is stopped in Software Standby mode.

14.6.3 Return from Snooze Mode

The ICU can return from Snooze mode using the interrupts provided for this mode.

To return to Normal mode from Snooze mode:

1. Use either of the following methods to select the event that you want to trigger a return from Snooze mode to Normal mode:
    - Set the event that you want to trigger a return from Snooze mode to Normal mode in SELSR0.SEL and set the value 02Dh (ICU_SNZCANCEL) in IELSRn.IELS
    - Set the event that you want to trigger a return from Snooze mode to Normal mode in IELSRn.IELS.
2. Select the CPU as the interrupt request destination.
3. Enable the interrupt in the NVIC.

Note: In Snooze mode, a clock is supplied to ICU. If an event selected in IELSRn is detected, the CPU can acknowledge the interrupt after returning to Normal mode from Software Standby mode. If an event selected in DELSRn is detected, the DMAC can acknowledge the interrupt after returning to Normal mode from Software Standby mode.

14.7 Using the WFI Instruction with Non-Maskable Interrupts

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

14.8 Reference

ARM® Cortex®-M4 Processor Technical Reference Manual (ARM DDI 0439D).

15. Buses

15.1 Overview

Table 15.1 lists the bus specifications, Figure 15.1 shows the bus configuration, and Table 15.2 lists the addresses assigned for each bus.

Table 15.1 Bus specifications

Bus type		Specifications
Main bus	ICode bus (CPU)	• Connected to the CPU • Connected to the on-chip memory (code flash memory, SRAMHS).
	DCode bus (CPU)	• Connected to the CPU • Connected to the on-chip memory (code flash memory, SRAMHS).
	System bus (CPU)	• Connected to the CPU • Connected to the on-chip memory, internal peripheral buses, and external bus.
	DMA bus	• Connected to the DMAC and DTC • Connected to the on-chip memory, internal peripheral buses, and external bus.
	ETHER bus	• Connected to the EDMAC • Connected to the on-chip memory, internal peripheral buses, and external bus.
	GPX bus	• Connected to the JPEG, GLCDC, and DRW • Connected to the on-chip memory and external bus.
Slave interface	Memory bus 1	• Connected to code flash memory
	Memory bus 2	• Connected to the SRAMHS
	Memory bus 3	• Connected to code flash memory and SRAMHS through the DMA bus, ETHER bus, and GPX bus
	Memory bus 4	• Connected to SRAM0
	Memory bus 5	• Connected to SRAM1 and the Standby SRAM
	Internal peripheral bus 1	• Connected to system control related to peripheral modules
	Internal peripheral bus 3	• Connected to peripheral modules (CAC, ELC, I/O ports, POEG, RTC, WDT, IWDT, IIC, CAN, SSIE, SRC, ADC12, DAC12, TSN, and DOC)
	Internal peripheral bus 4	• Connected to peripheral modules (GPT, ETHERC, EPTPC, EDMAC, USBHS, SCI, IrDA, SPI, CRC, and SDHI)
	Internal peripheral bus 5	• Connected to peripheral modules (KINT, AGT, USBFS, PDC, ACMPHS, and CTSU)
	Internal peripheral bus 7	• Connected to Secure IPs (SCE7)
	Internal peripheral bus 8	• Connected to graphic IPs (JPEG, GLCDC, and DRW)
	Internal peripheral bus 9	• Connected to flash memory (in P/E)*[1], data flash memory, and TSN
External bus	CS area	• Connected to the external devices
	SDRAM area	• Connected to SDRAM
	QSPI area	• Connected to the external SPI devices

Note 1. P/E: Programming and erasure.

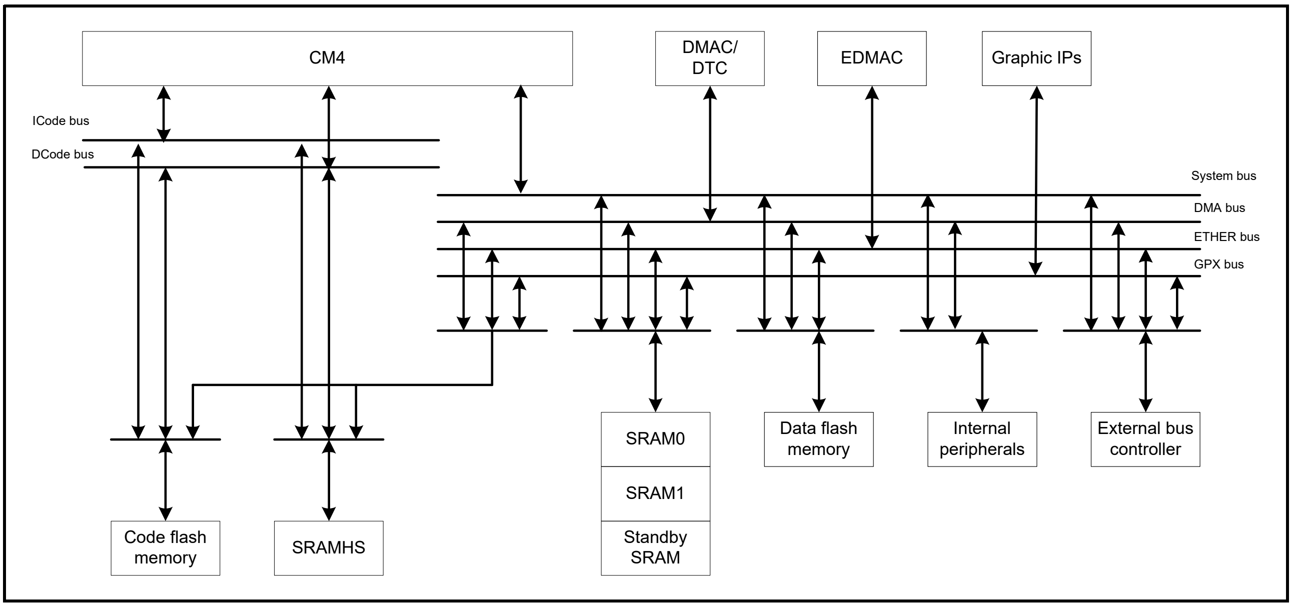


Figure 15.1 Bus configuration

Table 15.2 Addresses assigned for each bus

Addresses	Bus	Area
0000 0000h to 01FF FFFFh	Memory bus 1, 3	Code flash memory
1FFE 0000h to 1FFF FFFFh	Memory bus 2, 3	SRAMHS
2000 0000h to 2003 FFFFh	Memory bus 4	SRAM0
2004 0000h to 200F FFFFh	Memory bus 5	SRAM1 and Standby SRAM
4000 0000h to 4001 FFFFh	Internal peripheral bus 1	Peripheral I/O registers
4004 0000h to 4005 FFFFh	Internal peripheral bus 3	
4006 0000h to 4007 FFFFh	Internal peripheral bus 4	
4008 0000h to 4009 FFFFh	Internal peripheral bus 5	
400C 0000h to 400D FFFFh	Internal peripheral bus 7	Secure IPs
400E 0000h to 400F FFFFh	Internal peripheral bus 8	Graphic IPs (JPEG, GLCDC, and DRW)
4010 0000h to 407F FFFFh	Internal peripheral bus 9	Flash memory (in P/E*[1]), data flash memory, and TSN
6000 0000h to 67FF FFFFh	External bus	QSPI area
8000 0000h to 97FF FFFFh	External bus	CS area and SDRAM area

Note 1. P/E: Programming and erasure.

15.2 Description of Buses

15.2.1 Main Buses

The main buses for the CPU constitute the ICode bus, DCode bus, and system bus.

- The ICode and DCode buses are connected to the code flash memory and SRAMHS. The ICode bus is used for instruction access to the CPU, and the DCode bus is used for data access to the CPU.
- The system bus is connected to the SRAM0, SRAM1, Standby SRAM, data flash memory, internal peripheral buses, and external bus. It is used for instruction and data accesses to the CPU.

The main bus for modules other than the CPU consists of the DMA bus, ETHER bus, and GPX bus.

- The DMA bus is connected to the code flash memory, SRAMHS, SRAM0, SRAM1, Standby SRAM, data flash memory, and external bus.

- The ETHER bus is connected to the code flash memory, SRAMHS, SRAM0, SRAM1, Standby SRAM, data flash memory, and external bus.
- The GPX bus is connected to the code flash memory, SRAMHS, SRAM0, SRAM1, Standby SRAM, and external bus.

Different master and slave transfer combinations can proceed simultaneously.

Arbitration between the DMAC and DTC for the mastership of the DMA bus occurs in the DMAC and DTC. The following fixed-priority order is used:

DMAC0 > DMAC1 > DMAC2 > DMAC3 > DMAC4 > DMAC5 > DMAC6 > DMAC7 > DTC.

Only one DTC and DMAC channels that have accepted the activation requests can issue bus mastership requests. In addition, requests for bus access from masters other than the DTC are not accepted during reads of transfer control information for the DTC.

Requests for mastership of the GPX bus from the JPEG, GLCDC, and DRW are arbitrated. The arbitration protocol is selectable as either fixed-priority or round-robin. For more information, see section 15.3.20, Slave Bus Control Register (BUSSCNT<slave>).

15.2.2 Slave Interface

Products using the Cortex®-M4 core contain ICode and DCode bus areas and a system bus area.

To create the ICode and DCode bus areas, a bus matrix connects the ICode bus, DCode bus, and memory bus 3 from the main bus to the slave interfaces of the code flash memory and SRAMHS. Bus access to the slave interfaces is arbitrated between the three buses. The arbitration protocol is selectable as either fixed-priority or round-robin. For more information, see section 15.3.20, Slave Bus Control Register (BUSSCNT<slave>).

To create the system bus area, a bus matrix connects the system bus, DMA bus, ETHER bus, and GPX bus from the main bus to the slave interfaces of the SRAM0, SRAM1, Standby SRAM, data flash memory, internal peripherals, and external bus. Bus access to the slave interfaces is arbitrated between the four buses. The arbitration protocol is selectable as either fixed-priority or round-robin. For more information, see section 15.3.20, Slave Bus Control Register (BUSSCNT<slave>).

For connections from the main bus to the slave interfaces, see the slave interfaces in Table 15.1. For a description of the external bus, see section 15.2.3, External Bus.

Different master and slave transfer combinations can proceed simultaneously.

15.2.3 External Bus

The external bus controller arbitrates requests for bus access on the external address space from the CPU system bus, DMAC bus, ETHER bus, and GPX bus. The priority order can be set using the external bus priority control bits (BUSSCNT.ARBMET[1:0]). For more information, see section 15.3.20, Slave Bus Control Register (BUSSCNT<slave>).

The bus system provides an external space for the QSPI. See section 39, Quad Serial Peripheral Interface (QSPI).

Table 15.3 lists the external bus specifications and Table 15.4 lists the I/O pins.

Table 15.3 External bus specifications (1 of 2)

Parameter	Specifications
External address space	• The external address space is divided into 8 CS areas (CS0 to CS7) and the SDRAM area (SDCS) for management • Chip select signals can be output for each area • The bus width can be set for each area: - Separate bus: Selectable to 8-bit or 16-bit bus space - Address/data multiplexed bus: Selectable to 8-bit or 16-bit bus space • Endian mode can be specified for each area.

Table 15.3 External bus specifications (2 of 2)

Parameter	Specifications
CS area controller	• Recovery cycles can be inserted: - Read recovery: Up to 15 cycles - Write recovery: Up to 15 cycles • Cycle wait function: Wait for up to 31 cycles (for page access, up to 7 cycles) • Use wait control to set up: - Assertion and negation timing of chip select signals (CS0 to CS7) - Assertion timing of the read signal (RD) and write signals (WR0/WR and WR1) - Timing of data output starts and ends • Write access modes: - Single-write strobe mode and byte strobe mode • Separate bus or address/data multiplexed bus can be set for each area.
SDRAM area controller	• Multiplexed output of row address and column address (8, 9, 10, or 11 bits) • Self-refresh and auto-refresh selectable • CAS latency can be specified from 1 to 3 cycles.
Write buffer function	When write data from the bus master is written to the write buffer, write access by the bus master is complete
Frequency	• The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK)*1 • The frequency of the EBCLK pin output is the same as BCLK by default. Half of the BCLK cycles can be supplied by setting the EBCLK Pin Output Select bit, BCKCR.BCLKDIV, in the External Bus Clock Control Register. For more information, see section 9, Clock Generation Circuit. • The SDRAM area controller (SDRAMC) operates in synchronization with the SDRAM clock (SDCLK).

Note 1. BCLK and SDCLK must operate at the same frequency when the SDRAM is in use.

Table 15.4 External bus I/O pins (1 of 2)

Pin name	I/O		Description
EBCLK, SDCLK*1	Output	CSC, SDRAMC	Clock output pin
A23 to A00*2	Output	CSC, SDRAMC	Address output pins
D15 to D00 DQ15 to DQ00	I/O	CSC, SDRAMC	D15 to D00 are CSC data input/output pins DQ15 to DQ00 are SDRAMC data input/output pins • D015 to D00, DQ15 to DQ00 pins are enabled when the 16-bit bus space is specified • D07 to D00, DQ07 to DQ00 pins are enabled when the 8-bit bus space is specified.
BC0	Output	CSC	• Strobe signal that indicates (when low) that D07 to D00 are valid during access to an external address space in single-write strobe mode, active-low • When an 8-bit bus space is specified, this output pin is always held low regardless of the write access mode.
BC1	Output	CSC	• Strobe signal that indicates (when low) that D15 to D08 are valid during access to an external address space in single-write strobe mode, active-low • This pin is not used when the 8-bit bus space is specified.
CS0*3	Output	CSC	Chip select signal for area 0 (CS0), active-low
CS1*3	Output	CSC	Chip select signal for area 1 (CS1), active-low
CS2*3	Output	CSC	Chip select signal for area 2 (CS2), active-low
CS3*3	Output	CSC	Chip select signal for area 3 (CS3), active-low
CS4	Output	CSC	Chip select signal for area 4 (CS4), active-low
CS5	Output	CSC	Chip select signal for area 5 (CS5), active-low
CS6	Output	CSC	Chip select signal for area 6 (CS6), active-low
CS7	Output	CSC	Chip select signal for area 7 (CS7), active-low
RD	Output	CSC	Strobe signal that indicates that a read from an external address space (CS0 to CS7) is in progress, active-low

Table 15.4 External bus I/O pins (2 of 2)

Pin name	I/O		Description
WR0/WR*4	Output	CSC	• WR0 signal is a strobe signal that indicates that a write to an external address space is in progress in byte strobe mode, and D07 to D00 are valid, active-low • WR signal is a strobe signal that indicates that a write to an external address space is in progress in single-write strobe mode, active-low • When an 8-bit bus space is specified, this output pin is held low during a write access regardless of the write access mode.
WR1	Output	CSC	• Strobe signal that indicates that D15 to D08 are valid during a write to an external address space in byte strobe mode, active-low • This signal is invalid in single-write strobe mode • This pin is not used when the 8-bit bus space is specified.
ALE	Output	CSC	Address latch signal when address/data multiplexed bus is selected
WAIT	Input	CSC	Wait request signal used when accessing the external address space (CS0 to CS7), active-low
CKE	Output	SDRAMC	Clock enable signal
SDCS	Output	SDRAMC	Chip select signal, active-low
RAS	Output	SDRAMC	Row address strobe signal, active-low
CAS	Output	SDRAMC	Column address strobe signal, active-low
WE	Output	SDRAMC	Write enable signal, active-low
DQM0	Output	SDRAMC	I/O data mask enable signal for DQ07 to DQ00
DQM1	Output	SDRAMC	I/O data mask enable signal for DQ15 to DQ08

Note 1. The EBCLK and the SDCLK pin functions are shared by the CS area controller (CSC) and the SDRAM area controller (SDRAMC).
When using the CSC and the SDRAMC simultaneously, the SDCLK pin function is valid.
Note 2. The A23 to A00 pin functions are shared by the CSC and the SDRAMC.
When using the CSC only:
The A00 and BC0 pin functions share the same pin, and either becomes effective according to the area, with the function being A00 in byte strobe mode and BC0 in single-write strobe mode. Setting the 8-bit external bus width is prohibited in single-write strobe mode.
When using the SDRAMC only:
The A15 to A00 pin functions are valid.
The A00 and DQM1 pin functions share the same pin, and either becomes effective according to the external bus width.
When selecting 8-bit bus width, the pin function is A00. When selecting 16-bit bus width, the pin function is DQM1.
When using the CSC and the SDRAMC simultaneously:
The A23 to A16 pin functions are valid for CSC. The A15 to A00 pin functions are shared by the CSC and the SDRAMC.
In the SDRAMC functions, the A00 and the DQM1 pin function works as described above.
In the CSC functions, the A00 and the BC0 pin function works as described above.
Note 3. The CS0 to CS3 (CSC) and SDRAMC pin functions share the same pin. When using the CSC and the SDRAMC simultaneously, the CS0 to CS3 pin functions are invalid.
Note 4. The WR0 signal and WR signal are identical. The WR0 signal is referred to as WR in single-write strobe mode.

15.2.4 Parallel Operations

Parallel operations are possible when different bus masters request access to different slave modules. For example, if the CPU fetches an instruction from the flash and an operand from the SRAM, the DMAC can handle transfers between a peripheral bus and the external bus at the same time.

An example of parallel operations is shown in Figure 15.2. In this example, the CPU uses the instruction and operand buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DMAC/DTC, EDMAC, and JPEG/GLCDC/DRW simultaneously use the DMA bus (DMAC/DTC), ETHER bus (EDMAC), and GPX bus (JPEG/GLCDC/DRW) for access to a peripheral bus or external bus during access to the flash memory and SRAM by the CPU.

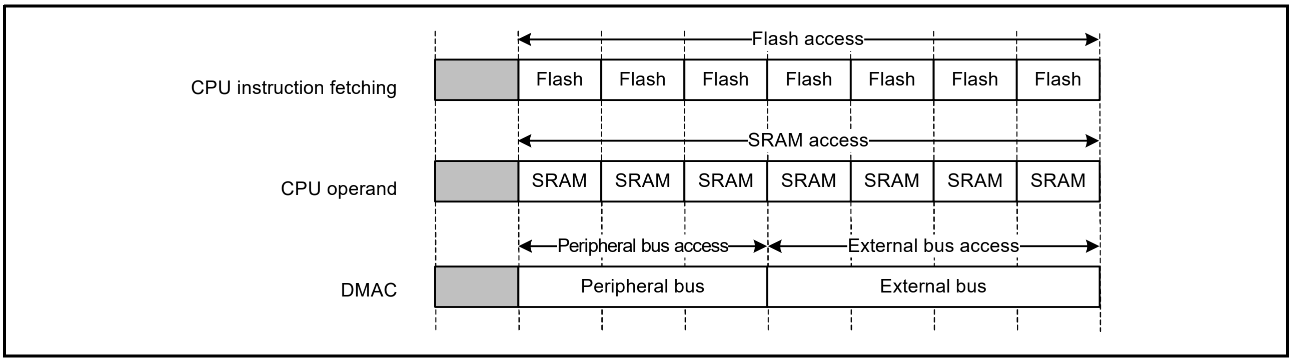


Figure 15.2 Example of parallel operations

15.2.5 Bus Settings

Set up the external bus with the following registers:

- Mode settings:
  CSn Mode Register (CSnMOD), CSn Wait Control Register 1 (CSnWCR1), CSn Wait Control Register 2 (CSnWCR2), CSn Control Register (CSnCR), CSn Recovery Cycle Setting Register (CSnREC), CS Recovery Cycle Insertion Enable Register (CSRECEN), and Bus Priority Control Register (BUSSCNT)
- I/O port assignments:
  PmnPFS.PMR = 1 and PmnPFS.PSEL[4:0] = 0Bh
- Frequency of the external bus clock (BCLK) and SDRAM clock (SDCLK):
  SCKDIVCR register.

See section 20, I/O Ports, for information on PmnPFS and section 9, Clock Generation Circuit for information on SCKDIVCR.

15.2.6 Restrictions

(1) Endianness constraint

Memory space must be little-endian to execute code on the Cortex-M4 core.

15.3 Register Descriptions

15.3.1 CSn Control Register (CSnCR) (n = 0 to 7)

Address(es): BUS.CS0CR 4000 3802h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Address(es): BUS.CS1CR 4000 3812h, BUS.CS2CR 4000 3822h, BUS.CS3CR 4000 3832h, BUS.CS4CR 4000 3842h, BUS.CS5CR 4000 3852h, BUS.CS6CR 4000 3862h, BUS.CS7CR 4000 3872h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MPXEN	—	—	—	EMODE	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EXENB	Operation Enable	0: Disable operation 1: Enable operation.	R/W

Bit	Symbol	Bit name	Description	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: 16-bit bus space 0 1: Setting prohibited 1 0: 8-bit bus space 1 1: Setting prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Little-endian 1: Big-endian.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	MPXEN	Address/Data Multiplexed I/O Interface Select	0: Separate bus interface is selected for area n 1: Address/data multiplexed I/O interface is selected for area n. (n = 0 to 7).	
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write to the CSnCR register while the external bus is being accessed.

EXENB bit (Operation Enable)

The EXENB bit enables operation of the associated CS area. On MCU reset, operation is enabled (EXENB = 1) only for area 0. Operation in other areas is disabled (EXENB = 0). Attempts to access disabled areas have no effect.

When the CSC and SDRAMC are in use at the same time, BCLK and SDCLK must operate at the same frequency.

BSIZE[1:0] bits (External Bus Width Select)

The BSIZE[1:0] bits specify the data bus width for the associated area.

EMODE bit (Endian Mode)

The EMODE bit specifies the endianness for the associated area. The Cortex-M4 core is fixed at little-endian order, so instruction code can only be allocated to external spaces with little-endian specified. If an area is specified as big-endian, no instruction code can be allocated to it.

MPXEN bit (Address/Data Multiplexed I/O Interface Select)

Th MPXEN bit specifies separate bus interface or address/data multiplexed I/O interface of each area.

15.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 7)

Address(es): BUS.CS0REC 4000 380Ah, BUS.CS1REC 4000 381Ah, BUS.CS2REC 4000 382Ah, BUS.CS3REC 4000 383Ah, BUS.CS4REC 4000 384Ah, BUS.CS5REC 4000 385Ah, BUS.CS6REC 4000 386Ah, BUS.CS7REC 4000 387Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WRCV[3:0]				—	—	—	—	RRCV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	RRCV[3:0]	Read Recovery	b3 b0 0 0 0 0: Do not insert any recovery cycles 0 0 0 1: Insert 1 recovery cycle 0 0 1 0: Insert 2 recovery cycles 0 0 1 1: Insert 3 recovery cycles 0 1 0 0: Insert 4 recovery cycles 0 1 0 1: Insert 5 recovery cycles 0 1 1 0: Insert 6 recovery cycles 0 1 1 1: Insert 7 recovery cycles 1 0 0 0: Insert 8 recovery cycles 1 0 0 1: Insert 9 recovery cycles 1 0 1 0: Insert 10 recovery cycles 1 0 1 1: Insert 11 recovery cycles 1 1 0 0: Insert 12 recovery cycles 1 1 0 1: Insert 13 recovery cycles 1 1 1 0: Insert 14 recovery cycles 1 1 1 1: Insert 15 recovery cycles.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	WRCV[3:0]	Write Recovery	b11 b8 0 0 0 0: Do not insert any recovery cycles 0 0 0 1: Insert 1 recovery cycle 0 0 1 0: Insert 2 recovery cycles 0 0 1 1: Insert 3 recovery cycles 0 1 0 0: Insert 4 recovery cycles 0 1 0 1: Insert 5 recovery cycles 0 1 1 0: Insert 6 recovery cycles 0 1 1 1: Insert 7 recovery cycles 1 0 0 0: Insert 8 recovery cycles 1 0 0 1: Insert 9 recovery cycles 1 0 1 0: Insert 10 recovery cycles 1 0 1 1: Insert 11 recovery cycles 1 1 0 0: Insert 12 recovery cycles 1 1 0 1: Insert 13 recovery cycles 1 1 1 0: Insert 14 recovery cycles 1 1 1 1: Insert 15 recovery cycles.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write to the CSnREC register while the external bus is being accessed.

When the preceding bus access is from a separate bus, CSnREC is valid when the recovery cycle insertion is enabled in the Separate Bus Recovery Cycle Insertion Enable bit (RCVENi (i = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the Multiplexed Bus Recovery Cycle Insertion Enable bit (RCVENMj (j = 0 to 7)) in CSRECEN. For more information, see section 15.5.4, Insertion of Recovery Cycles.

RRCV[3:0] bits (Read Recovery)

The RRCV[3:0] bits specify the number of recovery cycles inserted after a read access on the external bus for CSn (n = 0 to 7). When recovery cycle insertion is enabled and a value other than 0000b is set, 1 to 15 recovery cycles are inserted when:

- After a read access to the external bus, a read access is made to the external bus in the same area

- After a read access to the external bus, a read access is made to the external bus in a different area
- After a read access to the external bus, a write access is made to the external bus in the same area
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] bits (Write Recovery)

The WRCV[3:0] bits specify the number of recovery cycles inserted after a write access on the external bus for CSn (n = 0 to 7). When recovery cycle insertion is enabled and a value other than 0000b is set, 1 to 15 recovery cycles are inserted when:

- After a write access to the external bus, a read access is made to the external bus in the same area
- After a write access to the external bus, a read access is made to the external bus in a different area
- After a write access to the external bus, a write access is made to the external bus in the same area
- After a write access to the external bus, a write access is made to the external bus in a different area.

15.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): BUS.CSRECEN 4000 3880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0	0: Disable 1: Enable.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1	0: Disable 1: Enable.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Disable 1: Enable.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Disable 1: Enable.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Disable 1: Enable.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Disable 1: Enable.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Disable 1: Enable.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Disable 1: Enable.	R/W
b8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0	0: Disable 1: Enable.	R/W
b9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1	0: Disable 1: Enable.	R/W
b10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2	0: Disable 1: Enable.	R/W
b11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3	0: Disable 1: Enable.	R/W
b12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4	0: Disable 1: Enable.	R/W
b13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5	0: Disable 1: Enable.	R/W

Bit	Symbol	Bit name	Description	R/W
b14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6	0: Disable 1: Enable.	R/W
b15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7	0: Disable 1: Enable.	R/W

Do not attempt to write to the CSRECEN register while the external bus is being accessed. For more information on insertion recovery cycles, see 15.5.4 Insertion of Recovery Cycles.

RCVENi bit (Separate Bus Recovery Cycle Insertion Enable i) (i = 0 to 7)

This bit enables the insertion of read or write recovery cycles when, after a read or write access on the external bus, a read or write access is made on the external bus to the same or different area.

RCVENMj bit (Multiplexed Bus Recovery Cycle Insertion Enable j) (j = 0 to 7)

This bit enables the insertion of read or write recovery cycles when, after a read or write access on the external bus, a read or write access is made on the external bus to the same or different area.

Table 15.5 Insertion of recovery cycles

Access type	External address space	Insertion of recovery cycles	Corresponding bits (Separate/Multiplexed)
Read access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN0/RCVENM0
	Different area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN1/RCVENM1
Write access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN2/RCVENM2
	Different area	Recovery cycles specified in the RRCV[3:0] bits are inserted for the priority access area	RCVEN3/RCVENM3
Read access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN4/RCVENM4
	Different area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN6/RCVENM6
	Different area	Recovery cycles specified in the WRCV[3:0] bits are inserted for the priority access area	RCVEN7/RCVENM7

15.3.4 CSn Mode Register (CSnMOD) (n = 0 to 7)

Address(es): BUS.CS0MOD 4000 3002h, BUS.CS1MOD 4000 3012h, BUS.CS2MOD 4000 3022h, BUS.CS3MOD 4000 3032h, BUS.CS4MOD 4000 3042h, BUS.CS5MOD 4000 3052h, BUS.CS6MOD 4000 3062h, BUS.CS7MOD 4000 3072h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single-write strobe mode.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: Disable 1: Enable.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Disable 1: Enable.	R/W
b9	PWENB	Page Write Access Enable	0: Disable 1: Enable.	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode.	R/W

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD bit (Write Access Mode Select)

The WRMOD bit selects the write access operating mode. Writing 0 selects byte strobe mode, in which data writes are controlled by the WRn signals (n = 0 to 1) associated with the respective byte positions. Writing 1 selects single-write strobe mode, in which data writes are controlled by the BCn (n = 0 to 1) and WR signals associated with the respective byte positions.

Note: Setting the external bus width to 8 bits is prohibited in single-write strobe mode.

Table 15.6 Control signals for write access modes

	Pin name			
Write access mode	**WR1**	**WR0/WR**	**BC1**	**BC0**
Byte strobe mode	✓	✓ (WR0)	×	×
Single-write strobe mode	×	✓ (WR)	✓	✓

✓: Enabled, ×: Disabled

EWENB bit (External Wait Enable)

The EWENB bit enables external waits. Writing 0 disables the WAIT signal. Writing 1 selects external wait and allows the WAIT signal to control the number of waits per cycle. In this state, wait cycles are inserted when the WAIT signal is low.

PRENB bit (Page Read Access Enable)

The PRENB bit enables page read accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, PRENB should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PWENB bit (Page Write Access Enable)

The PWENB bit enables page write accesses.

Note: When the address/data multiplexed I/O interface is selected with the CSnCR.MPXEN bit, PWENB should not be set to enable page write accesses. Page write accesses are not supported in the address/data multiplexed I/O interface.

PRMOD bit (Page Read Access Mode Select)

The PRMOD bit selects the operating mode for page read accesses. Writing 0 selects normal access compatible mode, in which the RD signal is negated and an RD assert wait is inserted each time a unit of data is read. When there is no RD assert wait, the RD signal is negated only in the final transfer of the external bus access.

Writing 1 selects external data read continuous assertion mode, in which an RD assert wait is inserted and the RD signal is continuously asserted during the wait.

15.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7)

Address(es): BUS.CS0WCR1 4000 3004h, BUS.CS1WCR1 4000 3014h, BUS.CS2WCR1 4000 3024h, BUS.CS3WCR1 4000 3034h, BUS.CS4WCR1 4000 3044h, BUS.CS5WCR1 4000 3054h, BUS.CS6WCR1 4000 3064h, BUS.CS7WCR1 4000 3074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSPRWAIT[2:0]			—	—	—	—	—	CSPWWAIT[2:0]		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b0 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b8 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: Do not insert wait 0 0 0 0 1: Insert wait of 1 clock cycle 0 0 0 1 0: Insert wait of 2 clock cycles 0 0 0 1 1: Insert wait of 3 clock cycles ... 1 1 1 0 1: Insert wait of 29 clock cycles 1 1 1 1 0: Insert wait of 30 clock cycles 1 1 1 1 1: Insert wait of 31 clock cycles.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28 b24 0 0 0 0 0: Do not insert wait 0 0 0 0 1: Insert wait of 1 clock cycle 0 0 0 1 0: Insert wait of 2 clock cycles 0 0 0 1 1: Insert wait of 3 clock cycles. ... 1 1 1 0 1: Insert wait of 29 clock cycles 1 1 1 1 0: Insert wait of 30 clock cycles 1 1 1 1 1: Insert wait of 31 clock cycles.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CSPWWAIT[2:0] value is only valid when the CSnMOD.PWENB bit is set to 1.

Note 2. The CSPRWAIT[2:0] value is only valid when the CSnMOD.PRENB bit is set to 1.

Do not attempt to write to the CSnWCR1 register while the external bus is being accessed. Set each of these bits within a range of the restrictions described in section 15.5.7, Constraints, (1) Constraints on using a separate bus interface or section 15.5.7, Constraints, (2) Constraints on using address/data multiplexed bus interface, according to the bus interface used.

CSPWWAIT[2:0] bits (Page Write Cycle Wait Select)

The CSPWWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle. The setting is enabled when the CSnMOD.PWENB bit is set to 1.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] bits (Page Read Cycle Wait Select)

The CSPRWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle. The setting is enabled when the CSnMOD.PRENB bit is set to 1.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] bits (Normal Write Cycle Wait Select)

The CSWWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] bits (Normal Read Cycle Wait Select)

The CSRWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

15.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7)

Address(es): BUS.CS0WCR2 4000 3008h, BUS.CS1WCR2 4000 3018h, BUS.CS2WCR2 4000 3028h, BUS.CS3WCR2 4000 3038h, BUS.CS4WCR2 4000 3048h, BUS.CS5WCR2 4000 3058h, BUS.CS6WCR2 4000 3068h, BUS.CS7WCR2 4000 3078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AWAIT[1:0]		—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b0 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b4 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b8 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	AWAIT[1:0]	Address Cycle Wait Select	b13 b12 0 0: Do not insert wait 0 1: Insert wait of 1 clock cycle 1 0: Insert wait of 2 clock cycles 1 1: Insert wait of 3 clock cycles.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b18 to b16	RDON[2:0]	RD Assert Wait Select	b18 b16 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Do not attempt to write to the CSnWCR2 register while the external bus is being accessed. Set each of these bits within a range of the restrictions described in section 15.5.7, Constraints, (1) Constraints on using a separate bus interface. or section 15.5.7, Constraints, (2) Constraints on using address/data multiplexed bus interface, according to the bus interface used.

CSROFF[2:0] bits (Read-Access CS Extension Cycle Select)

The CSROFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (RD signal negated) until the CSn signal (n = 0 to 7) is negated in read access mode.

CSWOFF[2:0] bits (Write-Access CS Extension Cycle Select)

The CSWOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0, 1) negated) until the CSn signal (n = 0 to 7) is negated in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

WDOFF[2:0] bits (Write Data Output Extension Cycle Select)

The WDOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0, 1) negated) until the write-data output is complete in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

AWAIT[1:0] bits (Address Cycle Wait Select)

The AWAIT[1:0] bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.
For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

RDON[2:0] bits (RD Assert Wait Select)

The RDON[2:0] bits specify the number of wait cycles to be inserted before the RD signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

WRON[2:0] bits (WR Assert Wait Select)

The WRON[2:0] bits specify the number of wait cycles to be inserted before the WRn signal (n = 0, 1) is asserted.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

WDON[2:0] bits (Write Data Output Wait Select)

The WDON[2:0] bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSON[2:0] bits (CS Assert Wait Select)

The CSON[2:0] bits specify the number of wait cycles to be inserted before the CSn signal (n = 0 to 7) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.CSON[2:0] value ≤

CSnWCR2.AWAIT[1:0] value.

15.3.7 SDC Control Register (SDCCR)

Address(es): BUS.SDCCR 4000 3C00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EXENB	Operation Enable	0: Disable 1: Enable.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	SDRAM Bus Width Select	b5 b4 0 0: 16-bit bus space 0 1: Setting prohibited 1 0: 8-bit bus space 1 1: Setting prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

EXENB bit (Operation Enable)

The EXENB bit enables the operation of the SDRAM address space. On reset, operation is disabled (EXENB = 0). Attempts to access disabled areas have no effect.

When CSC and SDRAMC are in use at the same time, BCLK and SDCLK must operate at the same frequency.

15.3.8 SDC Mode Register (SDCMOD)

Address(es): BUS.SDCMOD 4000 3C01h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EMODE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EMODE	Endian Mode	0: Endian order of SDRAM address space is the same as the endian order of the operating mode 1: Endian order of SDRAM address space is not the endian order of the operating mode.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

EMODE bit (Endian Mode)

The EMODE bit specifies the endianness for the SDRAM address space. The Cortex-M4 core is fixed at little-endian order, so instruction code can only be allocated to external spaces with little-endian specified. If an area is specified as big-endian, no instruction code can be allocated to it.

15.3.9 SDRAM Access Mode Register (SDAMOD)

Address(es): BUS.SDAMOD 4000 3C02h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BE	Continuous Access Enable	0: Disable 1: Enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set the SDAMOD register only when the conditions in Table 15.14 are satisfied. Otherwise, operation is not guaranteed.

BE bit (Continuous Access Enable)

The BE bit enables continuous access to the SDRAM access space.

Note: When the SDRAM area is accessed from bus masters other than graphic IPs, continuous access is always disabled regardless of the setting.

15.3.10 SDRAM Self-Refresh Control Register (SDSELF)

Address(es): BUS.SDSELF 4000 3C10h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SFEN	SDRAM Self-Refresh Enable	0: Disable 1: Enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set the SDSELF register only when the conditions in Table 15.14 are satisfied. Otherwise, operation is not guaranteed.

SFEN bit (SDRAM Self-Refresh Enable)

The SFEN bit controls the self-refresh operation. Setting this bit to 1 initiates an auto-refresh cycle, after which self-refresh begins. Clearing this bit to 0 ends the self-refresh, and auto-refresh resumes. When the bit is set to 1, the write value takes effect when the self-refresh operation starts. When it is cleared to 0, the write value has already taken effect when auto-refresh starts after the end of the self-refresh.

15.3.11 SDRAM Refresh Control Register (SDRFCR)

Address(es): BUS.SDRFCR 4000 3C14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFW[3:0]				RFC[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b11 to b0	RFC[11:0]	Auto-Refresh Request Interval Setting	b11 b0 0 0 0 0 0 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 0 0 0 0 0 1: 2 cycles 0 0 0 0 0 0 0 0 0 0 1 0: 3 cycles : 1 1 1 1 1 1 1 1 1 1 1 1: 4096 cycles.	R/W
b15 to b12	REFW[3:0]	Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting	b15 b12 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles 0 0 1 0: 3 cycles 0 0 1 1: 4 cycles 0 1 0 0: 5 cycles 0 1 0 1: 6 cycles 0 1 1 0: 7 cycles 0 1 1 1: 8 cycles 1 0 0 0: 9 cycles 1 0 0 1: 10 cycles 1 0 1 0: 11 cycles 1 0 1 1: 12 cycles 1 1 0 0: 13 cycles 1 1 0 1: 14 cycles 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles.	R/W

RFC[11:0] bits (Auto-Refresh Request Interval Setting)

The RFC[11:0] bits specify the auto-refresh request interval. They can be written to at any time, regardless of the state of the Auto-Refresh Operation Enable bit (RFEN) in SDRFEN. If auto-refresh is enabled, the write value takes effect after the end of the auto-refresh cycles. The refresh counter uses SDCLK.

REFW[3:0] bits (Auto-Refresh Cycle/Self-Refresh Clearing Cycle Count Setting)

The REFW[3:0] bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles. They can be written to at any time, regardless of the state of the Auto-Refresh Operation Enable bit (RFEN) in SDRFEN. If an auto-refresh cycle is in progress, the value written to the bits while auto-refresh is enabled takes effect after the cycle completes.

Note: Auto-refresh requests are not accepted while the SDRAM is being accessed. This means they must sometimes wait until the access completes for the auto-refresh interval to be extended. Set the RFC[11:0] bits to an auto-refresh request interval value that meets the specifications of the SDRAM being used. Additionally, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle. The auto-refresh interval cannot be automatically adjusted when the frequency is changed during operation. In this case, perform a self-refresh operation and set the auto-refresh interval to an appropriate value for the frequency again.

15.3.11.1 Auto-refresh request interval and RFC set value

The SDRAMC (SDRAM area controller) includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. Use the following equation to calculate the set value for the RFC[11:0] bits from the auto-refresh request interval:

$$\text{RFC} = (\text{Auto-refresh request interval} / \text{SDCLK cycle}) - 1$$

Note: Auto-refresh requests are not accepted while the SDRAM is being accessed. They must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. If two or more auto-refresh requests are generated while the SDRAM is being accessed, the second and subsequent requests are ignored.

15.3.12 SDRAM Auto-Refresh Control Register (SDRFEN)

Address(es): BUS.SDRFEN 4000 3C16h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RFEN	Auto-Refresh Operation Enable	0: Disable 1: Enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RFEN bit (Auto-Refresh Operation Enable)

The RFEN bit enables auto-refresh operation. When auto-refresh is required, set the RFEN bit to 1 before SDRAM access.

Clearing this bit to 0 while auto-refreshing is enabled causes RFEN to be cleared to 0 and auto-refresh operation to halt after the end of the auto-refresh cycle.The interval at which refresh requests are generated is determined by the value in the Auto-Refresh Request Interval Setting bits (RFC[11:0]) in the SDRAM Refresh Control Register (SDRFCR).

15.3.13 SDRAM Initialization Sequence Control Register (SDICR)

Address(es): BUS.SDICR 4000 3C20h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	INIRQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	INIRQ	Initialization Sequence Start	0: Invalid 1: Start initialization sequence.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

INIRQ bit (Initialization Sequence Start)

Setting the INIRQ bit to 1 starts the SDRAM initialization sequence and automatically sets the Initialization Status bit (INIST) in the SDRAM Status Register (SDSR) to 1. The INIST bit clears automatically after the initialization sequence ends. The value written to the INIRQ bit is not retained.

Note: Set the INIRQ bit to start the SDRAM initialization sequence only when the conditions in Table 15.14 are satisfied. Otherwise, operation is not guaranteed.

15.3.14 SDRAM Initialization Register (SDIR)

Address(es): BUS.SDIR 4000 3C24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PRC[2:0]			ARFC[3:0]				ARFI[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	ARFI[3:0]	Initialization Auto-Refresh Interval	b3 b0 0 0 0 0: 3 cycles 0 0 0 1: 4 cycles 0 0 1 0: 5 cycles 0 0 1 1: 6 cycles 0 1 0 0: 7 cycles 0 1 0 1: 8 cycles 0 1 1 0: 9 cycles 0 1 1 1: 10 cycles 1 0 0 0: 11 cycles 1 0 0 1: 12 cycles 1 0 1 0: 13 cycles 1 0 1 1: 14 cycles 1 1 0 0: 15 cycles 1 1 0 1: 16 cycles 1 1 1 0: 17 cycles 1 1 1 1: 18 cycles.	R/W
b7 to b4	ARFC[3:0]	Initialization Auto-Refresh Count	b7 b4 0 0 0 0: Setting prohibited 0 0 0 1: 1 time 0 0 1 0: 2 times 0 0 1 1: 3 times 0 1 0 0: 4 times 0 1 0 1: 5 times 0 1 1 0: 6 times 0 1 1 1: 7 times 1 0 0 0: 8 times 1 0 0 1: 9 times 1 0 1 0: 10 times 1 0 1 1: 11 times 1 1 0 0: 12 times 1 1 0 1: 13 times 1 1 1 0: 14 times 1 1 1 1: 15 times.	R/W
b10 to b8	PRC[2:0]	Initialization Precharge Cycle Count	b10 b8 0 0 0: 3 cycles 0 0 1: 4 cycles 0 1 0: 5 cycles 0 1 1: 6 cycles 1 0 0: 7 cycles 1 0 1: 8 cycles 1 1 0: 9 cycles 1 1 1: 10 cycles.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

ARFI[3:0] bits (Initialization Auto-Refresh Interval)

The ARFI[3:0] bits specify the interval at which the auto-refresh commands are issued in the SDRAM initialization sequence.

ARFC[3:0] bits (Initialization Auto-Refresh Count)

The ARFC[3:0] bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence.

PRC[2:0] bits (Initialization Precharge Cycle Count)

The PRC[2:0] bits specify the number of precharged cycles in the SDRAM initialization sequence.

Note: Implement settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

15.3.15 SDRAM Address Register (SDADR)

Address(es): BUS.SDADR 4000 3C40h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MXC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	MXC[1:0]	Address Multiplex Select	b1 b0 0 0: 8-bit shift 0 1: 9-bit shift 1 0: 10-bit shift 1 1: 11-bit shift.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set SDADR only when the conditions in Table 15.14 are satisfied. Otherwise, operation is not guaranteed.

MXC[1:0] bits (Address Multiplex Select)

The MXC[1:0] bits select the size of the shift towards the lower half of the row address in row address/column address multiplexing. These bits also select the row address bits to be used for comparison in SDRAMC continuous access operation. For details, see Table 15.19.

15.3.16 SDRAM Timing Register (SDTR)

Address(es): BUS.SDTR 4000 3C44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RAS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RCD[1:0]		RP[2:0]			WR	—	—	—	—	—	CL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CL[2:0]	SDRAMC Column Latency	b2 b0 0 0 0: Setting prohibited 0 0 1: 1 cycle 0 1 0: 2 cycles 0 1 1: 3 cycles 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited.	R/W

Bit	Symbol	Bit name	Description	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	WR	Write Recovery Interval	0: 1 cycle 1: 2 cycles.	R/W
b11 to b9	RP[2:0]	Row Precharge Interval	b11 b9 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: 8 cycles.	R/W
b13, b12	RCD[1:0]	Row Column Latency	b13 b12 0 0: 1 cycle 0 1: 2 cycles 1 0: 3 cycles 1 1: 4 cycles.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RAS[2:0]	Row Active Interval	b18 b16 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: Setting prohibited.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SDTR register specifies the timing for read and write accesses to the SDRAM. For more information, see section 15.6.11.3, Timing register settings and access timing.

Set the SDTR register only when the conditions in Table 15.14 are satisfied. Otherwise, operation is not guaranteed.

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

CL[2:0] bits (SDRAMC Column Latency)

The CL[2:0] bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, use the SDRAM mode register (SDMOD).

WR bit (Write Recovery Interval)

The WR bit specifies the interval that must elapse between the SDRAM write command (WRIT) and deactivation (PALL).

RP[2:0] bits (Row Precharge Interval)

The RP[2:0] bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PALL) and the next valid command.

RAS[2:0] bits (Row Active Interval)

The RAS[2:0] bits specify the minimum interval that must elapse between the SDRAM row activation command (ACTV) and deactivation (PALL). The value specified in these bits must be less than or equal to the sum of the row column latency (RCD[1:0]) and column latency (CL[2:0]) settings.

15.3.17 SDRAM Mode Register (SDMOD)

Address(es): BUS.SDMOD 4000 3C48h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MR[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b14 to b0	MR[14:0]	Mode Register Setting	Writing to these bits triggers a mode register set command.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The SDMOD register specifies the value to be written to the SDRAM mode register. Writing to SDMOD causes a mode register set command to be issued automatically to the SDRAM. Set SDMOD only when the conditions in Table 15.14 are satisfied. Otherwise, operation is not guaranteed.

Writing to this register is possible only once after release from reset. Operation is not guaranteed if more than one write access is attempted.

MR[14:0] bits (Mode Register Setting)

Writing to the MR[14:0] bits causes a mode register set command to be issued to the SDRAM, and the setting in the MR[14:0] bits is output to the lower bits of the address. For more information, see section 15.6.10, Setting the Mode Register.

Note: Set a burst length of 1 for the SDRAM. Operation is not guaranteed with any other burst length setting.

Note: The SDRAM column latency must match the setting in the SDRAMC Column Latency setting (CL[2:0]) in the SDRAM Timing Register (SDTR). Operation is not guaranteed if the latency settings do not agree.

Note: Make sure the SRFST, INIST, and MRSST status bits in the SDRAM Status Register (SDSR) are all 0.

15.3.18 SDRAM Status Register (SDSR)

Address(es): BUS.SDSR 4000 3C50h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SRFST	INIST	—	—	MRSST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MRSST	Mode Register Setting Status	0: Mode register setting not in progress 1: Mode register setting in progress.	R
b2, b1	—	Reserved	These bits are read as 0.	R
b3	INIST	Initialization Status	0: Initialization sequence not in progress 1: Initialization sequence in progress.	R
b4	SRFST	Self-Refresh Transition/Recovery Status	0: Transition/recovery not in progress 1: Transition/recovery in progress.	R
b7 to b5	—	Reserved	These bits are read as 0.	R

MRSST bit (Mode Register Setting Status)

When set to 1, the MRSST bit indicates that SDRAM mode register setting is in progress.

INIST bit (Initialization Status)

When set to 1, the INIST bit indicates that the SDRAM initialization sequence is in progress.

SRFST bit (Self-Refresh Transition/Recovery Status)

When set to 1, the SRFST bit indicates that a transition to or recovery from a self-refresh operation is in progress for the SDRAM. The in progress interval begins when the bits in Table 15.7 are written to and lasts until the associated commands are issued.

Note: Execution of a self-refresh, initialization sequence, or mode register setting can only be performed when all the status bits are 0. Do not rewrite the registers and bits in Table 15.7 when any of the SRFST, INIST, or MRSST status bits is set to 1.

Table 15.7 Registers and bits requiring status bit checking

Function	Register	Bits
Self-refresh	SDSELF	SFEN
Initialization sequence	SDICR	INIRQ
Mode register setting	SDMOD	MR[14:0]

15.3.19 Master Bus Control Register (BUSMCNT<master>)

Address(es): BUS.BUSMCNTM4I 4000 4000h, BUS.BUSMCNTM4D 4000 4004h, BUS.BUSMCNTSYS 4000 4008h, BUS.BUSMCNTDMA 4000 400Ch, BUS.BUSMCNTEDM 4000 4010h, BUS.BUSMCNTGPX 4000 4014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IERES	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	IERES	Ignore Error Responses	0: Report bus errors 1: Do not report bus errors.	R/W

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

Table 15.8 shows the registers associated with each bus type.

Table 15.8 Associations between bus types and registers

Bus type	Master Bus Control Register	Slave Bus Control Register	Bus Error Address Register	Bus Error Status Register
ICode bus (CPU)	BUSMCNTM4I	-	BUS1ERRADD	BUS1ERRSTAT
DCode bus (CPU)	BUSMCNTM4D	-	BUS2ERRADD	BUS2ERRSTAT
System bus (CPU)	BUSMCNTSYS	-	BUS3ERRADD	BUS3ERRSTAT
DMA bus	BUSMCNTDMA	-	BUS4ERRADD	BUS4ERRSTAT
EDMAC bus	BUSMCNTEDM	-	BUS5ERRADD	BUS5ERRSTAT
GPX bus from JPEG (data input)	BUSMCNTGPX	BUSSCNTGPX	BUS6ERRADD	BUS6ERRSTAT
GPX bus from JPEG (data output)	BUSMCNTGPX	BUSSCNTGPX	BUS7ERRADD	BUS7ERRSTAT
GPX bus from GLCDC (Graphic1)	BUSMCNTGPX	BUSSCNTGPX	BUS8ERRADD	BUS8ERRSTAT
GPX bus from GLCDC (Graphic2)	BUSMCNTGPX	BUSSCNTGPX	BUS9ERRADD	BUS9ERRSTAT
GPX bus from DRW (texture)	BUSMCNTGPX	BUSSCNTGPX	BUS10ERRADD	BUS10ERRSTAT
GPX bus from DRW (data)	BUSMCNTGPX	BUSSCNTGPX	BUS11ERRADD	BUS11ERRSTAT
Memory bus 1	-	BUSSCNTFLI	-	-
Memory bus 2	-	BUSSCNTRAMH	-	-
Memory bus 3	-	BUSSCNTMBIU	-	-
Memory bus 4	-	BUSSCNTRAM0	-	-
Memory bus 5	-	BUSSCNTRAM1	-	-
Internal peripheral bus 1, 3, 4, 5, 7, 8	-	BUSSCNTPnB (n = 0, 2, 3, 4, 6, 7)	-	-
Internal peripheral bus 9	-	BUSSCNTFBU	-	-
External bus (CS and SDRAM areas)	-	BUSSCNTEXT	-	-
External bus (QSPI area)	-	BUSSCNTEXT2	-	-

IERES bit (Ignore Error Responses)

The IERES bit, when set, disables the AHB-Lite protocol error response.

15.3.20 Slave Bus Control Register (BUSSCNT<slave>)

Address(es): BUS.BUSSCNTFLI 4000 4100h, BUS.BUSSCNTRAMH 4000 4104h, BUS.BUSSCNTMBIU 4000 4108h,
BUS.BUSSCNTRAM0 4000 410Ch, BUS.BUSSCNTRAM1 4000 4110h, BUS.BUSSCNTP0B 4000 4114h,
BUS.BUSSCNTP2B 4000 4118h, BUS.BUSSCNTP3B 4000 411Ch, BUS.BUSSCNTP4B 4000 4120h,
BUS.BUSSCNTP6B 4000 4128h, BUS.BUSSCNTP7B 4000 412Ch, BUS.BUSSCNTFBU 4000 4130h,
BUS.BUSSCNTEXT 4000 4134h, BUS.BUSSCNTEXT2 4000 4138h, BUS.BUSSCNTGPX 4000 413Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EWRES	—	—	ARBMET[1:0]		—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b5, b4	ARBMET[1:0]	Arbitration Method	Specifies the group priorities. b5 b4 0 0: Fixed priority 0 1: Round-robin 1 0: Setting prohibited 1 1: Setting prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EWRES	Early Write Response	0: Disable early write response 1: Enable early write response.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

Table 15.8 lists the registers associated with each bus type.

ARBMET[1:0] bits (Arbitration Method)

The ARBMET[1:0] bits specify the arbitration protocol, with priority defined for all bus masters. For fixed priority, see Table 15.9. For round-robin, see Table 15.10.

EWRES bit (Early Write Response)

The EWRES bit indicates whether the next write request is accepted before the response for the current write transaction occurs. When the value is 1, the next write request is accepted and high-speed transfer is possible, but AHB-Lite error responses are not detected. Bus errors are returned to the requesting master IP using the error response protocol for AHB-Lite. For details on errors that occur on each bus, see section 15.7, Bus Error Monitoring Section. Only use the BUSSCNTMBIU, BUSSCNTP0B, and BUSSCNTEXT registers.

Table 15.9 Bus priorities with fixed-priority arbitration (ARBMET[1:0] = 00b)

Slave Bus Control Register	Slave interface	Priority order
BUSSCNTFLI	Memory bus 1	Memory bus 3 > DCode bus (CPU) > ICode bus (CPU)
BUSSCNTRAMH	Memory bus 2	Memory bus 3 > DCode bus (CPU) > ICode bus (CPU)
BUSSCNTMBIU	Memory bus 3	GPX bus > ETHER bus > DMA bus
BUSSCNTRAM0	Memory bus 4	GPX bus > ETHER bus > DMA bus > system bus (CPU)
BUSSCNTRAM1	Memory bus 5	GPX bus > ETHER bus > DMA bus > system bus (CPU)
BUSSCNTPnB (n = 0, 2, 3, 4, 6, 7)	Internal peripheral bus 1, 3, 4, 5, 7, 8	DMA bus > system bus (CPU)
BUSSCNTFBU	Internal peripheral bus 9	ETHER bus > DMA bus > system bus (CPU)
BUSSCNTEXT	External bus (CS and SDRAM areas)	GPX bus > ETHER bus > DMA bus > system bus (CPU)
BUSSCNTEXT2	External bus (QSPI area)	GPX bus > ETHER bus > DMA bus > system bus (CPU)
BUSSCNTGPX	GPX bus	GLCDC (Graphic 1) > DRW (texture) > DRW (data) > JPEG (input) > GLCDC (Graphic 2) > JPEG (output)

Table 15.10 Bus priorities with round-robin priority arbitration (ARBMET[1:0] = 01b)

Slave Bus Control Register	Slave interface	Priority order*1
BUSSCNTFLI	Memory bus 1	Memory bus 3 <=> DCode bus (CPU) <=> ICode bus (CPU)
BUSSCNTRAMH	Memory bus 2	Memory bus 3 <=> DCode bus (CPU) <=> ICode bus (CPU)
BUSSCNTMBIU	Memory bus 3	GPX bus <=> ETHER bus <=> DMA bus
BUSSCNTRAM0	Memory bus 4	GPX bus <=> ETHER bus <=> DMA bus <=> system bus (CPU)
BUSSCNTRAM1	Memory bus 5	GPX bus <=> ETHER bus <=> DMA bus <=> system bus (CPU)
BUSSCNTPnB (n = 0, 2, 3, 4, 6, 7)	Internal peripheral bus 1, 3, 4, 5, 7, 8	DMA bus <=> system bus (CPU)
BUSSCNTFBU	Internal peripheral bus 9	ETHER bus <=> DMA bus <=> system bus (CPU)
BUSSCNTEXT	External bus (CS and SDRAM areas)	GPX bus <=> ETHER bus <=> DMA bus <=> system bus (CPU)
BUSSCNTEXT2	External bus (QSPI area)	GPX bus <=> ETHER bus <=> DMA bus <=> system bus (CPU)
BUSSCNTGPX	GPX bus	GLCDC (Graphic 1) > DRW (texture) > DRW (data) > JPEG (input) <=> GLCDC (Graphic 2) > JPEG (output)

Note 1. Round-robin priority is denoted by <=>.

15.3.21 Bus Error Address Register (BUSnERRADD) (n = 1 to 11)

Address(es): BUS.BUS1ERRADD 4000 4800h, BUS.BUS2ERRADD 4000 4810h, BUS.BUS3ERRADD 4000 4820h, BUS.BUS4ERRADD 4000 4830h, BUS.BUS5ERRADD 4000 4840h, BUS.BUS6ERRADD 4000 4850h, BUS.BUS7ERRADD 4000 4860h, BUS.BUS8ERRADD 4000 4870h, BUS.BUS9ERRADD 4000 4880h, BUS.BUS10ERRADD 4000 4890h, BUS.BUS11ERRADD 4000 48A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BERAD[31:16]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BERAD[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Bit name	Description	R/W
b31 to b0	BERAD[31:0]	Bus Error Address	When a bus error occurs, these bits store the error address.	R

Note: This register is cleared only by reset other than MPU related reset. For more information, see section 6, Resets, and section 16, Memory Protection Unit (MPU).

Table 15.8 lists the registers associated with each bus type.

BERAD[31:0] bits (Bus Error Address)

When a bus error occurs, the BERAD[31:0] bits store the access address. For more information, see the BUSnERRSTAT.ERRSTAT bit description and section 15.7, Bus Error Monitoring Section.

A value of BUSnERRADD.BERAD[31:0] (n = 1 to 11) is only valid when BUSnERRSTAT.ERRSTAT (n = 1 to 11) is set to 1.

15.3.22 Bus Error Status Register (BUSnERRSTAT) (n = 1 to 11)

Address(es): BUS.BUS1ERRSTAT 4000 4804h, BUS.BUS2ERRSTAT 4000 4814h, BUS.BUS3ERRSTAT 4000 4824h, BUS.BUS4ERRSTAT 4000 4834h, BUS.BUS5ERRSTAT 4000 4844h, BUS.BUS6ERRSTAT 4000 4854h, BUS.BUS7ERRSTAT 4000 4864h, BUS.BUS8ERRSTAT 4000 4874h, BUS.BUS9ERRSTAT 4000 4884h, BUS.BUS10ERRSTAT 4000 4894h, BUS.BUS11ERRSTAT 4000 48A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	ERRSTAT	—	—	—	—	—	—	ACCSTAT
Value after reset:	0	0	0	0	0	0	0	x

Bit	Symbol	Bit name	Description	R/W
b0	ACCSTAT	Error Access Status	Access status when the error occurred: 1: Write access 0: Read access.	R
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ERRSTAT	Bus Error Status	0: No bus error occurred 1: Bus error occurred.	R

Note: This register is cleared only by reset other than MPU related reset. For more information, see section 6, Resets, and section 16, Memory Protection Unit (MPU).

Table 15.8 lists the registers associated with each bus type.

ACCSTAT bit (Error Access Status)

The ACCSTAT bit indicates the access status, write access or read access, when an error occurs on the associated bus. For more information, see the BUSnERRSTAT.ERRSTAT bit description and section 15.7, Bus Error Monitoring Section.

The value is only valid when BUSnERRSTAT.ERRSTAT (n = 1 to 11) is set to 1.

ERRSTAT bit (Bus Error Status)

The ERRSTAT bit indicates whether a bus error occurred. When an error occurs on the associated bus, the access address and status of write or read access are stored. BUSnERRSTATn.ERRSTAT (n = 1 to 11) is set to 1.

The following types of errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- Bus slave MPU error
- Time out.

When detecting bus master MPU errors or bus slave MPU errors, and reset is selected in the respective OAD bit, BUSnERRSTAT.ERRSTAT (n = 1 to 11) is not set to 1 if the bus access that caused the MPU error completes later than the internal reset signal being generated, which may occur depending on the wait setting.

When detecting bus master MPU errors or bus slave MPU errors, and NMI is selected in the respective OAD bit, BUSnERRSTAT.ERRSTAT (n = 1 to 11) is set to 1 when the bus access that caused the MPU error completes.

For more information on errors that occur on each bus, see section 15.7, Bus Error Monitoring Section, and section 16, Memory Protection Unit (MPU). For the GPX bus, BUSnERRSTAT.ERRSTAT (n = 6 to 11) is normally set to 1 unless a transfer request by different master bus is made after access with a bus master MPU error.

15.4 Endianness and Data Alignment

The external bus has a data alignment function to control which byte of the data bus (D15 to D08, D07 to D00) is used when accessing the external address space (the CS and SDRAM areas). Alignment is based on the bus specifications of the area to be accessed (8-bit or 16-bit bus space), the data size, and the endian order.

15.4.1 Data Alignment Control for the CS Areas

(1) 16-bit bus space

When a 16-bit bus space is selected in the BSIZE[1:0] bits in CSnCR, address buses A23 to A01 are enabled to output address signals in 16-bit units, and the address bus A00 is disabled (always outputs low).

When byte strobe mode is selected (WRMOD = 0 in CSnMOD), the WR0 and WR1 pins are enabled. The BC0 and BC1 pins are not used.

When single-write strobe mode is selected (WRMOD = 1 in CSnMOD), only the WR0 pin is enabled, and it always outputs low during write access, regardless of the data size. The WR1 pin is invalid (always outputs high). The valid byte position is indicated by the BC0 and BC1 pins.

The valid positions of control signals and data external to the chip differ depending on the endian order. See Figure 15.3 and Figure 15.4.

Page access can occur for accesses to data in 32-bit units. Page access can only occur when an access does not extend over a 32-bit boundary and causes no change in the BC0 and BC1 signals. The situations in which page access occurs are indicated by the letter (p) in Figure 15.3 and Figure 15.4.

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		WR1/BC1 RD Data bus D15 D08	WR0/BC0 RD Data bus D07 D00
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n		7 0	
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+2		7 0	
16 bits	4n	One	First	16 bits	4n		15 8	7 0
	4n+2	One	First	16 bits	4n+2		15 8	7 0
32 bits	4n	Two	First	16 bits	4n		15 8	7 0
			Second	16 bits	4n+2	(p)	31 24	23 16

(p): Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD).

Figure 15.3 Data alignment in 16-bit bus space with little-endian order for CS areas

						WR1/BC1	WR0/BC0
						RD	RD
Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	Data bus D15 D08	D07 D00
8 bits	4n	One	First	8 bits	4n	7 0	
	4n+1	One	First	8 bits	4n		7 0
	4n+2	One	First	8 bits	4n+2	7 0	
	4n+3	One	First	8 bits	4n+2		7 0
16 bits	4n	One	First	16 bits	4n	15 8	7 0
	4n+2	One	First	16 bits	4n+2	15 8	7 0
32 bits	4n	Two	First	16 bits	4n	31 24	23 16
			Second	16 bits	4n+2 (p)	15 8	7 0

(p): Page access (only when page access is enabled in the PRENB and PWENB bits in CSnMOD).

Figure 15.4 Data alignment in 16-bit bus space with big-endian order for CS areas

(2) 8-bit bus space

When an 8-bit bus space is selected in the BSIZE[1:0] bits in CSnCR, the address buses A23 to A00 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0 pin is valid, regardless of the write access mode, and it always outputs low during write access. The WR1 and BC0 pins are not used.

The valid positions of data external to the chip are D07 to D00, and WR0 is used as the control signal, regardless of the endian mode. See Figure 15.5 and Figure 15.6.

Page access can occur for accesses to data in 16-bit or 32-bit units. Page access can only occur when an access does not extend over a 32-bit boundary. The situations in which page access occurs are indicated by the letter (p) in Figure 15.5 and Figure 15.6.

						WR1/BC1	WR0/BC0
						RD	
						Data bus	
Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	D15 D08	D07 D00
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n+1		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+3		7 0
16 bits	4n	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+1 (p)		15 8
	4n+2	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3 (p)		15 8
32 bits	4n	Four	First	8 bits	4n		7 0
			Second	8 bits	4n+1 (p)		15 8
			Third	8 bits	4n+2 (p)		23 16
			Fourth	8 bits	4n+3 (p)		31 24

(p): Page access (only when page access is enabled
in the PRENB and PWENB bits in CSnMOD)

Figure 15.5 Data alignment in 8-bit bus space with little-endian order for CS areas

						WR1/BC1	WR0/BC0
						RD	
						Data bus	
Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address	D15 D08	D07 D00
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n+1		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+3		7 0
16 bits	4n	Two	First	8 bits	4n		15 8
			Second	8 bits	4n+1 (p)		7 0
	4n+2	Two	First	8 bits	4n+2		15 8
			Second	8 bits	4n+3 (p)		7 0
32 bits	4n	Four	First	8 bits	4n		31 24
			Second	8 bits	4n+1 (p)		23 16
			Third	8 bits	4n+2 (p)		15 8
			Fourth	8 bits	4n+3 (p)		7 0

(p): Page access (only when page access is enabled
in the PRENB and PWENB bits in CSnMOD)

Figure 15.6 Data alignment in 8-bit bus space with big-endian order for CS areas

15.4.2 Data Alignment Control for the SDRAM Area

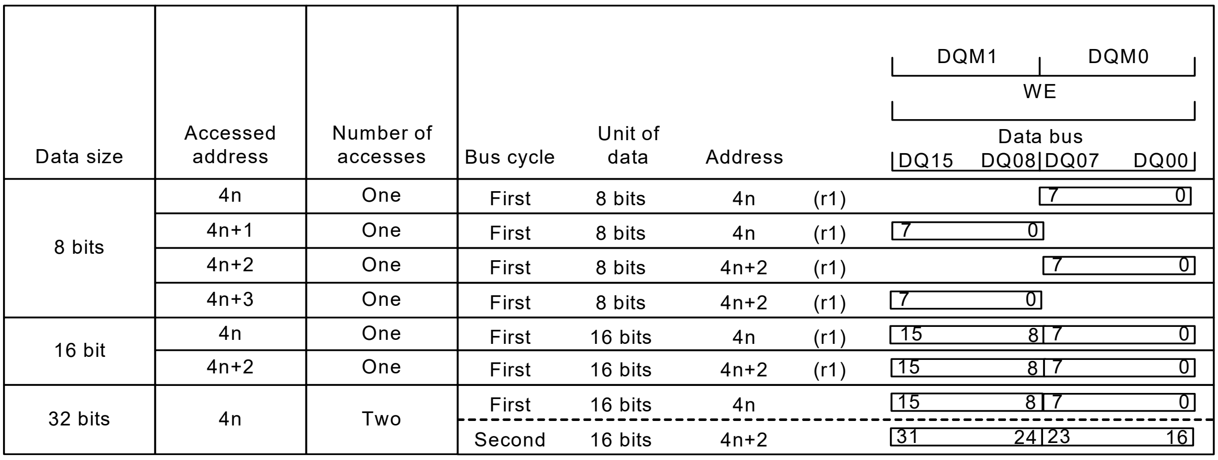
(1) 16-bit bus space

When a 16-bit bus space is selected in the BSIZE[1:0] bits in SDCCR, address buses A26 to A01 are enabled to output address signals in 16-bit units, and the address bus A00 is disabled (always outputs low). The valid byte position is indicated by the DQM0 and DQM1 signals.

External data is accessed using the DQ15 to DQ08 and DQ07 to DQ00 pins and DQM0 and DQM1 control signals. Data can be accessed in either 8-bit or 16-bit units at a time.

The valid positions of control signals and data external to the chip differ depending on the endian order. See Figure 15.7 and Figure 15.8.

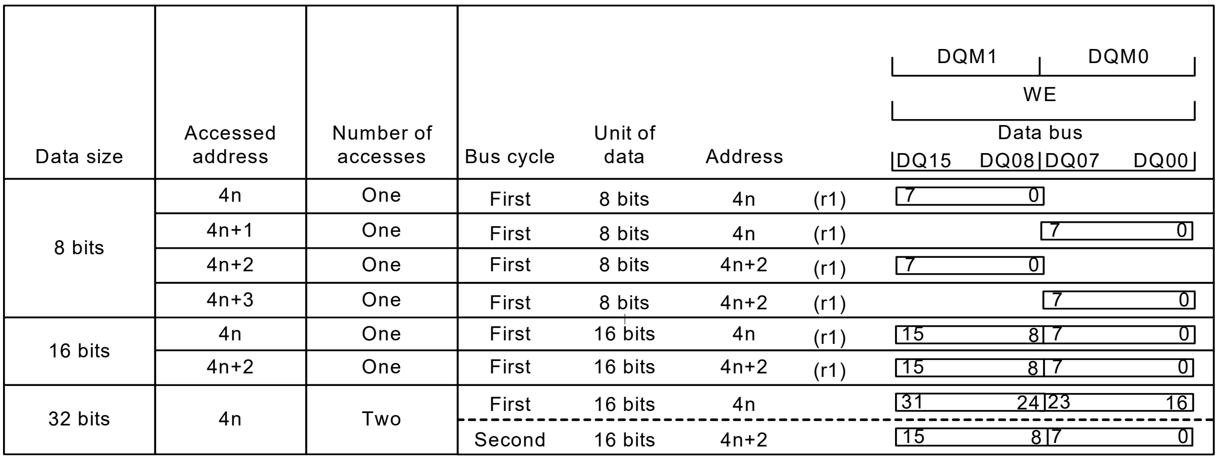
Consecutive access can occur for accesses to data in 8- or 16-bit units. Consecutive access can only occur when a single bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by "(r1)" in Figure 15.7 and Figure 15.8.



Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQ15 DQ08 (DQM1)	DQ07 DQ00 (DQM0)
8 bits	4n	One	First	8 bits	4n	(r1)		7 0
	4n+1	One	First	8 bits	4n	(r1)	7 0	
	4n+2	One	First	8 bits	4n+2	(r1)		7 0
	4n+3	One	First	8 bits	4n+2	(r1)	7 0	
16 bit	4n	One	First	16 bits	4n	(r1)	15 8	7 0
	4n+2	One	First	16 bits	4n+2	(r1)	15 8	7 0
32 bits	4n	Two	First	16 bits	4n		15 8	7 0
			Second	16 bits	4n+2		31 24	23 16

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the HMI burst transfer.)

Figure 15.7 Data alignment in 16-bit bus space with little-endian order for SDRAM area



Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQ15 DQ08 (DQM1)	DQ07 DQ00 (DQM0)
8 bits	4n	One	First	8 bits	4n	(r1)	7 0	
	4n+1	One	First	8 bits	4n	(r1)		7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0	
	4n+3	One	First	8 bits	4n+2	(r1)		7 0
16 bits	4n	One	First	16 bits	4n	(r1)	15 8	7 0
	4n+2	One	First	16 bits	4n+2	(r1)	15 8	7 0
32 bits	4n	Two	First	16 bits	4n		31 24	23 16
			Second	16 bits	4n+2		15 8	7 0

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the HMI burst transfer.)

Figure 15.8 Data alignment in 16-bit bus space with big-endian order for SDRAM area

(2) 8-bit bus space

When an 8-bit width is selected in the BSIZE[1:0] bits in SDCCR, address buses A26 to A00 are enabled to output address signals in 8-bit units.

External data is accessed using the DQ07 to DQ00 pins and DQM0 control signal. 8-bit data is accessed in single 8-bit accesses, 16-bit data with two 8-bit accesses, and 32-bit data with four 8-bit accesses.

The valid positions of control signals and data external to the chip differ depending on the endian order. See Figure 15.9 and Figure 15.10.

Consecutive access can occur in access to data in 8-bit units. Consecutive access can only occur when a single bus access is generated in response to a single transfer request. The situations in which consecutive access occurs are indicated by “(r1)” in Figure 15.9 and Figure 15.10.

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1 DQM0 WE Data bus DQ15 DQ08 DQ07 DQ00
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n+1	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+3	(r1)	7 0
16 bits	4n	Two	First	8 bits	4n		7 0
			Second	8 bits	4n+1		15 8
	4n+2	Two	First	8 bits	4n+2		7 0
			Second	8 bits	4n+3		15 8
32 bits	4n	Four	First	8 bits	4n		7 0
			Second	8 bits	4n+1		15 8
			Third	8 bits	4n+2		23 16
			Fourth	8 bits	4n+3		31 24

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the HMI burst transfer.)

Figure 15.9 Data alignment in 8-bit bus space with little-endian order for SDRAM area

Data size	Accessed address	Number of accesses	Bus cycle	Unit of data	Address		DQM1 DQM0 WE Data bus DQ15 DQ08 DQ07 DQ00
8 bits	4n	One	First	8 bits	4n	(r1)	7 0
	4n+1	One	First	8 bits	4n+1	(r1)	7 0
	4n+2	One	First	8 bits	4n+2	(r1)	7 0
	4n+3	One	First	8 bits	4n+3	(r1)	7 0
16 bits	4n	Two	First	8 bits	4n		15 8
			Second	8 bits	4n+1		7 0
	4n+2	Two	First	8 bits	4n+2		15 8
			Second	8 bits	4n+3		7 0
32 bits	4n	Four	First	8 bits	4n		31 24
			Second	8 bits	4n+1		23 16
			Third	8 bits	4n+2		15 8
			Fourth	8 bits	4n+3		7 0

(r1): Consecutive access (only when consecutive access is enabled by BE = 1 in SDAMOD during the HMI burst transfer.)

Figure 15.10 Data alignment in 8-bit bus space with big-endian order for SDRAM area

15.5 Operation of CS Area Controller

15.5.1 Separate Bus

This section describes the periods shown in the timing diagrams. The CS area controller (CSC) operates in synchronization with the external bus clock, BCLK. Operation cycles, such as wait cycles, specified in the CSC register, are counted on BCLK. In the following description, the frequencies of BCLK and EBCLK pin output are the same, unless otherwise noted. Access through the external bus starts at the same point as the output of a rising edge on the EBCLK pin. However, if the external bus clock, BCLK, and the output on the EBCLK pin are at different frequencies, the wait settings can cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin. See Figure 15.16 to Figure 15.20. If recovery cycles are inserted for bus access, the setting for the number of recovery cycles can also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin. See Figure 15.38.

(a) Tw1 to Twn (clock cycles for waiting for a normal read cycle or normal write cycle)

The period from Tw1 to Twn is the number of clock cycles from the start of access through the external bus clock to 1 cycle before the strobe signal is valid. The number of cycles is selectable from 0 to 31. Within this period, the timing of CSn, RD, and WRn assertion (driving the signals low) is determined by the respective wait settings. The wait periods are controlled by the CS Assert Wait Select bits (CSON), the RD Assert Wait Select bits (RDON), the WR Assert Wait Select bits (WRON), and the Write-Data Output Wait Select bits (WDON) in CSn Wait Control Register 2 (CSnWCR2). The number of clock cycles for each of these wait periods is selectable as a value from 0 to 7, counted from the start of external bus access. The selectable number of cycles is also within the overall number of clock cycles required for waiting to read or write.

(b) Tend (clock cycle where the strobe signal is valid)

Tend is the next clock cycle after completion of the wait period for a normal cycle of read or write, or for a cycle of page reading or page writing. If the wait select bit for these cycles is 0, bus access starts on the clock cycle where the strobe signal is valid. The RD and WRn signals are negated in the next clock cycle. For a read access, the clock cycle where the strobe signal is valid is where the data to be read is sampled. If an external wait is enabled, the wait signal is sampled on the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is low. The bus cycle completes in the next clock cycle if the wait signal is high. Tend indicates the cycle where sampling of the wait signal starts.

After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (see section (e), Tpw1 to Tpwn (page read cycle wait or page write cycle wait)) start in the next cycle, except during write access with a setting other than 0 for write-data output extension clock cycles (see section (d), Tdw1 to Tdwn (clock cycles for write-data output extension)). If the setting for the RD or WR assertion wait is any value other than 0, the RD and WRn signals are negated in the next clock cycle. If the setting is 0, assertion continues. Additionally, the CSn signal continues to be asserted rather than negated.

(c) Tn1 to Tnm (clock cycles for CS extension)

For normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn signal. For read or write access, the negation timing can be controlled by the read-access CS Extension Cycle Select bits (CSROFF) and the write-access CS Extension Cycle Select bits (CSWOFF) in the CSn Wait Control Register 2 (CSnWCR2). The number of cycles is counted from the cycle following the cycle where the strobe signal is valid.

For page access, Tn1 to Tnm represent the clock cycles of the period following the last cycle where the strobe signal is valid up to negation of the CSn signal.

For write access, setting the Write Data Output Extension Cycle Select bits (WDOFF) controls extension of the period where the address and output data is valid.

(d) Tdw1 to Tdwn (clock cycles for write-data output extension)

For write access, if the wait setting for the write-data output extension is any value other than 0, the specified clock cycles are inserted from the cycle following the cycle where the strobe signal is valid (Tend).

For normal access, this period is inserted within the clock cycle period for CS extension (Tn1 to Tnm).

For page access, this period is inserted within the clock cycle period where the strobe signal is valid and subsequent page accesses, or within the clock cycle period for the CS extension (Tn1 to Tnm). Valid address and data output are extended over this period, and the WRn signal is negated.

(e) Tpw1 to Tpwn (page read cycle wait or page write cycle wait)

For the second and subsequent bus cycles during page access, the values for a page read cycle wait or page write cycle wait are used instead of the settings for a normal read or write cycle wait. The settings in the WR Assert Wait Select bits become enabled in the same way as for the first access. The RD assertion control operation depends on the page read access mode setting (the PRMOD bit in CSnMOD) as follows:

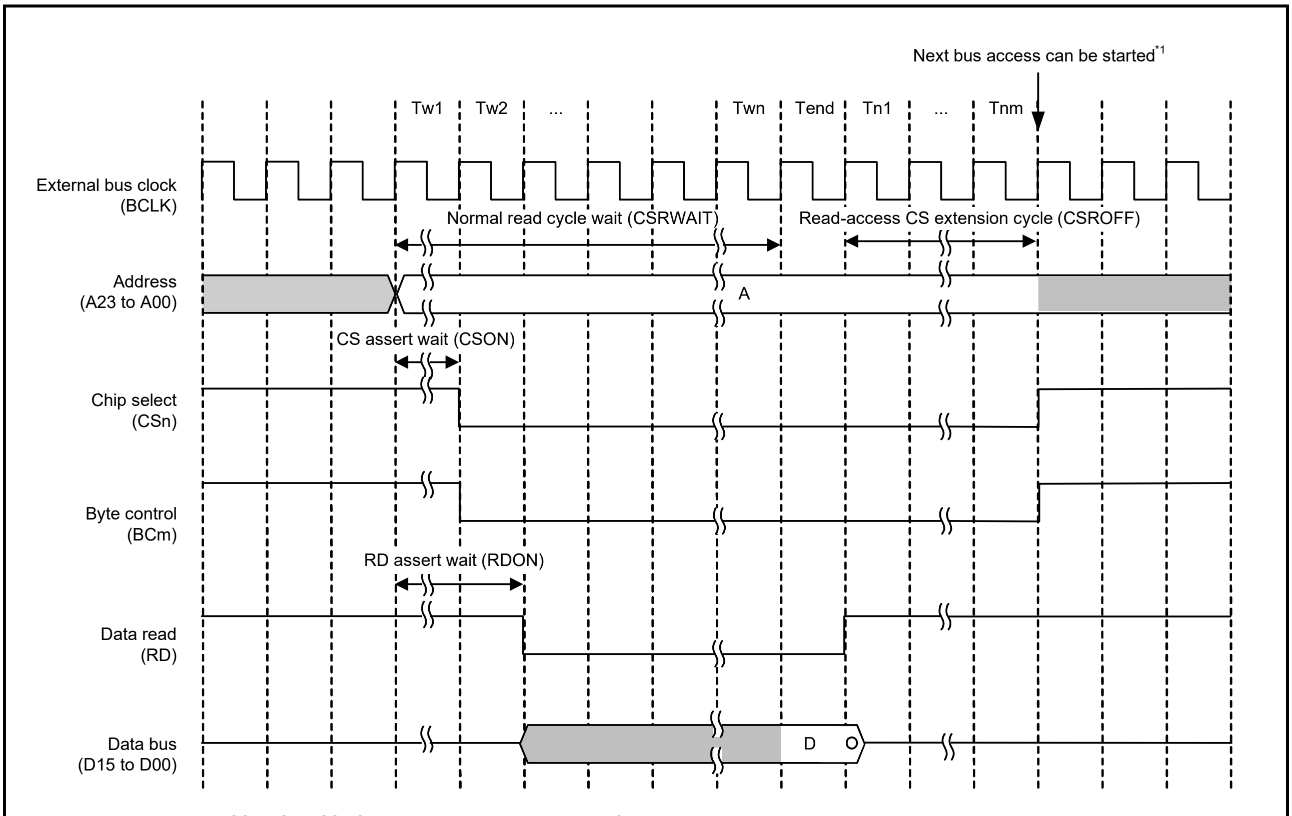
- CSnMOD.PRMOD = 0: A wait for RD assertion is inserted in the same way as for the first access, and the RD signal is negated
- CSnMOD.PRMOD = 1: Although a wait for RD assertion is inserted in the same way as for normal-access compatibility mode, the RD signal continues to be asserted over this period.

(f) Tr1 to Trn (recovery cycles)

Recovery cycles can be inserted from the point where a bus cycle is complete (CSn signal negation). The number of recovery cycles can be controlled by setting the Read Recovery (RRCV) or Write Recovery (WRCV) bits in the CSn Recovery Cycle Register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn negation) and can be selected from 0 to 15 cycles. For more information, see section 15.5.4, Insertion of Recovery Cycles.

(1) Normal access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page read and page write access, all bus accesses take the form of normal read and write operations. Even when these bits are set to 1 to enable page read and page write access, bus access other than page access takes the form of normal read and write operations. Figure 15.11 to Figure 15.13 show the normal access operations.



Note 1: When CSnWCR2.CSROFF[2:0] = 000b, the next round of bus access can start one cycle later.

O: Indicates sampling point

Figure 15.11 Bus timing for normal read operation (n = 0 to 7, m = 0, 1)

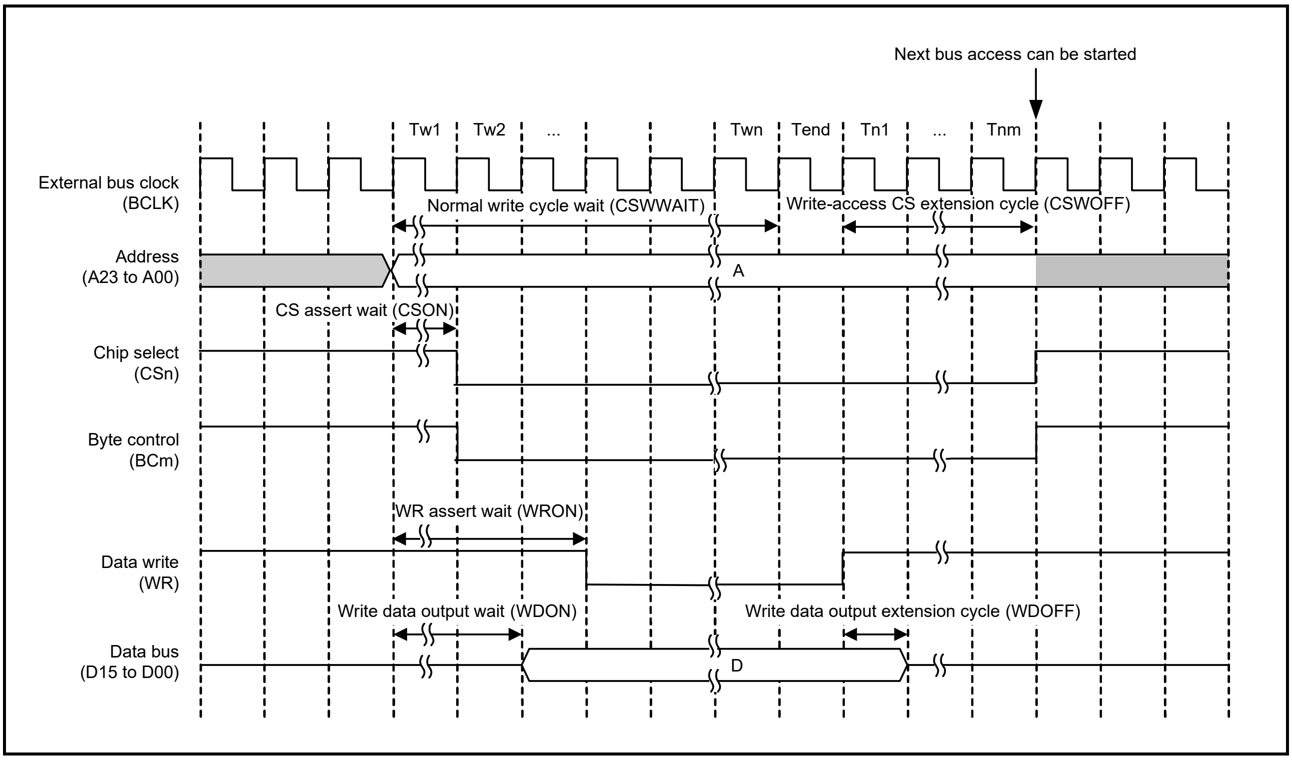


Figure 15.12 Bus timing for normal write operation in single-write strobe mode (n = 0 to 7, m = 0, 1)

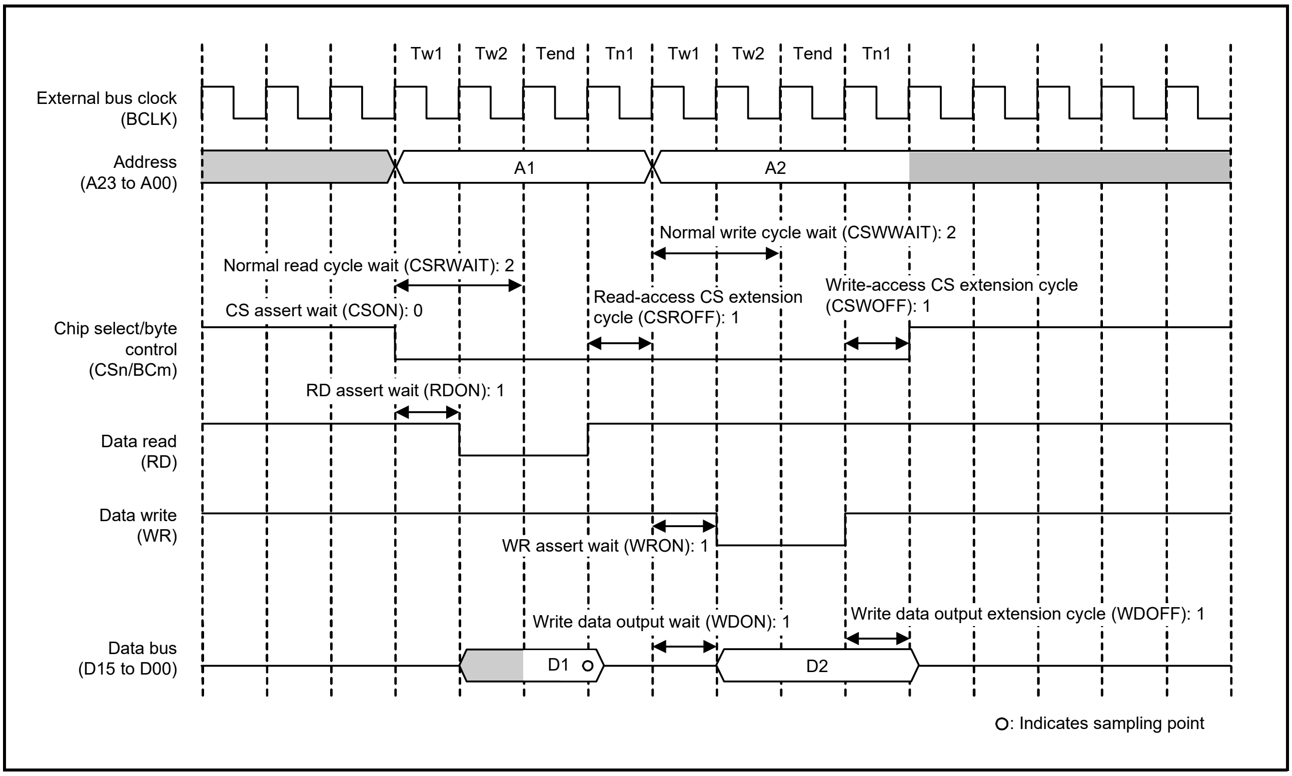


Figure 15.13 Example of normal access operation for read and write (n = 0 to 7, m = 0, 1)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations are repeated. See section (a), Tw1 to Twn (clock cycles for waiting for a normal read cycle or normal write cycle) to section (d), Tdw1 to Tdwn (clock cycles for write-data output extension). Figure 15.14 and Figure 15.15 show examples of operations when two rounds of bus access are generated in response to a single transfer request. If the recovery cycle insertion condition is satisfied, recovery cycles (section (f), Tr1 to Trn (recovery cycles)) are also inserted in the second and subsequent external bus accesses. See Figure 15.36.

The values in the wait control registers shown in the figures are example settings. In your application, set the registers appropriately for the specifications of connected devices.

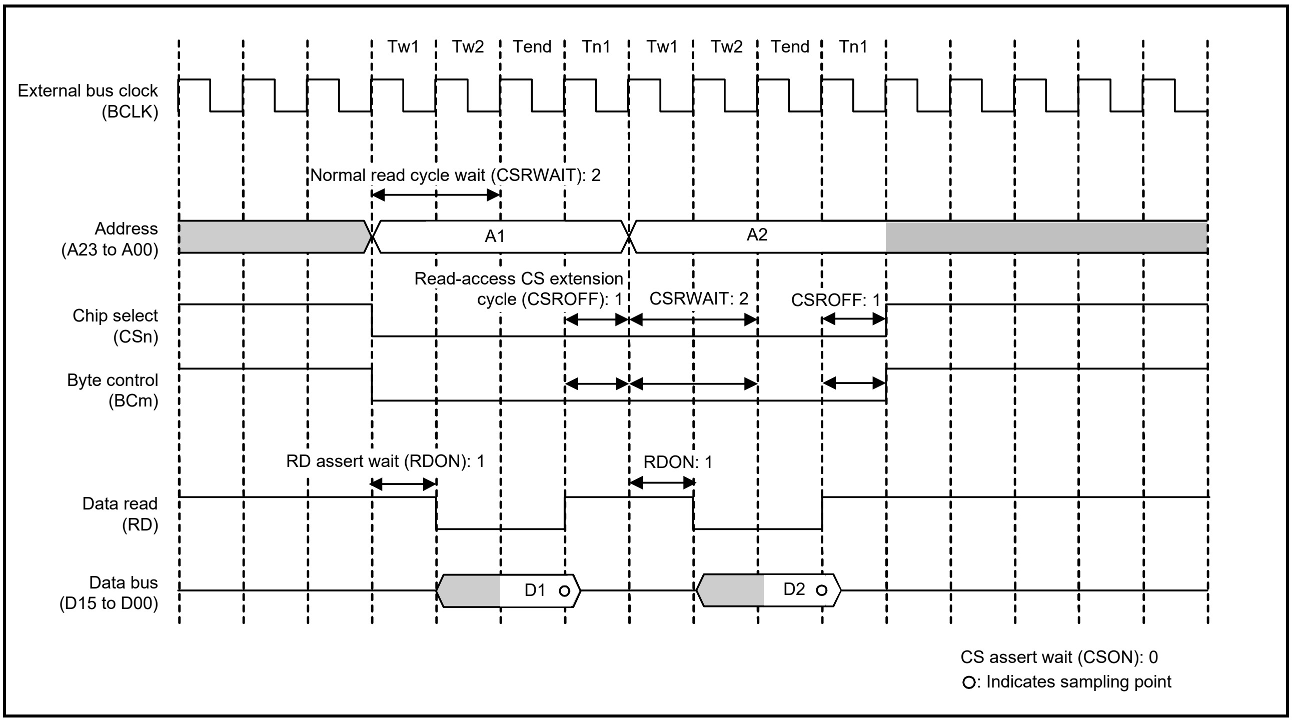


Figure 15.14 Example of normal read operation when two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0, 1)

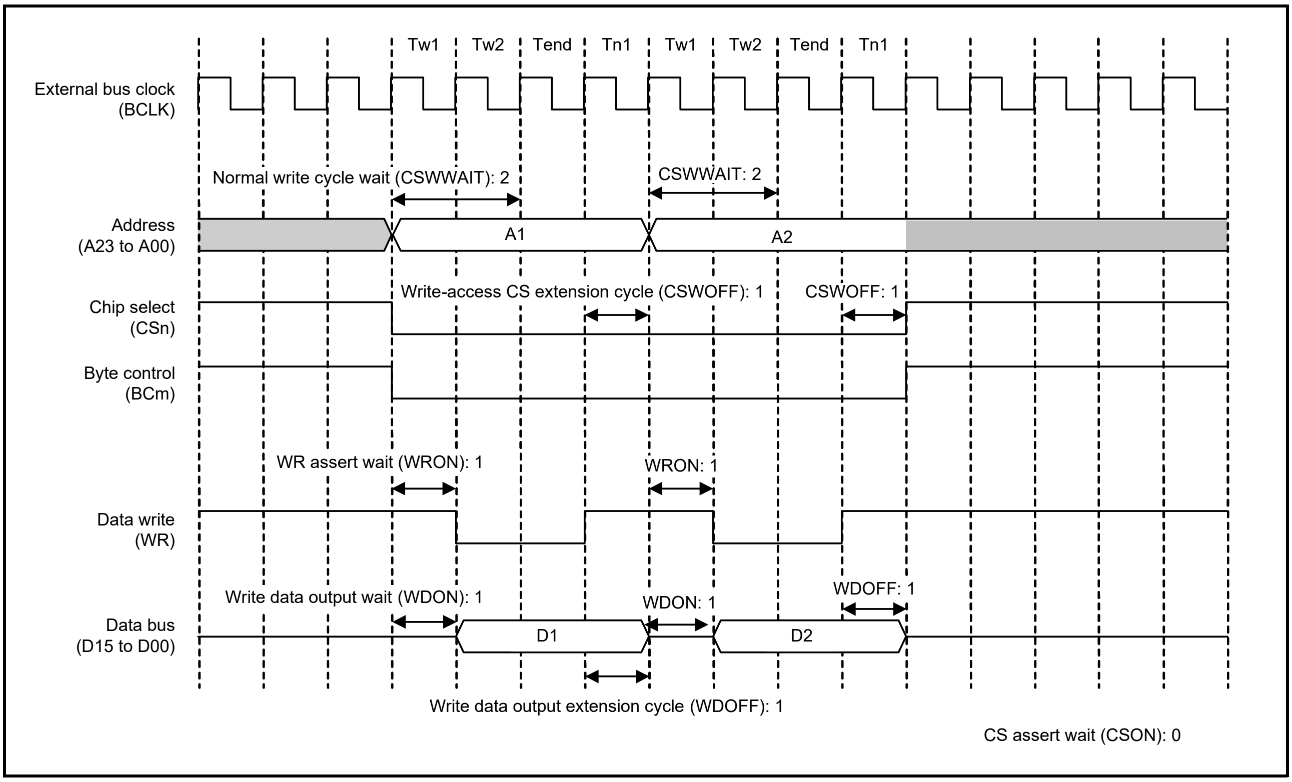


Figure 15.15 Example of normal write operation when two rounds of bus access are generated in response to a single transfer request in single-write strobe mode (n = 0 to 7, m = 0, 1)

Figure 15.16 to Figure 15.20 show examples of normal accesses made when BCLK/2 is selected as the frequency division in the EBCLK Pin Output Select bit.

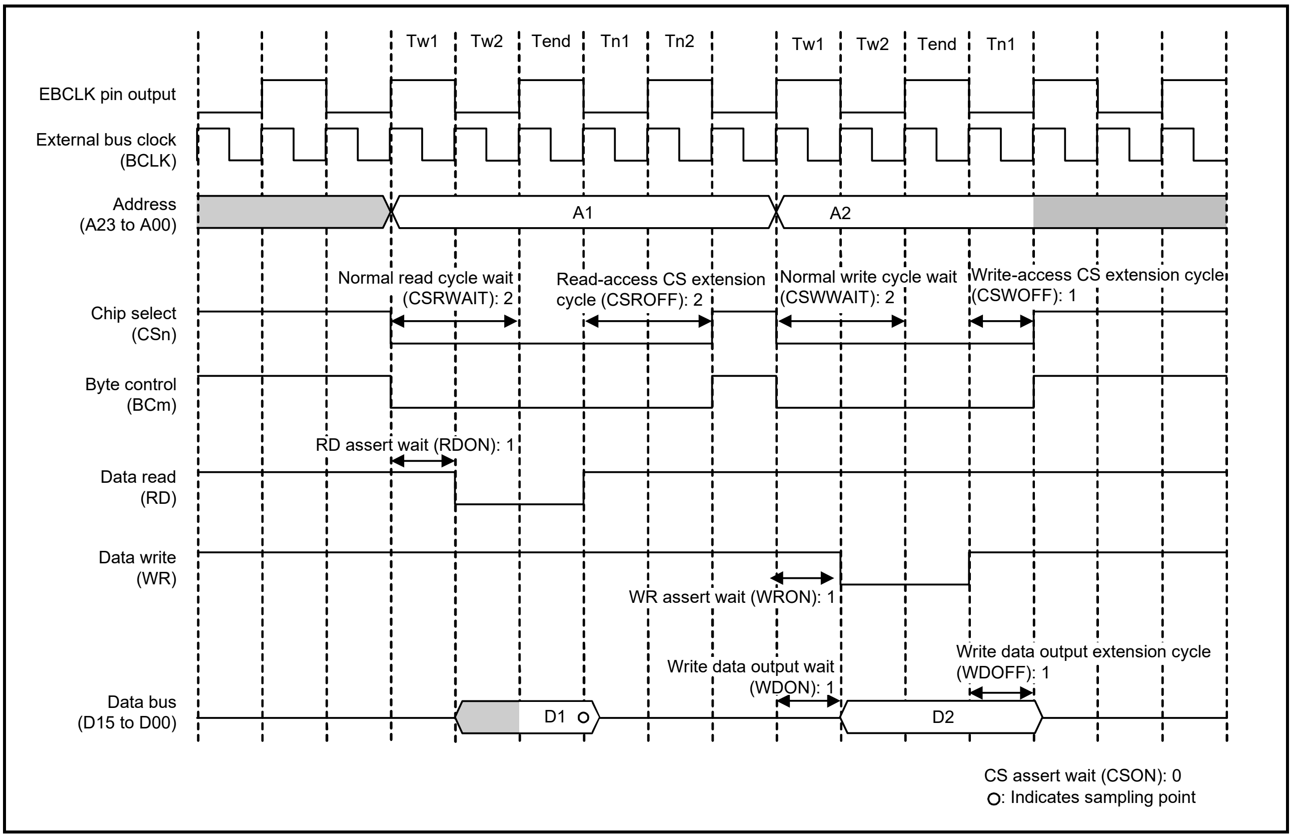


Figure 15.16 Example of normal access when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0, 1)

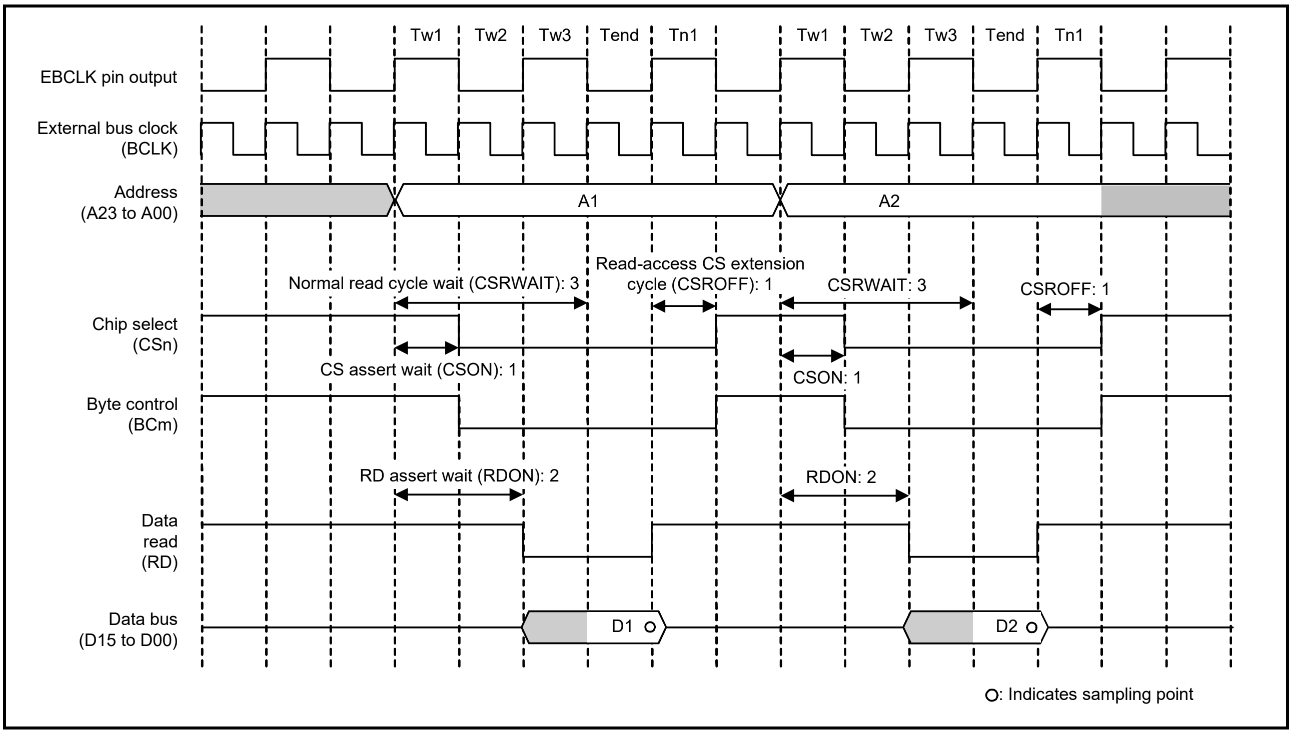


Figure 15.17 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0, 1)

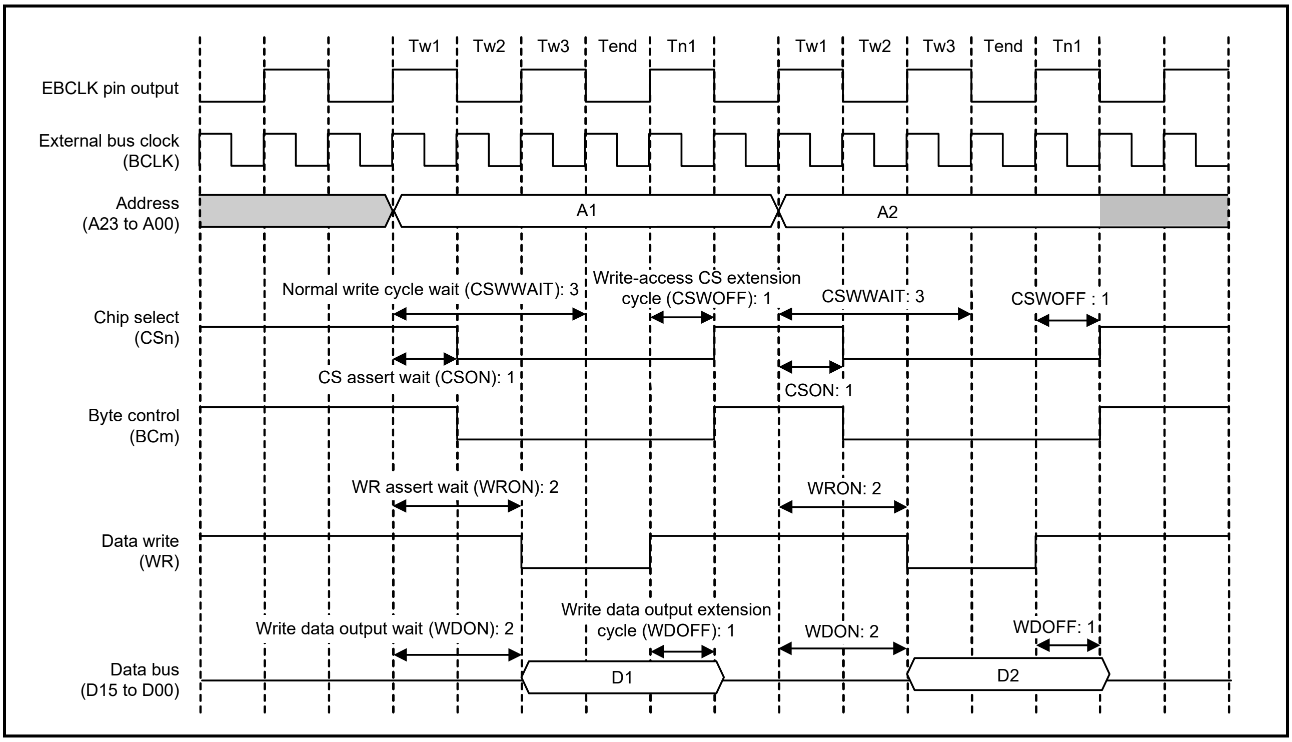


Figure 15.18 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 7, m = 0, 1)

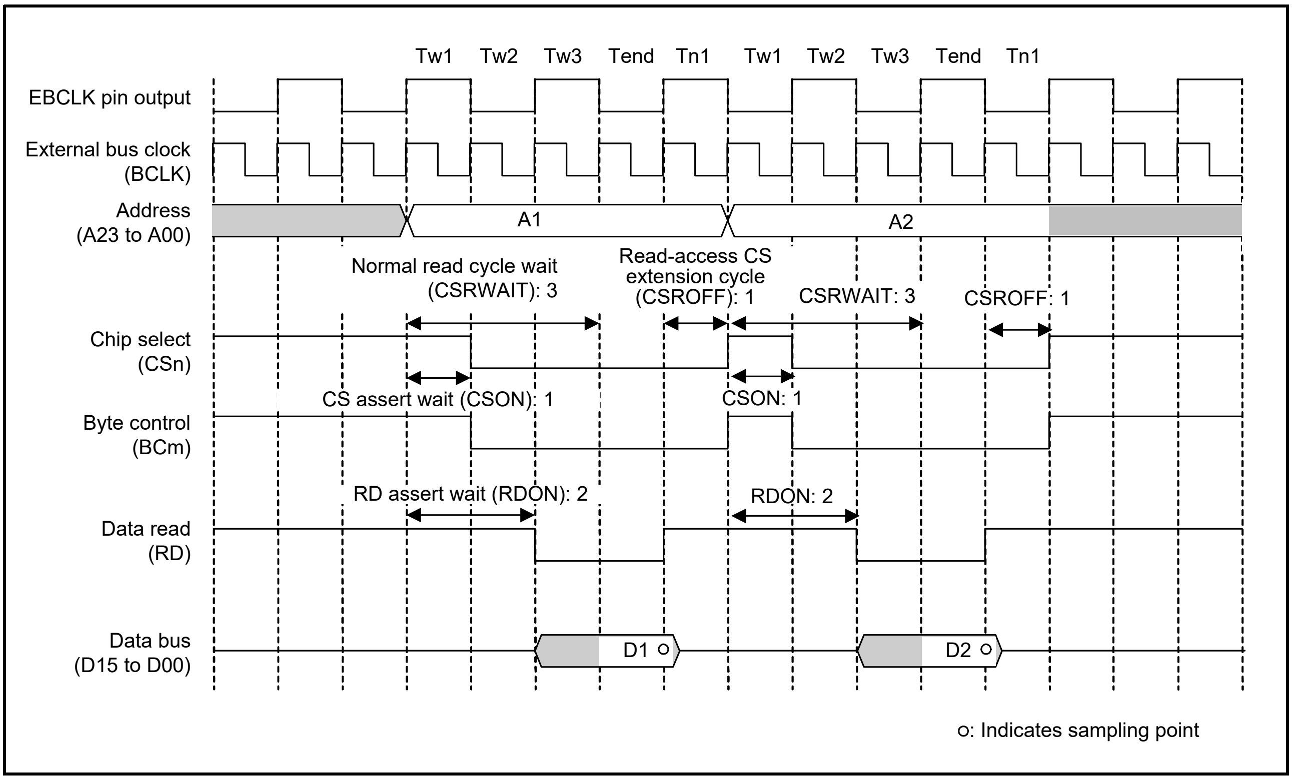


Figure 15.19 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0, 1)

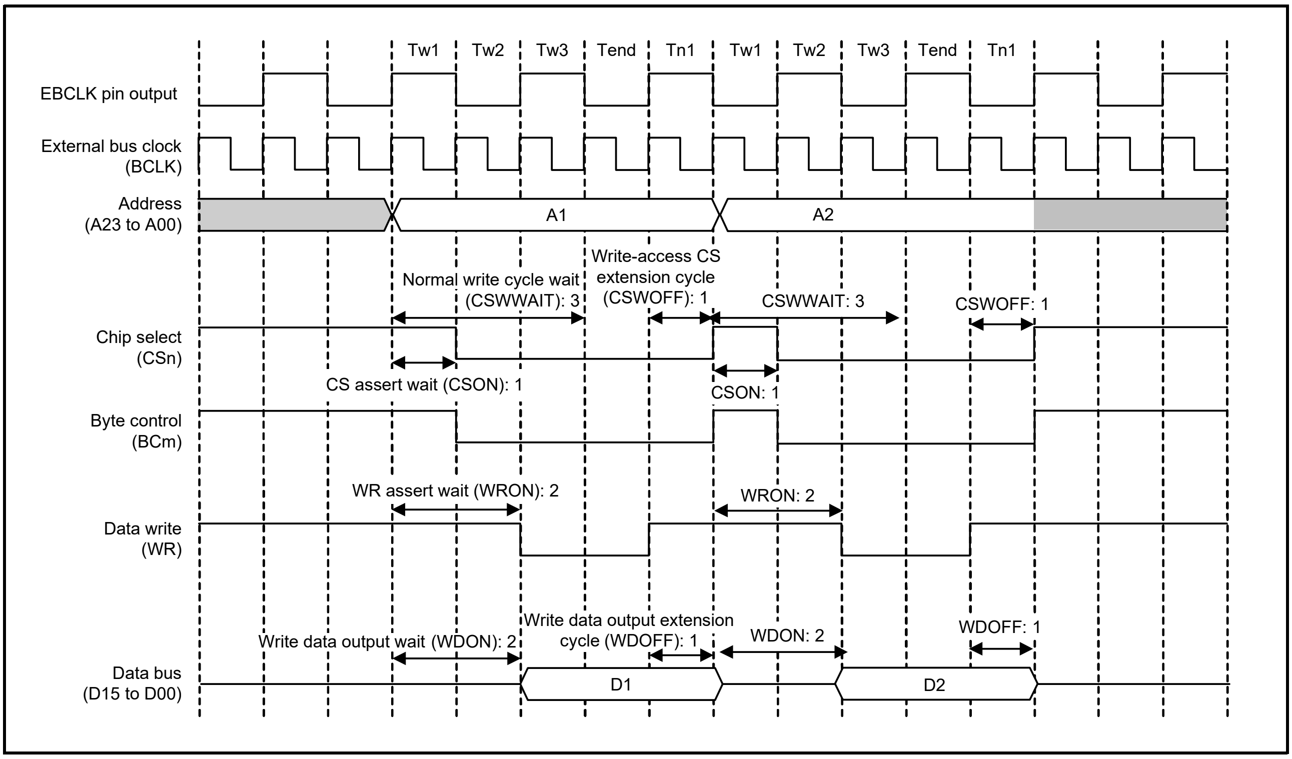
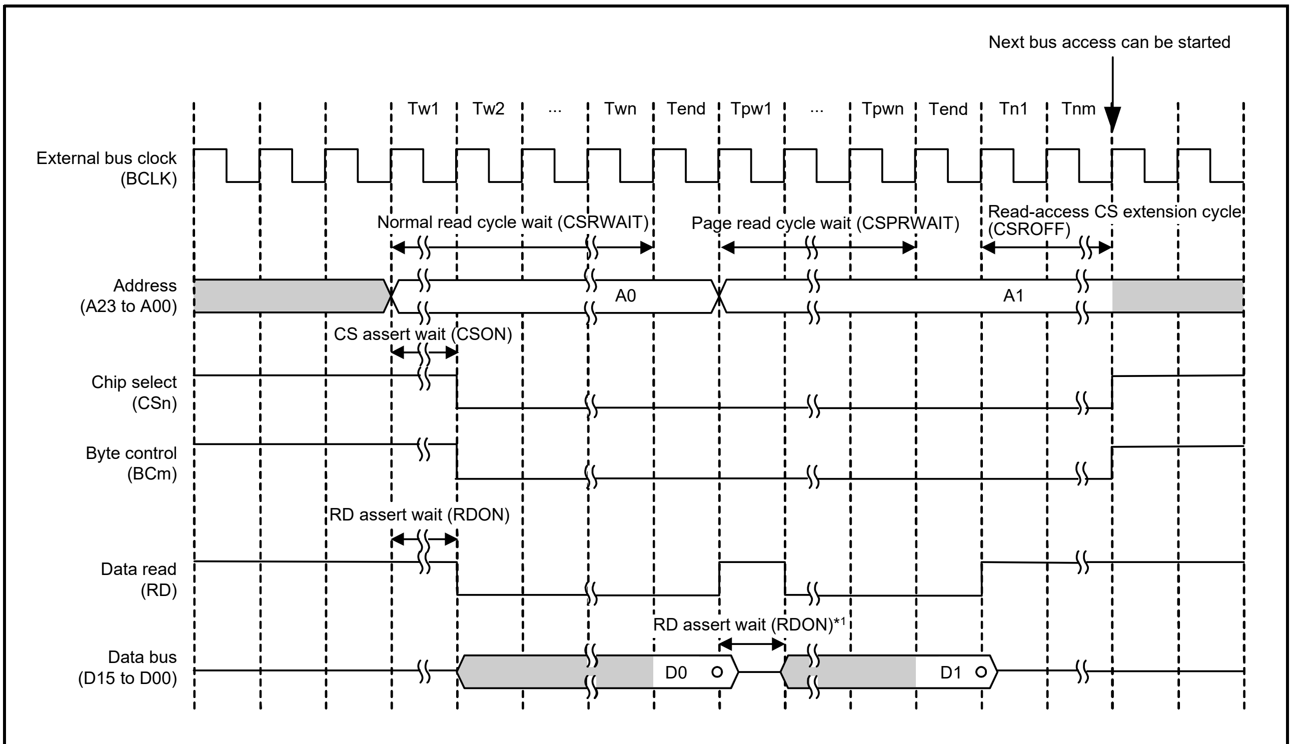


Figure 15.20 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7, m = 0, 1)

(2) Page access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page read and page write access, the bus access for page access operations becomes page reading and writing. Page access can only occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when split accesses are not aligned or access extends across the 32-bit boundary. See Figure 15.3 to Figure 15.6 for the conditions under which page access occurs.

Figure 15.21 and Figure 15.22 show examples of page access operations.



Note 1. The RD assert wait operation in the second and subsequent bus accesses depends on the page read access mode setting.

○: Indicates sampling point

Figure 15.21 Page read access timing (n = 0 to 7, m = 0, 1)

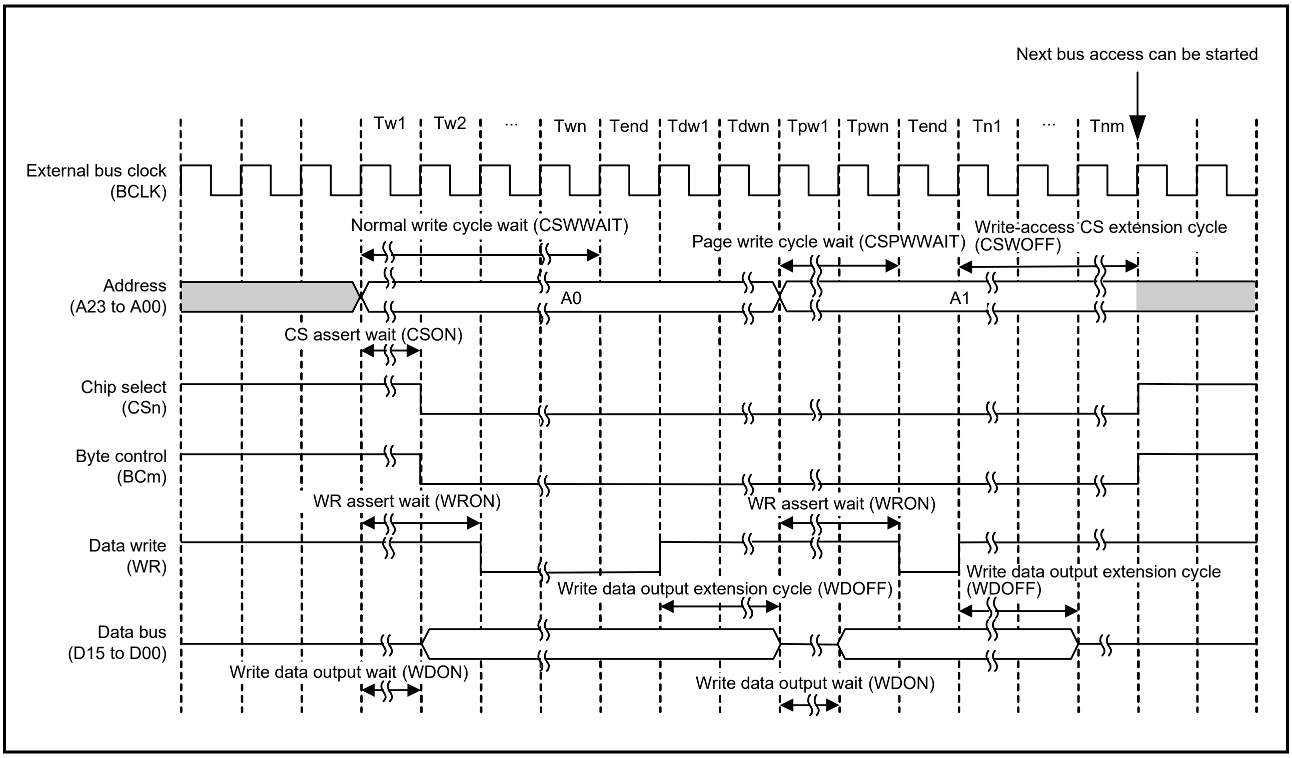


Figure 15.22 Page write access timing (n = 0 to 7, m = 0, 1)

Figure 15.23 and Figure 15.24 show examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers shown in the figures are example settings. In your application, set the registers appropriately for the specifications of connected devices.

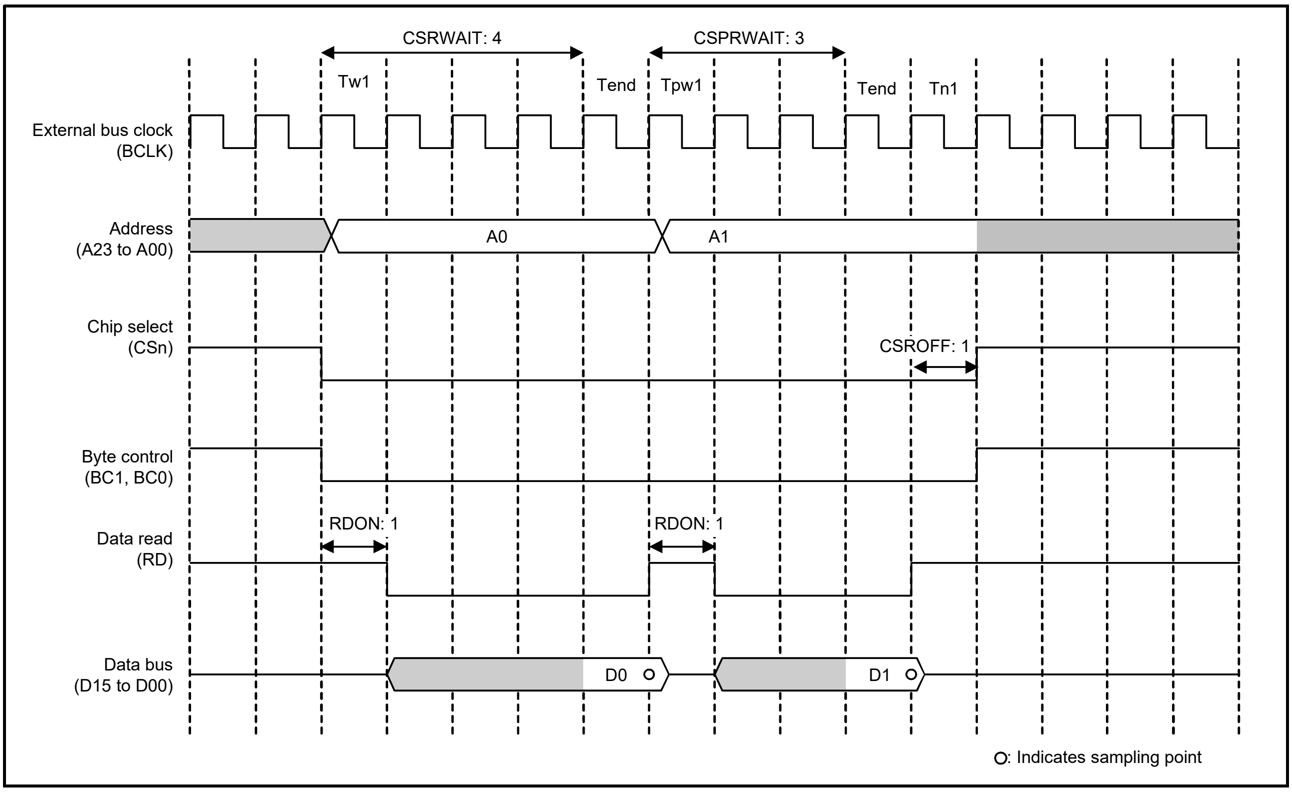


Figure 15.23 Example page read access operation when 16-bit bus space is accessed in 32 bits (n = 0 to 7)

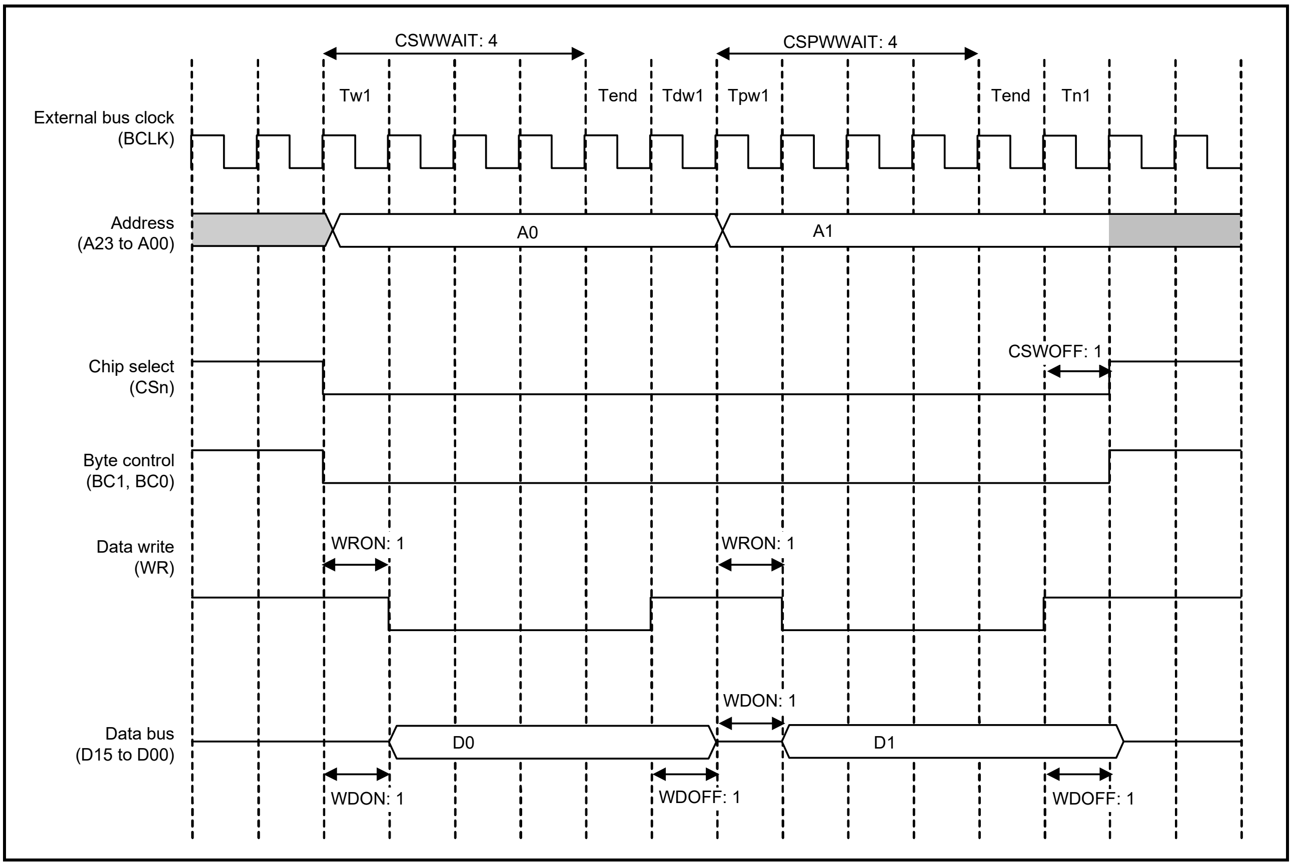


Figure 15.24 Example page write access operation when 16-bit bus space is accessed in 32 bits in single-write strobe mode (n = 0 to 7)

Figure 15.25 and Figure 15.26 show examples of page access operations when BCLK/2 is selected as the frequency division in the EBCLK Pin Output Select bit.

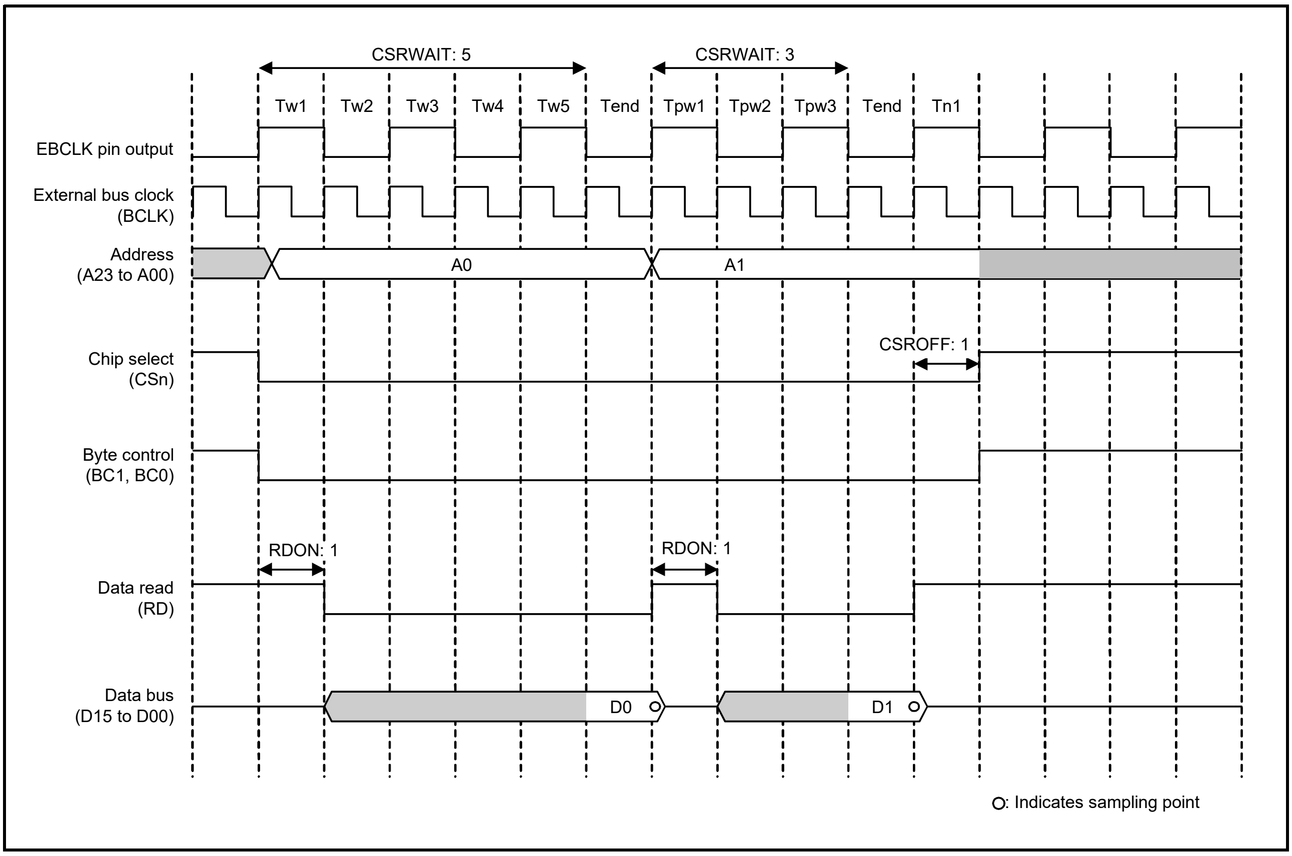


Figure 15.25 Example page read access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 7)

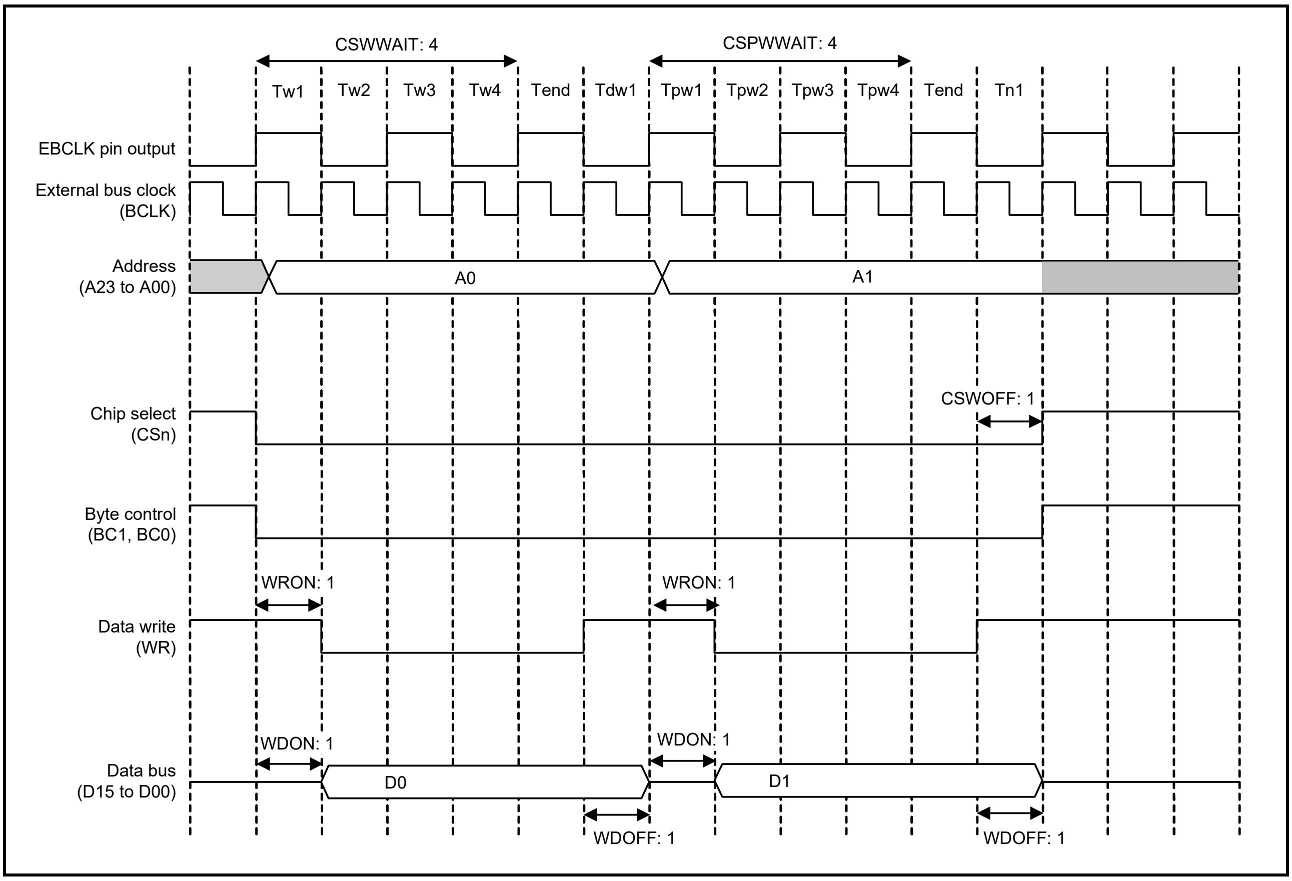


Figure 15.26 **Example page write access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request, in single-write strobe mode (n = 0 to 7)**

15.5.2 Address/Data Multiplexed Bus

When the address/data Multiplexed I/O Interface Select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexing input/output to/from the D15 to D00 pins in the corresponding area. Using this function enables direct connection of this LSI to peripheral LSIs requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D00 are multiplexed with A07 to A00. When 16-bit width is selected, D15 to D00 are multiplexed with A15 to A00. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD, WRn, and BCn signals.

Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn signals within the address cycle, the byte-control signal is output for the data being read or written.

During the address/data multiplexed I/O space access, after the number of wait cycles specified by the address cycle wait select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

Ta1 to Tan (Address Cycle Wait)
The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and 1 cycle before the address latch (ALE) signal is negated. The number of cycles are selectable within the range from zero to three. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.

Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 15.27 to Figure 15.29 show examples of operations with the address/data multiplexed I/O interface

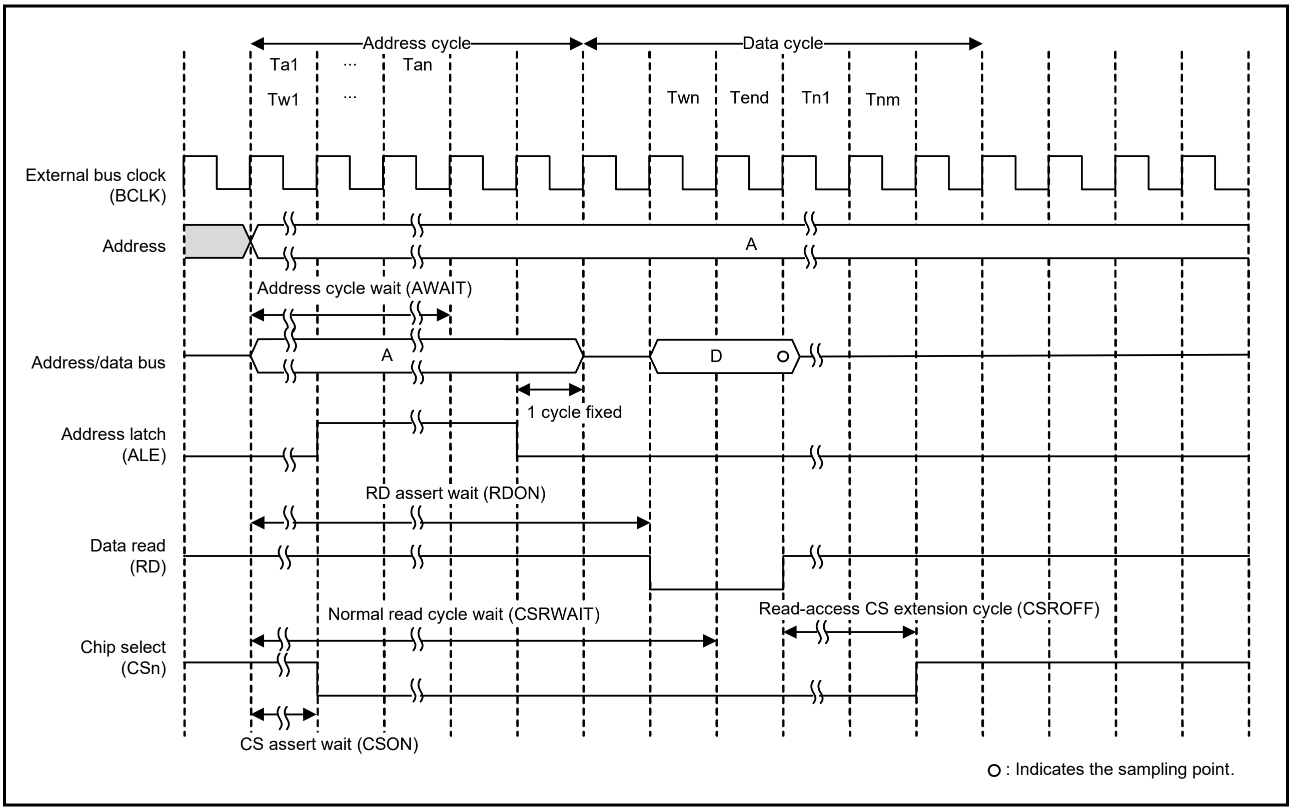


Figure 15.27 Example of read access operation with address/data multiplexed I/O interface (n = 0 to 7)

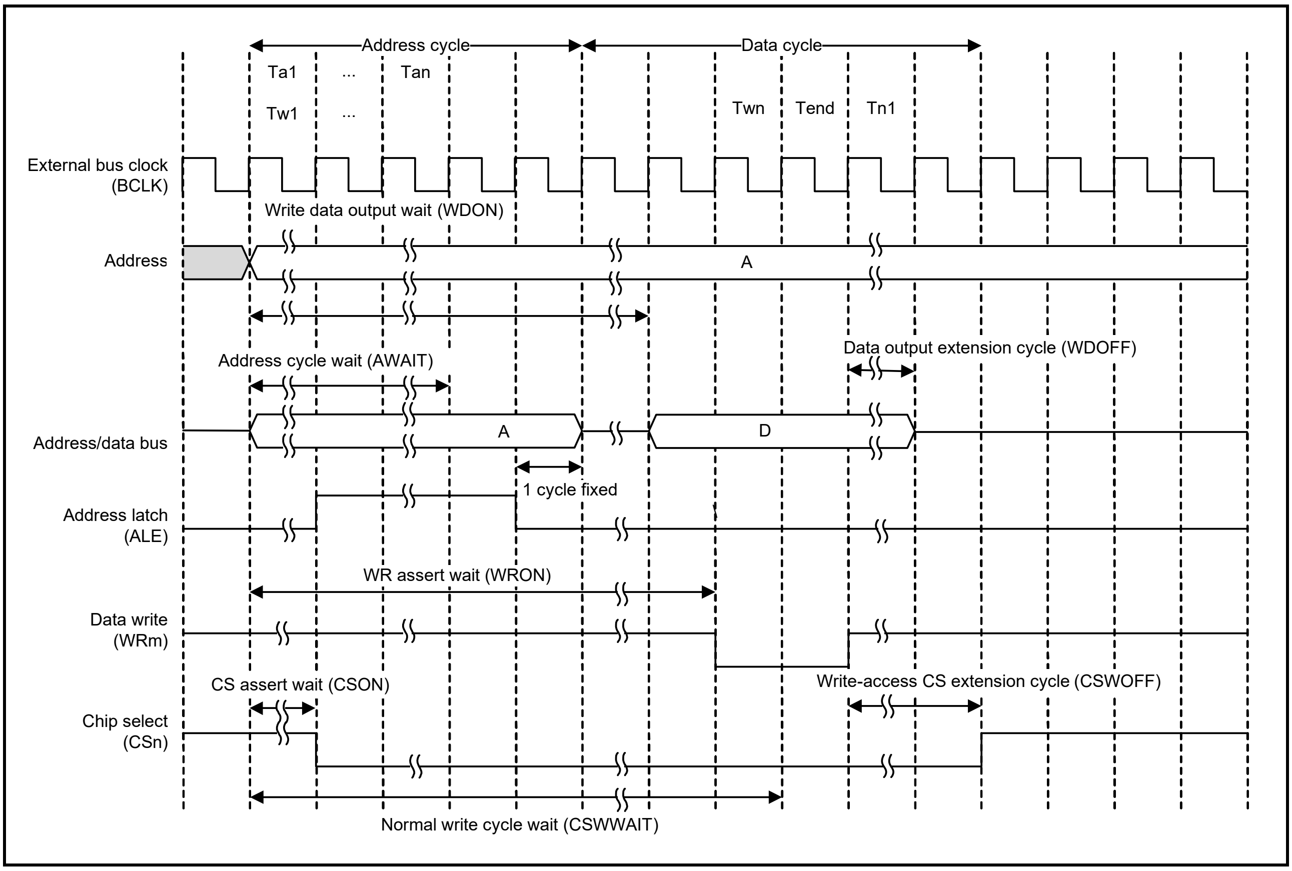


Figure 15.28 Example of write access operation with address/data multiplexed I/O interface (m = 0, 1)

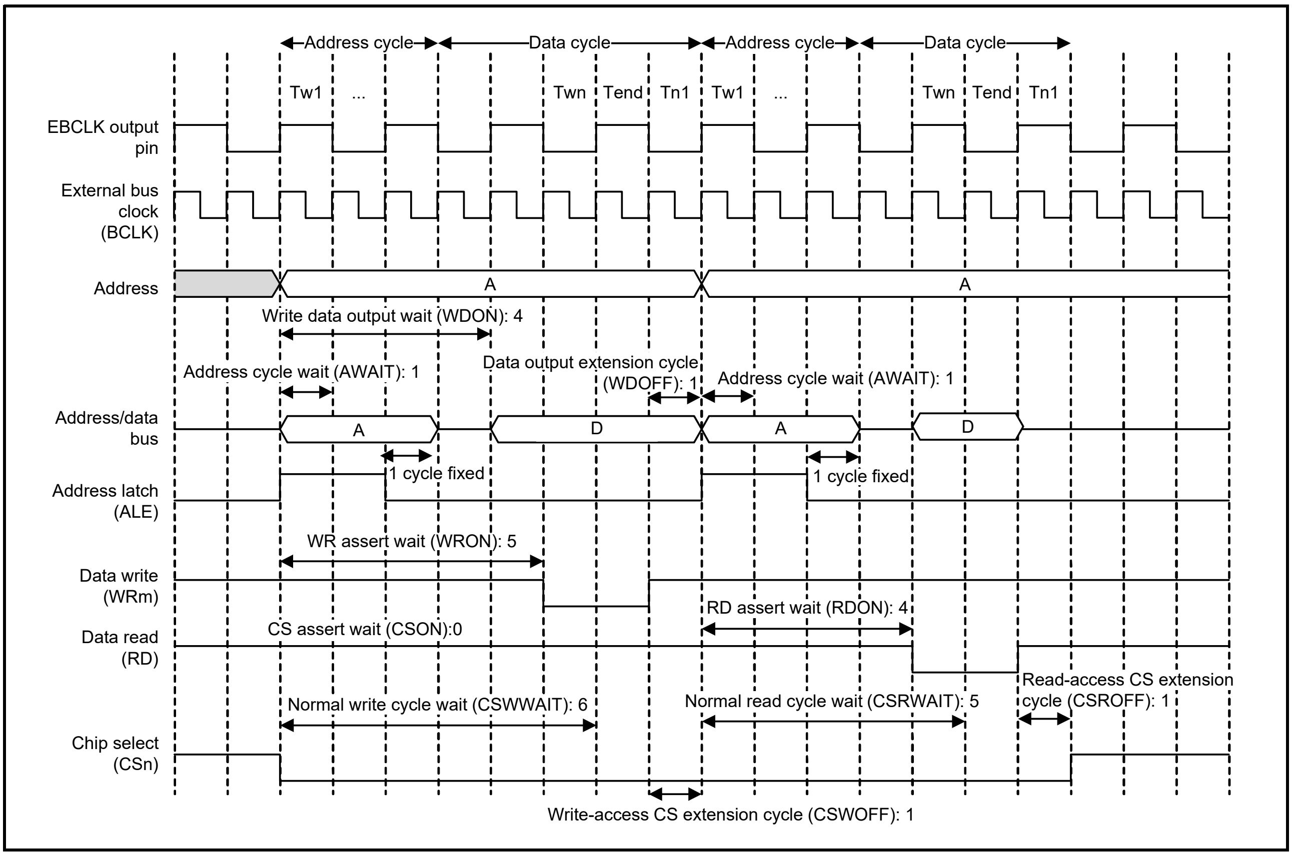


Figure 15.29 Example of bus timing with address/data multiplexed I/O interface (m = 0, 1)

15.5.3 External Wait Function

Wait cycles can be extended by the WAIT signal beyond the length of the normal access cycle wait specified in the CSRWAIT[4:0] and CSWWAIT[4:0] bits in CSnWCR1, and the page access cycle wait specified in the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1.

When external wait is enabled (EWENB = 1 in CSnMOD), wait cycles are inserted while the WAIT signal is held low. When external wait is disabled (EWENB = 0 in CSnMOD), the WAIT signal has no effect. All wait cycles specified in CSnWCR1 are inserted independently of the WAIT signal.

(1) Normal access

Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT signal goes high.

(2) Page access

The first access operation is the same as the normal access operation. Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT signal is held low. The wait cycle (Tend) ends at the next cycle after the WAIT signal goes high.

For the second and subsequent accesses, sampling of the WAIT signal begins on completion of the page access wait cycle (Tend). The page access wait cycle is extended while the WAIT signal is held low, and ends (Tend) at the next cycle after the WAIT signal goes high.

Figure 15.30 to Figure 15.33 show examples of external wait insertion timing with the separate bus interface.

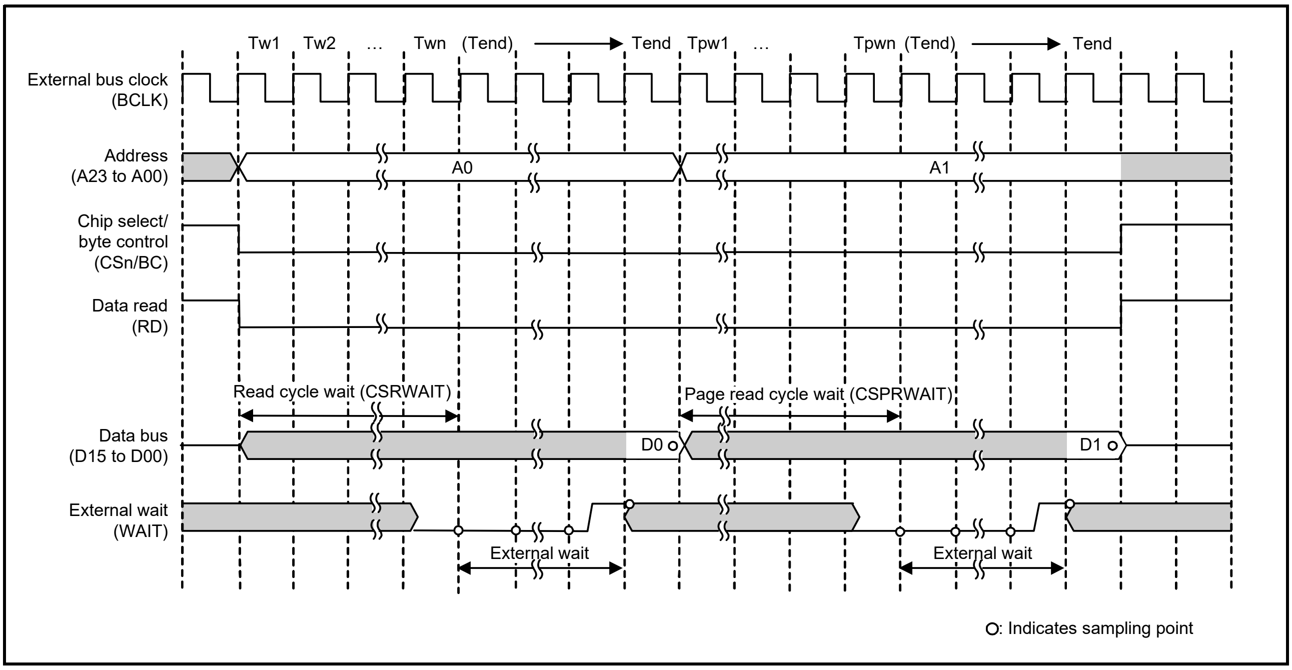


Figure 15.30 Example external wait timing for page read access to 16-bit bus space (when 1/1 BCLK is selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

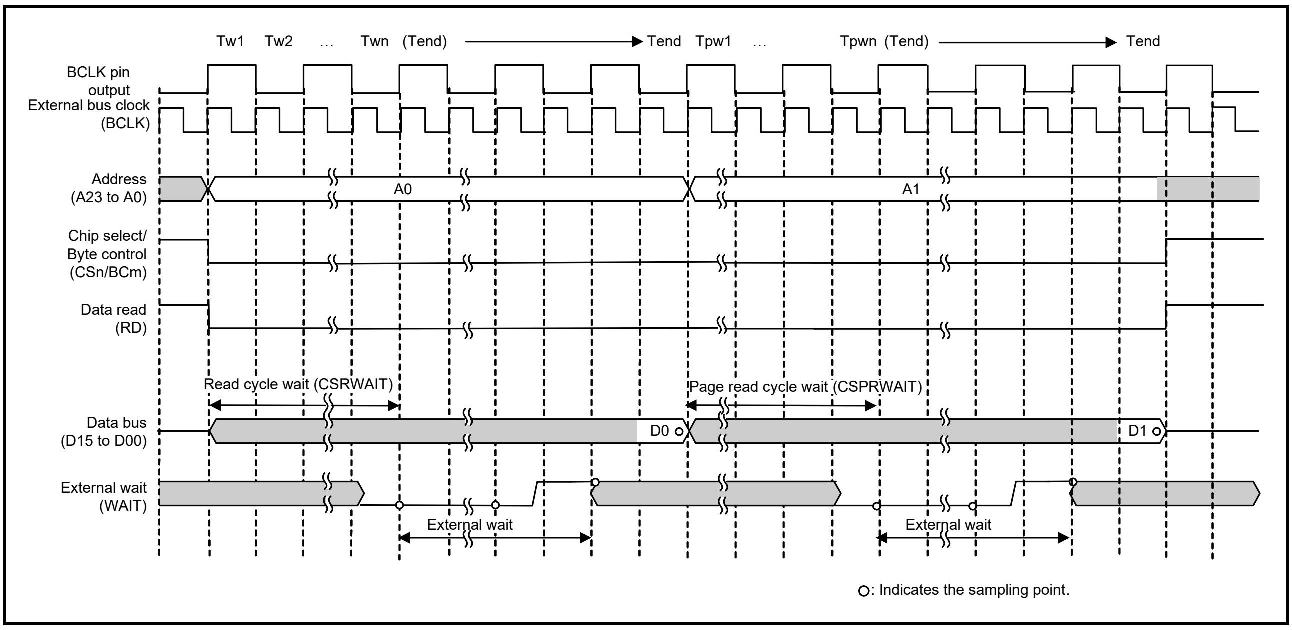


Figure 15.31 Example external wait timing for page read access to 16-bit bus space (when 1/2 BCLK is Selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

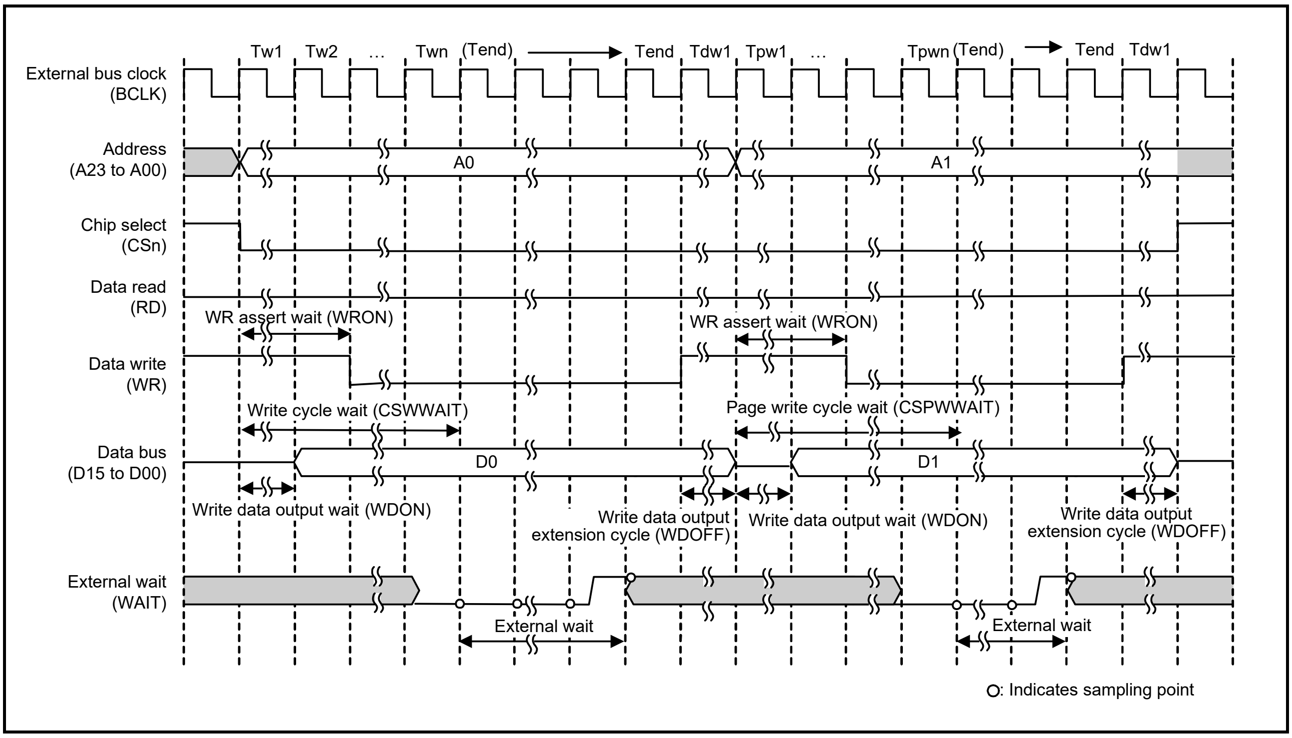


Figure 15.32 Example external wait timing for page write access to 16-bit bus space in byte strobe mode (when 1/1 BCLK is selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

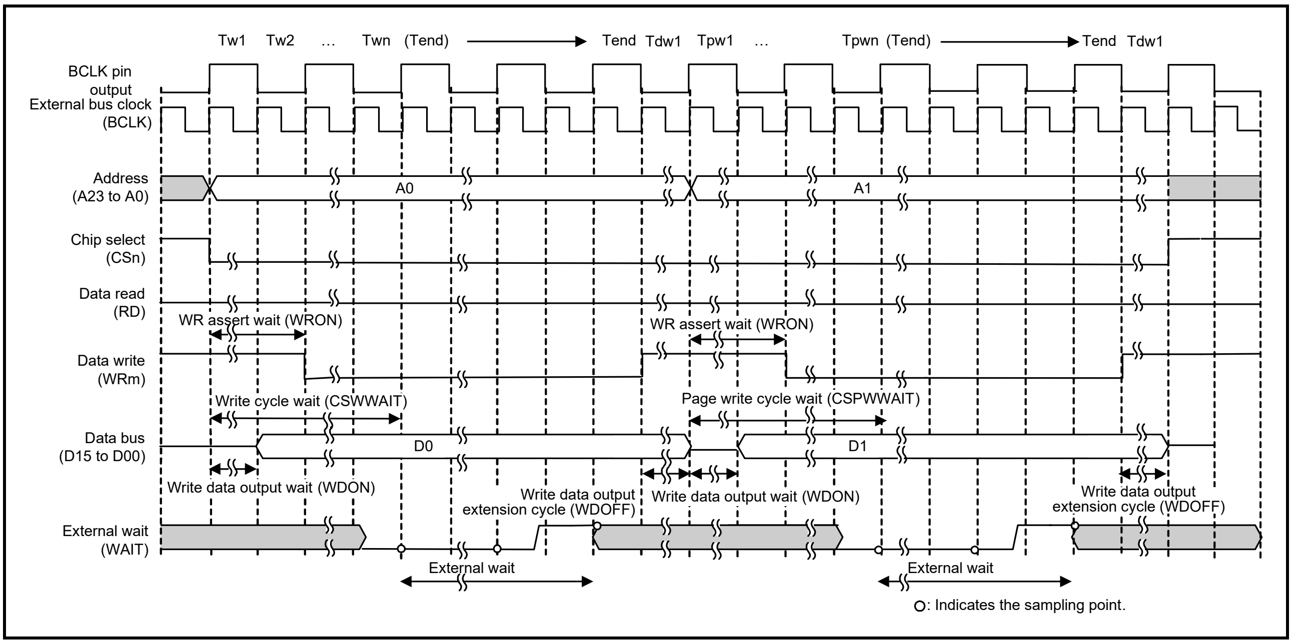


Figure 15.33 Example external wait timing for page write access to 16-bit bus space in byte strobe mode (when 1/2 BCLK is selected with the BCLK Pin Output Select bit) (n = 0 to 7, m = 0, 1)

(3) Address/data multiplexed I/O interface

In a data cycle with the address/data multiplexed I/O interface, programmed waits and pin waits using the WAIT pin can be inserted in the same way as that with the separate bus interface.

Address cycles are not affected by the wait control settings. Figure 15.34 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

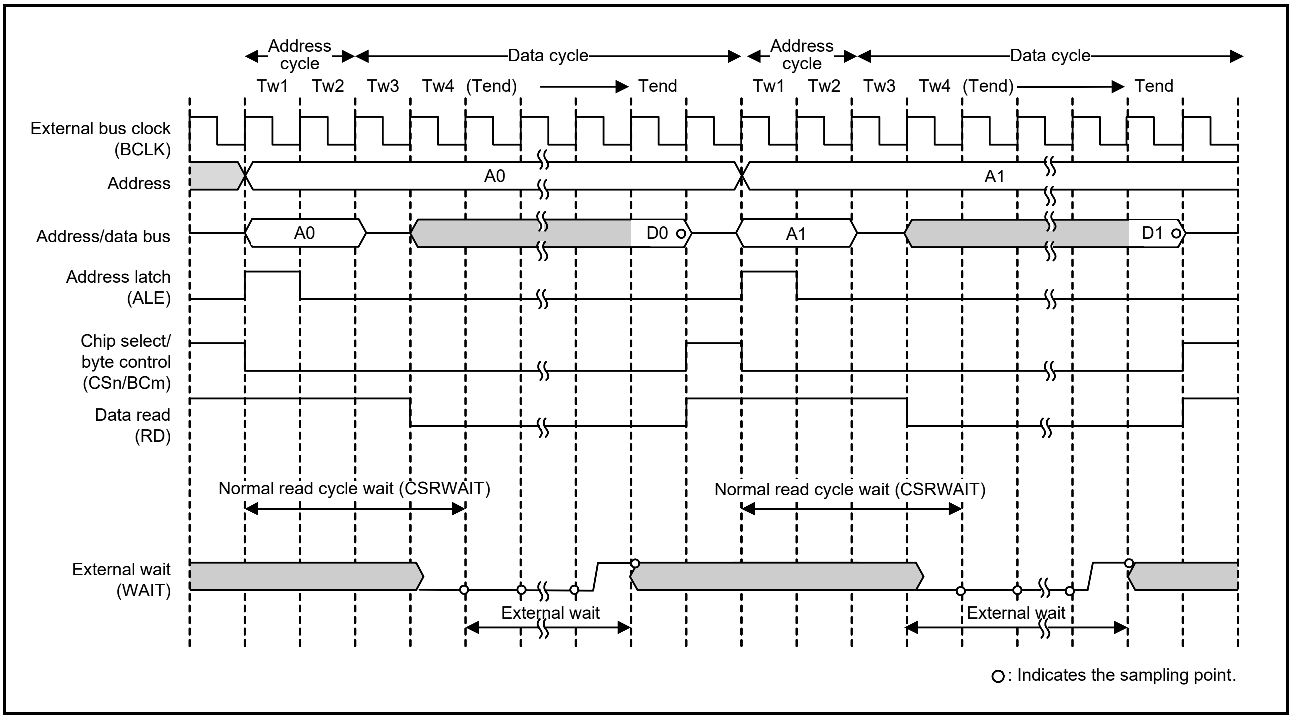


Figure 15.34 Example external wait Insertion timing with address/data multiplexed I/O interface (m = 0, 1)

15.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the Recovery Cycle Insertion Enable bit in CSRECEN to 1.

The number of recovery cycles to be inserted after read cycles and write cycles can be independently set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles must be set with the WRCV[3:0] bits for the associated area. When the preceding bus cycle is a read access, the number of read recovery cycles must be set with the RRCV[3:0] bits for the associated area. For example, when a CS1 read access occurs after a CS0 read access, the number of recovery cycles to be inserted between them is set in the RRCV[3:0] bits in CS0REC.

Recovery cycle insertion can be enabled or disabled with RCVENi (i = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access, and with RCVENMj (j = 0 to 7) when the preceding bus access is an address/data multiplexed bus access.

Recovery cycles can be inserted on any of the following conditions:

- After a read access to the external bus, a read access is made to the external bus in the same area
- After a read access to the external bus, a read access is made to the external bus in a different area
- After a read access to the external bus, a write access is made to the external bus in the same area
- After a read access to the external bus, a write access is made to the external bus in a different area
- After a write access to the external bus, a read access is made to the external bus in the same area
- After a write access to the external bus, a read access is made to the external bus in a different area
- After a write access to the external bus, a write access is made to the external bus in the same area
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, for example when the CSn signal (n = 0 to 7) is negated. A high-level period of the CSn signal is inserted for the specified recovery cycle period starting from this point.

In the fastest case, the CSn signal for the next round of bus access is asserted immediately after the end of the recovery cycles. Even if the next request for access to an external address space is generated during the recovery period, the next access over the external bus starts immediately after the end of the recovery cycles.

When two or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles. However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer, even if the recovery cycle insertion condition is satisfied. See Figure 15.37.

Similarly, during normal access with page access enabled, recovery cycles are not inserted between bus access cycles but only after the last bus access cycle of the transfer. With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 15.35 to Figure 15.37 show examples of recovery cycle insertion with the separate bus interface.

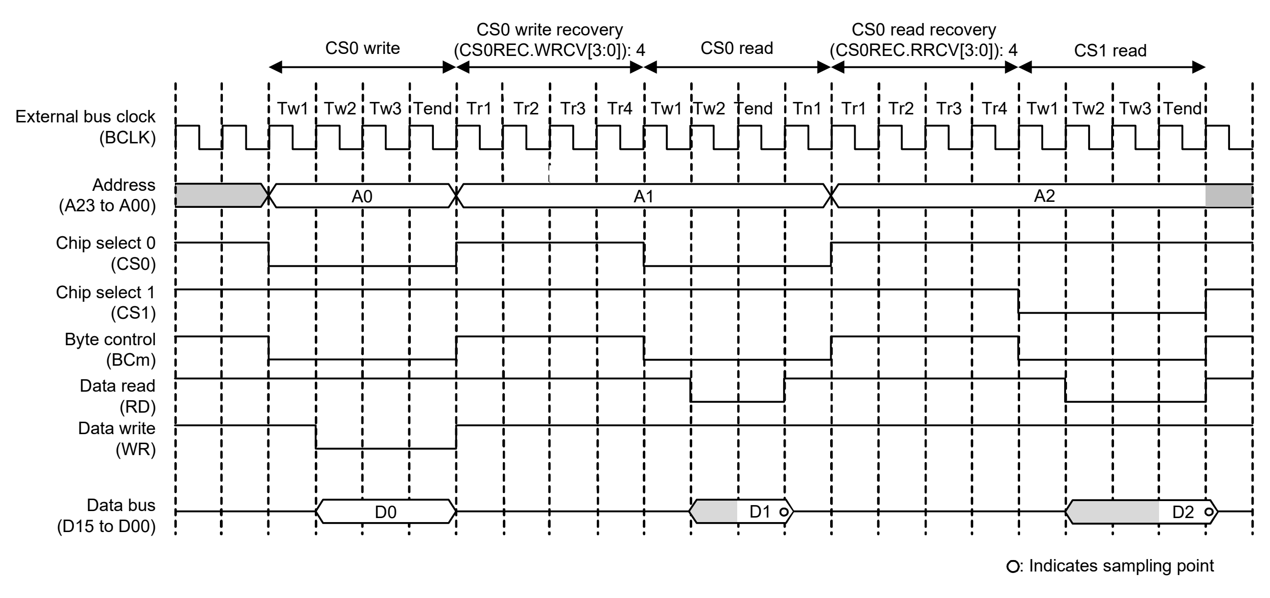


Figure 15.35 Example recovery cycle insertion with separate bus interface (m = 0, 1)

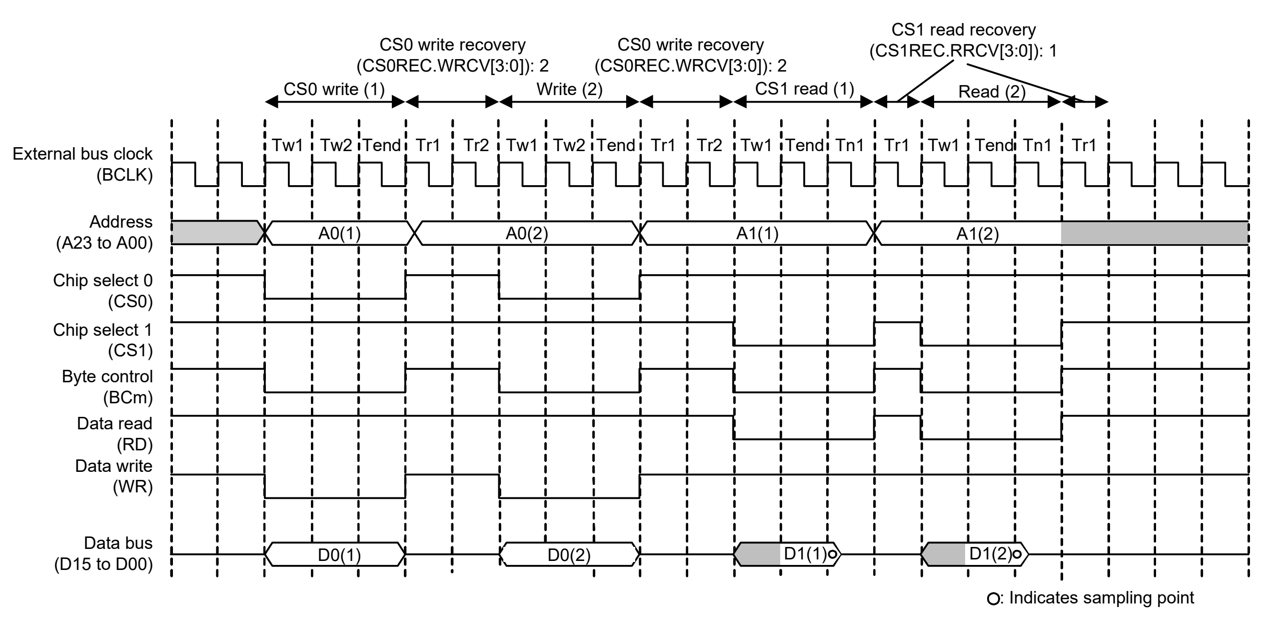


Figure 15.36 Example recovery cycle insertion when bus access is split, with separate bus interface and normal access (m = 0, 1)

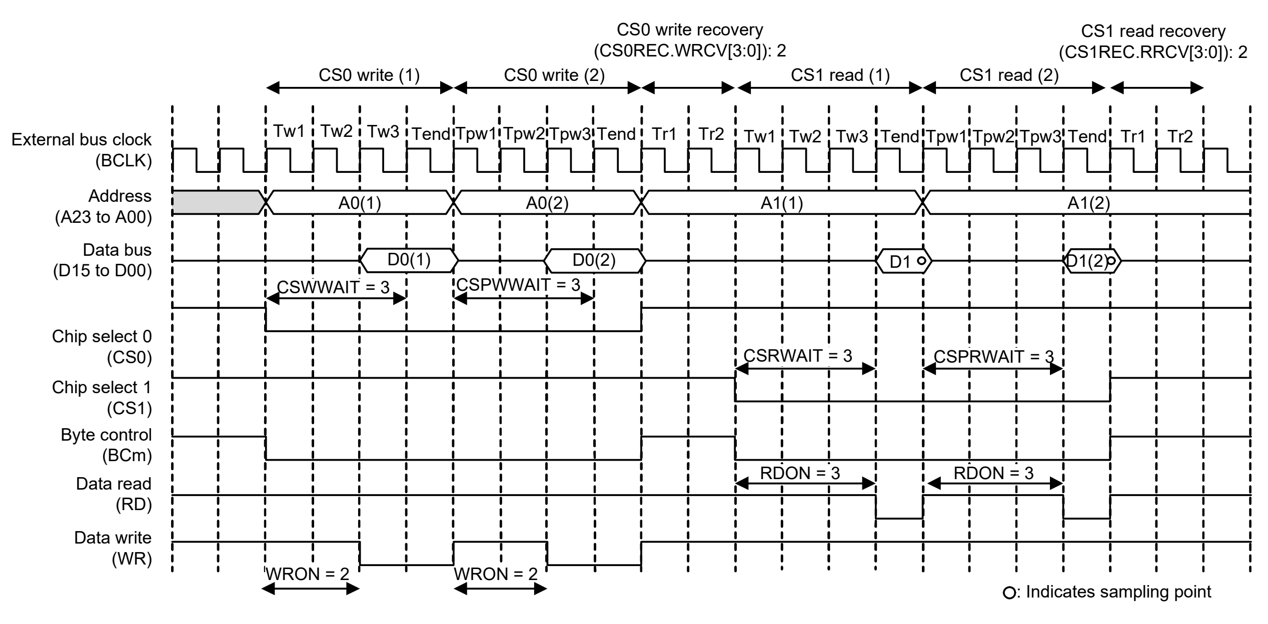


Figure 15.37 Example recovery cycle insertion when bus access is split, with separate bus interface and page access (m = 0, 1)

Figure 15.38 shows an example operation when BCLK/2 is selected as the frequency division in the EBCLK Pin Output Select bit.

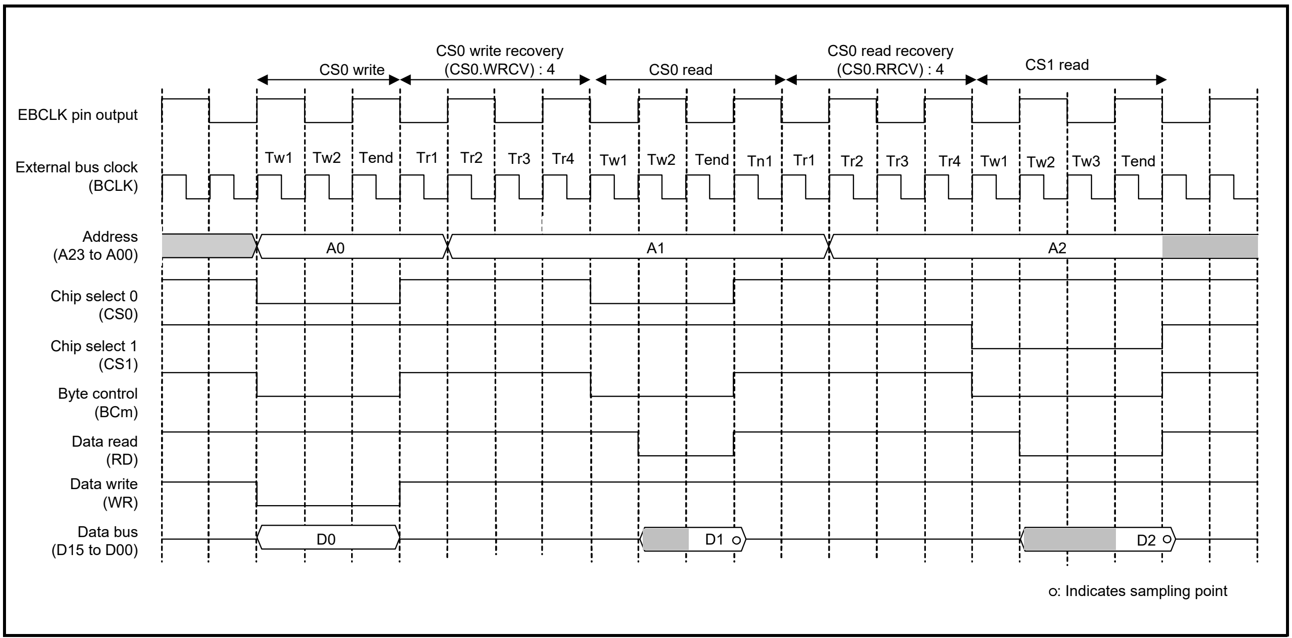


Figure 15.38 Example operation for recovery cycles when BCLK/2 is selected in the EBCLK Pin Output Select bit, with normal access through a separate bus interface (m = 0, 1)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 15.39 and Figure 15.40 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

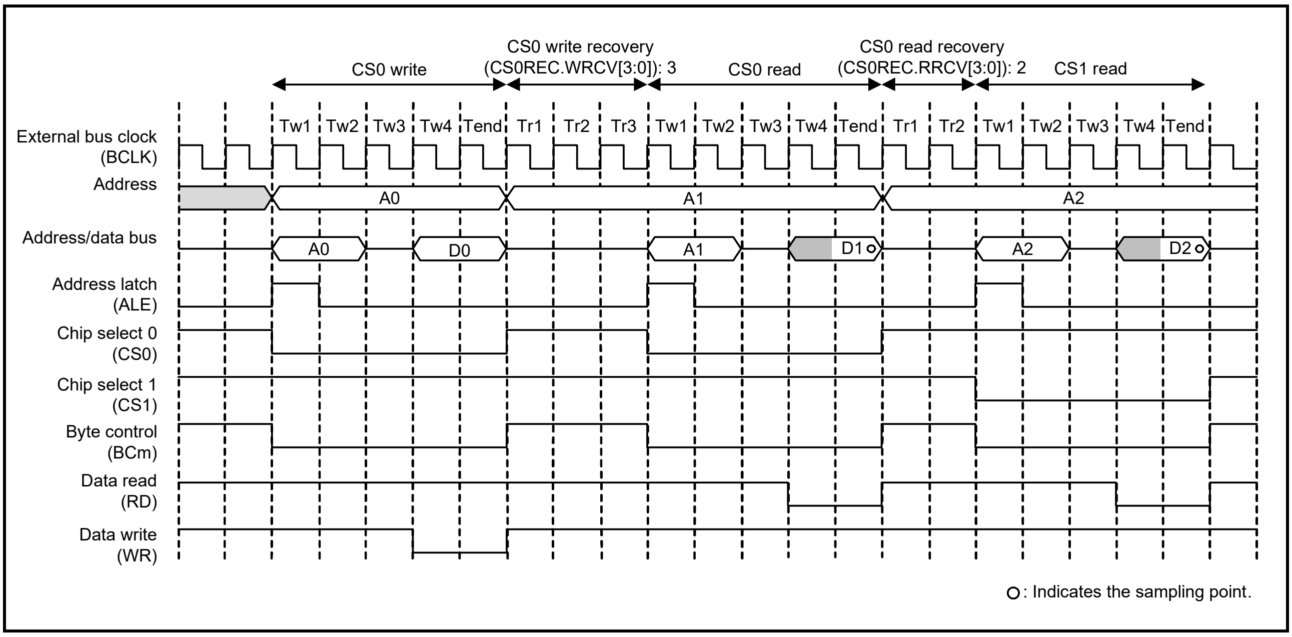


Figure 15.39 Example of recovery cycle insertion with address/data multiplexed I/O interface (m = 0, 1)

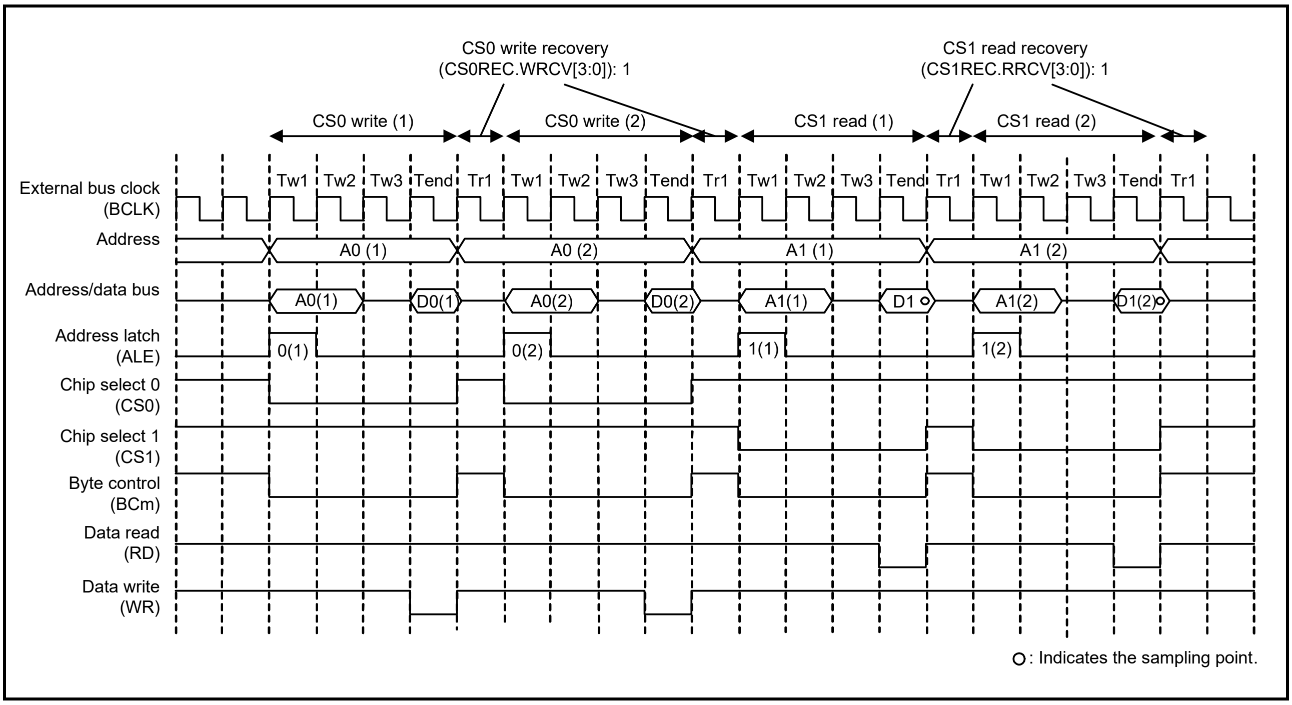


Figure 15.40 Example of recovery cycle insertion when a bus access is split with address/data multiplexed I/O interface (m = 0, 1)

15.5.5 No Access State

When no external address space is accessed, the CSn, BCn, WRn, RD signals are high, ALE signal is low, and D15 to D00 are in the high-impedance state.

15.5.6 Write Buffer Function (External Bus)

In write access, the main bus is released by writing data to the write buffer before the access is complete. This allows the next round of bus access to start. However, if the next access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are complete.

Figure 15.41 shows an example of operation when the write buffer function is in use. When this function is in use, if the next operation after an external write is an internal access, the internal access is executed in parallel with the external write, for example without waiting for completion of the latter operation.

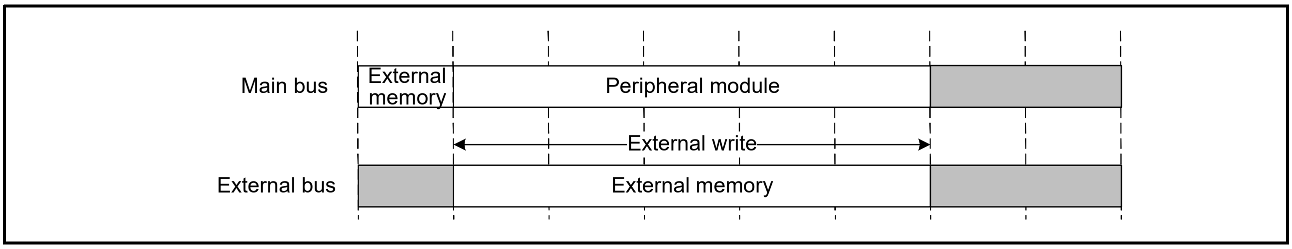


Figure 15.41 Example operation when the write buffer function is in use

15.5.7 Constraints

(1) Constraints on using a separate bus interface

Table 15.11 lists the constraints that apply to bits in the CSn Wait Control Register 1 (CSnWCR1) and CSn Wait Control Register 2 (CSnWCR2) when normal and page accesses occur.

Even if the Page Read Access Enable bit or Page Write Access Enable bit in the CSn Mode Register is set to enable (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first page access or access that does not fall within the scope of a page access is a normal access operation. Because of this, constraints on normal access must be satisfied.

Table 15.11 Constraints on normal access and page access

Constraints on normal access		Constraints on page access	
Reading	Writing	Reading	Writing
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON

Note: When two or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles and are inserted only after the last bus access cycle of the transfer.

(2) Constraints on using address/data multiplexed bus interface

In the address/data multiplexed I/O space, page accesses are invalid. If a page access setting is specified, the setting is ignored and the normal read or write operation is performed.

Table 15.12 Constraints at the time of normal access

Constraints at the time of normal access	
Reading	Writing
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON AWAIT[1:0] + 2 ≤ RDON CSON[2:0] ≤ AWAIT	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON AWAIT[1:0] + 2 ≤ WRON AWAIT[1:0] + 2 ≤ WDON CSON[2:0] ≤ AWAIT

(3) Constraint on pin multiplexing between the A00 and BC0 functions

Setting the single-write strobe mode is prohibited in the 8-bit bus space.

(4) Constraints when BCLK/2 is selected in the EBCLK Pin Output Select bit

When 1/2 cycle of BCLK is selected in the EBCLK Pin Output Select bit, the external bus access cycle starts on the rising edge of the EBCLK pin output. However, when 2 or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle can start on the falling edge of the EBCLK pin output, depending on the wait cycle settings. Set the registers appropriately for the specifications of connected devices.

(5) Instruction code constraint

You must fix the instruction code to little-endian order.

15.6 SDRAM Area Controller Operation

This section describes how the SDRAM area controller (SDRAMC) is enabled and the SDRAM bus width is set, followed by a description of the SDRAMC operations, including read, write, auto-refresh, self-refresh, initialization sequence, and mode register settings.

15.6.1 Enabling/Disabling SDRAM Access and Setting the SDRAM Bus Width

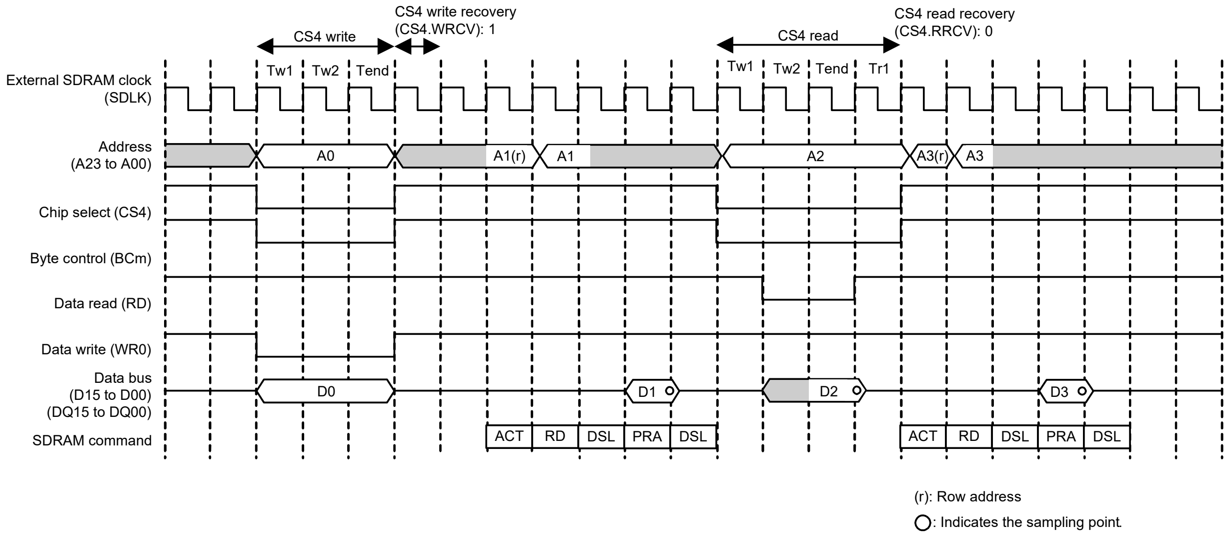
SDRAM access can be enabled or disabled using the SDC Control Register (SDCCR). The SDRAM bus width can also be set using SDCCR. The refresh operation is available even when the operation of the SDRAM address space is disabled, as long as self-refresh or auto-refresh is enabled.

15.6.2 No Access State

When no external address space is accessed, the SDCS, WE, RAS and CAS signals are high.

15.6.3 Insertion of Recovery Cycles

When access to the SDRAM area follows access to the CS area, data recovery cycles are inserted for the CS area controller (CSC). If the number of recovery cycles for the CSC is 0, the ACT command for the next SDRAM access is issued immediately after negation of CSn signal at the earliest. If the number of recovery cycles are not 0, the ACT command is issued 2 cycles after the specified recovery cycle period elapsed after negation of CSn signal at the earliest. Because no data conflicts can occur during access to the SDRAM area, there is no need to set data recovery cycles for the SDRAM (fixed to 0 cycle).



Note: When using SDRAM and CS Area access simultaneously.
- EBCLK cannot be used.
- CS0 to CS3 cannot be used.

Figure 15.42 Example of recovery timing for SDRAM access

15.6.4 Write Buffer Function

In write access, the main bus is released by writing data to the write buffer before access is complete. This allows the next round of bus access to start. However, if the next access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are complete.

15.6.5 SDRAM Commands

To control the SDRAM, the SDRAMC issues a command for each bus cycle. Commands are defined by a combination of the SDCS, RAS, CAS, WE, CKE, and other signals. Table 15.13 lists the commands issued by the SDRAMC.

Table 15.13 SDRAMC commands

							CKE			
Name	**Abbreviation**	**Command**	**SDCS**	**RAS**	**CAS**	**WE**	**n-1**	**n**	**BA1**	**BA0**
DESL	DSL	Device deselect	H	x	x	x	H	x	x	x
ACTV	ACT	Bank active	L	L	H	H	H	x	V	V
READ	RD	Read	L	H	L	H	H	x	V	V
WRIT	WRI	Write	L	H	L	L	H	x	V	V
PALL	PRA	All bank precharge	L	L	H	L	H	x	x	x
REF	RFA	Auto-refresh	L	L	L	H	H	x	x	x
MRS	MRS	Mode register set	L	L	L	L	H	x	L	L
SELF	RFS	Self-refresh entry	L	L	L	H	H	L	x	x
SELFX	RFX	Self-refresh end	H	x	x	x	L	H	x	x

Note: H = high level, L = low level, V = valid, x = don't care.
n = command issue cycle, n - 1 = 1 cycle before the command is issued.

15.6.6 Conditions for Setting the SDRAMC Registers

The SDRAMC registers must only be modified when all the conditions shown in Table 15.14 are satisfied.

Table 15.14 Conditions for register modification

Function or operation	Registers	Conditions
Self-refresh	SDSELF*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is enabled (SDRFEN.RFEN = 1).
Auto-refresh	SDRFCR	Self-refresh operation is disabled (SDSELF.SFEN = 0)
	SDRFEN	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Initialization sequence	SDIR*1	SDICR is not set yet, and the same conditions as for SDICR modification are satisfied
	SDICR*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Address register	SDADR	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Timing register	SDTR	• Self-refresh operation is in progress (SDSELF.SFEN = 1) or • SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Mode register	SDMOD*1	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled (SDSELF.SFEN = 0).
Access mode register	SDAMOD	• SDRAM access is disabled (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled (SDRFEN.RFEN = 0) • Self-refresh operation is disabled (SDSELF.SFEN = 0).

Note 1. Before modifying this register, confirm that all the status bits in SDSR are 0.
Note 2. After writing 0 to the EXENB bit, confirm that it is cleared to 0.

15.6.7 Self-Refresh

Transition to or recovery from self-refresh mode is controlled with the SDRAM Self-Refresh Control Register (SDSELF). Immediately before the transition to self-refresh mode, an auto-refresh operation is performed. In self-refresh mode, the CKE signal is low. Immediately after recovery from self-refresh mode, the auto-refresh cycle starts.

Figure 15.43 and Figure 15.44 show timing examples of the transition to and recovery from self-refresh mode.

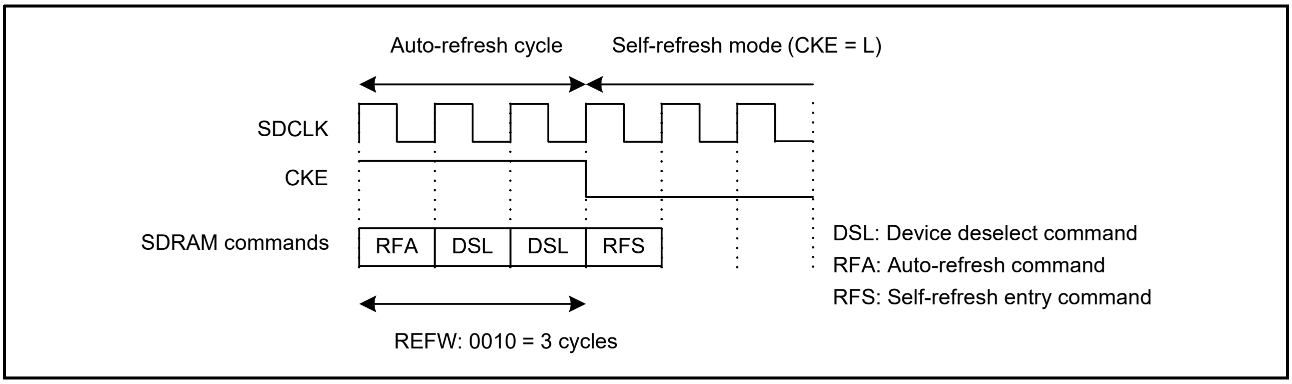


Figure 15.43 Example timing for transition to self-refresh mode when SDRFCR.REFW[3:0] = 0010b (3 cycles)

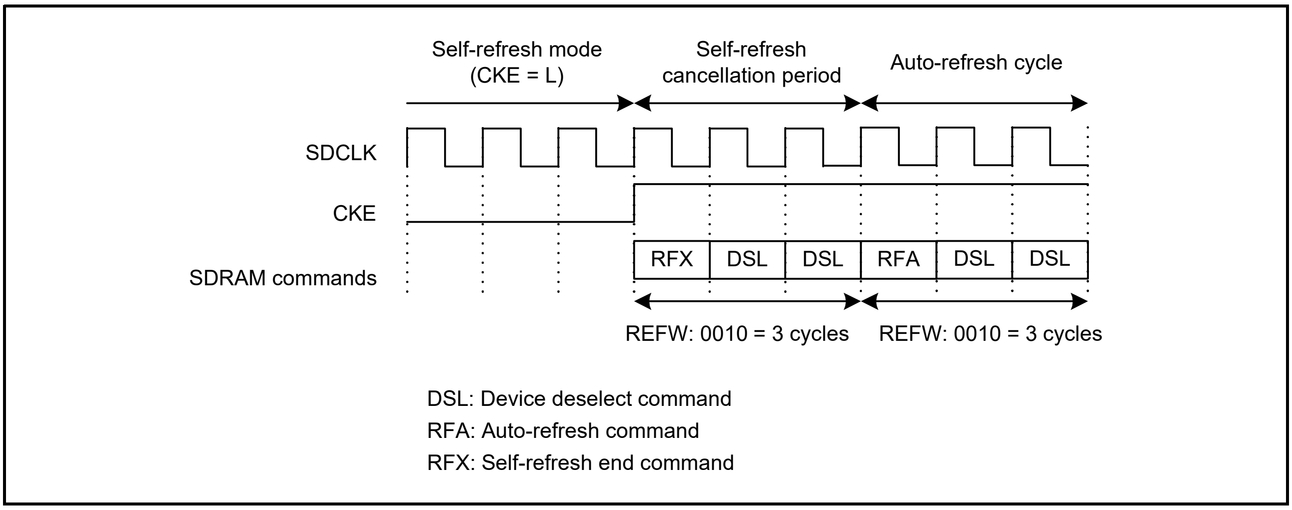


Figure 15.44 Example timing for recovery from self-refresh mode

(1) Self-refresh in Software Standby mode

When invoking self-refresh in Software Standby mode, first follow the procedure shown in section 15.6.11.2, Procedure for transitioning to and recovering from self-refresh mode. Next set up the transition to Software Standby mode. In this mode, set the Output Port Enable bit (OPE) in the Standby Control Register (SBYCR) to 1 to hold the output state of the address bus and bus control signals.

After canceling Software Standby mode, follow the procedure shown in section 15.6.11.2, Procedure for transitioning to and recovering from self-refresh mode. For details on invoking and canceling Software Standby mode, see section 11, Low Power Modes.

(2) Self-refresh in Deep Software Standby mode

Deep Software Standby mode is invoked from within Software Standby mode. On this transition, the pin states remain unchanged. Therefore, invoking of self-refresh in Deep Software Standby mode can be handled the same as for Software Standby mode with one additional setting. You must also set the I/O Port Keep bit (IOKEEP) in the Deep Software Standby mode Control Register (DPSBYCR) to 1.

Because the SDRAMC is reset internally when Deep Software Standby mode is canceled, the SDRAM control registers must be set again. After canceling Software Standby mode, follow the procedure in this section to cancel self-refresh.

Figure 15.45 shows self-refresh timing in Deep Software Standby mode. For details on invoking and canceling Deep Software Standby mode, see section 11, Low Power Modes.

To cancel self-refresh mode:

1. Set DPSBYCR.IOKEEP to 1 to keep the CKE signal output low in Deep Software Standby mode.
2. Start the clock supply to the SDRAMC.
3. Set the SDRAM control registers (SDCMOD, SDAMOD, SDADR, and SDTR) again. These registers were initialized by an internal reset on entering Deep Software Standby mode.
4. Enable an auto-refresh operation by setting SDRFEN.RFEN to 1.
5. Check that all the status bits in SDSR are cleared to 0 and set SDSELF.SFEN to 1 to select self-refresh mode again.
6. Modify the port settings for the SDRAM interface.
7. Set SDCKOCR.SDCKOEN to 1 to start the clock supply to the SDRAM with the SDCLK pin.
8. Check that all the status bits in SDSR are cleared to 0 and set SDSELF.SFEN to 0 to cancel self-refresh mode.

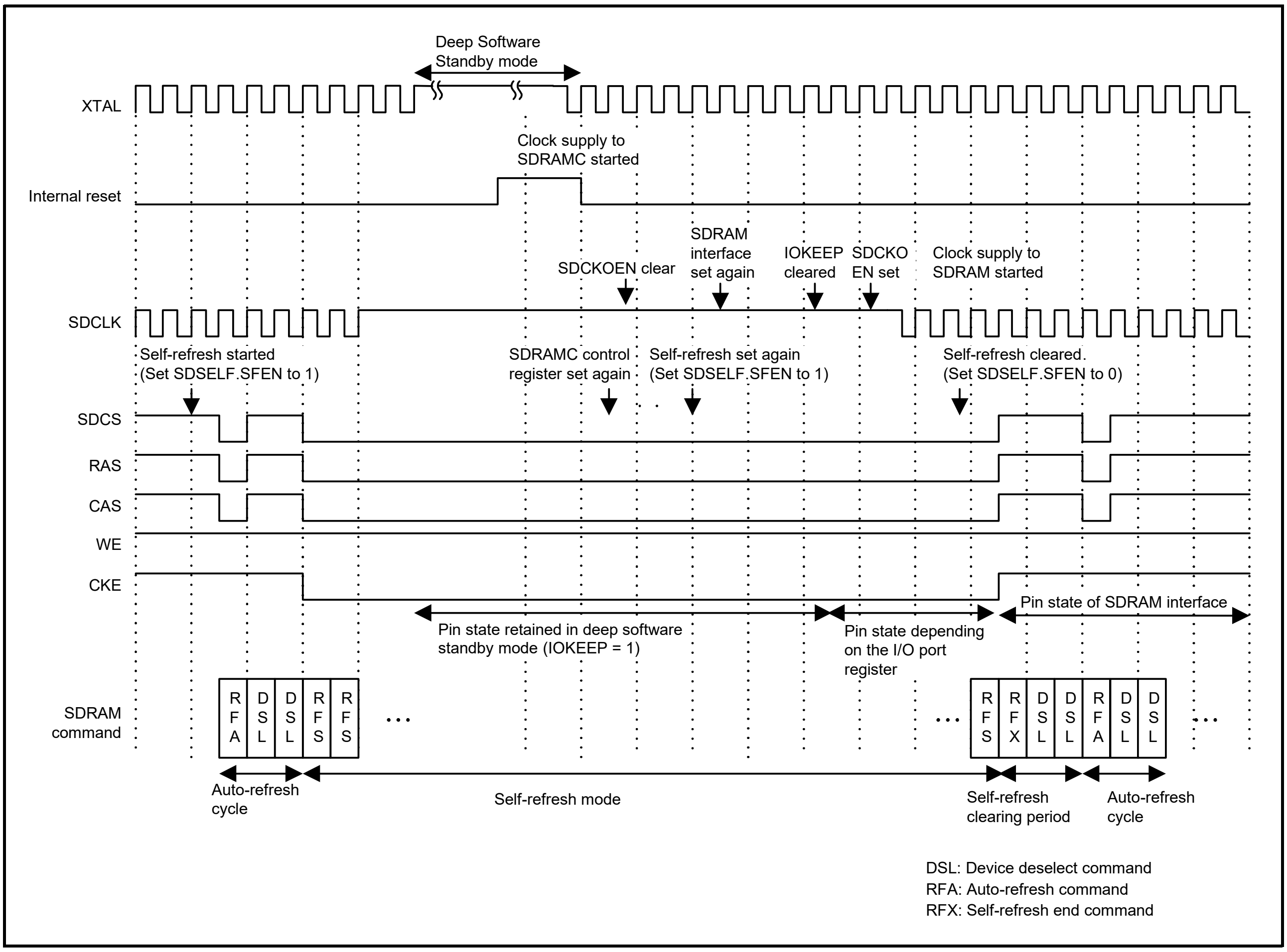


Figure 15.45 Example timing for self-refresh cycle in Deep Software Standby mode

15.6.8 Auto-Refresh

The auto-refresh cycle can be started by setting the Auto-Refresh Operation Enable bit (RFEN) in the SDRAM Auto-Refresh Control Register (SDRFEN) to 1. After the cycle starts, refresh requests are generated at fixed intervals determined by the refresh counter. However, because refresh requests are not accepted during read or write access, the auto-refresh cycle might be suspended. If an auto-refresh request is issued during consecutive accesses to the SDRAM, the auto-refresh cycle starts after completion of the bus access in response to a single transfer request from the bus master.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence. A CS area access and a refresh request can be made at the same if the SDCS, RAS, CAS, WE, and CKE signals, which are required for issuing the refresh command, are exclusively provided for SDRAM access.

The refresh counter is halted during a self-refresh operation. After recovery from the self-refresh mode, the auto-refresh cycle starts and the counter value is reset, resuming the counter operation.

Figure 15.46 shows a timing example of an auto-refresh cycle.

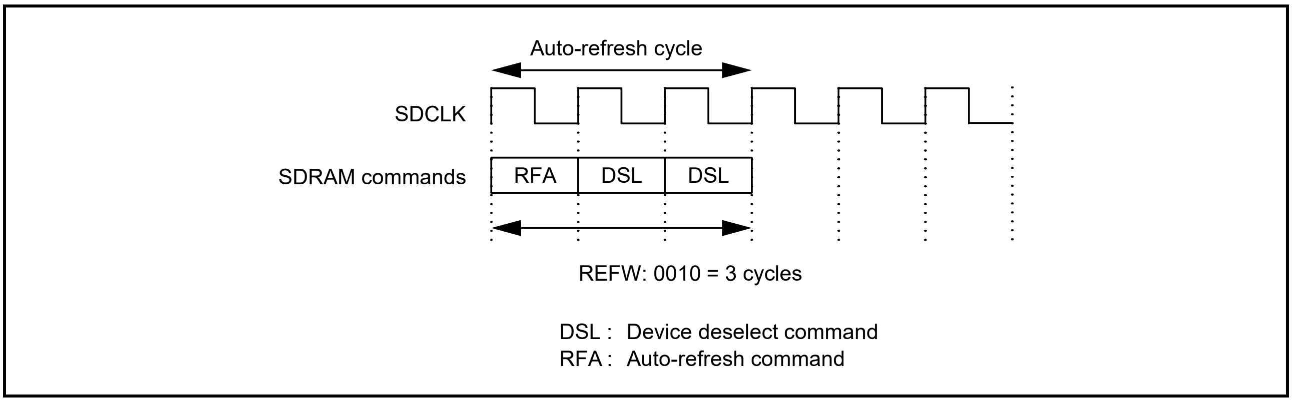


Figure 15.46 Example timing for auto-refresh cycle (1)

Figure 15.47 and Figure 15.48 show examples of operation when an auto-refresh request is generated during single access and continuous access.

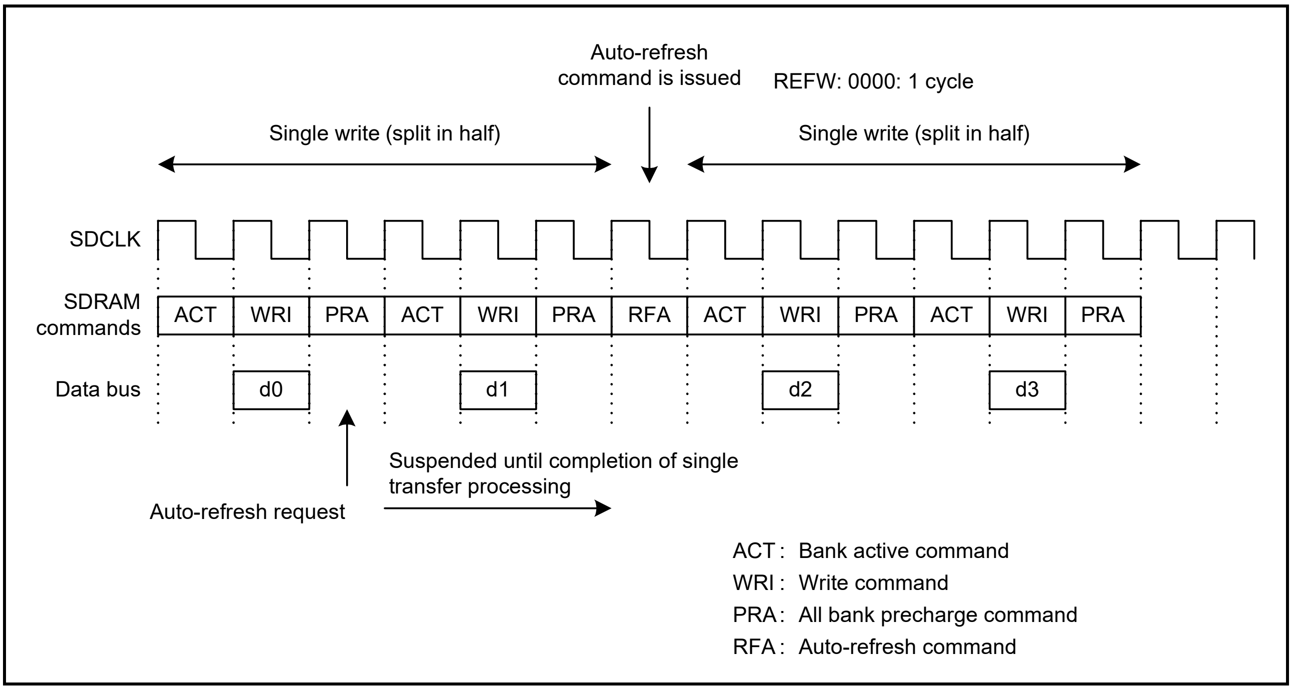


Figure 15.47 Example timing for auto-refresh cycle (2), when the auto-refresh request is made during single access

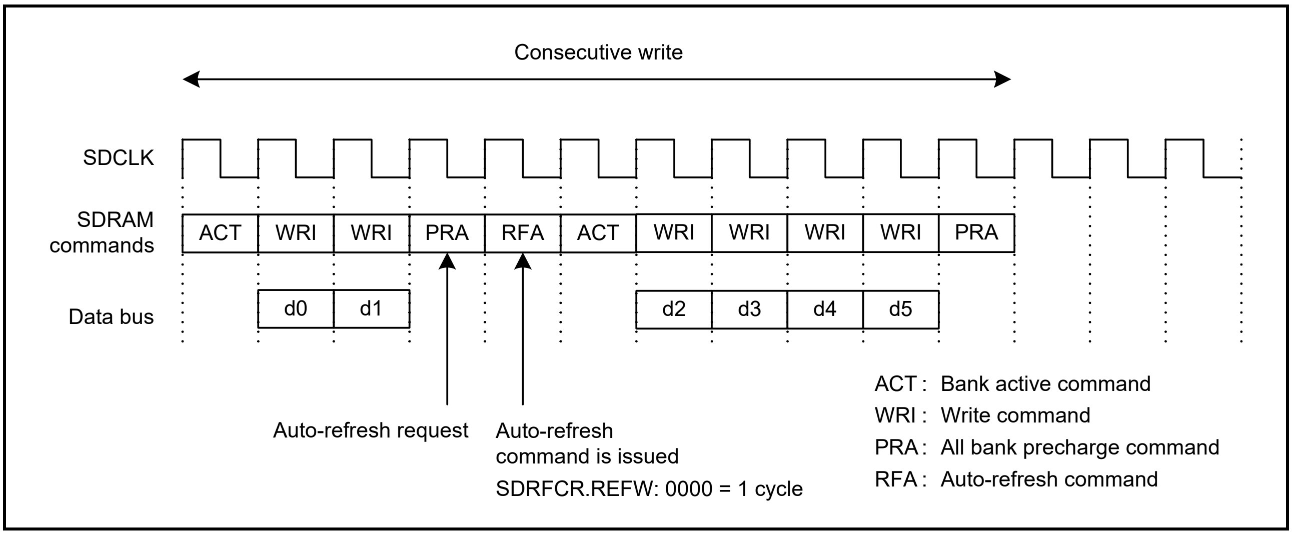


Figure 15.48 Example timing for auto-refresh cycle (3) when auto-refresh request is made during continuous access

15.6.9 Initialization Sequencer

The SDRAMC has a sequencer to issue SDRAM initialization commands. After a reset, the initialization sequencer must be activated without fail. Operation is not guaranteed if the SDRAM is not initialized.

The SDRAM initialization sequencer issues an all-bank precharge command followed by auto-refresh commands n times, where n = 1 to 15. The SDRAM initialization sequence timing can be set using the SDRAM Initialization Register (SDIR). The SDRAM initialization sequence can be activated using the SDRAM Initialization Sequence Control Register (SDICR). These registers must be set only when the conditions listed in Table 15.14 are satisfied.

Figure 15.49 shows a timing example of the SDRAM initialization sequence. When the ARFC[3:0] bits in SDIR are set so that auto-refresh operation is performed two or more times, auto-refresh cycles are repeated in the initialization sequence accordingly.

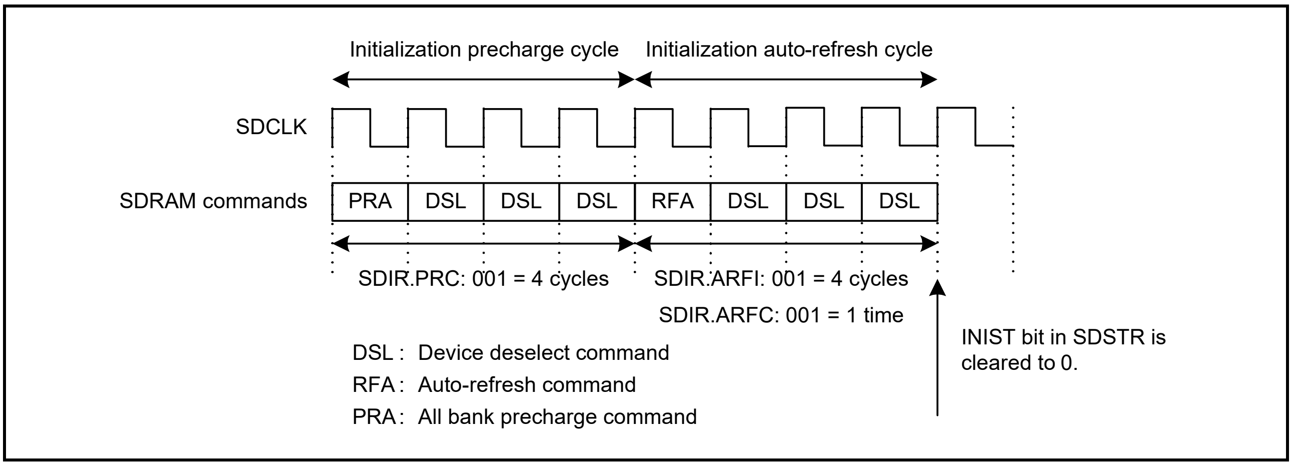


Figure 15.49 Example timing for SDRAM initialization sequence

15.6.10 Setting the Mode Register

Setting the SDRAM Mode Register (SDMOD) allows the mode register set command to be issued to the SDRAM and the value set in the MR[14:0] bits in SDMOD to be output to the lower bits of the address, specifically to A14 to A00 for 8-bit bus width or A15 to A01 for 16-bit bus width. Before setting the mode register, set the SDRAM Bus Width Select bits in the SDC Control Register (SDCCR.BSIZE[1:0]) to determine the data bus width of the SDRAM.

Figure 15.50 shows the mode register setting timing.

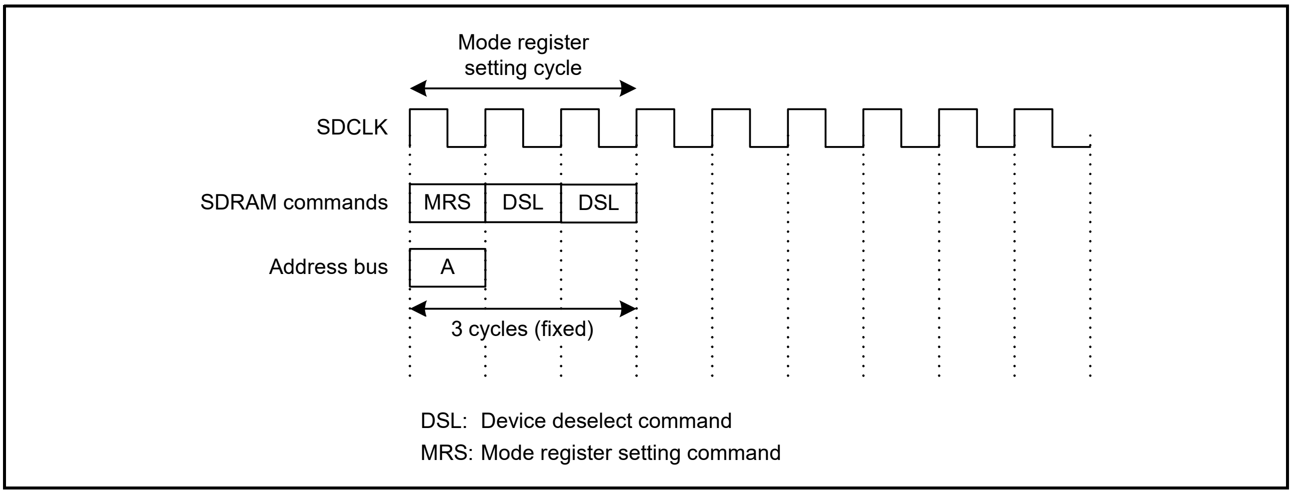


Figure 15.50 Mode register setting timing

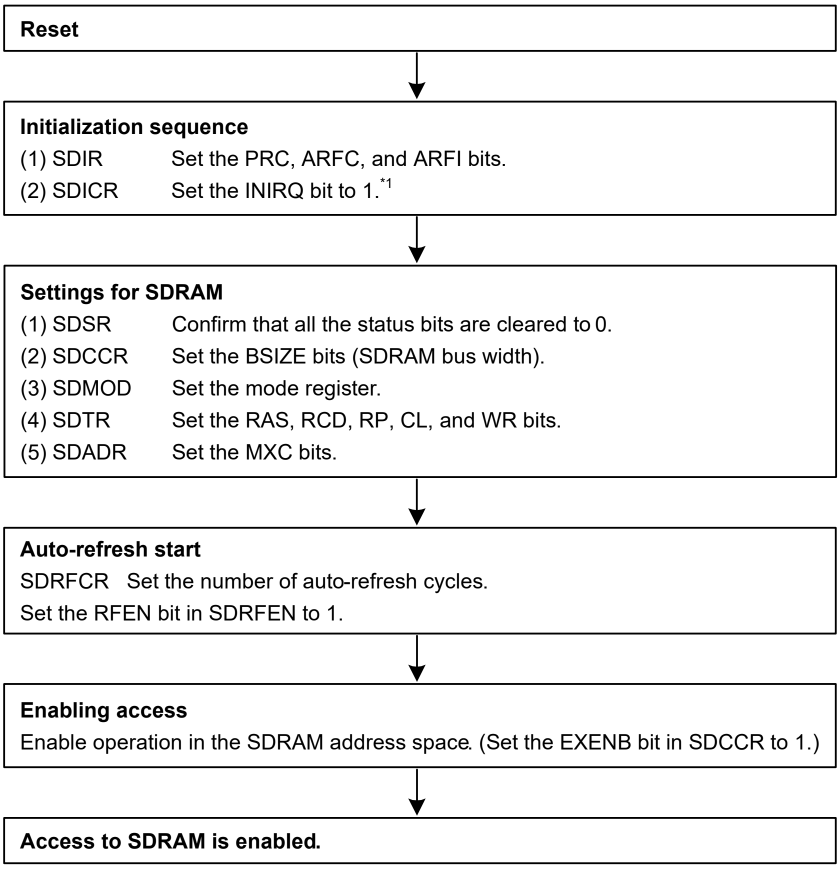
15.6.11 SDRAMC Setting Examples

This section describes the following:

- SDRAMC setting procedure
- Timing register setting examples
- Procedure for transitioning to and recovering from self-refresh mode.

15.6.11.1 SDRAMC access procedure

Figure 15.51 shows the SDRAMC setting procedure.



Reset

Initialization sequence

(1) SDIR Set the PRC, ARFC, and ARFI bits.

(2) SDICR Set the INIRQ bit to 1.[*1]

Settings for SDRAM

(1) SDSR Confirm that all the status bits are cleared to 0.

(2) SDCCR Set the BSIZE bits (SDRAM bus width).

(3) SDMOD Set the mode register.

(4) SDTR Set the RAS, RCD, RP, CL, and WR bits.

(5) SDADR Set the MXC bits.

Auto-refresh start

SDRFCR Set the number of auto-refresh cycles.

Set the RFEN bit in SDRFEN to 1.

Enabling access

Enable operation in the SDRAM address space. (Set the EXENB bit in SDCCR to 1.)

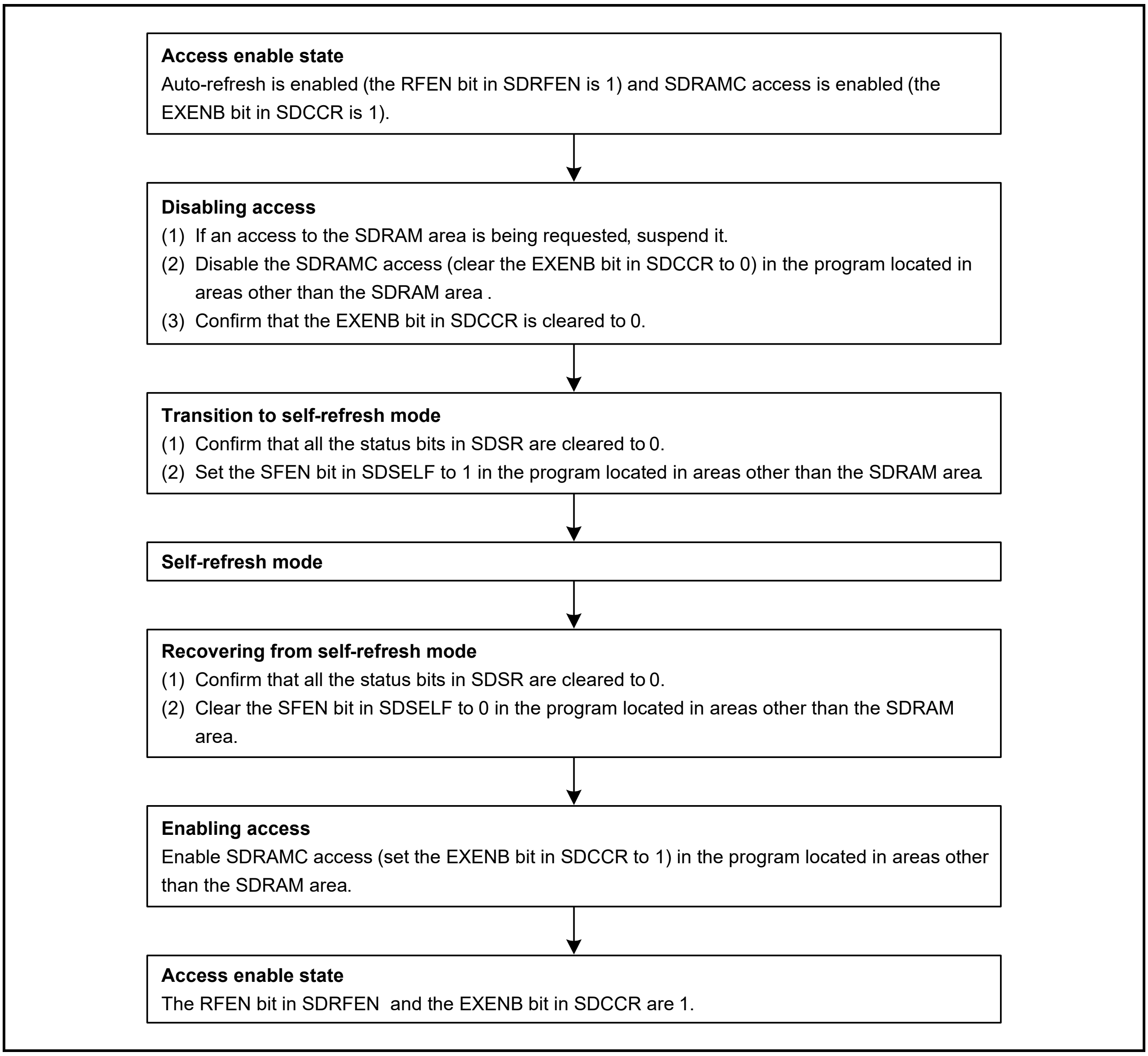
Access to SDRAM is enabled.

Note 1. When SDRAM bus width is 8-bit, before SDICR setting, set the BSIZE[1:0] bits in SDCCR to 10b.

Figure 15.51 SDRAMC setting procedure

15.6.11.2 Procedure for transitioning to and recovering from self-refresh mode

Figure 15.52 shows the procedure for transitioning to and recovering from self-refresh mode.



Access enable state
Auto-refresh is enabled (the RFEN bit in SDRFEN is 1) and SDRAMC access is enabled (the EXENB bit in SDCCR is 1).

Disabling access
(1) If an access to the SDRAM area is being requested, suspend it.
(2) Disable the SDRAMC access (clear the EXENB bit in SDCCR to 0) in the program located in areas other than the SDRAM area .
(3) Confirm that the EXENB bit in SDCCR is cleared to 0.

Transition to self-refresh mode
(1) Confirm that all the status bits in SDSR are cleared to 0.
(2) Set the SFEN bit in SDSELF to 1 in the program located in areas other than the SDRAM area.

Self-refresh mode

Recovering from self-refresh mode
(1) Confirm that all the status bits in SDSR are cleared to 0.
(2) Clear the SFEN bit in SDSELF to 0 in the program located in areas other than the SDRAM area.

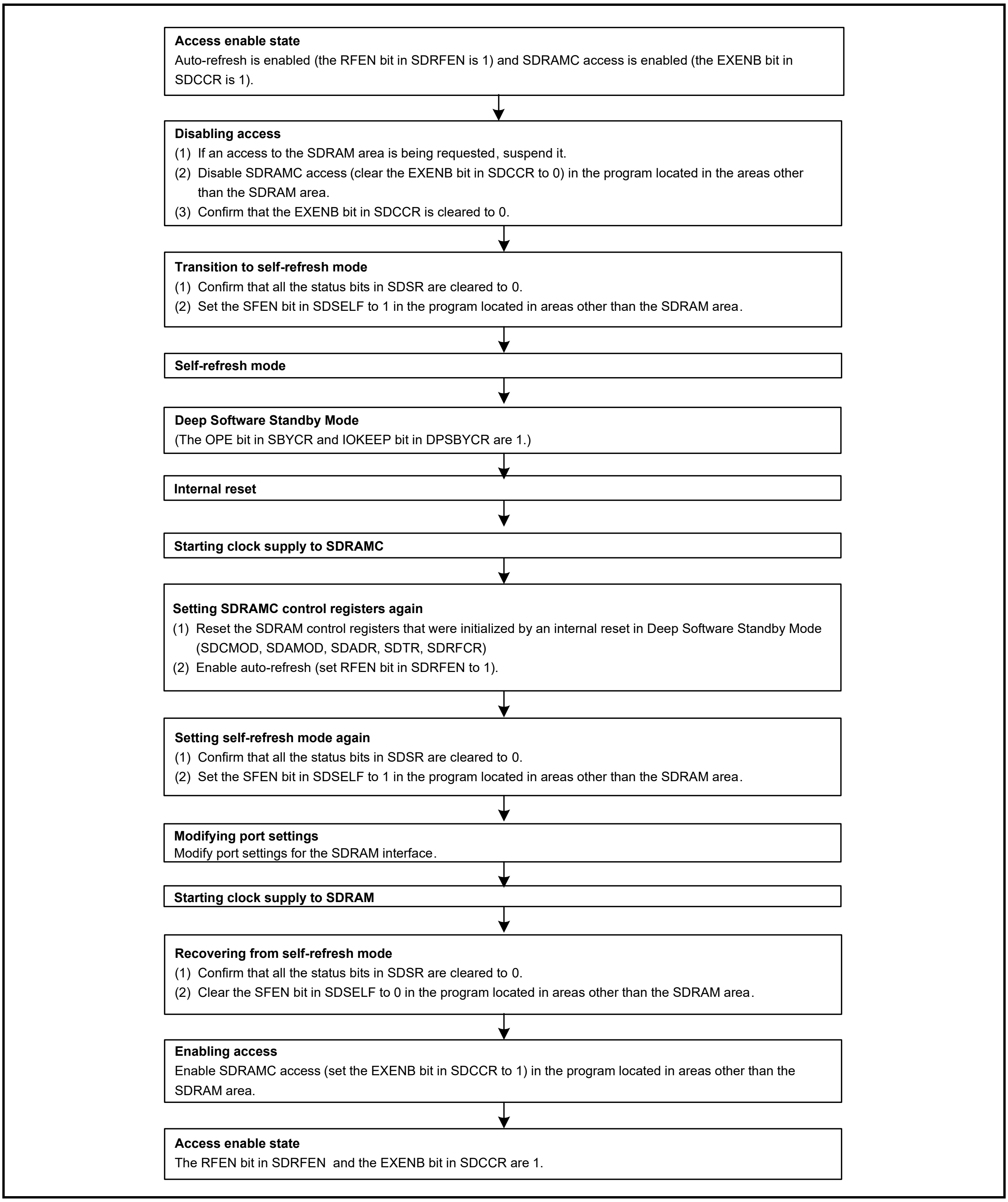
Enabling access
Enable SDRAMC access (set the EXENB bit in SDCCR to 1) in the program located in areas other than the SDRAM area.

Access enable state
The RFEN bit in SDRFEN and the EXENB bit in SDCCR are 1.

Figure 15.52 Procedure for transitioning to and recovering from self-refresh mode

Note: Self-refresh mode cannot be invoked during SDRAM access. SDRAM access must be disabled during both transition to and recovery from self-refresh mode. Follow the programming instructions shown in Figure 15.53.
Before transitioning to self-refresh mode, disable access to the SDRAM area.
During transition to self-refresh mode, self-refresh operation, and recovery from self-refresh mode, do not allow any operand access or instruction fetch, including prefetch to the SDRAM area, to be generated.

Figure 15.53 shows the procedure for transitioning to and recovering from self-refresh mode in Deep Software Standby mode.



Access enable state
Auto-refresh is enabled (the RFEN bit in SDRFEN is 1) and SDRAMC access is enabled (the EXENB bit in SDCCR is 1).

Disabling access
(1) If an access to the SDRAM area is being requested, suspend it.
(2) Disable SDRAMC access (clear the EXENB bit in SDCCR to 0) in the program located in the areas other than the SDRAM area.
(3) Confirm that the EXENB bit in SDCCR is cleared to 0.

Transition to self-refresh mode
(1) Confirm that all the status bits in SDSR are cleared to 0.
(2) Set the SFEN bit in SDSELF to 1 in the program located in areas other than the SDRAM area.

Self-refresh mode

Deep Software Standby Mode
(The OPE bit in SBYCR and IOKEEP bit in DPSBYCR are 1.)

Internal reset

Starting clock supply to SDRAMC

Setting SDRAMC control registers again
(1) Reset the SDRAM control registers that were initialized by an internal reset in Deep Software Standby Mode (SDCMOD, SDAMOD, SDADR, SDTR, SDRFCR)
(2) Enable auto-refresh (set RFEN bit in SDRFEN to 1).

Setting self-refresh mode again
(1) Confirm that all the status bits in SDSR are cleared to 0.
(2) Set the SFEN bit in SDSELF to 1 in the program located in areas other than the SDRAM area.

Modifying port settings
Modify port settings for the SDRAM interface.

Starting clock supply to SDRAM

Recovering from self-refresh mode
(1) Confirm that all the status bits in SDSR are cleared to 0.
(2) Clear the SFEN bit in SDSELF to 0 in the program located in areas other than the SDRAM area.

Enabling access
Enable SDRAMC access (set the EXENB bit in SDCCR to 1) in the program located in areas other than the SDRAM area.

Access enable state
The RFEN bit in SDRFEN and the EXENB bit in SDCCR are 1.

Figure 15.53 Procedure for transitioning to and recovering from self-refresh mode in Deep Software Standby mode

15.6.11.3 Timing register settings and access timing

This section describes the relationship between read and write timing and the settings in the SDRAM Timing Register (SDTR).

(1) Single read timing examples

Figure 15.54 to Figure 15.58 show the relationship between single read timing and the SDTR register settings. Table 15.15 shows the association between the figures and the SDTR register settings.

During read access, the next bus access is enabled at the earliest 2 cycles after the read data becomes valid. However, if two or more accesses occur for one transfer request, the next bus access is enabled at the earliest 1 cycle after the read data becomes valid, as shown in Figure 15.58.

Table 15.15 Association between timing figures and STDR register settings for single read timing

Figure number	RAS[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	CL[2:0] settings	Number of cycles
Figure 15.54	010	3	00	1	001	2	010	2
Figure 15.55	000	1	01	2	001	2	010	2
Figure 15.56	000	1	01	2	001	2	011	3
Figure 15.57, Figure 15.58	010	3	00	1	000	1	010	2

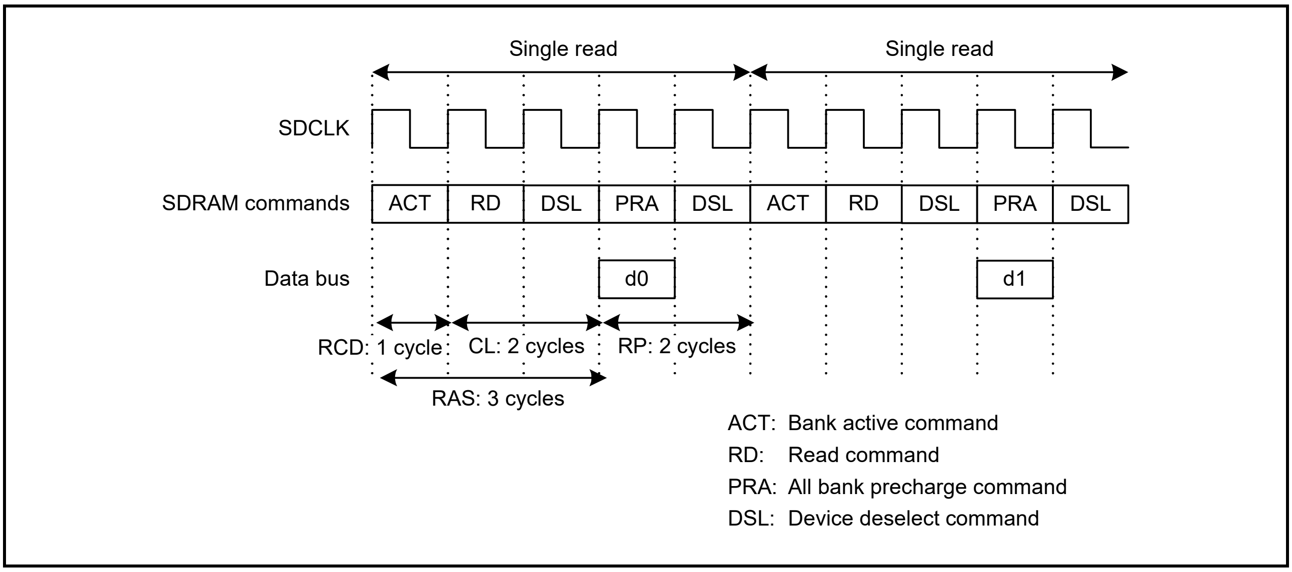
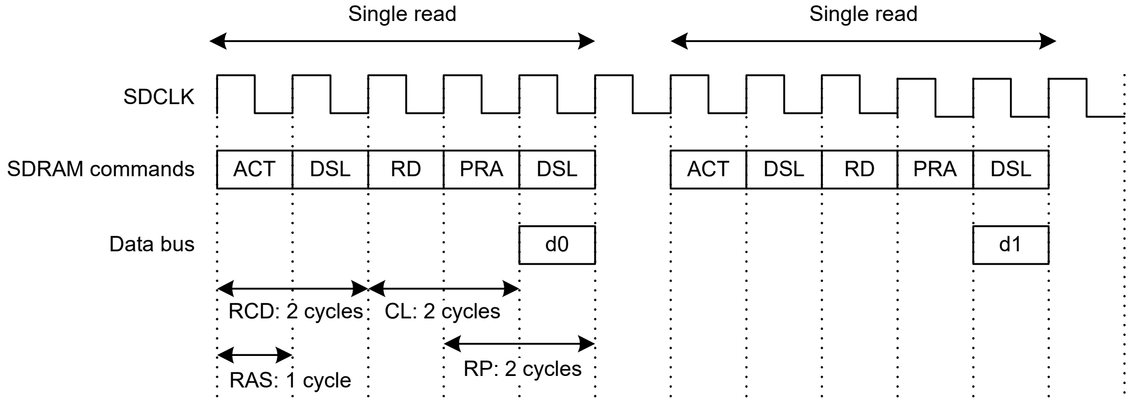


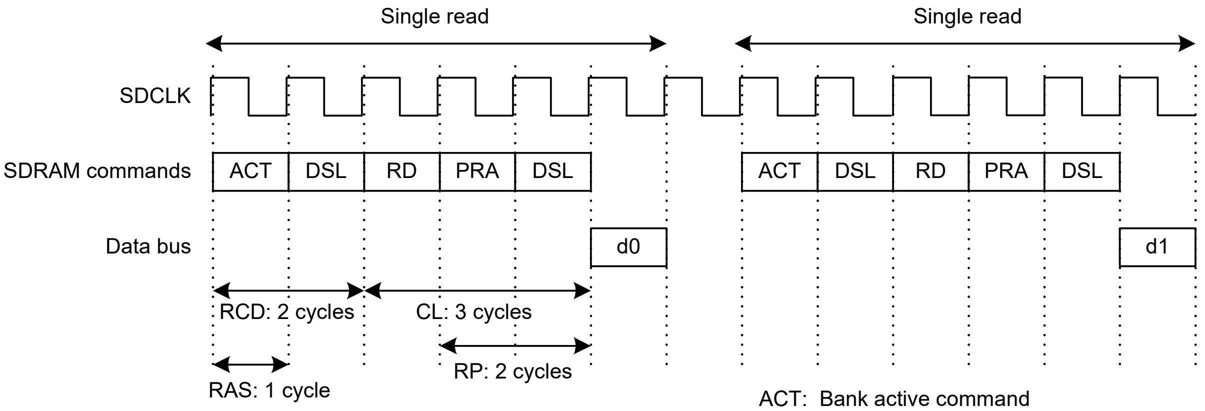
Figure 15.54 Example timing for single read (1)



Note: When the period set in the RAS bits ends earlier than the issue of the RD command, the PRA command is issued in the next cycle of the RD command cycle.

ACT: Bank active command
RD: Read command
PRA: All bank precharge command
DSL: Device deselect command

Figure 15.55 Example timing for single read (2)



ACT: Bank active command
RD: Read command
PRA: All bank precharge command
DSL: Device deselect command

Figure 15.56 Example timing for single read (3)

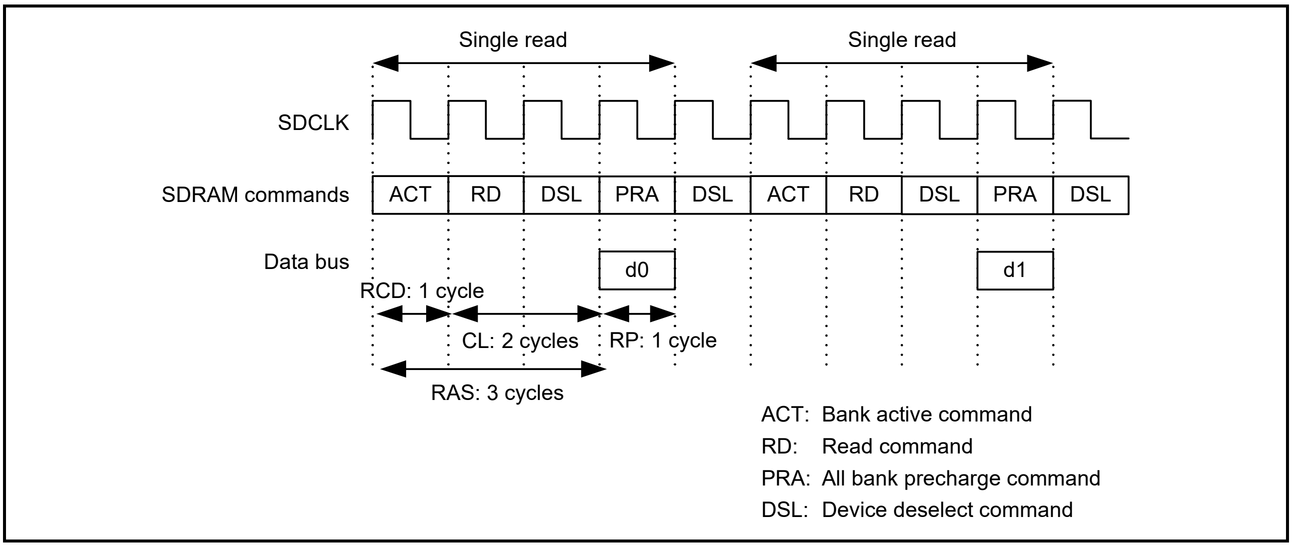


Figure 15.57 Example timing for single read (4)

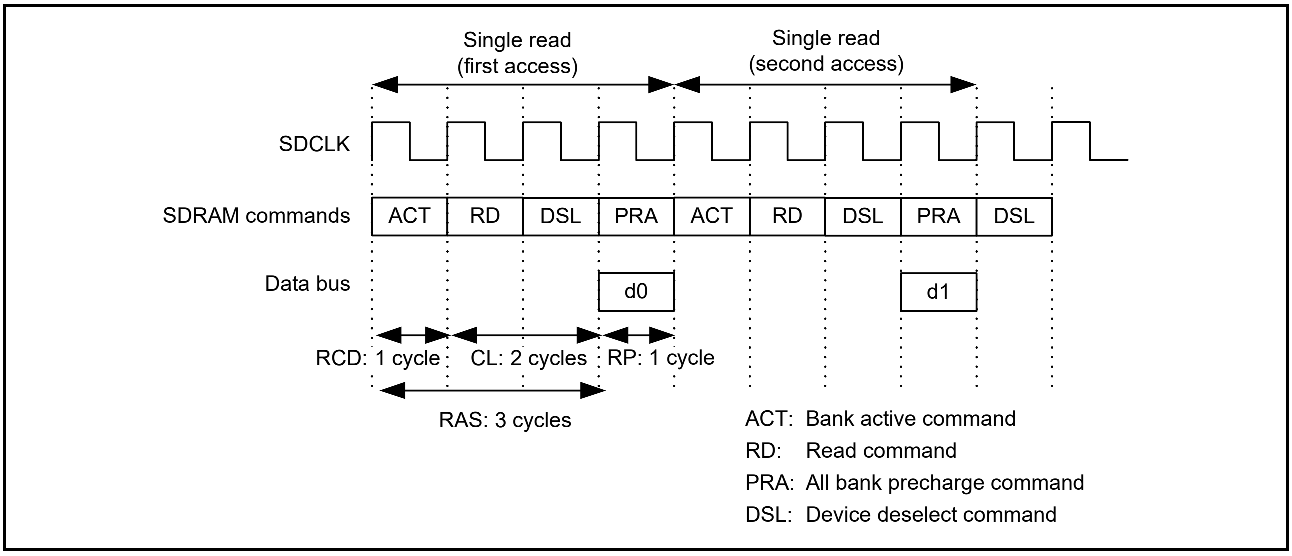


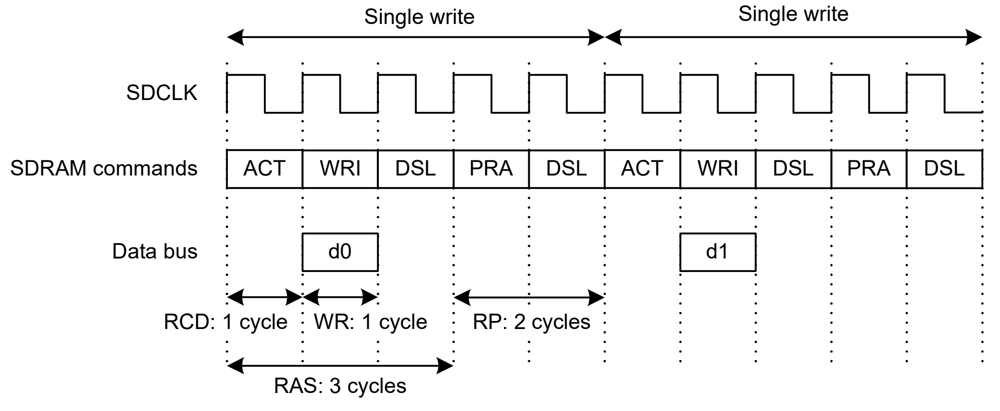
Figure 15.58 Example timing for single read (5), when two bus accesses occur for one transfer request

(2) Single write timing examples

Figure 15.59 to Figure 15.60 show the relationship between the single write timing and the SDTR register settings. Table 15.16 shows the association between the figures and the SDTR register settings. During write access, the next bus access is enabled at the earliest 2 cycles after an all bank precharge command (PRA) is issued. However, if two or more accesses occur for one transfer request, the next bus access is enabled at the earliest 1 cycle after the PRA is issued, as shown in Figure 15.63.

Table 15.16 Association between timing figures and STDR register settings for single write timing

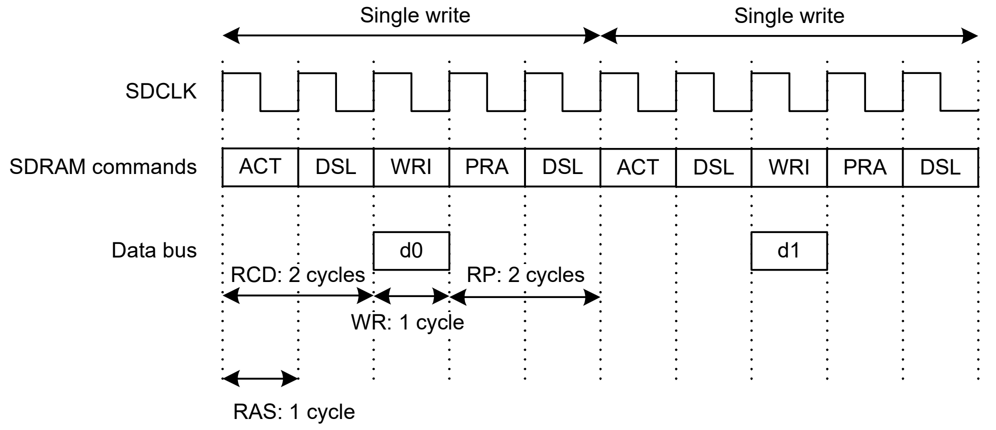
Figure number	RAS[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	WR settings	Number of cycles
Figure 15.59	010	3	00	1	001	2	0	1
Figure 15.60	000	1	01	2	001	2	0	1
Figure 15.61	000	1	01	2	001	2	1	2
Figure 15.62, Figure 15.63	010	3	00	0	000	2	0	1



Note: When the period specified in the RAS bits is longer than the period specified in WR bits, starting from the issue of the WRI command, the RAS setting is applied.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

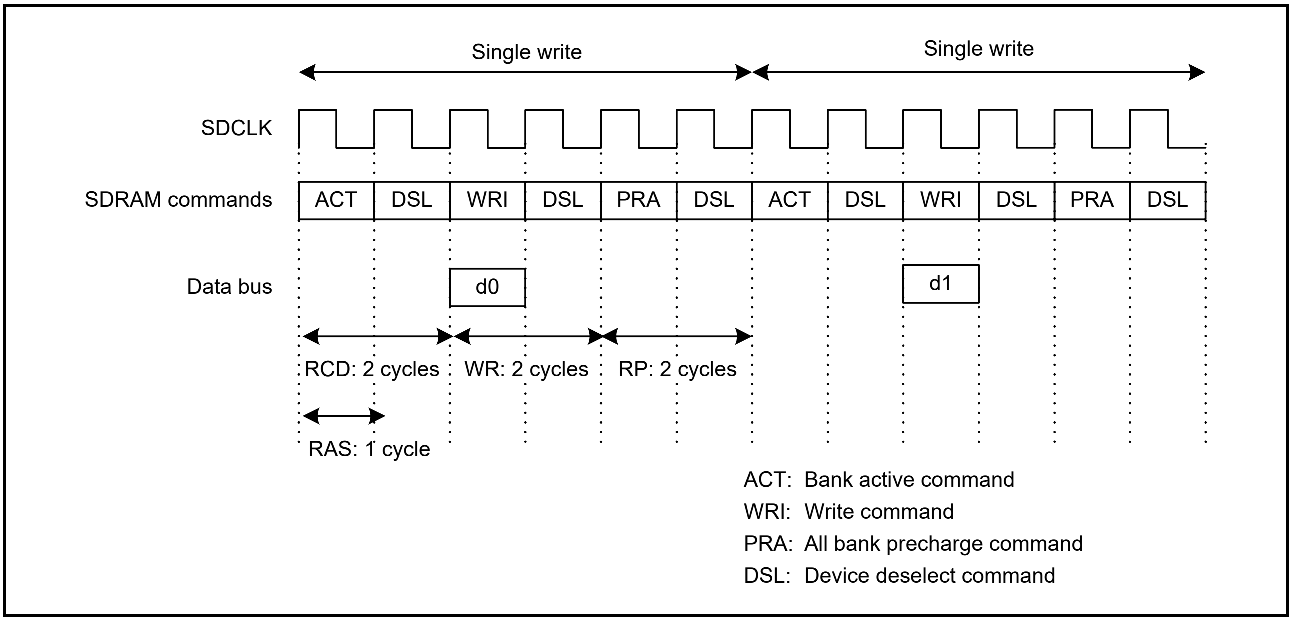
Figure 15.59 Example timing for single write (1)



Note: When the period specified in the RAS bits is shorter than the period specified in WR bits, starting from the issue of the WRI command, the WR setting is applied.

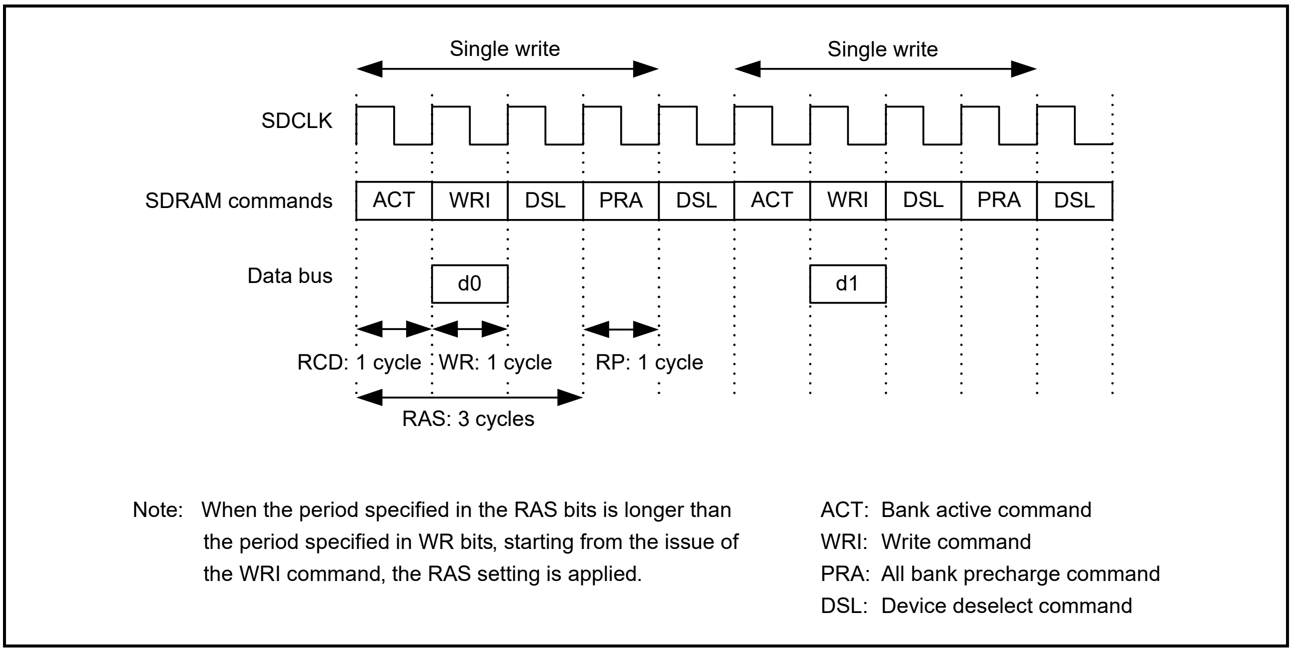
ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

Figure 15.60 Example timing for single write (2)



ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

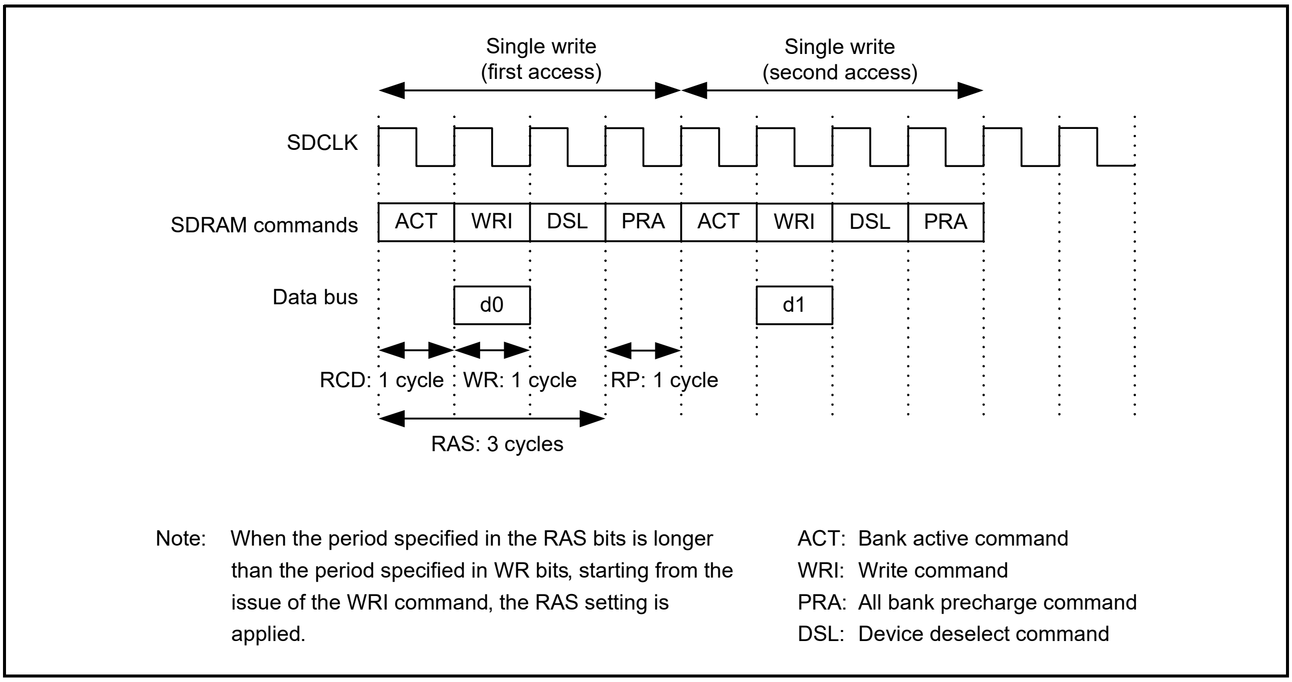
Figure 15.61 Example timing for single write (3)



Note: When the period specified in the RAS bits is longer than the period specified in WR bits, starting from the issue of the WRI command, the RAS setting is applied.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

Figure 15.62 Example timing for single write (4)



Note: When the period specified in the RAS bits is longer than the period specified in WR bits, starting from the issue of the WRI command, the RAS setting is applied.

ACT: Bank active command
WRI: Write command
PRA: All bank precharge command
DSL: Device deselect command

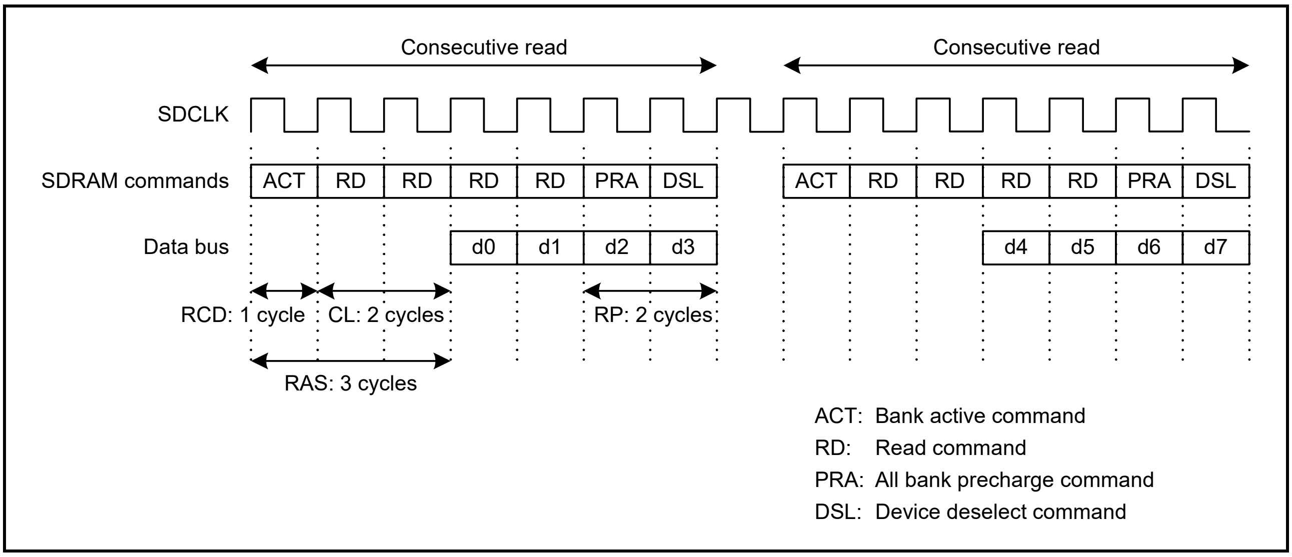
Figure 15.63 Example timing for single write (5), when two bus accesses occur for one transfer request

(3) Consecutive read timing examples

Figure 15.64 to Figure 15.66 show the relationship between the consecutive read timing for four data reads and the SDTR register settings. Table 15.17 shows the correspondence between the figures and the SDTR register settings.

Table 15.17 Correspondence between timing figures and STDR register settings for consecutive read timing

Figure number	RAS[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	CL[2:0] settings	Number of cycles
Figure 15.64	010	3	00	1	001	2	010	2
Figure 15.65	000	1	01	2	001	2	010	2
Figure 15.66	000	1	01	2	001	2	011	3



ACT: Bank active command
RD: Read command
PRA: All bank precharge command
DSL: Device deselect command

Figure 15.64 Example timing for consecutive read (1)

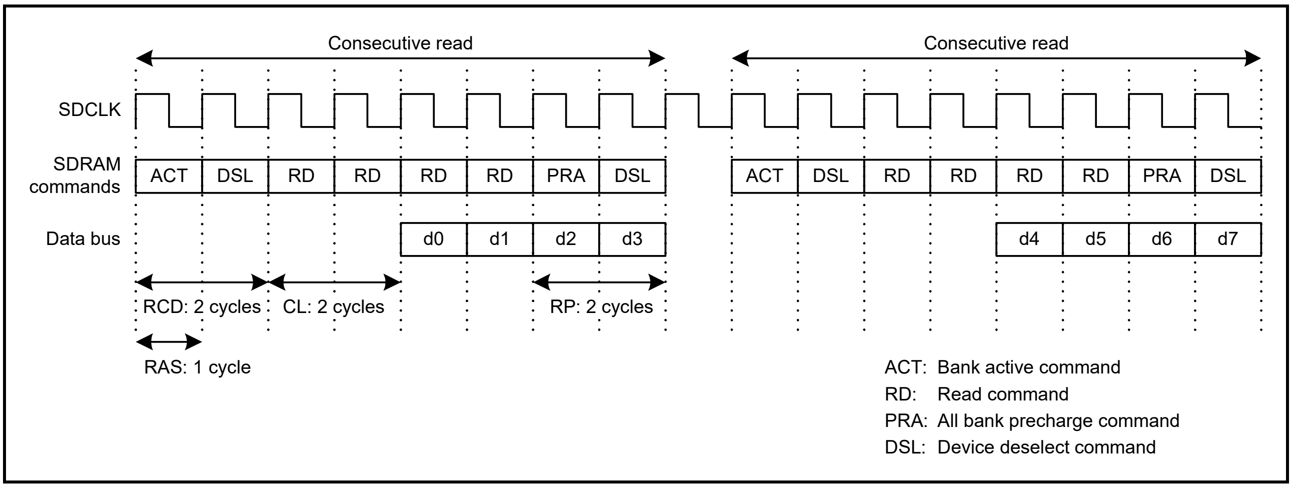


Figure 15.65 Example timing for consecutive read (2)

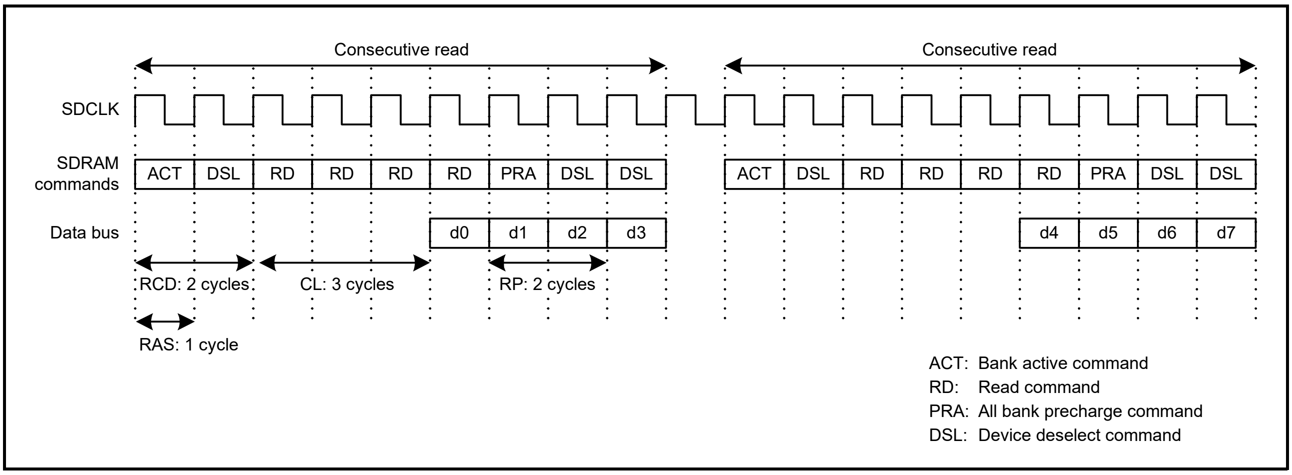


Figure 15.66 Example timing for consecutive read (3)

(4) Consecutive write timing examples

Figure 15.67 to Figure 15.69 show the relationship between the consecutive write timing for four data reads and the SDTR register settings. Table 15.18 shows the association between the figures and the SDTR register settings.

Table 15.18 Association between timing figures and STDR register settings for consecutive write timing

Figure number	RAS[2:0] settings	Number of cycles	RCD[1:0] settings	Number of cycles	RP[2:0] settings	Number of cycles	WR settings	Number of cycles
Figure 15.67	010	3	00	1	001	2	0	1
Figure 15.68	000	1	01	2	001	2	0	1
Figure 15.69	000	1	01	2	001	2	1	2

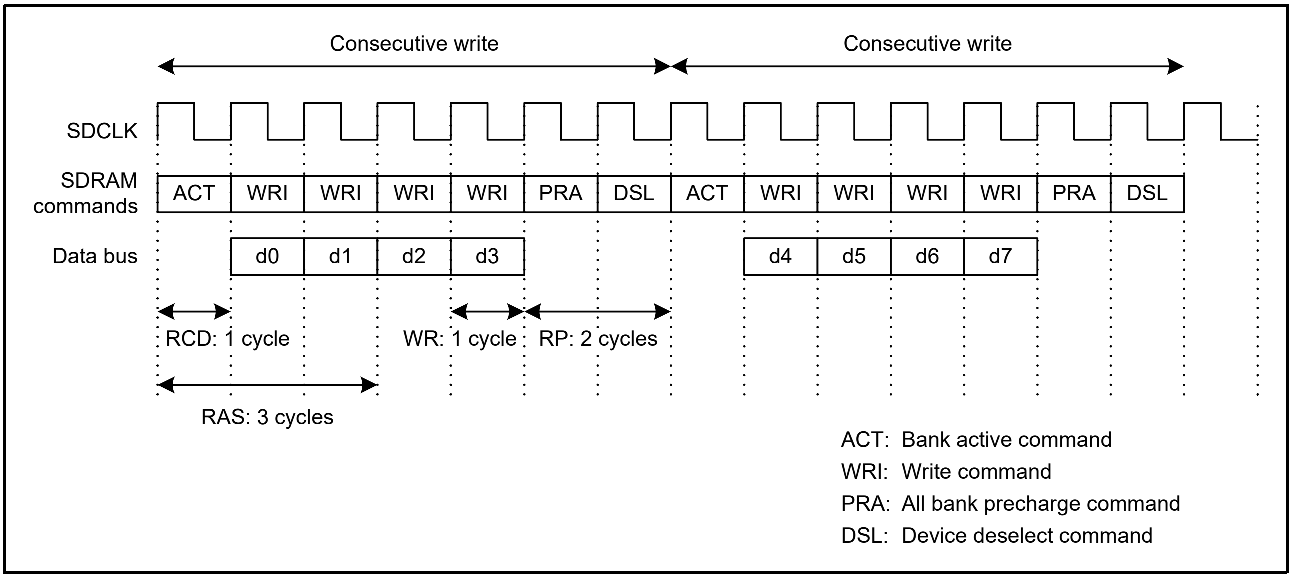


Figure 15.67 Example timing for consecutive write (1)

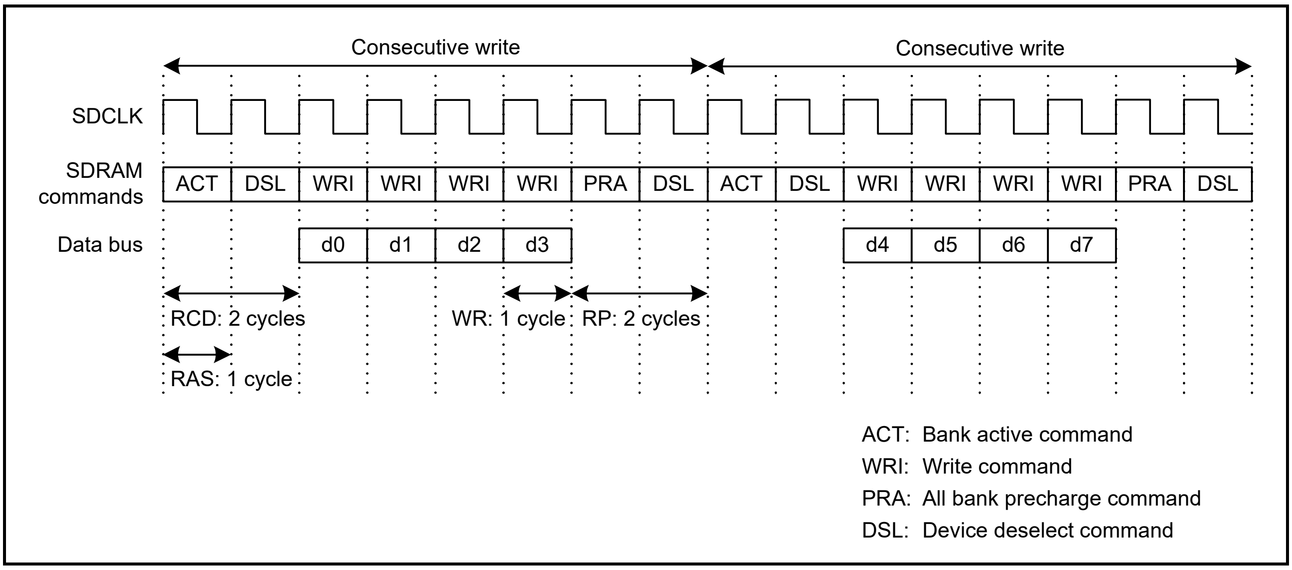


Figure 15.68 Example timing for consecutive write (2)

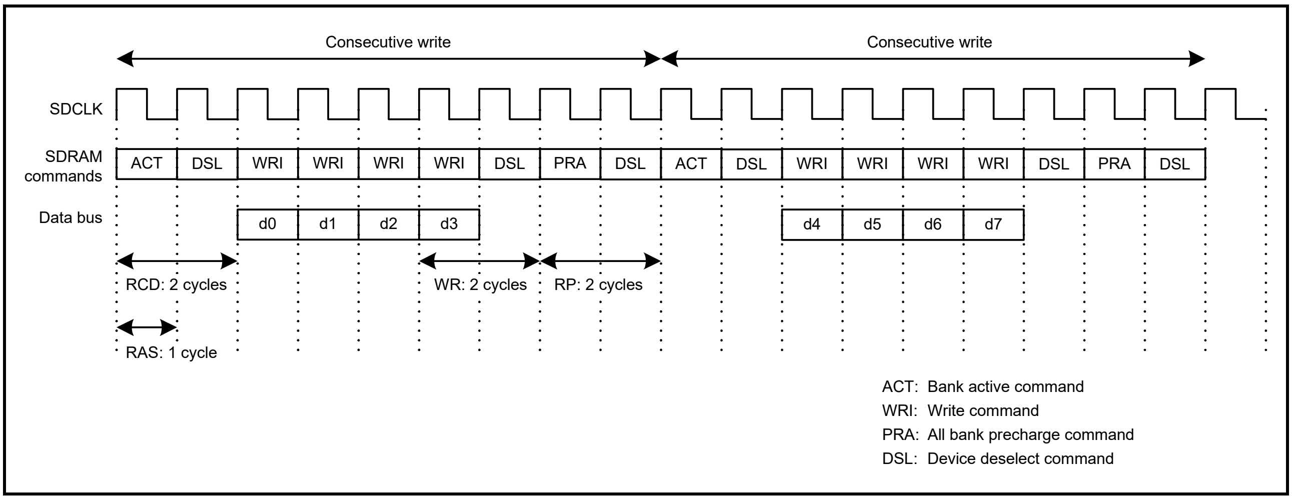


Figure 15.69 Example timing for consecutive write (3)

15.6.12 Address Multiplexing

In the SDRAM space, row and column addresses are multiplexed. The size of the shift in a row address must be specified in the Address Multiplex Select bits (SDADR.MXC[1:0]) in the SDRAM Address Register (SDADR). Additionally, in the SDRAM space, the address precharge-select command (Precharge-sel) is output to the upper bits of column addresses. Table 15.19 shows the relationship between the SDADR.MXC[1:0] settings and the shift amount.

Table 15.19 Address multiplexing

MXC [1:0]	Shift amount	Data bus width	Address	Address pins external to the MCU															
				A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
00	8 bits	8 bits	Row	A23	A22	A21	A20	A19	A18*	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A23	A22	A21	A20	A19	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A23	A22	A21	A20	A19*	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08
			Column	A23	A22	A21	A20	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
01	9 bits	8 bits	Row	A24	A23	A22	A21	A20	A20*	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A24	A23	A22	A21	A20	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09
			Column	A24	A23	A22	A21	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
10	10 bits	8 bits	Row	A25	A24	A23	A22	A21	A20*	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A25	A24	A23	A22	A21	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			Column	A25	A24	A23	A22	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
11	11 bits	8 bits	Row	A26	A25	A24	A23	A22	A21*	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	A26	A25	A24	A23	A10	P	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00
		16 bits	Row	A26	A25	A24	A23	A22*	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
			Column	A26	A25	A24	A11	P	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00

Note: P: Precharge-select command (Precharge-sel) is output.
*: When the PALL command is issued, Precharge-sel = 1 (high) is output. When the Active command is issued, the associated address is output.

15.6.13 Example SDRAM Connections

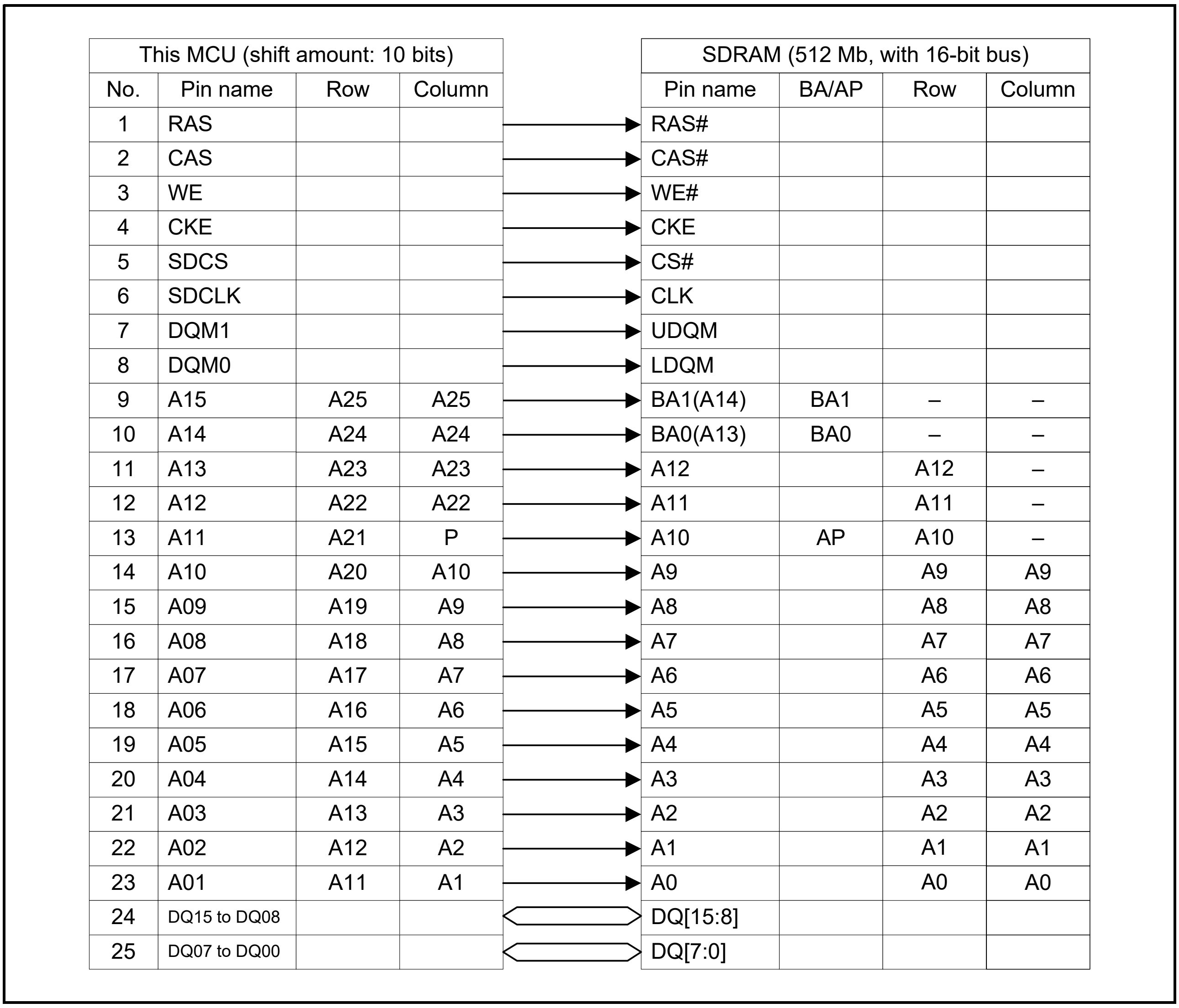
15.6.13.1 16-Bit bus space

Figure 15.70 shows an example connection to two 512-Mb SDRAMs with a 13-bit row address, 11-bit column address, and 8-bit bus.

This MCU (shift amount: 11 bits)					SDRAM (512 Mb, with 8-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM0			→	LDQM			
8	A15	A26	A26	→	BA1(A14)	BA1	–	–
9	A14	A25	A25	→	BA0(A13)	BA0	–	–
10	A13	A24	A24	→	A12		A12	–
11	A12	A23	A11	→	A11		A11	A11
12	A11	A22	P	→	A10	AP	A10	–
13	A10	A21	A10	→	A9		A9	A9
14	A09	A20	A9	→	A8		A8	A8
15	A08	A19	A8	→	A7		A7	A7
16	A07	A18	A7	→	A6		A6	A6
17	A06	A17	A6	→	A5		A5	A5
18	A05	A16	A5	→	A4		A4	A4
19	A04	A15	A4	→	A3		A3	A3
20	A03	A14	A3	→	A2		A2	A2
21	A02	A13	A2	→	A1		A1	A1
22	A01	A12	A1	→	A0		A0	A0
23	DQ07 to DQ00			⟷	DQ[7:0]			
No.					SDRAM (512 Mb, with 8-bit bus)			
(1)	RAS			→	RAS#			
(2)	CAS			→	CAS#			
(3)	WE			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS			→	CS#			
(6)	SDCLK			→	CLK			
24	DQM1			→	LDQM			
(8)	A15	A26	A26	→	BA1(A14)	BA1	–	–
(9)	A14	A25	A25	→	BA0(A13)	BA0	–	–
(10)	A13	A24	A24	→	A12		A12	–
(11)	A12	A23	A11	→	A11		A11	A11
(12)	A11	A22	P	→	A10	AP	A10	–
(13)	A10	A21	A10	→	A9		A9	A9
(14)	A09	A20	A9	→	A8		A8	A8
(15)	A08	A19	A8	→	A7		A7	A7
(16)	A07	A18	A7	→	A6		A6	A6
(17)	A06	A17	A6	→	A5		A5	A5
(18)	A05	A16	A5	→	A4		A4	A4
(19)	A04	A15	A4	→	A3		A3	A3
(20)	A03	A14	A3	→	A2		A2	A2
(21)	A02	A13	A2	→	A1		A1	A1
(22)	A01	A12	A1	→	A0		A0	A0
25	DQ15 to DQ08			⟷	DQ[7:0]			

Figure 15.70 SDRAM connection example with 512-Mb × 2 and 8-bit bus

Figure 15.71 shows an example connection to a 512-Mb SDRAMs with a 13-bit row address, 10-bit column address, and 16-bit bus.



This MCU (shift amount: 10 bits)					SDRAM (512 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A25	A25	→	BA1(A14)	BA1	–	–
10	A14	A24	A24	→	BA0(A13)	BA0	–	–
11	A13	A23	A23	→	A12		A12	–
12	A12	A22	A22	→	A11		A11	–
13	A11	A21	P	→	A10	AP	A10	–
14	A10	A20	A10	→	A9		A9	A9
15	A09	A19	A9	→	A8		A8	A8
16	A08	A18	A8	→	A7		A7	A7
17	A07	A17	A7	→	A6		A6	A6
18	A06	A16	A6	→	A5		A5	A5
19	A05	A15	A5	→	A4		A4	A4
20	A04	A14	A4	→	A3		A3	A3
21	A03	A13	A3	→	A2		A2	A2
22	A02	A12	A2	→	A1		A1	A1
23	A01	A11	A1	→	A0		A0	A0
24	DQ15 to DQ08			↔	DQ[15:8]			
25	DQ07 to DQ00			↔	DQ[7:0]			

Figure 15.71 SDRAM connection example with 512-Mb × 1 and 16-bit bus

Figure 15.72 shows an example connection to a 256-Mb SDRAMs with a 13-bit row address, 9-bit column address, and 16-bit bus.

This MCU (shift amount: 9 bits)					SDRAM (256 Mb, with 16-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS			→	RAS#			
2	CAS			→	CAS#			
3	WE			→	WE#			
4	CKE			→	CKE			
5	SDCS			→	CS#			
6	SDCLK			→	CLK			
7	DQM1			→	UDQM			
8	DQM0			→	LDQM			
9	A15	A24	A24	→	BA1(A14)	BA1	–	–
10	A14	A23	A23	→	BA0(A13)	BA0	–	–
11	A13	A22	A22	→	A12		A12	–
12	A12	A21	A21	→	A11		A11	–
13	A11	A20	P	→	A10	AP	A10	–
14	A10	A19	A10	→	A9		A9	–
15	A09	A18	A9	→	A8		A8	A8
16	A08	A17	A8	→	A7		A7	A7
17	A07	A16	A7	→	A6		A6	A6
18	A06	A15	A6	→	A5		A5	A5
19	A05	A14	A5	→	A4		A4	A4
20	A04	A13	A4	→	A3		A3	A3
21	A03	A12	A3	→	A2		A2	A2
22	A02	A11	A2	→	A1		A1	A1
23	A01	A10	A1	→	A0		A0	A0
24	DQ15 to DQ08			⟺	DQ[15:8]			
25	DQ07 to DQ00			⟺	DQ[7:0]			

Figure 15.72 SDRAM connection example with 256-Mb × 1 and 16-bit bus

15.6.14 Constraints

(1) Low-power states

In Software Standby and Deep Software Standby modes, auto-refresh operation is not available because the clock supply to the SDRAMC is stopped. To retain the data in the SDRAM when the SDRAM is externally connected, use the self-refresh function. For the procedure for transitioning to and recovering from self-refresh mode, see section 15.6.7, Self-Refresh.

(2) Setting the SDRAM Timing Register

Set the RAS[2:0] bits in the SDRAM Timing Register (SDTR) to a value less than or equal to the sum of the row column latency (SDTR.RCD[1:0]) and column latency (SDTR.CL[2:0]) settings. Operation is not guaranteed if this condition is not satisfied.

(3) Instruction code constraint

You must fix the instruction code to little-endian order.

15.7 Bus Error Monitoring Section

This monitoring system monitors each individual area, and whenever it detects an error, it returns the error to the requesting master IP using the AHB-Lite error response protocol.

15.7.1 Bus Error Types

The following types of errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- Bus slave MPU error
- Timeout.

Table 15.20 lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error. For more information on the bus master MPU and bus slave MPU, see section 16, Memory Protection Unit (MPU).

15.7.2 Operation When a Bus Error Occurs

When a bus error occurs, operation is not guaranteed and the error is returned to the requesting master IP. The bus errors that occur for each master are stored in the BUSnERRADD and BUSnERRSTAT registers. These registers must only be cleared by a reset. For more information, see section 15.3.21, Bus Error Address Register (BUSnERRADD) (n = 1 to 11) and section 15.3.22, Bus Error Status Register (BUSnERRSTAT) (n = 1 to 11).

Note: The DMAC and DTC do not receive bus errors. If the DMAC or DTC accesses the bus, the transfer continues. For other masters that receive bus errors, see the following sections:

- section 31, Ethernet DMA Controller (EDMAC)
- section 56, 2D Drawing Engine (DRW)
- section 57, JPEG Codec (JPEG)
- section 58, Graphics LCD Controller (GLCDC).

15.7.3 Conditions Leading to Illegal Address Access Errors

Table 15.20 lists the address spaces for each bus that trigger illegal address access errors.

Table 15.20 Conditions leading to illegal address access errors (1 of 2)

Address	Slave bus name	Master bus			
		CPU (ICode, DCode, System)	DMA	ETHER	GPX
0000 0000h to 01FF FFFFh	Memory bus 1 Memory bus 3	—	—	—	—
0200 0000h to 027F FFFFh	Memory mapping area	*1	E	E	E
0280 0000h to 1FFD FFFFh	Reserved	E	E	E	E
1FFE 0000h to 1FFF FFFFh	Memory bus 2 Memory bus 3	—	—	—	—
2000 0000h to 2003 FFFFh	Memory bus 4	—	—	—	—
2004 0000h to 200F FFFFh	Memory bus 5	—	—	—	—
2010 0000h to 3FFF FFFFh	Reserved	E	E	E	E
4000 0000h to 4001 FFFFh	Peripheral bus 1	—	—	E	E
4002 0000h to 4003 FFFFh	Reserved	E	E	E	E
4004 0000h to 4005 FFFFh	Peripheral bus 3	—	—	E	E
4006 0000h to 4007 FFFFh	Peripheral bus 4	—	—	E	E
4008 0000h to 4009 FFFFh	Peripheral bus 5	—	—	E	E
400A 0000h to 400B FFFFh	Reserved	—	—	E	E
400C 0000h to 400D FFFFh	Peripheral bus 7	—	—	E	E
400E 0000h to 400F FFFFh	Peripheral bus 8	—	—	E	E
4010 0000h to 407F FFFFh	Peripheral bus 9	—	—	—	E

Table 15.20 Conditions leading to illegal address access errors (2 of 2)

Address	Slave bus name	Master bus			
		CPU (ICode, DCode, System)	DMA	ETHER	GPX
4080 0000h to 5FFF FFFFh	Reserved	E	E	E	E
6000 0000h to 67FF FFFFh	QSPI area	—	—	—	—
6800 0000h to 7FFF FFFFh	Reserved	E	E	E	E
8000 0000h to 97FF FFFFh	CS/SDRAM area	—	—	—	—
9800 0000h to DFFF FFFFh	Reserved	E	E	E	E
E000 0000h to FFFF FFFFh	System for Cortex-M4	—	E	E	E

E: Path where an illegal address access error occurs

Note 1. The bus module does not detect whether the MMF switched the address. Therefore, if the MMF is enabled and the CPU accesses 0200 0000h, no error occurs. This depends on the switched address.
If the MMF is disabled and the CPU accesses 0200 0000h, the bus module can detect the error.

The bus module detects an access error resulting from access to a reserved area, for example if no area is assigned to the slave.

- 0200 0000h to 1FFD FFFFh: access error detection
- 0000 0000h to 01FF FFFFh: memory bus 1 no access error detection.

15.7.4 Timeout

For some peripheral modules, a timeout error occurs with the module-stop function. When there is no response from the slave for a certain period of time, a timeout error is detected. A timeout error is returned to the requesting master IP using the AHB-Lite error response protocol.

15.8 Notes on Using Flash Cache

When using flash cache through access from the CPU, the Arm® MPU should also be set to cacheable. See references 1. and 2. for more information.

15.9 References

1. *ARM®v7-M Architecture Reference Manual* (ARM DDI 0403D)
2. *ARM® Cortex®-M4 Devices Generic User Guide* (ARM DUI 0553A).

16. Memory Protection Unit (MPU)

16.1 Overview

The MCU provides four Memory Protection Units (MPUs) and a CPU stack pointer monitor function. Table 16.1 lists the MPU specifications and Table 16.2 shows the behavior on detection of each MPU error.

Table 16.1 MPU specifications

Classification	Module/Function	Description
Illegal memory access	Arm® Cortex®-M4 CPU	• Arm CPU has a default memory map. If the CPU makes an illegal access, an exception interrupt occurs. • MPU can change a default memory map.
	CPU stack pointer monitor	2 regions: • Main Stack Pointer (MSP) • Process Stack Pointer (PSP).
Memory protection	Arm MPU	Memory protection function for the CPU: • 8 MPU regions with subregions and background region.
	Bus master MPU	Memory protection function for each bus master except for the CPU: • Bus master MPU group A: 32 regions • Bus master MPU group B: 8 regions • Bus master MPU group C: 8 regions.
	Bus slave MPU	Memory protection function for each bus slave.
Security	Security MPU	Protects against non-secure program access to the following secure regions: • 2 regions (PC) • 4 regions (code flash, SRAM, two secure functions).

Table 16.2 Behavior on MPU error detection

MPU type	Notification type	Bus access on error detection	Storing of error access information
CPU stack pointer monitor	Reset or non-maskable interrupt	Don't care	Not stored
Arm MPU	Hard fault	• Does not correctly have write access • Does not correctly have read access.	Stored in the Cortex-M4 processor
Bus master MPU	Reset or non-maskable interrupt	• Write access to the protected region • Read access to the protected region.	Stored
Bus slave MPU	• Reset or non-maskable interrupt • Hard fault.	• Write access ignored • Read access read as 0.	Stored
Security MPU	Not notified	• Does not correctly have write access • Does not correctly have read access.	Do not hold

For information on error access for the Arm MPU, see section 16.7. For information on error access for other MPUs, see section 15.3.21, Bus Error Address Register (BUSnERRADD) (n = 1 to 11) and section 15.3.22, Bus Error Status Register (BUSnERRSTAT) (n = 1 to 11) in section 15, Buses.

16.2 CPU Stack Pointer Monitor

The CPU stack pointer monitor detects underflows and overflows of the stack pointer. Because the Arm CPU has two stack pointers, a Main Stack Pointer (MSP) and Process Stack Pointer (PSP), it supports two CPU stack pointer monitors. If a stack pointer underflow or overflow is detected, the CPU stack pointer monitor generates a reset or a non-maskable interrupt.

To enable the CPU stack pointer monitor, set the Stack Pointer Monitor Enable bit in the Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL) to 1.

Table 16.3 lists the specifications of the CPU stack pointer monitor, Figure 16.1 shows a block diagram, and Figure 16.2 shows the register setting flow.

Table 16.3 CPU stack pointer monitor specifications

Parameter	Description
Protected region	SRAM region
Number of regions	2 regions: • Main Stack Pointer • Process Stack Pointer.
Address specification for individual regions	Region start and end addresses configurable
Stack pointer monitor enable or disable setting for individual regions	Stack pointer monitor for individual regions can be enabled or disabled
Operation on error detection	Reset or non-maskable interrupts can be generated
Register protection	Registers can be protected from illegal writes

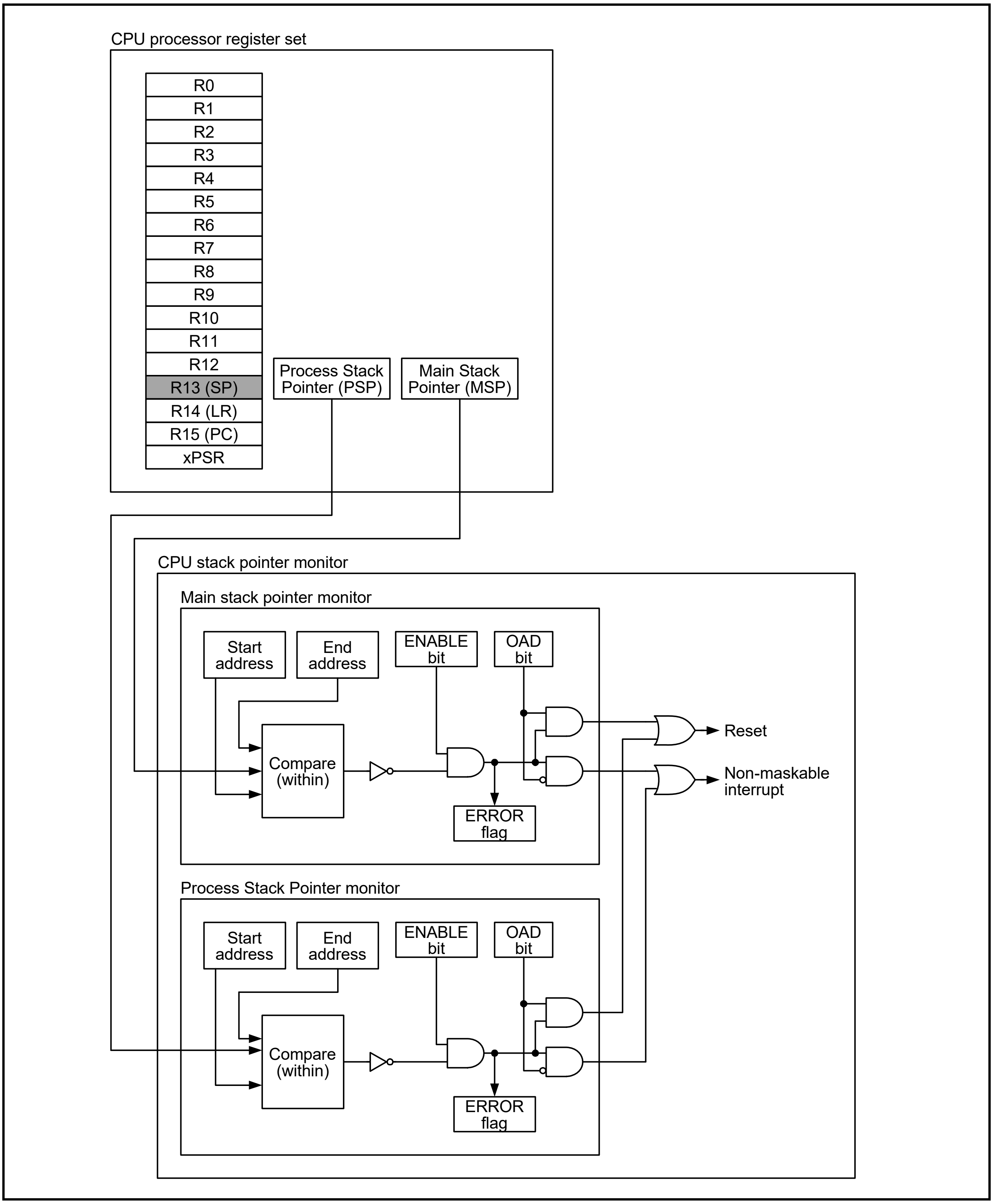


Figure 16.1 CPU stack pointer monitor block diagram

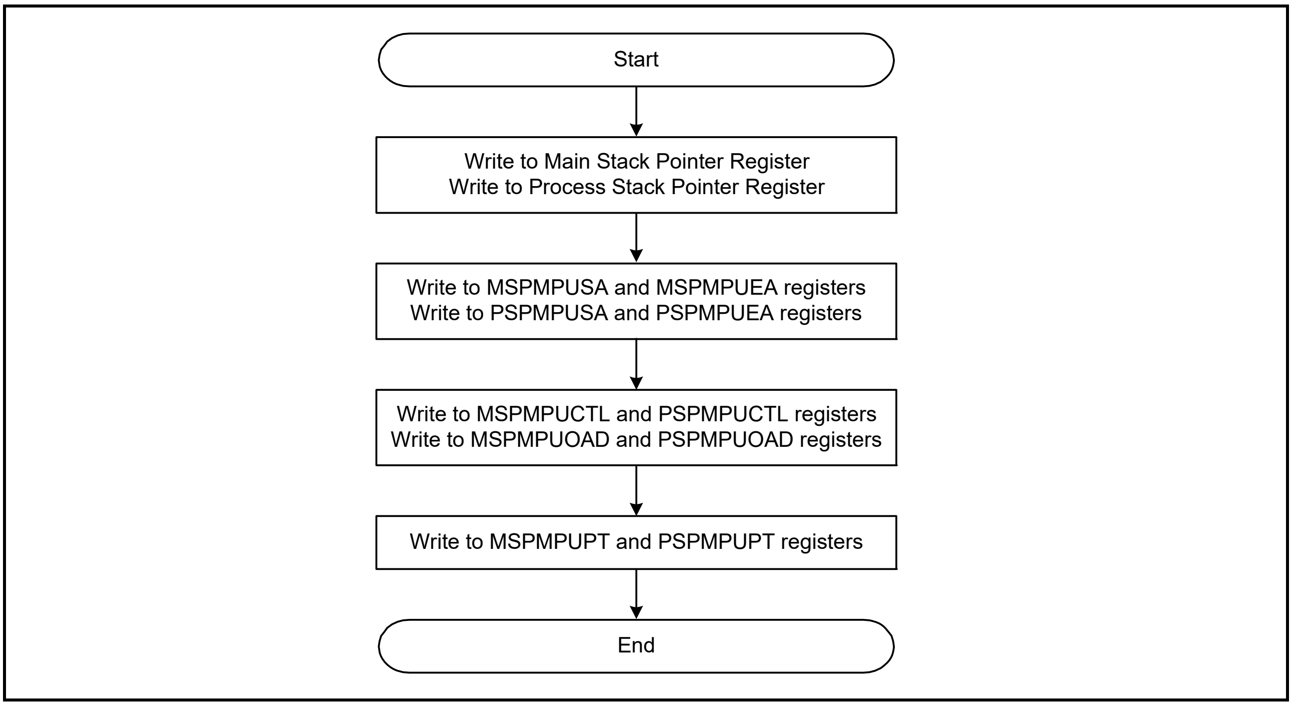


Figure 16.2 Register setting flow

16.2.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

16.2.1.1 Main Stack Pointer Monitor Start Address Register (MSPMPUSA)

Address(es): SPMON.MSPMPUSA 4000 0D08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSPMPUSA[31:16]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSPMPUSA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MSPMPUSA[31:0]	Region Start Address	Address where the region starts, for use in region determination. The lower 2 bits should be 0. The value range must be 1FF0 0000h to 200F FFFCh, excluding reserved areas.	R/W

The MSPMPUSA and MSPMPUEA registers specify the CPU stack region in the SRAM (1FF0 0000h to 200F FFFFh, excluding reserved areas). For SRAM area to be covered, see Figure 4.1, Memory map.

16.2.1.2 Main Stack Pointer Monitor End Address Register (MSPMPUEA)

Address(es): SPMON.MSPMPUEA 4000 0D0Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSPMPUEA[31:16]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSPMPUEA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MSPMPUEA[31:0]	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits should be 1. The value range must be 1FF0 0003h to 200F FFFFh, excluding reserved areas.	R/W

16.2.1.3 Process Stack Pointer Monitor Start Address Register (PSPMPUSA)

Address(es): SPMON.PSPMPUSA 4000 0D18h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PSPMPUSA[31:16]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSPMPUSA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	PSPMPUSA[31:0]	Region Start Address	Address where the region starts, for use in region determination. The lower 2 bits should be 0. The value range must be 1FF0 0000h to 200F FFFCh, excluding reserved areas.	R/W

The PSPMPUSA and PSPMPUEA registers specify the CPU stack region in the SRAM (1FF0 0000h to 200F FFFFh, excluding reserved areas). For SRAM area to be covered, see Figure 4.1, Memory map.

16.2.1.4 Process Stack Pointer Monitor End Address Register (PSPMPUEA)

Address(es): SPMON.PSPMPUEA 4000 0D1Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PSPMPUEA[31:16]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSPMPUEA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	PSPMPUEA[31:0]	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits should be 1. The value range must be 1FF0 0003h to 200F FFFFh, excluding reserved areas.	R/W

16.2.1.5 Stack Pointer Monitor Operation After Detection Register (MSPMPUOAD, PSPMPUOAD)

Address(es): SPMON.MSPMPUOAD 4000 0D00h, SPMON.PSPMPUOAD 4000 0D10h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OAD	Operation after Detection	0: Non-maskable interrupt 1: Reset.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits enable or disable writes to the OAD bit.	R/(W)*1

Note 1. Write data is not saved.

OAD bit (Operation after Detection)

The OAD bit selects either a reset or a non-maskable interrupt to occur when a stack pointer underflow or overflow is detected by the CPU stack pointer monitor. The main and the process stack pointer monitors each use an OAD bit to determine which signal is generated when a stack pointer underflow or overflow is detected. When writing to the OAD bit, write A5h simultaneously to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write A5h simultaneously to the KEY[7:0] bits. When values other than A5h are written to the KEY[7:0] bits, the OAD bit is not updated. The KEY[7:0] bits are always read as 00h.

16.2.1.6 Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL)

Address(es): SPMON.MSPMPUCTL 4000 0D04h, SPMON.PSPMPUCTL 4000 0D14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ERROR	—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0/1*1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ENABLE	Stack Pointer Monitor Enable	0: Disable stack pointer monitor 1: Enable stack pointer monitor.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ERROR	Stack Pointer Monitor Error Flag	0: No stack pointer overflow and underflow occurred 1: Stack pointer overflow or underflow occurred.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value depends on the reset generation source.

ENABLE bit (Stack Pointer Monitor Enable)

The ENABLE bit enables or disables the stack pointer monitor function, independently set for the main stack pointer monitor and the process stack pointer monitor.

When the MSPMPUCTL.ENABLE bit is set to 1, the following registers are available:

- MSPMPUSA
- MSPMPUEA
- MSPMPUOAD.

When the PSPMPUCTL.ENABLE bit is set to 1, the following registers are available:

- PSPMPUSA
- PSPMPUEA
- PSPMPUOAD.

ERROR bit (Stack Pointer Monitor Error Flag)

The ERROR bit indicates the status of the stack pointer monitor. Each stack pointer monitor has an independent ERROR bit. Only 0 can be written to this bit.

[Setting condition]

- Overflow or underflow of the stack pointer.

[Clearing conditions]

- 0 is written to this bit.
- A reset other than the bus master MPU error reset, bus slave MPU error reset, and stack pointer error reset.

Note: Only 0 can be written to the ERROR bit.

16.2.1.7 Stack Pointer Monitor Protection Register (MSPMPUPT, PSPMPUPT)

Address(es): SPMON.MSPMPUPT 4000 0D06h, SPMON.PSPMPUPT 4000 0D16h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PROTECT	Protection of Register	0: Stack pointer monitor register writes are permitted 1: Stack pointer monitor register writes are protected. Reads are permitted.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits enable or disable writes to the PROTECT bit.	R/(W)*1

Note 1. Write data is not saved.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected, independently set for the Main Stack Pointer monitor and the Process Stack Pointer monitor.

MSPMPUPT.PROTECT controls the following Main Stack Pointer protection registers:

- MSPMPUCTL
- MSPMPUSA
- MSPMPUEA.

PSPMPUT.PROTECT controls the following Process Stack Pointer protection registers:

- PSPMPUCTL
- PSPMPUSA
- PSPMPUEA.

When writing to the PROTECT bit, write A5h simultaneously to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write A5h simultaneously to the KEY[7:0] bits. When other values are written, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 00h.

16.2.2 Operation

16.2.2.1 Protecting the registers

To protect registers related to the CPU stack pointer monitor, set the associated PROTECT bit.

16.2.2.2 Overflow and underflow errors

The CPU stack pointer monitor generates an error if an overflow or underflow error is detected. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset. The non-maskable interrupt status is indicated in ICU.NMISR.SPEST, see section 14, Interrupt Controller Unit (ICU). Reset status is indicated in SYSTEM.RSTSR1.SPERF, see section 6, Resets.

When ICU.NMISR.SPEST indicates that a CPU stack pointer monitor interrupt occurred, confirm it by checking the ERROR bit in MSPMPUCTL and PSPMPUCTL to determine whether the error is a main stack pointer monitor error or process stack pointer monitor error.

A non-maskable interrupt remains set when a stack pointer overflows or underflows. To clear the error, clear the non-maskable interrupt flag by writing 1 to ICU.NMICLR.SPECLR. Write 0 to clear the ERROR bit in MSPMPUCTL and PSPMPUCTL.

16.3 Arm MPU

The Arm MPU has eight region MPUs and provides full support for:

- Protected regions
- Overlapping protected regions, with ascending priority:
  7 = highest priority
  0 = lowest priority.
- Access permissions
- Export of memory attributes to the system.

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (HardFault) handler. For details, see section 16.7.

16.4 Bus Master MPU

The bus master MPU monitors the addresses accessed by the bus masters in the entire address space (0000 0000h to FFFF FFFFh). The access control information, consisting of read and write permissions, can be independently set for up to 32 regions. The bus master MPU monitors access to each region based on these settings. If access to a protected region is detected, the bus master MPU generates a reset or a non-maskable interrupt. For details on error access, see section 15.3.21 and section 15.3.22 in section 15, Buses.

Table 16.4 lists the specifications of the bus master MPU and Figure 16.3 shows a block diagram. Figure 16.4 shows bus master MPU groups A, B, and C.

Table 16.4 Bus master MPU specifications

Parameter	Specifications
Protected master groups	• Bus master MPU group A: DMA bus • Bus master MPU group B: ETHER bus • Bus master MPU group C: GPX bus.
Protected region	0000 0000h to FFFF FFFFh
Number of regions	• Bus master MPU group A: 32 regions • Bus master MPU group B: 8 regions • Bus master MPU group C: 8 regions.
Address specification for individual regions	Region start and end addresses configurable
Enable/disable setting for memory protection in individual regions	Settings enabled or disabled for the associated region
Access-control settings for individual regions	Permission to read and write
Operation on error detection	Reset or non-maskable interrupt
Register protection	Register can be protected from illegal writes

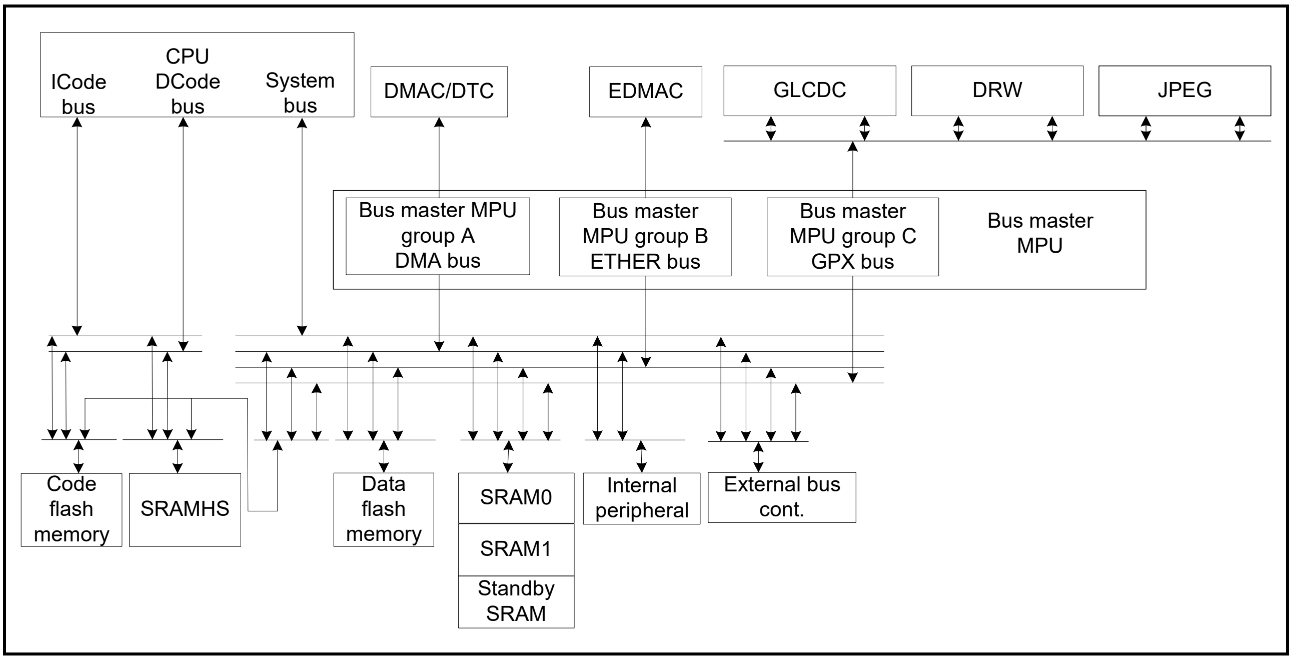


Figure 16.3 Bus master MPU block diagram

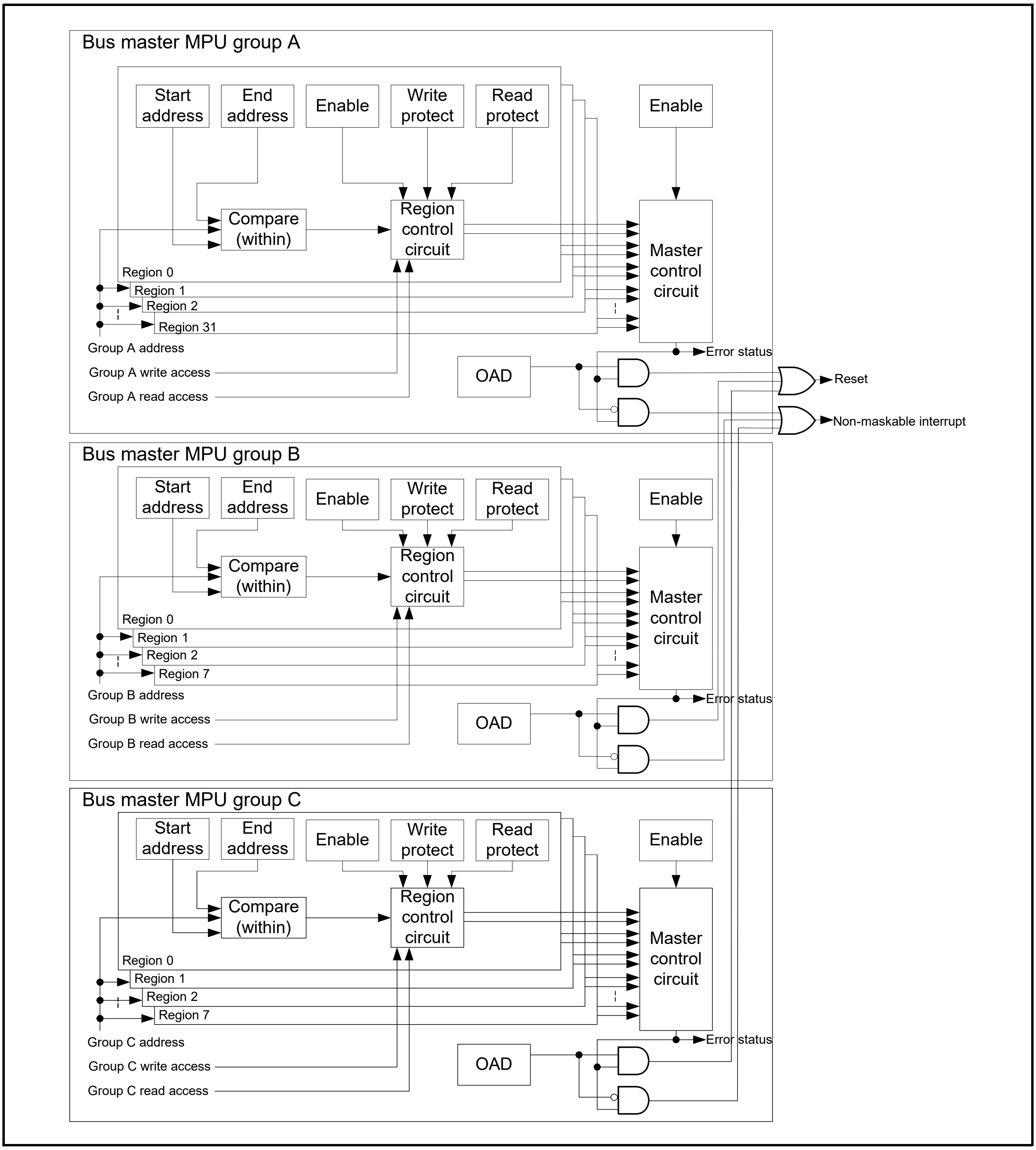


Figure 16.4 Bus master MPU groups A, B, and C

16.4.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

16.4.1.1 Group m Region n Start Address Register (MMPUSmn) (m = A to C; n = 0 to 31)

Address(es):
MMPU.MMPUSA0 4000 0204h, MMPU.MMPUSA1 4000 0214h, MMPU.MMPUSA2 4000 0224h, MMPU.MMPUSA3 4000 0234h,
MMPU.MMPUSA4 4000 0244h, MMPU.MMPUSA5 4000 0254h, MMPU.MMPUSA6 4000 0264h, MMPU.MMPUSA7 4000 0274h,
MMPU.MMPUSA8 4000 0284h, MMPU.MMPUSA9 4000 0294h, MMPU.MMPUSA10 4000 02A4h, MMPU.MMPUSA11 4000 02B4h,
MMPU.MMPUSA12 4000 02C4h, MMPU.MMPUSA13 4000 02D4h, MMPU.MMPUSA14 4000 02E4h, MMPU.MMPUSA15 4000 02F4h,
MMPU.MMPUSA16 4000 0304h, MMPU.MMPUSA17 4000 0314h, MMPU.MMPUSA18 4000 0324h, MMPU.MMPUSA19 4000 0334h,
MMPU.MMPUSA20 4000 0344h, MMPU.MMPUSA21 4000 0354h, MMPU.MMPUSA22 4000 0364h, MMPU.MMPUSA23 4000 0374h,
MMPU.MMPUSA24 4000 0384h, MMPU.MMPUSA25 4000 0394h, MMPU.MMPUSA26 4000 03A4h, MMPU.MMPUSA27 4000 03B4h,
MMPU.MMPUSA28 4000 03C4h, MMPU.MMPUSA29 4000 03D4h, MMPU.MMPUSA30 4000 03E4h, MMPU.MMPUSA31 4000 03F4h,
MMPU.MMPUSB0 4000 0604h, MMPU.MMPUSB1 4000 0614h, MMPU.MMPUSB2 4000 0624h, MMPU.MMPUSB3 4000 0634h,
MMPU.MMPUSB4 4000 0644h, MMPU.MMPUSB5 4000 0654h, MMPU.MMPUSB6 4000 0664h, MMPU.MMPUSB7 4000 0674h,
MMPU.MMPUSC0 4000 0A04h, MMPU.MMPUSC1 4000 0A14h, MMPU.MMPUSC2 4000 0A24h, MMPU.MMPUSC3 4000 0A34h,
MMPU.MMPUSC4 4000 0A44h, MMPU.MMPUSC5 4000 0A54h, MMPU.MMPUSC6 4000 0A64h, MMPU.MMPUSC7 4000 0A74h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MMPUSm[31:16](m = A to C)															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MMPUSm[15:0](m = A to C)															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MMPUSmn[31:0](m = A to C)	Region Start Address	Address where the region starts, for use in region determination. The lower 2 bits to 00b.	R/W

16.4.1.2 Group m Region n End Address Register (MMPUEmn) (m = A to C; n = 0 to 31)

Address(es):
MMPU.MMPUEA0 4000 0208h, MMPU.MMPUEA1 4000 0218h, MMPU.MMPUEA2 4000 0228h, MMPU.MMPUEA3 4000 0238h,
MMPU.MMPUEA4 4000 0248h, MMPU.MMPUEA5 4000 0258h, MMPU.MMPUEA6 4000 0268h, MMPU.MMPUEA7 4000 0278h,
MMPU.MMPUEA8 4000 0288h, MMPU.MMPUEA9 4000 0298h, MMPU.MMPUEA10 4000 02A8h, MMPU.MMPUEA11 4000 02B8h,
MMPU.MMPUEA12 4000 02C8h, MMPU.MMPUEA13 4000 02D8h, MMPU.MMPUEA14 4000 02E8h, MMPU.MMPUEA15 4000 02F8h,
MMPU.MMPUEA16 4000 0308h, MMPU.MMPUEA17 4000 0318h, MMPU.MMPUEA18 4000 0328h, MMPU.MMPUEA19 4000 0338h,
MMPU.MMPUEA20 4000 0348h, MMPU.MMPUEA21 4000 0358h, MMPU.MMPUEA22 4000 0368h, MMPU.MMPUEA23 4000 0378h,
MMPU.MMPUEA24 4000 0388h, MMPU.MMPUEA25 4000 0398h, MMPU.MMPUEA26 4000 03A8h, MMPU.MMPUEA27 4000 03B8h,
MMPU.MMPUEA28 4000 03C8h, MMPU.MMPUEA29 4000 03D8h, MMPU.MMPUEA30 4000 03E8h, MMPU.MMPUEA31 4000 03F8h,
MMPU.MMPUEB0 4000 0608h, MMPU.MMPUEB1 4000 0618h, MMPU.MMPUEB2 4000 0628h, MMPU.MMPUEB3 4000 0638h,
MMPU.MMPUEB4 4000 0648h, MMPU.MMPUEB5 4000 0658h, MMPU.MMPUEB6 4000 0668h, MMPU.MMPUEB7 4000 0678h,
MMPU.MMPUEC0 4000 0A08h, MMPU.MMPUEC1 4000 0A18h, MMPU.MMPUEC2 4000 0A28h, MMPU.MMPUEC3 4000 0A38h,
MMPU.MMPUEC4 4000 0A48h, MMPU.MMPUEC5 4000 0A58h, MMPU.MMPUEC6 4000 0A68h, MMPU.MMPUEC7 4000 0A78h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MMPUEm[31:16](m = A to C)															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MMPUEm[15:0](m = A to C)															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MMPUEm[31:0](m = A to C)	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits should be 1.	R/W

16.4.1.3 Group m Region n Access Control Register (MMPUACmn) (m = A to C; n = 0 to 31)

Address(es): MMPU.MMPUACA0 4000 0200h, MMPU.MMPUACA1 4000 0210h, MMPU.MMPUACA2 4000 0220h, MMPU.MMPUACA3 4000 0230h,
MMPU.MMPUACA4 4000 0240h, MMPU.MMPUACA5 4000 0250h, MMPU.MMPUACA6 4000 0260h, MMPU.MMPUACA7 4000 0270h,
MMPU.MMPUACA8 4000 0280h, MMPU.MMPUACA9 4000 0290h, MMPU.MMPUACA10 4000 02A0h,
MMPU.MMPUACA11 4000 02B0h, MMPU.MMPUACA12 4000 02C0h, MMPU.MMPUACA13 4000 02D0h,
MMPU.MMPUACA14 4000 02E0h, MMPU.MMPUACA15 4000 02F0h,
MMPU.MMPUACA16 4000 0300h, MMPU.MMPUACA17 4000 0310h, MMPU.MMPUACA18 4000 0320h,
MMPU.MMPUACA19 4000 0330h, MMPU.MMPUACA20 4000 0340h, MMPU.MMPUACA21 4000 0350h,
MMPU.MMPUACA22 4000 0360h, MMPU.MMPUACA23 4000 0370h, MMPU.MMPUACA24 4000 0380h,
MMPU.MMPUACA25 4000 0390h, MMPU.MMPUACA26 4000 03A0h, MMPU.MMPUACA27 4000 03B0h,
MMPU.MMPUACA28 4000 03C0h, MMPU.MMPUACA29 4000 03D0h, MMPU.MMPUACA30 4000 03E0h,
MMPU.MMPUACA31 4000 03F0h, MMPU.MMPUACB0 4000 0600h, MMPU.MMPUACB1 4000 0610h,
MMPU.MMPUACB2 4000 0620h, MMPU.MMPUACB3 4000 0630h, MMPU.MMPUACB4 4000 0640h,
MMPU.MMPUACB5 4000 0650h, MMPU.MMPUACB6 4000 0660h, MMPU.MMPUACB7 4000 0670h,
MMPU.MMPUACC0 4000 0A00h, MMPU.MMPUACC1 4000 0A10h, MMPU.MMPUACC2 4000 0A20h,
MMPU.MMPUACC3 4000 0A30h, MMPU.MMPUACC4 4000 0A40h, MMPU.MMPUACC5 4000 0A50h,
MMPU.MMPUACC6 4000 0A60h, MMPU.MMPUACC7 4000 0A70h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ENABLE	Region Enable	0: Group m region n unit disabled 1: Group m region n unit enabled.	R/W
b1	RP	Read Protection	0: Read access permitted 1: Read access protected.	R/W
b2	WP	Write Protection	0: Write access permitted 1: Write access protected.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Individually configurable ENABLE, RP, and WP bits are provided for each group m region n unit.

ENABLE bit (Region Enable)

The ENABLE bit enables or disables the group m region n unit. When the ENABLE bit is set to 1, the RP and WP bits can be set to permit or protect access to the region that is set in MMPUSmn and MMPUEmn. When the ENABLE bit is set to 0, no region is specified for group m region n access.

RP bit (Read Protection)

The RP bit enables or disables read protection for group m region n. The RP bit is available when the ENABLE bit is set to 1.

WP bit (Write Protection)

The WP bit enables or disables write protection for group m region n. The WP bit is available when the ENABLE bit is set to 1.

Table 16.5 Function of region control circuit (1 of 2)

MMPUACmn.ENABLE*1	MMPUACmn.RP*1	MMPUACmn.WP*1	Access	Region	Output of group m region n unit*1
0	—	—	Read	—	Outside of region
			Write	—	Outside of region

Table 16.5 Function of region control circuit (2 of 2)

MMPUACmn.ENABLE*1	MMPUACmn.RP*1	MMPUACmn.WP*1	Access	Region	Output of group m region n unit*1
1	0	0	Read	Inside	Permitted region
				Outside	Outside of region
			Write	Inside	Permitted region
				Outside	Outside of region
	0	1	Read	Inside	Permitted region
				Outside	Outside of region
			Write	Inside	Protected region
				Outside	Outside of region
	1	0	Read	Inside	Protected region
				Outside	Outside of region
			Write	Inside	Permitted region
				Outside	Outside of region
	1	1	Read	Inside	Protected region
				Outside	Outside of region
			Write	Inside	Protected region
				Outside	Outside of region

Note 1. m = A to C,
In the case of m = A: n = 0 to 31
In the case of m = B or C: n = 0 to 7.

Table 16.6 Function of master control circuit

MMPUCTLm.ENABLE*1	Output of group m region 0 unit*1	Output of group m region 1 unit*1	Output of group A Region 2 to 31 unit, Output of group B or C Region 2 to 7 unit	Function of group m*1
1	Protected region	Don't care	Don't care	Generate error
1	Don't care	Protected region	Don't care	Generate error
1	Don't care	Don't care	Protected region	Generate error
1	Outside of region	Outside of region	Outside of region	Generate error
Other case				No error

Note 1. m = A to C.

A master MPU error occurs on the following conditions:

- MMPUCTLm.ENABLE = 1, and output of one or more region n units is to a protected region
- MMPUCTLm.ENABLE = 1, and output of all region n units is outside of region.

Other cases are handled as permitted regions.

16.4.1.4 Bus Master MPU Control Register (MMPUCTLm) (m = A to C)

Address(es): MMPU.MMPUCTLA 4000 0000h, MMPU.MMPUCTLB 4000 0400h, MMPU.MMPUCTLC 4000 0800h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	OAD	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ENABLE	Master Group Enable	0: Master group m disabled 1: Master group m enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	OAD	Operation After Detection	0: Non-maskable interrupt 1: Reset.	R/W
b7 to b2	—	Reserved	These bits are read as 0.The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits enable or disable writes to the OAD and ENABLE bits.	R/(W)*1

Note 1. Write data is not saved.

ENABLE bit (Master Group Enable)

The ENABLE bit enables or disables the bus master MPU function for each master group. When this bit is set to 1, MMPUACmn is available. When this bit is set to 0, MMPUACmn is unavailable, including permission for all regions. The bus master MPU function of each master group uses the ENABLE bit. When writing to the ENABLE bit, write A5h to the KEY[7:0] bits simultaneously using halfword access.

OAD bit (Operation After Detection)

The OAD bit selects either a reset or a non-maskable interrupt to occur when access to the protected region is detected by the bus master MPU. The bus master MPU function for each master group uses its OAD bit independently. When writing to the OAD bit, write A5h to the KEY[7:0] bits simultaneously using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the ENABLE and OAD bits. When writing to the ENABLE and OAD bits, write A5h to the KEY[7:0] bits simultaneously. When other values are written, the ENABLE and OAD bits are not updated. The KEY[7:0] bits are always read as 00h.

16.4.1.5 Group m Protection of Register (MMPUPTm) (m = A to C)

Address(es): MMPU.MMPUPTA 4000 0102h, MMPU.MMPUPTB 4000 0502h, MMPU.MMPUPTC 4000 0902h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PROTECT	Protection of register	0: All bus master MPU group m register writes are permitted 1: All bus master MPU group m register writes are protected. Read is permitted.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits enable or disable writes to the PROTECT bit.	R/(W)*1

Note 1. Write data is not saved.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected. MMPUTm.PROTECT controls the following bus master MPU group m protection registers:

- MMPUSmn
- MMPUEmn
- MMPUACmn
- MMPUCTLm.

When writing to the PROTECT bit, write A5h to the KEY[7:0] bits simultaneously using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write A5h simultaneously to the KEY[7:0] bits. When other values are written, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 00h.

16.4.2 Operation

16.4.2.1 Memory protection

The bus master MPU monitors memory access using control settings made individually for the access control regions. If access to a protected region is detected, the bus master MPU generates a memory protection error.

The bus master MPU can be configured for up to 32 protection regions. Protected regions include those with overlapping permitted and protected regions, and those with two overlapping permitted regions.

The bus master MPU provides three groups: A, B, and C. The memory protection function checks the address of the bus for a unified master group, and all accesses by a master group are protected. The bus master MPU sets the permission for all of the regions after reset. Setting MMPUCTLm.ENABLE to 1 protects all of the regions. A permitted region is set up within the protected region for each region. If access to a protected region is detected, the bus master MPU generates an error.

Figure 16.5 shows the use case of a bus master MPU.

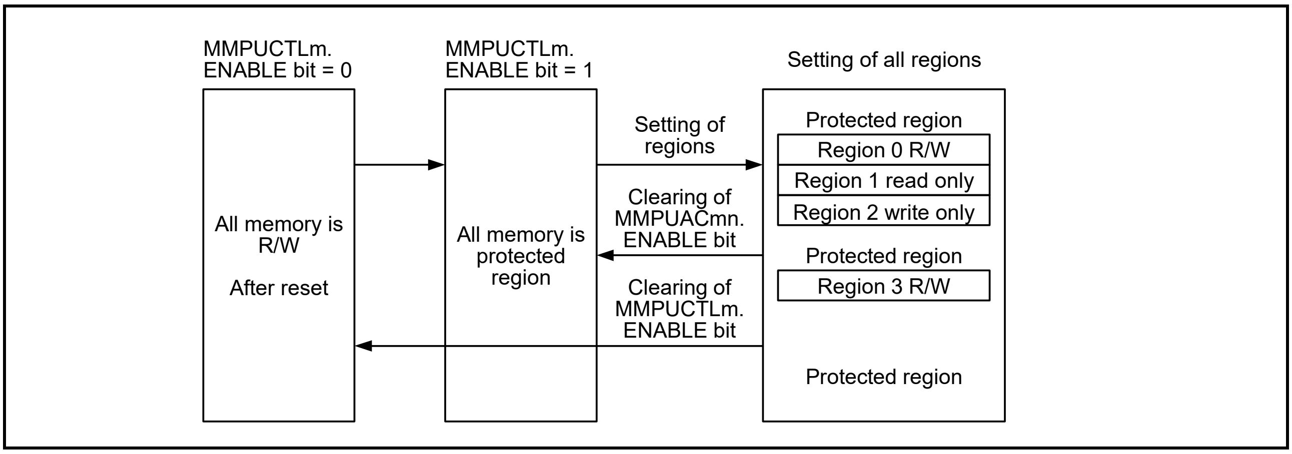


Figure 16.5 Use case of bus master MPU

Figure 16.6 shows the access permission or protection for overlapping bus master MPU regions. Access control for overlapping regions is as follows:

- The region is handled as protected when output of one or more region units is a protected region
- The region is handled as protected when output of all region units is outside of the regions
- Other cases are handled as permitted regions.

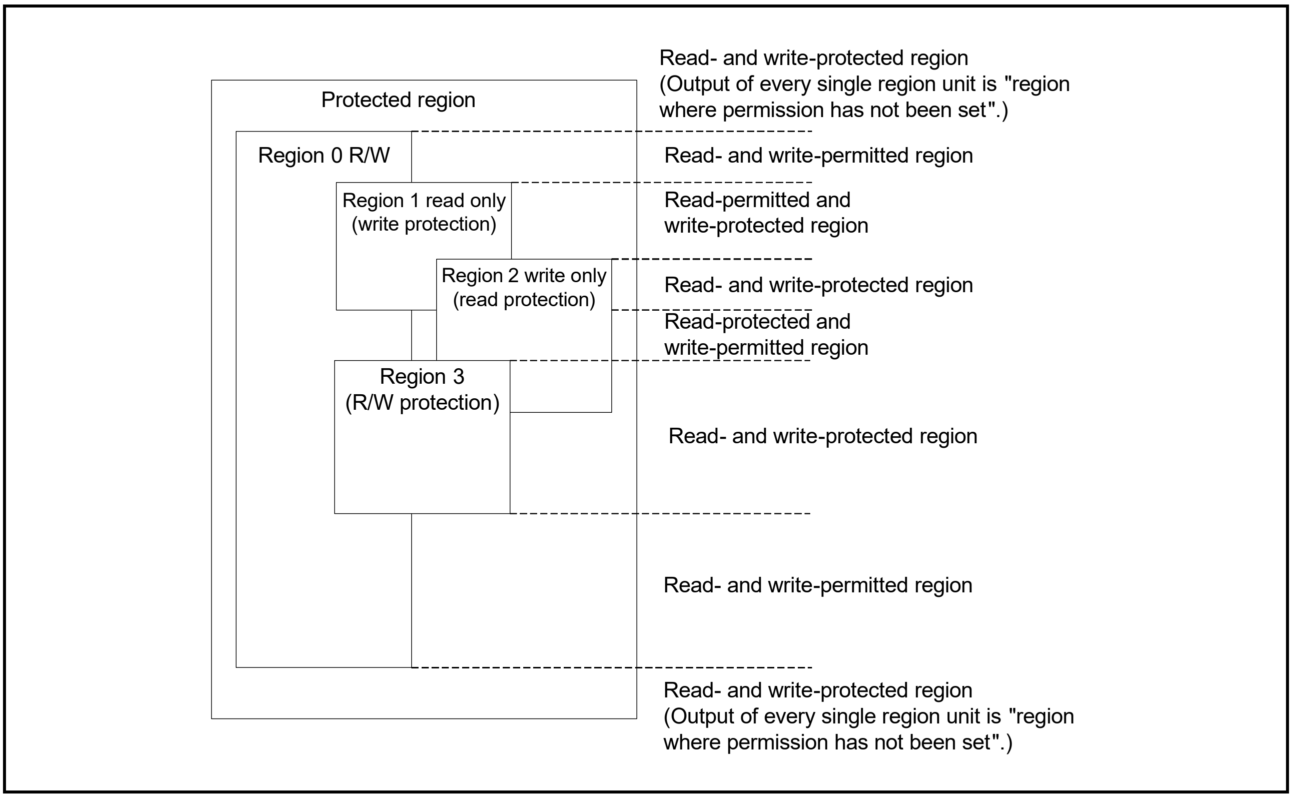


Figure 16.6 **Access permission or protection by overlap of the bus master MPU region**

Figure 16.7 shows the register setting flow after reset. During this register setting, stop all masters except the CPU.

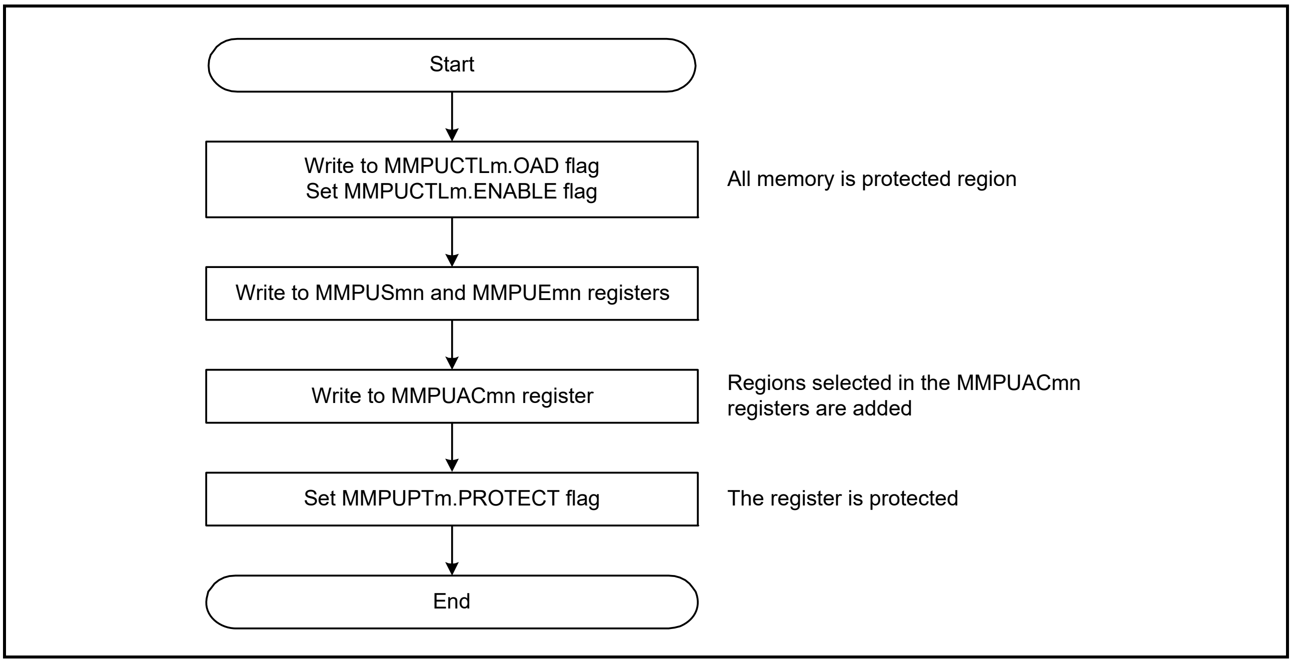


Figure 16.7 **Register setting flow after reset**

Figure 16.8 shows the register setting flow for adding regions. During this register setting, stop all masters except the CPU.

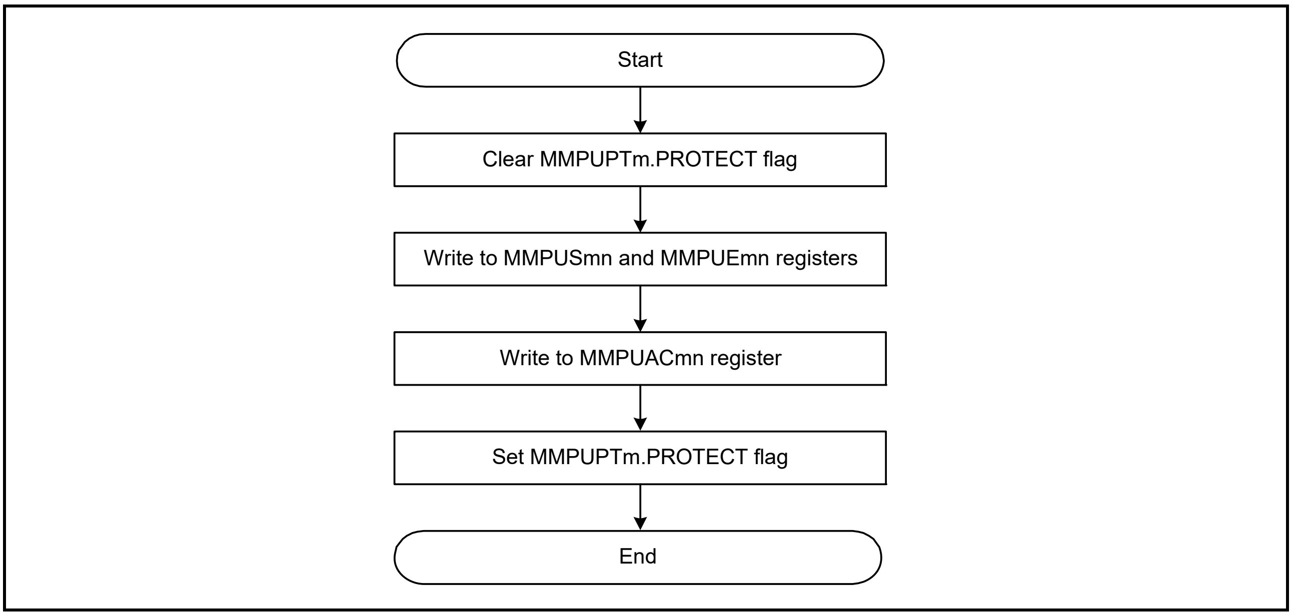


Figure 16.8 Register setting flow for region addition

16.4.2.2 Protecting the registers

To protect registers related to the bus master MPU, set the PROTECT bit in the associated MMPUPTm register.

16.4.2.3 Memory protection error

The bus master MPU generates an error if access to a protected region is detected. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset. The non-maskable interrupt or reset is shared between bus master MPU groups A, B, and C. The non-maskable interrupt status is indicated in ICU.NMISR.BUSMST, see section 14, Interrupt Controller Unit (ICU). Reset status is indicated in SYSTEM.RSTSR1.BUSMRF, see section 6, Resets.

16.5 Bus Slave MPU

The bus slave MPU monitors access to the bus slave functions, such as flash or SRAM. The function can be accessed from four bus masters, the CPU, and bus master MPU groups A, B, and C. The bus slave MPU has a separate protection register for each of the four bus masters, with independent access protection control, consisting of read and write permission. If access to a protected region is detected, the bus slave MPU generates a reset or a non-maskable interrupt, and can store the bus error address, bus error status, and error access status. For details, see 15.3.21 and 15.3.22 in section 15, Buses.

Table 16.7 lists the specifications of the bus slave MPU and Figure 16.9 shows a block diagram.

Table 16.7 Bus slave MPU specifications (1 of 2)

Specifications	Description
Protected bus masters	• Bus master MPU group A: DMA bus • Bus master MPU group B: ETHER bus • Bus master MPU group C: GPX bus.
Protected slave functions	• Memory bus 3: Code flash memory, SRAMHS • Internal peripheral bus 9: Flash memory (in P/E), data flash memory, and TSN • Memory bus 4: SRAM0 • Memory bus 5: SRAM1, Standby SRAM • Internal peripheral bus 1: DTC, DMAC, interrupt controller, flash registers, MPU, CSC, SDRAMC, SRAM registers, system controller and bus controller • Internal peripheral bus 3, 4, 5: Other peripherals • Internal peripheral bus 7: Secure IPs (SCE7) • Internal peripheral bus 8: Graphic IPs (JPEG/GLCDC/DRW) • EXBIU: External memory interface (SDRAM, CSC) • EXBIU2: External device interface (QSPI).

Table 16.7 Bus slave MPU specifications (2 of 2)

Specifications	Description
Access-control settings for individual regions	Permission to read and write
Operation on error detection	Reset, non-maskable interrupt, or exception
Register protection	Register can be protected from illegal writes

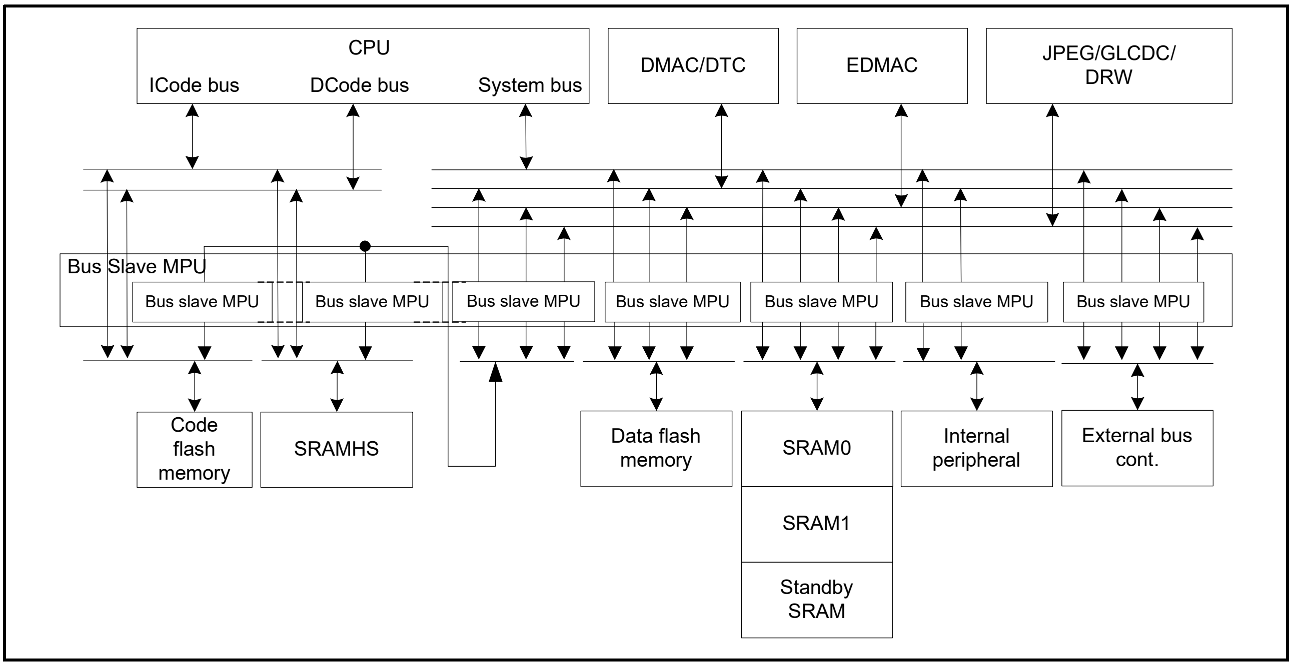


Figure 16.9 Bus slave MPU block diagram

16.5.1 Register Descriptions

Note: Bus access must be stopped before writing to the MPU registers.

16.5.1.1 Access Control Register for Memory Bus 3 (SMPUMBIU)

Address(es): SMPU.SMPUMBIU 4000 0C10h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WPSRAMHS	RPSRAMHS	WPFLI	RPFLI	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	0: Memory protection for master group B reads disabled 1: Memory protection for master group B reads enabled.	R/W
b5	WPGRPB	Master Group B Write Protection	0: Memory protection for master group B writes disabled 1: Memory protection for master group B writes enabled.	R/W
b6	RPGRPC	Master Group C Read Protection	0: Memory protection for master group C reads disabled 1: Memory protection for master group C reads enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b7	WPGRPC	Master Group C Write Protection	0: Memory protection for master group C writes disabled 1: Memory protection for master group C writes enabled.	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	RPFLI	Code Flash Memory Read Protection	0: Memory protection for code flash memory reads from master group A, B, and C disabled 1: Memory protection for code flash memory reads from master group A, B, and C enabled.	R/W
b13	WPFLI	Code Flash Memory Write Protection	1: Memory protection for code flash memory writes from master group A, B, and C enabled. This bit is read as 1. The write value should be 1.	R/W
b14	RPSRAMHS	SRAMHS Read Protection	0: Memory protection for SRAMHS reads from master group A, B, and C disabled 1: Memory protection for SRAMHS reads from master group A, B, and C enabled.	R/W
b15	WPSRAMHS	SRAMHS Write Protection	0: Memory protection for SRAMHS writes from master group A, B, and C disabled 1: Memory protection for SRAMHS writes from master group A, B, and C enabled.	R/W

The SMPUMBIU register enables memory protection for the specified master and slave for access from master group A, B, or C to code flash memory and SRAMHS.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A on memory bus 3.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A on memory bus 3.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables or disables memory protection for reads by master group B on memory bus 3.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables or disables memory protection for writes by master group B on memory bus 3.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables or disables memory protection for reads by master group C on memory bus 3.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables or disables memory protection for writes by master group C on memory bus 3.

RPFLI bit (Code Flash Memory Read Protection)

The RPFLI bit enables or disables memory protection for reads by master group A, B, or C on the code flash memory.

WPFLI bit (Code Flash Memory Write Protection)

The WPFLI bit enables memory protection for writes by master group A, B, or C on the code flash memory.

RPSRAMHS bit (SRAMHS Read Protection)

The RPSRAMHS bit enables or disables memory protection for reads by master group A, B, or C on the SRAMHS.

WPSRAMHS bit (SRAMHS Write Protection)

The WPSRAMHS bit enables or disables memory protection for writes by master group A, B, or C on the SRAMHS.

16.5.1.2 Access Control Register for Internal Peripheral Bus 9 (SMPUFBIU)

Address(es): SMPU.SMPUFBIU 4000 0C14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	0: Memory protection for master group B reads disabled 1: Memory protection for master group B reads enabled.	R/W
b5	WPGRPB	Master Group B Write Protection	0: Memory protection for master group B writes disabled 1: Memory protection for master group B writes enabled.	R/W
b6	RPGRPC	Master Group C Read Protection	1: Memory protection for master group C reads enabled. This bit is read as 1. The write value should be 1.	R/W
b7	WPGRPC	Master Group C Write Protection	1: Memory protection for master group C writes enabled. This bit is read as 1. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for reads by the CPU on internal peripheral bus 9.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for writes by the CPU on internal peripheral bus 9.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A on internal peripheral bus 9.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A on internal peripheral bus 9.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables or disables memory protection for reads by master group B on internal peripheral bus 9.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables or disables memory protection for writes by master group B on internal peripheral bus 9.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables memory protection for reads by master group C on internal peripheral bus 9. There is no connection between master group C and internal peripheral bus 9. This bit is read as 1, and the write value should be 1.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables memory protection for writes by master group C on internal peripheral bus 9. There is no connection between master group C and internal peripheral bus 9. This bit is read as 1, and the write value should be 1.

16.5.1.3 Access Control Register for Memory Bus 4 (SMPUSRAM0)

Address(es): SMPU.SMPUSRAM0 4000 0C18h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read protection	0: Memory protection for master group B reads disabled 1: Memory protection for master group B reads enabled.	R/W
b5	WPGRPB	Master Group B Write protection	0: Memory protection for master group B writes disabled 1: Memory protection for master group B writes enabled.	R/W
b6	RPGRPC	Master Group C Read protection	0: Memory protection for master group C reads disabled 1: Memory protection for master group C reads enabled.	R/W
b7	WPGRPC	Master Group C Write protection	0: Memory protection for master group C writes disabled 1: Memory protection for master group C writes enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read protection)

The RPCPU bit enables or disables memory protection for reads by the CPU on memory bus 4.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for writes by the CPU on memory bus 4.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for reads by master group A on memory bus 4.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for writes by master group A on memory bus 4.

RPGRPB bit (Master Group B Read protection)

The RPGRPB bit enables or disables memory protection for reads by master group B on memory bus 4.

WPGRPB bit (Master Group B Write protection)

The WPGRPB bit enables or disables memory protection for writes by master group B on memory bus 4.

RPGRPC bit (Master Group C Read protection)

The RPGRPC bit enables or disables memory protection for reads by master group C on memory bus 4.

WPGRPC bit (Master Group C Write protection)

The WPGRPC bit enables or disables memory protection for writes by master group C on memory bus 4.

16.5.1.4 Access Control Register for Memory Bus 5 (SMPUSRAM1)

Address(es): SMPU.SMPUSRAM1 4000 0C1Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	0: Memory protection for master group B reads disabled 1: Memory protection for master group B reads enabled.	R/W
b5	WPGRPB	Master Group B Write Protection	0: Memory protection for master group B writes disabled 1: Memory protection for master group B writes enabled.	R/W
b6	RPGRPC	Master Group C Read Protection	0: Memory protection for master group C reads disabled 1: Memory protection for master group C reads enabled.	R/W
b7	WPGRPC	Master Group C Write Protection	0: Memory protection for master group C writes disabled 1: Memory protection for master group C writes enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for reads by the CPU on memory bus 5.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for writes by the CPU on memory bus 5.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A on memory bus 5.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A on memory bus 5.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables or disables memory protection for reads by master group B on memory bus 5.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables or disables memory protection for writes by master group B on memory bus 5.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables or disables memory protection for reads by master group C on memory bus 5.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables or disables memory protection for writes by master group C on memory bus 5.

16.5.1.5 Access Control Register for Internal Peripheral Bus 1 (SMPUP0BIU)

Address(es): SMPU.SMPUP0BIU 4000 0C20h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	1: Memory protection for master group B reads enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b5	WPGRPB	Master Group B Write Protection	1: Memory protection for master group B writes enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b6	RPGRPC	Master Group C Read Protection	0: Memory protection for master group C reads disabled 1: Memory protection for master group C reads enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b7	WPGRPC	Master Group C Write Protection	0: Memory protection for master group C writes disabled 1: Memory protection for master group C writes enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for reads by the CPU on internal peripheral bus 1.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for writes by the CPU on internal peripheral bus 1.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A on internal peripheral bus 1.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A on internal peripheral bus 1.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables memory protection for reads by master group B on internal peripheral bus 1. There is no connection between master group B and internal peripheral bus 1. This bit is read as 1, and the write value should be 1.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables memory protection for writes by master group B on internal peripheral bus 1. There is no connection between master group B and internal peripheral bus 1. This bit is read as 1, and the write value should be 1.

RPGRPC bit (**Master Group C Read Protection**)

The RPGRPC bit enables memory protection for reads by master group C on internal peripheral bus 1. There is no connection between master group C and internal peripheral bus 1. This bit is read as 1, and the write value should be 1.

WPGRPC bit (**Master Group C Write Protection**)

The WPGRPC bit enables memory protection for writes by master group C on internal peripheral bus 1. There is no connection between master group C and internal peripheral bus 1. This bit is read as 1, and the write value should be 1.

16.5.1.6 Access Control Register for Internal Peripheral Bus 3 (SMPUP2BIU)

Address(es): SMPU.SMPUP2BIU 4000 0C24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	1: Memory protection for master group B reads enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b5	WPGRPB	Master Group B Write Protection	1: Memory protection for master group B writes enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b6	RPGRPC	Master Group C Read Protection	1: Memory protection for master group C reads enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b7	WPGRPC	Master Group C Write Protection	1: Memory protection for master group C writes enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (**CPU Read Protection**)

The RPCPU bit enables or disables memory protection for reads by the CPU on internal peripheral buses 3, 4, and 5.

WPCPU bit (**CPU Write Protection**)

The WPCPU bit enables or disables memory protection for writes by the CPU on internal peripheral buses 3, 4, and 5.

RPGRPA bit (**Master Group A Read Protection**)

The RPGRPA bit enables or disables memory protection for reads by master group A on internal peripheral buses 3, 4, and 5.

WPGRPA bit (**Master Group A Write Protection**)

The WPGRPA bit enables or disables memory protection for writes by master group A on internal peripheral buses 3, 4, and 5.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables memory protection for reads by master group B on internal peripheral buses 3, 4, and 5. There is no connection between master group B and internal peripheral buses 3, 4, and 5. This bit is read as 1, and the write value should be 1.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables memory protection for writes by master group B on internal peripheral buses 3, 4, and 5. There is no connection between master group B and internal peripheral buses 3, 4, and 5. This bit is read as 1, and the write value should be 1.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables memory protection for reads by master group C on internal peripheral buses 3, 4, and 5. There is no connection between master group C and internal peripheral buses 3, 4, and 5. This bit is read as 1, and the write value should be 1.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables memory protection for writes by master group C on internal peripheral buses 3, 4, and 5. There is no connection between master group C and internal peripheral buses 3, 4, and 5. This bit is read as 1, and the write value should be 1.

16.5.1.7 Access Control Register for Internal Peripheral Bus 7 (SMPUP6BIU)

Address(es): SMPU.SMPUP6BIU 4000 0C28h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	1: Memory protection for master group B reads enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b5	WPGRPB	Master Group B Write Protection	1: Memory protection for master group B writes enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b6	RPGRPC	Master Group C Read Protection	1: Memory protection for master group C reads enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b7	WPGRPC	Master Group C Write Protection	1: Memory protection for master group C writes enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for reads by the CPU on internal peripheral bus 7.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for writes by the CPU on internal peripheral bus 7.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A on internal peripheral bus 7.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A on internal peripheral bus 7.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables memory protection for reads by master group B on internal peripheral bus 7. There is no connection between master group B and internal peripheral bus 7. This bit is read as 1, and the write value should be 1.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables memory protection for writes by master group B on internal peripheral bus 7. There is no connection between master group B and internal peripheral bus 7. This bit is read as 1, and the write value should be 1.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables memory protection for reads by master group C on internal peripheral bus 7. There is no connection between master group C and internal peripheral bus 7. This bit is read as 1, and the write value should be 1.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables memory protection for writes by master group C on internal peripheral bus 7. There is no connection between master group C and internal peripheral bus 7. This bit is read as 1, and the write value should be 1.

16.5.1.8 Access Control Register for Internal Peripheral Bus 8 (SMPUP7BIU)

Address(es): SMPU.SMPUP7BIU 4000 0C2Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	1: Memory protection for master group B reads enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b5	WPGRPB	Master Group B Write Protection	1: Memory protection for master group B writes enabled. Master group B is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b6	RPGRPC	Master Group C Read Protection	1: Memory protection for master group C reads enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W
b7	WPGRPC	Master Group C Write Protection	1: Memory protection for master group C writes enabled. Master group C is protected, and is not detected. This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit name	Description	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for reads by the CPU on internal peripheral bus 8.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for writes by the CPU on internal peripheral bus 8.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A on internal peripheral bus 8.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A on internal peripheral bus 8.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables memory protection for reads by master group B on internal peripheral bus 8. There is no connection between master group B and internal peripheral bus 8. This bit is read as 1, and the write value should be 1.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables memory protection for writes by master group B on internal peripheral bus 8. There is no connection between master group B and internal peripheral bus 8. This bit is read as 1, and the write value should be 1.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables memory protection for reads by master group C on internal peripheral bus 8. There is no connection between master group C and internal peripheral bus 8. This bit is read as 1, and the write value should be 1.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables memory protection for writes by master group C on internal peripheral bus 8. There is no connection between master group C and internal peripheral bus 8. This bit is read as 1, and the write value should be 1.

16.5.1.9 Access Control Register for CS Area and SDRAM Area (SMPUEXBIU)

Address(es): SMPU.SMPUEXBIU 4000 0C30h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	0: Memory protection for master group B reads disabled 1: Memory protection for master group B reads enabled.	R/W
b5	WPGRPB	Master Group B Write Protection	0: Memory protection for master group B writes disabled 1: Memory protection for master group B writes enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	RPGRPC	Master Group C Read Protection	0: Memory protection for master group C reads disabled 1: Memory protection for master group C reads enabled.	R/W
b7	WPGRPC	Master Group C Write Protection	0: Memory protection for master group C writes disabled 1: Memory protection for master group C writes enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for reads by the CPU in the CS and SDRAM areas.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for writes by the CPU in the CS and SDRAM areas.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A in the CS and SDRAM areas.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A in the CS and SDRAM areas.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables or disables memory protection for reads by master group B in the CS and SDRAM areas.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables or disables memory protection for writes by master group B in the CS and SDRAM areas.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables or disables memory protection for reads by master group C in the CS and SDRAM areas.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables or disables memory protection for writes by master group C in the CS and SDRAM areas.

16.5.1.10 Access Control Register for QSPI Area (SMPUEXBIU2)

Address(es): SMPU.SMPUEXBIU2 4000 0C34h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	WPGRPC	RPGRPC	WPGRPB	RPGRPB	WPGRPA	RPGRPA	WPCPU	RPCPU
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPCPU	CPU Read Protection	0: Memory protection for CPU reads disabled 1: Memory protection for CPU reads enabled.	R/W
b1	WPCPU	CPU Write Protection	0: Memory protection for CPU writes disabled 1: Memory protection for CPU writes enabled.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Memory protection for master group A reads disabled 1: Memory protection for master group A reads enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Memory protection for master group A writes disabled 1: Memory protection for master group A writes enabled.	R/W
b4	RPGRPB	Master Group B Read Protection	0: Memory protection for master group B reads disabled 1: Memory protection for master group B reads enabled.	R/W
b5	WPGRPB	Master Group B Write Protection	0: Memory protection for master group B writes disabled 1: Memory protection for master group B writes enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	RPGRPC	Master Group C Read Protection	0: Memory protection for master group C reads disabled 1: Memory protection for master group C reads enabled.	R/W
b7	WPGRPC	Master Group C Write Protection	0: Memory protection for master group C writes disabled 1: Memory protection for master group C writes enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for reads by the CPU in the QSPI area.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for writes by the CPU in the QSPI area.

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for reads by master group A in the QSPI area.

WPGRPA bit (Master Group A Write Protection)

The WPGRPA bit enables or disables memory protection for writes by master group A in the QSPI area.

RPGRPB bit (Master Group B Read Protection)

The RPGRPB bit enables or disables memory protection for reads by master group B in the QSPI area.

WPGRPB bit (Master Group B Write Protection)

The WPGRPB bit enables or disables memory protection for writes by master group B in the QSPI area.

RPGRPC bit (Master Group C Read Protection)

The RPGRPC bit enables or disables memory protection for reads by master group C in the QSPI area.

WPGRPC bit (Master Group C Write Protection)

The WPGRPC bit enables or disables memory protection for writes by master group C in the QSPI area.

16.5.1.11 Slave MPU Control Register (SMPUCTL)

Address(es): SMPU.SMPUCTL 4000 0C00h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	PROTECT	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OAD	Operation After Detection	0: Non-maskable interrupt 1: Reset.	R/W
b1	PROTECT	Protection of Register	0: All bus slave MPU register writes are permitted 1: All bus slave MPU register writes are protected. Reads are permitted.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits enable or disable writes to the OAD and PROTECT bits.	R/(W)*1

Note 1. Write data is not saved.

OAD bit (Operation After Detection)

The OAD bit selects either a reset or non-maskable interrupt to occur when access to the protected region is detected by the bus slave MPU. When writing to the OAD bit, write A5h simultaneously to the KEY[7:0] bits using halfword access.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected. SMPUCTL.PROTECT controls the following registers:

- SMPUMBIU
- SMPUFBIU
- SMPUSRAM0
- SMPUSRAM1
- SMPUP0BIU
- SMPUP2BIU
- SMPUP6BIU
- SMPUP7BIU
- SMPUEXBIU
- SMPUEXBIU2.

When writing to the PROTECT bit, write A5h simultaneously to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD and PROTECT bits. When writing to the OAD and PROTECT bits, write A5h simultaneously to the KEY[7:0] bits. When other values are written, the OAD and PROTECT bits are not updated. The KEY[7:0] bits always read as 00h.

16.5.2 Operation

16.5.2.1 Memory protection

The bus slave MPU monitoring functions with access control information that is set for the individual access control registers. If access to a protected region is detected, the bus slave MPU generates a memory protection error.

The bus slave MPU is enabled by writing 1 to the Write Protect (WPCPU or WPGRPA) bit or the Read Protect (RPCPU or RPGRPA) bit in the access control register (SMPUMBIU, SMPUFBIU, SMPUSRAM0, SMPUSRAM1, SMPUP0BIU, SMPUP2BIU, SMPUP6BIU, SMPUP7BIU, SMPUEXBIU and SMPUEXBIU2).

16.5.2.2 Protecting the registers

To protect registers related to the bus slave MPU, set the PROTECT bit in the SMPUCTL register.

16.5.2.3 Memory protection error

The slave master MPU generates an error if access to a protected region is detected. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset. The non-maskable interrupt status is indicated in ICU.NMISR.BUSSST. (See section 14, Interrupt Controller Unit (ICU).) Reset status is indicated in SYSTEM.RSTSR1.BUSSRF. (See section 6, Resets.)

16.6 Security MPU

The MCU incorporates a security MPU with four secure regions that include the code flash, the SRAM, and two security functions. The secure regions can be protected from non-secure program accesses. Access to a protected region from a non-secure program is not permitted.

Table 16.8 lists the specifications of the security MPU and Figure 16.10 shows a block diagram of the security MPU.

Table 16.8 Security MPU specifications

Specifications	Description
Secure regions	Code flash, SRAM, two security function

Table 16.8 Security MPU specifications

Specifications	Description
Protected regions	0000 0000h to FFFF FFFFh
Number of regions	Program Counter: 2 regions Data access: 4 regions
Address specification for individual regions	Setting the address where regions start and end
Enable/disable setting for memory protection in individual regions	Settings enabled or disabled for the associated region

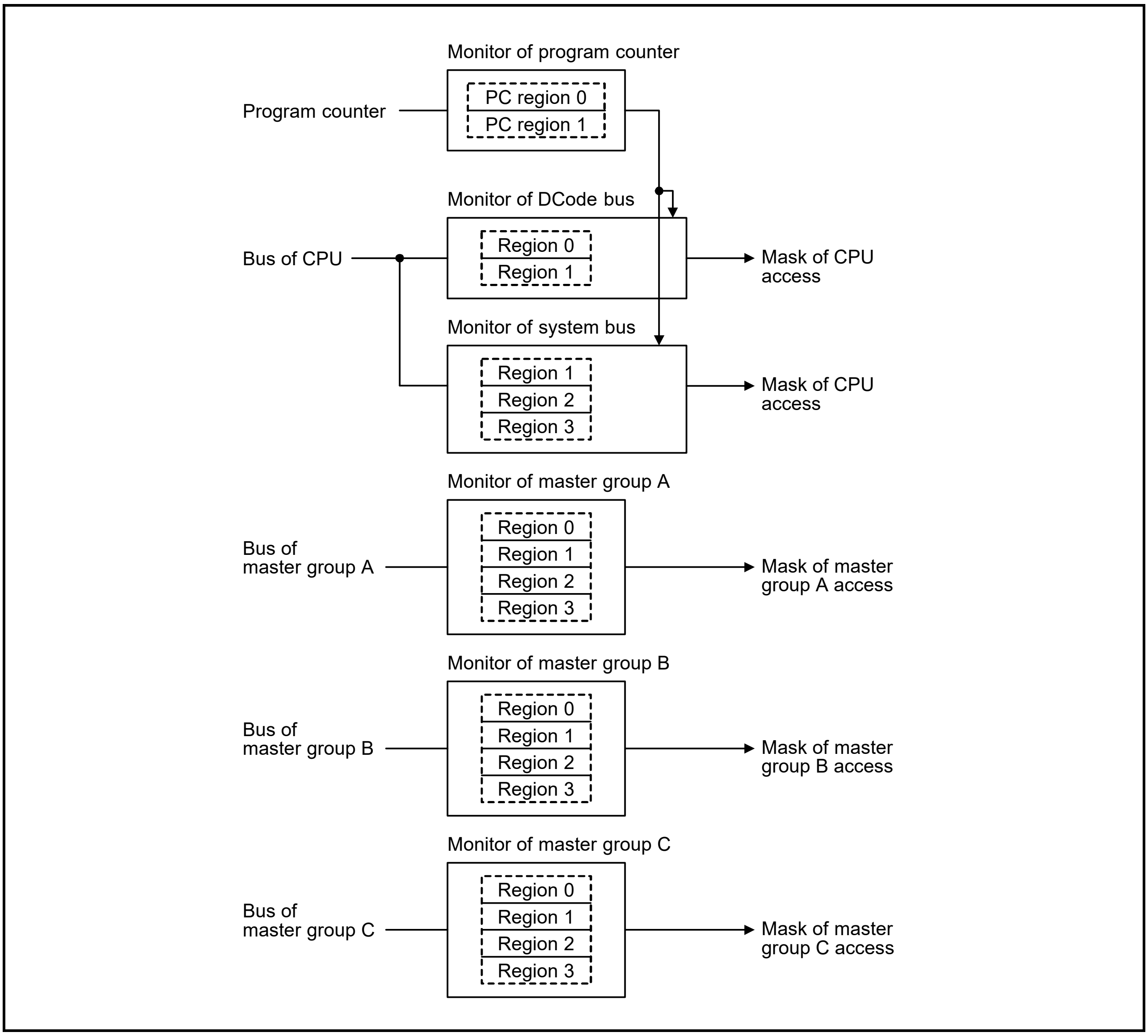


Figure 16.10 Security MPU block diagram

16.6.1 Register Descriptions (Option-Setting memory)

All security MPU registers are option-setting memory. Option-setting memory refers to a set of registers that are provided for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the flash.

16.6.1.1 Security MPU Program Counter Start Address Register (SECMPUPCSn) (n = 0, 1)

Address(es): SECMPUPCS0 0000 0408h, SECMPUPCS1 0000 0410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SECMPUPCS[31:16]															
Value after reset:	The value set by user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUPCS[15:2]														—	—
Value after reset:	The value set by user														0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b31 to b2	SECMPUPCS[31:2]	Region Start Address	Address where the region starts, for use in region determination.	R

The SECMPUPCSn and SECMPUPCEn registers specify the security fetch region of the code flash or SRAM (0000 0000h to FFFF FFFFh). The secure program is executed in the memory space defined by the SECMPUPCSn and SECMPUPCEn registers and can access the secure data specified in the SECMPUSm and SECMPUEm registers (m = 0 to 3).

The SECMPUPCSn register specifies the start address where the region starts. Setting of the memory mirror space (0200 0000h to 027F FFFFh) for MMF is prohibited.

An address space of greater than 12 bytes is required between the last instruction of the non-secure program and the first instruction of the secure program.

16.6.1.2 Security MPU Program Counter End Address Register (SECMPUPCEn) (n = 0, 1)

Address(es): SECMPUPCE0 0000 040Ch, SECMPUPCE1 0000 0414h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SECMPUPCE[31:16]															
Value after reset:	The value set by user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUPCE[15:2]														—	—
Value after reset:	The value set by user														1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. When writing to flash, the write value should always be 1.	R
b31 to b2	SECMPUPCE[31:2]	Region End Address	Address where the region ends, for use in region determination.	R

The SECMPUPCSn and SECMPUPCEn registers specify the security fetch region of code flash or SRAM (0000 0000h to FFFF FFFFh).

The SECMPUPCEn register specifies the end address where the region ends.

16.6.1.3 Security MPU Region 0 Start Address Register (SECMPUS0)

Address(es): SECMPUS0 0000 0418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SECMPUS0[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	The value set by user							

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUS0[15:2]														—	—
Value after reset:	The value set by user														0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b23 to b2	SECMPUS0[23:2]	Region Start Address	Address where the region starts, for use in region determination.	R
b31 to b24	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R

The SECMPUS0 and SECMPUE0 registers specify the secure region of flash (0000 0000 to 00FF FFFFh), which can be accessed only from the secure program set up by SECMPUPCSn and SECMPUPCEn.

The SECMPUS0 register specifies the start address where the region starts. Setting of the vector table area is prohibited.

16.6.1.4 Security MPU Region 0 End Address Register (SECMPUE0)

Address(es): SECMPUE0 0000 041Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SECMPUS0[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	The value set by user							

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUS0[15:2]														—	—
Value after reset:	The value set by user														1	1

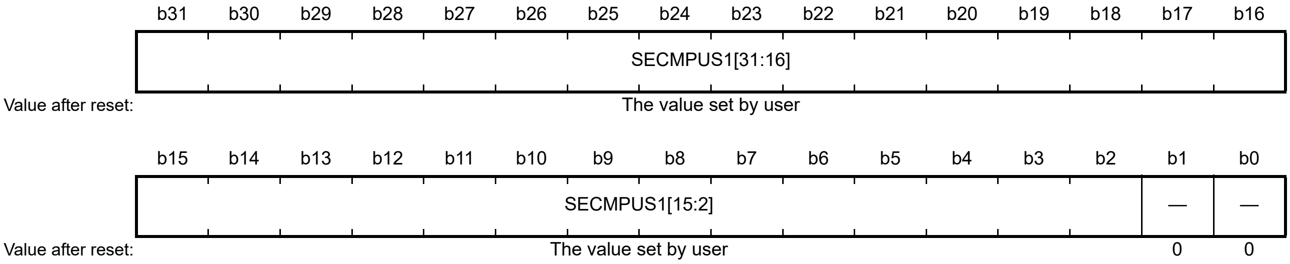
Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. When writing to flash, the write value should always be 1.	R
b23 to b2	SECMPUE0[23:2]	Region End Address	Address where the region end, for use in region determination.	R
b31 to b24	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R

The SECMPUS0 and SECMPUE0 registers specify the secure region of flash (0000 0000 to 00FF FFFFh). The memory space defined in the SECMPUS0 and SECMPUE0 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

The SECMPUE0 register specifies the end address where the region ends.

16.6.1.5 Security MPU Region 1 Start Address Register (SECMPUS1)

Address(es): SECMPUS1 0000 0420h



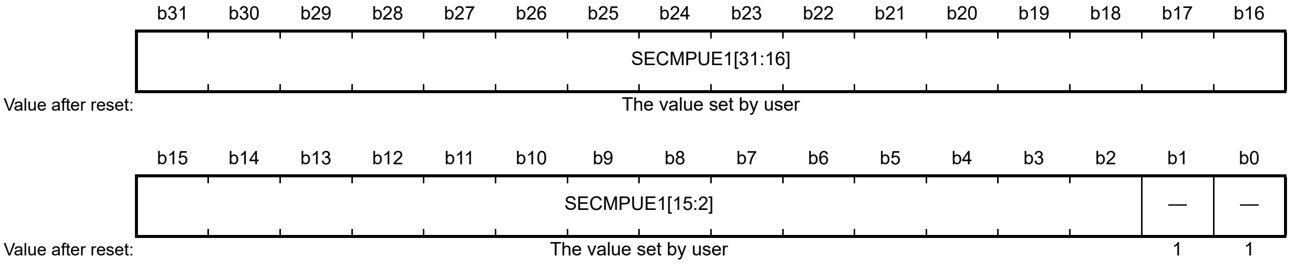
Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b19 to b2	SECMPUS1[19:2]	Region Start Address	Address where the region starts, for use in region determination.	R
b31 to b20	SECMPUS1[31:20]	Region Start Address	Address where the region starts, for use in region determination. The write value should always be 1FFh or 200h.	R

The SECMPUS1 and SECMPUE1 registers specify the secure region of SRAM (1FF0 0000h to 200F FFFFh), which can be accessed only from the secure program set up by SECMPUPCSn and SECMPUPCEn.

The SECMPUS1 register specifies the start address where the region starts. Setting of the stack area and the vector table are prohibited.

16.6.1.6 Security MPU Region 1 End Address Register (SECMPUE1)

Address(es): SECMPUE1 0000 0424h



Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. When writing to flash, the write value should always be 1.	R
b19 to b2	SECMPUE1[19:2]	Region End Address	Address where the region ends, for use in region determination.	R
b31 to b20	SECMPUE1[31:20]	Region End Address	Address where the region end, for use in region determination. The write value should always be 1FFh or 200h.	R

The SECMPUS1 and SECMPUE1 registers specify the secure region of SRAM (1FF0 0000h to 200F FFFFh). The memory space defined in the SECMPUS1 and SECMPUE1 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

The SECMPUE1 register specifies the end address where the region ends.

16.6.1.7 Security MPU Region 2 Start Address Register (SECMPUS2)

Address(es): SECMPUS2 0000 0428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUS2[22:16]						
Value after reset:	0	1	0	0	0	0	0	0	0	The value set by user						

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUS2[15:2]														—	—
Value after reset:	The value set by user														0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b22 to b2	SECMPUS2[22:2]	Region Start Address	Address where the region starts, for use in region determination	R
b29 to b23	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b30	—	Reserved	This bit is read as 1. When writing to flash, the write value should always be 1.	R
b31	—	Reserved	This bit is read as 0. When writing to flash, the write value should always be 0.	R

The SECMPUS2 and SECMPUE2 registers specify the secure region for security function 1 (400C 0000 to 400D FFFFh and 4010 0000 to 407F FFFFh). The secure region can be accessed only from the secure program set up by SECMPUPCSn and SECMPUPCEn.

The SECMPUS2 register specifies the start address where the region starts.

16.6.1.8 Security MPU Region 2 End Address Register (SECMPUE2)

Address(es): SECMPUE2 0000 042Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUE2[22:16]						
Value after reset:	0	1	0	0	0	0	0	0	0	The value set by user						

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUE2[15:2]														—	—
Value after reset:	The value set by user														1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. When writing to flash, the write value should always be 1.	R
b22 to b2	SECMPUE2[22:2]	Region End Address	Address where the region ends, for use in region determination.	R
b29 to b23	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b30	—	Reserved	This bit is read as 1. When writing to flash, the write value should always be 1.	R

Bit	Symbol	Bit name	Description	R/W
b31	—	Reserved	This bit is read as 0. When writing to flash, the write value should always be 0.	R

The SECMPUS2 and SECMPUE2 registers specify the secure region for security function 1 (400C 0000 to 400D FFFFh and 4010 0000 to 407F FFFFh). The memory space defined in the SECMPUS2 and SECMPUE2 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

The SECMPUE2 register specifies the end address where the region ends.

16.6.1.9 Security MPU Region 3 Start Address Register (SECMPUS3)

Address(es): SECMPUS3 0000 0430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUS3[22:16]						
Value after reset:	0	1	0	0	0	0	0	0	0	The value set by user						

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUS3[15:2]														—	—
Value after reset:	The value set by user														0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b22 to b2	SECMPUS3[22:2]	Region Start Address	Address where the region starts, for use in region determination.	R
b29 to b23	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b30	—	Reserved	This bit is read as 1. When writing to flash, the write value should always be 1.	R
b31	—	Reserved	This bit is read as 0. When writing to flash, the write value should always be 0.	R

The SECMPUS3 and SECMPUE3 registers specify the secure region for security function 2 (400C 0000h to 400D FFFFh and 4010 0000h to 407F FFFFh). The secure region can be accessed only from the secure program set up by SECMPUPCSn and SECMPUPCEn.

The SECMPUS3 register specifies the start address where the region starts.

16.6.1.10 Security MPU Region 3 End Address Register (SECMPUE3)

Address(es): SECMPUE3 0000 0434h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	SECMPUE3[22:16]						
Value after reset:	0	1	0	0	0	0	0	0	0	The value set by user						

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUE3[15:2]														—	—
Value after reset:	The value set by user														1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. When writing to flash, the write value should always be 1.	R
b22 to b2	SECMPUE3[22:2]	Region End Address	Address where the region ends, for use in region determination.	R
b29 to b23	—	Reserved	These bits are read as 0. When writing to flash, the write value should always be 0.	R
b30	—	Reserved	This bit is read as 1. When writing to flash, the write value should always be 1.	R
b31	—	Reserved	This bit is read as 0. When writing to flash, the write value should always be 0.	R

The SECMPUS3 and SECMPUE3 registers specify the secure region for security function 2 (400C 0000h to 400D FFFFh and 4010 0000h to 407F FFFFh). The memory space defined in the SECMPUS3 and SECMPUE3 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers.

The SECMPUE3 register specifies the end address where the region ends.

16.6.1.11 Security MPU Access Control Register (SECMPUAC)

Address(es): SECMPUAC 0000 0438h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DISPC1	DISPC0	—	—	—	—	DIS3	DIS2	DIS1	DIS0
Value after reset:	1	1	1	1	1	1	The value set by user		1	1	1	1	The value set by user			

Bit	Symbol	Bit name	Description	R/W
b0	DIS0	Region 0 Disable	0: Security MPU region 0 is enabled 1: Security MPU region 0 is disabled.	R
b1	DIS1	Region 1 Disable	0: Security MPU region 1 is enabled 1: Security MPU region 1 is disabled.	R
b2	DIS2	Region 2 Disable	0: Security MPU region 2 is enabled 1: Security MPU region 2 is disabled.	R
b3	DIS3	Region 3 Disable	0: Security MPU region 3 is enabled 1: Security MPU region 3 is disabled.	R
b7 to b4	—	Reserved	These bits are read as 1. When writing to flash, the write value should always be 1.	R
b8	DISPC0	PC Region 0 Disable	0: Security MPU PC region 0 is enabled 1: Security MPU PC region 0 is disabled.	R

Bit	Symbol	Bit name	Description	R/W
b9	DISPC1	PC Region 1 Disable	0: Security MPU PC region 1 is enabled 1: Security MPU PC region 1 is disabled.	R
b15 to b10	—	Reserved	These bits are read as 1. When writing to flash, the write value should always be 1.	R

Note: When flash memory is erased, the security MPU is disabled.
To enable and disable the security MPU, see section 16.6.2.1, Memory protection.

DIS0 bit (Region 0 Disable)

DIS0 bit enables or disables security MPU region 0.

If security MPU region 0 is enabled, the code flash region within the limits set up by SECMPUS0 and SECMPUE0 is a secure region.

DIS1 bit (Region 1 Disable)

DIS1 bit enables or disables security MPU region 1.

If security MPU region 1 is enabled, the SRAM region within the limits set up by SECMPUS1 and SECMPUE1 is a secure region.

DIS2 bit (Region 2 Disable)

DIS2 bit enables or disables security MPU region 2.

If security MPU region 2 is enabled, the region within the limits set up by SECMPUS2 and SECMPUE2 is a secure region.

DIS3 bit (Region 3 Disable)

DIS3 bit enables or disables security MPU region 3.

If security MPU region 3 is enabled, the region within the limits set up by SECMPUS3 and SECMPUE3 is a secure region.

DISPC0 bit (PC Region 0 Disable)

DISPC0 bit enables or disables security MPU PC region 0.

If security MPU PC region 0 is enabled, the code flash or SRAM region within the limits set up by SECMPUPCS0 and SECMPUPCE0 is a secure program.

DISPC1 bit (PC Region 1 Disable)

DISPC1 bit enables or disables security MPU PC region 1.

If security MPU PC region 1 is enabled, the code flash or SRAM region within the limits set up by SECMPUPCS1 and SECMPUPCE1 is a secure program.

16.6.2 Operation

16.6.2.1 Memory protection

The security MPU protects the regions (code flash, SRAM, two security function regions) from non-secure program access. If access to the protected region is detected, access becomes invalid.

When the security MPU is enabled, DISPC0 or DISPC1 in the Security MPU Access Control Register (SECMPUAC) must be cleared to 0 and DIS0, DIS1, DIS2, or DIS3 in the Security MPU Access Control Register (SECMPUAC) must be cleared to 0.

When security MPU is disabled, all of bits DISPC0, DISPC1, DIS0, DIS1, DIS2, and DIS3 in the Security MPU Access Control Register (SECMPUAC) must be set to 1.

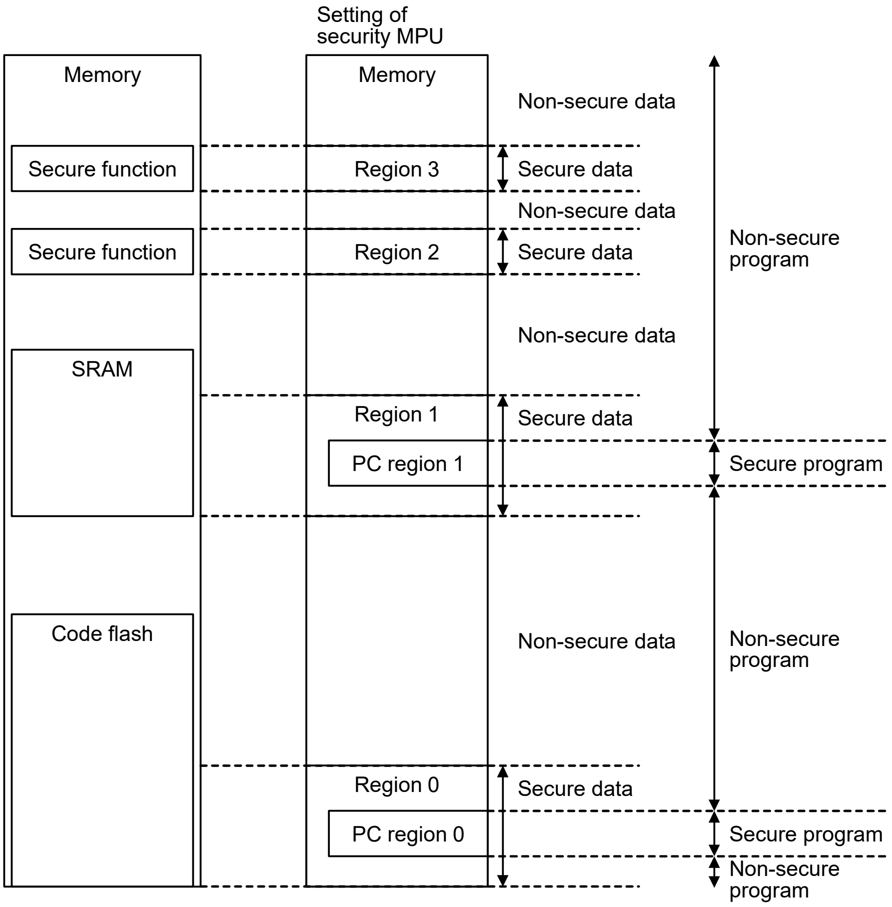
Other settings of the Security MPU Access Control Register (SECMPUAC) are prohibited.

The security MPU provides protection of secure regions when:

- Secure data is accessed from a non-secure program
- Secure data is accessed from other than the CPU (DMAC, DTC, EDMAC, GLCDC, DRW, JPEG)
- Secure data is accessed from the debugger.

Secure data can be accessed from a secure program.

Note: Secure program: Code flash or SRAM region within the limits set up by SECMPUPCS0 and SECMPUPCE0.
Code flash or SRAM region within the limits set up by SECMPUPCS1 and SECMPUPCE1.
Non-secure program: All regions without the secure program.
Secure data: Code flash region within the limits set up by SECMPUS0 and SECMPUE0.
SRAM region within the limits set up by SECMPUS1 and SECMPUE1.
Security function region within the limits set up by SECMPUS2 and SECMPUE2.
Security Function region within the limits set up by SECMPUS3 and SECMPUE3.



Secure program in code flash (PC region 0) can access all.
Secure program in SRAM (PC region 1) can access all.
Non-secure program (not PC region 0 or PC region 1) cannot access secure data (region 0, region 1, region 2, region 3).
Non-secure program (not PC region 0 or PC region 1) can access non-secure data.

Figure 16.11 Use case of security MPU

16.6.2.2 Notes on debug

The protected memory cannot be debugged if the security MPU is enable. Disable the security MPU when debugging a security program.

16.7 References

1. *ARM®v7-M Architecture Reference Manual* (ARM DDI 0403D).
2. *ARM® Cortex®-M4 Processor Technical Reference Manual* (ARM DDI 0439D).
3. *ARM® Cortex®-M4 Devices Generic User Guide* (ARM DUI 0553A).

17. DMA Controller (DMAC)

The 8-channel DMA Controller (DMAC) can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

17.1 Overview

Table 17.1 lists the DMAC specifications and Figure 17.1 shows a block diagram.

Table 17.1 DMAC specifications

Parameter		Specifications
Number of channels		8 channels (DMACm, m = 0 to 7)
Transfer space		4 GB (0000 0000h to FFFF FFFFh, excluding reserved areas)
Maximum transfer volume		64M data units (maximum number of transfers in block transfer mode: 1,024 data units × 65,536 blocks)
DMA activation source		Selectable for each channel: • Software trigger • Interrupt requests from peripheral modules or trigger from external interrupt input pins.*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3... > Channel 7 (Channel 0: highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Selectable free running mode (total number of data transfers is not specified).
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination • Maximum settable repeat size: 1,024.
	Block transfer mode	• One data block transfer by one DMA transfer request • Maximum settable block size: 1,024 data.
Selective functions	Extended repeat area function	• Allows data to be transferred by repeating the address values in the specified range, with the upper bit values in the transfer address register remaining fixed • Area of 2 bytes to 128 MB individually selectable as the extended repeat area for transfer source and destination.
Interrupt request (DMACm_INT)	Transfer end interrupt	Generated on completion of transferring data volume specified in the transfer counter
	Transfer escape end interrupt	Generated when: • The repeat size of data transfer is complete • The source address of the extended repeat area overflows • The destination address of the extended repeat area overflows.
Event link activation (DMACm_INT)		An event link request is generated after each data transfer (for block transfer, after each block is transferred)
Module-stop function		Module-stop state can be set to reduce power consumption

Note 1. For details on DMAC activation sources, see Table 14.3, Interrupt vector table, in section 14, Interrupt Controller Unit (ICU).

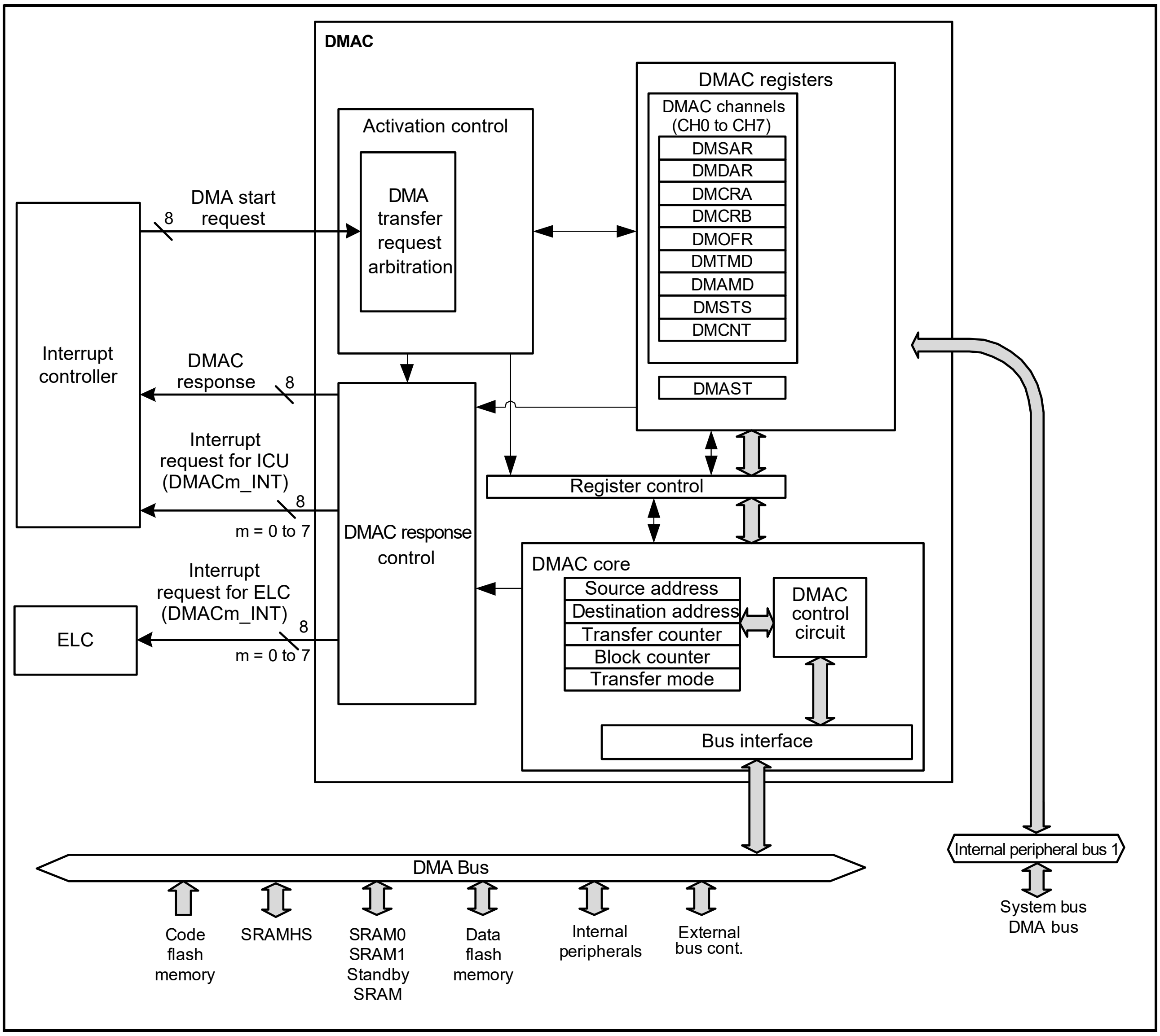


Figure 17.1 **DMAC block diagram**

17.2 Register Descriptions

17.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 4000 5000h, DMAC1.DMSAR 4000 5040h, DMAC2.DMSAR 4000 5080h, DMAC3.DMSAR 4000 50C0h, DMAC4.DMSAR 4000 5100h, DMAC5.DMSAR 4000 5140h, DMAC6.DMSAR 4000 5180h, DMAC7.DMSAR 4000 51C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting range	R/W
b31 to b0	Specifies the transfer source start address	0000 0000h to FFFF FFFFh (4 GB)	R/W

Set DMSAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Note: Address alignment in this register must match the transfer data size value selected in the SZ bit in DMTMD.

17.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 4000 5004h, DMAC1.DMDAR 4000 5044h, DMAC2.DMDAR 4000 5084h, DMAC3.DMDAR 4000 50C4h, DMAC4.DMDAR 4000 5104h, DMAC5.DMDAR 4000 5144h, DMAC6.DMDAR 4000 5184h, DMAC7.DMDAR 4000 51C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting range	R/W
b31 to b0	Specifies the transfer destination start address	0000 0000h to FFFF FFFFh (4 GB)	R/W

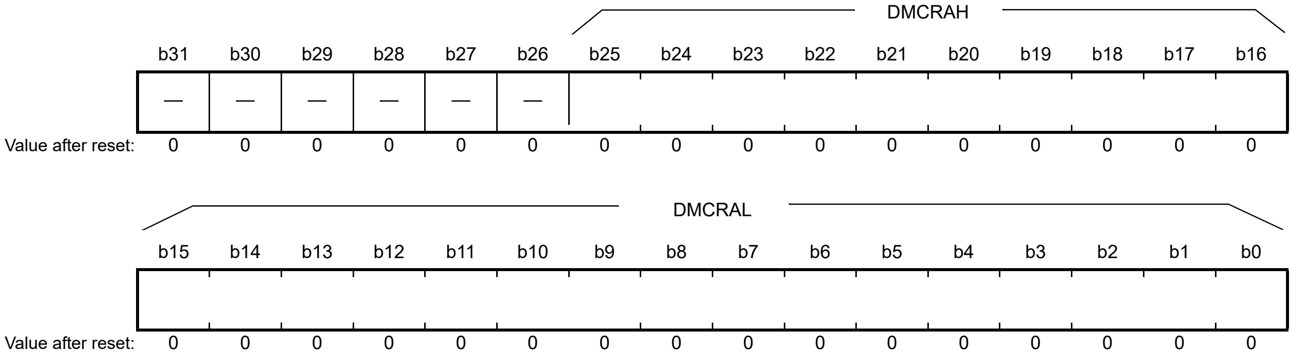
Set DMDAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Note: Address alignment in this register must match the transfer data size value selected in the SZ bit in DMTMD.

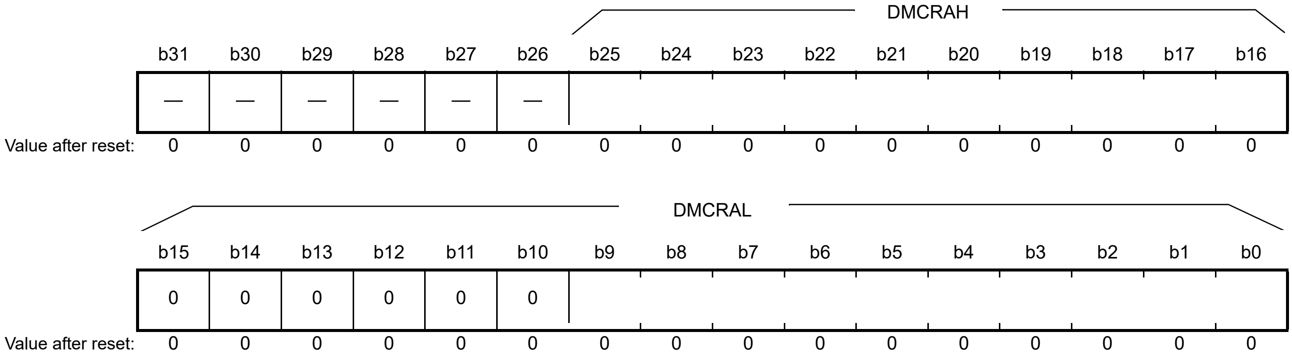
17.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 4000 5008h, DMAC1.DMCRA 4000 5048h, DMAC2.DMCRA 4000 5088h, DMAC3.DMCRA 4000 50C8h
DMAC4.DMCRA 4000 5108h, DMAC5.DMCRA 4000 5148h, DMAC6.DMCRA 4000 5188h, DMAC7.DMCRA 4000 51C8h

- Normal transfer mode



- Repeat transfer mode, block transfer mode



Symbol	Bit name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: In repeat and block transfer modes, set the same value for DMCRAH and DMCRAL.

(1) Normal transfer mode (MD[1:0] bits in DMACm.DMTMD = 00b)

In normal transfer mode, DMCRAL functions as a 16-bit transfer counter. The number of transfer operations is one when the setting is 0001h, and 65,535 when it is FFFFh. The value is decremented by one each time data is transferred.

A setting of 0000h indicates an unspecified number of transfer operations. Data transfer is performed with the transfer counter stopped, that is, in free running mode.

Do not use DMCRAH in normal transfer mode. Write 0000h to DMCRAH.

(2) Repeat transfer mode (MD[1:0] bits in DMACm.DMTMD = 01b)

In repeat transfer mode, DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter. The number of transfer operations is one when the setting is 001h, 1,023 when it is 3FFh, and 1,024 when it is 000h. In this mode, a value in the range of 000h to 3FFh (1 to 1,024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits. The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which time the value in DMCRAH is loaded into DMCRAL.

(3) Block transfer mode (MD[1:0] bits in DMACm.DMTMD = 10b)

In block transfer mode, DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter. The block size is one when the setting is 001h, 1,023 when it is 3FFh, and 1,024 when it is 000h. In this mode, a value in the range of 000h to 3FFh can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits. The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which time the value in DMCRAH is loaded into DMCRAL.

17.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 4000 500Ch, DMAC1.DMCRB 4000 504Ch, DMAC2.DMCRB 4000 508Ch, DMAC3.DMCRB 4000 50CCh, DMAC4.DMCRB 4000 510Ch, DMAC5.DMCRB 4000 514Ch, DMAC6.DMCRB 4000 518Ch, DMAC7.DMCRB 4000 51CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting range	R/W
b15 to b0	Specifies the number of block or repeat transfer operations	0001h to FFFFh (1 to 65,535) 0000h (65,536).	R/W

DMCRB specifies the number of operations in block and repeat transfer modes. The number of transfer operations is one when the setting is 0001h, 65,535 when it is FFFFh, and 65,536 when it is 0000h.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred. In block transfer mode, the value is decremented by one when the final data of one block size is transferred. Do not use DMCRB in normal transfer mode as the setting is invalid.

17.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 4000 5010h, DMAC1.DMTMD 4000 5050h, DMAC2.DMTMD 4000 5090h, DMAC3.DMTMD 4000 50D0h, DMAC4.DMTMD 4000 5110h, DMAC5.DMTMD 4000 5150h, DMAC6.DMTMD 4000 5190h, DMAC7.DMTMD 4000 51D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		DTS[1:0]		—	—	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*[1] from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: Specify destination as the repeat area or block area 0 1: Specify source as the repeat area or block area 1 0: Do not specify repeat area or block area 1 1: Setting prohibited.	R/W

Bit	Symbol	Bit name	Description	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited.	R/W

Note 1. To select the DMAC activation source, use the DELSRn registers of the ICU. For details on DMAC activation sources, see Table 14.4, Event table in section 14, Interrupt Controller Unit (ICU).

DTS[1:0] bits (Repeat Area Select)

The DTS[1:0] bits select either the source or destination as the repeat area in repeat transfer mode and the block area in block transfer mode. In normal transfer mode, these bit settings are invalid.

17.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 4000 5013h, DMAC1.DMINT 4000 5053h, DMAC2.DMINT 4000 5093h, DMAC3.DMINT 4000 50D3h, DMAC4.DMINT 4000 5113h, DMAC5.DMINT 4000 5153h, DMAC6.DMINT 4000 5193h, DMAC7.DMINT 4000 51D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disable 1: Enable.	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disable 1: Enable.	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disable 1: Enable.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disable 1: Enable.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disable 1: Enable.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DARIE bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow occurs on the destination address while the DARIE bit is set to 1, the DTE bit in DMCNT clears to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate an interrupt request triggered by an extended repeat area overflow on the destination address.

When block transfer mode is used with the extended repeat area function, an interrupt occurs after completion of a 1-block size transfer. When the DTE bit is set to 1 in DMACm.DMCNT of the channel associated with the stopped transfer, the transfer resumes from the state it was in when the transfer stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow occurs on the source address while the SARIE bit is set to 1, the DTE bit in DMCNT clears to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate an interrupt request triggered by an extended repeat area overflow on the source address.

When block transfer mode is used with the extended repeat area function, an interrupt occurs after completion of a 1-block size transfer. When the DTE bit is set to 1 in DMACm.DMCNT of the channel associated with the stopped transfer, the transfer resumes from the state it was in when the transfer stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE bit (Repeat Size End Interrupt Enable)

When the RPTIE bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT clears to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request occurred. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (repeat area or block area is not specified).

When the RPTIE bit is set to 1 in block transfer mode, the DTE bit in DMCNT clears to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request occurred. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (repeat area or block area is not specified).

ESIE bit (Transfer Escape End Interrupt Enable)

The ESIE bit enables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that occur during DMA transfer. The interrupt occurs when this bit is 1 and the ESIF flag in DMSTS is set to 1. To clear the transfer escape end interrupt, clear this bit or the ESIF flag in DMSTS to 0.

DTIE bit (Transfer End Interrupt Enable)

The DTIE bit enables the transfer end interrupt request that occurs on completion of a specified number of data transfers. The interrupt occurs when this bit is 1 and the DTIF flag in DMSTS is set to 1. To clear the transfer end interrupt, clear this bit or the DTIF flag in DMSTS to 0.

17.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 4000 5014h, DMAC1.DMAMD 4000 5054h, DMAC2.DMAMD 4000 5094h, DMAC3.DMAMD 4000 50D4h, DMAC4.DMAMD 4000 5114h, DMAC5.DMAMD 4000 5154h, DMAC6.DMAMD 4000 5194h, DMAC7.DMAMD 4000 51D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 17.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Fixed address 0 1: Offset addition 1 0: Incremented address 1 1: Decremented address.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 17.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Fixed address 0 1: Offset addition 1 0: Incremented address 1 1: Decremented address.	R/W

DARA[4:0] bits (Destination Address Extended Repeat Area)

The DARA[4:0] bits specify the extended repeat area on the destination address. The extended repeat area function is realized through an update of the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 MB.

The start address of the extended repeat area is set when the lower address overflows the extended repeat area on an address increment. Similarly, the end address of the extended repeat area is set when the lower address underflows the extended repeat area on an address decrement.

Do not specify the extended repeat area on the destination address when a repeat or block area is specified as the transfer destination. When repeat or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat or block area), write 00000b in the DARA[4:0] bits.

To request an interrupt when an overflow or underflow occurs in the extended repeat area, set the DARIE bit in DMINT to 1. Table 17.2 lists the extended repeat areas associated with each setting.

DM[1:0] bits (Destination Address Update Mode)

The DM[1:0] bits select the update mode for the destination address:

- When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively
- When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively
- When offset addition is selected, the offset specified in the DMACm.DMOFR register is added to the address.

SARA[4:0] bits (Source Address Extended Repeat Area)

The SARA[4:0] bits specify the extended repeat area on the source address. The extended repeat area function is realized through an update of the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 MB.

The start address of the extended repeat area is set when the lower address overflows the extended repeat area on an address increment. Similarly, the end address of the extended repeat area is set when the lower address underflows the extended repeat area on an address decrement.

Do not specify the extended repeat area on the source address when a repeat or block area is specified as the transfer source. When repeat or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat or block area), write 00000b in the SARA[4:0] bits.

To request an interrupt when an overflow or underflow occurs in the extended repeat area, set the SARIE bit in DMINT to 1. Table 17.2 lists the extended repeat areas associated with each setting.

SM[1:0] bits (Source Address Update Mode)

The SM[1:0] bits select the update mode for the source address:

- When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively
- When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively
- When offset addition is selected, the offset specified in the DMACm.DMOFR register is added to the address.

.

Table 17.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (1 of 2)

SARA[4:0] or DARA[4:0]	Extended repeat area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 KB specified as extended repeat area by the lower 10 bits of the address

Table 17.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (2 of 2)

SARA[4:0] or DARA[4:0]	Extended repeat area
01011b	2 KB specified as extended repeat area by the lower 11 bits of the address
01100b	4 KB specified as extended repeat area by the lower 12 bits of the address
01101b	8 KB specified as extended repeat area by the lower 13 bits of the address
01110b	16 KB specified as extended repeat area by the lower 14 bits of the address
01111b	32 KB specified as extended repeat area by the lower 15 bits of the address
10000b	64 KB specified as extended repeat area by the lower 16 bits of the address
10001b	128 KB specified as extended repeat area by the lower 17 bits of the address
10010b	256 KB specified as extended repeat area by the lower 18 bits of the address
10011b	512 KB specified as extended repeat area by the lower 19 bits of the address
10100b	1 MB specified as extended repeat area by the lower 20 bits of the address
10101b	2 MB specified as extended repeat area by the lower 21 bits of the address
10110b	4 MB specified as extended repeat area by the lower 22 bits of the address
10111b	8 MB specified as extended repeat area by the lower 23 bits of the address
11000b	16 MB specified as extended repeat area by the lower 24 bits of the address
11001b	32 MB specified as extended repeat area by the lower 25 bits of the address
11010b	64 MB specified as extended repeat area by the lower 26 bits of the address
11011b	128 MB specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited

17.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 4000 5018h, DMAC1.DMOFR 4000 5058h, DMAC2.DMOFR 4000 5098h, DMAC3.DMOFR 4000 50D8h, DMAC4.DMOFR 4000 5118h, DMAC5.DMOFR 4000 5158h, DMAC6.DMOFR 4000 5198h, DMAC7.DMOFR 4000 51D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	Setting range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination	0000 0000h to 00FF FFFFh (0 bytes to (16 MB - 1 byte)) FF00 0000h to FFFF FFFFh (-16 MB to -1 byte).	R/W

Only write to this register while DMAC operation is stopped or DMA transfer is disabled, not during data transfer. Setting bits 31 to 25 is invalid. The value in bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

17.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 4000 501Ch, DMAC1.DMCNT 4000 505Ch, DMAC2.DMCNT 4000 509Ch, DMAC3.DMCNT 4000 50DCh, DMAC4.DMCNT 4000 511Ch, DMAC5.DMCNT 4000 515Ch, DMAC6.DMCNT 4000 519Ch, DMAC7.DMCNT 4000 51DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disable 1: Enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTE bit (DMA Transfer Enable)

The DTE bit enables DMA transfer. To enable DMA transfer, set the DMST bit in DMAST to 1 to enable DMAC activation, then set the DTE bit to 1 to enable DMA transfer for the associated channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit
- When the specified volume of data transfer is complete
- When DMA transfer is stopped by a repeat size end interrupt
- When DMA transfer is stopped by an extended repeat area overflow interrupt.

17.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 4000 501Dh, DMAC1.DMREQ 4000 505Dh, DMAC2.DMREQ 4000 509Dh, DMAC3.DMREQ 4000 50DDh, DMAC4.DMREQ 4000 511Dh, DMAC5.DMREQ 4000 515Dh, DMAC6.DMREQ 4000 519Dh, DMAC7.DMREQ 4000 51DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SWREQ	DMA Software Start	0: Do not request DMA transfer 1: Request DMA transfer.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: Clear SWREQ bit after DMA transfer is started by software 1: Do not clear SWREQ bit after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SWREQ bit (DMA Software Start)

Writing 1 to the SWREQ bit generates a DMA transfer request. After DMA transfer starts in response, SWREQ clears to 0 if the CLRS bit is 0.

SWREQ does not clear to 0 if CLRS is 1. A DMA transfer request is issued again after completion of the transfer.

Note: Setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b, specifying software as the DMA activation source. Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to any value other than 00b.

To start DMA transfer by software with the CLRS bit set to 0, ensure that the SWREQ bit is 0, then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started with the CLRS bit set to 0 (the SWREQ bit is cleared after DMA transfer is started by software)
- When 0 is written to this bit.

CLRS bit (DMA Software Start Bit Auto Clear Select)

When an SWREQ setting of 1 triggers a transfer request, the CLRS bit specifies whether to clear the SWREQ bit to 0 after DMA transfer starts in response:

- When CLRS is set to 0, SWREQ clears to 0 after DMA transfer starts.
- When CLRS is set to 1, SWREQ does not clear to 0. A DMA transfer request is issued again after completion of the transfer.

17.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 4000 501Eh, DMAC1.DMSTS 4000 505Eh, DMAC2.DMSTS 4000 509Eh, DMAC3.DMSTS 4000 50DEh, DMAC4.DMSTS 4000 511Eh, DMAC5.DMSTS 4000 515Eh, DMAC6.DMSTS 4000 519Eh, DMAC7.DMSTS 4000 51DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: No interrupt occurred 1: Interrupt occurred.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: No interrupt occurred 1: Interrupt occurred.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation suspended 1: DMAC operating.	R

Note 1. Only 0 can be written, to clear the flag.

ESIF flag (Transfer Escape End Interrupt Flag)

The ESIF flag indicates that a transfer escape end interrupt occurred.

[Setting conditions]

- In repeat transfer mode, when one repeat size data transfer completes with the RPTIE bit in DMINT set to 1
- In block transfer mode, when one block data transfer completes with the RPTIE bit in DMINT set to 1
- When an extended repeat area overflow on the source address occurs with the SARIE bit in DMINT set to 1, and the SARA[4:0] bits in DMAMD set to any value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs with the DARIE bit in DMINT set to 1,

and the DARA[4:0] bits in DMAMD set to any value other than 00000b (extended repeat area is specified on the transfer destination address).

[Clearing conditions]

- When 0 is written to this flag
- When 1 is written to the DTE bit in DMCNT.

DTIF flag (Transfer End Interrupt Flag)

The DTIF flag indicates that a transfer end interrupt occurred.

[Setting conditions]

- In normal transfer mode, when the specified number of unit transfers completes (the value of DMCRAL becomes 0 on completion of transfer)
- In repeat transfer mode, when the specified number of repeat transfer operations completes (the value of DMCRB becomes 0 on completion of transfer)
- In block transfer mode, when the specified number of blocks is transferred (the value of DMCRB becomes 0 on completion of transfer).

[Clearing conditions]

- When 0 is written to this flag
- When 1 is written to the DTE bit in DMCNT.

ACT flag (DMA Active Flag)

The ACT flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts a data transfer.

[Clearing condition]

- When the data transfer in response to one transfer request completes.

17.2.12 DMAC Module Activation Register (DMAST)

Address(es): DMA.DMAST 4000 5200h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DMST	DMAC Operation Enable	0: Disable 1: Enable.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST bit (DMAC Operation Enable)

Setting the DMST bit to 1 enables DMAC activation for all channels. When the DMST bit is set to 1 (DMAC activation is enabled), and 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) for multiple channels, all of the associated channels can be placed in the transfer request ready state at the same time.

When the DMST bit is cleared to 0 during DMA transfer, DMA transfer is suspended after the current data transfer corresponding to a single transfer request completes. To resume DMA transfer, set the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit.

[Clearing condition]

- When 0 is written to this bit.

17.3 Operation

17.3.1 Transfer Mode

(1) Normal transfer mode

In normal transfer mode, one data unit is transferred for one transfer request. You can specify the number of transfer operations, up a maximum of 65,535, in DMACm.DMCRAL. When these bits are set to 0000h, no number of operations is specified and data transfer is performed with the transfer counter stopped (free running mode).

A transfer end interrupt request can be generated after completion of the specified number of transfer operations, except when data transfers are occurring in free running mode.

Setting DMACm.DMCRB is invalid in normal transfer mode.

Table 17.3 summarizes the register update operation in normal transfer mode.

Table 17.3 Register update operation in normal transfer mode

Register	Function	Update operation after completion of a transfer for one transfer request
DMACm.DMSAR	Transfer source address	Increment, decrement, fixed, or offset addition
DMACm.DMDAR	Transfer destination address	Increment, decrement, fixed, or offset addition
DMACm.DMCRAL	Transfer count	Decremented by one or not updated (in free running mode)
DMACm.DMCRAH	-	Not updated (not used in normal transfer mode)
DMACm.DMCRB	-	Not updated (not used in normal transfer mode)

Figure 17.2 shows the operation in normal transfer mode.

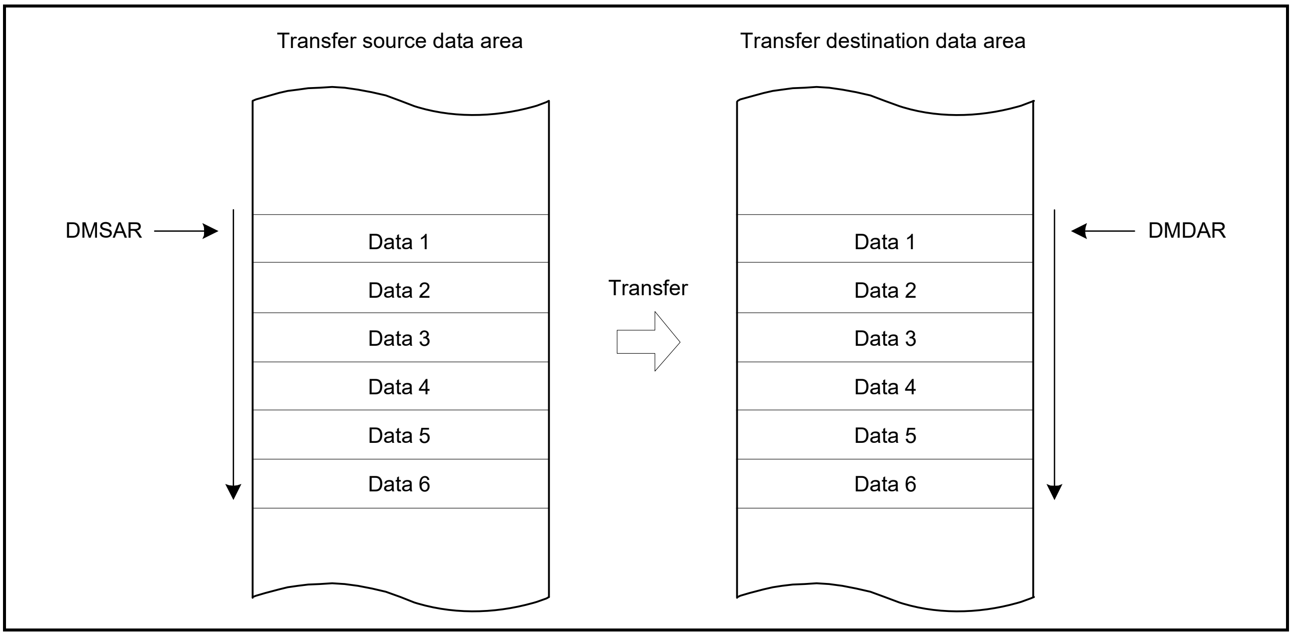


Figure 17.2 Operation in normal transfer mode

(2) Repeat transfer mode

In repeat transfer mode, one data unit is transferred for one transfer request. The repeat transfer size, up to a maximum of 1K data units, is set in DMACm.DMCRA.

The number of repeat transfer operations, up to a maximum of 64K, is set in DMACm.DMCRB. A maximum of 64M data units (1K data units × 64K repeat transfer operations) can be set as a total data transfer size.

You can specify either the transfer source or destination as a repeat area. When transfer of the repeat size data is complete, the address of the specified repeat area (DMSAR or DMDAR in DMACm) returns to the transfer start address. In this mode, when all data of the specified repeat size is transferred, DMA transfer can be stopped and a repeat size end interrupt can be requested. To resume DMA transfer, write 1 to the DTE bit in DMACm.DMCNT during repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfers.

Table 17.4 summarizes the register update operation in repeat transfer mode, and Figure 17.3 shows the operation in repeat transfer mode.

Table 17.4 Register update operation in repeat transfer mode

		Update operation after completion of a transfer for one transfer request	
Register	**Function**	**When DMACm.DMCRAL is not 1**	**When DMACm.DMCRAL is 1 (transfer of the last repeat size data unit)**
DMACm.DMSAR	Transfer source address	Increment, decrement, fixed, or offset addition	• DMACm.DMTMD.DTS[1:0] = 00b Increment, decrement, fixed, or offset addition • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement, fixed, or offset addition
DMACm.DMDAR	Transfer destination address	Increment, decrement, fixed, or offset addition	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment, decrement, fixed, or offset addition • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement, fixed, or offset addition
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer count	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Count of repeat transfer operations	Not updated	Decremented by one

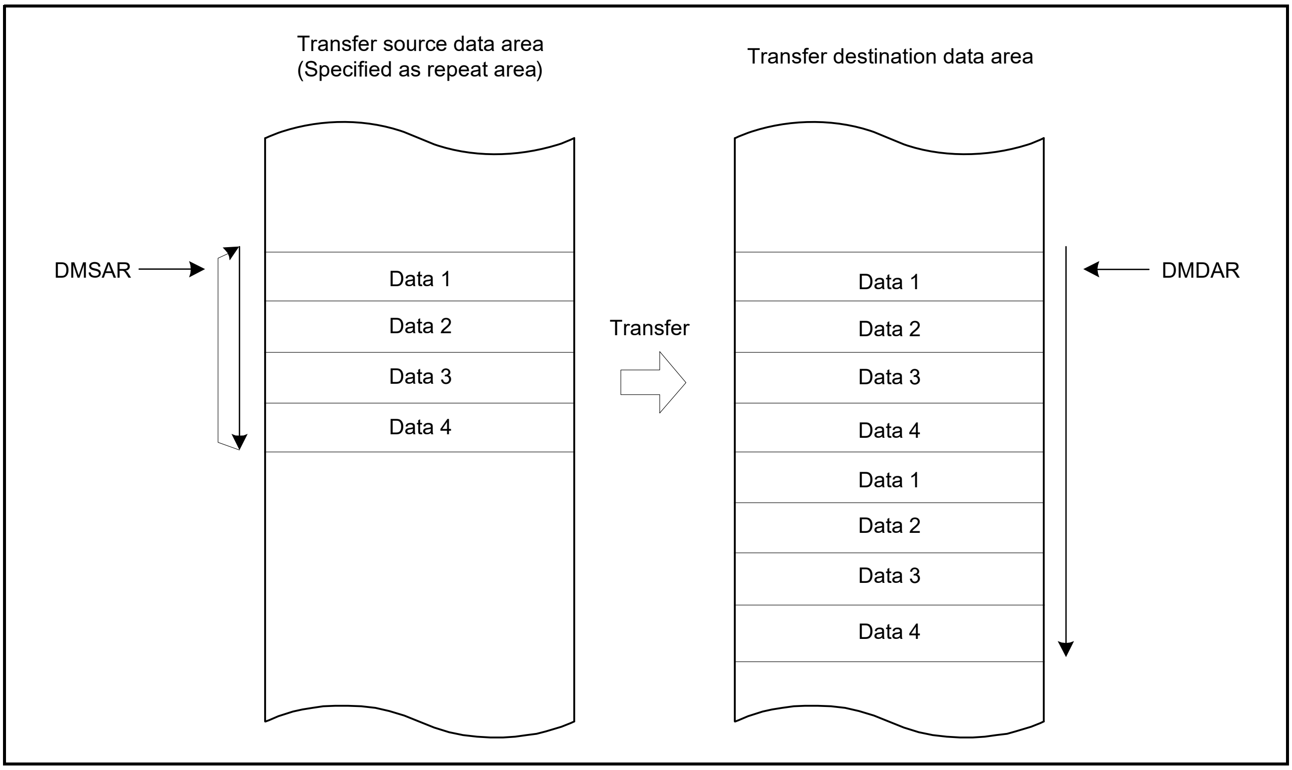


Figure 17.3 Operation in repeat transfer mode

(3) Block transfer mode

In block transfer mode, a single data block is transferred for one transfer request. The block size, up to a maximum of 1K data units, is set in DMACm.DMCRA.

The number of block transfers, up to a maximum of 64K, is set in DMACm.DMCRB. A maximum of 64M data units (1K data units × 64K block transfer operations) can be set as a total data transfer size.

You can specify either the transfer source or destination as a block area. When transfer of a single data block is complete, the address of the specified block area (DMSAR or DMDAR in DMACm) returns to the transfer start address. In this mode, when all data in a single block is transferred, DMA transfer can be stopped and a repeat size end interrupt can be requested. To resume DMA transfer, write 1 to the DTE bit in DMACm.DMCNT during repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of block transfers.

Table 17.5 summarizes the register update operation in block transfer mode, and Figure 17.4 shows the operation in block transfer mode.

Table 17.5 Register update operation in block transfer mode (1 of 2)

Register	Function	Update operation after completion of single-block transfer for one transfer request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment, decrement, fixed, or offset addition • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement, fixed, or offset addition.
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment, decrement, fixed, or offset addition • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement, fixed, or offset addition.

Table 17.5 Register update operation in block transfer mode (2 of 2)

Register	Function	Update operation after completion of single-block transfer for one transfer request
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Transfer count	DMACm.DMCRAH
DMACm.DMCRB	Count of block transfer operations	Decremented by one

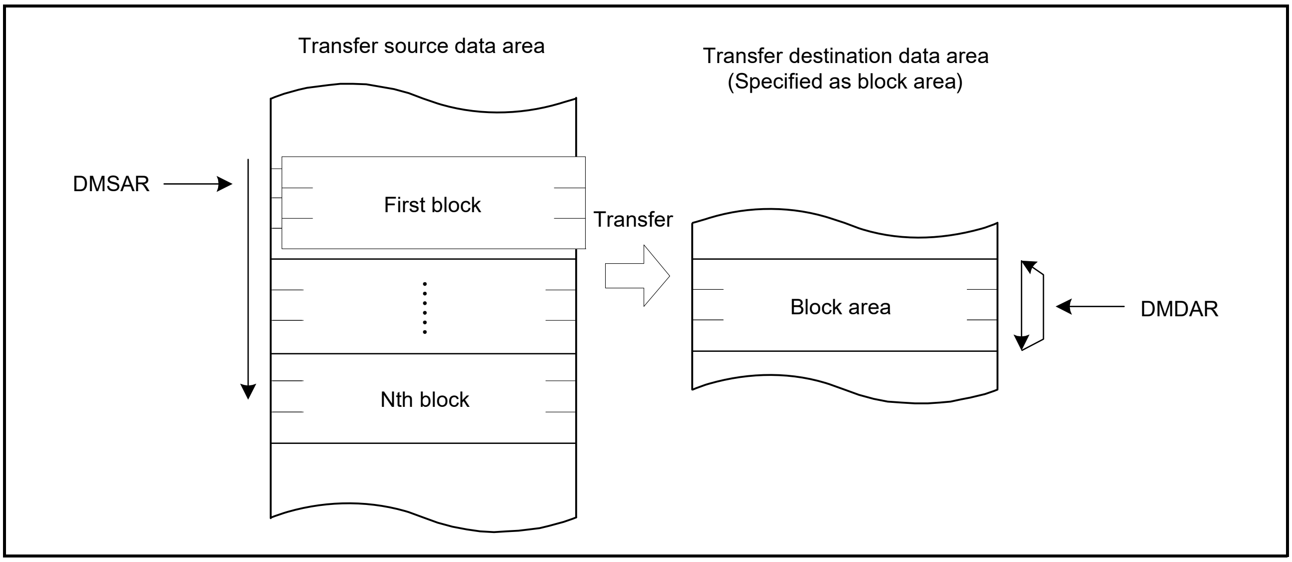


Figure 17.4 Operation in block transfer mode

17.3.2 Extended Repeat Area Function

The DMAC supports extended repeat areas on the transfer source and destination addresses, specified separately in the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm. When this function is set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat area on the source address is specified in the SARA[4:0] bits in DMACm.DMAMD. The extended repeat area on the destination address is specified in the DARA[4:0] bits in DMACm.DMAMD. You can specify different sizes for the source and destination. However, you must not specify a transfer source or destination that is set as the repeat or block area as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an extended repeat area overflow interrupt can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMACm.DMINT is set to 1, the ESIF flag in DMACm.DMSTS is set to 1 and the DTE bit in DMACm.DMCNT is cleared to 0 to stop DMA transfer. At this point, if the ESIE bit in DMACm.DMINT is set to 1, an extended repeat area overflow interrupt is requested. When the DARIE bit in DMACm.DMINT is set to 1, the destination address register becomes a target for the function. To resume DMA transfer, write 1 to the DTE bit in DMACm.DMCNT during interrupt handling.

Figure 17.5 shows an example of the extended repeat area operation.

Example:
8 bytes are specified as an extended repeat area in the lower 3 bits of DMACm.DMSAR (SARA[4:0] bits in DMACm.DMAMD = 00011b)
The data size is eight bits (SZ[1:0] bits in DMACm.DMTMD = 00b)

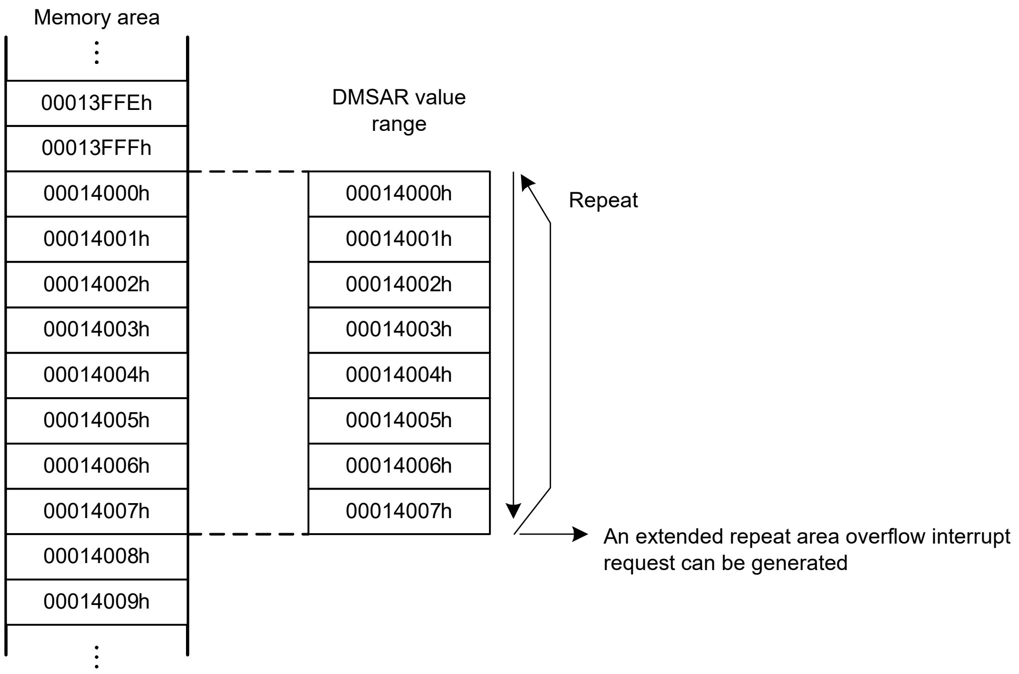


Figure 17.5 Example of extended repeat area operation

When using extended repeat area overflow interrupts in block transfer mode, consider the following points:

- When a transfer is stopped by an extended repeat area overflow interrupt, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the overflow interrupt is suspended until transfer of the block is complete, and the transfer overruns.

Figure 17.6 shows an example of using the extended repeat area function in block transfer mode.

Example:
8 bytes are specified as an extended repeat area in the lower3 bits of DMACm.DMSAR (SARA[4:0] bits in DMACm.DMAMD = 00011b)
Block transfer mode with block size 5 is set (DMACm.DMCRA = 00050005h)
The transfer source address is not specified as a block area
The data size is 8 bits (SZ[1:0] bits in DMACm.DTMD = 00b)

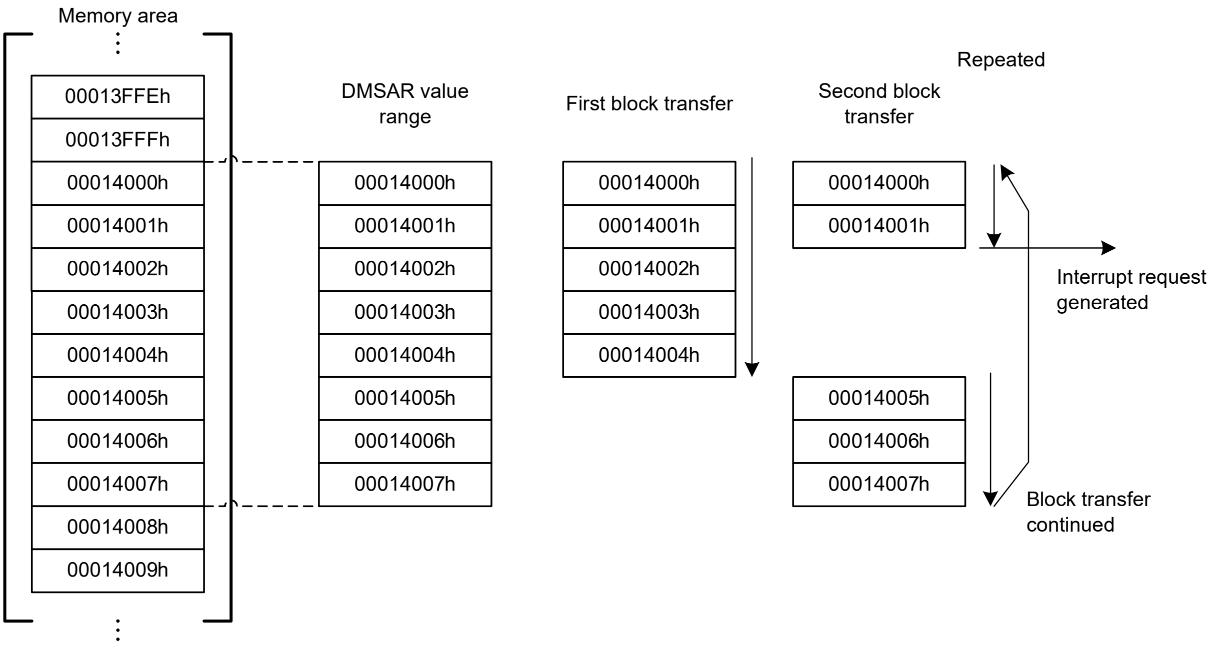


Figure 17.6 Example of extended repeat area function in block transfer mode

17.3.3 Address Update Function Using Offset

The source and destination addresses can be updated by fixing, incrementing, decrementing, or adding an offset. When offset addition is selected, the offset specified in the DMA Offset Register (DMACm.DMOFR) is added to the address every time the DMAC performs one data transfer. You can also subtract an offset by setting a negative value in DMACm.DMOFR. The negative value must be in two's complement.

Table 17.6 shows the address update method in each address update mode.

Table 17.6 Address update method in each address update mode

Address update mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for address update modes	Address update method for different SZ[1:0] settings in DMACm.DMTMD		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*[1]		
Increment	10b	+1	+2	+4
Decrement	11b	-1	-2	-4

Note 1. When setting a negative value in the DMA Offset Register, the value must be in two's complement, obtained by the following formula:
two's complement of a negative offset value = ~ (offset) + 1 (~ = bit inversion)

(1) Basic transfer using offset addition

Figure 17.7 shows an example of address updating using offset addition.

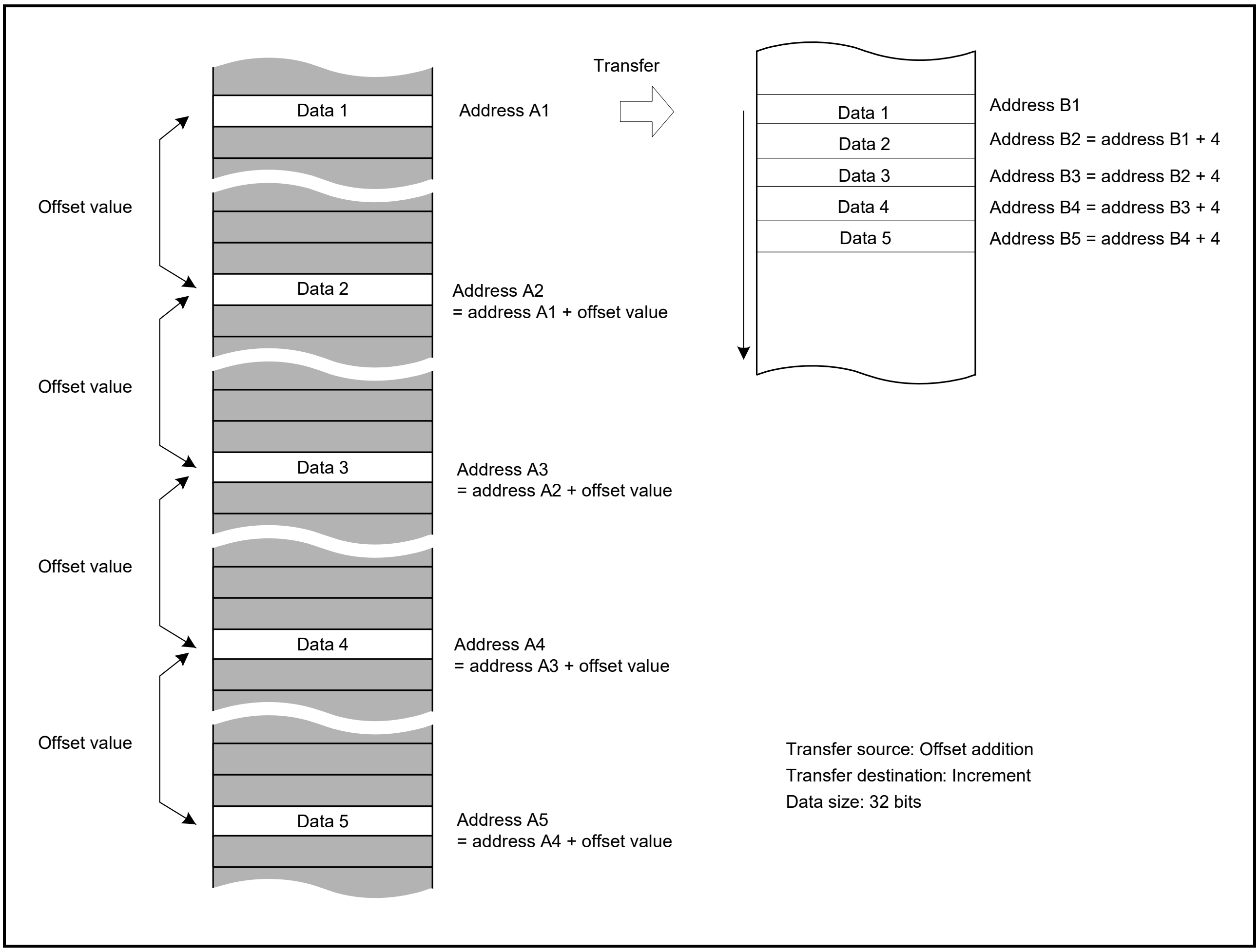


Figure 17.7 Example of address updating through offset addition

In Figure 17.7:

- The transfer data is 32 bits long
- Offset addition is set as the transfer source address update mode
- Increment is set as the transfer destination address update mode.

The second and subsequent data units are each read from the source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to continuous locations on the destination.

(2) Example of XY conversion using offset addition

Figure 17.8 shows XY conversion using offset addition in repeat transfer mode. The settings are as follows:

- DMAC0.DMAMD — Transfer source address update mode: offset addition
- DMAC0.DMAMD — Transfer destination address update mode: destination address is incremented
- DMAC0.DMTMD — Transfer data size select: 32 bits
- DMAC0.DMTMD — Transfer mode select: repeat transfer
- DMAC0.DMTMD — Repeat area select: the source is specified as the repeat area
- DMAC0.DMOFR — Offset address: 10h
- DMAC0.DMCRA — Repeat size: 4h
- DMAC0.DMINT — The repeat size end interrupt is enabled.

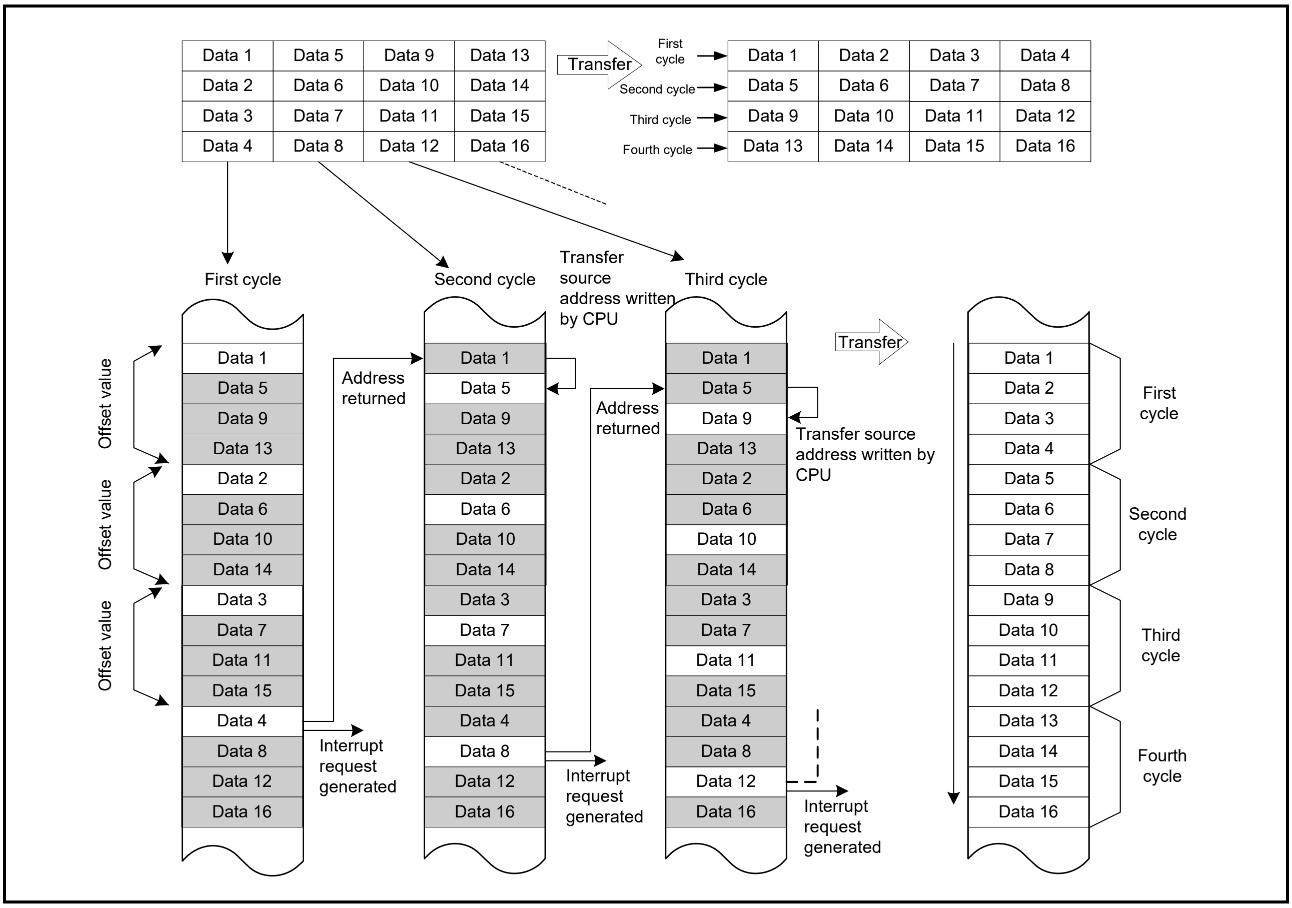


Figure 17.8 XY conversion operation using offset addition in repeat transfer mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to continuous destination addresses. When data 4 is transferred:

- The repeat size of the transfers is complete
- The transfer source address returns to the transfer start address (the address of data 1 on the transfer source)
- A repeat size end interrupt is requested.

During the time this interrupt pauses the transfer, perform the following:

- DMAC0.DMSAR — Rewrite the DMA transfer source address to the address of data 5 (in this example, the data 1 address + 4)
- DMAC0.DMCNT — Set the DTE bit to 1.

The DMA transfer resumes from the state when the DMA transfer was stopped. The same operations are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 17.9 shows the flow of the XY conversion.

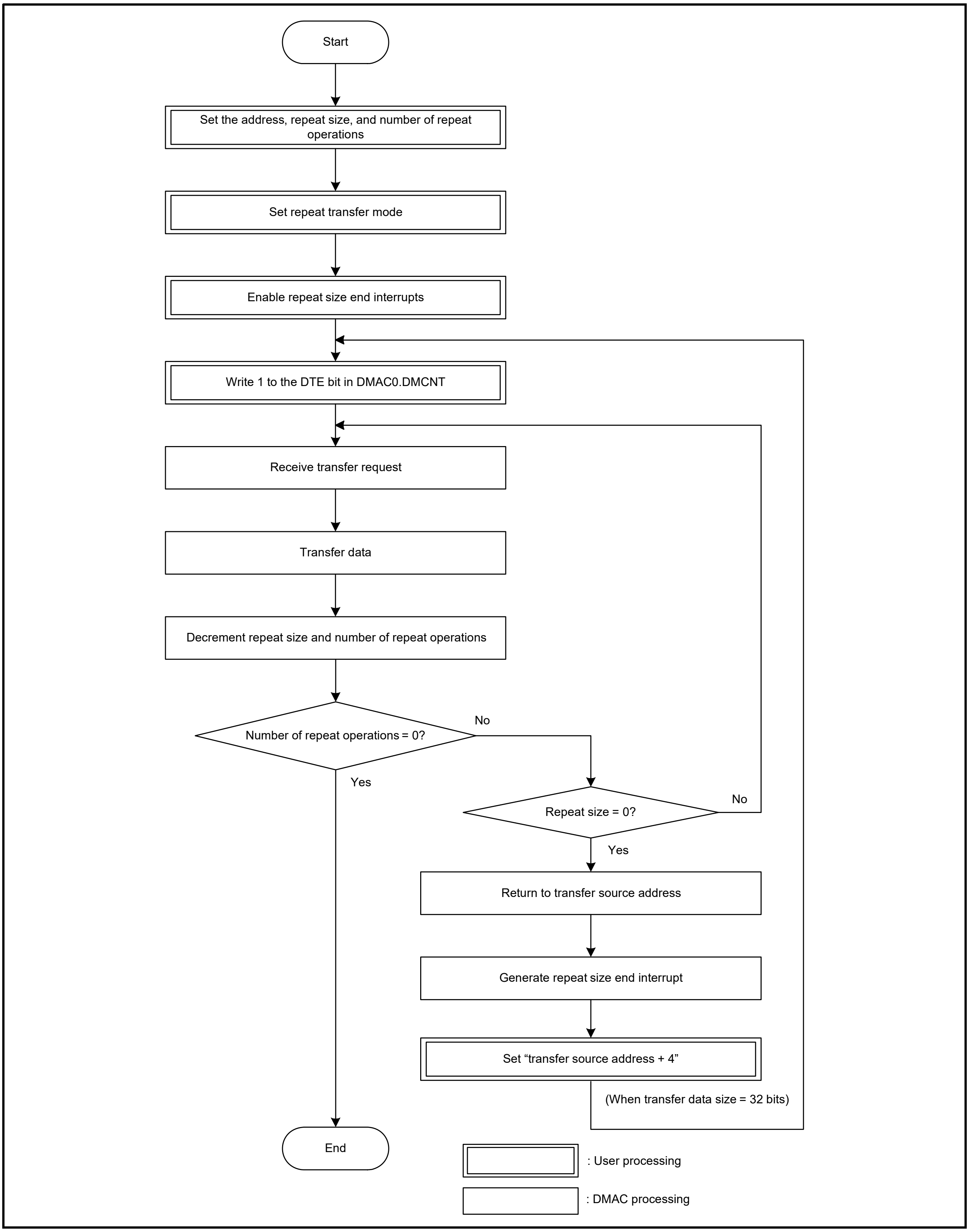


Figure 17.9 XY conversion flow using offset addition in repeat transfer mode

17.3.4 Activation Sources

Software, interrupt requests from the peripheral modules, and external interrupt requests can all be specified as DMAC activation sources. Set the DCTG[1:0] bits in DMACm.DMTMD to select the activation source.

(1) DMAC activation through software

To start DMA transfer through software:

1. Set the DCTG[1:0] bits in DMACm.DMTMD to 00b.
2. Set the DTE bit in DMACm.DMCNT to 1 (enable DMA transfer).
3. Set the DMST bit in DMAST to 1 (enable DMAC activation).
4. Set the SWREQ bit in DMACm.DMREQ to 1 (request DMA).

When the DMAC is activated by software while the CLRS bit in DMACm.DMREQ is 0, the SWREQ bit in DMACm.DMREQ clears to 0 after data transfer starts in response to a DMA transfer request.

When the DMAC is activated by software while the CLRS bit is 1, SWREQ does not clear to 0 after data transfer starts. A DMA transfer request is issued again after completion of a transfer.

(2) DMAC activation through interrupt requests from on-chip peripheral modules or external interrupt requests

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC activation sources. The activation source can be individually selected for each channel in ICU.DELSRn.DELS[8:0] (n = 0 to 7).

To start DMAC transfer through an interrupt request from an on-chip peripheral module or an external interrupt request:

1. Set the DCTG[1:0] bits in DMACm.DMTMD to 01b (select interrupts from the peripheral modules and the external interrupt pins).
2. Set the DTE bit in DMACm.DMCNT to 1 (enable DMA transfer).
3. Set ICU.DELSRn.DSEL to the event number (select the DMAC event link).
4. Set the DMST bit in DMAST to 1 (enable DMAC activation).

For interrupt requests specified as DMAC activation sources, see Table 14.3, Interrupt vector table, in section 14, Interrupt Controller Unit (ICU).

17.3.5 Operation Timing

The following diagrams show the timing with the minimum number of execution cycles.

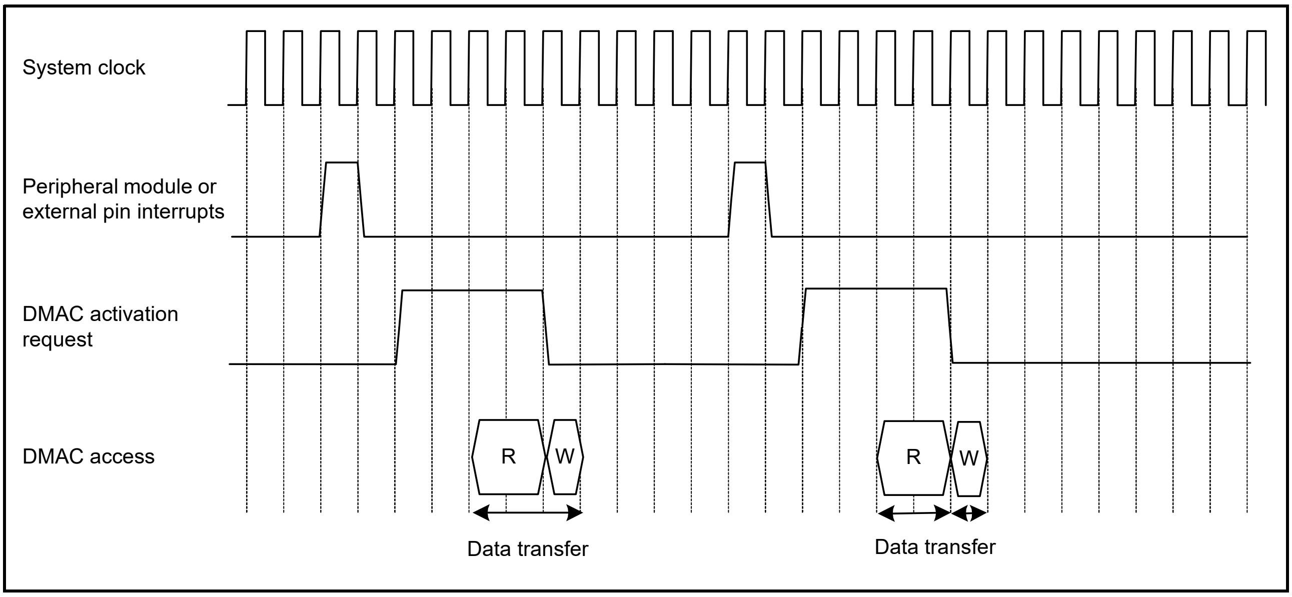


Figure 17.10 **DMAC operation timing example 1: DMA activation by interrupt from peripheral module or external interrupt input pin, in normal transfer mode or repeat transfer mode**

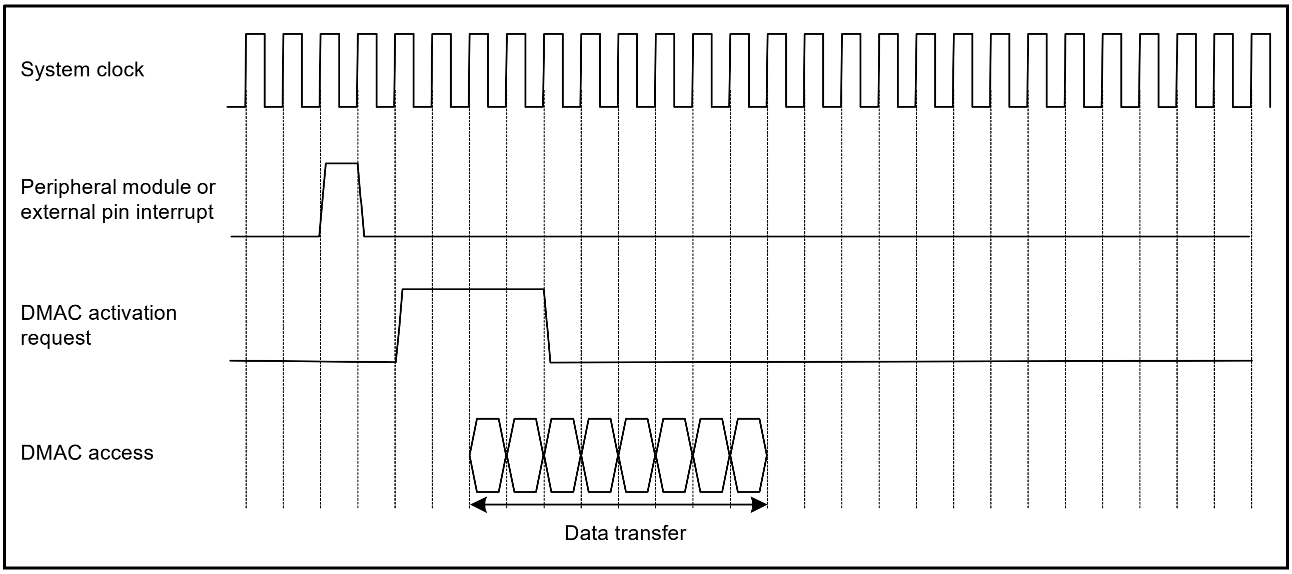


Figure 17.11 **DMAC operation timing example 2: DMA activation by interrupt from peripheral module or external interrupt input pin, in block transfer mode with block size = 4**

17.3.6 Execution Cycles of DMAC

Table 17.7 lists the execution cycles in one DMAC data transfer operation.

Table 17.7 **DMAC execution cycles**

Transfer mode	Data transfer (read)	Data transfer (write)
Normal	Cr+Cs+1	Cw+Cs
Repeat	Cr+Cs+1	Cw+Cs
Block*1	P × (Cr+Cs)	P × (Cw+Cs)

Note: P = Block size (DMCRAH register setting).
Cr = Read destination access cycle.
Cw = Data write destination access cycle.
Cs = When accessing SRAMHS and peripheral modules related to system control: 2 cycles.
When accessing elsewhere: 0 cycles.
When a slave bus changes by a read/write data transfer, add 1 more cycle.

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle applies.

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 53, SRAM, section 55, Flash Memory, and section 15.2.3, External Bus. The frequency ratio of the system clock and the peripheral clock is also taken into consideration.

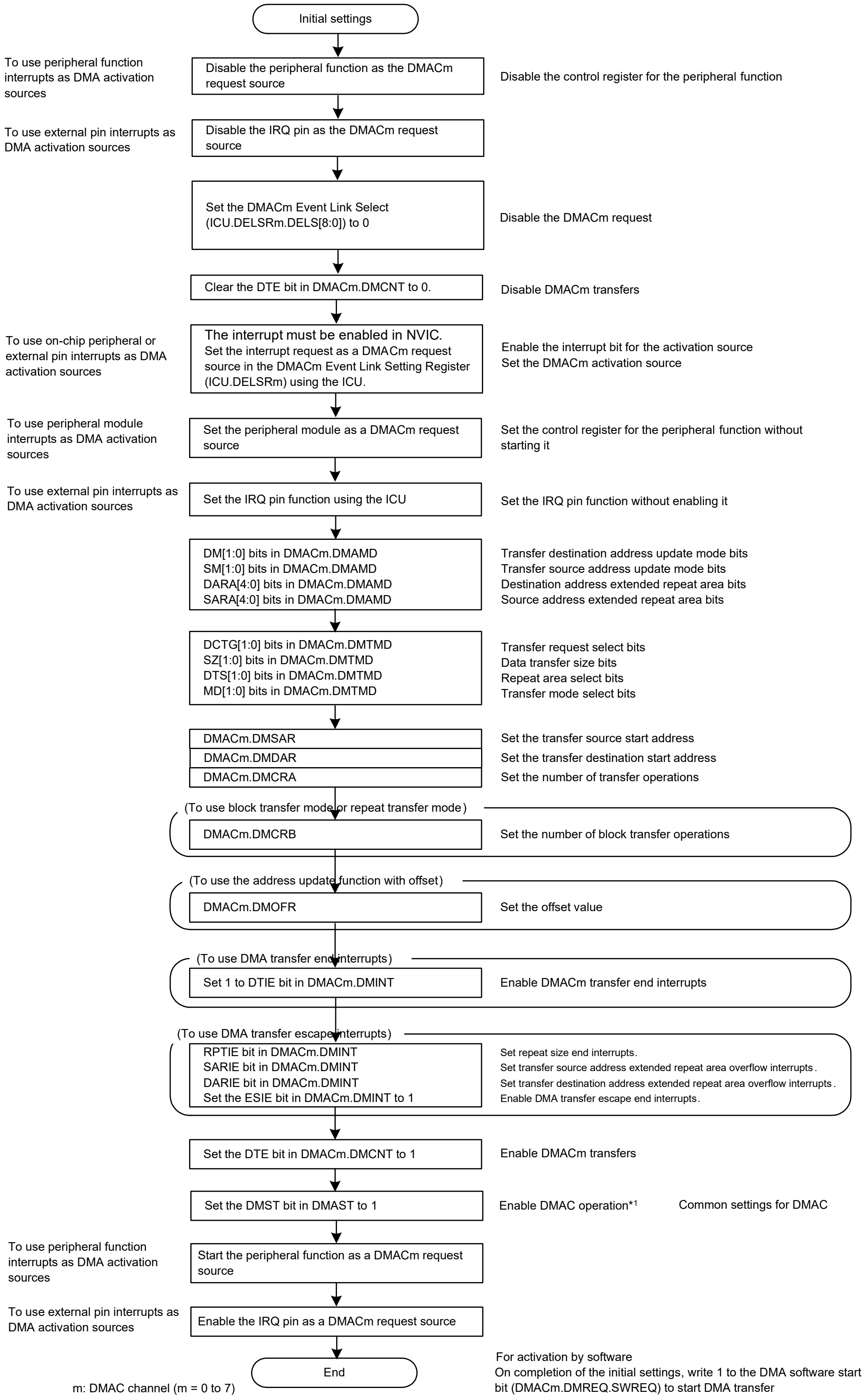
The unit for +1 in the data transfer (read) column is 1 system clock cycle, ICLK. For the operation example, see section 17.3.5, Operation Timing.

The DMAC response time is the time from when the DMAC activation source is detected until the DMAC transfer starts.

Table 17.7 does not include the time until the DMAC data transfer starts after the DMAC activation source becomes active.

17.3.7 Activating the DMAC

Figure 17.12 shows the register setting procedure.



Note 1. Setting of the DMAST.DMST bit does not necessarily have to follow the settings for the individual activation sources.

Figure 17.12 Register setting procedure

17.3.8 Starting DMA Transfer

To enable a DMA transfer of channel m, set the DTE bit in DMACm.DMCNT to 1 (DMA transfer enabled) and set the DMST bit in DMAST to 1 (DMAC activation enabled). New activation requests are not accepted during the transfer of another DMAC channel or DTC. When the proceeding transfer is complete, channel arbitration selects the DMA transfer request of the highest priority channel, and DMA transfer of that channel starts. When DMA transfer starts, the ACT flag in DMACm.DMSTS sets to 1 (the DMAC is in the active state).

17.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers that are updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMACm.DMSTS, described in the following sections. For details on register update operations in each transfer mode, see Table 17.3 to Table 17.5.

(1) DMA Source Address Register (DMACm.DMSAR)

After the data for one transfer request is transferred, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

(2) DMA Destination Address Register (DMACm.DMDAR)

After the data for one transfer request is transferred, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

(3) DMA Transfer Count Register (DMACm.DMCRA)

After the data for one transfer request is transferred, the count value is updated. The update operation depends on the transfer mode selected.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

After the data for one transfer request is transferred, the count value is updated. The update operation depends on the transfer mode selected.

(5) DMA Transfer Enable bit (DMACm.DMCNT.DTE)

The DMACm.DMCNT.DTE bit enables or disables data transfer through register write access. It is automatically cleared to 0 by the DMAC based on the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is complete
- When DMA transfer is stopped by a repeat size end interrupt
- When DMA transfer is stopped by an extended repeat area overflow interrupt.

Writing to the registers for channels whose associated DMACm.DMCNT.DTE bit is set to 1 is prohibited except for DMACm.DMCNT. Writes are only possible after the bit clears to 0.

(6) DMA Active flag (DMACm.DMSTS.ACT)

The ACT flag in DMSTS of DMACm indicates whether the DMACm is in the idle or active state. This flag sets to 1 when the DMAC starts data transfer, and clears to 0 when data transfer for one transfer request is complete. Even when DMA transfer is stopped by write of 0 to the DTE bit in DMACm.DMCNT, this flag remains 1 until DMA transfer is complete.

(7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DTIF flag in DMACm.DMSTS sets to 1 after DMA transfer of the total transfer size is complete. When both this flag and the DTIE bit in DMACm.DMINT are 1, a transfer end interrupt is requested. This flag sets to 1 when the DMA transfer bus cycle is complete and the ACT flag in DMACm.DMSTS clears to 0, indicating the DMA transfer end. The flag automatically clears to 0 when the DTE bit in DMACm.DMCNT is set to 1 during interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The ESIF flag in DMACm.DMSTS sets to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMACm.DMINT are 1, a transfer escape end interrupt is requested. This flag sets to 1 when the bus cycle of the DMA transfer that caused the interrupt request is complete and the ACT flag in DMACm.DMSTS clears to 0, indicating the DMA transfer end. The flag automatically clears to 0 when the DTE bit in DMACm.DMCNT is set to 1 during interrupt handling.

You must set the interrupt control register before sending an interrupt request from the DMAC to the CPU or the DTC. For more information, see section 14, Interrupt Controller Unit (ICU).

17.3.10 Channel Priority

When multiple DMA transfer requests occur, the DMAC determines the priority of channels that have DMA transfer requests.

The priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7. (Channel 0 is the highest.)

When a DMA transfer request occurs during data transfer, channel arbitration starts after the final data unit is transferred, and DMA transfer of the highest-priority channel starts.

17.4 Ending DMA Transfer

The operation for ending a DMA transfer depends on the transfer end conditions. When a DMA transfer ends, the DTE bit in DMCNT and the ACT flag in DMACm.DMSTS change from 1 to 0.

17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In normal transfer mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of DMACm.DMCRAL changes from 1 to 0, DMA transfer ends on the associated channel, the DTE bit in DMACm.DMCNT clears to 0, and the DTIF flag in DMACm.DMSTS sets to 1. If the DTIE bit in DMACm.DMINT is 1 at this time, a transfer end interrupt request is sent to the CPU or the DTC.

(2) In repeat transfer mode (DMACm.DMTMD.MD[1:0] = 01b)

When the value of DMACm.DMCRB changes from 1 to 0, DMA transfer ends on the associated channel, the DTE bit in DMACm.DMCNT clears to 0, and the DTIF flag in DMACm.DMSTS sets to 1. If the DTIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC.

(3) In block transfer mode (DMACm.DMTMD.MD[1:0] = 10b)

When the value of DMACm.DMCRB changes from 1 to 0, DMA transfer ends on the associated channel, the DTE bit in DMACm.DMCNT clears to 0, and the DTIF flag in DMACm.DMSTS sets to 1. If the DTIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC.

You must set the interrupt control register before sending an interrupt request from the DMAC to the CPU or the DTC. For more information, see section 14, Interrupt Controller Unit (ICU).

17.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, if the RPTIE bit in DMACm.DMINT is 1, a repeat size end interrupt is requested when transfer of a single repeat size of data is complete. The DTE bit in DMACm.DMCNT clears to 0 and the ESIF flag in DMACm.DMSTS sets to 1. If the ESIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC. To resume the transfer, write 1 to the DTE bit in DMACm.DMCNT.

A repeat size end interrupt can also be requested in block transfer mode. When transfer of a single block size of data is complete, the interrupt is requested in the same way as in repeat transfer mode.

You must set the interrupt control register before sending an interrupt request from the DMAC to the CPU or the DTC. For more information, see section 14, Interrupt Controller Unit (ICU).

17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMACm.DMINT is 1, an extended repeat area overflow interrupt is requested. The DMA transfer is terminated, the DTE bit in DMACm.DMCNT clears to 0, and the ESIF flag in DMACm.DMSTS sets to 1. If the ESIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC.

If this interrupt is requested during a read cycle, the subsequent write cycle is performed. In block transfer mode, if the interrupt is requested during a 1-block transfer, the remaining data in the block is transferred before transfer stops.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For more information, see section 14, Interrupt Controller Unit (ICU).

17.4.4 Precautions for the End of DMA Transfer

A DMA activation request source might occur in the next request after a DMA transfer completes. If this happens, the DMA transfer starts and the DMA activation request is held in DMAC. To prevent this, stop the DMA activation requests by clearing the DELSRn.DSELS[8:0] bits in the ICU to 0.

When the DMA activation request occurs after the last round of the DMA transfer is generated, clear the DMA activation request by setting ICU.DELSRm.IR bit to 0.

17.5 Interrupts

Each DMAC channel can output an interrupt request (DMACm_INT) to the CPU or DTC after transfer for one request is complete. When the transfer destination is the external bus, an interrupt request is generated after completion of a data write to the write buffer, and not to the actual transfer destination.

Table 17.8 lists the interrupt sources and their associated status flags and enable bits. Figure 17.13 shows the schematic logic diagram of the interrupt outputs (DMAC0 to DMAC7). Figure 17.14 shows the DMAC interrupt handling routine for resuming and terminating DMA transfers.

Table 17.8 Associations among interrupt sources, interrupt status flags, and interrupt enable bits

Interrupt sources		Interrupt enable bits	Interrupt status flags	Request output enable bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

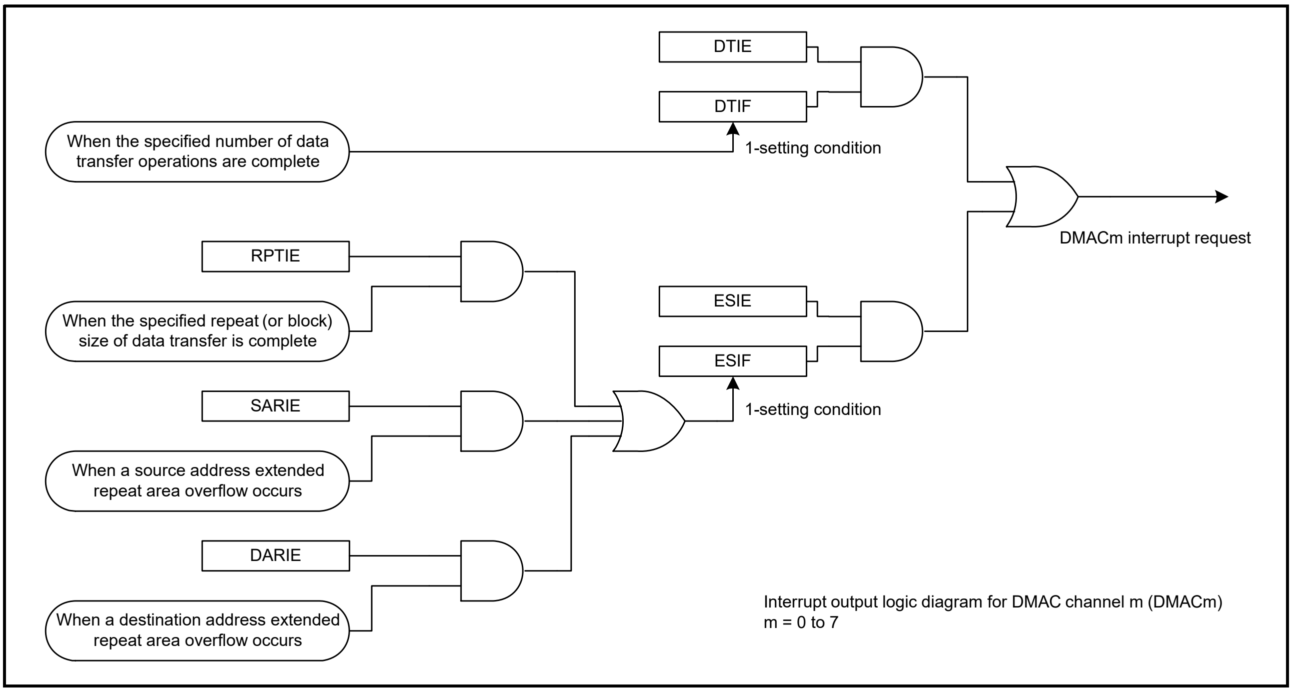


Figure 17.13 Schematic logic diagram of interrupt outputs for DMAC0 to DMAC7

Different procedures are used for canceling an interrupt to restart a DMA transfer in the following cases:

- When terminating a DMA transfer
- When continuing a DMA transfer.

(1) When terminating a DMA transfer

Write 0 to the DTIF flag in DMACm.DMSTS to clear a transfer end interrupt, and to the ESIF flag in DMACm.DMSTS to clear a repeat size interrupt or an extended repeat area overflow interrupt. DMACm remains in the stopped state. When starting another DMA transfer, set the appropriate registers and set the DTE bit in DMACm.DMCNT to 1 (DMA transfer enabled).

(2) When continuing a DMA transfer

Write 1 to the DTE bit in DMACm.DMCNT. The ESIF flag in DMSTS of DMACm automatically clears to 0 (interrupt source cleared), and the DMA transfer resumes.

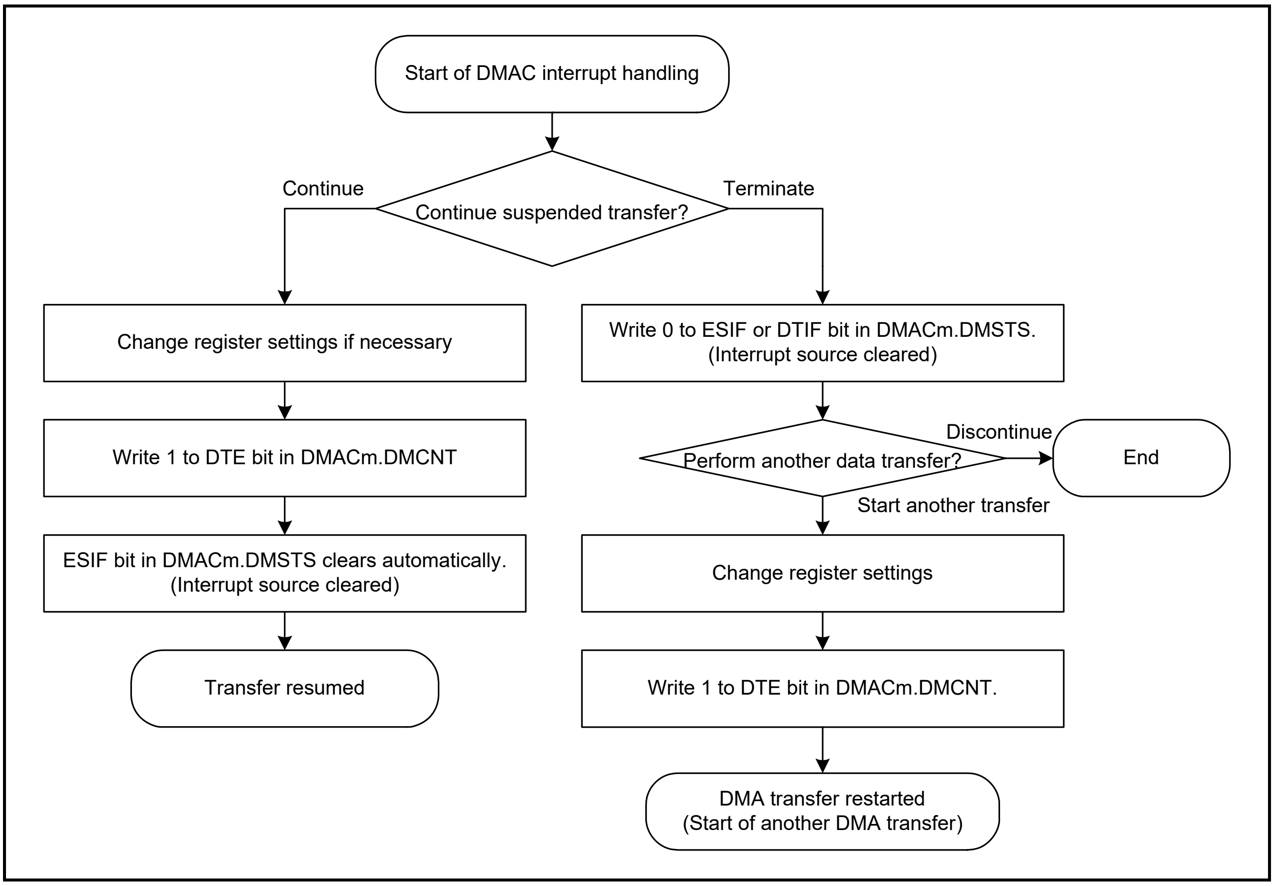


Figure 17.14 DMAC interrupt handling routine to resume or terminate a DMA transfer

17.6 Event Link

Each DMAC channel outputs an event link request signal (DMACm_INT) every time it completes a data transfer, or a block transfer in block transfer mode. When the transfer destination is the external bus, the signal is generated when writing to the write buffer is accepted. For more information, see section 19, Event Link Controller (ELC).

17.7 Low-Power Functions

Before entering the module-stop state, Software Standby mode, or Deep Software Standby mode, you must first clear the DMST bit in DMAST to 0 (DMAC suspended) and use the settings in the sections that follow.

(1) Module-stop function

Writing 1 to the MSTPA22 bit in MSTPCRA enables the module-stop function of the DMAC. If a DMA transfer is in progress when 1 is written to MSTPA22, the transition to the module-stop state continues after DMA transfer ends. Access to the DMAC registers is prohibited while MSTPA22 is 1. Writing 0 to the MSTPA22 bit releases the DMAC from the module-stop state.

(2) Software Standby and Deep Software Standby modes

Use the settings described in section 11.7.1, Transition to Software Standby Mode and section 11.9.1, Transition to Deep Software Standby Mode.

If DMA transfer operations are in progress when the WFI instruction is executed, the DMA transfer completes before the transition to Software Standby mode or Deep Software Standby mode.

(3) Notes on the low-power function

For information on the WFI instruction and register settings, see section 11.10.7, Timing of WFI Instruction.

To perform a DMA transfer after returning from a low power mode, set the DMST bit in DMAST to 1 again. To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination, as described in section 14.4.2, Selecting Interrupt Request Destinations, then execute the WFI instruction.

17.8 Usage Notes

17.8.1 DMA Transfer to External Devices

In a DMA transfer to an external device, the ACT flag in DMACm.DMSTS may be cleared to 0 (DMAC transfer suspended) from the beginning of the final data write to the end of the external bus access.

17.8.2 Access to Registers during DMA Transfer

Do not write to the following registers of DMACm while the ACT flag in DMSTS of the associated channel is set to 1 (DMAC active state) or the DTE bit in DMCNT of the associated channel is set to 1 (DMA transfer enabled):

- DMSAR
- DMDAR
- DMCRA
- DMCRB
- DMTMD
- DMINT
- DMAMD
- DMOFR.

17.8.3 DMA Transfer to Reserved Areas

DMA transfer to reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on reserved areas, see section 4, Address Space.

17.8.4 Setting the DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU) (ICU.DELSRn)

Before setting the DMAC Event Link Setting Register (ICU.DELSRn), make sure the DMA transfer enable bit (DMACm.DMCNT.DTE) is cleared to 0, disabling DMA transfer. Additionally, ensure that the DTC activation enable register (ICU.IELSRn.DTCE) associated with the event number set in the ICU.DELSRn register is not set to 1. For details on ICU.IELSRn.DTCE and ICU.DELSRn, see section 14, Interrupt Controller Unit (ICU).

17.8.5 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the DMAC Event Link select bits (ICU.DELSRn.DELS[8:0]). To restart the DMA transfer, write the event number to the ICU.DELSRn.DELS[8:0] bit with the settings shown in section 17.3.7, Activating the DMAC.

18. Data Transfer Controller (DTC)

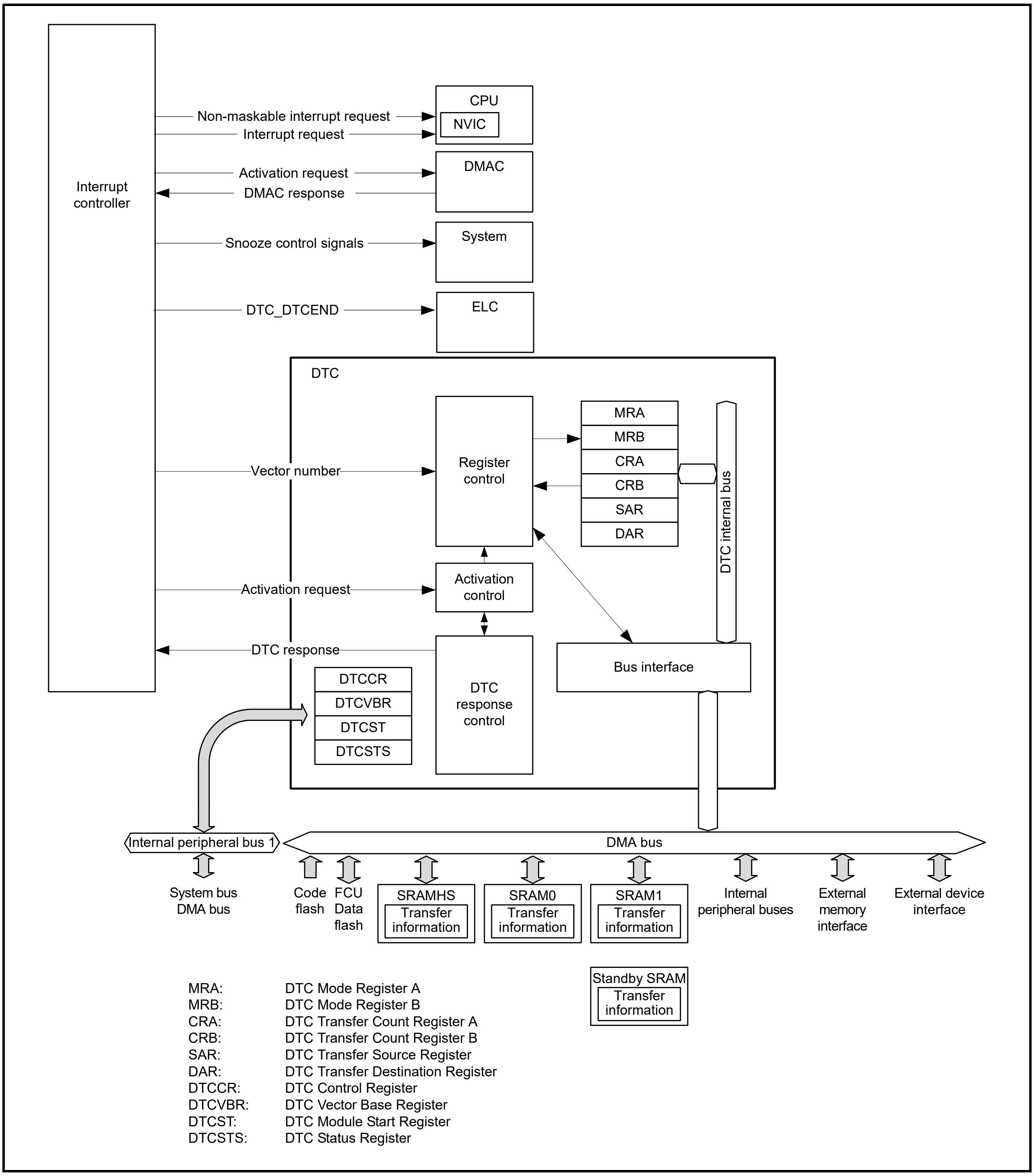
18.1 Overview

The Data Transfer Controller (DTC) performs data transfers when activated by an interrupt request.

Table 18.1 lists the DTC specifications and Figure 18.1 shows a block diagram.

Table 18.1 DTC specifications

Parameter	Specifications
Transfer modes	• Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation leads to a single data transfer. The transfer address returns to the start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256 × 32 bits (1024 bytes). • Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Transfer channel	• Channel transfer can be associated with the interrupt source (transferred by a DTC activation request from the ICU) • Multiple data units can be transferred on a single activation source (chain transfer) • Chain transfers can be set to either execute when the counter is 0 or always execute
Transfer space	• 4 GB (0000 0000h to FFFF FFFFh, excluding reserved areas)
Data transfer units	• Single data unit: 1 byte (8 bits), 1 halfword (16 bits), or 1 word (32 bits) • Single block size: 1 to 256 data units
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt • An interrupt request can be generated to the CPU after a single data transfer • An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption



MRA:	DTC Mode Register A
MRB:	DTC Mode Register B
CRA:	DTC Transfer Count Register A
CRB:	DTC Transfer Count Register B
SAR:	DTC Transfer Source Register
DAR:	DTC Transfer Destination Register
DTCCR:	DTC Control Register
DTCVBR:	DTC Vector Base Register
DTCST:	DTC Module Start Register
DTCSTS:	DTC Status Register

Figure 18.1 DTC block diagram

See 14.1 Overview, in section 14, Interrupt Controller Unit (ICU), for the connections between the DTC and NVIC (in the CPU).

18.2 Register Descriptions

MRA, MRB, SAR, DAR, CRA, and CRB are all DTC internal registers that cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the SRAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the SRAM area and sets it in its internal registers. After the data transfer ends, the internal register contents are written back to the SRAM area as transfer information.

18.2.1 DTC Mode Register A (MRA)

Address(es): (Inaccessible directly from the CPU. See section 18.3.1.)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: Address in the SAR register is fixed (write-back to SAR is skipped) 0 1: Address in the SAR register is fixed (write-back to SAR is skipped) 1 0: SAR value is incremented after data transfer: +1 when SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b. 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b.	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Setting prohibited.	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited.	—

The MRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 03h) and the DTC automatically transfers the MRA transfer information to and from the MRA register. See section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

18.2.2 DTC Mode Register B (MRB)

Address(es): (Inaccessible directly from the CPU. See section 18.3.1.)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—

Bit	Symbol	Bit name	Description	R/W
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when MRA.SZ[1:0] = 01b +4 when MRA.SZ[1:0] = 10b. 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when MRA.SZ[1:0] = 01b -4 when MRA.SZ[1:0] = 10b.	—
b4	DTS	DTC Transfer Mode Select	0: Select transfer destination as repeat or block area 1: Select transfer source as repeat or block area.	—
b5	DISEL	DTC Interrupt Select	0: Generate an interrupt request to CPU when specified data transfer is complete 1: Generate an interrupt request to CPU each time DTC data transfer is performed.	—
b6	CHNS	DTC Chain Transfer Select	0: Select continuous chain transfer 1: Select chain transfer occurring only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer disabled 1: Chain transfer enabled.	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 02h) and the DTC automatically transfers the MRB transfer information to and from the MRB register. See section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

DTS bit (DTC Transfer Mode Select)

The DTS bit selects either the transfer source or transfer destination as the repeat area or block area in repeat or block transfer mode.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see Table 18.3, Chain transfer conditions.

When the next transfer is a chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected in the CHNS bit. For details on chain transfer, see section 18.4.6, Chain Transfer.

18.2.3 DTC Transfer Source Register (SAR)

Address(es): (Inaccessible directly from the CPU. See section 18.3.1.)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

The SAR register sets the transfer source start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 04h) and the DTC automatically transfers the SAR transfer information to and from the SAR register. See section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

Note: Misalignment is prohibited for DTC transfers. Bit [0] must be 0 when MRA.SZ[1:0] = 01b.
Bits [1] and [0] must be 0 when MRA.SZ[1:0] = 10b.

18.2.4 DTC Transfer Destination Register (DAR)

Address(es): (Inaccessible directly from the CPU. See section 18.3.1.)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

The DAR register sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 08h) and the DTC automatically transfers the DAR transfer information to and from the DAR register. See section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

Note: Misalignment is prohibited for DTC transfers. Bit [0] must be 0 when MRA.SZ[1:0] = 01b.
Bits [1] and [0] must be 0 when MRA.SZ[1:0] = 10b.

18.2.5 DTC Transfer Count Register A (CRA)

Address(es): (Inaccessible directly from the CPU. See section 18.3.1.)

- Normal transfer mode

	CRA															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

- Repeat transfer mode/block transfer mode

	CRAH								CRAL							
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Symbol	Register name	Description	R/W
CRAL	Transfer Counter A Lower Register	Specify the transfer count	—
CRAH	Transfer Counter A Upper Register		—

Note: The function depends on the transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

The CRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0Eh) and the DTC automatically transfers the CRA transfer information to and from the CRA register. See section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

(1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, CRA functions as a 16-bit transfer counter. The transfer count is 1, 65,535, and 65,536 when the set value is 0001h, FFFFh, and 0000h, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MRA.MD[1:0] = 10b)

In block transfer mode, the CRAH register holds the block size and the CRAL register functions as an 8-bit block size counter. The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively. The CRAL value is decremented (-1) at each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

18.2.6 DTC Transfer Count Register B (CRB)

Address(es): (Inaccessible directly from the CPU. See section 18.3.1.)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

The CRB register sets the block transfer count for block transfer mode. The transfer count is 1, 65,535, and 65,536 when the set value is 0001h, FFFFh, and 0000h, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

CRB cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0ch) and the DTC automatically transfers the CRB transfer information to and from the CRB register. See section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

18.2.7 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 4000 5400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable	0: Transfer information read is not skipped 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RRS bit (DTC Transfer Information Read Skip Enable)

The RRS bit enables skipping of transfer information reads when vector numbers match.

The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the value in the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

18.2.8 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 4000 5404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Bit name	Description	R/W
b31 to b0	DTC Vector Base Address	Specify the DTC vector base address. The lower 10 bits should be 0.	R/W

The DTCVBR register sets the base address for calculating the DTC vector table address, which can be set in the range of 0000 0000h to FFFF FFFFh (4 GB) in 1-KB units.

18.2.9 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 4000 540Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stopped 1: DTC module started.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCST bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When DTCST is set to 0, transfer requests are no longer accepted. If DTCST is set to 0 during a data transfer, the accepted transfer request is active until processing is complete.

DTCST must be set to 0 before transitioning to any of the following state or mode:

- Module-stop state
- Software Standby mode without Snooze mode transition
- Deep Software Standby mode.

For details on these transitions, see section 18.10, Module-Stop Function, and section 11, Low Power Modes.

18.2.10 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 4000 540Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	VECN[7:0]	DTC-Activating Vector Number Monitoring	These bits indicate the vector number for the activation source when a DTC transfer is in progress. The value is only valid if a DTC transfer is in progress (ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: DTC transfer operation is not in progress 1: DTC transfer operation is in progress.	R

VECN[7:0] bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the value of the ACT flag is 1, indicating a DTC transfer is in progress, and invalid if the value of the ACT flag is 0, indicating no DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

18.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output *n* number set in ICU.IELSRn is defined as the interrupt vector number, where n = 0 to 95. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number n is selected in ICU.IELSRn.IELS[8:0], as listed in Table 14.4, Event table, in section 14, Interrupt Controller Unit (ICU).

For activation by software, see section 19.2.2, Event Link Software Event Generation Register n (ELSEGRn) (n = 0, 1).

The interrupt vector number is equivalent to the DTC vector table number. After the DTC accepts an activation request, it does not accept another activation request until transfer for that single request is complete, regardless of the priority of the requests. When multiple activation requests are generated during a DMAC or DTC transfer, a highest priority request is accepted on completion of the transfer. When multiple activation requests are generated while the DTC module start bit (DTCST.DTCST) is 0, the DTC accepts the highest priority request when DTCST.DTCST is subsequently set to 1. The smaller interrupt vector number has higher priority.

The DTC performs the following operations at the start of a single data transfer or for a chain transfer, after the last of the consecutive transfers:

- On completion of a specified round of data transfer, the ICU.IELSRn.DTCE bit is set to 0, and an interrupt request is sent to the CPU
- If the MRB.DISEL bit is 1, an interrupt request is sent to the CPU on completion of a data transfer
- For other transfers, the ICU.IELSRn.IR bit of the activation source is set to 0 at the start of the data transfer.

18.3.1 Allocating Transfer Information and the DTC Vector Table

The DTC reads the start address of the transfer information associated with each activation source from the vector table and reads the transfer information starting at that address.

The vector table must be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC Vector Base Register (DTCVBR) to set the base address of the DTC vector table. Transfer information is allocated in the SRAM area. In the SRAM area, the start address of the transfer information (n) with vector number n must be 4n added to the base address in the vector table.

Figure 18.2 shows the relationship between the DTC vector table and transfer information. Figure 18.3 shows the allocation of transfer information in the SRAM area.

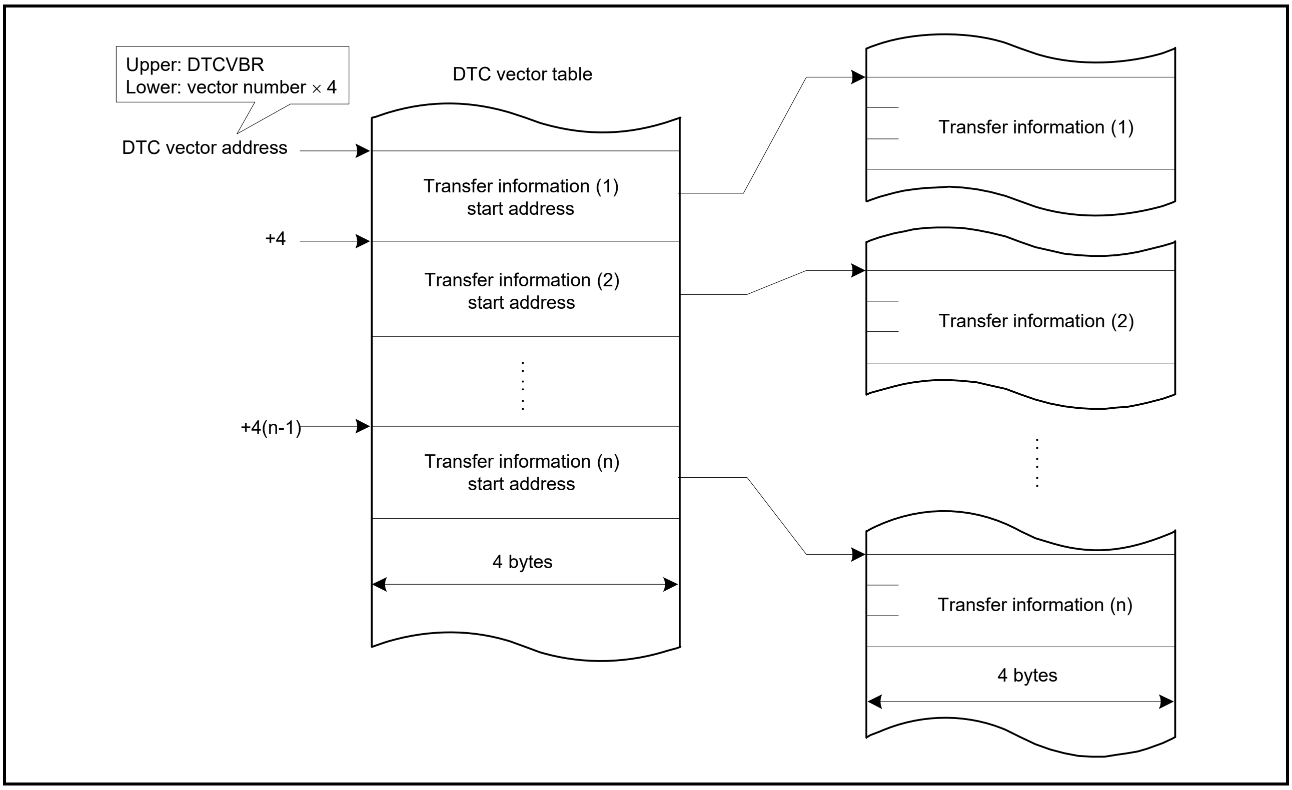


Figure 18.2 DTC vector table and transfer information

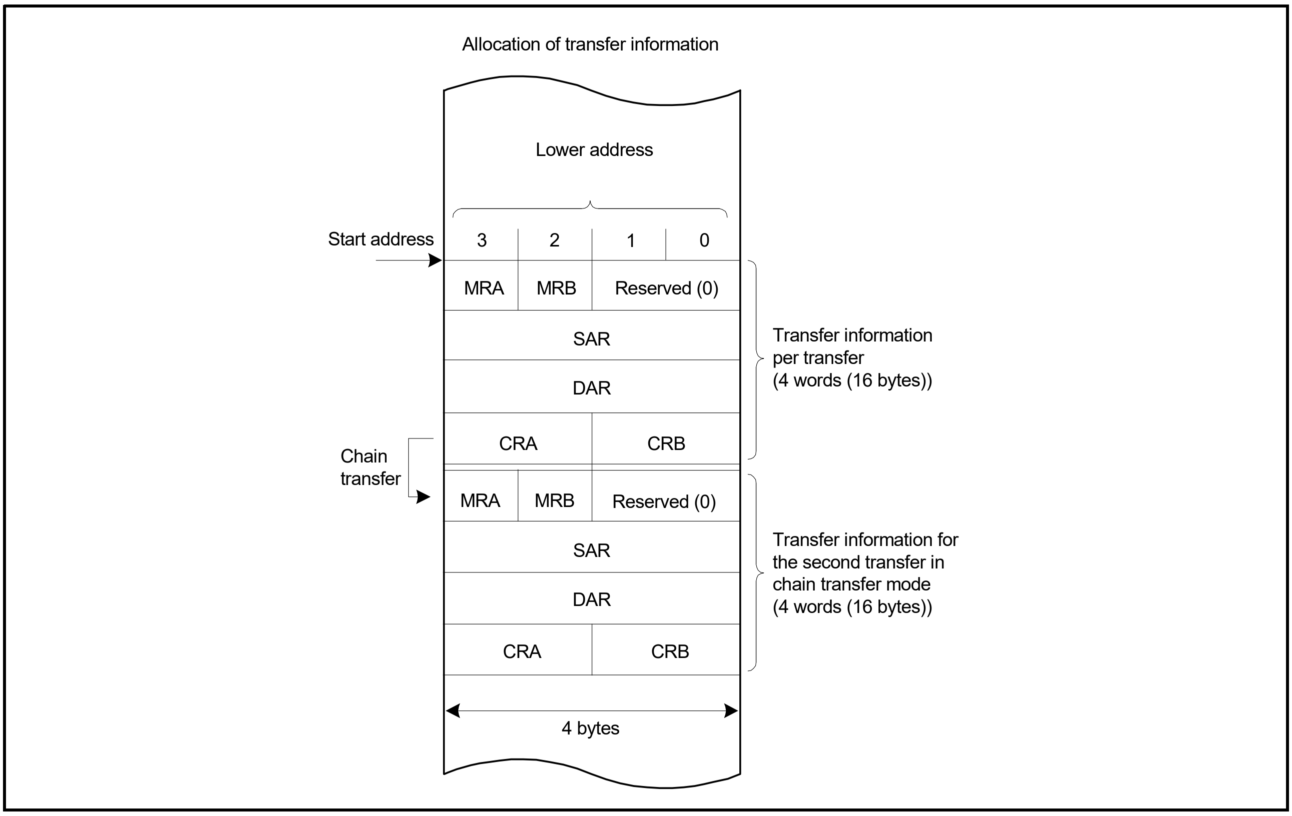


Figure 18.3 Allocation of transfer information in the SRAM area

18.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC then reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

The transfer modes include:

- Normal transfer mode
- Repeat transfer mode
- Block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values in these registers are incremented, decremented, or address-fixed independently after the data transfer.

Table 18.2 describes the DTC transfer modes.

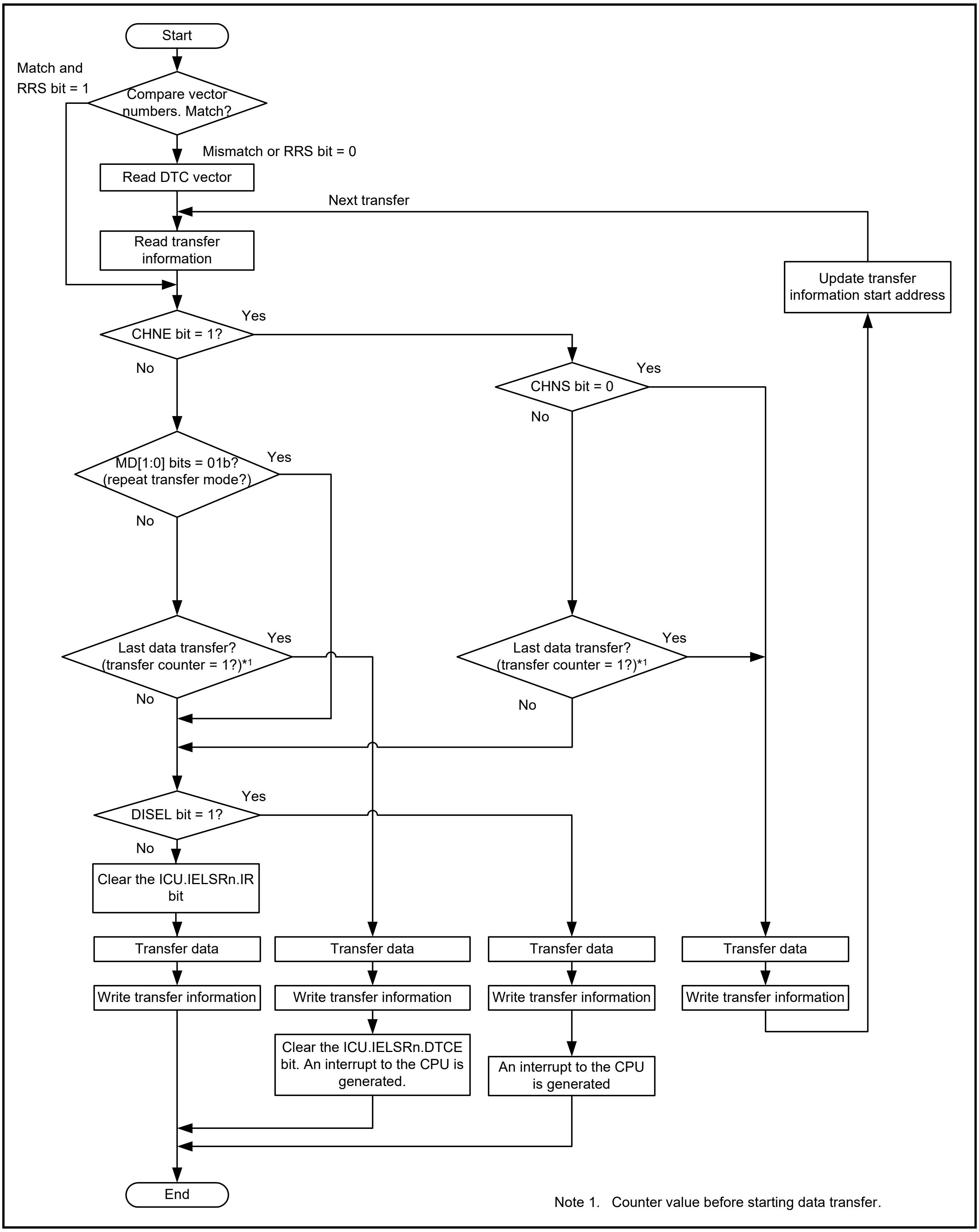
Table 18.2 DTC transfer modes

Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Normal transfer mode	1 byte (8 bits), 1 halfword (16 bits), or 1 word (32 bits)	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 65,536
Repeat transfer mode[*1]	1 byte (8 bits), 1 halfword (16 bits), or 1 word (32 bits)	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 256[*3]
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes))	Incremented or decremented by 1, 2, or 4 or address fixed	1 to 65,536

Note 1. Set the transfer source or destination as the repeat area.
Note 2. Set the transfer source or destination as the block area.
Note 3. After a data transfer of the specified count, the initial state is restored and operation restarts.

Setting the MRB.CHNE bit to 1 allows multiple transfers or a chain transfer on a single activation source. It also enables a chain transfer when the specified data transfer is complete.

Figure 18.4 shows the operation flow of the DTC. Table 18.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.



Note 1. Counter value before starting data transfer.

Figure 18.4 DTC operation flow

Table 18.3 Chain transfer conditions

First transfer				Second transfer*3				
CHNE bit	**CHNS bit**	**DISEL bit**	**Transfer counter*1,*2**	**CHNE bit**	**CHNS bit**	**DISEL bit**	**Transfer counter*1,*2**	**DTC transfer**
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE bit = 1 is omitted.

18.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared to the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer is a chain transfer, the vector address and transfer information are read. Additionally, when the transfer counter (CRA register) becomes 0 during the previous normal transfer, or when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 18.12 shows an example of a transfer information read skip.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. The stored vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

18.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to address fixed, a part of the transfer information is not written back. Table 18.4 lists the transfer information write-back skip conditions and associated registers. The CRA and CRB registers are written back, and the write-back of the MRA and MRB registers is skipped.

Table 18.4 Transfer information write-back skip conditions and applicable registers

MRA.SM[1:0] bits		MRB.DM[1:0] bits		SAR register	DAR register
b3	b2	b3	b2		
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

18.4.3 Normal Transfer Mode

Normal transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. The transfer count can be set from 1 to 65,536. Transfer source and destination addresses can be independently set to increment, decrement, or remain fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

Table 18.5 lists register functions in normal transfer mode, and Figure 18.5 shows the memory map of normal transfer mode.

Table 18.5 Register functions in normal transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	Increment, decrement, or fixed*[1]
DAR	Transfer destination address	Increment, decrement, or fixed*[1]
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

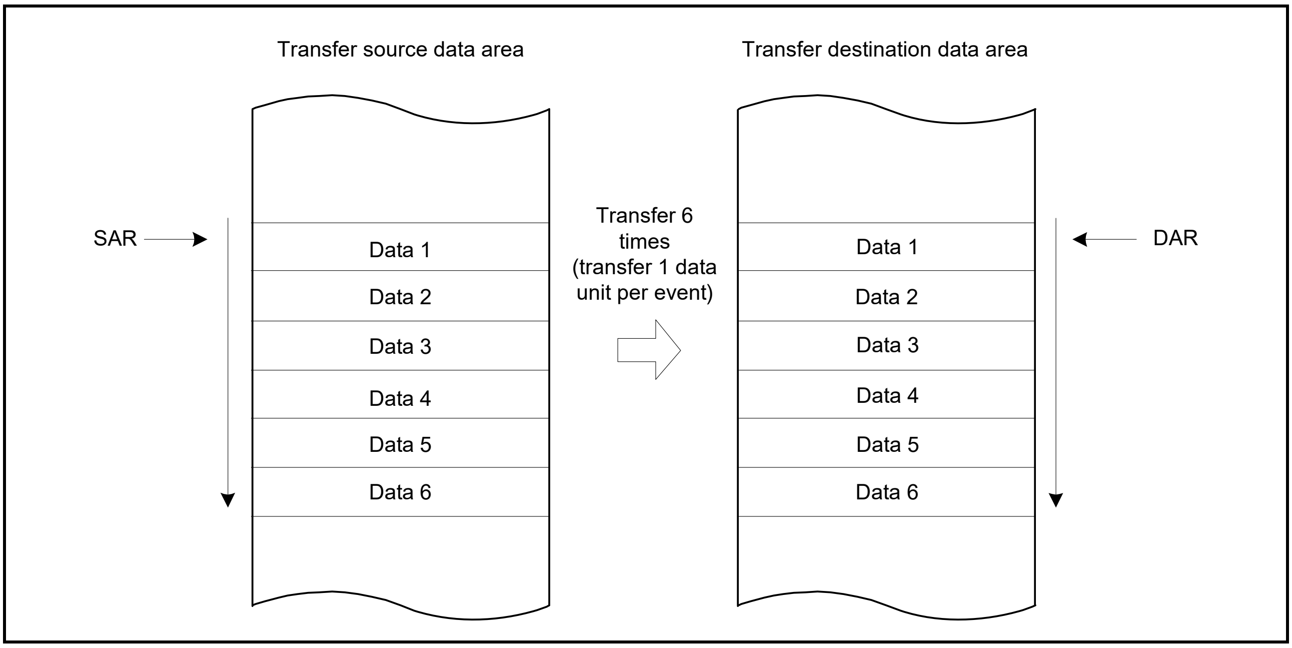


Figure 18.5 **Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0006h)**

18.4.4 Repeat Transfer Mode

Repeat transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. Specify either transfer source or transfer destination for the repeat area in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified-count transfer is complete, the initial value of the address register specified in the repeat area is restored, the initial value of the transfer counter is restored, and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL decrements to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. As a result, the transfer counter does not become 00h, which disables interrupt requests to the CPU when the MRB.DISEL bit is set to 0. An interrupt request to the CPU is generated when the specified data transfer is complete.

Table 18.6 lists the register functions in repeat transfer mode, and Figure 18.6 shows the memory map of repeat transfer mode.

Table 18.6 **Register functions in repeat transfer mode**

Register	Description	Value written back by writing transfer information	
		When CRAL is not 1	**When CRAL is 1**
SAR	Transfer source address	Increment, decrement, or fixed*1	• When the MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When the MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	Increment, decrement, or fixed*1	• When the MRB.DTS bit is 0 DAR register initial value • When the MRB.DTS bit is 1 Increment, decrement, or fixed.*1
CRAH	Holds transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

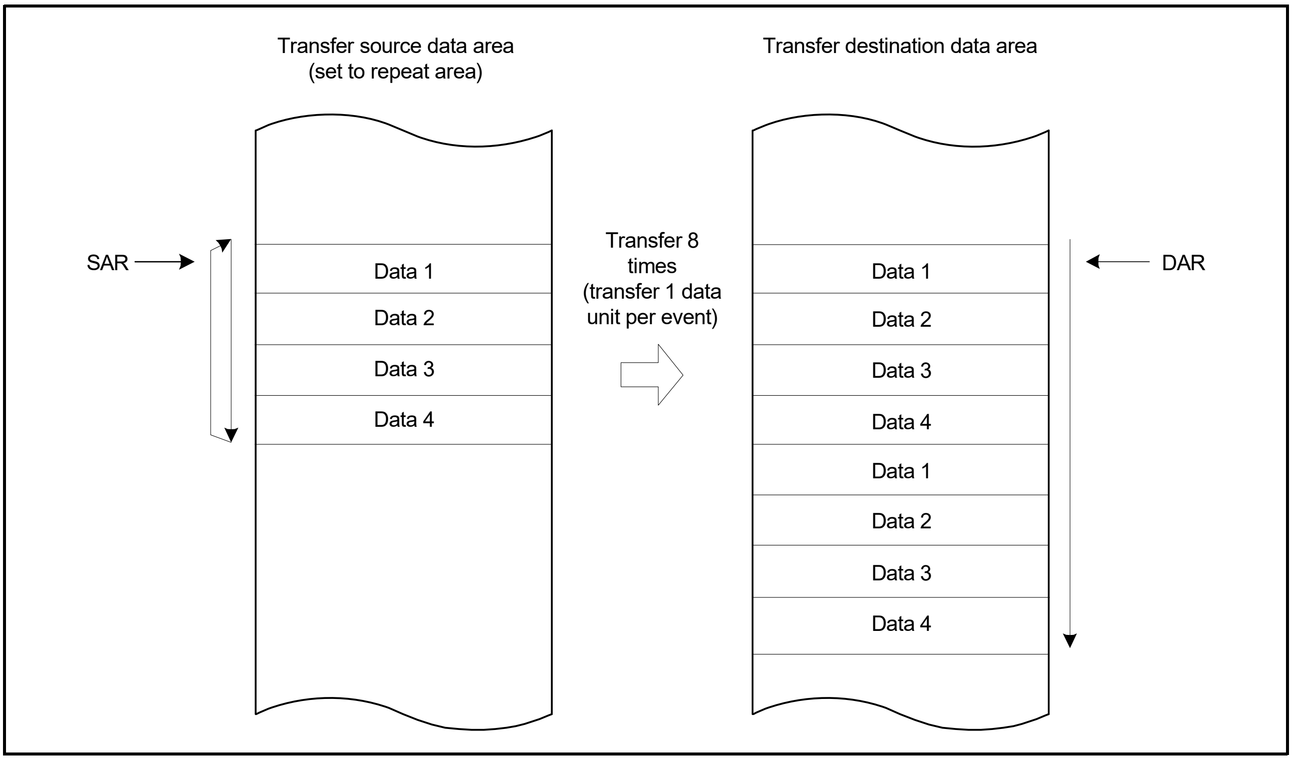


Figure 18.6 Memory map of repeat transfer mode when the transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 04h)

18.4.5 Block Transfer Mode

Block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit = 1 or the DAR register when the DTS bit = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

The transfer count (block count) can be set from 1 to 65,536. This mode enables an interrupt request to the CPU to be generated at the end of the specified-count block transfer.

Table 18.7 lists register functions in block transfer mode, and Figure 18.7 shows the memory map of block transfer mode.

Table 18.7 Register functions in block transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	• When MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	• When MRB.DTS bit is 0 DAR register initial value • When MRB.DTS bit is 1 Increment, decrement, or fixed.*1
CRAH	Holds the block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

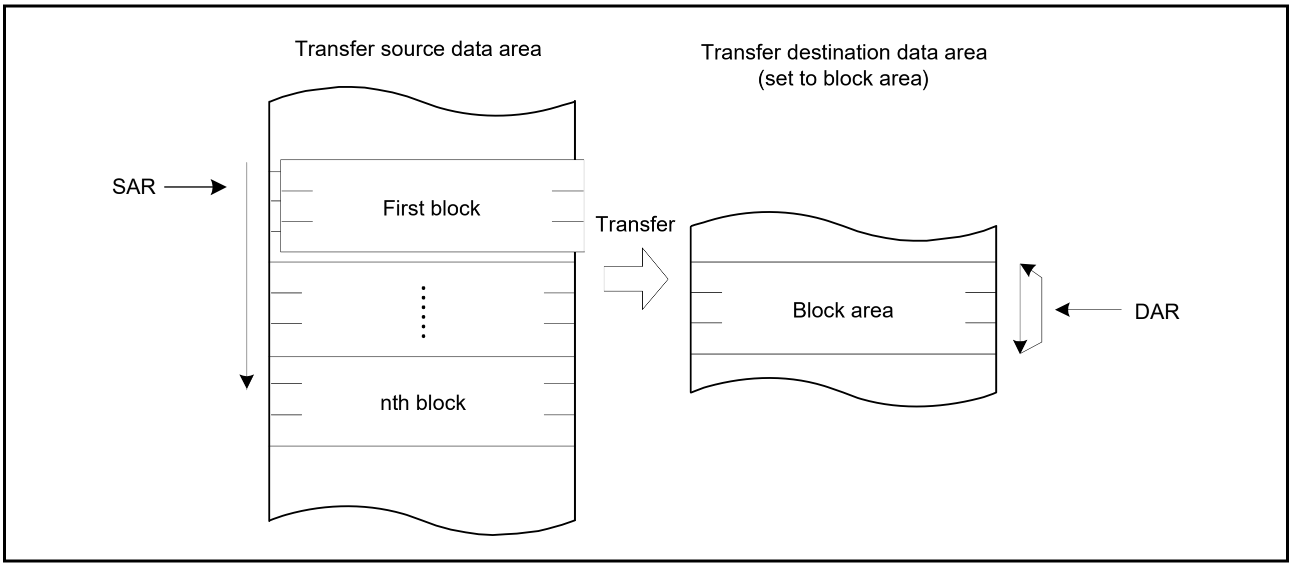


Figure 18.7 Memory map of block transfer mode

18.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR bit of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define the data transfer. Figure 18.8 shows a chain transfer operation.

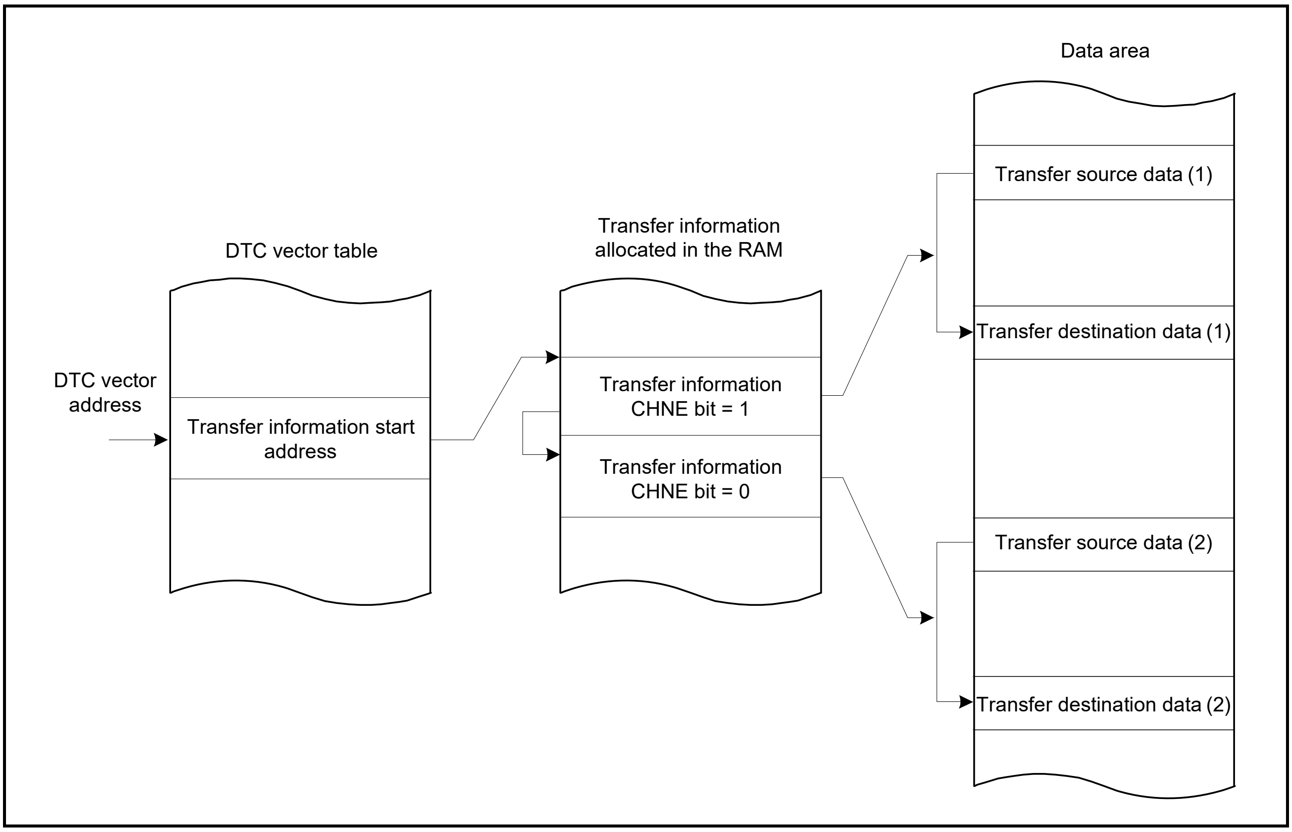


Figure 18.8 Chain transfer operation

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of the specified data transfer. In repeat transfer mode, chain transfer is performed after completion of the specified data transfer. For details on chain transfer conditions, see Table 18.3, Chain transfer conditions.

18.4.7 Operation Timing

Figure 18.9 to Figure 18.12 are timing diagrams that show the minimum number of execution cycles.

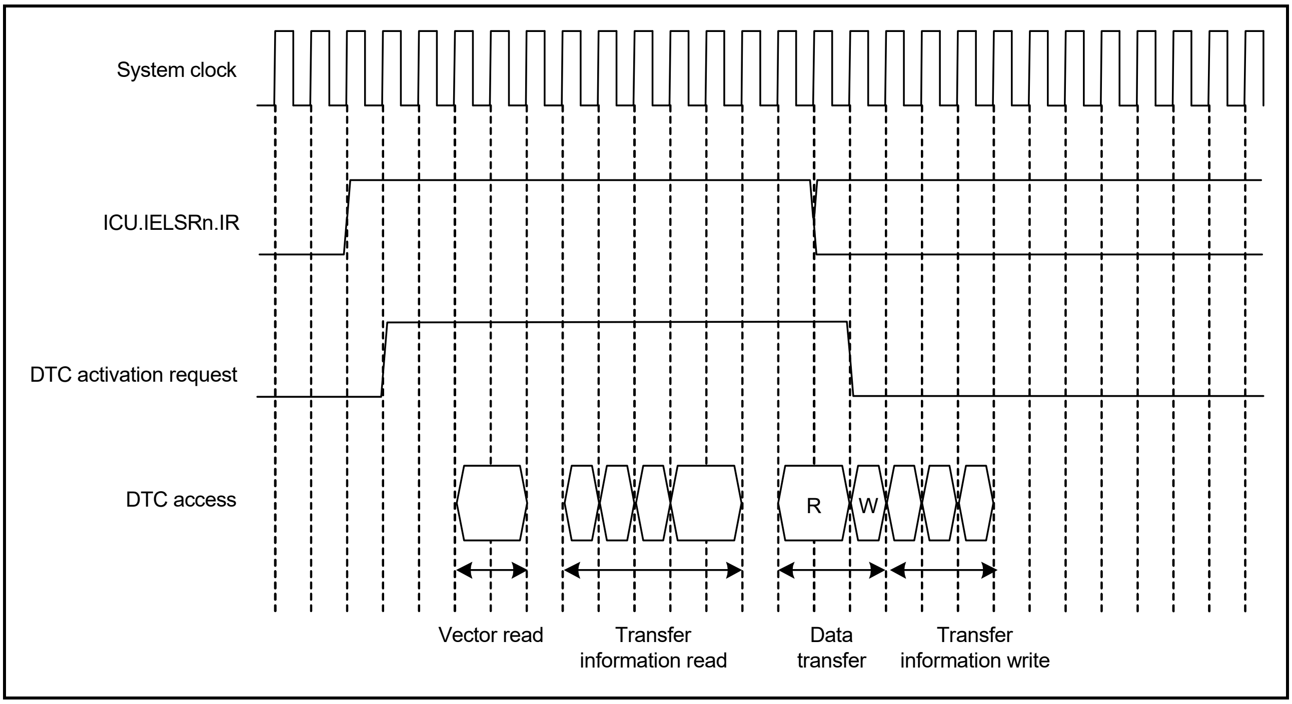


Figure 18.9 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

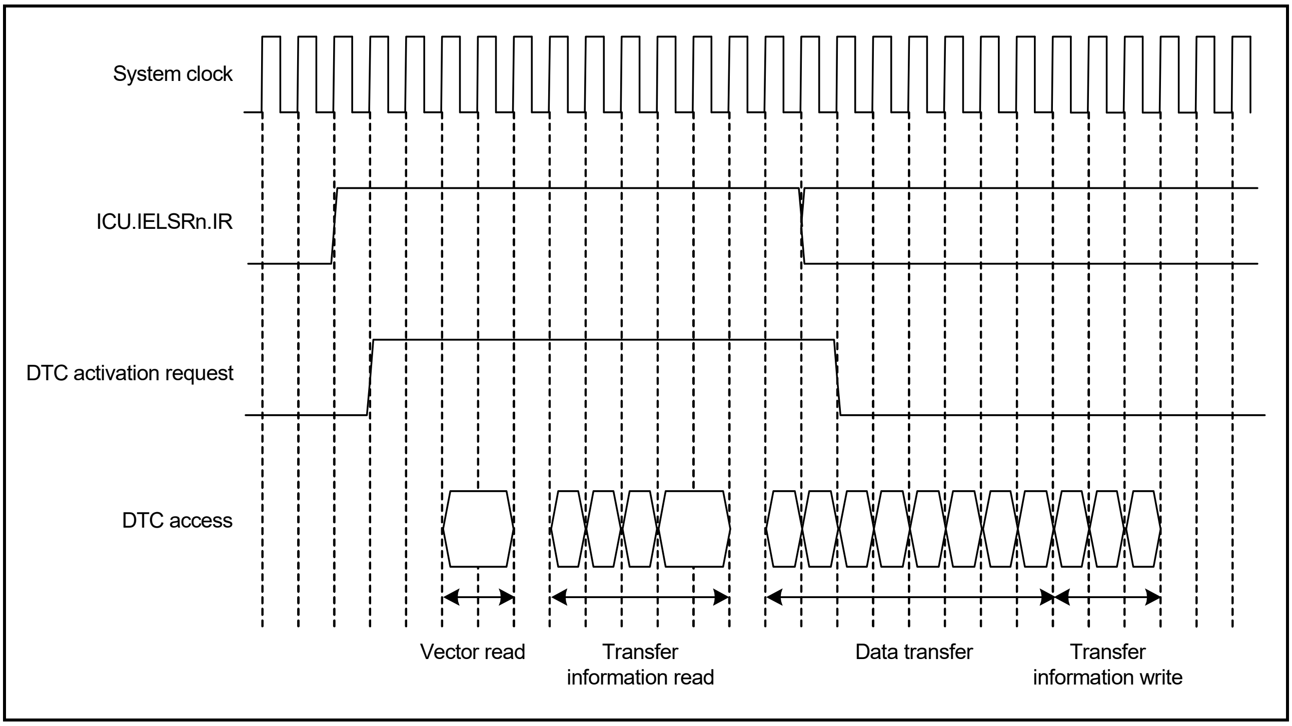


Figure 18.10 Example 2 of DTC operation timing in block transfer mode when the block size = 4

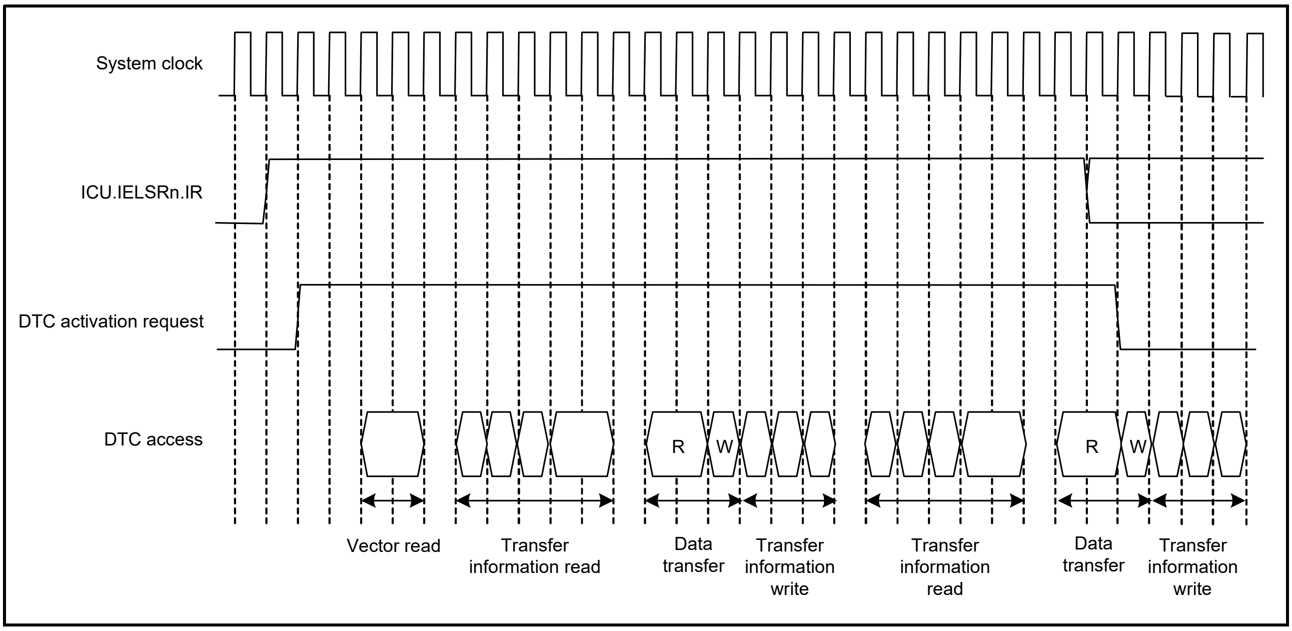
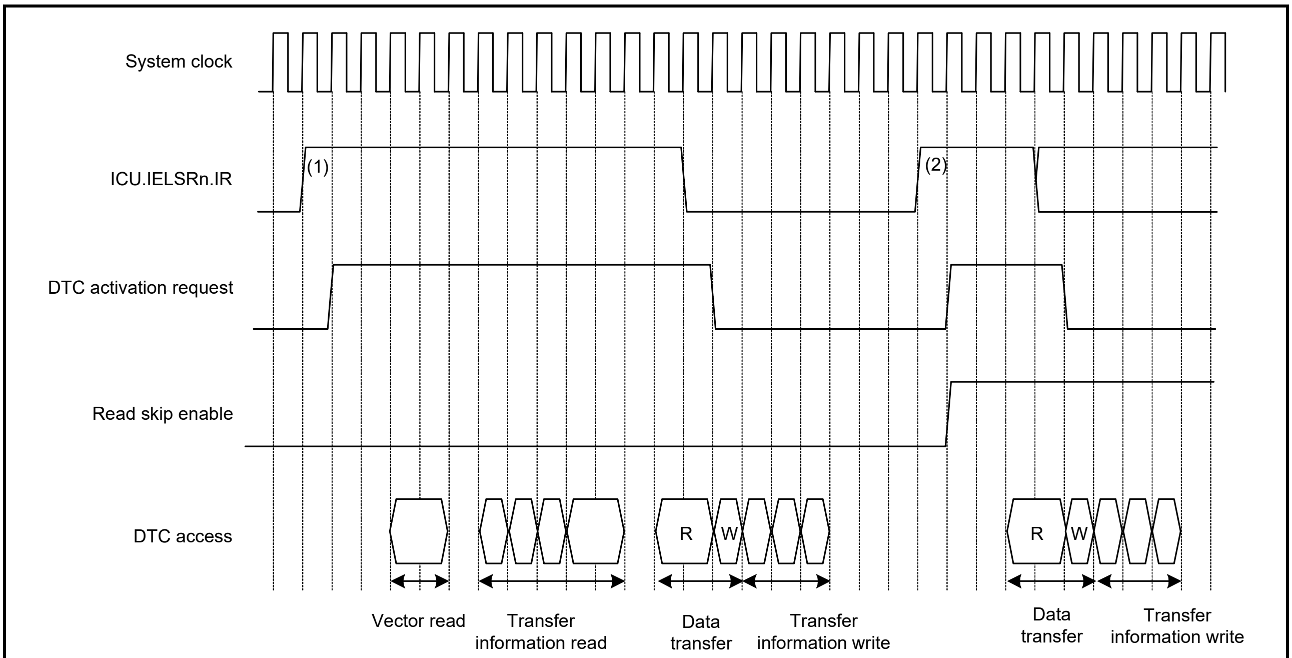


Figure 18.11 Example 3 of DTC operation timing for chain transfer



Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS bit = 1, the transfer information read for request (2) is skipped.

Figure 18.12 Example of operation when a transfer information read is skipped, with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

18.4.8 Execution Cycles of DTC

Table 18.8 lists the execution cycles of single data transfer of the DTC.

For the order of the execution states, see section 18.4.7, Operation Timing.

Table 18.8 Execution cycles of DTC

Transfer mode	Vector read		Transfer information read		Transfer information write			Data transfer		Internal operation	
								Read	Write		
Normal	Cv + Cs1 + 1	0*1	4 × (Ci + Cs1) + 1	0*1	3 × (Ci + Cs1) + 1*2	2 × (Ci + Cs1) + 1*3	(Ci + Cs1)*4	Cr + Cs2 + 1	Cw + Cs2 + 1	2	0*1
Repeat								Cr + Cs2 + 1	Cw + Cs2 + 1		
Block*5								P × (Cr + Cs2)	P × (Cw + Cs2)		

Note 1. When transfer information read is skipped.
Note 2. When neither SAR nor DAR is set to address-fixed mode.
Note 3. When SAR or DAR is set to address-fixed mode.
Note 4. When SAR and DAR are set to address-fixed mode.
Note 5. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

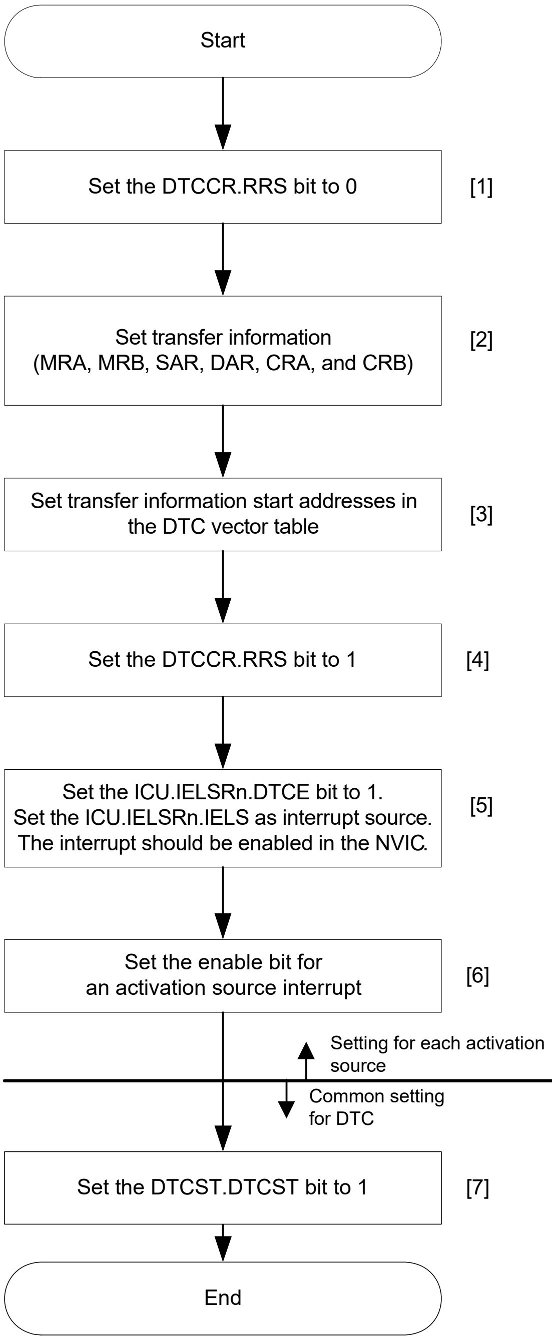
P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
Cs1: When accessing SRAMHS: 2 cycles.
When accessing elsewhere: 0 cycles.
When a slave bus changes by a read/write data transfer, add 1 more cycle.
Cs2: When accessing SRAMHS and peripheral modules related to system control: 2 cycles.
When accessing elsewhere: 0 cycle.
When a slave bus change by a read/write data transfer, add 1 more cycle.
The unit is system clocks (ICLK) + 1 in the Vector read, Transfer information read, and Data transfer read columns and 2 in the Internal operation column.
Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, see section 53, SRAM, section 55, Flash Memory, and section 15.2.3, External Bus.
The frequency ratio of the system clock and peripheral clock is also taken into consideration.
The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts.
This table does not include the time until DTC data transfer starts after the DTC activation source becomes active.

18.4.9 DTC Bus Mastership Release Timing

The DTC does not release bus mastership during transfer information reads. Before the transfer information is read or written, the bus is arbitrated according to the priority determined by the bus master arbitrator. For bus arbitration, see section 15, Buses.

18.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Figure 18.13 shows the procedure for setting the DTC.



Set the ICU.IELSRn.IELS[8:0] bit to 0. Disable the interrupt in the NVIC and provide the following settings:

[1] Set the DTCCR.RRS bit to 0 to reset the transfer information read skip flag. After that, transfer information read is not skipped while the DTC is activated. Specify this setting when transfer information is updated.

[2] Allocate the transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see section 18.2, Register Descriptions. To allocate transfer information, see section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

[3] Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see section 18.3.1, Allocating Transfer Information and the DTC Vector Table.

[4] Setting the DTCCR.RRS bit to 1 enables skipping of the second and the subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.

[5] Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS as the interrupt sources that trigger the DTC. The interrupts must be enabled in NVIC. See the Table 14.4, Event table.

[6] Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.

[7] Set the DTC module start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for the activation sources is not complete.

Figure 18.13 DTC setting procedure

18.6 Examples of DTC Usage

18.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC vector table setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, reception stops. To manage this, use settings that allow the CPU to accept receive error interrupts.

(5) DTC transfer

Every time a reception of 1 byte by the SCI is complete, an RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt handling

After 128 rounds of data transfer are complete and the value in the CRA register becomes 0, an RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

18.6.2 Chain Transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the General PWM Timer (GPT). You can use chain transfer to transfer PWM timer compare data and change the period of the PWM timer for GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPT32m.GTCCRC register. For the second transfer, normal transfer mode is specified for transfer to the GPT32m.GTCCRE register. For the third transfer, normal transfer mode is specified for transfer to the GPT32m.GTPBR register. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of transfers are restricted to the third of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the counter overflow interrupt with a GPT32EH0.GTPR register as an activating source for the DTC.

(1) First transfer information settings

Set up transfer to the GPT32EH0.GTCCRC register:

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT32EH0.GTCCRC register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(2) Second transfer information settings

Set up transfer to the GPT32EH0.GTCCRE register:

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).

4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT32EH0.GTCCRE register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(3) Third transfer information settings

Set up transfer to the GPT32EH0.GTPBR registers:

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value.
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT32EH0.GTPBR register.
6. Set the CRA register to the size of the data table. The CRB register can be set to any value.

(4) Transfer information assignment

Place the transfer information for use in the transfer to GPT32EH0.GTPBR immediately after the transfer control information for use in the GPT32EH0.GTCCRC and GPT32EH0.GTCCRE registers.

(5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the GPT32EH0.GTCCRC and GPT32EH0.GTCCRE registers starts.

(6) ICU settings and DTC module activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT32EH0 counter overflow interrupt.
2. Set ICU.IELSRn.IELS[8:0] to 182 (B6h) for the GPT32EH0 counter overflow.
3. Set the DTCST.DTCST bit to 1.

(7) GPT settings

1. Set the GPT32EH0.GTIOR register so that the GTCCRA and GTCCRB registers operate as output compare registers.
2. Set the default PWM timer compare values in the GPT32EH0.GTCCRA and GPT32EH0.GTCCRB registers and the next PWM timer compare values in the GPT32EH0.GTCCRC and GPT32EH0.GTCCRE registers.
3. Set the default PWM timer period values in the GPT32EH0.GTPR register and the next PWM timer period values in the GPT32EH0.GTPBR register.
4. Set 1 to the output bit in PmnPFS.PDR, and set 00011b to the peripheral select bits in PmnPFS.PSEL[4:0].

(8) GPT activation

Set the GPT32EH0.GTSTR.CSTRT bit to 1 to start the GPT32EH0.GTCNT counter.

(9) DTC transfer

Each time a GPT32EH0 counter overflow is generated with the GPT32EH0.GTPR register, the next PWM timer compare values are transferred to the GPT32EH0.GTCCRC and GPT32EH0.GTCCRE registers. The setting for the next PWM timer period is transferred to the GPT32EH0.GTPBR register.

(10) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for GPT transfer becomes 0, a GPT counter overflow interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

18.6.3 Chain Transfer When Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Chain transfer enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 128-KB input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 18.14 shows a chain transfer when the counter = 0.

1. Set the normal transfer mode to input data for the first data transfer. Set the following:
   a. Transfer source address = fixed.
   b. CRA register = 0000h (65,536) times.
   c. MRB.CHNE bit = 1 (chain transfer is enabled).
   d. MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0).
   e. MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
2. Prepare the upper 8-bit address of the start address at every 65,536 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 20 0000h to 21 FFFFh, prepare 21h and 20h.
3. For the second data transfer:
   a. Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
   b. Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
   c. Set the MRB.CHNE bit = 0 (chain transfer is disabled).
   d. Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   e. When setting the input buffer to 20 0000h to 21 FFFFh, also set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 21h. The lower 16 bits of the transfer destination address and the transfer counter of the first data transfer become 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 times as specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 20h. The lower 16 bits of the transfer destination address and the transfer counter of the first data transfer become 0000h.
6. Steps 4 and 5 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

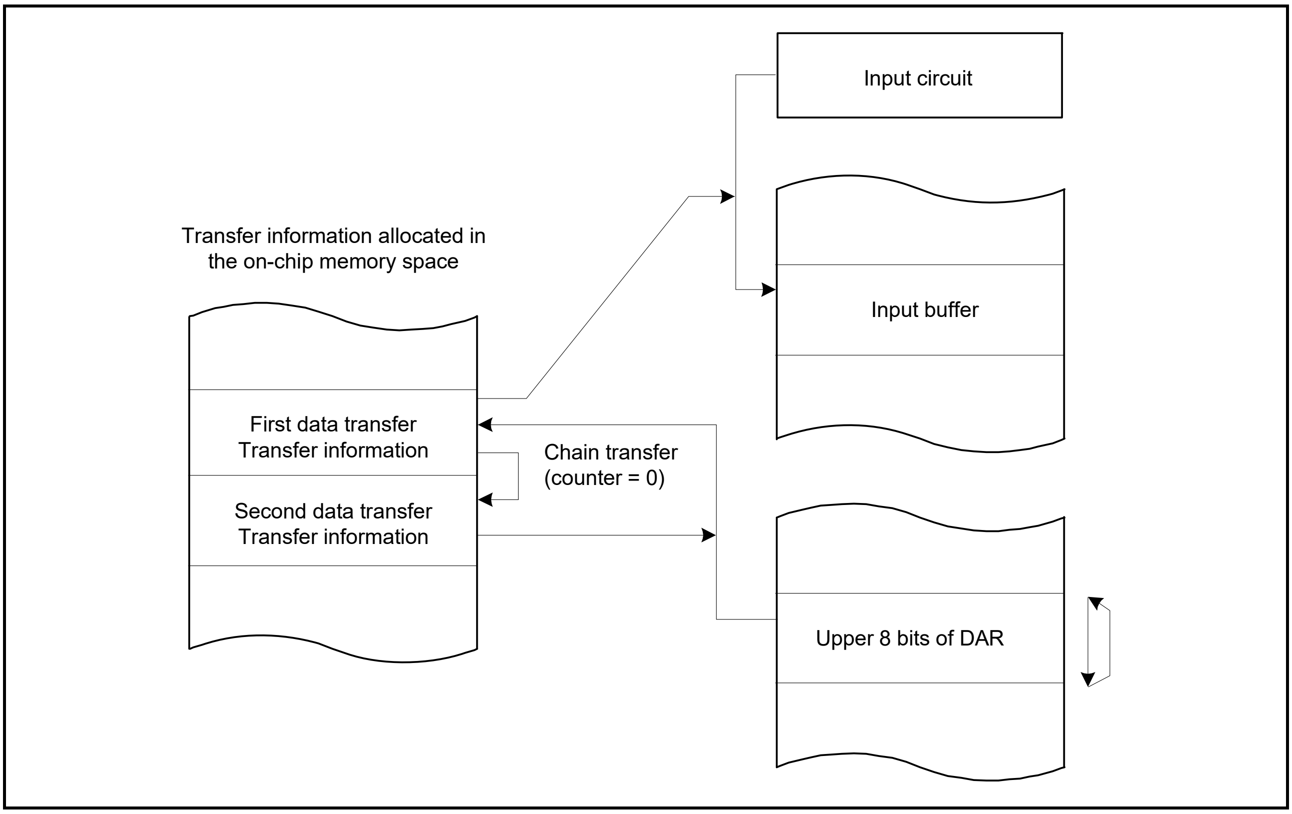


Figure 18.14 Chain transfer when counter = 0

18.7 Interrupt Sources

When the DTC finishes data transfer of the specified count or when data transfer with the MRB.DISEL bit set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Interrupts to the CPU are controlled according to the settings in the NVIC and ICU.IELSRn.IELS[8:0]. See section 14, Interrupt Controller Unit (ICU).

The DTC prioritizes activation sources by granting the smaller interrupt vector numbers higher priority. The priority of interrupts to the CPU is determined by the NVIC priority.

18.8 Event Link

The DTC is capable of producing an event link request on completion of one transfer request. When the destination for the transfer is an external bus, the event link request is issued after completion of writing to the write buffer rather than after completion of writing to the actual transfer destination.

18.9 Snooze Control Interface

To return to Software Standby mode from Snooze mode through the DTC, set SYSTEM.SNZEDCR.DTCZRED or SYSTEM.SNZEDCR.DTCNZRED to 1. See section 11.8.3, Return to Software Standby Mode.

SYSTEM.SNZEDCR.DTCZRED enables or disables a Snooze end request on completion of the last DTC transmission, detected on DTC transmission completion when CRA and CRB are 0.

SYSTEM.SNZEDCR.DTCNZRED enables or disables a Snooze end request on a not last DTC transmission completion, detected on DTC transmission completion when CRA and CRB are not 0.

18.10 Module-Stop Function

Before transitioning to the module-stop function, Software Standby mode without a Snooze mode transition, or Deep Software Standby mode, set the DTCST.DTCST bit to 0, then perform the operations described in the following sections. The DTC is available in Snooze mode by setting LPW.SNZCR.SNZDTCEN to 1. See section 11, Low Power Modes.

(1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DTC. If the DTC transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA22 bit, the transition to the module-stop state proceeds after DTC transfer ends. When the MSTPCRA.MSTPA22 bit is 1, accessing the DTC registers is prohibited.

Writing 0 to the MSTPCRA.MSTPA22 bit releases the DTC from the module-stop state.

(2) Software Standby and Deep Software Standby modes

Use the settings described in section 11.7.1, Transition to Software Standby Mode, or section 11.9.1, Transition to Deep Software Standby Mode.

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode or Deep Software Standby mode follows the completion of the DTC transfer.

When the Snooze control circuit receives a Snooze request in Software Standby mode, the MCU transfers to Snooze mode. See section 11.8.1, Transition to Snooze Mode. DTC operation in Snooze mode can be selected in the SYSTEM.SNZCR.SNZDTCEN bit. If DTC operation is enabled in Snooze mode, transitioning to Software Standby mode, set the DTCST.DTCST bit to 1. To return to Software Standby mode through the DTC, set SYSTEM.SNZEDCR.DTCZRED or SYSTEM.SNZEDCR.DTCNZRED to 1. See section 11.8.3, Return to Software Standby Mode. The DTC activation request from the ICU is stopped during Software Standby mode but not during Snooze mode.

(3) Notes on the module-stop function

For the WFI instruction and the register setting procedure, see section 11, Low Power Modes.

To perform a DTC transfer after returning from a low power mode without Snooze mode transition, set the DTCST.DTCST bit to 1 again.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in section 14.4.2, Selecting Interrupt Request Destinations, then execute a WFI instruction. If DTC operation is enabled in Snooze mode, do not use the module-stop function of the DTC.

18.11 Usage Notes

18.11.1 Transfer information Start Address

You must set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

19. Event Link Controller (ELC)

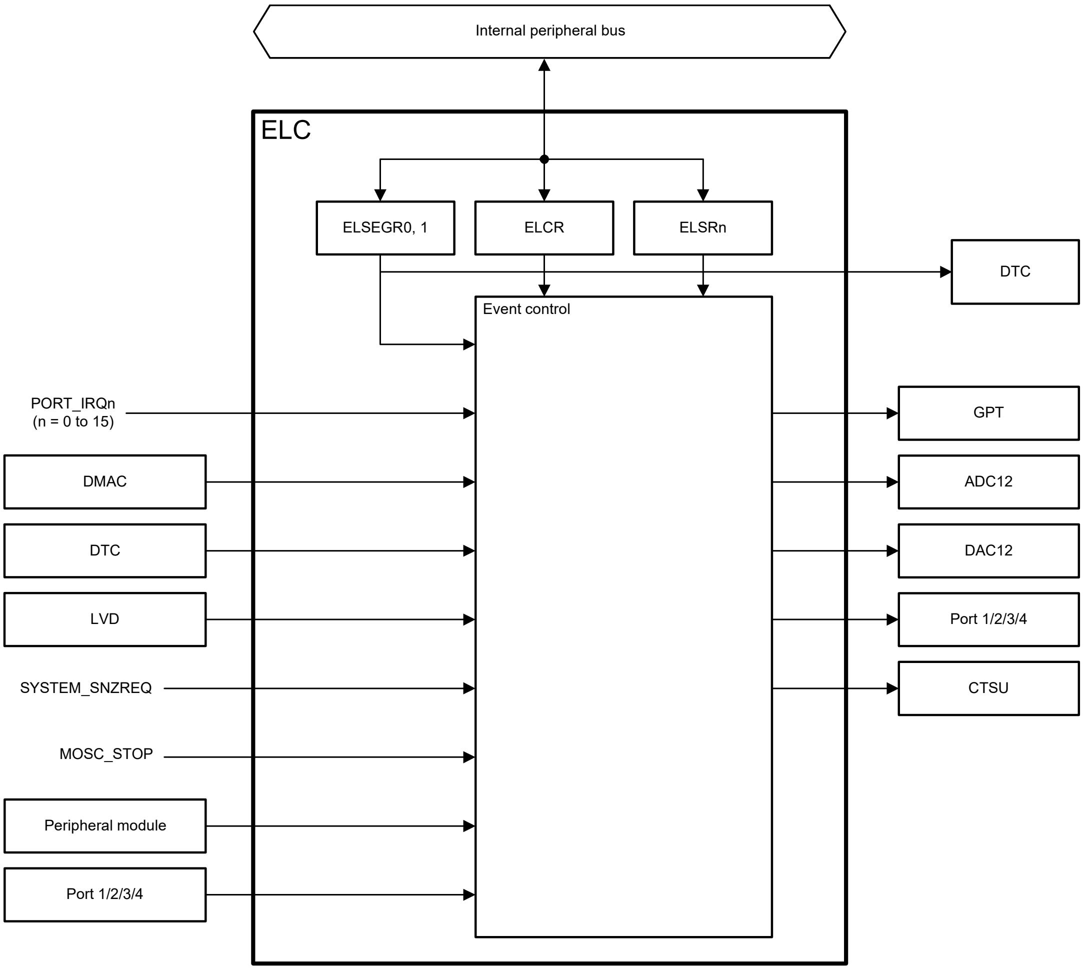
19.1 Overview

The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between modules without CPU intervention.

Table 19.1 lists the ELC specifications and Figure 19.1 shows a block diagram.

Table 19.1 ELC specifications

Parameter	Specifications
Event link function	270 types of event signals can be directly connected to modules.The ELC can generate an ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set to reduce power consumption



ELSEGR0, 1: Event Link Software Event Generation Register
ELCR: Event Link Control Register
ELSRn: Event Link Setting Register n (n = 0 to 18)

Figure 19.1 ELC block diagram

19.2 Register Descriptions

19.2.1 Event Link Controller Register (ELCR)

Address(es): ELC.ELCR 4004 1000h

	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ELCON	All Event Link Enable	0: ELC function disabled 1: ELC function enabled.	R/W

The ELCR register controls the ELC operation.

19.2.2 Event Link Software Event Generation Register n (ELSEGRn) (n = 0, 1)

Address(es): ELC.ELSEGR0 4004 1002h, ELC.ELSEGR1 4004 1004h

	b7	b6	b5	b4	b3	b2	b1	b0
	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SEG	Software Event Generation	0: Normal operation 1: Software event is generated.	W
b5 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	WE	SEG Bit Write Enable	0: Writes to SEG bit disabled 1: Writes to SEG bit enabled.	R/W
b7	WI	ELSEGR Register Write Disable	0: Writes to ELSEGR register enabled 1: Writes to ELSEGR register disabled.	W

SEG bit (Software Event Generation)

When 1 is written to the SEG bit while the WE bit is 1, a software event is generated. This bit is read as 0. Even when 1 is written to this bit, data is not stored. The WE bit must be set to 1 before writing to this bit.

A software event can trigger a linked DTC event.

WE bit (SEG Bit Write Enable)

The SEG bit can only be written to when the WE bit is 1. Clear the WI bit to 0 before writing to this bit.

[Setting condition]

- If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

- If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

WI bit (ELSEGR Register Write Disable)

The ELSEGR register can only be written to when the write value to the WI bit is 0. This bit is read as 1. Before setting the WE or SEG bit, the WI bit must be set to 0.

19.2.3 Event Link Setting Register n (ELSRn) (n = 0 to 18)

Address(es): ELC.ELSR0 4004 1010h, ELC.ELSR1 4004 1014h, ELC.ELSR2 4004 1018h, ELC.ELSR3 4004 101Ch, ELC.ELSR4 4004 1020h, ELC.ELSR5 4004 1024h, ELC.ELSR6 4004 1028h, ELC.ELSR7 4004 102Ch, ELC.ELSR8 4004 1030h, ELC.ELSR9 4004 1034h, ELC.ELSR10 4004 1038h, ELC.ELSR11 4004 103Ch, ELC.ELSR12 4004 1040h, ELC.ELSR13 4004 1044h, ELC.ELSR14 4004 1048h, ELC.ELSR15 4004 104Ch, ELC.ELSR16 4004 1050h, ELC.ELSR17 4004 1054h, ELC.ELSR18 4004 1058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ELS[8:0]	Event Link Select	b8 b0 000000000: Event output disabled for the associated peripheral module 000000001 to 111000101b: Number setting for the event signal to be linked. Other settings are prohibited.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 19.2 shows the association between the ELSRn registers and the peripheral modules. Table 19.3 shows the association between the event signal names set in the ELSRn register and the signal numbers.

Table 19.2 Association between the ELSRn registers and peripheral functions

Register name	Peripheral function (module)	Event name
ELSR0	GPT (A)	ELC_GPTA
ELSR1	GPT (B)	ELC_GPTB
ELSR2	GPT (C)	ELC_GPTC
ELSR3	GPT (D)	ELC_GPTD
ELSR4	GPT (E)	ELC_GPTE
ELSR5	GPT (F)	ELC_GPTF
ELSR6	GPT (G)	ELC_GPTG
ELSR7	GPT (H)	ELC_GPTH
ELSR8	ADC12A0	ELC_AD00
ELSR9	ADC12B0	ELC_AD01
ELSR10	ADC12A1	ELC_AD10
ELSR11	ADC12B1	ELC_AD11
ELSR12	DAC12 channel 0	ELC_DA0
ELSR13	DAC12 channel 1	ELC_DA1
ELSR14	PORT 1	ELC_PORT1
ELSR15	PORT 2	ELC_PORT2
ELSR16	PORT 3	ELC_PORT3
ELSR17	PORT 4	ELC_PORT4
ELSR18	CTSU	ELC_CTSU

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (1 of 7)

Event number	Interrupt request source	Name	Description
001h	Port	PORT_IRQ0*1	External pin interrupt 0
002h		PORT_IRQ1*1	External pin interrupt 1
003h		PORT_IRQ2*1	External pin interrupt 2
004h		PORT_IRQ3*1	External pin interrupt 3
005h		PORT_IRQ4*1	External pin interrupt 4
006h		PORT_IRQ5*1	External pin interrupt 5
007h		PORT_IRQ6*1	External pin interrupt 6
008h		PORT_IRQ7*1	External pin interrupt 7
009h		PORT_IRQ8*1	External pin interrupt 8
00Ah		PORT_IRQ9*1	External pin interrupt 9
00Bh		PORT_IRQ10*1	External pin interrupt 10
00Ch		PORT_IRQ11*1	External pin interrupt 11
00Dh		PORT_IRQ12*1	External pin interrupt 12
00Eh		PORT_IRQ13*1	External pin interrupt 13
00Fh		PORT_IRQ14*1	External pin interrupt 14
010h		PORT_IRQ15*1	External pin interrupt 15
020h	DMAC0	DMAC0_INT	DMAC transfer end 0
021h	DMAC1	DMAC1_INT	DMAC transfer end 1
022h	DMAC2	DMAC2_INT	DMAC transfer end 2
023h	DMAC3	DMAC3_INT	DMAC transfer end 3
024h	DMAC4	DMAC4_INT	DMAC transfer end 4
025h	DMAC5	DMAC5_INT	DMAC transfer end 5
026h	DMAC6	DMAC6_INT	DMAC transfer end 6
027h	DMAC7	DMAC7_INT	DMAC transfer end 7
02Ah	DTC	DTC_DTCEND*3	DTC transfer end
038h	LVD	LVD_LVD1	Voltage monitor 1 interrupt
039h		LVD_LVD2	Voltage monitor 2 interrupt
03Bh	MOSC	MOSC_STOP	Main clock oscillation stop
03Ch	Low-power mode	SYSTEM_SNZREQ*2, *3	Snooze entry
040h	AGT0	AGT0_AGTI	AGT interrupt
041h		AGT0_AGTCMAI	Compare match A
042h		AGT0_AGTCMBI	Compare match B
043h	AGT1	AGT1_AGTI	AGT interrupt
044h		AGT1_AGTCMAI	Compare match A
045h		AGT1_AGTCMBI	Compare match B
046h	IWDT	IWDT_NMIUNDF	IWDT underflow
047h	WDT	WDT_NMIUNDF	WDT underflow
049h	RTC	RTC_PRD	Periodic interrupt
04Bh	ADC120	ADC120_ADI	A/D scan end interrupt
04Fh		ADC120_WCMPM*3	Compare match
050h		ADC120_WCMPUM*3	Compare mismatch
051h	ADC121	ADC121_ADI	A/D scan end interrupt
055h		ADC121_WCMPM*3	Compare match
056h		ADC121_WCMPUM*3	Compare mismatch

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (2 of 7)

Event number	Interrupt request source	Name	Description
057h	ACMPHS	ACMP_HS0*1	High-Speed Analog Comparator interrupt 0
058h		ACMP_HS1*1	High-Speed Analog Comparator interrupt 1
059h		ACMP_HS2*1	High-Speed Analog Comparator interrupt 2
05Ah		ACMP_HS3*1	High-Speed Analog Comparator interrupt 3
05Bh		ACMP_HS4*1	High-Speed Analog Comparator interrupt 4
05Ch		ACMP_HS5*1	High-Speed Analog Comparator interrupt 5
063h	IIC0	IIC0_RXI	Receive data full
064h		IIC0_TXI	Transmit data empty
065h		IIC0_TEI	Transmit end
066h		IIC0_EEI	Transfer error
068h	IIC1	IIC1_RXI	Receive data full
069h		IIC1_TXI	Transmit data empty
06Ah		IIC1_TEI	Transmit end
06Bh		IIC1_EEI	Transfer error
06Dh	IIC2	IIC2_RXI	Receive data full
06Eh		IIC2_TXI	Transmit data empty
06Fh		IIC2_TEI	Transmit end
070h		IIC2_EEI	Transfer error
086h	DOC	DOC_DOPCI*3	Data Operation Circuit interrupt
094h	I/O port	IOPORT_GROUP1	Port 1 event
095h		IOPORT_GROUP2	Port 2 event
096h		IOPORT_GROUP3	Port 3 event
097h		IOPORT_GROUP4	Port 4 event
098h	ELC	ELC_SWEVT0	Software event 0
099h		ELC_SWEVT1	Software event 1
0B0h	GPT32EH0	GPT0_CCMPA	Compare match A
0B1h		GPT0_CCMPB	Compare match B
0B2h		GPT0_CMPC	Compare match C
0B3h		GPT0_CMPD	Compare match D
0B4h		GPT0_CMPE	Compare match E
0B5h		GPT0_CMPF	Compare match F
0B6h		GPT0_OVF	Overflow
0B7h		GPT0_UDF	Underflow
0B8h		GPT0_ADTRGA	A/D converter start request A
0B9h		GPT0_ADTRGB	A/D converter start request B
0BAh	GPT32EH1	GPT1_CCMPA	Compare match A
0BBh		GPT1_CCMPB	Compare match B
0BCh		GPT1_CMPC	Compare match C
0BDh		GPT1_CMPD	Compare match D
0BEh		GPT1_CMPE	Compare match E
0BFh		GPT1_CMPF	Compare match F
0C0h		GPT1_OVF	Overflow
0C1h		GPT1_UDF	Underflow
0C2h		GPT1_ADTRGA	A/D converter start request A
0C3h		GPT1_ADTRGB	A/D converter start request B

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (3 of 7)

Event number	Interrupt request source	Name	Description
0C4h	GPT32EH2	GPT2_CCMPA	Compare match A
0C5h		GPT2_CCMPB	Compare match B
0C6h		GPT2_CMPC	Compare match C
0C7h		GPT2_CMPD	Compare match D
0C8h		GPT2_CMPE	Compare match E
0C9h		GPT2_CMPF	Compare match F
0CAh		GPT2_OVF	Overflow
0CBh		GPT2_UDF	Underflow
0CCh		GPT2_ADTRGA	A/D converter start request A
0CDh		GPT2_ADTRGB	A/D converter start request B
0CEh	GPT32EH3	GPT3_CCMPA	Compare match A
0CFh		GPT3_CCMPB	Compare match B
0D0h		GPT3_CMPC	Compare match C
0D1h		GPT3_CMPD	Compare match D
0D2h		GPT3_CMPE	Compare match E
0D3h		GPT3_CMPF	Compare match F
0D4h		GPT3_OVF	Overflow
0D5h		GPT3_UDF	Underflow
0D6h		GPT3_ADTRGA	A/D converter start request A
0D7h		GPT3_ADTRGB	A/D converter start request B
0D8h	GPT32E4	GPT4_CCMPA	Compare match A
0D9h		GPT4_CCMPB	Compare match B
0DAh		GPT4_CMPC	Compare match C
0DBh		GPT4_CMPD	Compare match D
0DCh		GPT4_CMPE	Compare match E
0DDh		GPT4_CMPF	Compare match F
0DEh		GPT4_OVF	Overflow
0DFh		GPT4_UDF	Underflow
0E0h		GPT4_ADTRGA	A/D converter start request A
0E1h		GPT4_ADTRGB	A/D converter start request B
0E2h	GPT32E5	GPT5_CCMPA	Compare match A
0E3h		GPT5_CCMPB	Compare match B
0E4h		GPT5_CMPC	Compare match C
0E5h		GPT5_CMPD	Compare match D
0E6h		GPT5_CMPE	Compare match E
0E7h		GPT5_CMPF	Compare match F
0E8h		GPT5_OVF	Overflow
0E9h		GPT5_UDF	Underflow
0EAh		GPT5_ADTRGA	A/D converter start request A
0EBh		GPT5_ADTRGB	A/D converter start request B

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (4 of 7)

Event number	Interrupt request source	Name	Description
0ECh	GPT32E6	GPT6_CCMPA	Compare match A
0EDh		GPT6_CCMPB	Compare match B
0EEh		GPT6_CMPC	Compare match C
0EFh		GPT6_CMPD	Compare match D
0F0h		GPT6_CMPE	Compare match E
0F1h		GPT6_CMPF	Compare match F
0F2h		GPT6_OVF	Overflow
0F3h		GPT6_UDF	Underflow
0F4h		GPT6_ADTRGA	A/D converter start request A
0F5h		GPT6_ADTRGB	A/D converter start request B
0F6h	GPT32E7	GPT7_CCMPA	Compare match A
0F7h		GPT7_CCMPB	Compare match B
0F8h		GPT7_CMPC	Compare match C
0F9h		GPT7_CMPD	Compare match D
0FAh		GPT7_CMPE	Compare match E
0FBh		GPT7_CMPF	Compare match F
0FCh		GPT7_OVF	Overflow
0FDh		GPT7_UDF	Underflow
0FEh		GPT7_ADTRGA	A/D converter start request A
0FFh		GPT7_ADTRGB	A/D converter start request B
100h	GPT328	GPT8_CCMPA	Compare match A
101h		GPT8_CCMPB	Compare match B
102h		GPT8_CMPC	Compare match C
103h		GPT8_CMPD	Compare match D
104h		GPT8_CMPE	Compare match E
105h		GPT8_CMPF	Compare match F
106h		GPT8_OVF	Overflow
107h		GPT8_UDF	Underflow
10Ah	GPT329	GPT9_CCMPA	Compare match A
10Bh		GPT9_CCMPB	Compare match B
10Ch		GPT9_CMPC	Compare match C
10Dh		GPT9_CMPD	Compare match D
10Eh		GPT9_CMPE	Compare match E
10Fh		GPT9_CMPF	Compare match F
110h		GPT9_OVF	Overflow
111h		GPT9_UDF	Underflow
114h	GPT3210	GPT10_CCMPA	Compare match A
115h		GPT10_CCMPB	Compare match B
116h		GPT10_CMPC	Compare match C
117h		GPT10_CMPD	Compare match D
118h		GPT10_CMPE	Compare match E
119h		GPT10_CMPF	Compare match F
11Ah		GPT10_OVF	Overflow
11Bh		GPT10_UDF	Underflow

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (5 of 7)

Event number	Interrupt request source	Name	Description
11Eh	GPT3211	GPT11_CCMPA	Compare match A
11Fh		GPT11_CCMPB	Compare match B
120h		GPT11_CMPC	Compare match C
121h		GPT11_CMPD	Compare match D
122h		GPT11_CMPE	Compare match E
123h		GPT11_CMPF	Compare match F
124h		GPT11_OVF	Overflow
125h		GPT11_UDF	Underflow
128h	GPT3212	GPT12_CCMPA	Compare match A
129h		GPT12_CCMPB	Compare match B
12Ah		GPT12_CMPC	Compare match C
12Bh		GPT12_CMPD	Compare match D
12Ch		GPT12_CMPE	Compare match E
12Dh		GPT12_CMPF	Compare match F
12Eh		GPT12_OVF	Overflow
12Fh		GPT12_UDF	Underflow
132h	GPT3213	GPT13_CCMPA	Compare match A
133h		GPT13_CCMPB	Compare match B
134h		GPT13_CMPC	Compare match C
135h		GPT13_CMPD	Compare match D
136h		GPT13_CMPE	Compare match E
137h		GPT13_CMPF	Compare match F
138h		GPT13_OVF	Overflow
139h		GPT13_UDF	Underflow
150h	GPT	GPT_UVWEDGE	UVW edge event
165h	Ethernet Controller	ETHER_RISE0	Pulse output timer 0 rising edge detection
166h		ETHER_RISE1	Pulse output timer 1 rising edge detection
167h		ETHER_RISE2	Pulse output timer 2 rising edge detection
168h		ETHER_RISE3	Pulse output timer 3 rising edge detection
169h		ETHER_RISE4	Pulse output timer 4 rising edge detection
16Ah		ETHER_RISE5	Pulse output timer 5 rising edge detection
16Bh		ETHER_FALL0	Pulse output timer 0 falling edge detection
16Ch		ETHER_FALL1	Pulse output timer 1 falling edge detection
16Dh		ETHER_FALL2	Pulse output timer 2 falling edge detection
16Eh		ETHER_FALL3	Pulse output timer 3 falling edge detection
16Fh		ETHER_FALL4	Pulse output timer 4 falling edge detection
170h		ETHER_FALL5	Pulse output timer 5 falling edge detection
174h	SCI0	SCI0_RXI *4	Receive data full
175h		SCI0_TXI *4	Transmit data empty
176h		SCI0_TEI	Transmit end
177h		SCI0_ERI *4	Receive error
178h		SCI0_AM	Address match event

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (6 of 7)

Event number	Interrupt request source	Name	Description
17Ah	SCI1	SCI1_RXI *4	Receive data full
17Bh		SCI1_TXI *4	Transmit data empty
17Ch		SCI1_TEI	Transmit end
17Dh		SCI1_ERI *4	Receive error
17Eh		SCI1_AM	Address match event
180h	SCI2	SCI2_RXI *4	Receive data full
181h		SCI2_TXI *4	Transmit data empty
182h		SCI2_TEI	Transmit end
183h		SCI2_ERI *4	Receive error
184h		SCI2_AM	Address match event
186h	SCI3	SCI3_RXI *4	Receive data full
187h		SCI3_TXI *4	Transmit data empty
188h		SCI3_TEI	Transmit end
189h		SCI3_ERI *4	Receive error
18Ah		SCI3_AM	Address match event
18Ch	SCI4	SCI4_RXI *4	Receive data full
18Dh		SCI4_TXI *4	Transmit data empty
18Eh		SCI4_TEI	Transmit end
18Fh		SCI4_ERI *4	Receive error
190h		SCI4_AM	Address match event
192h	SCI5	SCI5_RXI *4	Receive data full
193h		SCI5_TXI *4	Transmit data empty
194h		SCI5_TEI	Transmit end
195h		SCI5_ERI *4	Receive error
196h		SCI5_AM	Address match event
198h	SCI6	SCI6_RXI *4	Receive data full
199h		SCI6_TXI *4	Transmit data empty
19Ah		SCI6_TEI	Transmit end
19Bh		SCI6_ERI *4	Receive error
19Ch		SCI6_AM	Address match event
19Eh	SCI7	SCI7_RXI *4	Receive data full
19Fh		SCI7_TXI *4	Transmit data empty
1A0h		SCI7_TEI	Transmit end
1A1h		SCI7_ERI *4	Receive error
1A2h		SCI7_AM	Address match event
1A4h	SCI8	SCI8_RXI *4	Receive data full
1A5h		SCI8_TXI *4	Transmit data empty
1A6h		SCI8_TEI	Transmit end
1A7h		SCI8_ERI *4	Receive error
1A8h		SCI8_AM	Address match event
1AAh	SCI9	SCI9_RXI *4	Receive data full
1ABh		SCI9_TXI *4	Transmit data empty
1ACh		SCI9_TEI	Transmit end
1ADh		SCI9_ERI *4	Receive error
1AEh		SCI9_AM	Address match event

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (7 of 7)

Event number	Interrupt request source	Name	Description
1BCh	SPI0	SPI0_SPRI	Receive data full
1BDh		SPI0_SPTI	Transmit data empty
1BEh		SPI0_SPII	Idle
1BFh		SPI0_SPEI	Receive error
1C0h		SPI0_SPTEND	Transmit end
1C1h	SPI1	SPI1_SPRI	Receive data full
1C2h		SPI1_SPTI	Transmit data empty
1C3h		SPI1_SPII	Idle
1C4h		SPI1_SPEI	Receive error
1C5h		SPI1_SPTEND	Transmit end

Note 1. Only pulse (edge detection) is supported.
Note 2. ELSR8 to ELSR11, ELSR14 to ELSR17, and ELSR18 can select this event.
Note 3. This event can occur in Snooze mode.
Note 4. This event is not supported in FIFO mode.

19.3 Operation

19.3.1 Relation between Interrupt Handling and Event Linking

Event number for an event link is the same as that for the associated interrupt source. For information on generating event signals, see the explanation in the chapter for each event source module.

19.3.2 Linking Events

When an event occurs and that event is already set as a trigger in the Event Link Setting Register (ELSRn), the associated module is activated. The operation of the module must be set up in advance. Table 19.4 lists the operations of modules when an event occurs.

Table 19.4 Module operations when event occurs

Module	Operations when event occurs
GPT	• Start counting • Stop counting • Clear counting • Up counting • Down counting • Input capture.
ADC12	Start A/D conversion
DAC12	Start D/A conversion
I/O ports	• Change pin output based on the EORR (reset) or EOSR (set) • Latch pin state to EIDR • The following ports can be used for the ELC: PORT 1 PORT 2 PORT 3 PORT 4.
CTSU	Start measurement operation
DTC	Start DTC data transfer

19.3.3 Example Procedure for Linking Events

To link events:

1. Set the operation of the module for which an event is to be linked.
2. Set the appropriate ELSRn register for the module to be linked.

3. Set the ELCR.ELCON bit to 1 to enable linkage of all events.
4. Configure the module from which an event is output and activate the module. The link between the two modules is now active.
5. To stop event linkage of modules individually, set 000000000b in the ELSRn.ELS[8:0] bits associated with the modules. To stop linkage of all events, set the ELCR.ELCON bit to 0.

If the event link output from the RTC is to be used, set the ELC after the RTC is set, for example, for initialization and time settings. Unintended events can be generated if the RTC settings are made after the ELC settings.

19.4 Usage Notes

19.4.1 Linking DMAC or DTC Transfer End Signals as Events

When linking the DMAC or DTC transfer end signals as events, do not set the same peripheral module as the DMAC or DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC or DTC transfer to the peripheral module is complete.

19.4.2 Setting Clocks

To link events, you must enable the ELC and the related modules. The modules cannot operate if the related modules are in the module-stop state or in low power modes in which the module is stopped (Software Standby mode or Deep Software Standby mode). Some modules can perform in Snooze mode. For more information, see Table 19.3 and section 11, Low Power Modes.

19.4.3 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For more information, see Table 19.3 and section 11, Low Power Modes. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register.

19.4.4 ELC Delay Time

In Figure 19.2, module A uses ELC and accesses module B through the ELC. There is a delay time in the ELC module between module A and module B. See Table 19.5.

If the clock domains in both module A and module B are the same, the delay time is 0. But, if the clock domains in module A and B are different, ELC module has some delay. The time delay is defined by the slower clock frequency between module A and module B clocks.

Table 19.5 ELC delay time

Clock domain	Clock frequency	ELC delay time
clock_A = clock_B	clock_A = clock_B	0 cycle
clock_A ≠ clock_B	clock_A = clock_B	1 cycle to 2 cycles
	clock_A > clock_B	1 cycle to 2 cycles of B
	clock_A < clock_B	1 cycle to 2 cycles of A

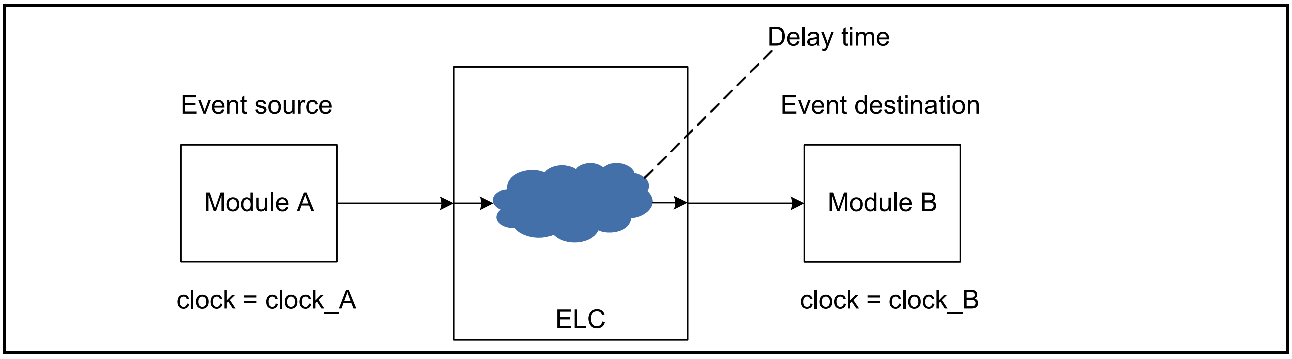


Figure 19.2 ELC delay time

20. I/O Ports

20.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, port group function for the ELC, or bus control pins. All pins operate as input pins immediately after a reset, and pin functions are switched by register settings. You can specify the associated I/O ports and peripheral modules for each pin in the registers.

Figure 20.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs depending on the package. Table 20.1 lists the I/O port specifications by package, and Table 20.2 lists the port functions.

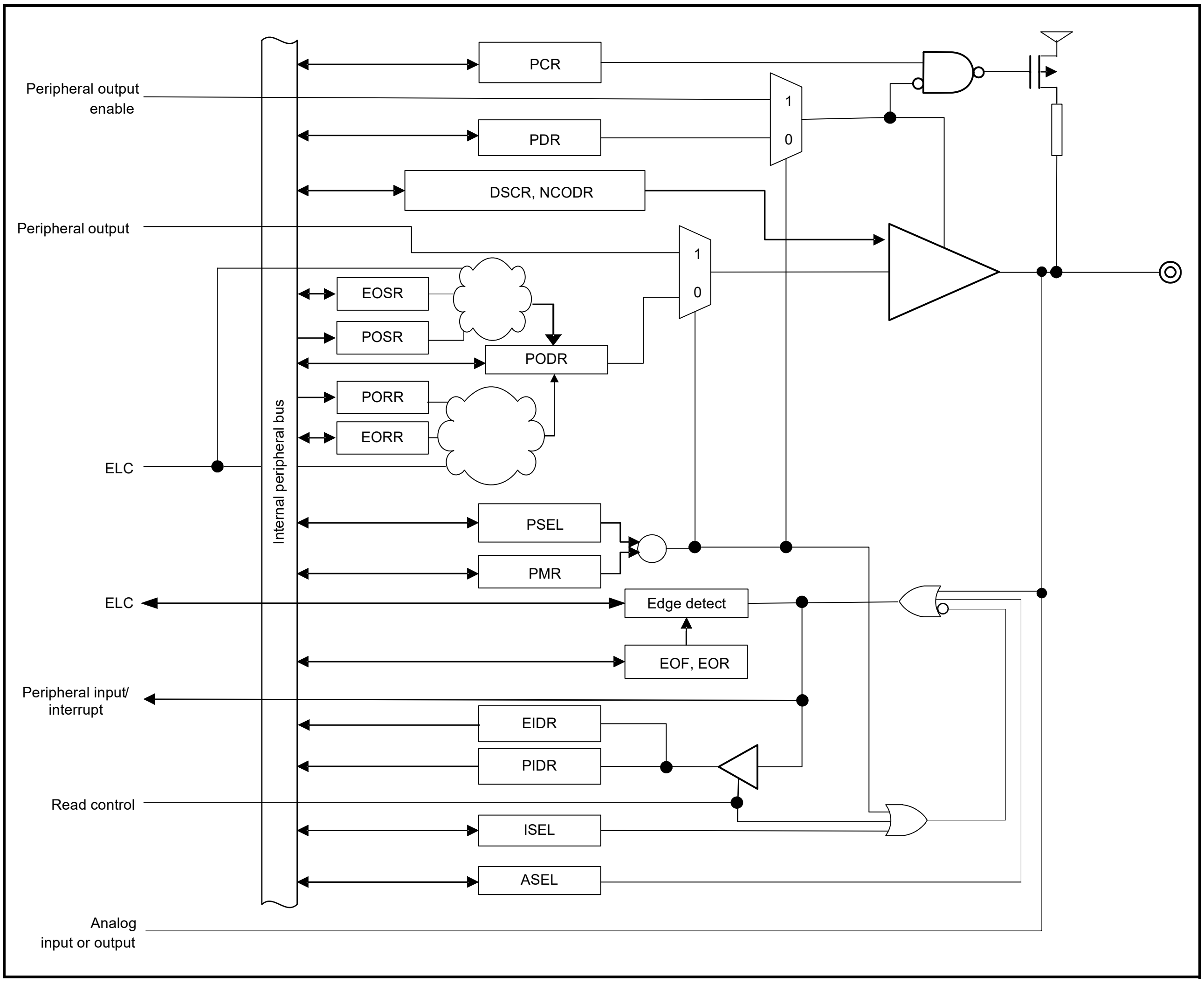


Figure 20.1 Connection diagram for I/O port registers

Note: Figure 20.1 shows a basic port configuration. The configuration differs depending on the ports.

Table 20.1 I/O port specifications

Port	Package		Package		Package	
	176 pins	Number of pins	144 pins, 145 pins	Number of pins	100 pins	Number of pins
PORT0	P000 to P010, P014, P015	13	P000 to P009, P014, P015	12	P000 to P008, P014, P015	11
PORT1	P100 to P115	16	P100 to P115	16	P100 to P115	16
PORT2	P200 to P214	15	P200 to P214	15	P200, P201, P205 to P214	12
PORT3	P300 to P315	16	P300 to P313	14	P300 to P307	8
PORT4	P400 to P415	16	P400 to P415	16	P400 to P415	16
PORT5	P500 to P508, P511 to P513	12	P500 to P506, P508, P511, P512	10	P500 to P504, P508	6
PORT6	P600 to P615	16	P600 to P605, P608 to P614	13	P600 to P602, P608 to P610	6
PORT7	P700 to P708	9	P700 to P705, P708 to P713	12	P708	1
PORT8	P800 to P806	7	P800, P801	2	N/A	0
PORT9	P900, P901, P905 to P908	6	N/A	0	N/A	0
PORTA	PA00, PA01, PA08 to PA10	5	N/A	0	N/A	0
PORTB	PB00, PB01	2	N/A	0	N/A	0
	Total pins	133	Total pins	110	Total pins	76

Table 20.2 I/O port functions

Port	Port name	Input pull-up	Open-drain output	Drive capacity switching	5-V tolerant
PORT0	P000 to P007	-	-	-	-
	P008 to P010, P014, P015	✓	✓	-	-
PORT1	P100 to P115	✓	✓	Low, middle, high	-
PORT2	P200	✓	-	-	-
	P201	✓	✓	-	-
	P202 to P204, P207 to P214	✓	✓	Low, middle, high	-
	P205, P206	✓	✓	Low, middle, high	✓
PORT3	P300 to P315	✓	✓	Low, middle, high	-
PORT4	P400, P401, P407 to P415	✓	✓	Low, middle, high	✓
	P402 to P406	✓	✓	Low, middle, high	-
PORT5	P500 to P508, P513	✓	✓	Low, middle, high	-
	P511, P512	✓	✓	Low, middle, high	✓
PORT6	P600 to P615	✓	✓	Low, middle, high	-
PORT7	P700 to P707	✓	✓	Low, middle, high	-
	P708 to P713	✓	✓	Low, middle, high	✓
PORT8	P800 to P806	✓	✓	Low, middle, high	-
PORT9	P900, P901, P905 to P908	✓	✓	Low, middle, high	-
PORTA	PA00, PA01, PA08 to PA10	✓	✓	Low, middle, high	-
PORTB	PB00	✓	✓	Low, middle, high	-
	PB01	✓	✓	Low, middle, high	✓

✓: Available

20.2 Register Descriptions

20.2.1 Port Control Register 1 (PCNTR1/PODR/PDR)

Address(es): PORT0.PCNTR1 4004 0000h, PORT1.PCNTR1 4004 0020h, PORT2.PCNTR1 4004 0040h, PORT3.PCNTR1 4004 0060h, PORT4.PCNTR1 4004 0080h, PORT5.PCNTR1 4004 00A0h, PORT6.PCNTR1 4004 00C0h, PORT7.PCNTR1 4004 00E0h, PORT8.PCNTR1 4004 0100h, PORT9.PCNTR1 4004 0120h, PORTA.PCNTR1 4004 0140h, PORTB.PCNTR1 4004 0160h

PORT0.PODR 4004 0000h, PORT1.PODR 4004 0020h, PORT2.PODR 4004 0040h, PORT3.PODR 4004 0060h, PORT4.PODR 4004 0080h, PORT5.PODR 4004 00A0h, PORT6.PODR 4004 00C0h, PORT7.PODR 4004 00E0h, PORT8.PODR 4004 0100h, PORT9.PODR 4004 0120h, PORTA.PODR 4004 0140h, PORTB.PODR 4004 0160h

PORT0.PDR 4004 0002h, PORT1.PDR 4004 0022h, PORT2.PDR 4004 0042h, PORT3.PDR 4004 0062h, PORT4.PDR 4004 0082h, PORT5.PDR 4004 00A2h, PORT6.PDR 4004 00C2h, PORT7.PDR 4004 00E2h, PORT8.PDR 4004 0102h, PORT9.PDR 4004 0122h, PORTA.PDR 4004 0142h, PORTB.PDR 4004 0162h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PODR 15	PODR 14	PODR 13	PODR 12	PODR 11	PODR 10	PODR 09	PODR 08	PODR 07	PODR 06	PODR 05	PODR 04	PODR 03	PODR 02	PODR 01	PODR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	PDRn	Pmn Direction	0: Input (functions as an input pin) 1: Output (functions as an output pin).	R/W
b31 to b16	PODRn	Pmn Output Data	0: Low output 1: High output.	R/W

m = 0 to 9, A, B
n = 00 to 15

The Port Control Register 1 is a 32-bit and 16-bit read/write register that controls port direction and port output data.

The PCNTR1 specifies the port direction and the output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit units. Bits associated with non-existent pins are reserved. The write value should be 0. P000 to P007 and P200 are input only, so PORT0.PCNTR1.PDR00-PDR07 and PORT2.PCNTR1.PDR00 are reserved. The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

The PODRn bits hold data to be output from the general I/O pins. Bits associated with non-existent pins are reserved. The write value should be 0. P000 to P007 and P200 are input only, so PORT0.PCNTR1.PODR00-PODR07 and PORT2.PCNTR1.PODR00 are reserved. Writes to P000 to P007 and P200 have no effect. The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

20.2.2 Port Control Register 2 (PCNTR2/EIDR/PIDR)

Address(es): PORT0.PCNTR2 4004 0004h, PORT1.PCNTR2 4004 0024h, PORT2.PCNTR2 4004 0044h, PORT3.PCNTR2 4004 0064h, PORT4.PCNTR2 4004 0084h, PORT5.PCNTR2 4004 00A4h, PORT6.PCNTR2 4004 00C4h, PORT7.PCNTR2 4004 00E4h, PORT8.PCNTR2 4004 0104h, PORT9.PCNTR2 4004 0124h, PORTA.PCNTR2 4004 0144h, PORTB.PCNTR2 4004 0164h

PORT1.EIDR 4004 0024h, PORT2.EIDR 4004 0044h, PORT3.EIDR 4004 0064h, PORT4.EIDR 4004 0084h

PORT0.PIDR 4004 0006h, PORT1.PIDR 4004 0026h, PORT2.PIDR 4004 0046h, PORT3.PIDR 4004 0066h, PORT4.PIDR 4004 0086h, PORT5.PIDR 4004 00A6h, PORT6.PIDR 4004 00C6h, PORT7.PIDR 4004 00E6h, PORT8.PIDR 4004 0106h, PORT9.PIDR 4004 0126h, PORTA.PIDR 4004 0146h, PORTB.PIDR 4004 0166h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EIDR15	EIDR14	EIDR13	EIDR12	EIDR11	EIDR10	EIDR09	EIDR08	EIDR07	EIDR06	EIDR05	EIDR04	EIDR03	EIDR02	EIDR01	EIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b15 to b0	PIDRn	Pmn State	0: Low level 1: High level.	R
b31 to b16	EIDRn	Port Event Input Data*1	When an ELC_PORTx occurs: 0: Low input 1: High input.	R

m = 0 to 9, A, B
n = 00 to 15
x = 1 to 4

Note 1. Supported for PORT1 to PORT4.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 specifies the Pmn state and the port event input data, and is accessed in 32-bit units. The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

The PIDRn bits reflect individual pin states of the port, regardless of the values set in PmnPFS.PMR and PORTm.PCNTR1.PDRn. The PIDRn bit in the PORTm.PCNTR2 register serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- Main clock oscillator (MOSC)
- CS area controller (CSC)
- Analog function (ASEL = 1)
- Capacitive Touch Sensing Unit (CTSU)
- USB 2.0 Full-Speed (USBFS) module.

The EIDRn bits latch a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTm.PCNTR1.PDRn are 0. When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

20.2.3 Port Control Register 3 (PCNTR3/PORR/POSR)

Address(es): PORT0.PCNTR3 4004 0008h, PORT1.PCNTR3 4004 0028h, PORT2.PCNTR3 4004 0048h, PORT3.PCNTR3 4004 0068h, PORT4.PCNTR3 4004 0088h, PORT5.PCNTR3 4004 00A8h, PORT6.PCNTR3 4004 00C8h, PORT7.PCNTR3 4004 00E8h, PORT8.PCNTR3 4004 0108h, PORT9.PCNTR3 4004 0128h, PORTA.PCNTR3 4004 0148h, PORTB.PCNTR3 4004 0168h

PORT0.PORR 4004 0008h, PORT1.PORR 4004 0028h, PORT2.PORR 4004 0048h, PORT3.PORR 4004 0068h, PORT4.PORR 4004 0088h, PORT5.PORR 4004 00A8h, PORT6.PORR 4004 00C8h, PORT7.PORR 4004 00E8h, PORT8.PORR 4004 0108h, PORT9.PORR 4004 0128h, PORTA.PORR 4004 0148h, PORTB.PORR 4004 0168h

PORT0.POSR 4004 000Ah, PORT1.POSR 4004 002Ah, PORT2.POSR 4004 004Ah, PORT3.POSR 4004 006Ah, PORT4.POSR 4004 008Ah, PORT5.POSR 4004 00AAh, PORT6.POSR 4004 00CAh, PORT7.POSR 4004 00EAh, PORT8.POSR 4004 010Ah, PORT9.POSR 4004 012Ah, PORTA.POSR 4004 014Ah, PORTB.POSR 4004 016Ah

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PORR 15	PORR 14	PORR 13	PORR 12	PORR 11	PORR 10	PORR 09	PORR 08	PORR 07	PORR 06	PORR 05	PORR 04	PORR 03	PORR 02	PORR 01	PORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	POSRn	Pmn Output Set	0: No effect on output 1: High output.	W
b31 to b16	PORRn	Pmn Output Reset	0: No effect on output 1: Low output.	W

m = 0 to 9, A, B
n = 00 to 15

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit and 16-bit write register that controls the setting or resetting of the port output data.

The PCNTR3 controls the setting or resetting of the port output data, and is accessed in 32-bit units. The POSRn (bits [15:0] in PCNTR3) and PORRn (bits [31:16] in PCNTR3) respectively, are accessed in 16-bit units.

POSR changes PODR when set by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 is 1, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P000 to P007 and P200 are input only, so PORT0.PCNTR3.POSR00-07 and PORT2.PCNTR3.POSR00 are reserved.

PORR changes PODR when reset by a software write. For example, for P100, when PORT1.PCNTR3.PORR00 is 1, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P000 to P007 and P200 are input only, so PORT0.PCNTR3.PORR00-07 and PORT2.PCNTR3.PORR00 are reserved.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: PORRn and POSRn should not be set at the same time.

20.2.4 Port Control Register 4 (PCNTR4/EORR/EOSR)

Address(es): PORT1.PCNTR4 4004 002Ch, PORT2.PCNTR4 4004 004Ch, PORT3.PCNTR4 4004 006Ch, PORT4.PCNTR4 4004 008Ch
PORT1.EORR 4004 002Ch, PORT2.EORR 4004 004Ch, PORT3.EORR 4004 006Ch, PORT4.EORR 4004 008Ch
PORT1.EOSR 4004 002Eh, PORT2.EOSR 4004 004Eh, PORT3.EOSR 4004 006Eh, PORT4.EOSR 4004 008Eh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EORR 15	EORR 14	EORR 13	EORR 12	EORR 11	EORR 10	EORR 09	EORR 08	EORR 07	EORR 06	EORR 05	EORR 04	EORR 03	EORR 02	EORR 01	EORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	EOSRn	Pmn Event Output Set	When an ELC_PORTx signal occurs: 0: No effect on output 1: High output.	R/W
b31 to b16	EORRn	Pmn Event Output Reset	When an ELC_PORTx signal occurs: 0: No effect on output 1: Low output.	R/W

m = 1 to 4
n = 00 to 15
x = 1 to 4

Port Control Register 4 (PCNTR4/EORR/EOSR) is a 32-bit and 16-bit read/write register that controls the setting or resetting of the port output data by event input from the ELC.

The PCNTR4 controls the setting or resetting of the port output data by event input from the ELC, and is accessed in 32-bit units. The EOSRn (bits [15:0] in PCNTR4) and EORRn (bits [31:16] in PCNTR4) respectively, are accessed in 16-bit units.

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for P100, if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P000 to P007 and P200 are input only, so PORT0.PCNTR4.EOSR00-07 and PORT2.PCNTR4.EOSR00 are reserved.

EORR changes PODR when reset because an ELC_PORTx signal occurs. For example, for P100, if PORT1.PCNTR4.EORR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P000 to P007 and P200 are input only, so PORT0.PCNTR4.EORR00-07 and PORT2.PCNTR4.EORR00 are reserved.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: EORRn and EOSRn should not be set at the same time.

20.2.5 Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9, A, B; n = 00 to 15)

Address(es):
PFS.P000PFS 4004 0800h to PFS.P015PFS 4004 083Ch, PFS.P100PFS 4004 0840h to PFS.P115PFS 4004 087Ch,
PFS.P200PFS 4004 0880h to PFS.P214PFS 4004 08B8h, PFS.P300PFS 4004 08C0h to PFS.P315PFS 4004 08FCh,
PFS.P400PFS 4004 0900h to PFS.P415PFS 4004 093Ch, PFS.P500PFS 4004 0940h to PFS.P513PFS 4004 0974h,
PFS.P600PFS 4004 0980h to PFS.P615PFS 4004 09BCh, PFS.P700PFS 4004 09C0h to PFS.P713PFS 4004 09F4h,
PFS.P800PFS 4004 0A00h to PFS.P806PFS 4004 0A18h, PFS.P900PFS 4004 0A40h to PFS.P908PFS 4004 0A60h,
PFS.PA00PFS 4004 0A80h to PFS.PA10PFS 4004 0AA8h, PFS.PB00PFS 4004 0AC0h to PFS.PB01PFS 4004 0AC4h

PFS.P000PFS_HA 4004 0802h to PFS.P015PFS_HA 4004 083Eh, PFS.P100PFS_HA 4004 0842h to PFS.P115PFS_HA 4004 087Eh,
PFS.P200PFS_HA 4004 0882h to PFS.P214PFS_HA 4004 08BAh, PFS.P300PFS_HA 4004 08C2h to PFS.P315PFS_HA 4004 08FEh,
PFS.P400PFS_HA 4004 0902h to PFS.P415PFS_HA 4004 093Eh, PFS.P500PFS_HA 4004 0942h to PFS.P513PFS_HA 4004 0976h,
PFS.P600PFS_HA 4004 0982h to PFS.P615PFS_HA 4004 09BEh, PFS.P700PFS_HA 4004 09C2h to PFS.P713PFS_HA 4004 09F6h,
PFS.P800PFS_HA 4004 0A02h to PFS.P806PFS_HA 4004 0A1Ah, PFS.P900PFS_HA 4004 0A42h to PFS.P908PFS_HA 4004 0A62h,
PFS.PA00PFS_HA 4004 0A82h to PFS.PA10PFS_HA 4004 0AAAh, PFS.PB00PFS_HA 4004 0AC2h to PFS.PB01PFS_HA 4004 0AC6h

PFS.P000PFS_BY 4004 0803h to PFS.P015PFS_BY 4004 083Fh, PFS.P100PFS_BY 4004 0843h to PFS.P115PFS_BY 4004 087Fh,
PFS.P200PFS_BY 4004 0883h to PFS.P214PFS_BY 4004 08BBh, PFS.P300PFS_BY 4004 08C3h to PFS.P315PFS_BY 4004 08FFh,
PFS.P400PFS_BY 4004 0903h to PFS.P415PFS_BY 4004 093Fh, PFS.P500PFS_BY 4004 0943h to PFS.P513PFS_BY 4004 0977h,
PFS.P600PFS_BY 4004 0983h to PFS.P615PFS_BY 4004 09BFh, PFS.P700PFS_BY 4004 09C3h to PFS.P713PFS_BY 4004 09F7h,
PFS.P800PFS_BY 4004 0A03h to PFS.P806PFS_BY 4004 0A1Bh, PFS.P900PFS_BY 4004 0A43h to PFS.P908PFS_BY 4004 0A63h,
PFS.PA00PFS_BY 4004 0A83h to PFS.PA10PFS_BY 4004 0AABh, PFS.PB00PFS_BY 4004 0AC3h to PFS.PB01PFS_BY 4004 0AC7h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*2

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	EOF	EOR	DSCR[1:0]		—	—	—	NCODR	—	PCR	—	PDR	PIDR	PODR
Value after reset:	0*2	0	0	0	0	0*2	0	0	0	0	0	0*2	0	0	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	PODR	Port Output Data	0: Low output 1: High output.	R/W
b1	PIDR	Pmn State	0: Low level 1: High level.	R
b2	PDR	Port Direction	0: Input (functions as an input pin) 1: Output (functions as an output pin).	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	PCR	Pull-up Control	0: Disable input pull-up 1: Enable input pull-up.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	NCODR	N-Channel Open-Drain Control	0: CMOS output 1: NMOS open-drain output.	R/W
b9 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11, b10	DSCR[1:0]	Port Drive Capability	b11 b10 0 0: Low drive 0 1: Middle drive 1 0: Setting prohibited 1 1: High drive.	R/W
b13, b12	EOF/EOR	Event on Falling/Event on Rising*1	b13 b12 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges.	R/W
b14	ISEL	IRQ Input Enable	0: Not used as IRQn input pin 1: Used as IRQn input pin.	R/W
b15	ASEL	Analog Input Enable	0: Not used as analog pin 1: Used as analog pin.	R/W
b16	PMR	Port Mode Control	0: Used as general I/O pin 1: Used as I/O port for peripheral functions.	R/W

Bit	Symbol	Bit name	Description	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	PSEL[4:0]	Peripheral Select	These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: P011PFS to P013PFS, P509PFS, P510PFS, P902PFS to P904PFS, and PA02PFS to PA07PFS for 32-bit, 16-bit, and 8-bit access are not available.

Note 1. Supported for PORT1 to PORT4.

Note 2. The initial value of P000 to P007, P108, P109, P110, P201 and P300 is not 0000_0000h.
P000 to P007 is 0000_8000h, P108 is 0001_0410h, P109 is 0001_0400h, P110 is 0001_0010h, P201 is 0000_0010h, and P300 is 0001_0010h.

The Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) is a 32-bit, 16-bit, and 8-bit read/write control register that selects the port mn pin function, and is accessed in 32-bit units. PmnPFS_HA (bits [15:0] in PmnPFS) is accessed in 16-bit units. PmnPFS_BY (bits [7:0]) is accessed in 8-bit units.

The PDR/PIDR/PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

The PCR bit enables or disables an input pull-up resistor on the individual port pins. When a pin is in the input state with the associated bit in PmnPFS.PCR set to 1, the pull-up resistor connected to the pin is enabled. When a pin is set as an external bus pin, a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. The write value should be 0.

The NCODR bit specifies the output type for the port pins. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

The DSCR bit switches the drive capacity of the port. If the drive capacity of a pin is fixed, the associated bit is read/write, but the drive capacity cannot be changed. Bits associated with non-existent pins are reserved. The write value should be 0.

The EOR and EOF bits select the edge detection method for the port group input signal. These bits support rising, falling, or both edge detections. When the EOR and EOF bits are set to 01b, 10b, or 11b, the input enable of the I/O cell is asserted. Following that, the event pulse is input from the external pin, and the GPIO outputs the event pulse to the ELC. Bits associated with non-existent pins are reserved. The write value should be 0.

The ISEL bit specifies IRQ input pins. This setting can be used in combination with the peripheral functions, although an IRQn (external pin interrupt) of the same number must only be enabled for one pin.

The ASEL bit specifies analog pins. When a pin is set to analog pin by this bit:

1. Specify it as a general I/O port with the Port Mode Control bit (PmnPFS.PMR).
2. Disable the pull-up resistor with the Pull-up Control bit (PmnPFS.PCR).
3. Specify the input in the Port Direction bit (PmnPFS.PDR). The pin state cannot be read at this point. The PmnPFS register is protected by the Write-Protect Register (PWPR). Release write-protect before modifying the register.

The ISEL bit for an unspecified IRQn is reserved. The ASEL bit for an unspecified analog I/O pin is reserved.

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. The write value should be 0.

The PSEL[4:0] bits assign the peripheral function.

For details on the peripheral settings for each product, see section 20.6, Peripheral Select Settings for each Product.

20.2.6 Write-Protect Register (PWPR)

Address(es): PMISC.PWPR 4004 0D03h

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PmnPFS Register Write Enable	0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled.	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled.	R/W

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

20.2.7 Ethernet Control Register (PFENET)

Address(es): PMISC.PFENET 4004 0D00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PHYMODE0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	PHYMODE0	Ethernet Mode Setting ch0	0: RMII mode (ETHERC channel 0) 1: MII mode (ETHERC channel 0).	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PHYMODE0 bit (Ethernet Mode Setting ch0)

The PHYMODE0 bit specifies the PHY mode of ETHERC channel 0. Select the same mode as that specified in the pin function select bits (PmnPFS.PSEL[4:0]). When the signals for the RMII mode are specified in the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 0 (RMII mode). When the signals for the MII mode are specified in the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 1 (MII mode).

20.3 Operation

20.3.1 General I/O Ports

All pins except P000 to P007, P108 to P110, and P300 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pin with the Pin Function Select Registers. For details on these registers, see section 20, Register Descriptions.

Each port has the following bits:

- Port Direction bit (PDRn), which selects input or output direction
- Port Output Data bit (PODRn), which holds data for output
- Port Input Data bit (PIDRn), which indicates the pin state
- Event Input Data bit (EIDRn), which indicates the pin state when an ELC_PORT1, 2, 3, or 4 signal occurs
- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC_PORT1, 2, 3 or 4 signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORT1, 2, 3 or 4 signal occurs.

20.3.2 Port Function Select

The following port functions are available for configuring each pin:

- I/O configuration: Complementary or open-drain output, pull-up control, and drive strength
- General I/O port: Port direction, output data setting, and reading input data
- Alternate function: Configured function mapping to the pin.

Each pin is associated with a Pin Function Select register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS register includes:

- PCR: Pull-up resistor control bit that turns the input pull-up MOS on or off
- NCODR: N-channel open-drain control bit that selects the output type for each pin
- DSCR: Drive capacity control bit that selects the drive capacity
- EOR: Event on rising bit used to detect rising edges on the port input
- EOF: Event on falling bit used to detect falling edges on the port input
- ISEL: IRQ input enable bit to specify an IRQ input pin
- ASEL: Analog input enable bit to specify an analog pin
- PMR: Port mode bit to specify the pin function of each port
- PSEL[4:0]: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Pin Function Select Register. For details, see section 20, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9, A, B; n = 00 to 15).

20.3.3 Port Group Function for the ELC

In the MCU, PORT1 to PORT4 are assigned for the port group function.

20.3.3.1 Behavior when ELC_PORT1, 2, 3, or 4 is input from the ELC

The MCU supports the two functions described in this section when an ELC_PORT1, 2, 3, or 4 signal comes from the ELC.

(1) Input to EIDR

For the GPI function (PDR = 0 and PMR = 0 in the PmnPFS register), when an ELC_PORT1, 2, 3, or 4 signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins are read into the EIDR bit.

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input to the EIDR bit from the external pins.

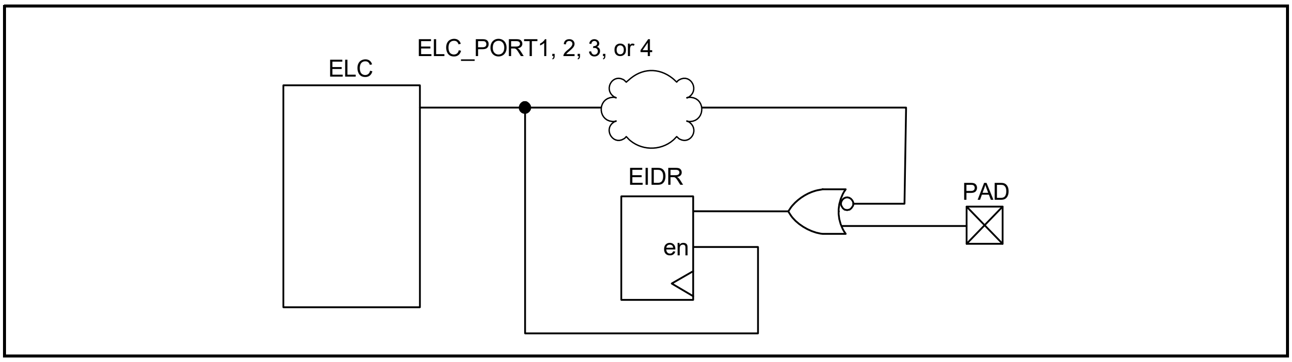


Figure 20.2 Event ports input data

(2) Output from PODR by EOSR/EORR

When an ELC_PORT1, 2, 3, or 4 signal occurs, the data is output from the PODR to the external pin based on the EOSR/EORR bit settings as follows:

- If EOSR is set to 1, when an ELC_PORT1, 2, 3, or 4 signal occurs, the PODR register outputs 1 to the external pin. Otherwise, when EOSR = 0, the PODR value is kept.
- If EORR is set to 1, when an ELC_PORT1, 2, 3, or 4 signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is kept.

See Figure 20.3.

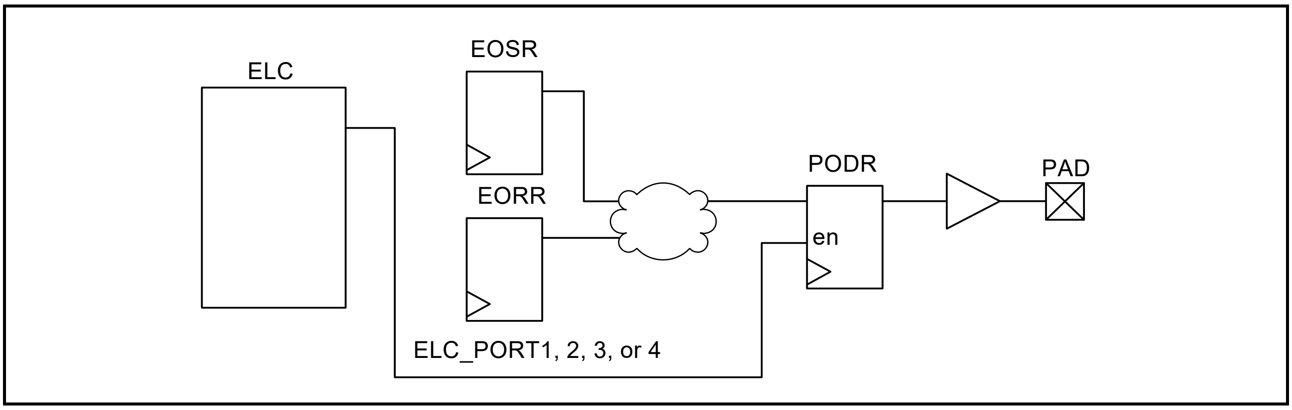


Figure 20.3 Event ports output data

20.3.3.2 Behavior when an event pulse is output to the ELC

To output the event pulse from the external pins to the ELC, set the EOR/EOF bits in the PmnPFS register. For details, see section 20.2.5, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9, A, B; n = 00 to 15). When the EOR/EOF bits are set, the input enable of the I/O cell is asserted.

Data from the external pin is the input. For example, for PORT1, when the data is input from P100 to P115, the data of those 16 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of PORT2 to PORT4 is the same.

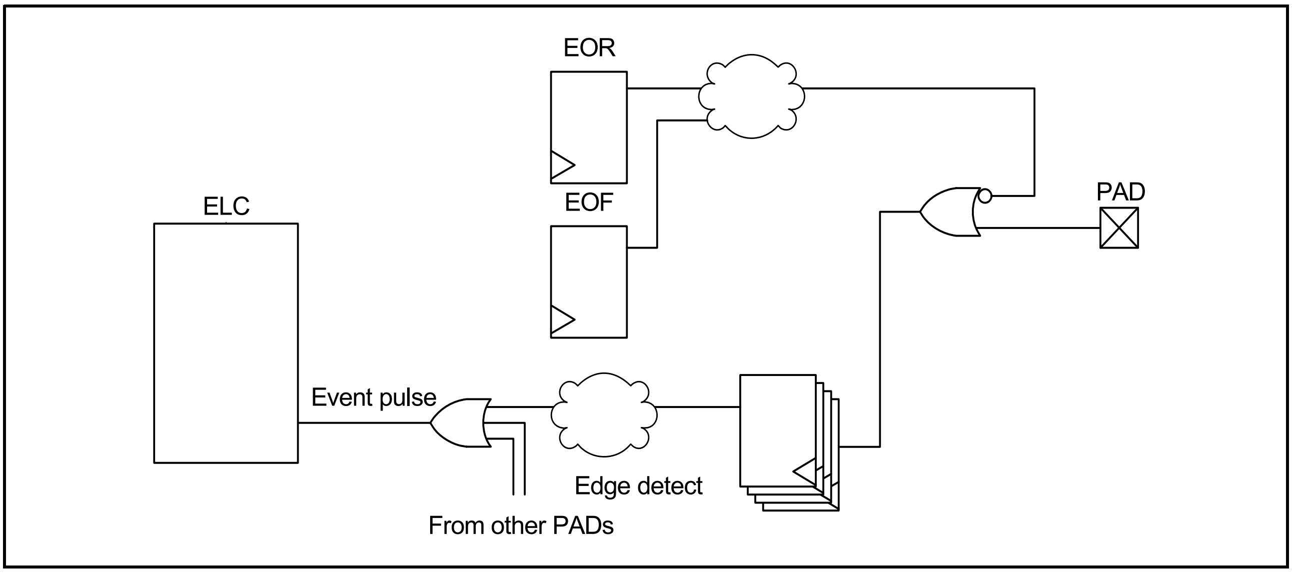


Figure 20.4 Generation of event pulse

20.4 Handling of Unused Pins

Table 20.3 shows how to handle unused pins.

Table 20.3 Handling of unused pins

Pin name	Handling when unused
MD	Use as a mode pin
RES	Connect to VCC through a resistor (pulling up)
USB_DP	Keep pin open
USB_DM	Keep pin open
P200/NMI	Connect to VCC through a resistor (pulling up)
EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as ports 1 to 9.
XTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P213). When the external clock is input to the EXTAL pin, the XTAL pin functions as P213. When this pin is not used as port P213, configure it in the same way as ports 1 to 9.
XCIN	Connect to VSS through a resistor (pulling down)
XCOUT	Keep pin open
P000 to P007	Connect to AVCC0 (pulled up) through a resistor or to AVSS0 (pulled down) through a resistor*1, *4
P008 to P010 P014 to P015	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to AVCC0 (pulled up) through a resistor or to AVSS0 (pulled down) through a resistor*1 • If the direction is set to output (PCNTR1.PDRn = 1), release the pin*1
P1x to P9x PAx to PBx	• If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC (pulled up) through a resistor or to VSS (pulled down) through a resistor*1, *2 • If the direction is set to output (PCNTR1.PDRn = 1), release the pin*1, *3
VREFH0, VREFH	Connect to AVCC0
VREFL0, VREFL	Connect to AVSS0
USBHS_DP USBHS_DM USBHS_RREF	• Preconditions: AVCC_USBHS = VCC_USBHS: Connect to VCC AVSS_USBHS = PVSS_USBHS = VSS1_USBHS = VSS2_USBHS: Connect to VSS Set the module-stop state for USBHS (MSTPCRB.MSTPB12 = 1) • Processing details: USBHS_DP, USBHS_DM, and USBHS_RREF: Open.
VBATT	Connect to VCC or VSS.

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. P108, P110, P300 are recommended for pull up VCC (pulled up) through a resistor, because these pins are input pull-up enabled from the initial value (PmnPFS.PCR=1).
Note 3. P109 is recommended to be set as an output (PCNTR1.PDRn = 1) because this pin is output from the initial value.
Note 4. To reduce input leakage current of P003 and P007, set the P003PFS.ASEL and P007PFS.ASEL bits to 0.

20.5 Usage Notes

20.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR register. This enables writing to the PFSWE bit in the PWPR register.
2. Set 1 to the PFSWE bit in the PWPR register. This enables writing to the PmnPFS register.
3. Clear the port mode control in the PMR for the target pin to select the general I/O port.
4. Specify the I/O function for the pin through the PSEL[4:0] bit settings in the PmnPFS register.
5. Set the PMR bit to 1 as required to switch to the selected I/O function for the pin.
6. Clear the PFSWE bit in the PWPR register. This disables writing to the PmnPFS register.
7. Set 1 to the B0WI bit in the PWPR register. This disables writing to the PFSWE bit in the PWPR register.

20.5.2 Procedure for Using Port Group Input

To use the port group input (PORT1 to PORT4):

1. Set the ELSRx.ELS[8:0] bits to 0000 0000b to ignore unexpected pulses. For more information, see section 19, Event Link Controller (ELC).
2. Set the EOF/EOR bit of the PmnPFS register to specify the rising, falling, or both edge detections.
3. Execute a dummy read or wait for a short time, for example 100 ns. Ignoring of unexpected pulses depends on the initial value of the external pin.
4. Set the ELSRx.ELS[8:0] bits to enable the event signals.

20.5.3 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Output 0 if PCNTR4.EORR is set to 1 when an ELC_PORT1, 2, 3, or 4 signal occurs.
2. Output 1 if PCNTR4.EOSR is set to 1 when an ELC_PORT1, 2, 3, or 4 signal occurs.
3. Output 0 if PCNTR3.PORR is set to 1.
4. Output 1 if PCNTR3.POSR is set to 1.
5. Output 0 or 1 because PCNTR1.PODR is set.
6. Output 0 or 1 because PmnPFS.PODR is set.

Numbers in this list correspond to the priority for writing to the PODR. For example, if 1. and 3. from the list occur at the same time, the higher priority event 1. is executed.

20.5.4 Notes on Using Analog Functions

To use an analog function, set the Port Mode Control bit (PMR) and Port Direction bit (PDR) to 0 so that the pin acts as a general input port. Next, set the Analog Input Enable bit (ASEL) in the Port mn Pin Function Select register (PmnPFS.ASEL) to 1.

20.5.5 I/O Buffer Specification

The P402, P403, and P404 can be used as the RTC input, AGT input, and other peripheral functions.

Table 20.4 lists the P402, P403, P404 specifications.

Table 20.4 P402, P403, P404 specifications

Functions					
I/O port	RTC and AGT			Other peripheral	
	RTC and AGT input enable register	RTC	AGT	Other peripheral enable register	CAC, GPT, CAN, SCI, SSIE, ETHERC (MII), ETHERC (RMII), SDHI, Interrupt, and PDC
P402	VBTICTLR .VCH0INEN	RTCIC0	AGTIO0 AGTIO1	P402PFS.PSEL and PMR	For details, see Table 20.13 Register settings for I/O pin functions on PORT4
P403	VBTICTLR .VCH1INEN	RTCIC1	AGTIO0 AGTIO1	P403PFS.PSEL and PMR	
P404	VBTICTLR .VCH2INEN	RTCIC2	—	P404PFS.PSEL and PMR	

These RTC and AGT inputs are controlled by the VBTICTLR register, which has the highest priority for selecting the RTC and AGT input functions. See Figure 20.5.

The VBTICTLR register is not initialized on reset. Therefore, when not using the RTC or AGT inputs, the associated bit of the VBTICTLR register must be set to 0 after reset.

For more information on the VBTICTLR register, see section 12.2.2, VBATT Input Control Register (VBTICTLR).

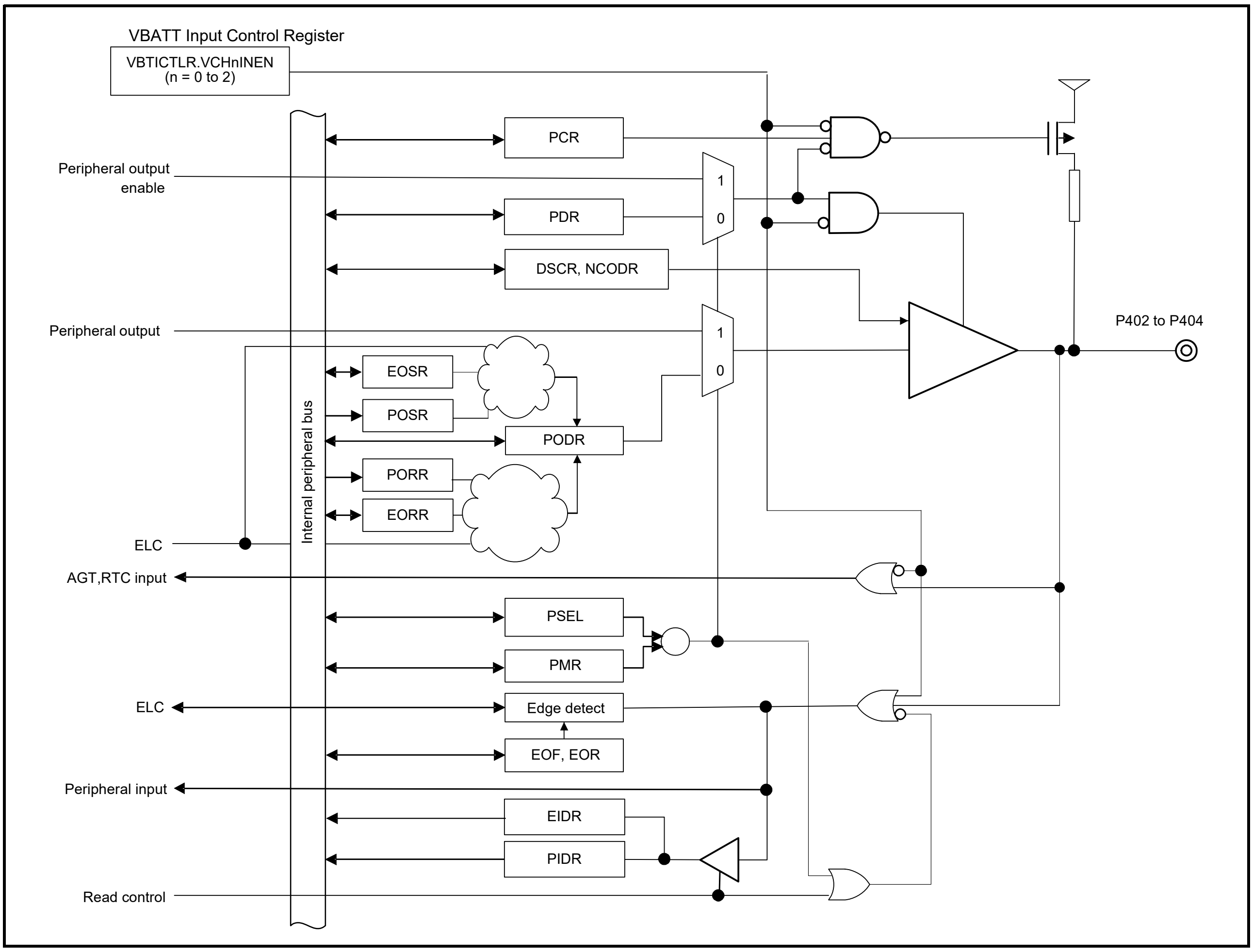


Figure 20.5 P402, P403, P404 diagram

20.6 Peripheral Select Settings for each Product

This section describes the pin function select configuration by the PmnPFS register. Some pin names have added _A, _B, and _C suffixes. When assigning IIC, SPI, SSIE, ETHERC, and SDHI functionality, select the functional pins having the same suffix. The other pins can be selected regardless of the suffix. Assigning the same function to two or more pins simultaneously is prohibited.

20.7 Notes on the PmnPFS Register Setting

(1) In the Port mn Pin Function Select register (PmnPFS), the PSEL bits must be set when the PMR bit of the target pin is 0. If the PSEL bits are set when the PMR bit is 1, unexpected edges might be input for the input function or unexpected pulses might be output to the external pin for the output function.

(2) Only the allowed values (functions) should be specified in the PSEL bits of PmnPFS. If a value that is not allowed for the register is specified, the correct operation is not guaranteed.

(3) A single function should not be assigned to multiple pins by the PmnPFS register.

(4) PORT0 and PORT5 have the analog functions such as A/D converter and D/A converter. When these pins are used as an analog function, to avoid loss of resolution, the PMR and PDR bits should be set to 0. After that, the ASEL bit should be set to 1.

(5) The initial value of the ASEL bit of P003 and P007 is 1. When these pins are not used as an analog function, to reduce the input leakage current, the ASEL bit should be set to 0.

Table 20.5 Register settings for I/O pin functions (PORT0)

PSEL[4:0] settings	Function	Pin							
		P000	P001	P002	P003	P004	P005	P006	P007
ASEL bit		AN000/ IVCMP2	AN001/ IVCMP2	AN002/ IVCMP2	PGAVSS000/ AN007	AN100/ IVCMP2	AN101/ IVCMP2	AN102/ IVCMP2	PGAVSS100/ AN107
ISEL bit		IRQ6-DS	IRQ7-DS	IRQ8-DS		IRQ9-DS	IRQ10-DS	IRQ11-DS	
DSCR[1:0] bits	Drive capacity control	-	-	-	-	-	-	-	-
NCODR bit	N-ch open-drain	-	-	-	-	-	-	-	-
PCR bit	Pull-up								
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available

Table 20.6 Register settings for I/O pin functions (PORT0)

PSEL[4:0] settings	Function	Pin				
		P008	P009	P010	P014	P015
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z				
ASEL bit		AN003	AN004	AN103	AN005/ AN105/ DA0/ IVREF3	AN006/ AN106/ DA1/ IVCMP1
ISEL bit		IRQ12-DS	IRQ13-DS	IRQ14-DS		IRQ13
DSCR[1:0] bits	Drive capacity control	*1	*1	*1	*1	*1
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	-	✓	✓
	100 pins	✓	-	-	✓	✓

✓: Available

Note 1. The drive strength of this port cannot be controlled by PmnPFS.DSCR[1:0] bits.

Table 20.7 Register settings for I/O pin functions (PORT1)

PSEL[4:0] settings	Function	Pin							
		P100	P101	P102	P103	P104	P105	P106	P107
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	AGTIO0	AGTEE0	AGTO0	—	—	—	AGTOB0	AGTOA0
00010b	GPT	GTETRGA	GTETRGB	GTOWLO	GTOWUP	GTETRGB	GTETRGA	—	—
00011b	GPT*2	GTIOC5B	GTIOC5A	GTIOC2B_A	GTIOC2A_A	GTIOC1B	GTIOC1A	GTIOC8B	GTIOC8A
00100b	SCI	RXD0/MISO0/ SCL0	TXD0/MOSI0/ SDA0	SCK0	CTS0_RTS0/ SS0	RXD8/MISO8/ SCL8	TXD8/MOSI8/ SDA8	SCK8	CTS8_RTS8/ SS8
00101b	SCI	SCK1	CTS1_RTS1/ SS1	—	—	—	—	—	—
00110b	SPI*1	MISOA_A	MOSIA_A	RSPCKA_A	SSLA0_A	SSLA1_A	SSLA2_A	SSLA3_A	—
00111b	IIC*1	SCL1_B	SDA1_B	—	—	—	—	—	
01000b	KINT	KR00	KR01	KR02	KR03	KR04	KR05	KR06	KR07
01010b	CAC/ADC12	—	—	ADTRG0	—	—	—	—	—
01011b	BUS	D00[A00/D00]/ DQ00	D01[A01/D01]/ DQ01	D02[A02/D02]/ DQ02	D03[A03/D03]/ DQ03	D04[A04/D04]/ DQ04	D05[A05/D05]/ DQ05	D06[A06/D06]/ DQ06	D07[A07/D07]/ DQ07
10000b	CAN	—	—	CRX0	CTX0	—	—	—	—
11001b	GLCDC	LCD_EXTCLK _A	LCD_CLK_A	LCD_TCON0_ A	LCD_TCON1_ A	LCD_TCON2_ A	LCD_TCON3_ A	LCD_DATA00_ A	LCD_DATA01_ A
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		IRQ2	IRQ1	-	-	IRQ1	IRQ0	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.8 Register settings for I/O pin functions (PORT1)

PSEL[4:0] settings	Function	Pin							
		P108	P109	P110	P111	P112	P113	P114	P115
00000b (value after reset)	Hi-Z/JTAG/SWD	TMS/SWDIO	TDO/SWO	TDI	Hi-Z				
00010b	GPT	GTOULO	GTOVUP	GTOVLO	—	—	—	—	—
00011b	GPT*2	GTIOC0B_A	GTIOC1A_A	GTIOC1B_A	GTIOC3A_A	GTIOC3B_A	GTIOC2A	GTIOC2B	GTIOC4A
00100b	SCI	—	—	CTS2_RTS2/ SS2	SCK2	TXD2/MOSI2/ SDA2	RXD2/MISO2/ SCL2	—	—
00101b	SCI	CTS9_RTS9/ SS9	TXD9/MOSI9/ SDA9	RXD9/MISO9/ SCL9	SCK9	SCK1	—	—	—
00110b	SPI*1	SSLB0_B	MOSIB_B	MISOB_B	RSPCKB_B	SSLB0_B	—	—	—
01001b	CLKOUT/ACMPHS/RTC	—	CLKOUT	VCOUT	—	—	—	—	—
01011b	BUS	—	—	—	A05	A04	A03	A02	A01
10000b	CAN	—	CTX1	CRX1	—	—	—	—	—
10010b	SSIE*1	—	—	—	—	SSIBCK0_B	SSILRCK0/SSIFS0_B	SSIRXD0_B	SSITXD0_B
11001b	GLCDC	—	—	—	LCD_DATA12_A	LCD_DATA11_A	LCD_DATA10_A	LCD_DATA09_A	LCD_DATA08_A
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		-	-	IRQ3	IRQ4	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance “_A” or “_B”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.9 Register settings for I/O pin functions (PORT2)

PSEL[4:0] settings	Function	Pin							
		P200	P201	P202	P203	P204	P205	P206	P207
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	—	—	—	—	AGTIO1	AGTO1	—	—
00010b	GPT	—	—	—	—	GTIW	GTIV	GTIU	—
00011b	GPT*2	—	—	GTIOC5B	GTIOC5A	GTIOC4B	GTIOC4A	—	—
00100b	SCI	—	—	SCK2	CTS2_RTS2/ SS2	SCK4	TXD4/MOSI4/ SDA4	RXD4/MISO4/ SCL4	—
00101b	SCI	—	—	RXD9/MISO9/ SCL9	TXD9/MOSI9/ SDA9	SCK9	CTS9_RTS9/ SS9	—	—
00110b	SPI*1	—	—	MISOB_A	MOSIB_A	RSPCKB_A	SSLB0_A	SSLB1_A	SSLB2_A
00111b	IIC*1	—	—	—	—	SCL0_B	SCL1_A	SDA1_A	—
01001b	CLKOUT/ACMPHS/RTC	—	—	—	—	—	CLKOUT	—	—
01010b	CAC/ADC12	—	—	—	—	CACREF	—	—	—
01011b	BUS	—	—	WR1/BC1	A19	A18	A16	WAIT	A17
01100b	CTSU	—	—	—	TSCAP	TS00	TSCAP	TS01	TS02
10000b	CAN	—	—	CRX0	CTX0	—	—	—	—
10001b	QSPI	—	—	—	—	—	—	—	QSSL
10010b	SSIE*1	—	—	—	—	SSIBCK1_A	SSILRCK1/SSIFS1_A	SSIDATA1_A	—
10011b	USBFS	—	—	—	—	USB_OVRCURB-DS	USB_OVRCURA-DS	USB_VBUSEN	—
10101b	SDHI*1	—	—	SD0DAT6_A	SD0DAT5_A	SD0DAT4_A	SD0DAT3_A	SD0DAT2_A	—
10110b	ETHERC (MII)	—	—	ET0_ERXD2	ET0_COL	ET0_RX_DV	ET0_WOL	ET0_LINKSTA	—
10111b	ETHERC (RMII)	—	—	—	—	—	ET0_WOL	ET0_LINKSTA	—
11001b	GLCDC	—	—	LCD_TCON3_B	—	—	—	—	LCD_DATA23_B
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		NMI	-	IRQ3-DS	IRQ2-DS	-	IRQ1-DS	IRQ0-DS	-
DSCR[1:0] bits	Drive capacity control	-	*3	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	-	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	-	-	-	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Note 3. The drive strength of this port cannot be controlled by PmnPFS.DSCR[1:0] bits.

Table 20.10 Register settings for I/O pin functions (PORT2)

PSEL[4:0] settings	Function	Pin						
		P208	P209	P210	P211	P212	P213	P214
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z						
00001b	AGT	—	—	—	—	AGTEE1	—	—
00010b	GPT	GTOVL0	GTOVUP	GTIW	GTIV	GTETRGD	GTETRGC	GTIU
00011b	GPT*2	—	—	—	—	GTIOC0B	GTIOC0A	—
00101b	SCI	—	—	—	—	RXD1/MISO1/ SCL1	TXD1/MOSI1/ SDA1	—
01010b	CAC/ADC12	—	—	—	—	—	ADTRG1	—
10001b	QSPI	QIO3	QIO2	QIO1	QIO0	—	—	QSPCLK
10101b	SDHI*1	SD0DAT0_B	SD0WP	SD0CD	SD0CMD_B	—	—	SD0CLK_B
10110b	ETHERC (MII)	ET0_LINKSTA	ET0_EXOUT	ET0_WOL	ET0_MDIO	—	—	ET0_MDC
10111b	ETHERC (RMII)	ET0_LINKSTA	ET0_EXOUT	ET0_WOL	ET0_MDIO	—	—	ET0_MDC
11001b	GLCDC	LCD_DATA18_B	LCD_DATA19_B	LCD_DATA20_B	LCD_DATA21_B	—	—	LCD_DATA22_B
11010b	Trace (Debug)	TDATA3	TDATA2	TDATA1	TDATA0	—	—	TCLK
ASEL bit		-	-	-	-	-	-	-
ISEL bit		-	-	-	-	IRQ3	IRQ2	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.11 Register settings for I/O pin functions (PORT3)

PSEL[4:0] settings	Function	Pin							
		P300	P301	P302	P303	P304	P305	P306	P307
00000b (value after reset)	Hi-Z/JTAG/SWD	TCK/SWCLK	Hi-Z						
00001b	AGT	—	AGTIO0	—	—	—	—	—	—
00010b	GPT	—	GTOULO	GTOUUP	—	GTOWLO	GTOWUP	GTOULO	GTOUUP
00011b	GPT*2	GTIOC0A_A	GTIOC4B	GTIOC4A	GTIOC7B	GTIOC7A	—	—	—
00100b	SCI	—	RXD2/MISO2/ SCL2	TXD2/MOSI2/ SDA2	—	RXD6/MISO6/ SCL6	TXD6/MOSI6/ SDA6	SCK6	CTS6_RTS6/ SS6
00101b	SCI	—	CTS9_RTS9/ SS9	—	—	—	—	—	—
00110b	SPI*1	SSLB1_B	SSLB2_B	SSLB3_B	—	—	—	—	—
01011b	BUS	—	A06	A07	A08	A09	A10	A11	A12
10001b	QSPI	—	—	—	—	—	QSPCLK	QSSL	QIO0
11001b	GLCDC	—	LCD_DATA13_ A	LCD_DATA14_ A	LCD_DATA15_ A	LCD_DATA16_ A	LCD_DATA17_ A	LCD_DATA18_ A	LCD_DATA19_ A
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		-	IRQ6	IRQ5	-	IRQ9	IRQ8	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance “_A” or “_B”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.12 Register settings for I/O pin functions (PORT3)

PSEL[4:0] settings	**Function**	**Pin**							
		P308	**P309**	**P310**	**P311**	**P312**	**P313**	**P314**	**P315**
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	—	—	AGTEE1	AGTOB1	AGTOA1	—	—	—
00100b	SCI	—	—	—	—	—	—	—	RXD4
00101b	SCI	—	RXD3	TXD3	SCK3	CTS3_RTS3/ SS3	—	—	—
01010b	CAC/ADC12	—	—	—	—	—	—	ADTRG0	—
01011b	BUS	A13	A14	A15	CS2/RAS	CS3/CAS	A20	A21	A22
10001b	QSPI	QIO1	QIO2	QIO3	—	—	—	—	—
10101b	SDHI*1	—	—	—	—	—	SD0DAT7_A	—	—
10110b	ETHERC (MII)	—	—	—	—	—	ET0_ERXD3	—	—
11001b	GLCDC	LCD_DATA20_A	LCD_DATA21_A	LCD_DATA22_A	LCD_DATA23_A	—	LCD_TCON2_B	LCD_TCON1_B	LCD_TCON0_B
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		-	-	-	-	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	-	-
	100 pins	-	-	-	-	-	-	-	-

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance “_A” or “_B”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Table 20.13 Register settings for I/O pin functions (PORT4)

PSEL[4:0] settings	Function	Pin							
		P400	P401	P402	P403	P404	P405	P406	P407
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	AGTIO1	—	—	—	—	—	—	AGTIO0
00010b	GPT	—	GTETRGA	—	—	—	—	—	—
00011b	GPT*3	GTIOC6A	GTIOC6B	—	GTIOC3A	GTIOC3B	GTIOC1A	GTIOC1B	—
00100b	SCI	SCK4	CTS4_RTS4/SS4	—	—	—	—	—	CTS4_RTS4/SS4
00101b	SCI	SCK7	TXD7/MOSI7/SDA7	RXD7/MISO7/SCL7	CTS7_RTS7/SS7	—	—	—	—
00110b	SPI*2	—	—	—	—	—	—	SSLB3_C	SSLB3_A
00111b	IIC*2	SCL0_A	SDA0_A	—	—	—	—	—	SDA0_B
01001b	CLKOUT/ACMPHS/RTC	—	—	—	—	—	—	—	RTCOUT
01010b	CAC/ADC12	ADTRG1	—	CACREF	—	—	—	—	ADTRG0
01100b	CTSU	—	—	—	—	—	—	—	TS03
10000b	CAN	—	CTX0	CRX0	—	—	—	—	—
10010b	SSIE*2	AUDIO_CLK	—	AUDIO_CLK	SSIBCK0_A	SSILRCK0/SSIFS0_A	SSITXD0_A	SSIRXD0_A	—
10011b	USBFS	—	—	—	—	—	—	—	USB_VBUS
10101b	SDHI*2	—	—	—	SD1DAT7_B	SD1DAT6_B	SD1DAT5_B	SD1DAT4_B	—
10110b	ETHERC (MII)	ET0_WOL	ET0_MDC	ET0_MDIO	ET0_LINKSTA	ET0_EXOUT	ET0_TX_EN	ET0_RX_ER	ET0_EXOUT
10111b	ETHERC (RMII)*2	ET0_WOL	ET0_MDC	ET0_MDIO	ET0_LINKSTA	ET0_EXOUT	RMII0_TXD_EN_B	RMII0_TXD1_B	ET0_EXOUT
11000b	PDC	—	—	VSYNC	PIXD7	PIXD6	PIXD5	PIXD4	—
Don't-care	AGT, RTC	—	—	AGTIO0*1/AGTIO1*1/RTCIC0*1	AGTIO0*1/AGTIO1*1/RTCIC1*1	RTCIC2*1	—	—	—
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		IRQ0	IRQ5-DS	IRQ4-DS	-	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. To use this pin function, set the associated pin as a general input (set the PmnPFS.PDR and PmnPFS.PMR bits to 0).

Note 2. Renesas recommends using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 3. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.14 Register settings for I/O pin functions (PORT4)

PSEL[4:0] settings	Function	Pin							
		P408	P409	P410	P411	P412	P413	P414	P415
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	—	—	AGTOB1	AGTOA1	AGTEE1	—	—	—
00010b	GPT	GTOWLO	GTOWUP	GTOVLO	GTOVUP	GTOULO	GTOUUP	—	—
00011b	GPT*2	GTIOC10B	GTIOC10A	GTIOC9B	GTIOC9A	—	—	GTIOC0B	GTIOC0A
00100b	SCI	—	—	RXD0/MISO0/ SCL0	TXD0/MOSI0/ SDA0	SCK0	CTS0_RTS0/ SS0	—	—
00101b	SCI	RXD3/MISO3/ SCL3	TXD3/MOSI3/ SDA3	SCK3	CTS3_RTS3/ SS3	—	—	—	—
00110b	SPI*1	—	—	MISOA_B	MOSIA_B	RSPCKA_B	SSLA0_B	SSLA1_B	SSLA2_B
00111b	IIC*1	SCL0_B	—	—	—	—	—	—	—
01100b	CTSU	TS04	TS05	TS06	TS07	TS08	TS09	TS10	TS11
10011b	USBFS	USB_ID	USB_EXICEN	—	—	—	—	—	USB_VBUSEN
10100b	USBHS	USBHS_ID	USBHS_EXICEN	—	—	—	—	—	—
10101b	SDHI*1	—	—	SD0DAT1_A	SD0DAT0_A	SD0CMD_A	SD0CLK_A	SD0WP	SD0CD
10110b	ETHERC (MII)	ET0_CRS	ET0_RX_CLK	ET0_ERXD0	ET0_ERXD1	ET0_ETXD0	ET0_ETXD1		ET0_TX_EN
10111b	ETHERC (RMII)*1	RMII0_CRS_DV_A	RMII0_RX_ER_A	RMII0_RXD1_A	RMII0_RXD0_A	REF50CK0_A	RMII0_TXD0_A	RMII0_TXD1_A	RMII0_TXD_EN_A
11000b	PDC	PIXCLK	HSYNC	PIXD0	PIXD1	PIXD2	PIXD3	PIXD4	PIXD5
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		IRQ7	IRQ6	IRQ5	IRQ4	-	-	IRQ9	IRQ8
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	✓
	100 pins	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance “_A” or “_B”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.15 Register settings for I/O pin functions (PORT5)

PSEL[4:0] settings	Function	Pin							
		P500	P501	P502	P503	P504	P505	P506	P507
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	AGTOA0	AGTOB0	—	—	—	—	—	—
00010b	GPT	GTIU	GTIV	GTIW	GTETRGC	GTETRGD	—	—	—
00011b	GPT*2	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	GTIOC13B	—	—
00100b	SCI	—	—	—	CTS6_RTS6/SS6	SCK6	RXD6/MISO6/SCL6	TXD6/MOSI6/SDA6	—
00101b	SCI	—	TXD5/MOSI5/SDA5	RXD5/MISO5/SCL5	SCK5	CTS5_RTS5/SS5	—	—	CTS5_RTS5/SS5
01011b	BUS	—	—	—	—	ALE	—	—	—
10001b	QSPI	QSPCLK	QSSL	QIO0	QIO1	QIO2	QIO3	—	—
10011b	USBFS	USB_VBUSEN	USB_OVRCURA	USB_OVRCURB	USB_EXICEN	USB_ID	—	—	—
10101b	SDHI*1	SD1CLK_A	SD1CMD_A	SD1DAT0_A	SD1DAT1_A	SD1DAT2_A	SD1DAT3_A	SD1CD	SD1WP_A
ASEL bit		AN016/IVREF0	AN116/IVREF1	AN017/IVCMP0	AN117	AN018	AN118	AN019	AN119
ISEL bit		-	IRQ11	IRQ12	-	-	IRQ14	IRQ15	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	-
	100 pins	✓	✓	✓	✓	✓	-	-	-

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.16 Register settings for I/O pin functions (PORT5)

PSEL[4:0] settings	Function	Pin			
		P508	P511	P512	P513
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z			
00010b	GPT	—	—	—	—
00011b	GPT*[1]	—	GTIOC0B	GTIOC0A	—
00100b	SCI	SCK6	RXD4/MISO4/ SCL4	TXD4/MOSI4/ SDA4	—
00101b	SCI	SCK5	—	—	RXD5
00111b	IIC	—	SDA2	SCL2	—
10000b	CAN	—	CRX1	CTX1	—
11000b	PDC	—	PCKO	VSYNC	—
11001b	GLCDC	—	—	—	LCD_DATA16_B
ASEL bit		AN020	-	-	-
ISEL bit		-	IRQ15	IRQ14	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	-
	100 pins	✓	-	-	-

✓: Available

—: Setting prohibited

Note 1. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.17 Register settings for I/O pin functions (PORT6)

PSEL[4:0] settings	Function	Pin							
		P600	P601	P602	P603	P604	P605	P606	P607
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00011b	GPT*1	GTIOC6B	GTIOC6A	GTIOC7B	GTIOC7A	GTIOC8B	GTIOC8A	—	—
00100b	SCI	—	—	—	—	—	—	CTS8_RTS8/ SS8	RXD8
00101b	SCI	SCK9	RXD9	TXD9	CTS9_RTS9/ SS9	—	—	—	—
01001b	CLKOUT/ACMPHS/RTC	CLKOUT	—	—	—	—	—	RTCOUT	—
01010b	CAC/ADC12	CACREF	—	—	—	—	—	—	—
01011b	BUS	RD	WR/ WR0/ DQM00	EBCLK/ SDCLK	D13[A13/D13]/ DQ13	D12[A12/D12]/ DQ12	D11[A11/D11]/ DQ11	—	—
11001b	GLCDC	LCD_DATA02_A	LCD_DATA03_A	LCD_DATA04_A	—	—	—	LCD_DATA03_B	LCD_DATA04_B
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		-	-	-	-	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	-	-
	100 pins	✓	✓	✓	-	-	-	-	-

✓: Available
—: Setting prohibited

Note 1. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.18 Register settings for I/O pin functions (PORT6)

PSEL[4:0] settings	Function	Pin							
		P608	P609	P610	P611	P612	P613	P614	P615
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00011b	GPT*1	GTIOC4B	GTIOC5A	GTIOC5B	—	—	—	—	—
00101b	SCI	—	—	—	CTS7_RTS7/SS7	SCK7	TXD7	RXD7	—
01001b	CLKOUT/ACMPHS/RTC	—	—	—	CLKOUT	—	—	—	—
01010b	CAC/ADC12	—	—	—	CACREF	—	—	—	—
01011b	BUS	A00/BC0/DQM1	CS1/CKE	CS0/WE	SDCS	D08[A08/D08]/DQ08	D09[A09/D09]/DQ09	D10[A10/D10]/DQ10	—
10000b	CAN	—	CTX1	CRX1	—	—	—	—	—
11001b	GLCDC	LCD_DATA07_A	LCD_DATA06_A	LCD_DATA05_A	—	—	—	—	LCD_DATA10_B
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		-	-	-	-	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	✓	-
	100 pins	✓	✓	✓	-	-	-	-	-

✓: Available
—: Setting prohibited

Note 1. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.19 Register settings for I/O pin functions (PORT7)

PSEL[4:0] settings	Function	Pin							
		P700	P701	P702	P703	P704	P705	P706	P707
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	—	—	—	—	AGTO0	AGTIO0	—	—
00011b	GPT*2	GTIOC5A	GTIOC5B	GTIOC6A	GTIOC6B	—	—	—	—
00101b	SCI	—	—	—	—	—	—	RXD3/MISO3/SCL3	TXD3/MOSI3/SDA3
00110b	SPI*1	MISOB_C	MOSIB_C	RSPCKB_C	SSLB0_C	SSLB1_C	SSLB2_C	—	—
01001b	CLKOUT/ACMPHS/RTC	—	—	—	VCOUT	—	—	—	—
10000b	CAN	—	—	—	—	CTX0	CRX0	—	—
10100b	USBHS	—	—	—	—	—	—	USBHS_OVRCURB	USBHS_OVRCURA
10101b	SDHI*1	SD1DAT3_B	SD1DAT2_B	SD1DAT1_B	SD1DAT0_B	SD1CLK_B	SD1CMD_B	SD1CD_B	SD1WP_B
10110b	ETHERC (MII)	ET0_ETXD1	ET0_ETXD0	ET0_ERXD1	ET0_ERXD0	ET0_RX_CLK	ET0_CRS	—	—
10111b	ETHERC (RMII)*1	RMII0_TXD0_B	REF50CK0_B	RMII0_RXD0_B	RMII0_RXD1_B	RMII0_RX_ER_B	RMII0_CRS_DV_B	—	—
11000b	PDC	PIXD3	PIXD2	PIXD1	PIXD0	HSYNC	PIXCLK	—	—
ASEL bit		-	-	-	-	-	-	-	-
ISEL bit		-	-	-	-	-	-	IRQ7	IRQ8
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓	-	-
	100 pins	-	-	-	-	-	-	-	-

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance “_A” or “_B”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.20 Register settings for I/O pin functions (PORT7)

PSEL[4:0] settings	Function	Pin					
		P708	P709	P710	P711	P712	P713
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z					
00001b	AGT	—	—	—	AGTEE0	AGTOB0	AGTOA0
00011b	GPT*2	—	—	—	—	GTIOC2B	GTIOC2A
00101b	SCI	RXD1/MISO1/ SCL1	TXD1/MOSI1/ SDA1	SCK1	CTS1_RTS1/ SS1	—	—
00110b	SPI*1	SSLA3_B	—	—	—	—	—
01010b	CAC/ADC12	CACREF	—	—	—	—	—
01100b	CTSU	TS12	TS13	TS14	TS15	TS16	TS17
10010b	SSIE	AUDIO_CLK	—	—	—	—	—
10110b	ETHERC (MII)	ET0_ETXD3	ET0_ETXD2	ET0_TX_ER	ET0_TX_CLK	—	—
11000b	PDC	PCKO	—	—	—	—	—
ASEL bit		-	-	-	-	-	-
ISEL bit		IRQ11	IRQ10	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	-	-	-	-	-
	144 pins, 145 pins	✓	✓	✓	✓	✓	✓
	100 pins	✓	-	-	-	-	-

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.21 Register settings for I/O pin functions (PORT8)

PSEL[4:0] settings	Function	Pin						
		P800	P801	P802	P803	P804	P805	P806
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z						
00101b	SCI	—	—	—	—	—	TXD5	—
01011b	BUS	D14[A14/D14]/DQ14	D15[A15/D15]/DQ15	—	—	—	—	—
10101b	SDHI*1	—	SD1DAT4_A	SD1DAT5_A	SD1DAT6_A	SD1DAT7_A	—	—
11001b	GLCDC	—	—	LCD_DATA02_B	LCD_DATA01_B	LCD_DATA00_B	LCD_DATA17_B	LCD_EXTCLK_B
ASEL bit		-	-	-	-	-	-	-
ISEL bit		-	-	-	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓	✓	✓
	144 pins, 145 pins	✓	✓	-	-	-	-	-
	100 pins	-	-	-	-	-	-	-

✓: Available
—: Setting prohibited

Note 1. Renesas recommends using pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Table 20.22 Register settings for I/O pin functions (PORT9)

PSEL[4:0] settings	Function	Pin				
		P900	P901	P905	P906	P907
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z				
00001b	AGT	—	AGTIO1	—	—	—
00011b	GPT*1	—	—	GTIOC13B	GTIOC13A	GTIOC12B
00100b	SCI	TXD4	SCK4	—	—	—
01011b	BUS	A23	—	CS4	CS5	CS6
11001b	GLCDC	LCD_CLK_B	LCD_DATA15_B	LCD_DATA11_B	LCD_DATA12_B	LCD_DATA13_B
ASEL bit		-	-	-	-	-
ISEL bit		-	-	-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓
Number of pins	176 pins	✓	✓	✓	✓	✓
	144 pins, 145 pins	-	-	-	-	-
	100 pins	-	-	-	-	-

✓: Available
—: Setting prohibited

Note 1. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.23 Register settings for I/O pin functions (PORT9)

PSEL[4:0] settings	Function	Pin P908
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z
00011b	GPT*1	GTIOC12A
01011b	BUS	CS7
11001b	GLCDC	LCD_DATA14_B
ASEL bit		-
ISEL bit		-
DSCR[1:0] bits	Drive capacity control	L/M/H
NCODR bit	N-ch open-drain	✓
PCR bit	Pull-up	✓
Number of pins	176 pins	✓
	144 pins, 145 pins	-
	100 pins	-

✓: Available
—: Setting prohibited

Note 1. There are two types of output buffer, middle drive and high drive. Renesas recommends using the same drive buffer for output skew spec (t_{GTISK}).

Table 20.24 Register settings for I/O pin functions (PORTA)

PSEL[4:0] settings	Function	Pin PA00	 PA01
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z	
00100b	SCI	TXD8	SCK8
11001b	GLCDC	LCD_DATA05_B	LCD_DATA06_B
ASEL bit		-	-
ISEL bit		-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓
PCR bit	Pull-up	✓	✓
Number of pins	176 pins	✓	✓
	144 pins, 145 pins	-	-
	100 pins	-	-

✓: Available
—: Setting prohibited

Table 20.25 Register settings for I/O pin functions (PORTA)

PSEL[4:0] settings	Function	Pin		
		PA08	PA09	PA10
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z		
11001b	GLCDC	LCD_DATA09_B	LCD_DATA08_B	LCD_DATA07_B
ASEL bit		-	-	-
ISEL bit		-	-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓
Number of pins	176 pins	✓	✓	✓
	144 pins, 145 pins	-	-	-
	100 pins	-	-	-

✓: Available
—: Setting prohibited

Table 20.26 Register settings for I/O pin functions (PORTB)

PSEL[4:0] settings	Function	Pin	
		PB00	PB01
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z	
00101b	SCI	SCK3	CTS3_RTS3/ SS3
10100b	USBHS	USBHS_VBUSEN	USBHS_VBUS
ASEL bit		-	-
ISEL bit		-	-
DSCR[1:0] bits	Drive capacity control	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓
PCR bit	Pull-up	✓	✓
Number of pins	176 pins	✓	✓
	144 pins, 145 pins	-	-
	100 pins	-	-

✓: Available
—: Setting prohibited

21. Key Interrupt Function (KINT)

21.1 Overview

A key interrupt (KEY_INTKR) can be generated by setting the Key Return Mode Register (KRM) and inputting a rising or falling edge on the key interrupt input pins, KR0 to KR7.

Table 21.1 shows the pin assignment for key interrupt detection, Table 21.2 shows the function configuration, and Figure 21.1 shows a block diagram.

Table 21.1 Assignment of key interrupt detection pins

Key Interrupt Mode Control n (n = 0 to 7)	Description
KRM0	Controls KR00 signal in 1-bit units
KRM1	Controls KR01 signal in 1-bit units
KRM2	Controls KR02 signal in 1-bit units
KRM3	Controls KR03 signal in 1-bit units
KRM4	Controls KR04 signal in 1-bit units
KRM5	Controls KR05 signal in 1-bit units
KRM6	Controls KR06 signal in 1-bit units
KRM7	Controls KR07 signal in 1-bit units

Table 21.2 Configuration of key interrupt function

Parameter	Configuration
Input	KR00 to KR07
Control registers	Key Return Control Register (KRCTL) Key Return Mode Register (KRM) Key Return Flag Register (KRF)

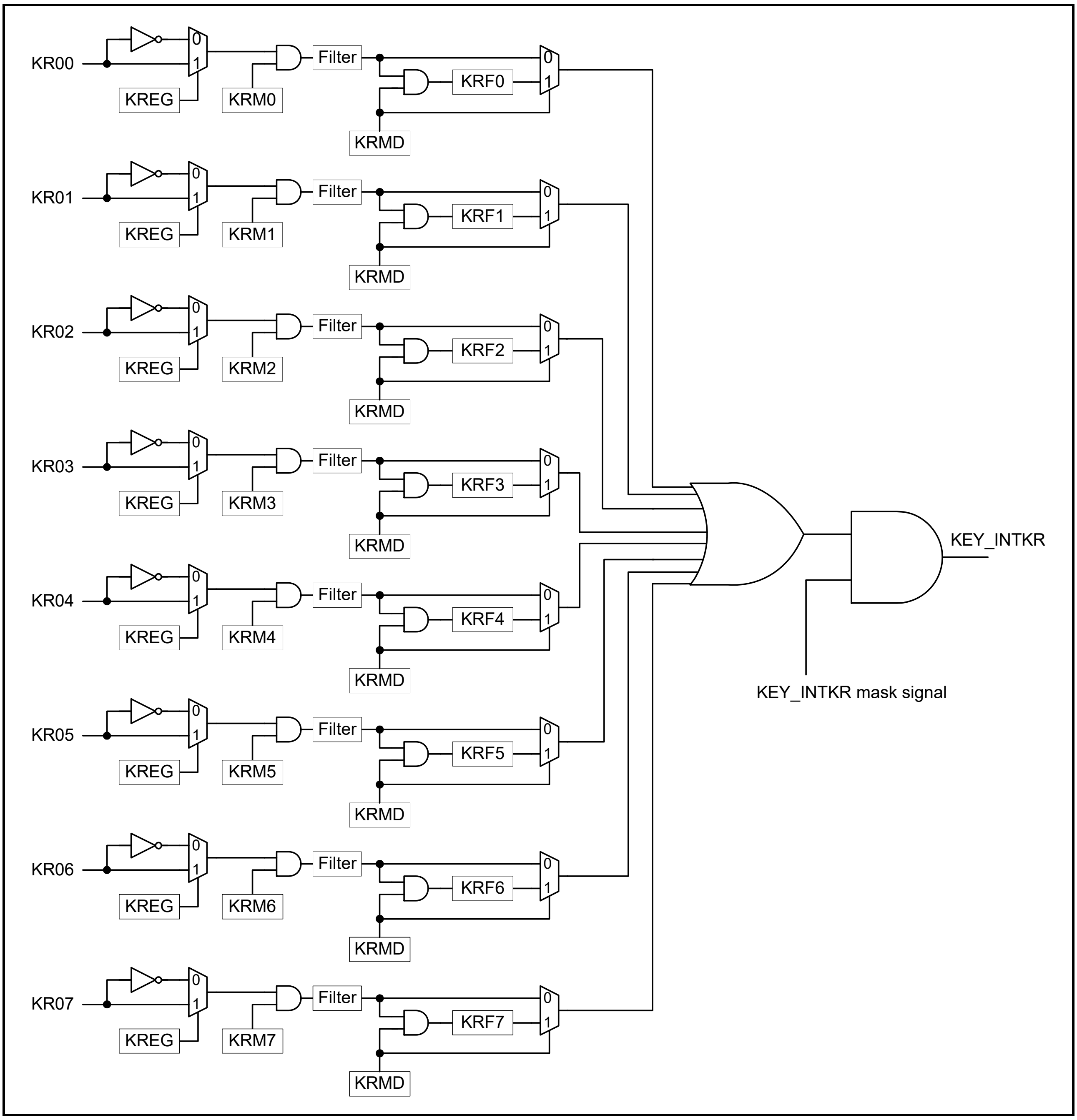


Figure 21.1 Key interrupt block diagram

All key return factors are merged by OR gate. The key interrupt KEY_INTKR is the output of the AND gate to mask merged key return factor by KEY_INTKR mask signal. When using KRFn (KRMD = 1), KEY_INTKR mask signal is used as the output mask that is asserted by clearing KRFn.

21.2 Register Descriptions

21.2.1 Key Return Control Register (KRCTL)

Address(es): KINT.KRCTL 4008 0000h

	b7	b6	b5	b4	b3	b2	b1	b0
	KRMD	—	—	—	—	—	—	KREG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	KREG	Detection Edge Selection (KR00 to KR07)	0: Falling edge 1: Rising edge.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	KRMD	Usage of Key Interrupt Flags (KRF0 to KRF7)	0: Do not use key interrupt flags 1: Use key interrupt flags.	R/W

The KRCTL register controls the usage of the key interrupt flags, KRF0 to KRF7, and sets the detection edge.

21.2.2 Key Return Flag Register (KRF)

Address(es): KINT.KRF 4008 0004h

	b7	b6	b5	b4	b3	b2	b1	b0
	KRF7	KRF6	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	KRFn	Key Interrupt Flag n	0: No key interrupt detected 1: Key interrupt detected.	R/W

n = 0 to 7

Note: When KRMD = 0, setting the KRFn bit to 1 is prohibited.
When setting the KRFn bit to 1, the KRFn value does not change. To clear the KRFn bit, confirm the target bit is 1 before writing 0 to the bit, then write 1 to the other bits.

The KRF register controls the key interrupt flags, KRF0 to KRF7.

21.2.3 Key Return Mode Register (KRM)

Address(es): KINT.KRM 4008 0008h

	b7	b6	b5	b4	b3	b2	b1	b0
	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	KRMn	Key Interrupt Mode Control n	0: No key interrupt signal detected 1: Key interrupt signal detected.	R/W

n = 0 to 7

Note: The on-chip pull-up resistors can be applied by setting the pull-up function for the associated key interrupt input pin. For more information, see section 20, I/O Ports.
Key interrupts can be assigned in the PmnPFS.PSEL bits. For more information, see section 20, I/O Ports.
An interrupt is generated when the target bit in the KRM register is set while a low level (KREG = 0) or a high level (KREG = 1) is being input to the key interrupt input pin. To ignore this interrupt, set the KRM register after disabling the interrupt handling.

The KRM register sets the key interrupt mode.

21.3 Operation

21.3.1 Operation When Not Using Key Interrupt Flag (KRMD = 0)

A key interrupt (KEY_INTKR) is generated when the valid edge specified in the KREG bit is input to a key interrupt pin, KR00 to KR07. To identify the channel to which the valid edge is input, read the port register and check the port level after the key interrupt (KEY_INTKR) is generated.

The KEY_INTKR signal changes based on the input level of the key interrupt input pin, KR00 to KR07.

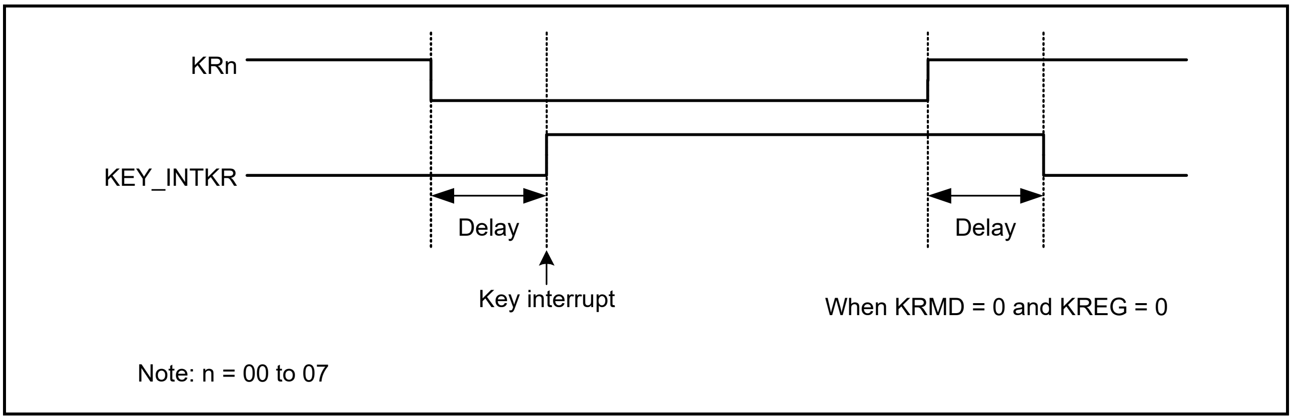


Figure 21.2 Operation of KEY_INTKR signal when a key interrupt is input to a single channel

Figure 21.3 shows the operation when a valid edge is input to multiple key interrupt input pins. The KEY_INTKR signal is set while a low level is being input to one pin (when KREG = 0). Therefore, even if a falling edge is input to another pin in this period, a key interrupt (KEY_INTKR) is not generated again. See [1] in Figure 21.3.

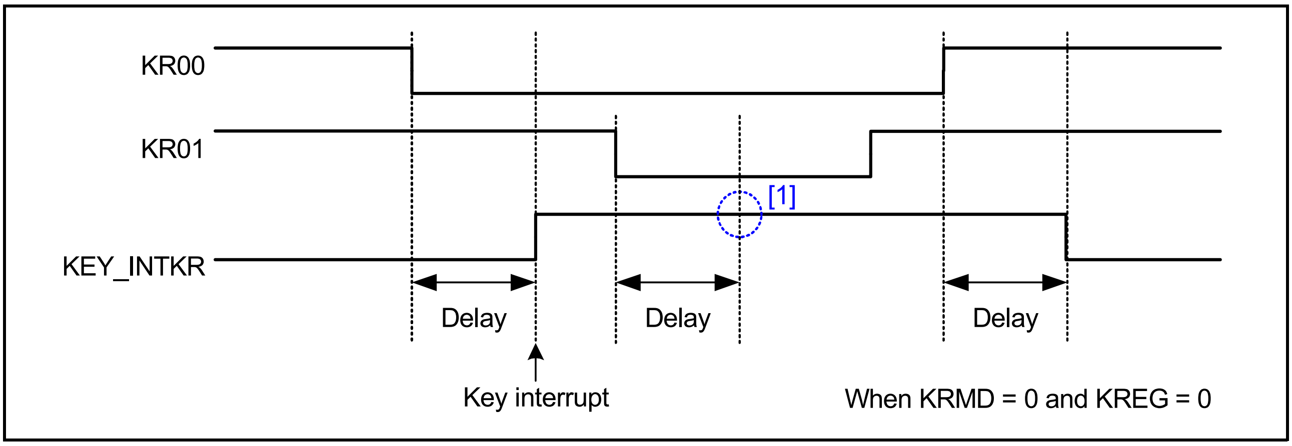


Figure 21.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

21.3.2 Operation When Using the Key Interrupt Flags (KRMD = 1)

A key interrupt (KEY_INTKR) is generated when the valid edge specified in the KREG bit is input to a key interrupt pin, KR00 to KR07. To identify the channels to which the valid edge is input, read the Key Return Flag Register (KRF) after the key interrupt (KEY_INTKR) is generated. If the KRMD bit is set to 1, clear the KEY_INTKR signal by clearing the associated bit in the KRF register.

As Figure 21.4 shows, only one interrupt is generated each time a falling edge is input to one channel, that is, when KREG = 0, regardless of whether the KRFn bit is cleared before or after a rising edge is input.

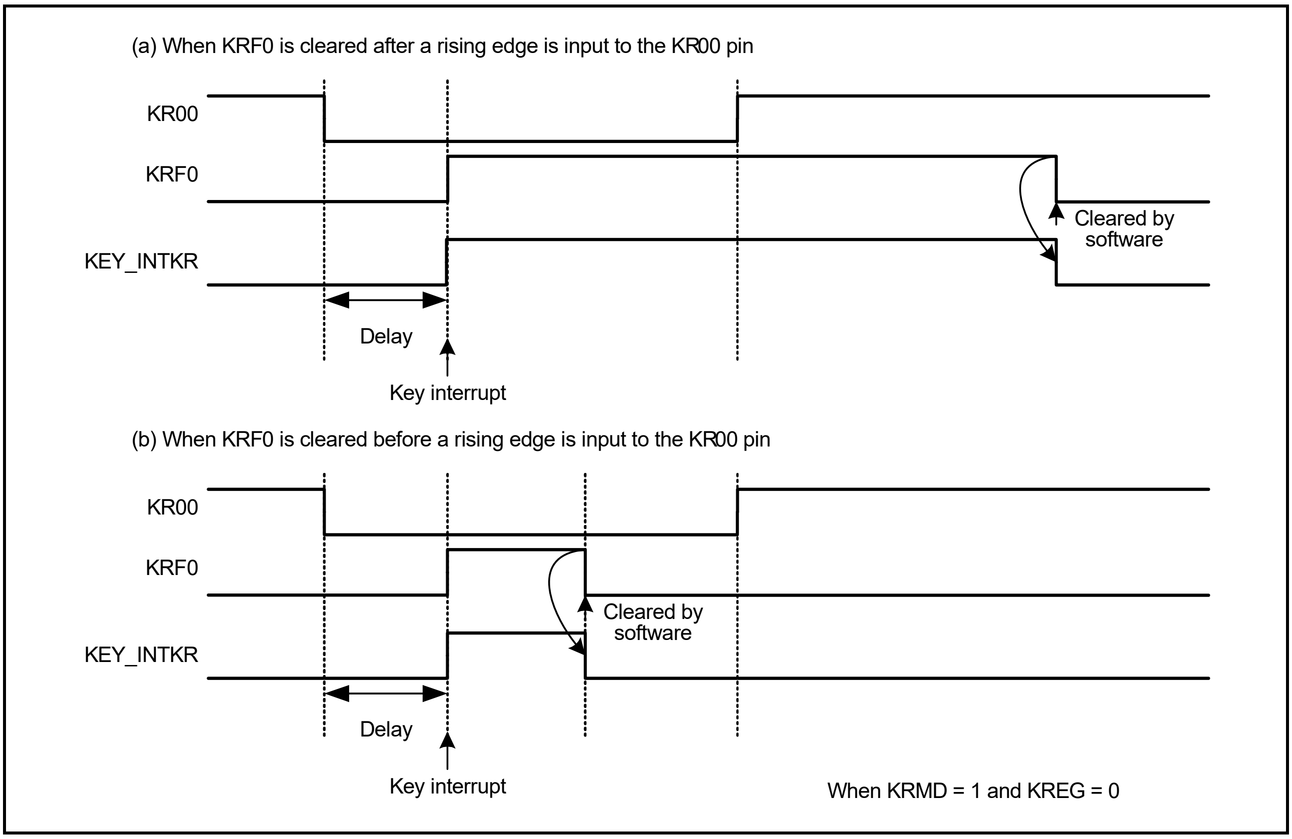


Figure 21.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 21.5. A falling edge is also input to the KR01 and KR05 pins after a falling edge is input to the KR00 pin (when KREG = 0). The KRF1 bit is set when the KRF0 bit is cleared. A key interrupt is generated 1 PCLKB clock cycle, after the KRF0 bit is cleared. See [1] in Figure 21.5.

Also, after a falling edge is input to the KR05 pin, the KRF5 bit is set. The KRF1 bit is cleared at time [2] in the figure. A key interrupt is generated 1 PCLKB clock cycle, after the KRF1 bit is cleared. See [3] in the figure. It is therefore possible to generate a key interrupt when a valid edge is input to multiple channels.

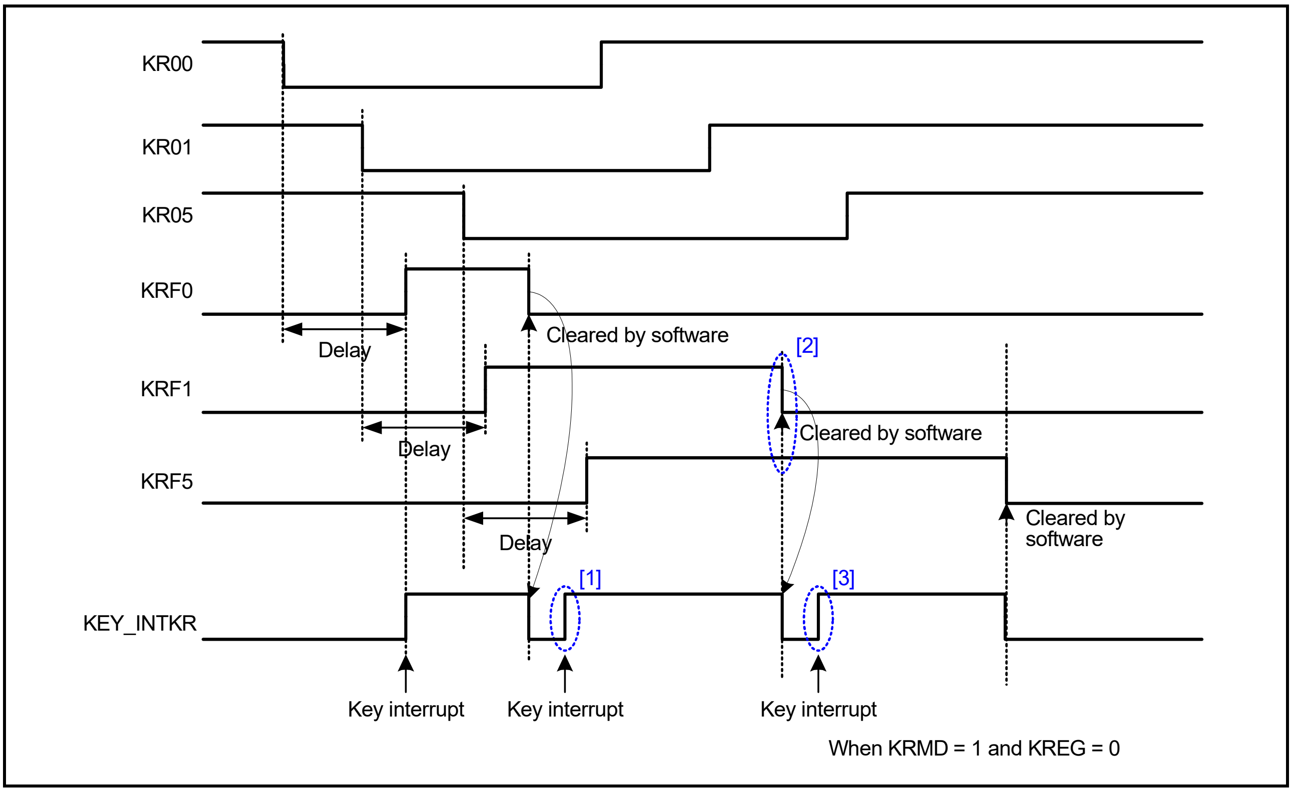


Figure 21.5 **Operation of KEY_INTKR signal when key interrupts are input to multiple channels**

21.4 Usage Notes

- If KEY_INTKR is used as the Snooze request, the KRMD bit must be set to 0.
- If KEY_INTKR is used as the interrupt source for returning to Normal mode from Snooze mode and Software Standby mode, the KRMD bit must be set to 1.
- When the Key Interrupt function (KINT) is assigned to a pin, this pin input is always enabled in Software Standby mode, and if the pin level changes, the associated KRFn flag can be set. Therefore, a key interrupt might occur on canceling Software Standby mode.

To ignore changes to the key interrupt pin during a software standby, clear the associated KRM bit before entering Software Standby mode. After canceling Software Standby mode, you must clear KRFn before the associated KRM bit can be set.

22. Port Output Enable for GPT (POEG)

22.1 Overview

Use the Port Output Enable for GPT (POEG) function to place the General PWM Timer (GPT) output pins in the output-disable state in one of the following ways:

- Input level detection of the GTETRGn (n = A, B, C, D) pins
- Output-disable request from the GPT
- Comparator interrupt request detection
- Oscillation stop detection of the clock generation circuit
- Register settings.

The GTETRGn (n = A, B, C, D) pins can also be used as GPT external trigger input pins.

Table 22.1 lists the POEG specifications, Figure 22.1 shows a block diagram, and Table 22.2 lists the input pins.

Table 22.1 POEG specifications

Parameter	Specifications
Output-disable control through input level detection	The GPT output pins can be disabled when a GTETRGn rising edge or high level is sampled after polarity and filter selection
Output-disable request from the GPT	• When the GTIOCA pin and the GTIOCB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POEG.Through reception of these requests, the POEG can control whether the GTIOCA and GTIOCB pins are output-disabled. • GPT output pins can be set to be disabled when the GPT output pins detect a dead time error.
Output-disable control through comparator (ACMPHS) interrupt detection	The GPT output pins can be disabled when an interrupt request is generated by a change in the output results of any of the comparators
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of the clock generation circuit stops
Output-disable control by software (registers)	The GPT output pins can be disabled by modifying the register settings
Interrupt	• Allows output-disable control by input level detection • Allows output-disable requests from the GPT or ACMPHS.
External trigger output to the GPT (count start, count stop, count clear, up-count, down-count, or input capture function)	The GTETRGn signals can be output to the GPT after polarity and filter selection
Noise filtering	• Three times sampling for every PCLKB/1, PCLKB/8, PCLKB/32, or PCLKB/128 can be set for any of the input pins GTETRGn • Positive or negative polarity can be selected for any of the input pins, GTETRGn • Signal state after polarity and filter selection can be monitored.

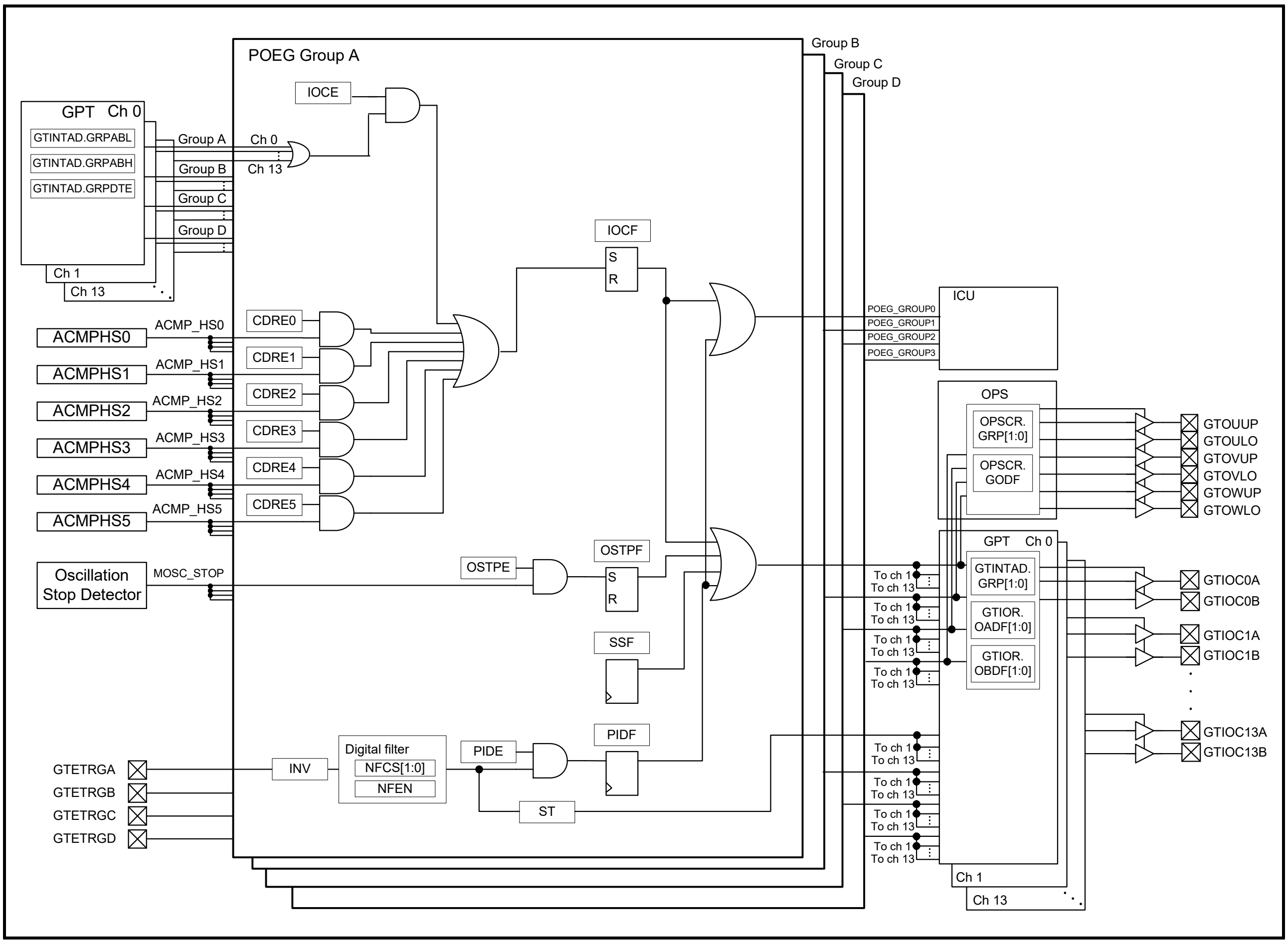


Figure 22.1 POEG block diagram

Table 22.2 POEG input pins

Pin name	I/O	Description
GTETRGA	Input	GPT output pin output-disable request signal and GPT external trigger input pin A
GTETRGB	Input	GPT output pin output-disable request signal and GPT external trigger input pin B
GTETRGC	Input	GPT output pin output-disable request signal and GPT external trigger input pin C
GTETRGD	Input	GPT output pin output-disable request signal and GPT external trigger input pin D

22.2 Register Descriptions

22.2.1 POEG Group n Setting Register (POEGGn) (n = A to D)

Address(es): POEG.POEGGA 4004 2000h, POEG.POEGGB 4004 2100h, POEG.POEGGC 4004 2200h, POEG.POEGGD 4004 2300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CDRE5	CDRE4	CDRE3	CDRE2	CDRE1	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIDF	Port Input Detection Flag	0: No output-disable request from the GTETRGn pin occurred 1: Output-disable request from the GTETRGn pin occurred.	R(/W)*1
b1	IOCF	Detection Flag for GPT or ACMPHS Output-Disable Request	0: No output-disable request from GPT disable request or comparator interrupt occurred 1: Output-disable request from GPT disable request or comparator interrupt occurred.	R(/W)*1
b2	OSTPF	Oscillation Stop Detection Flag	0: No output-disable request from oscillation stop detection occurred 1: Output-disable request from oscillation stop detection occurred.	R(/W)*1
b3	SSF	Software Stop Flag	0: No output-disable request from software occurred 1: Output-disable request from software occurred.	R/W
b4	PIDE	Port Input Detection Enable	0: Output-disable requests from the GTETRGn pins disabled 1: Output-disable requests from the GTETRGn pins enabled.	R/W*2
b5	IOCE	Enable for GPT Output-Disable Request	0: Output-disable requests from GPT disable request disabled 1: Output-disable requests from GPT disable request enabled.	R/W*2
b6	OSTPE	Oscillation Stop Detection Enable	0: Output-disable requests from oscillation stop detection disabled 1: Output-disable requests from oscillation stop detection enabled.	R/W*2
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CDRE0	ACMP_HS0 Enable	0: Comparator 0 disable requests disabled 1: Comparator 0 disable requests enabled.	R/W*2
b9	CDRE1	ACMP_HS1 Enable	0: Comparator 1 disable requests disabled 1: Comparator 1 disable requests enabled.	R/W*2
b10	CDRE2	ACMP_HS2 Enable	0: Comparator 2 disable requests disabled 1: Comparator 2 disable requests enabled.	R/W*2
b11	CDRE3	ACMP_HS3 Enable	0: Comparator 3 disable requests disabled 1: Comparator 3 disable requests enabled.	R/W*2
b12	CDRE4	ACMP_HS4 Enable	0: Comparator 4 disable requests disabled 1: Comparator 4 disable requests enabled.	R/W*2
b13	CDRE5	ACMP_HS5 Enable	0: Comparator 5 disable requests disabled 1: Comparator 5 disable requests enabled.	R/W*2
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ST	GTETRGn Input Status Flag	0: GTETRGn input after filtering was 0 1: GTETRGn input after filtering was 1.	R
b27 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	INV	GTETRGn Input Reverse	0: Input GTETRGn as-is 1: Input GTETRGn reversed.	R/W
b29	NFEN	Noise Filter Enable	0: Noise filtering disabled 1: Noise filtering enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b31, b30	NFCS[1:0]	Noise Filter Clock Select	b1 b0 0 0: GTETRGn pin input level sampled three times every PCLKB 0 1: GTETRGn pin input level sampled three times every PCLKB/8 1 0: GTETRGn pin input level sampled three times every PCLKB/32 1 1: GTETRGn pin input level sampled three times every PCLKB/128.	R/W

Note 1. Only 0 can be written, to clear the flag.
Note 2. Can be modified only once after a reset.

The POEGGA to POEGGD registers control the output-disable state of the GPT pins, interrupts, and the external trigger input to the GPT. In the descriptions, POEGGn represents all of the POEGGA to POEGGD registers.

22.3 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the 3-phase PWM output for BLDC motor control pins can be set to output-disable:

- Input level or edge detection of the GTETRGn pins
  When POEGGn.PIDE is 1, the POEGGn.PIDF flag is set to 1.
- Output-disable request from the GPT
  When POEGGn.IOCE is 1, the POEGGn.IOCF flag is set to 1 if the disable request is enabled in the GTINTAD register. The GTINTAD.GRPDTE, GTINTAD.GRPABH, and GTINTAD.GRPABL settings apply to the group selected in the GPT registers GTINTAD.GRP[1:0] and OPSCR.GRP[1:0].
- Comparator (ACMPHS) interrupt request detection
  Comparator interrupt detection is activated when any of the POEGGn.CDRE[5:0] registers is 1. When the associated comparator interrupt is generated, the GPT output pins are disabled. POEGGn.IOCF indicates the detection status.
- Oscillation stop detection for the clock generation circuit
  When POEGGn.OSTPE is 1, the POEGGn.OSTPF flag is set to 1.
- SSF bit setting
  When POEGGn.SSF is set to 1, the PWM output is disabled.

The output-disable state is controlled in the GPT. The output-disable of the GTIOCxA and GTIOCxB pins is set in the GTINTAD.GRP[1:0], GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in GPTx. The output-disable of the 3-phase PWM output for BLDC motor control pins is set in the OPSCR.GRP[1:0] bits and OPSCR.GODF bit in GPT_OPS.

22.3.1 Pin Input Level Detection Operation

If the input conditions set in POEGGn.PIDE, POEGGn.NFCS[1:0], POEGGn.NFEN, and POEGGn.INV occur on the GTETRGn pins, the GPT output pins are output-disabled.

22.3.1.1 Digital filter

Figure 22.2 shows high level detection by the digital filter. When a high level associated with the POEGGn.INV polarity setting is detected three times consecutively with the sampling clock selected in POEGGn.NFCS[1:0] and POEGGn.NFEN, the detected level is recognized as high, and the GPT output pins are output-disabled. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not being output, changes of the levels on the GTETRGn pins are ignored.

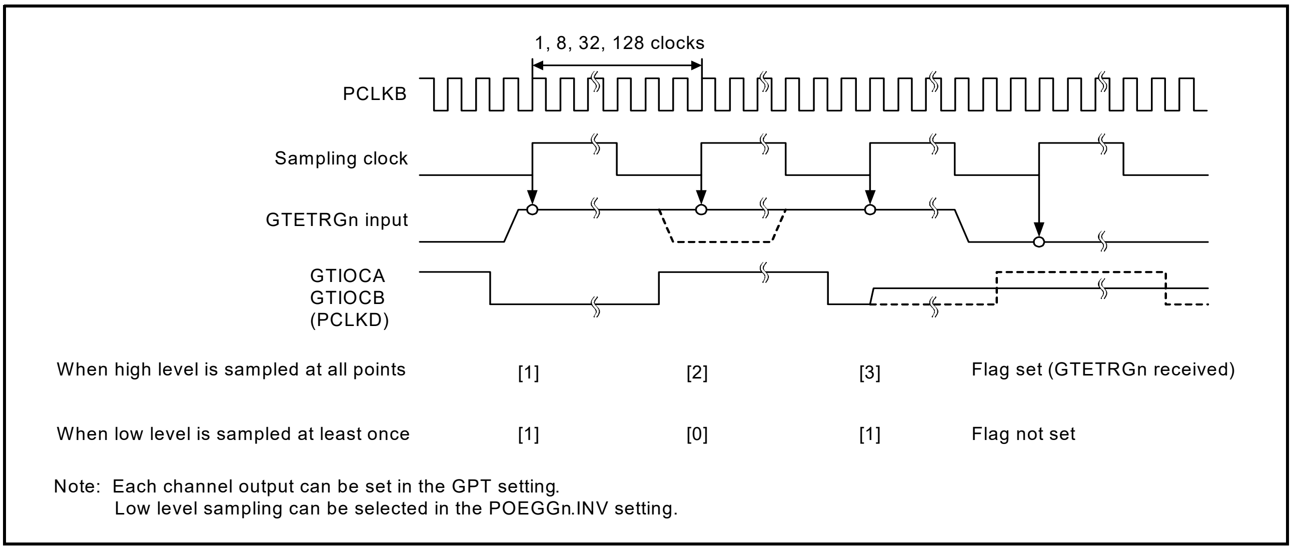


Figure 22.2 Example of digital filter operation

22.3.2 Output-Disable Requests from the GPT

For details on this operation, see the description of GTIOC Pin Output Negate Control in section 23, General PWM Timer (GPT).

22.3.3 Comparator Interrupt Detection

If POEGGn.CDRE[5:0] is 1 when an associated comparator interrupt request is generated, the GPT output pins are output-disabled for each group. The status flag is POEGGn.IOCF, which is shared with GPT output-disable detection.

22.3.4 Output-Disable Control Using Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while POEGGn.OSTPE is 1, the GPT output pins are output-disabled for each group.

22.3.5 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing to the Software Stop flag, POEGGn.SSF.

22.3.6 Release from Output-Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset or clear all of the following flags:

- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF.

Writing 0 to the POEGGn.PIDF flag is ignored (the flag is not cleared) if the external input pins, GTETRGn, are not disabled and the POEGGn.ST bit is not set to 0.

Writing 0 to the POEGGn.IOCF flag is valid (the flag is cleared) only if all of the GTST.DTEF, GTST.OABHF, and GTST.OABLF flags in the GPT are set to 0.

Writing 0 to the POEGGn.OSTPF flag is ignored (the flag is not cleared) if the OSTDSR.OSTDF flag in the clock generation circuit is not set to 0. In addition, when the flag set and release occur at the same time, the flag set takes precedence.

Figure 22.3 shows the release timing for output-disable. The output-disable is released at the beginning of the next count cycle of the GPT after the flag is cleared.

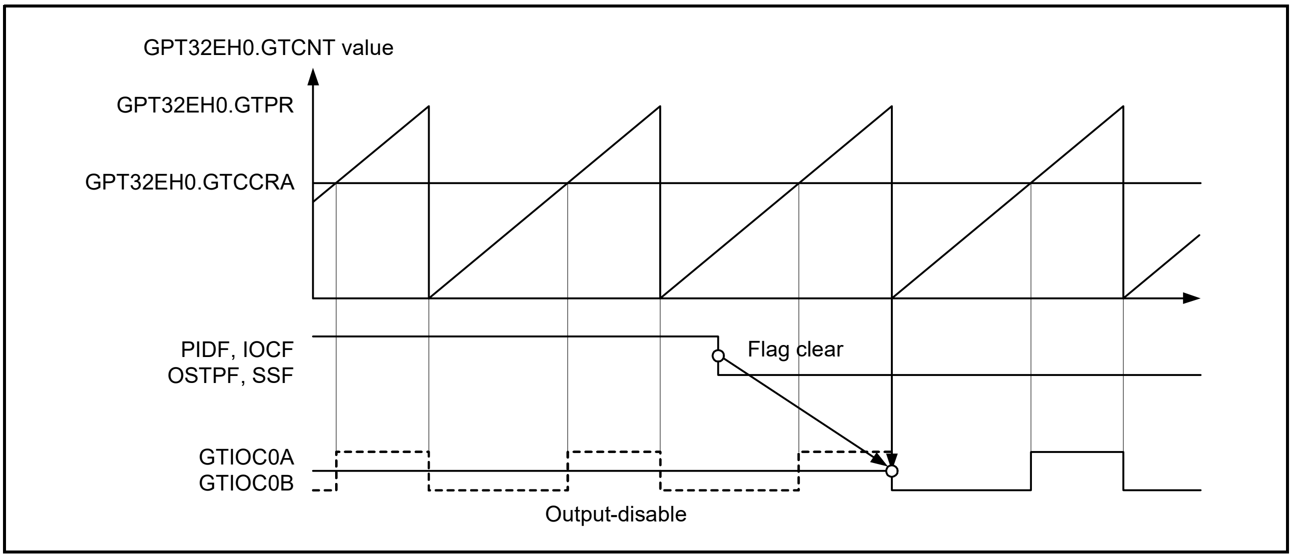


Figure 22.3 Output-disable release timing for GPT pin outputs

22.4 Interrupt Sources

The POEG generates an interrupt request when triggered by these sources:

- Output-disable control by input level detection
- Output-disable request from the GPT
- Comparator interrupt request detection.

Table 22.3 lists the conditions for interrupt requests.

Table 22.3 Interrupt sources and conditions

Interrupt source	Symbol	Associated flag	Trigger conditions
POEG group A interrupt	POEG_GROUP0	POEGGA.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt occurred
		POEGGA.PIDF	An output-disable request from the GTETRGA pin occurred
POEG group B interrupt	POEG_GROUP1	POEGGB.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt occurred
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred
POEG group C interrupt	POEG_GROUP2	POEGGC.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt occurred
		POEGGC.PIDF	An output-disable request from the GTETRGC pin occurred
POEG group D interrupt	POEG_GROUP3	POEGGD.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt occurred
		POEGGD.PIDF	An output-disable request from the GTETRGD pin occurred

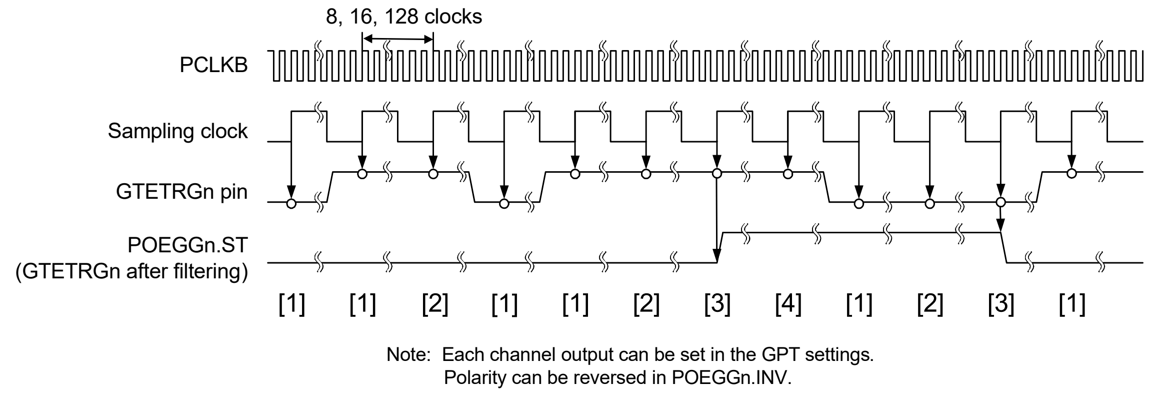
22.5 External Trigger Output to the GPT

The POEG outputs the GTETRGn signals as the GPT operation trigger signal for the following:

- Count start
- Count stop
- Count clear
- Up-count
- Down-count
- Input capture.

For the POEGGn.INV polarity setting signal, when the same level is input three times continuously with the sampling clock selected in POEGGn.NFCS[1:0] and POEGGn.NFEN, that value is output. Set the control registers the same as for the input level detection operation described in section 22.3.1, Pin Input Level Detection Operation. The state after filtering can be monitored in POEGGn.ST.

Figure 22.4 shows the output timing of an external trigger to the GPT.



Note: Each channel output can be set in the GPT settings.
Polarity can be reversed in POEGGn.INV.

Figure 22.4 Output timing of external trigger to GPT

22.6 Usage Notes

22.6.1 Transition to Software Standby Mode

When using the POEG, do not invoke Software Standby mode. In this mode, the POEG stops and therefore output-disable of the pins cannot be controlled.

22.6.2 Specifying Pins Associated with the GPT

The POEG controls output-disable only when a pin is associated with the GPT in the PmnPFS.PMR and PmnPFS.PSEL settings. When the pin is specified as a general I/O pin, the POEG does not perform output-disable control.

23. General PWM Timer (GPT)

23.1 Overview

The General PWM Timer (GPT) is a 32-bit timer with six GPT32 channels, four GPT32E channels, and four GPT32EH channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. The GPT can also be used as a general-purpose timer.

Table 23.1 lists the GPT specifications, Table 23.2 shows the GPT functions, Figure 23.1 shows a block diagram, Figure 23.2 shows the correspondence between the GPT channels and module names, and Table 23.3 lists the I/O pins.

Table 23.1 GPT specifications

Parameter	Specifications
Functions	• 32 bits × 14 channels • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter • Clock sources independently selectable for each channel • Two I/O pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms • Registers for setting up frame cycles in each channel (with capability for generating interrupts at overflow or underflow) • Generation of dead times in PWM operation • Synchronous starting, stopping, and clearing counters for arbitrary channels • Starting, stopping, and clearing up/down counters in response to a maximum of eight ELC events • Starting, stopping, and clearing up/down counters in response to input level comparison • Starting, stopping, and clearing up/down counters in response to a maximum of four external triggers • Output pin disable function by dead time error and detected short-circuits between output pins • A/D converter start triggers can be generated • PWM waveform for controlling brushless DC motors can be generated • Compare match A to F event, overflow/underflow event, and input UVW edge event can be output to the ELC • Enables the noise filter for input capture and input UVW • Bus clock: PCLKA • Core clock: PCLKD • Frequency ratio: PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64).

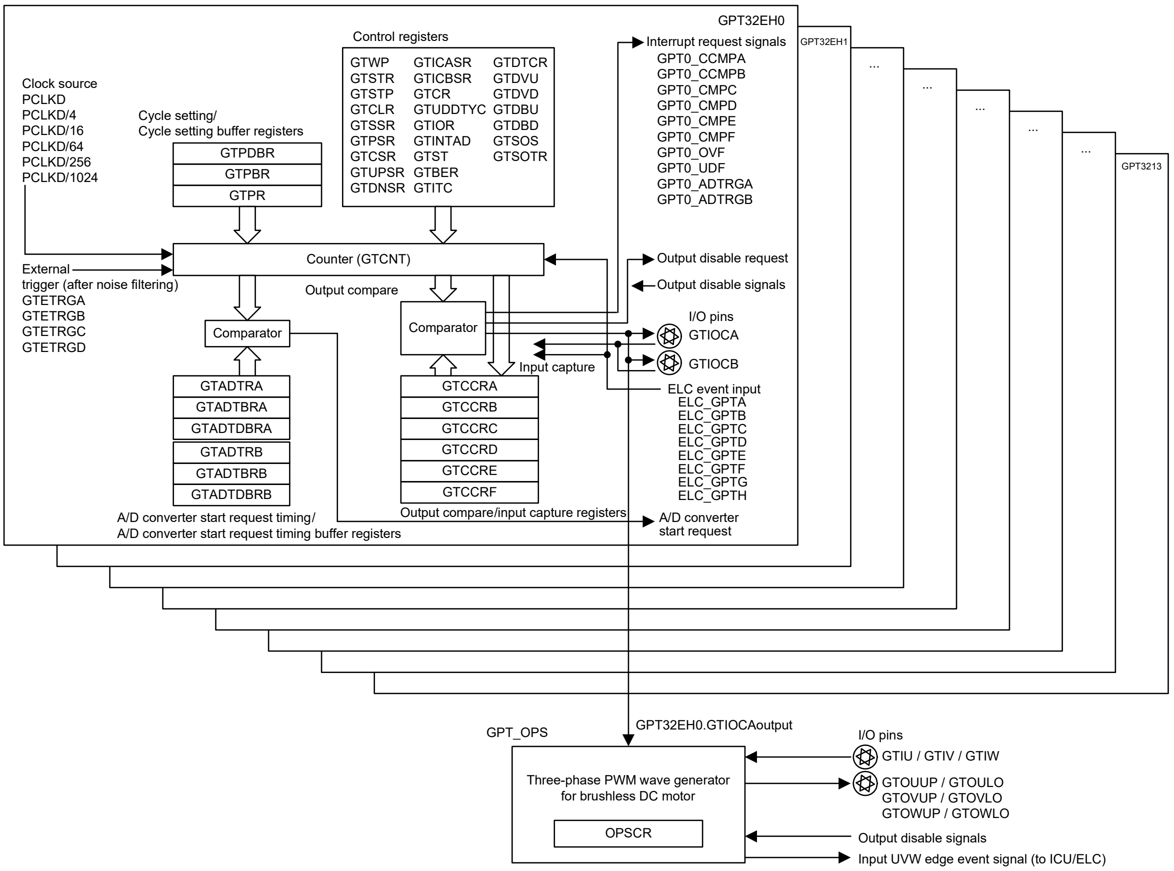
Table 23.2 GPT functions (1 of 2)

Parameter	GPT32EH, GPT32E	GPT32
Count clock	PCLKD PCLKD/4 PCLKD/16 PCLKD/64 PCLKD/256 PCLKD/1024	PCLKD PCLKD/4 PCLKD/16 PCLKD/64 PCLKD/256 PCLKD/1024
Output compare/input capture registers (GTCCR)	GTCCRA GTCCRB	GTCCRA GTCCRB
Compare/buffer registers	GTCCRC GTCCRD GTCCRE GTCCRF	GTCCRC GTCCRD GTCCRE GTCCRF
Cycle setting register	GTPR	GTPR
Cycle setting buffer registers	GTPBR GTPDBR	GTPBR
I/O pins	GTIOCA GTIOCB	GTIOCA GTIOCB

Table 23.2 GPT functions (2 of 2)

Parameter		GPT32EH, GPT32E	GPT32
External trigger input pin*1		GTETRGA GTETRGB GTETRGC GTETRGD	GTETRGA GTETRGB GTETRGC GTETRGD
Counter clear sources		GTPR register compare match, input capture, input pin status, ELC event input, and GTETRGn (n = A, B, C, D) pin input	GTPR register compare match, input capture, input pin status, ELC event input, and GTETRGn (n = A, B, C, D) pin input
Compare match output	Low output	Available	Available
	High output	Available	Available
	Toggle output	Available	Available
Input capture function		Available	Available
Automatic addition of dead time		Available	Available (no dead time buffer)
PWM mode		Available	Available
Phase count function		Available	Available
Buffer operation		Double buffer	Double buffer
One-shot operation		Available	Available
DTC activation		All the interrupt sources	All the interrupt sources
A/D converter start trigger		Compare match of GTADTRA or GTADTRB	-
Brushless DC motor control function		Available	Available
Interrupt sources		10 sources • GTCCRA compare match/input capture (GPTn_CCMPA) • GTCCRB compare match/input capture (GPTn_CCMPB) • GTCCRC compare match (GPTn_CMPC) • GTCCRD compare match (GPTn_CMPD) • GTCCRE compare match (GPTn_CMPCE) • GTCCRF compare match (GPTn_CMPF) • GTADTRA compare match (GPTn_ADTRGA) • GTADTRB compare match (GPTn_ADTRGB) • GTCNT overflow (GTPR compare match) (GPTn_OVF) • GTCNT underflow (GPTn_UDF)	8 sources • GTCCRA compare match/input capture (GPTn_CCMPA) • GTCCRB compare match/input capture (GPTn_CCMPB) • GTCCRC compare match (GPTn_CMPC) • GTCCRD compare match (GPTn_CMPD) • GTCCRE compare match (GPTn_CMPE) • GTCCRF compare match (GPTn_CMPF) • GTCNT overflow (GTPR compare match) (GPTn_OVF) • GTCNT underflow (GPTn_UDF)
Interrupt skipping function		Skips GTCNT overflows (GTPR compare match) (GPTn_OVF)/ GTCNT underflow (GPTn_UDF) interrupts (with interlocking function for other interrupts or A/D conversion requests).	-
Event linking (ELC) function		Available	Available
Noise filtering function		Available	Available

Note 1. GTRETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPD14 bit.



GTWP:	General PWM Timer Write-Protection Register
GTSTR:	General PWM Timer Software Start Register
GTSTP:	General PWM Timer Software Stop Register
GTCLR:	General PWM Timer Software Clear Register
GTSSR:	General PWM Timer Start Source Select Register
GTPSR:	General PWM Timer Stop Source Select Register
GTCSR:	General PWM Timer Clear Source Select Register
GTUPSR:	General PWM Timer Up Count Source Select Register
GTDNSR:	General PWM Timer Down Count Source Select Register
GTICASR:	General PWM Timer Input Capture Source Select Register A
GTICBSR:	General PWM Timer Input Capture Source Select Register B
GTCR:	General PWM Timer Control Register
GTUDDTYC:	General PWM Timer Count Direction and Duty Setting Register
GTIOR:	General PWM Timer I/O Control Register
GTINTAD:	General PWM Timer Interrupt Output Setting Register
GTST:	General PWM Timer Status Register
GTBER:	General PWM Timer Buffer Enable Register
GTITC:	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register
GTCNT:	General PWM Timer Counter
GTCCRA:	General PWM Timer Compare Capture Register A
GTCCRB:	General PWM Timer Compare Capture Register B
GTCCRC:	General PWM Timer Compare Capture Register C
GTCCRD:	General PWM Timer Compare Capture Register D
GTCCRE:	General PWM Timer Compare Capture Register E
GTCCRF:	General PWM Timer Compare Capture Register F
GTPR:	General PWM Timer Cycle Setting Register
GTPBR:	General PWM Timer Cycle Setting Buffer Register
GTPDBR:	General PWM Timer Cycle Setting Double-Buffer Register
GTADTRA:	General PWM Timer A/D Converter Start Request Timing Register A
GTADTBRA:	General PWM Timer A/D Converter Start Request Timing Buffer Register A
GTADTDBRA:	General PWM Timer A/D Converter Start Request Timing Double-Buffer Register A
GTADTRB:	General PWM Timer A/D Converter Start Request Timing Register B
GTADTBRB:	General PWM Timer A/D Converter Start Request Timing Buffer Register B
GTADTDBRB:	General PWM Timer A/D Converter Start Request Timing Double-Buffer Register B
GTDTCR:	General PWM Timer Dead Time Control Register
GTDVU:	General PWM Timer Dead Time Value Register U
GTDVD:	General PWM Timer Dead Time Value Register D
GTDBU:	General PWM Timer Dead Time Buffer Register U
GTDBD:	General PWM Timer Dead Time Buffer Register D
GTSOS:	General PWM Timer Output Protection Function Status Register
GTSOTR:	General PWM Timer Output Protection Function Temporary Release Register
OPSCR:	Output Phase Switching Control Register

Figure 23.1 GPT block diagram

Figure 23.2 shows an example using multiple GPTs.

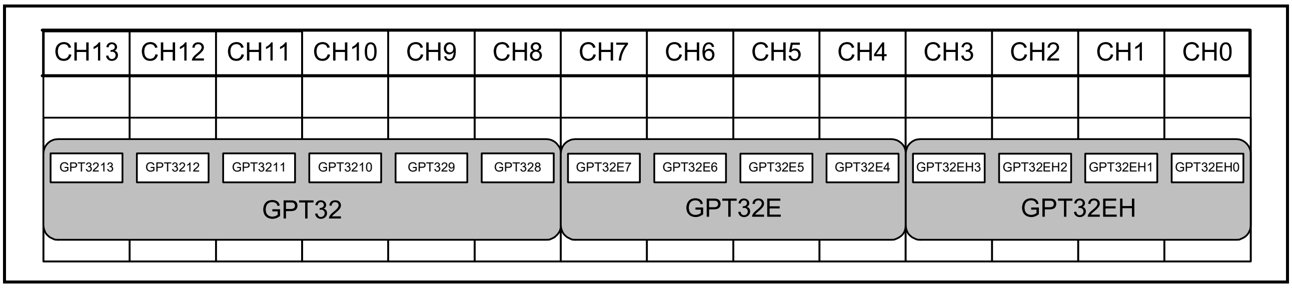


Figure 23.2 Correspondence between GPT channels and module names

Table 23.3 GPT I/O pins (1 of 2)

Channel	Pin name	I/O	Function
Shared	GTETRGA	Input	External trigger input pin A (after noise filtering)
	GTETRGB	Input	External trigger input pin B (after noise filtering)
	GTETRGC	Input	External trigger input pin C (after noise filtering)
	GTETRGD	Input	External trigger input pin D (after noise filtering)
GPT32EH0	GTIOC0A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC0B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT32EH1	GTIOC1A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC1B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT32EH2	GTIOC2A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC2B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT32EH3	GTIOC3A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC3B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT32E4	GTIOC4A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC4B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT32E5	GTIOC5A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC5B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT32E6	GTIOC6A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC6B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT32E7	GTIOC7A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC7B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT328	GTIOC8A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC8B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT329	GTIOC9A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC9B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT3210	GTIOC10A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC10B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT3211	GTIOC11A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC11B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT3212	GTIOC12A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC12B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT3213	GTIOC13A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC13B	I/O	GTCCRB register input capture input/output compare output/PWM output pin

Table 23.3 GPT I/O pins (2 of 2)

Channel	Pin name	I/O	Function
GPT_OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

23.2 Register Descriptions

Table 23.4 lists the registers in the GPT.

Table 23.4 GPT registers (1 of 2)

Module symbol	Register name	Register symbol	Reset value	Address (m = 0 to 13)	Access size	GPT32EH/ GPT32E	GPT32
GPT32EHm (m = 0 to 3) GPT32Em (m = 4 to 7) GPT32m (m = 8 to 13)	General PWM Timer Write Protection Register	GTWP	0000_0000h	4007 8000h + 0100h × m	32	✓	✓
	General PWM Timer Software Start Register	GTSTR	0000_0000h	4007 8004h + 0100h × m	32	✓	✓
	General PWM Timer Software Stop Register	GTSTP	FFFF_FFFFh	4007 8008h + 0100h × m	32	✓	✓
	General PWM Timer Software Clear Register	GTCLR	0000_0000h	4007 800Ch + 0100h × m	32	✓	✓
	General PWM Timer Start Source Select Register	GTSSR	0000_0000h	4007 8010h + 0100h × m	32	✓	✓
	General PWM Timer Stop Source Select Register	GTPSR	0000_0000h	4007 8014h + 0100h × m	32	✓	✓
	General PWM Timer Clear Source Select Register	GTCSR	0000_0000h	4007 8018h + 0100h × m	32	✓	✓
	General PWM Timer Up Count Source Select Register	GTUPSR	0000_0000h	4007 801Ch + 0100h × m	32	✓	✓
	General PWM Timer Down Count Source Select Register	GTDNSR	0000_0000h	4007 8020h + 0100h × m	32	✓	✓
	General PWM Timer Input Capture Source Select Register A	GTICASR	0000_0000h	4007 8024h + 0100h × m	32	✓	✓
	General PWM Timer Input Capture Source Select Register B	GTICBSR	0000_0000h	4007 8028h + 0100h × m	32	✓	✓
	General PWM Timer Control Register	GTCR	0000_0000h	4007 802Ch + 0100h × m	32	✓	✓
	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	0000_0001h	4007 8030h + 0100h × m	32	✓	✓
	General PWM Timer I/O Control Register	GTIOR	0000_0000h	4007 8034h + 0100h × m	32	✓	✓
	General PWM Timer Interrupt Output Setting Register	GTINTAD	0000_0000h	4007 8038h + 0100h × m	32	✓	(✓)*1
	General PWM Timer Status Register	GTST	0000_8000h	4007 803Ch + 0100h × m	32	✓	(✓)*1
	General PWM Timer Buffer Enable Register	GTBER	0000_0000h	4007 8040h + 0100h × m	32	✓	(✓)*1
GPT32EHm (m = 0 to 3) GPT32Em (m = 4 to 7)	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	GTITC	0000_0000h	4007 8044h + 0100h × m	32	✓	-

Table 23.4 GPT registers (2 of 2)

Module symbol	Register name	Register symbol	Reset value	Address (m = 0 to 13)	Access size	GPT32EH/ GPT32E	GPT32
GPT32EHm (m = 0 to 3) GPT32Em (m = 4 to 7) GPT32m (m = 8 to 13)	General PWM Timer Counter	GTCNT	0000_0000h	4007 8048h + 0100h × m	32	✓	✓
	General PWM Timer Compare Capture Register A	GTCCRA	FFFF_FFFFh	4007 804Ch + 0100h × m	32	✓	✓
	General PWM Timer Compare Capture Register B	GTCCRB	FFFF_FFFFh	4007 8050h + 0100h × m	32	✓	✓
	General PWM Timer Compare Capture Register C	GTCCRC	FFFF_FFFFh	4007 8054h + 0100h × m	32	✓	✓
	General PWM Timer Compare Capture Register E	GTCCRE	FFFF_FFFFh	4007 8058h + 0100h × m	32	✓	✓
	General PWM Timer Compare Capture Register D	GTCCRD	FFFF_FFFFh	4007 805Ch + 0100h × m	32	✓	✓
	General PWM Timer Compare Capture Register F	GTCCRF	FFFF_FFFFh	4007 8060h + 0100h × m	32	✓	✓
	General PWM Timer Cycle Setting Register	GTPR	FFFF_FFFFh	4007 8064h + 0100h × m	32	✓	✓
	General PWM Timer Cycle Setting Buffer Register	GTPBR	FFFF_FFFFh	4007 8068h + 0100h × m	32	✓	✓
GPT32EHm (m = 0 to 3) GPT32Em (m = 4 to 7)	General PWM Timer Cycle Setting Double-Buffer Register	GTPDBR	FFFF_FFFFh	4007 806Ch + 0100h × m	32	✓	-
	A/D Converter Start Request Timing Register A	GTADTRA	FFFF_FFFFh	4007 8070h + 0100h × m	32	✓	-
	A/D Converter Start Request Timing Buffer Register A	GTADTBRA	FFFF_FFFFh	4007 8074h + 0100h × m	32	✓	-
	A/D Converter Start Request Timing Double-Buffer Register A	GTADTDBRA	FFFF_FFFFh	4007 8078h + 0100h × m	32	✓	-
	A/D Converter Start Request Timing Register B	GTADTRB	FFFF_FFFFh	4007 807Ch + 0100h × m	32	✓	-
	A/D Converter Start Request Timing Buffer Register B	GTADTBRB	FFFF_FFFFh	4007 8080h + 0100h × m	32	✓	-
	A/D Converter Start Request Timing Double-Buffer Register B	GTADTDBRB	FFFF_FFFFh	4007 8084h + 0100h × m	32	✓	-
GPT32EHm (m = 0 to 3) GPT32Em (m = 4 to 7) GPT32m (m = 8 to 13)	General PWM Timer Dead Time Control Register	GTDTCR	0000_0000h	4007 8088h + 0100h × m	32	✓	(✓)*1
	General PWM Timer Dead Time Value Register U	GTDVU	FFFF_FFFFh	4007 808Ch + 0100h × m	32	✓	✓
GPT32EHm (m = 0 to 3) GPT32Em (m = 4 to 7)	General PWM Timer Dead Time Value Register D	GTDVD	FFFF_FFFFh	4007 8090h + 0100h × m	32	✓	-
	General PWM Timer Dead Time Buffer Register U	GTDBU	FFFF_FFFFh	4007 8094h + 0100h × m	32	✓	-
	General PWM Timer Dead Time Buffer Register D	GTDBD	FFFF_FFFFh	4007 8098h + 0100h × m	32	✓	-
	General PWM Timer Output Protection Function Status Register	GTSOS	0000_0000h	4007 809Ch + 0100h × m	32	✓	-
	General PWM Timer Output Protection Function Temporary Release Register	GTSOTR	0000_0000h	4007 80A0h + 0100h × m	32	✓	-
GPT_OPS	Output Phase Switching Control Register	OPSCR	0000_0000h	4007 8FF0h	32	✓	✓

Note 1. Some functions are reduced from GPT32EH/GPT32E.

23.2.1 General PWM Timer Write-Protection Register (GTWP)

Address(es): GPT32EHm.GTWP 4007 8000h + 0100h × m (m = 0 to 3)
GPT32Em.GTWP 4007 8000h + 0100h × m (m = 4 to 7)
GPT32m.GTWP 4007 8000h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	—	—	—	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	WP	Register Write Disable	0: Enable writes to the affected registers 1: Disable writes to the affected registers.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	GTWP Key Code	When A5h is written to these bits, writing to the WP bit is permitted. These bits are read as 0.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

To prevent accidental changes, the GTWP register enables or disables writing to registers. The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR.

23.2.2 General PWM Timer Software Start Register (GTSTR)

Address(es): GPT32EHm.GTSTR 4007 8004h + 0100h × m (m = 0 to 3)
GPT32Em.GTSTR 4007 8004h + 0100h × m (m = 4 to 7)
GPT32m.GTSTR 4007 8004h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CSTRT 13	CSTRT 12	CSTRT 11	CSTRT 10	CSTRT 9	CSTRT 8	CSTRT 7	CSTRT 6	CSTRT 5	CSTRT 4	CSTRT 3	CSTRT 2	CSTRT 1	CSTRT 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTSTR starts the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTSTR bit number represents the channel number. The GTSTR register is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTR register. For the association between the GTSTR bit number and a channel number, see Figure 23.2.

CSTRTn bit (Channel n GTCNT Count Start) (n = 0 to 13)

The CSTRTn bit starts channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 13) has no effect unless the GPTm.GTSSR.CSTRT bit is set to 1 (for GPT32EH, m = EH0 to EH3, for GPT32E, m = E4 to E7, for GPT32, m = 8 to 13).

Read data shows the counter status of each channel (GTCR.CST bit). Zero means the counter is stopped and 1 means the counter is running.

23.2.3 General PWM Timer Software Stop Register (GTSTP)

Address(es): GPT32EHm.GTSTP 4007 8008h + 0100h × m (m = 0 to 3)
GPT32Em.GTSTP 4007 8008h + 0100h × m (m = 4 to 7)
GPT32m.GTSTP 4007 8008h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CSTOP 13	CSTOP 12	CSTOP 11	CSTOP 10	CSTOP 9	CSTOP 8	CSTOP 7	CSTOP 6	CSTOP 5	CSTOP 4	CSTOP 3	CSTOP 2	CSTOP 1	CSTOP 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTSTP register stops the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTSTP bit number represents the channel number. The GTSTP register is shared by all of the channels. The GTCNT counter stops for the channel associated with the GTSTP bit in which 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTP register. For the association between the GTSTP bit number and a channel number, see Figure 23.2.

CSTOPn bit (Channel n GTCNT Count Stop) (n = 0 to 13)

The CSTOPn bit stops channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 13) has no effect unless the GPTm.GTPSR.CSTOP bit is set to 1 (for GPT32EH, m = EH0 to EH3, for GPT32E, m = E4 to E7, for GPT32, m = 8 to 13).

Read data shows the counter status of each channel (invert of the GTCR.CST bit). Zero means the counter is running and 1 means the counter stops.

23.2.4 General PWM Timer Software Clear Register (GTCLR)

Address(es): GPT32EHm.GTCLR 4007 800Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTCLR 4007 800Ch + 0100h × m (m = 4 to 7)
GPT32m.GTCLR 4007 800Ch + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CCLR 13	CCLR 12	CCLR 11	CCLR 10	CCLR 9	CCLR 8	CCLR 7	CCLR 6	CCLR 5	CCLR 4	CCLR 3	CCLR 2	CCLR 1	CCLR 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 13.

The GTCLR bit number represents the channel number. The GTCLR register is shared by all of the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter. For the association between the GTCLR bit number and a channel number, see Figure 23.2.

CCLRn bit (Channel n GTCNT Count Clear) (n = 0 to 13)

Channel n of the GTCNT counter value is cleared on writing 1 to the CCLRn bit. This bit is read as 0.

23.2.5 General PWM Timer Start Source Select Register (GTSSR)

Address(es): GPT32EHm.GTSSR 4007 8010h + 0100h × m (m = 0 to 3)
GPT32Em.GTSSR 4007 8010h + 0100h × m (m = 4 to 7)
GPT32m.GTSSR 4007 8010h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SSGTRGAR	GTETRGA Pin Rising Input Source Counter Start Enable	0: Disable counter start on the rising edge of GTETRGA input 1: Enable counter start on the rising edge of GTETRGA input.	R/W
b1	SSGTRGAF	GTETRGA Pin Falling Input Source Counter Start Enable	0: Disable counter start on the falling edge of GTETRGA input 1: Enable counter start on the falling edge of GTETRGA input.	R/W
b2	SSGTRGBR	GTETRGB Pin Rising Input Source Counter Start Enable	0: Disable counter start on the rising edge of GTETRGB input 1: Enable counter start on the rising edge of GTETRGB input.	R/W
b3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable	0: Disable counter start on the falling edge of GTETRGB input 1: Enable counter start on the falling edge of GTETRGB input.	R/W
b4	SSGTRGCR	GTETRGC Pin Rising Input Source Counter Start Enable	0: Disable counter start on the rising edge of GTETRGC input 1: Enable counter start on the rising edge of GTETRGC input.	R/W
b5	SSGTRGCF	GTETRGC Pin Falling Input Source Counter Start Enable	0: Disable counter start on the falling edge of GTETRGC input 1: Enable counter start on the falling edge of GTETRGC input	R/W
b6	SSGTRGDR	GTETRGD Pin Rising Input Source Counter Start Enable	0: Disable counter start on the rising edge of GTETRGD input 1: Enable counter start on the rising edge of GTETRGD input.	R/W
b7	SSGTRGDF	GTETRGD Pin Falling Input Source Counter Start Enable	0: Disable counter start on the falling edge of GTETRGD input 1: Enable counter start on the falling edge of GTETRGD input.	R/W
b8	SSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Start Enable	0: Disable counter start on the rising edge of GTIOCA input when GTIOCB input is 0 1: Enable counter start on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	SSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Start Enable	0: Disable counter start on the rising edge of GTIOCA input when GTIOCB input is 1 1: Enable counter start on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	SSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Start Enable	0: Disable counter start on the falling edge of GTIOCA input when GTIOCB input is 0 1: Enable counter start on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	SSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Start Enable	0: Disable counter start on the falling edge of GTIOCA input when GTIOCB input is 1 1: Enable counter start on the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	SSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Start Enable	0: Disable counter start on the rising edge of GTIOCB input when GTIOCA input is 0 1: Enable counter start on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	SSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Start Enable	0: Disable counter start on the rising edge of GTIOCB input when GTIOCA input is 1 1: Enable counter start on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	SSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Start Enable	0: Disable counter start on the falling edge of GTIOCB input when GTIOCA input is 0 1: Enable counter start on the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	SSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Start Enable	0: Disable counter start on the falling edge of GTIOCB input when GTIOCA input is 1 1: Enable counter start on the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	SSELCA	ELC_GPTA Event Source Counter Start Enable	0: Disable counter start on ELC_GPTA event input 1: Enable counter start on ELC_GPTA event input.	R/W
b17	SSELCB	ELC_GPTB Event Source Counter Start Enable	0: Disable counter start on ELC_GPTB event input 1: Enable counter start on ELC_GPTB event input.	R/W
b18	SSELCC	ELC_GPTC Event Source Counter Start Enable	0: Disable counter start on ELC_GPTC event input 1: Enable counter start on ELC_GPTC event input.	R/W
b19	SSELCD	ELC_GPTD Event Source Counter Start Enable	0: Disable counter start on ELC_GPTD event input 1: Enable counter start on ELC_GPTD event input.	R/W
b20	SSELCE	ELC_GPTE Event Source Counter Start Enable	0: Disable counter start on ELC_GPTE event input 1: Enable counter start on ELC_GPTE event input.	R/W
b21	SSELCF	ELC_GPTF Event Source Counter Start Enable	0: Disable counter start on ELC_GPTF event input 1: Enable counter start on ELC_GPTF event input.	R/W
b22	SSELCG	ELC_GPTG Event Source Counter Start Enable	0: Disable counter start on ELC_GPTG event input 1: Enable counter start on ELC_GPTG event input.	R/W
b23	SSELCH	ELC_GPTH Event Source Counter Start Enable	0: Disable counter start on ELC_GPTH event input 1: Enable counter start on ELC_GPTH event input.	R/W
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTRT	Software Source Counter Start Enable	0: Disable counter start by the GTSTR register 1: Enable counter start by the GTSTR register.	R/W

The GTSSR register sets the source to start the GTCNT counter.

SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)

The SSGTRGAR bit enables or disables GTCNT counter start on the rising edge of the GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

The SSGTRGAF bit enables or disables GTCNT counter start on the falling edge of the GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

The SSGTRGBR bit enables or disables GTCNT counter start on the rising edge of the GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

The SSGTRGBF bit enables or disables GTCNT counter start on the falling edge of the GTETRGB pin input.

SSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Start Enable)

The SSGTRGCR bit enables or disables GTCNT counter start on the rising edge of the GTETRGC pin input.

SSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Start Enable)

The SSGTRGCF bit enables or disables GTCNT counter start on the falling edge of the GTETRGC pin input.

SSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Start Enable)

The SSGTRGDR bit enables or disables GTCNT counter start on the rising edge of the GTETRGD pin input.

SSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Start Enable)

The SSGTRGDF bit enables or disables GTCNT counter start on the falling edge of the GTETRGD pin input.

SSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Start Enable)

The SSCARBL bit enables or disables GTCNT counter start on the rising edge of the GTIOCA pin input when the GTIOCB input is 0.

SSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Start Enable)

The SSCARBH bit enables or disables GTCNT counter start on the rising edge of the GTIOCA pin input when the GTIOCB input is 1.

SSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Start Enable)

The SSCAFBL bit enables or disables GTCNT counter start on the falling edge of the GTIOCA pin input when the GTIOCB input is 0.

SSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Start Enable)

The SSCAFBH bit enables or disables GTCNT counter start on the falling edge of the GTIOCA pin input when the GTIOCB input is 1.

SSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Start Enable)

The SSCBRAL bit enables or disables GTCNT counter start on the rising edge of the GTIOCB pin input when the GTIOCA input is 0.

SSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Start Enable)

The SSCBRAH bit enables or disables GTCNT counter start on the rising edge of the GTIOCB pin input when the GTIOCA input is 1.

SSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Start Enable)

The SSCBFAL bit enables or disables GTCNT counter start on the falling edge of the GTIOCB pin input when the GTIOCA input is 0.

SSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Start Enable)

The SSCBFAH bit enables or disables GTCNT counter start on the falling edge of the GTIOCB pin input when the GTIOCA input is 1.

SSELCm bit (ELC_GPTm Event Source Counter Start Enable) (m = A to H)

The SSELCm bit enables or disables GTCNT counter start on the ELC_GPTm event input.

CSTRT bit (Software Source Counter Start Enable)

The CSTRT bit enables or disables GTCNT counter start by the GTSTR register.

23.2.6 General PWM Timer Stop Source Select Register (GTPSR)

Address(es): GPT32EHm.GTPSR 4007 8014h + 0100h × m (m = 0 to 3)
GPT32Em.GTPSR 4007 8014h + 0100h × m (m = 4 to 7)
GPT32m.GTPSR 4007 8014h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTETRGA input 1: Enable counter stop on the rising edge of GTETRGA input.	R/W
b1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTETRGA input 1: Enable counter stop on the falling edge of GTETRGA input.	R/W
b2	PSGTRGBR	GTETRGB Pin Rising Input Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTETRGB input 1: Enable counter stop on the rising edge of GTETRGB input.	R/W
b3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTETRGB input 1: Enable counter stop on the falling edge of GTETRGB input.	R/W
b4	PSGTRGCR	GTETRGC Pin Rising Input Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTETRGC input 1: Enable counter stop on the rising edge of GTETRGC input.	R/W
b5	PSGTRGCF	GTETRGC Pin Falling Input Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTETRGC input 1: Enable counter stop on the falling edge of GTETRGC input.	R/W
b6	PSGTRGDR	GTETRGD Pin Rising Input Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTETRGD input 1: Enable counter stop on the rising edge of GTETRGD input.	R/W
b7	PSGTRGDF	GTETRGD Pin Falling Input Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTETRGD input 1: Enable counter stop on the falling edge of GTETRGD input.	R/W
b8	PSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTIOCA input when GTIOCB input is 0 1: Enable counter stop on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	PSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTIOCA input when GTIOCB input is 1 1: Enable counter stop on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	PSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTIOCA input when GTIOCB input is 0 1: Enable counter stop on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	PSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTIOCA input when GTIOCB input is 1 1: Enable counter stop on the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	PSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTIOCB input when GTIOCA input is 0 1: Enable counter stop on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	PSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Stop Enable	0: Disable counter stop on the rising edge of GTIOCB input when GTIOCA input is 1 1: Enable counter stop on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	PSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTIOCB input when GTIOCA input is 0 1: Enable counter stop on the falling edge of GTIOCB input when GTIOCA input is 0	R/W
b15	PSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Stop Enable	0: Disable counter stop on the falling edge of GTIOCB input when GTIOCA input is 1 1: Enable counter stop on the falling edge of GTIOCB input when GTIOCA input is 1	R/W
b16	PSELCA	ELC_GPTA Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTA event input 1: Enable counter stop on ELC_GPTA event input.	R/W
b17	PSELCB	ELC_GPTB Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTB event input 1: Enable counter stop on ELC_GPTB event input.	R/W
b18	PSELCC	ELC_GPTC Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTC event input 1: Enable counter stop on ELC_GPTC event input.	R/W
b19	PSELCD	ELC_GPTD Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTD event input 1: Enable counter stop on ELC_GPTD event input.	R/W
b20	PSELCE	ELC_GPTE Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTE event input 1: Enable counter stop on ELC_GPTE event input.	R/W
b21	PSELCF	ELC_GPTF Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTF event input 1: Enable counter stop on ELC_GPTF event input.	R/W
b22	PSELCG	ELC_GPTG Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTG event input 1: Enable counter stop on ELC_GPTG event input.	R/W
b23	PSELCH	ELC_GPTH Event Source Counter Stop Enable	0: Disable counter stop on ELC_GPTH event input 1: Enable counter stop on ELC_GPTH event input.	R/W
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTOP	Software Source Counter Stop Enable	0: Disable counter stop by the GTSTP register 1: Enable counter stop by the GTSTP register.	R/W

The GTPSR register sets the source to stop the GTCNT counter.

PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)

The PSGTRGAR bit enables or disables GTCNT counter stop on the rising edge of the GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

The PSGTRGAF bit enables or disables GTCNT counter stop on the falling edge of the GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

The PSGTRGBR bit enables or disables GTCNT counter stop on the rising edge of the GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

The PSGTRGBF bit enables or disables GTCNT counter stop on the falling edge of the GTETRGB pin input.

PSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Stop Enable)

The PSGTRGCR bit enables or disables GTCNT counter stop on the rising edge of the GTETRGC pin input.

PSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Stop Enable)

The PSGTRGCF bit enables or disables GTCNT counter stop on the falling edge of the GTETRGC pin input.

PSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Stop Enable)

The PSGTRGDR bit enables or disables GTCNT counter stop on the rising edge of the GTETRGD pin input.

PSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Stop Enable)

The PSGTRGDF bit enables or disables GTCNT counter stop on the falling edge of the GTETRGD pin input.

PSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Stop Enable)

The PSCARBL bit enables or disables GTCNT counter stop on the rising edge of the GTIOCA pin input when the GTIOCB input is 0.

PSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Stop Enable)

The PSCARBH bit enables or disables GTCNT counter stop on the rising edge of the GTIOCA pin input when the GTIOCB input is 1.

PSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Stop Enable)

The PSCAFBL bit enables or disables GTCNT counter stop on the falling edge of the GTIOCA pin input when the GTIOCB input is 0.

PSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Stop Enable)

The PSCAFBH bit enables or disables GTCNT counter stop on the falling edge of the GTIOCA pin input when the GTIOCB input is 1.

PSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Stop Enable)

The PSCBRAL bit enables or disables GTCNT counter stop on the rising edge of the GTIOCB pin input when the GTIOCA input is 0.

PSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Stop Enable)

The PSCBRAH bit enables or disables GTCNT counter stop on the rising edge of the GTIOCB pin input when the GTIOCA input is 1.

PSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Stop Enable)

The PSCBFAL bit enables or disables GTCNT counter stop on the falling edge of the GTIOCB pin input when the GTIOCA input is 0.

PSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Stop Enable)

The PSCBFAH bit enables or disables GTCNT counter stop on the falling edge of the GTIOCB pin input when the GTIOCA input is 1.

PSELCm bit (ELC_GPTm Event Source Counter Stop Enable) (m = A to H)

The PSELCm bit enables or disables GTCNT counter stop on the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

The CSTOP bit enables or disables GTCNT counter stop by the GTSTP register.

23.2.7 General PWM Timer Clear Source Select Register (GTCSR)

Address(es): GPT32EHm.GTCSR 4007 8018h + 0100h × m (m = 0 to 3)
GPT32Em.GTCSR 4007 8018h + 0100h × m (m = 4 to 7)
GPT32m.GTCSR 4007 8018h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CCLR	—	—	—	—	—	—	—	CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CSGTRGAR	GTETRGA Pin Rising Input Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTETRGA input 1: Enable counter clear on the rising edge of GTETRGA input.	R/W
b1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTETRGA input 1: Enable counter clear on the falling edge of GTETRGA input.	R/W
b2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear on the rising edge of GTETRGB input.	R/W
b3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTETRGB input 1: Enable counter clear on the falling edge of GTETRGB input.	R/W
b4	CSGTRGCR	GTETRGC Pin Rising Input Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTETRGC input 1: Enable counter clear on the rising edge of GTETRGC input.	R/W
b5	CSGTRGCF	GTETRGC Pin Falling Input Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTETRGC input 1: Enable counter clear on the falling edge of GTETRGC input.	R/W
b6	CSGTRGDR	GTETRGD Pin Rising Input Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTETRGD input 1: Enable counter clear on the rising edge of GTETRGD input.	R/W
b7	CSGTRGDF	GTETRGD Pin Falling Input Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTETRGD input 1: Enable counter clear on the falling edge of GTETRGD input.	R/W
b8	CSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTIOCA input when GTIOCB input is 0 1: Enable counter clear on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	CSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTIOCA input when GTIOCB input is 1 1: Enable counter clear on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	CSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTIOCA input when GTIOCB input is 0 1: Enable counter clear on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	CSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTIOCA input when GTIOCB input is 1 1: Enable counter clear on the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	CSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTIOCB input when GTIOCA input is 0 1: Enable counter clear on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	CSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Clear Enable	0: Disable counter clear on the rising edge of GTIOCB input when GTIOCA input is 1 1: Enable counter clear on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	CSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTIOCB input when GTIOCA input is 0 1: Enable counter clear on the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	CSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Clear Enable	0: Disable counter clear on the falling edge of GTIOCB input when GTIOCA input is 1 1: Enable counter clear on the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	CSELCA	ELC_GPTA Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTA event input 1: Enable counter clear on ELC_GPTA event input.	R/W
b17	CSELCB	ELC_GPTB Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTB event input 1: Enable counter clear on ELC_GPTB event input.	R/W
b18	CSELCC	ELC_GPTC Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTC event input 1: Enable counter clear on ELC_GPTC event input.	R/W
b19	CSELCD	ELC_GPTD Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTD event input 1: Enable counter clear on ELC_GPTD event input.	R/W
b20	CSELCE	ELC_GPTE Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTE event input 1: Enable counter clear on ELC_GPTE event input.	R/W
b21	CSELCF	ELC_GPTF Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTF event input 1: Enable counter clear on ELC_GPTF event input.	R/W
b22	CSELCG	ELC_GPTG Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTG event input 1: Enable counter clear on ELC_GPTG event input.	R/W
b23	CSELCH	ELC_GPTH Event Source Counter Clear Enable	0: Disable counter clear on ELC_GPTH event input 1: Enable counter clear on ELC_GPTH event input.	R/W
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CCLR	Software Source Counter Clear Enable	0: Disable counter clear by the GTCLR register 1: Enable counter clear by the GTCLR register.	R/W

GTCSR sets the source to clear the GTCNT counter.

CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)

The CSGTRGAR bit enables or disables GTCNT counter clear on the rising edge of the GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

The CSGTRGAF bit enables or disables GTCNT counter clear on the falling edge of the GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

The CSGTRGBR bit enables or disables GTCNT counter clear on the rising edge of the GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

The CSGTRGBF bit enables or disables GTCNT counter clear on the falling edge of the GTETRGB pin input.

CSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Clear Enable)

The CSGTRGCR bit enables or disables GTCNT counter clear on the rising edge of the GTETRGC pin input.

CSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Clear Enable)

The CSGTRGCF bit enables or disables GTCNT counter clear on the falling edge of the GTETRGC pin input.

CSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Clear Enable)

The CSGTRGDR bit enables or disables GTCNT counter clear on the rising edge of the GTETRGD pin input.

CSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Clear Enable)

The CSGTRGDF bit enables or disables GTCNT counter clear on the falling edge of the GTETRGD pin input.

CSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Clear Enable)

The CSCARBL bit enables or disables GTCNT counter clear on the rising edge of the GTIOCA pin input when the GTIOCB input is 0.

CSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Clear Enable)

The CSCARBH bit enables or disables GTCNT counter clear on the rising edge of the GTIOCA pin input when the GTIOCB input is 1.

CSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Clear Enable)

The CSCAFBL bit enables or disables GTCNT counter clear on the falling edge of the GTIOCA pin input when the GTIOCB input is 0.

CSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Clear Enable)

The CSCAFBH bit enables or disables GTCNT counter clear on the falling edge of the GTIOCA pin input when the GTIOCB input is 1.

CSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Clear Enable)

The CSCBRAL bit enables or disables GTCNT counter clear on the rising edge of the GTIOCB pin input when the GTIOCA input is 0.

CSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Clear Enable)

The CSCBRAH bit enables or disables GTCNT counter clear on the rising edge of the GTIOCB pin input when the GTIOCA input is 1.

CSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Clear Enable)

The CSCBFAL bit enables or disables GTCNT counter clear on the falling edge of the GTIOCB pin input when the GTIOCA input is 0.

CSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Clear Enable)

The CSCBFAH bit enables or disables GTCNT counter clear on the falling edge of the GTIOCB pin input when the GTIOCA input is 1.

CSELCm bit (ELC_GPTm Event Source Counter Clear Enable) (m = A to H)

The CSELCm bit enables or disables GTCNT counter clear on the ELC_GPTm event input.

CCLR bit (Software Source Counter Clear Enable)

The CCLR bit enables or disables GTCNT counter clear by the GTCLR register.

23.2.8 General PWM Timer Up Count Source Select Register (GTUPSR)

Address(es): GPT32EHm.GTUPSR 4007 801Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTUPSR 4007 801Ch + 0100h × m (m = 4 to 7)
GPT32m.GTUPSR 4007 801Ch + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	USGTRGAR	GTETRGA Pin Rising Input Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTETRGA input 1: Enable counter count up on the rising edge of GTETRGA input.	R/W
b1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTETRGA input 1: Enable counter count up on the falling edge of GTETRGA input.	R/W
b2	USGTRGBR	GTETRGB Pin Rising Input Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTETRGB input 1: Enable counter count up on the rising edge of GTETRGB input.	R/W
b3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTETRGB input 1: Enable counter count up on the falling edge of GTETRGB input.	R/W
b4	USGTRGCR	GTETRGC Pin Rising Input Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTETRGC input 1: Enable counter count up on the rising edge of GTETRGC input.	R/W
b5	USGTRGCF	GTETRGC Pin Falling Input Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTETRGC input 1: Enable counter count up on the falling edge of GTETRGC input.	R/W
b6	USGTRGDR	GTETRGD Pin Rising Input Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTETRGD input 1: Enable counter count up on the rising edge of GTETRGD input.	R/W
b7	USGTRGDF	GTETRGD Pin Falling Input Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTETRGD input 1: Enable counter count up on the falling edge of GTETRGD input.	R/W
b8	USCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTIOCA input when GTIOCB input is 0 1: Enable counter count up on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	USCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTIOCA input when GTIOCB input is 1 1: Enable counter count up on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	USCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTIOCA input when GTIOCB input is 0 1: Enable counter count up on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	USCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTIOCA input when GTIOCB input is 1 1: Enable counter count up on the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	USCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTIOCB input when GTIOCA input is 0 1: Enable counter count up on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	USCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Up Enable	0: Disable counter count up on the rising edge of GTIOCB input when GTIOCA input is 1 1: Enable counter count up on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	USCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTIOCB input when GTIOCA input is 0 1: Enable counter count up on the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	USCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Up Enable	0: Disable counter count up on the falling edge of GTIOCB input when GTIOCA input is 1 1: Enable counter count up on the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	USELCA	ELC_GPTA Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTA event input 1: Enable counter count up on ELC_GPTA event input.	R/W
b17	USELCB	ELC_GPTB Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTB event input 1: Enable counter count up on ELC_GPTB event input.	R/W
b18	USELCC	ELC_GPTC Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTC event input 1: Enable counter count up on ELC_GPTC event input.	R/W
b19	USELCD	ELC_GPTD Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTD event input 1: Enable counter count up on ELC_GPTD event input.	R/W
b20	USELCE	ELC_GPTE Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTE event input 1: Enable counter count up on ELC_GPTE event input.	R/W
b21	USELCF	ELC_GPTF Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTF event input 1: Enable counter count up on ELC_GPTF event input.	R/W
b22	USELCG	ELC_GPTG Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTG event input 1: Enable counter count up on ELC_GPTG event input.	R/W
b23	USELCH	ELC_GPTH Event Source Counter Count Up Enable	0: Disable counter count up on ELC_GPTH event input 1: Enable counter count up on ELC_GPTH event input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTUPSR register sets the source to count up the GTCNT counter.

When at least one bit in the GTUPSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)

The USGTRGAR bit enables or disables GTCNT counter count up on the rising edge of the GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

The USGTRGAF bit enables or disables GTCNT counter count up on the falling edge of the GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

The USGTRGBR bit enables or disables GTCNT counter count up on the rising edge of the GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

The USGTRGBF bit enables or disables GTCNT counter count up on the falling edge of the GTETRGB pin input.

USGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Up Enable)

The USGTRGCR bit enables or disables GTCNT counter count up on the rising edge of the GTETRGC pin input.

USGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Up Enable)

The USGTRGCF bit enables or disables GTCNT counter count up on the falling edge of the GTETRGC pin input.

USGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Up Enable)

The USGTRGDR bit enables or disables GTCNT counter count up on the rising edge of the GTETRGD pin input.

USGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Up Enable)

The USGTRGDF bit enables or disables GTCNT counter count up on the falling edge of the GTETRGD pin input.

USCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of the GTIOCA pin input when GTIOCB input is 0.

USCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Up Enable)

The USCARBH bit enables or disables GTCNT counter count up on the rising edge of the GTIOCA pin input when the GTIOCB input is 1.

USCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Up Enable)

The USCAFBL bit enables or disables GTCNT counter count up on the falling edge of the GTIOCA pin input when the GTIOCB input is 0.

USCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Up Enable)

The USCAFBH bit enables or disables GTCNT counter count up on the falling edge of the GTIOCA pin input when the GTIOCB input is 1.

USCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Up Enable)

The USCBRAL bit enables or disables GTCNT counter count up on the rising edge of the GTIOCB pin input when the GTIOCA input is 0.

USCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Up Enable)

The USCBRAH bit enables or disables GTCNT counter count up on the rising edge of the GTIOCB pin input when the GTIOCA input is 1.

USCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Up Enable)

The USCBFAL bit enables or disables GTCNT counter count up on the falling edge of the GTIOCB pin input when the GTIOCA input is 0.

USCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Up Enable)

The USCBFAH bit enables or disables GTCNT counter count up on the falling edge of the GTIOCB pin input when the GTIOCA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to H)

The USELCm bit enables or disables GTCNT counter count up on the ELC_GPTm event input.

23.2.9 General PWM Timer Down Count Source Select Register (GTDNSR)

Address(es): GPT32EHm.GTDNSR 4007 8020h + 0100h × m (m = 0 to 3)
GPT32Em.GTDNSR 4007 8020h + 0100h × m (m = 4 to 7)
GPT32m.GTDNSR 4007 8020h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTETRGA input 1: Enable counter count down on the rising edge of GTETRGA input.	R/W
b1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTETRGA input 1: Enable counter count down on the falling edge of GTETRGA input.	R/W
b2	DSGTRGBR	GTETRGB Pin Rising Input Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTETRGB input 1: Enable counter count down on the rising edge of GTETRGB input.	R/W
b3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTETRGB input 1: Enable counter count down on the falling edge of GTETRGB input.	R/W
b4	DSGTRGCR	GTETRGC Pin Rising Input Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTETRGC input 1: Enable counter count down on the rising edge of GTETRGC input.	R/W
b5	DSGTRGCF	GTETRGC Pin Falling Input Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTETRGC input 1: Enable counter count down on the falling edge of GTETRGC input.	R/W
b6	DSGTRGDR	GTETRGD Pin Rising Input Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTETRGD input 1: Enable counter count down on the rising edge of GTETRGD input.	R/W
b7	DSGTRGDF	GTETRGD Pin Falling Input Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTETRGD input 1: Enable counter count down on the falling edge of GTETRGD input.	R/W
b8	DSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTIOCA input when GTIOCB input is 0 1: Enable counter count down on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	DSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTIOCA input when GTIOCB input is 1 1: Enable counter count down on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	DSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTIOCA input when GTIOCB input is 0 1: Enable counter count down on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	DSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTIOCA input when GTIOCB input is 1 1: Enable counter count down on the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	DSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTIOCB input when GTIOCA input is 0 1: Enable counter count down on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	DSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Down Enable	0: Disable counter count down on the rising edge of GTIOCB input when GTIOCA input is 1 1: Enable counter count down on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	DSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTIOCB input when GTIOCA input is 0 1: Enable counter count down on the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	DSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Down Enable	0: Disable counter count down on the falling edge of GTIOCB input when GTIOCA input is 1 1: Enable counter count down on the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTA event input 1: Enable counter count down on ELC_GPTA event input.	R/W
b17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTB event input 1: Enable counter count down on ELC_GPTB event input.	R/W
b18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTC event input 1: Enable counter count down on ELC_GPTC event input.	R/W
b19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTD event input 1: Enable counter count down on ELC_GPTD event input.	R/W
b20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTE event input 1: Enable counter count down on ELC_GPTE event input.	R/W
b21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTF event input 1: Enable counter count down on ELC_GPTF event input.	R/W
b22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTG event input 1: Enable counter count down on ELC_GPTG event input.	R/W
b23	DSELCH	ELC_GPTH Event Source Counter Count Down Enable	0: Disable counter count down on ELC_GPTH event input 1: Enable counter count down on ELC_GPTH event input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDNSR register sets the source to count down the GTCNT counter.

When at least one bit in the GTDNSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

The DSGTRGAR bit enables or disables GTCNT counter count down on the rising edge of the GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

The DSGTRGAF bit enables or disables GTCNT counter count down on the falling edge of the GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

The DSGTRGBR bit enables or disables GTCNT counter count down on the rising edge of the GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

The DSGTRGBF bit enables or disables GTCNT counter count down on the falling edge of the GTETRGB pin input.

DSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Down Enable)

The DSGTRGCR bit enables or disables GTCNT counter count down on the rising edge of the GTETRGC pin input.

DSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Down Enable)

The DSGTRGCF bit enables or disables GTCNT counter count down on the falling edge of the GTETRGC pin input.

DSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Down Enable)

The DSGTRGDR bit enables or disables GTCNT counter count down on the rising edge of the GTETRGD pin input.

DSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Down Enable)

The DSGTRGDF bit enables or disables GTCNT counter count down on the falling edge of the GTETRGD pin input.

DSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Down Enable)

The DSCARBL bit enables or disables GTCNT counter count down on the rising edge of the GTIOCA pin input when the GTIOCB input is 0.

DSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Down Enable)

The DSCARBH bit enables or disables GTCNT counter count down on the rising edge of the GTIOCA pin input when the GTIOCB input is 1.

DSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Down Enable)

The DSCAFBL bit enables or disables GTCNT counter count down on the falling edge of the GTIOCA pin input when the GTIOCB input is 0.

DSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Down Enable)

The DSCAFBH bit enables or disables GTCNT counter count down on the falling edge of the GTIOCA pin input when the GTIOCB input is 1.

DSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Down Enable)

The DSCBRAL bit enables or disables GTCNT counter count down on the rising edge of the GTIOCB pin input when the GTIOCA input is 0.

DSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Down Enable)

The DSCBRAH bit enables or disables GTCNT counter count down on the rising edge of the GTIOCB pin input when the GTIOCA input is 1.

DSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Down Enable)

The DSCBFAL bit enables or disables GTCNT counter count down on the falling edge of the GTIOCB pin input when the GTIOCA input is 0.

DSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Down Enable)

The DSCBFAH bit enables or disables GTCNT counter count down on the falling edge of the GTIOCB pin input when the GTIOCA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to H)

The DSELCm bit enables or disables GTCNT counter count down on the ELC_GPTm event input.

23.2.10 General PWM Timer Input Capture Source Select Register A (GTICASR)

Address(es): GPT32EHm.GTICASR 4007 8024h + 0100h × m (m = 0 to 3)
GPT32Em.GTICASR 4007 8024h + 0100h × m (m = 4 to 7)
GPT32m.GTICASR 4007 8024h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ASGTRGAR	GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTETRGA input 1: Enable GTCCRA input capture on the rising edge of GTETRGA input.	R/W
b1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTETRGA input 1: Enable GTCCRA input capture on the falling edge of GTETRGA input.	R/W
b2	ASGTRGBR	GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTETRGB input 1: Enable GTCCRA input capture on the rising edge of GTETRGB input.	R/W
b3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTETRGB input 1: Enable GTCCRA input capture on the falling edge of GTETRGB input.	R/W
b4	ASGTRGCR	GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTETRGC input 1: Enable GTCCRA input capture on the rising edge of GTETRGC input.	R/W
b5	ASGTRGCF	GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTETRGC input 1: Enable GTCCRA input capture on the falling edge of GTETRGC input.	R/W
b6	ASGTRGDR	GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTETRGD input 1: Enable GTCCRA input capture on the rising edge of GTETRGD input.	R/W
b7	ASGTRGDF	GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTETRGD input 1: Enable GTCCRA input capture on the falling edge of GTETRGD input.	R/W
b8	ASCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTIOCA input when GTIOCB input is 0 1: Enable GTCCRA input capture on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	ASCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTIOCA input when GTIOCB input is 1 1: Enable GTCCRA input capture on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	ASCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTIOCA input when GTIOCB input is 0 1: Enable GTCCRA input capture on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	ASCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTIOCA input when GTIOCB input is 1 1: Enable GTCCRA input capture on the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	ASCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTIOCB input when GTIOCA input is 0 1: Enable GTCCRA input capture on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	ASCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the rising edge of GTIOCB input when GTIOCA input is 1 1: Enable GTCCRA input capture on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	ASCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTIOCB input when GTIOCA input is 0 1: Enable GTCCRA input capture on the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	ASCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on the falling edge of GTIOCB input when GTIOCA input is 1 1: Enable GTCCRA input capture on the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTA event input 1: Enable GTCCRA input capture on ELC_GPTA event input.	R/W
b17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTB event input 1: Enable GTCCRA input capture on ELC_GPTB event input.	R/W
b18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTC event input 1: Enable GTCCRA input capture on ELC_GPTC event input.	R/W
b19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTD event input 1: Enable GTCCRA input capture on ELC_GPTD event input.	R/W
b20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTE event input 1: Enable GTCCRA input capture on ELC_GPTE event input.	R/W
b21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTF event input 1: Enable GTCCRA input capture on ELC_GPTF event input.	R/W
b22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTG event input 1: Enable GTCCRA input capture on ELC_GPTG event input.	R/W
b23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable	0: Disable GTCCRA input capture on ELC_GPTH event input 1: Enable GTCCRA input capture on ELC_GPTH event input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTICASR register sets the source of input capture for GTCCRA.

ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGAR bit enables or disables input capture for GTCCRA on the rising edge of the GTETRGA pin input.

ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGAF bit enables or disables input capture for GTCCRA on the falling edge of the GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGBR bit enables or disables input capture for GTCCRA on the rising edge of the GTETRGB pin input.

ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGBF bit enables or disables input capture for GTCCRA on the falling edge of the GTETRGB pin input.

ASGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGCR bit enables or disables input capture for GTCCRA on the rising edge of the GTETRGC pin input.

ASGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGCF bit enables or disables input capture for GTCCRA on the falling edge of the GTETRGC pin input.

ASGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGDR bit enables or disables input capture for GTCCRA on the rising edge of the GTETRGD pin input.

ASGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGDF bit enables or disables input capture for GTCCRA on the falling edge of the GTETRGD pin input.

ASCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL bit enables or disables input capture for GTCCRA on the rising edge of the GTIOCA pin input when the GTIOCB input is 0.

ASCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRA Input Capture Enable)

The ASCARBH bit enables or disables input capture for GTCCRA on the rising edge of the GTIOCA pin input when the GTIOCB input is 1.

ASCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRA Input Capture Enable)

The ASCAFBL bit enables or disables input capture for GTCCRA on the falling edge of the GTIOCA pin input when the GTIOCB input is 0.

ASCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRA Input Capture Enable)

The ASCAFBH bit enables or disables input capture for GTCCRA on the falling edge of the GTIOCA pin input when the GTIOCB input is 1.

ASCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRA Input Capture Enable)

The ASCBRAL bit enables or disables input capture for GTCCRA on the rising edge of the GTIOCB pin input when the GTIOCA input is 0.

ASCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRA Input Capture Enable)

The ASCBRAH bit enables or disables input capture for GTCCRA on the rising edge of the GTIOCB pin input when the GTIOCA input is 1.

ASCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRA Input Capture Enable)

The ASCBFAL bit enables or disables input capture for GTCCRA on the falling edge of the GTIOCB pin input when the GTIOCA input is 0.

ASCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRA Input Capture Enable)

The ASCBFAH bit enables or disables input capture for GTCCRA on the falling edge of the GTIOCB pin input when the GTIOCA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to H)

The ASELCm bit enables or disables input capture for GTCCRA on the ELC_GPTm event input.

23.2.11 General PWM Timer Input Capture Source Select Register B (GTICBSR)

Address(es): GPT32EHm.GTICBSR 4007 8028h + 0100h × m (m = 0 to 3)
GPT32Em.GTICBSR 4007 8028h + 0100h × m (m = 4 to 7)
GPT32m.GTICBSR 4007 8028h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BSGTRGAR	GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTETRGA input 1: Enable GTCCRB input capture on the rising edge of GTETRGA input.	R/W
b1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTETRGA input 1: Enable GTCCRB input capture on the falling edge of GTETRGA input.	R/W
b2	BSGTRGBR	GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTETRGB input 1: Enable GTCCRB input capture on the rising edge of GTETRGB input.	R/W
b3	BSGTRGBF	GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTETRGB input 1: Enable GTCCRB input capture on the falling edge of GTETRGB input.	R/W
b4	BSGTRGCR	GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTETRGC input 1: Enable GTCCRB input capture on the rising edge of GTETRGC input.	R/W
b5	BSGTRGCF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTETRGC input 1: Enable GTCCRB input capture on the falling edge of GTETRGC input.	R/W
b6	BSGTRGDR	GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTETRGD input 1: Enable GTCCRB input capture on the rising edge of GTETRGD input.	R/W
b7	BSGTRGDF	GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTETRGD input 1: Enable GTCCRB input capture on the falling edge of GTETRGD input.	R/W

Bit	Symbol	Bit name	Description	R/W
b8	BSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTIOCA input when GTIOCB input is 0 1: Enable GTCCRB input capture on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	BSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTIOCA input when GTIOCB input is 1 1: Enable GTCCRB input capture on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	BSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTIOCA input when GTIOCB input is 0 1: Enable GTCCRB input capture on the falling edge of GTIOCA input when GTIOCB input is 0.	R/W
b11	BSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTIOCA input when GTIOCB input is 1 1: Enable GTCCRB input capture on the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	BSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTIOCB input when GTIOCA input is 0 1: Enable GTCCRB input capture on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	BSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the rising edge of GTIOCB input when GTIOCA input is 1 1: Enable GTCCRB input capture on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	BSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTIOCB input when GTIOCA input is 0 1: Enable GTCCRB input capture on the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	BSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on the falling edge of GTIOCB input when GTIOCA input is 1 1: Enable GTCCRB input capture on the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTA event input 1: Enable GTCCRB input capture on ELC_GPTA event input.	R/W
b17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTB event input 1: Enable GTCCRB input capture on ELC_GPTB event input.	R/W
b18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTC event input 1: Enable GTCCRB input capture on ELC_GPTC event input.	R/W
b19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTD event input 1: Enable GTCCRB input capture on ELC_GPTD event input.	R/W
b20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTE event input 1: Enable GTCCRB input capture on ELC_GPTE event input.	R/W
b21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTF event input 1: Enable GTCCRB input capture on ELC_GPTF event input.	R/W
b22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTG event input 1: Enable GTCCRB input capture on ELC_GPTG event input.	R/W

Bit	Symbol	Bit name	Description	R/W
b23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable	0: Disable GTCCRB input capture on ELC_GPTH event input 1: Enable GTCCRB input capture on ELC_GPTH event input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTICBSR register sets the source of input capture for GTCCRB.

BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGAR bit enables or disables input capture for GTCCRB on the rising edge of the GTETRGA pin input.

BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGAF bit enables or disables input capture for GTCCRB on the falling edge of the GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGBR bit enables or disables input capture for GTCCRB on the rising edge of the GTETRGB pin input.

BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGBF bit enables or disables input capture for GTCCRB on the falling edge of the GTETRGB pin input.

BSGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGCR bit enables or disables input capture for GTCCRB on the rising edge of the GTETRGC pin input.

BSGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGCF bit enables or disables input capture for GTCCRB on the falling edge of the GTETRGC pin input.

BSGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGDR bit enables or disables input capture for GTCCRB on the rising edge of the GTETRGD pin input.

BSGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGDF bit enables or disables input capture for GTCCRB on the falling edge of the GTETRGD pin input.

BSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL bit enables or disables input capture for GTCCRB on the rising edge of the GTIOCA pin input when the GTIOCB input is 0.

BSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRB Input Capture Enable)

The BSCARBH bit enables or disables input capture for GTCCRB on the rising edge of the GTIOCA pin input when the GTIOCB input is 1.

BSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRB Input Capture Enable)

The BSCAFBL bit enables or disables input capture for GTCCRB on the falling edge of the GTIOCA pin input when the GTIOCB input is 0.

BSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRB Input Capture Enable)

The BSCAFBH bit enables or disables input capture for GTCCRB on the falling edge of the GTIOCA pin input when the GTIOCB input is 1.

BSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRB Input Capture Enable)

The BSCBRAL bit enables or disables input capture for GTCCRB on the rising edge of the GTIOCB pin input when the

GTIOCA input is 0.

BSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRB Input Capture Enable)

The BSCBRAH bit enables or disables input capture for GTCCRB on the rising edge of the GTIOCB pin input when the GTIOCA input is 1.

BSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRB Input Capture Enable)

The BSCBFAL bit enables or disables input capture for GTCCRB on the falling edge of the GTIOCB pin input when the GTIOCA input is 0.

BSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRB Input Capture Enable)

The BSCBFAH bit enables or disables input capture for GTCCRB on the falling edge of the GTIOCB pin input when the GTIOCA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to H)

The BSELCm bit enables or disables input capture for GTCCRB on the ELC_GPTm event input.

23.2.12 General PWM Timer Control Register (GTCR)

Address(es): GPT32EHm.GTCR 4007 802Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTCR 4007 802Ch + 0100h × m (m = 4 to 7)
GPT32m.GTCR 4007 802Ch + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	TPCS[2:0]			—	—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CST	Count Start	0: Stop count operation 1: Perform count operation.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	MD[2:0]	Mode Select	b18 b16 0 0 0: Saw-wave PWM mode (single buffer or double buffer possible) 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b26 to b24	TPCS[2:0]	Timer Prescaler Select	b26 b24 0 0 0: PCLKD/1 0 0 1: PCLKD/4 0 1 0: PCLKD/16 0 1 1: PCLKD/64 1 0 0: PCLKD/256 1 0 1: PCLKD/1024.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTCR register controls GTCNT.

CST bit (Count Start)

The CST bit controls the GTCNT counter start and stop.

[Setting conditions]

- GTSTR value in which the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit being 1
- The ELC event input or the GTIOCA/GTIOCB/GTETRGn port input event enabled by GTSSR as the counter start source occurs
- 1 is written by software directly.

[Clearing conditions]

- GTSTP value in which the channel number associated with the bit number is set to 1 with the GTSSR.CSTOP bit being 1.
- The ELC event input or the GTIOCA/GTIOCB/GTETRGn port input event enabled by GTSSR as the counter stop source occurs
- 0 is written by software directly.

MD[2:0] bits (Mode Select)

The MD[2:0] bits select the GPT operating mode. The MD[2:0] bits must be set while the GTCNT operation is stopped.

TPCS[2:0] bits (Timer Prescaler Select)

The TPCS[2:0] bits select the clock for GTCNT. A clock prescaler can be selected independently for each channel. The TPCS[2:0] bits must be set while the GTCNT operation is stopped.

23.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)

Address(es): GPT32EHm.GTUDDTYC 4007 8030h + 0100h × m (m = 0 to 3)
GPT32Em.GTUDDTYC 4007 8030h + 0100h × m (m = 4 to 7)
GPT32m.GTUDDTYC 4007 8030h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	UD	Count Direction Setting	0: GTCNT counts down 1: GTCNT counts up.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	UDF	Forcible Count Direction Setting	0: Do not force setting 1: Force setting.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, 16	OADTY[1:0]	GTIOCA Output Duty Setting	b17 b16 0 x: GTIOCA pin duty depends on compare match 1 0: GTIOCA pin duty = 0% 1 1: GTIOCA pin duty = 100%.	R/W
b18	OADTYF	Forcible GTIOCA Output Duty Setting	0: Do not force setting 1: Force setting.	R/W
b19	OADTYR	GTIOCA Output Value Selecting after Releasing 0%/100% Duty Setting	0: Apply output value set in 0%/100% duty to GTIOA[3:2] function after releasing 0%/100% duty setting 1: Apply masked compare match output value to GTIOA[3:2] function after releasing 0%/100% duty setting.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	OBDTY[1:0]	GTIOCB Output Duty Setting	b25 b24 0 x: GTIOCB pin duty depends on compare match 1 0: GTIOCB pin duty = 0% 1 1: GTIOCB pin duty = 100%.	R/W
b26	OBDTYF	Forcible GTIOCB Output Duty Setting	0: Do not force setting 1: Force setting.	R/W
b27	OBDTYR	GTIOCB Output Value Selecting after Releasing 0%/100% Duty Setting	0: Apply output value set in 0%/100% duty to GTIOB[3:2] function after releasing 0%/100% duty setting 1: Apply masked compare match output value to GTIOB[3:2] function after releasing 0%/100% duty setting.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

The GTUDDTYC register sets the direction in which GTCNT counts (up-counting or down-counting) and sets the duty of GTIOCA/GTIOCB pin output.

Count direction in saw-wave mode

When the UD value is set to 0 during up-counting, the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value is set to 1 during down-counting, the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).

When the UD value changes from 1 to 0 with the UDF bit being 0 and while counting is stopped, the counter starts up-counting and the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes GTPR value). When the UD value changes from 0 to 1 with the UDF bit being 0 and while counting is stopped, the counter starts down-counting and the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).

When the UDF bit is set to 1 while counting is stopped, the UD bit value is reflected in the count direction when counting starts.

Count direction in triangle-wave mode

When the UD value changes during counting, the count direction does not change. When the UD value changes while the UDF bit is 0 and counting is stopped, the change is not reflected in the count direction when counting starts.

When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

The UD bit sets the count direction for GTCNT, either up-counting or down-counting.

UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only write 0 to this bit during counter operation. When 1 is written to UDF while counting is stopped, return UDF to 0 before counting starts.

Output duty in saw-wave mode

When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR). When the OADTY/OBDTY value changes during down-counting, the duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value changes to 1 with the OADTYF/OBDTYF bit being 0 and while counting is stopped, the output duty is not reflected at starting counter operation. When the count direction is up, the output duty is reflected at an overflow (GTCNT = GTPR). When the count direction is down, the output duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value changes to 0 with the OADTYF/OBDTYF bit being 1 and while counting is stopped, the output duty is reflected at starting counter operation.

Output duty in triangle-wave mode

When the OADTY/OBDTY value changes during counting, the duty is reflected at an underflow. When the OADTY/OBDTY value changes to 1 with the OADTYF/OBDTYF bit being 0 and while counting is stopped, the output duty is not reflected at starting counter operation. The output duty is reflected at an underflow.

When the OADTY/OBDTY value changes to 0 with the OADTYF/OBDTYF bit being 1 and while counting is stopped, the output duty is reflected at starting counter operation.

OmDTY[1:0] bits (GTIOCm Output Duty Setting) (m = A, B)

The OmDTY[1:0] bits set the output duty of the GTIOCm pin to either 0%, 100%, or compare match control.

OmDTYF bit (Forcible GTIOCm Output Duty Setting) (m = A, B)

The OmDTYF bit forcibly sets the output duty cycle to the OmDTY setting. Set this bit to 0 during counter operation. When OmDTYF bit is set to 1 while counting is stopped, return OmDTYF to 0 until the first period ends after the counter starts.

OmDTYR bit (GTIOCm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

The OmDTYR bits select the value that is the object of output retained or toggled at cycle end, when the control changes from 0%/100% duty setting to compare match for GTIOCm pin and GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end).

While the duty 0%/100% setting operation is running, the compare match operation continues inside the GPT32. When the OmDTYR bit is set to 1, the GTIOCm pin is in the output state selected by the GTIOR.GTIOm [3:2] bit at the end of the cycle in the compare match operation.

23.2.14 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPT32EHm.GTIOR 4007 8034h + 0100h × m (m = 0 to 3)
GPT32Em.GTIOR 4007 8034h + 0100h × m (m = 4 to 7)
GPT32m.GTIOR 4007 8034h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCSB[1:0]		NFBEN	—	—	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NFCSA[1:0]		NFAEN	—	—	OADF[1:0]		OAE	OAHLD	OADFLT	—	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	GTIOA[4:0]	GTIOCA Pin Function Select	See Table 23.5.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	OADFLT	GTIOCA Pin Output Value Setting at the Count Stop	0: Output low on GTIOCA pin when counting is stopped 1: Output high on GTIOCA pin when counting is stopped.	R/W

Bit	Symbol	Bit name	Description	R/W
b7	OAHLD	GTIOCA Pin Output Setting at the Start/Stop Count	0: Set GTIOCA pin output level on counting start and stop based on the register setting. 1: Retain GTIOCA pin output level on counting start and stop.	R/W
b8	OAE	GTIOCA Pin Output Enable	0: Disable output 1: Enable output.	R/W
b10, b9	OADF[1:0]	GTIOCA Pin Disable Value Setting	b10 b9 0 0: Prohibit output disable 0 1: Set GTIOCA pin to Hi-Z on output disable 1 0: Set GTIOCA pin to 0 on output disable 1 1: Set GTIOCA pin to 1 on output disable.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	NFAEN	Noise Filter A Enable	0: Disable noise filter for GTIOCA pin 1: Enable noise filter for GTIOCA pin.	R/W
b15, b14	NFCSA[1:0]	Noise Filter A Sampling Clock Select	b15 b14 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64.	R/W
b20 to b16	GTIOB[4:0]	GTIOCB Pin Function Select	See Table 23.5.	R/W
b21	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22	OBDFLT	GTIOCB Pin Output Value Setting at the Count Stop	0: Output low on GTIOCB pin when counting is stopped 1: Output high on GTIOCB pin when counting is stopped.	R/W
b23	OBHLD	GTIOCB Pin Output Setting at the Start/Stop Count	0: Set GTIOCB pin output level on counting start and stop based on the register setting 1: Retain GTIOCB pin output level on counting start and stop.	R/W
b24	OBE	GTIOCB Pin Output Enable	0: Disable output 1: Enable output.	R/W
b26, b25	OBDF[1:0]	GTIOCB Pin Disable Value Setting	b26 b25 0 0: Prohibit output disable 0 1: Set GTIOCB pin to Hi-Z on output disable 1 0: Set GTIOCB pin to 0 on output disable 1 1: Set GTIOCB pin to 1 on output disable.	R/W
b28, b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFBEN	Noise Filter B Enable	0: Disable noise filter for GTIOCB pin 1: Enable noise filter for GTIOCB pin.	R/W
b31, b30	NFCSB[1:0]	Noise Filter B Sampling Clock Select	b31 b30 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64.	R/W

The GTIOR register sets the functions of the GTIOCA and GTIOCB pins.

GTIOA[4:0] bits (GTIOCA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCA pin function. For details, see Table 23.5.

OADFLT bit (GTIOCA Pin Output Value Setting at the Count Stop)

The OADFLT bit selects whether the GTIOCA pin outputs high or low when counting is stopped.

OAHLD bit (GTIOCA Pin Output Setting at the Start/Stop Count)

The OAHLD bit specifies whether the GTIOCA pin output level is retained or the level depends on the register setting when counting is started or stopped.

[When the OAHLD bit is set to 0]

- The value specified in the GTIOA[4] bit is output when counting starts
- The value specified in the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting is stopped, the new value is immediately reflected in the output.

[When the OAHLD bit is set to 1]

- The output is retained when counting starts or stops.

OAE bit (GTIOCA Pin Output Enable)

The OAE bit disables or enables the GTIOCA pin output.

When GTCCRA register is used as the input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCA pin does not output regardless of the OAE bit value.

OADF[1:0] bits (GTIOCA Pin Disable Value Setting)

The OADF[1:0] bits select the output value of GTIOCA pin when an output disable request occurs.

NFAEN bit (Noise Filter A Enable)

The NFAEN bit disables or enables the noise filter for input from the GTIOCA pin. Because changing the value of the bit might lead to internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSA[1:0] bits (Noise Filter A Sampling Clock Select)

The NFCSA[1:0] bits set the sampling interval for the noise filter of the GTIOCA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

GTIOB[4:0] bits (GTIOCB Pin Function Select)

The GTIOB[4:0] bits select the GTIOCB pin function. For details, see Table 23.5.

OBDFLT bit (GTIOCB Pin Output Value Setting at the Count Stop)

The OBDFLT bit sets whether the GTIOCB pin outputs high or low when counting is stopped.

OBHLD bit (GTIOCB Pin Output Setting at the Start/Stop Count)

The OBHLD bit specifies whether the GTIOCB pin output level is retained or the level depends on the register setting when counting is started or stopped.

[When the OBHLD bit is set to 0]

- The value specified in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting is stopped, the new value is immediately reflected in the output.

[When the OBHLD bit is set to 1]

- The output is retained when counting starts or stops.

OBE bit (GTIOCB Pin Output Enable)

The OBE bit disables or enables the GTIOCB pin output.

When GTCCRB register is used as the input capture register (at least one bit in GTICBSR register is set to 1), the GTIOCB pin does not output regardless of the OBE bit value.

OBDF[1:0] bits (GTIOCB Pin Disable Value Setting)

The OBDF[1:0] bits select the output value of GTIOCB pin when an output disable request occurs.

NFBEN bit (Noise Filter B Enable)

The NFBEN bit disables or enables the noise filter for input from the GTIOCB pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSB[1:0] bits (Noise Filter B Sampling Clock Select)

The NFCSB[1:0] bits set the sampling interval for the noise filter of the GTIOCB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

Table 23.5 Settings of GTIOA[4:0] and GTIOB[4:0] bits

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	0	0	0	Set initial output low	Retain output at cycle end	Retain output at GTCCRA/GTCCRB compare match
0	0	0	0	1			Output low at GTCCRA/GTCCRB compare match
0	0	0	1	0			Output high at GTCCRA/GTCCRB compare match
0	0	0	1	1			Toggle output at GTCCRA/GTCCRB compare match
0	0	1	0	0		Output low at cycle end	Retain output at GTCCRA/GTCCRB compare match
0	0	1	0	1			Output low at GTCCRA/GTCCRB compare match
0	0	1	1	0			Output high at GTCCRA/GTCCRB compare match
0	0	1	1	1			Toggle output at GTCCRA/GTCCRB compare match
0	1	0	0	0		Output high at cycle end	Retain output at GTCCRA/GTCCRB compare match
0	1	0	0	1			Output low at GTCCRA/GTCCRB compare match
0	1	0	1	0			Output high at GTCCRA/GTCCRB compare match
0	1	0	1	1			Toggle output at GTCCRA/GTCCRB compare match
0	1	1	0	0		Toggle output at cycle end	Retain output at GTCCRA/GTCCRB compare match
0	1	1	0	1			Output low at GTCCRA/GTCCRB compare match
0	1	1	1	0			Output high at GTCCRA/GTCCRB compare match
0	1	1	1	1			Toggle output at GTCCRA/GTCCRB compare match
1	0	0	0	0	Set initial output high	Retain output at cycle end	Retain output at GTCCRA/GTCCRB compare match
1	0	0	0	1			Output low at GTCCRA/GTCCRB compare match
1	0	0	1	0			Output high at GTCCRA/GTCCRB compare match
1	0	0	1	1			Toggle output at GTCCRA/GTCCRB compare match
1	0	1	0	0		Output low at cycle end	Retain output at GTCCRA/GTCCRB compare match
1	0	1	0	1			Output low at GTCCRA/GTCCRB compare match
1	0	1	1	0			Output high at GTCCRA/GTCCRB compare match
1	0	1	1	1			Toggle output at GTCCRA/GTCCRB compare match
1	1	0	0	0		Output high at cycle end	Retain output at GTCCRA/GTCCRB compare match
1	1	0	0	1			Output low at GTCCRA/GTCCRB compare match
1	1	0	1	0			Output high at GTCCRA/GTCCRB compare match
1	1	0	1	1			Toggle output at GTCCRA/GTCCRB compare match
1	1	1	0	0		Toggle output at cycle end	Retain output at GTCCRA/GTCCRB compare match
1	1	1	0	1			Output low at GTCCRA/GTCCRB compare match
1	1	1	1	0			Output high at GTCCRA/GTCCRB compare match
1	1	1	1	1			Toggle output at GTCCRA/GTCCRB compare match

Note 1. The cycle end means an overflow (GTCNT is changed from GTPR to 0 in up-counting) or underflow (GTCNT is changed from 0 to GTPR in down-counting). In this case, the GTCNT counter is cleared for saw waves and for the trough (GTCNT is changed from 0 to 1) for triangle waves.

Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.

Note 3. In event count operation where at least one bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored.

23.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD)

Address(es): GPT32EHm.GTINTAD 4007 8038h + 0100h × m (m = 0 to 3)
GPT32Em.GTINTAD 4007 8038h + 0100h × m (m = 4 to 7)
GPT32m.GTINTAD 4007 8038h + 0100h × m (m = 8 to 13)

- GPT32EH, GPT32E

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- GPT32

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	GRPABL	GRPABH	—	—	—	GRP[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ADTRAUEN	GTADTRA Compare Match (Up-Counting) A/D Converter Start Request Enable	0: Disable A/D converter start request 1: Enable A/D converter start request.	R/W
b17	ADTRADEN	GTADTRA Compare Match (Down-Counting) A/D Converter Start Request Enable	0: Disable A/D converter start request 1: Enable A/D converter start request.	R/W
b18	ADTRBUEN	GTADTRB Compare Match (Up-Counting) A/D Converter Start Request Enable	0: Disable A/D converter start request 1: Enable A/D converter start request.	R/W
b19	ADTRBDEN	GTADTRB Compare Match (Down-Counting) A/D Converter Start Request Enable	0: Disable A/D converter start request 1: Enable A/D converter start request.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Disable Source Select	b25 b24 0 0: Select Group A output disable request 0 1: Select Group B output disable request 1 0: Select Group C output disable request 1 1: Select Group D output disable request.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	GRPDTE	Dead Time Error Output Disable Request Enable	0: Disable dead time error output disable request 1: Enable dead time error output disable request.	R/W
b29	GRPABH	Same Time Output Level High Disable Request Enable	0: Disable same time output level high disable request 1: Enable same time output level high disable request.	R/W
b30	GRPABL	Same Time Output Level Low Disable Request Enable	0: Disable same time output level low disable request 1: Enable same time output level low disable request.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

GTINTAD enables or disables interrupt requests, A/D converter start requests, and output disable requests.

ADTRAUEN bit (GTADTRA Compare Match (Up-Counting) A/D Converter Start Request Enable)

The ADTRAUEN bit enables or disables A/D converter start requests generated by GTADTRA compare matches during GTCNT up-counting. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

ADTRADEN bit (GTADTRA Compare Match (Down-Counting) A/D Converter Start Request Enable)

The ADTRADEN bit enables or disables A/D converter start requests generated by GTADTRA compare matches during GTCNT down-counting. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

ADTRBUEN bit (GTADTRB Compare Match (Up-Counting) A/D Converter Start Request Enable)

The ADTRBUEN bit enables or disables A/D converter start requests generated by GTADTRB compare matches during GTCNT up-counting. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

ADTRBDEN bit (GTADTRB Compare Match (Down-Counting) A/D Converter Start Request Enable)

The ADTRBDEN bit enables or disables A/D converter start requests generated by GTADTRB compare matches during GTCNT down-counting. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

GRP[1:0] bits (Output Disable Source Select)

The GRP[1:0] bits select GTIOCA pin and GTIOCB pin output disable source. The output disable request to POEG outputs to the group which is selected by GRP[1:0] bits when dead time error, same time output level high or low occurs according to each output disable request enable bits.

GTST.ODF shows the request of output disable source group that is selected with the GRP[1:0] bits.

Set the GRP[1:0] bits when both GTIOR.OAE and GTIOR.OBE are 0.

GRPDTE bit (Dead Time Error Output Disable Request Enable)

The GRPDTE bit enables or disables dead time error output disable request. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

GRPABH bit (Same Time Output Level High Disable Request Enable)

The GRPABH bit enables or disables output disable request when GTIOCA pin and GTIOCB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables output disable request when GTIOCA pin and GTIOCB pin output 0 at the same time.

23.2.16 General PWM Timer Status Register (GTST)

Address(es): GPT32EHm.GTST 4007 803Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTST 4007 803Ch + 0100h × m (m = 4 to 7)
GPT32m.GTST 4007 803Ch + 0100h × m (m = 8 to 13)

- GPT32EH, GPT32E

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	—	ITCNT[2:0]			TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- GPT32

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	OABLF	OABHF	—	—	—	—	ODF	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	—	—	—	—	TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TCFA	Input Capture/Compare Match Flag A	0: No input capture/compare match of GTCCRA occurred 1: Input capture/compare match of GTCCRA occurred.	R/(W)*1
b1	TCFB	Input Capture/Compare Match Flag B	0: No input capture/compare match of GTCCRB occurred 1: Input capture/compare match of GTCCRB occurred.	R/(W)*1
b2	TCFC	Input Compare Match Flag C	0: No compare match of GTCCRC occurred 1: Compare match of GTCCRC occurred.	R/(W)*1
b3	TCFD	Input Compare Match Flag D	0: No compare match of GTCCRD occurred 1: Compare match of GTCCRD occurred.	R/(W)*1
b4	TCFE	Input Compare Match Flag E	0: No compare match of GTCCRE occurred 1: Compare match of GTCCRE occurred.	R/(W)*1
b5	TCFF	Input Compare Match Flag F	0: No compare match of GTCCRF occurred 1: Compare match of GTCCRF occurred.	R/(W)*1
b6	TCFPO	Overflow Flag	0: No overflow (crest) occurred 1: Overflow (crest) occurred.	R/(W)*1
b7	TCFPU	Underflow Flag	0: No underflow (trough) occurred 1: Underflow (trough) occurred.	R/(W)*1
b10 to b8	ITCNT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter	Counter for counting the number of times a timer interrupt is skipped.	R
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TUCF	Count Direction Flag	0: GTCNT counter is counting down 1: GTCNT counter is counting up.	R
b16	ADTRAUF	GTADTRA Compare Match (Up-Counting) A/D Converter Start Request Flag	0: No compare match of GTADTRA at up-counting occurred 1: A compare match of GTADTRA at up-counting occurred.	R/(W)*1
b17	ADTRADF	GTADTRA Compare Match (Down-Counting) A/D Converter Start Request Flag	0: No compare match of GTADTRA at down-counting occurred 1: A compare match of GTADTRA at down-counting occurred.	R/(W)*1

Bit	Symbol	Bit name	Description	R/W
b18	ADTRBUF	GTADTRB Compare Match (Up-Counting) A/D Converter Start Request Flag	0: No compare match of GTADTRB at up-counting occurred 1: A compare match of GTADTRB at up-counting occurred.	R/(W)*1
b19	ADTRBDF	GTADTRB Compare Match (Down-Counting) A/D Converter Start Request Flag	0: No compare match of GTADTRB at down-counting occurred 1: A compare match of GTADTRB at down-counting occurred.	R/(W)*1
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ODF	Output Disable Flag	0: No output disable request occurred 1: Output disable request occurred.	R
b27 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	DTEF	Dead Time Error Flag	0: No dead time error occurred 1: Dead time error occurred.	R
b29	OABHF	Same Time Output Level High Flag	0: GTIOCA pin and GTIOCB pin did not output 1 at the same time 1: GTIOCA pin and GTIOCB pin output 1 at the same time.	R
b30	OABLF	Same Time Output Level Low Flag	0: GTIOCA pin and GTIOCB pin did not output 0 at the same time 1: GTIOCA pin and GTIOCB pin output 0 at the same time.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. Do not write 1.

The GTST register indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

The TCFS flag indicates the status for the input capture or compare match of GTCCRA.

[Setting conditions]

- GTCNT = GTCCRA when the GTCCRA register functions as a compare match register
- GTCNT counter value is transferred to GTCCRA by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of GTCCRB.

[Setting conditions]

- GTCNT = GTCCRB when the GTCCRB register functions as a compare match register
- GTCNT counter value is transferred to GTCCRB by the input capture signal when the GTCCRB register function as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC flag (Input Compare Match Flag C)

The TCFC flag indicates the status for the compare match of GTCCRC.

[Setting condition]

- GTCNT = GTCCRC

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (GTCCRC performs buffer operation).

TCFD flag (Input Compare Match Flag D)

The TCFD flag indicates the status for the compare match of GTCCRD.

[Setting condition]

- GTCNT = GTCCRD

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (GTCCRD performs buffer operation).

TCFE flag (Input Compare Match Flag E)

The TCFE flag indicates the status for the compare match of GTCCRE.

[Setting condition]

- GTCNT = GTCCRE

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (GTCCRE performs buffer operation).

TCFF flag (Input Compare Match Flag F)

The TCFF flag indicates the status for the compare match of GTCCRF.

[Setting condition]

- GTCNT = GTCCRF

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (GTCCRF performs buffer operation).

TCFPO flag (Overflow Flag)

The TCFPO flag indicates when an overflow or a crest has occurred.

[Setting conditions]

- In saw-wave mode, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (GTCNT changes from GTPR to GTPR-1) has occurred

- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU flag (Underflow Flag)

The TCFPU flag indicates when an underflow or a trough has occurred.

[Setting conditions]

- In saw-wave mode, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred
- In triangle-wave mode, a trough (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

ITCNT[2:0] bits (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter)

When the GPTn_OVF/GPTn_UDF (n = 0 to 7) interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter in the ITCNT[2:0] bits increments by 1 every time the GPTn_OVF/GPTn_UDF interrupt source that is selected in GTITC.IVTC[1:0] is generated.

Only GPT32EH and GPT32E have these bits. GPT32 does not have these bits.

[Clearing conditions]

- The GPTn_OVF/GPTn_UDF interrupt skipping function is not used (GTITC.IVTT[2:0] is 000b when GTITC.IVTC[1:0] is 00b)
- The GPTn_OVF/GPTn_UDF interrupt skipping count matches the specified count (ITCNT[2:0] matches the skipping count specified in GTITC.IVTT[2:0]).

TUCF flag (Count Direction Flag)

The TUCF flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and is set to 0 in down-counting.

ADTRAUF flag (GTADTRA Compare Match (Up-Counting) A/D Converter Start Request Flag)

The ADTRAUF is a status flag for the compare match of GTADTRA at up-counting.

[Setting condition]

- GTCNT = GTADTRA at up-counting.

[Clearing condition]

- 0 is written to this bit.

ADTRADF flag (GTADTRA Compare Match (Down-Counting) A/D Converter Start Request Flag)

The ADTRADF is a status flag for the compare match of GTADTRA at down-counting.

[Setting condition]

- GTCNT = GTADTRA at down-counting.

[Clearing condition]

- 0 is written to this bit.

ADTRBUF flag (GTADTRB Compare Match (Up-Counting) A/D Converter Start Request Flag)

The ADTRBUF is a status flag for the compare match of GTADTRB at up-counting.

[Setting condition]

- GTCNT = GTADTRB at up-counting.

[Clearing condition]

- 0 is written to this bit.

ADTRBDF flag (**GTADTRB Compare Match (Down-Counting) A/D Converter Start Request Flag**)

The ADTRBDF is a status flag for the compare match of GTADTRB at down-counting.

[Setting condition]

- GTCNT = GTADTRB at down-counting.

[Clearing condition]

- 0 is written to this bit.

ODF flag (Output Disable Flag)

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits. When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

DTEF flag (Dead Time Error Flag)

The DTEF flag indicates that the timer output toggle point after the automatic addition of dead time has exceeded the timer cycle.

DTEF returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the cycle. DTEF is read only. Writing 0 to clear the flag is not allowed.

[Setting condition]

- The timer output toggle point after the automatic addition of dead time has exceeded the timer cycle.
  For triangle wave in up-counting: GTCCRA - GTDVU ≤ 0
  For triangle wave in down-counting: GTCCRA - GTDVD < 0
  For saw-wave one-shot pulse mode in up-counting:
    GTCCRA - GTDVU < 0 or GTCCRA + GTDVD > GTPR
  For saw-wave one-shot pulse mode in down-counting:
    GTCCRA + GTDVU > GTPR or GTCCRA - GTDVD < 0

[Clearing condition]

- The timer output toggle point after the automatic addition of dead time is within the timer cycle.
  Only GPT32EH and GPT32E have this flag. GPT32 does not have this flag.
  GPT32 has the automatic dead time setting function but it does not generate dead time error.

OABHF flag (Same Time Output Level High Flag)

The OABHF flag indicates that the GTIOCA pin and the GTIOCB pin output 1 at the same time.

When the GTIOCA pin or GTIOCB pin outputs 0, OABHF returns to 0. OABHF is read only. Writing 0 to clear the flag is not allowed. When an interrupt by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to the POEG as an output disable request.

[Setting condition]

- GTIOCA pin and GTIOCB pin output 1 at the same time when both the OAE and OBE bits are set to 1.

[Clearing conditions]

- GTIOCA pin output value is different from GTIOCB pin output value when both the OAE and OBE bits are set to 1
- GTIOCA pin and GTIOCB pin output 0 at the same time when both the OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

The OABLF flag indicates that the GTIOCA pin and the GTIOCB pin output 0 at the same time.

When the GTIOCA pin or GTIOCB pin outputs 1, OABLF returns to 0. OABLF is read only. Writing 0 to clear the flag is not allowed. When an interrupt by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to the POEG as an output disable request.

[Setting condition]

- GTIOCA pin and GTIOCB pin output 0 at the same time when both the OAE and OBE bits are set to 1.

[Clearing conditions]

- GTIOCA pin output value is different from GTIOCB pin output value when both the OAE and OBE bits are set to 1
- GTIOCA pin and GTIOCB pin output 1 at the same time when both the OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flags are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. When the output disable state is active, a compare match is performed continuously in the GPT and the OABHF/OABLF flags are updated according to with the result of the compared values.

23.2.17 General PWM Timer Buffer Enable Register (GTBER)

Address(es): GPT32EHm.GTBER 4007 8040h + 0100h × m (m = 0 to 3)
GPT32Em.GTBER 4007 8040h + 0100h × m (m = 4 to 7)
GPT32m.GTBER 4007 8040h + 0100h × m (m = 8 to 13)

- GPT32EH, GPT32E

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	BD[3]	BD[2]	BD[1]	BD[0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- GPT32

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—		—	—	—		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BD[1]	BD[0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BD[0]	GTCCR Buffer Operation Disable	0: Enable buffer operation 1: Disable buffer operation.	R/W
b1	BD[1]	GTPR Buffer Operation Disable		R/W
b2	BD[2]	GTADTR Buffer Operation Disable		R/W
b3	BD[3]	GTDV Buffer Operation Disable		R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	CCRA[1:0]	GTCCRA Buffer Operation	b17 b16 0 0: No buffer operation 0 1: Single buffer operation (GTCCRA ↔ GTCCRC) 1 x: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD).	R/W

Bit	Symbol	Bit name	Description	R/W
b19, b18	CCRB[1:0]	GTCCRB Buffer Operation	b19 b18 0 0: No buffer operation 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) 1 x: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF).	R/W
b21, b20	PR[1:0]	GTPR Buffer Operation	b21 b20 0 0: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) 1 x: Double buffer operation (GTPDBR → GTPBR → GTPR).	R/W
b22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation	Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25, b24	ADTTA[1:0]	GTADTRA Buffer Transfer Timing Select	• Triangle waves b25 b24 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough. • Saw waves b25 b24 0 0: No transfer Values other than 0 0: Transfer on underflow (during down-counting) or on overflow (during up-counting).	R/W
b26	ADTDA	GTADTRA Double Buffer Operation	0: Single buffer operation (GTADTBRA → GTADTRA) 1: Double buffer operation (GTADTDBRA → GTADTBRA → GTADTDRA).	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b29, b28	ADTTB[1:0]	GTADTRB Buffer Transfer Timing Select	• Triangle waves b29 b28 0 0: No transfer 0 1: Transfer at crest 1 0: Transfer at trough 1 1: Transfer at both crest and trough. • Saw waves b29 b28 0 0: No transfer Values other than 0 0: Transfer on underflow (in down-counting) or on overflow (in up-counting).	R/W
b30	ADTDB	GTADTRB Double Buffer Operation	0: Single buffer operation (GTADTBRB → GTADTRB) 1: Double buffer operation (GTADTDBRB → GTADTBRB → GTADTDRB).	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The GTBER register provides settings for the buffer operation and must be set while the GTCNT operation is stopped.

BD[0] bit (GTCCR Buffer Operation Disable)

The BD[0] bit disables buffer operation using GTCCRA, GTCCRC, and GTCCRD combined and buffer operation using GTCCRB, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1, even if BD[0] is set to 0, GTCCRB does not perform buffer operation. The GTCCRB register is automatically set to a compare match value for a negative-phase waveform with dead time.

BD[1] bit (GTPR Buffer Operation Disable)

The BD[1] bit disables buffer operation using GTPR, GTPBR, and GTPDBR combined.

BD[2] bit (GTADTR Buffer Operation Disable)

The BD[2] bit disables buffer operation using GTADTRA, GTADTBRA, and GTADTDBRA combined and buffer operation using GTADTRB, GTADTBRB, and GTADTDBRB combined. In event count operation, this bit is not available and the GTADTR buffer operation is not performed. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

BD[3] bit (GTDV Buffer Operation Disable)

The BD[3] bit disables buffer operation using GTDVU and GTDBU combined and buffer operation using GTDVD and GTDBD combined.

When the GTDTCR.TDFER bit is set to 1, even if BD[3] is set to 0, buffer operation is not performed and the GTDVD value is set as a value of GTDVU automatically. In event count operation, this bit is not available and the GTDV buffer operation is not performed. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

CCRA[1:0] bits (GTCCRA Buffer Operation)

The CCRA[1:0] bits set buffer operation using GTCCRA, GTCCRC, and GTCCRD combined. When buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.*1

CCRB[1:0] bits (GTCCRB Buffer Operation)

The CCRB[1:0] bits set buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.*1

PR[1:0] bits (GTPR Buffer Operation)

The PR[1:0] bits set buffer operation using GTPR, GTPBR, and GTPDBR combined. GPT32 does not have the PR[1] bit. Only single buffer operation setting by PR[0] bit is available for GPT32.

CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forcibly performs a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0 and is only valid when counting is stopped with a compare match operation specified.

ADTTA[1:0] bits (GTADTRA Buffer Transfer Timing Select)

The ADTTA[1:0] bits set the transfer timing for buffer operation of GTADTRA, GTADTBRA, and GTADTDBRA. These bits are not available in event count operation. Only GPT32EH and GPT32E have these bits. GPT32 does not have these bits.

ADTDA bit (GTADTRA Double Buffer Operation)

The ADTDA bit sets buffer operation using GTADTRA, GTADTBRA, and GTADTDBRA combined. This bit is not available in event count operation. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

ADTTB[1:0] bits (GTADTRB Buffer Transfer Timing Select)

The ADTTB[1:0] bits set the transfer timing for buffer operation of GTADTRB, GTADTBRB, and GTADTDBRB. These bits are not available in event count operation. Only GPT32EH and GPT32E have these bits. GPT32 does not have these bits.

ADTDB bit (GTADTRB Double Buffer Operation)

The ADTDB bit sets buffer operation using GTADTRB, GTADTBRB, and GTADTDBRB combined. This bit is not available in event count operation. Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

Note 1. The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

23.2.18 General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register (GTITC)

Address(es): GPT32EHm.GTITC 4007 8044h + 0100h × m (m = 0 to 3)
GPT32Em.GTITC 4007 8044h + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ITLA	GTCCRA Compare Match/Input Capture Interrupt Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b1	ITLB	GTCCRB Compare Match/Input Capture Interrupt Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b2	ITLC	GTCCRC Compare Match Interrupt Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b3	ITLD	GTCCRD Compare Match Interrupt Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b4	ITLE	GTCCRE Compare Match Interrupt Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b5	ITLF	GTCCRF Compare Match Interrupt Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b7, b6	IVTC[1:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select	b7 b6 0 0: Do not perform skipping 0 1: Count and skip both overflow and underflow for saw waves and crest for triangle waves 1 0: Count and skip both overflow and underflow for saw waves and trough for triangle waves 1 1: Count and skip both overflow and underflow for saw waves and both crest and trough for triangle waves.	R/W
b10 to b8	IVTT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select	b10 b8 0 0 0: No skipping 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b12	ADTAL	GTADTRA A/D Converter Start Request Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	ADTBL	GTADTRB A/D Converter Start Request Link	0: Do not link with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Link with GPTn_OVF/GPTn_UDF interrupt skipping function.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTITC register sets the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF). It also specifies whether to link other interrupts and A/D converter start requests with the GPTn_OVF/GPTn_UDF interrupt skipping function. The output disable request to POEG cannot be linked with the GPTn_OVF/GPTn_UDF interrupt skipping function. This register is not available in event count operation. Only GPT32EH and GPT32E have this register. GPT32 does not have this register and it is read as 0.

ITLA bit (GTCCRA Compare Match/Input Capture Interrupt Link)

The ITLA bit specifies whether to link the GTCCRA compare match/input capture interrupt (GTCIA) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLB bit (GTCCRB Compare Match/Input Capture Interrupt Link)

The ITLB bit specifies whether to link the GTCCRB compare match/input capture interrupt (GTCIB) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLC bit (GTCCRC Compare Match Interrupt Link)

The ITLC bit specifies whether to link the GTCCRC compare match interrupt (GTCIC) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLD bit (GTCCRD Compare Match Interrupt Link)

The ITLD bit specifies whether to link the GTCCRD compare match interrupt (GTCID) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLE bit (GTCCRE Compare Match Interrupt Link)

The ITLE bit specifies whether to link the GTCCRE compare match interrupt (GTCIE) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLF bit (GTCCRF Compare Match Interrupt Link)

The ITLF bit specifies whether to link the GTCCRF compare match interrupt (GTCIF) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

IVTC[1:0] bits (GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select)

The IVTC[1:0] bits set the skipping function for the GTPR compare match (GTCNT overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF).

IVTT[2:0] bits (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select)

The IVTT[2:0] bits set the skipping count for the GTPR compare match (GTCNT overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF). When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL bit (GTADTRA A/D Converter Start Request Link)

The ADTAL bit specifies whether to link the GTADTRA A/D converter start request with GPTn_OVF/GPTn_UDF interrupt skipping function.

ADTBL bit (GTADTRB A/D Converter Start Request Link)

The ADTBL bit specifies whether to link the GTADTRB A/D converter start request with GPTn_OVF/GPTn_UDF interrupt skipping function.

23.2.19 General PWM Timer Counter (GTCNT)

Address(es): GPT32EHm.GTCNT 4007 8048h + 0100h × m (m = 0 to 3)
GPT32Em.GTCNT 4007 8048h + 0100h × m (m = 4 to 7)
GPT32m.GTCNT 4007 8048h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTCNT is a 32-bit read/write counter and can only be written to after counting stops. GTCNT must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited. GTCNT must be set within the range of 0 ≤ GTCNT ≤ GTPR.

23.2.20 General PWM Timer Compare Capture Register n (GTCCRn) (n = A to F)

Address(es): GPT32EHm.GTCCRA 4007 804Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTCCRA 4007 804Ch + 0100h × m (m = 4 to 7)
GPT32m.GTCCRA 4007 804Ch + 0100h × m (m = 8 to 13)
GPT32EHm.GTCCRB 4007 8050h + 0100h × m (m = 0 to 3)
GPT32Em.GTCCRB 4007 8050h + 0100h × m (m = 4 to 7)
GPT32m.GTCCRB 4007 8050h + 0100h × m (m = 8 to 13)
GPT32EHm.GTCCRC 4007 8054h + 0100h × m (m = 0 to 3)
GPT32Em.GTCCRC 4007 8054h + 0100h × m (m = 4 to 7)
GPT32m.GTCCRC 4007 8054h + 0100h × m (m = 8 to 13)
GPT32EHm.GTCCRE 4007 8058h + 0100h × m (m = 0 to 3)
GPT32Em.GTCCRE 4007 8058h + 0100h × m (m = 4 to 7)
GPT32m.GTCCRE 4007 8058h + 0100h × m (m = 8 to 13)
GPT32EHm.GTCCRD 4007 805Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTCCRD 4007 805Ch + 0100h × m (m = 4 to 7)
GPT32EHm.GTCCRF 4007 8060h + 0100h × m (m = 0 to 3)
GPT32Em.GTCCRF 4007 8060h + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GTCCRn registers are read/write registers.

GTCCRA and GTCCRB are registers used for both output compare and input capture.

GTCCRC and GTCCRE are compare match registers that can also function as buffer registers for GTCCRA and GTCCRB.

GTCCRD and GTCCRF are compare match registers that can also function as buffer registers for GTCCRC and GTCCRE (double-buffer registers for GTCCRA and GTCCRB).

23.2.21 General PWM Timer Cycle Setting Register (GTPR)

Address(es): GPT32EHm.GTPR 4007 8064h + 0100h × m (m = 0 to 3)
GPT32Em.GTPR 4007 8064h + 0100h × m (m = 4 to 7)
GPT32m.GTPR 4007 8064h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GTPR is a read/write register that sets the maximum count value of GTCNT. For saw waves, the value of (GTPR + 1) is the cycle. For triangle waves, the value of (GTPR value × 2) is the cycle.

23.2.22 General PWM Timer Cycle Setting Buffer Register (GTPBR)

Address(es): GPT32EHm.GTPBR 4007 8068h + 0100h × m (m = 0 to 3)
GPT32Em.GTPBR 4007 8068h + 0100h × m (m = 4 to 7)
GPT32m.GTPBR 4007 8068h + 0100h × m (m = 8 to 13)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GTPBR is a read/write register that functions as a buffer register for GTPR.

23.2.23 General PWM Timer Cycle Setting Double-Buffer Register (GTPDBR)

Address(es): GPT32EHm.GTPDBR 4007 806Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTPDBR 4007 806Ch + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GTPDBR is a 32-bit read/write register that functions as a buffer register for GTPBR (double-buffer register for GTPR). Only GPT32EH and GPT32E have this register. GPT32 does not have this register. This register is read with the value after reset.

23.2.24 A/D Converter Start Request Timing Register n (GTADTRn) (n = A, B)

Address(es): GPT32EHm.GTADTRA 4007 8070h + 0100h × m (m = 0 to 3)
GPT32Em.GTADTRA 4007 8070h + 0100h × m (m = 4 to 7)

GPT32EHm.GTADTRB 4007 807Ch + 0100h × m (m = 0 to 3)
GPT32Em.GTADTRB 4007 807Ch + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTRn registers are 32-bit read/write registers that set the timing of A/D converter start request generation. When the GTADTRn value matches the GTCNT counter value, an A/D converter start request is generated. GTADTRn must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited. Only GPT32EH and GPT32E have this register. GPT32 does not have this register. This register is read with the value after reset.

23.2.25 A/D Converter Start Request Timing Buffer Register n (GTADTBRn) (n = A, B)

Address(es): GPT32EHm.GTADTBRA 4007 8074h + 0100h × m (m = 0 to 3)
GPT32Em.GTADTBRA 4007 8074h + 0100h × m (m = 4 to 7)

GPT32EHm.GTADTBRB 4007 8080h + 0100h × m (m = 0 to 3)
GPT32Em.GTADTBRB 4007 8080h + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTBRn registers are 32-bit read/write registers that function as buffer registers for GTADTRn. GTADTBRn must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited. Only GPT32EH and GPT32E have this register. GPT32 does not have this register. This register is read with the value after reset.

23.2.26 A/D Converter Start Request Timing Double-Buffer Register n (GTADTDBRn) (n = A, B)

Address(es): GPT32EHm.GTADTDBRA 4007 8078h + 0100h × m (m = 0 to 3)
GPT32Em.GTADTDBRA 4007 8078h + 0100h × m (m = 4 to 7)

GPT32EHm.GTADTDBRB 4007 8084h + 0100h × m (m = 0 to 3)
GPT32Em.GTADTDBRB 4007 8084h + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTADTDBRn registers are 32-bit read/write registers that function as buffer registers for GTADTBRn (double-buffer registers for GTADTR). GTADTDBRn must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited. Only GPT32EH and GPT32E have this register. GPT32 does not have this register. This register is read with the value after reset.

23.2.27 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPT32EHm.GTDTCR 4007 8088h + 0100h × m (m = 0 to 3)
GPT32Em.GTDTCR 4007 8088h + 0100h × m (m = 4 to 7)
GPT32m.GTDTCR 4007 8088h + 0100h × m (m = 8 to 13)

- GPT32EH, GPT32E

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDFER	—	—	TDBDE	TDBUE	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- GPT32

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TDE	Negative-Phase Waveform Setting	0: Set GTCCRB without using GTDVU and GTDVD 1: Use GTDVU and GTDVD to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TDBUE	GTDVU Buffer Operation Enable	0: Disable GTDVU buffer operation 1: Enable GTDVU buffer operation.	R/W

Bit	Symbol	Bit name	Description	R/W
b5	TDBDE	GTDVD Buffer Operation Enable	0: Disable GTDVD buffer operation 1: Enable GTDVD buffer operation.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TDFER	GTDVD Setting	0: Set GTDVU and GTDVD separately 1: Automatically set the value written to GTDVU to GTDVD.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDTCR register enables automatic setting of a compare match value for negative-phase waveform with dead time.

GPT32EH, GPT32E and GPT32 have dead time control function. GPT32 does not have the dead time buffer function and only GTDVU register is used for setting dead time value.

TDE bit (Negative-Phase Waveform Setting)

The TDE bit specifies whether to use GTDVU and GTDVD. When GTDVU and GTDVD are used, the compare match value for a negative-phase waveform with dead time obtained by the compare match value of a positive-phase waveform (GTCCRA) and the dead time value (GTDVU and GTDVD) is automatically set in GTCCRB.

The TDE bit setting is ignored in saw-wave PWM mode, and automatic setting does not take place.

The GTCCRB value is automatically set and has the following upper and lower limit values. If the obtained GTCCRB value is not within the upper or lower limit, the following limit value is set in GTCCRB and the GTST.DTEF flag is set to 1. However, in triangle waves, when the obtained GTCCRB value exceeds the upper limit value, the GTST.DTEF flag is set to 0.

- Triangle waves
  Upper limit value: GTPR – 1
  Lower limit value: 1 in up-counting, 0 in down-counting
- Saw-wave one-shot pulse mode

  Upper limit value: GTPR

  Lower limit value: 0.

TDBUE bit (GTDVU Buffer Operation Enable)

The TDBUE bit enables buffer operation with GTDVU and GTDBU combined. The buffer transfer timing is the trough for triangle waves, and an overflow or underflow for saw waves.

Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

TDBDE bit (GTDVD Buffer Operation Enable)

The TDBDE bit enables buffer operation with GTDVD and GTDBD combined. The buffer transfer timing is the trough for triangle waves, and an overflow or underflow for saw waves. When this bit and the TDFER bit are set to 1 simultaneously, the TDFER bit setting is given priority.

Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

TDFER bit (GTDVD Setting)

The TDFER bits selects whether or not the value written to GTDVU is also set to GTDVD automatically.

Only GPT32EH and GPT32E have this bit. GPT32 does not have this bit.

23.2.28 General PWM Timer Dead Time Value Register n (GTDVn) (n = U, D)

Address(es): GPT32EHm.GTDVU 4007 808Ch + 0100h x m (m = 0 to 3)
GPT32Em.GTDVU 4007 808Ch + 0100h x m (m = 4 to 7)
GPT32m.GTDVU 4007 808Ch + 0100h x m (m = 8 to 13)
GPT32EHm.GTDVD 4007 8090h + 0100h x m (m = 0 to 3)
GPT32Em.GTDVD 4007 8090h + 0100h x m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GTDVn is a 32-bit read/write register that sets the dead time for generating PWM waveforms with dead time. GTDVU is used for up-counting and GTDVD is used for down-counting.

Setting a GTDVn value greater than or equal to GTPR is prohibited. Dead time setting beyond the cycle is prohibited. The compare match value set by the automatic dead time setting function for a negative waveform can be confirmed by reading from GTCCRB.

When GTDVn is used, writing to GTCCRB is not allowed. When this register is set to 0, waveforms without dead time are output. GTDVn must be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited. The way to rewrite GTDVn differs by GPT channel number.

GPT32EH0 to GPT32EH3 and GPT32E4 to GPT32E7

When GTDVm buffer operation is enabled, GTDBm can be written at anytime. GTDBm is transferred to GTDVm at the cycle end. When GTDVm buffer operation is disabled, stop the GPT using the CST bit in the GTCR register before changing GTDVm to a new value.

GPT328 to GPT3213

While the GPT is running, changing the GTDVU values is prohibited. To change GTDVU to a new value, stop the GPT with the CST bit in the GTCR register.

Only GPT32EH and GPT32E have the GTDVD register. GPT32 does not have the GTDVD register. This register is read with the value after reset.

23.2.29 General PWM Timer Dead Time Buffer Register n (GTDBn) (n = U, D)

Address(es): GPT32EHm.GTDBU 4007 8094h + 0100h × m (m = 0 to 3)
GPT32Em.GTDBU 4007 8094h + 0100h × m (m = 4 to 7)

GPT32EHm.GTDBD 4007 8098h + 0100h × m (m = 0 to 3)
GPT32Em.GTDBD 4007 8098h + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GTDBn is a 32-bit read/write register that functions as a buffer register for GTDVn.

Only GPT32EH and GPT32E have this register. GPT32 does not have this register. This register is read with the value after reset.

23.2.30 General PWM Timer Output Protection Function Status Register (GTSOS)

Address(es): GPT32EHm.GTSOS 4007 809Ch + 0100h x m (m = 0 to 3)
GPT32Em.GTSOS 4007 809Ch + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	SOS[1:0]	Output Protection Function Status	b1 b0 0 0: Normal operation 0 1: Protected state (set GTCCRA = 0 during transfer at trough or crest) 1 0: Protected state (set GTCCRA ≥ GTPR during transfer at trough) 1 1: Protected state (set GTCCRA ≥ GTPR during transfer at crest).	R
b31 to b2	—	Reserved	These bits are read as 0. Writing to these bits is ignored.	R

GTSOS is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (GTDTCR.TDE bit = 1) in triangle-wave mode.

Only GPT32EH and GPT32E have this register. GPT32 does not have this register. This register is read as 0000_0000h.

SOS[1:0] bits (Output Protection Function Status)

The SOS[1:0] bits indicate the status of the output protection function in triangle-wave PWM mode.

23.2.31 General PWM Timer Output Protection Function Temporary Release Register (GTSOTR)

Address(es): GPT32EHm.GTSOTR 4007 80A0h + 0100h x m (m = 0 to 3)
GPT32Em.GTSOTR 4007 80A0h + 0100h × m (m = 4 to 7)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SOTR	Output Protection Function Temporary Release	0: Do not release protected state 1: Release protected state.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTSOTR register temporarily releases the protected state of GTIOCB pin output when output protection is set. The protected state can be released only when GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA ≥ GTPR has occurred during transfer at trough). The protected state cannot be released in any other case.

Only GPT32EH and GPT32E have this register. GPT32 does not have this register. This register is read as 0000_0000h.

SOTR bit (Output Protection Function Temporary Release)

The SOTR bit specifies whether to temporarily release the protected state of the GTIOCB pin output in an output protected state. When the SOTR bit is set to 1, the output protection function is canceled from the first trough. When the SOTR bit is set to 0, output protection resumes from the first trough.

23.2.32 Output Phase Switching Control Register (OPSCR)

Address(es): GPT_OPS.OPSCR 4007 8FF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	—	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	UF	Input Phase Soft Setting	These bits set the input phase from the software settings. Setting these bits is valid when the OPSCR.FB bit = 1.	R/W
b1	VF			R/W
b2	WF			R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	U	Input U-Phase Monitor	These bits monitor the state of the input phase: OPSCR.FB = 0: External input monitoring by PCLKD OPSCR.FB = 1: Software settings (UF/VF/WF).	R
b5	V	Input V-Phase Monitor		R
b6	W	Input W-Phase Monitor		R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	EN	Enable-Phase Output Control	0: Do not output (Hi-Z on external pin) 1: Output.*1	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	FB	External Feedback Signal Enable	This bit selects the input phase from the software settings or external input: 0: Select the external input 1: Select the software settings (OPSCR.UF, VF, WF).	R/W
b17	P	Positive-Phase Output (P) Control	0: Output level signal 1: Output PWM signal (PWM of GPT32EH0).	R/W
b18	N	Negative-Phase Output (N) Control	0: Output level signal 1: Output PWM signal (PWM of GPT32EH0).	R/W
b19	INV	Invert-Phase Output Control	0: Output positive logic (active-high) 1: Output negative logic (active-low).	R/W
b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b21	ALIGN	Input Phase Alignment	0: Align input phase to PCLKD 1: Align input phase to PWM.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Disabled Source Selection	b25 b24 0 0: Select Group A output disable source 0 1: Select Group B output disable source 1 0: Select Group C output disable source 1 1: Select Group D output disable source.	R/W
b26	GODF	Group Output Disable Function	0: Ignore this bit function 1: Clear the OPSCR.EN bit on group disable.*1	R/W
b28, b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b29	NFEN	External Input Noise Filter Enable	0: Do not use a noise filter on the external input 1: Use a noise filter on the external input.	R/W
b31, b30	NFCS[1:0]	External Input Noise Filter Clock Selection	Noise filter sampling clock setting of the external input: b31 b30 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64.	R/W

Note 1. When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP bit is high, the OPSCR.EN bit is cleared to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF, VF, WF bits (Input Phase Soft Setting)

The UF, VF, and WF bits set the input phase from the software settings. When OPSCR.FB bit = 1, these bits are valid. The set value of the UF/VF/WF bits take the place of the U/V/W external inputs.

U, V, W bits (Input Phase Monitor)

When OPSCR.FB bit = 0, external inputs that are synchronized by PCLKD are monitored by the U, V, and W bits. When OPSCR.FB bit = 1, the OPSCR.U, OPSCR.V, and OPSCR.W bits can read the OPSCR.UF, OPSCR.VF, and OPSCR.WF bits.

EN bit (Enable-Phase Output Control)

The EN bit controls the output enable signal output phase (positive phase/reverse phase).

When OPSCR.EN bit = 1, the signal waveform is output.

When OPSCR.EN bit = 0, first set OPSCR.FB, OPSCR.UF/VF/WF (software setting is selected), OPSCR.P/N, OPSCR.INV, OPSCR.RV, OPSCR.ALIGN, OPSCR.GRP, OPSCR.GODF, OPSCR.NFEN, and OPSCR.NFCS. Then, set this bit to 1. Also, when OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP bit is high, the OPSCR.EN bit is cleared to 0.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects the level signal output or PWM signal output for the positive-phase output (GTOUUP pin, GTOVUP pin, GTOWUP pin).

N bit (Negative-Phase Output (N) Control)

The N bit selects the level signal output or PWM signal output for the negative-phase output (GTOULO pin, GTOVLO pin, GTOWLO pin).

INV bit (Invert-Phase Output Control)

The INV bit selects either positive logic (active-high) output or negative logic (active-low) output for the output phase.

ALIGN bit (Input Phase Alignment)

The ALIGN bit selects PCLKD or PWM for the sampling of the input phase (input phase is specified in the OPSCR.FB bit).

When OPSCR.ALIGN bit = 0, input phase is aligned to PCLKD.

Note: When PWM output is selected (OPSCR.P/N = 1) and the PCLKD input phase is aligned, the PWM pulse might be short-pulsed.

Note: When OPSCR.ALIGN bit = 1, input phase is aligned with PWM output.

GRP[1:0] bits (Output Disabled Source Selection)

The GRP[1:0] bits select the output disable source (A to D).

GODF bit (Group Output Disable Function)

When the GODF bit = 1 and signal value selected by the OPSCR.GRP bit is high, the OPSCR.EN bit is cleared to 0. When the GODF bit = 0, the bit is ignored.

NFEN bit (External Input Noise Filter Enable)

The NFEN bit selects the noise filter for external input.

When OPSCR.NFEN = 0, a noise filter is not used for the external input.

When OPSCR.NFEN = 1, a noise filter is used for the external input.

Note: When this bit is switched, because an unintentional internal edge occurs, first set the OPSCR.EN bit to 0.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

The NFCS[1:0] bits select the clock for the external input noise filter. When the OPSCR.NFEN bit = 1, noise filter sampling clock setting for external input is enabled.

Note: After setting the NFCS[1:0] bits, wait 2 cycles of the selected sampling clock, then set OPSCR.EN to 1.

23.3 Operation

23.3.1 Basic Operation

Each channel has a 32-bit timer that performs a periodic count operation using the count clock and hardware sources. The count function provides both up-counting and down-counting. The GTPR register controls the count cycle. When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the associated pin GTIOCA or GTIOCB can be changed. GTCCRA or GTCCRB can be used as an input capture register with hardware resources.

GTCCRC and GTCCRD can function as buffer registers for GTCCRA. GTCCRE and GTCCRF can function as buffer registers for GTCCRB.

23.3.1.1 Counter operation

(1) Counter start and stop

The counter of each channel starts the count operation when GTCR.CST is set to 1. The GTCR.CST bit value is changed by following sources:

- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit is set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit is set to 1
- The hardware source selected in the GTSSR register
- The hardware source selected in the GTPSR register.

(2) Periodic count operation in up-counting by count clock

The GTCNT counter in each channel starts up-counting when the associated GTCR.CST bit is set to 1 with the GTUPSR and GTDNSR registers set to 0000 0000h. When the GTCNT value changes from the GTPR value to 0 (overflow), the GTST.TCFPO flag is set to 1. When GTCNT overflows, up-counting resumes from 0000 0000h.

Figure 23.3 shows an example of a periodic count operation in up-counting.

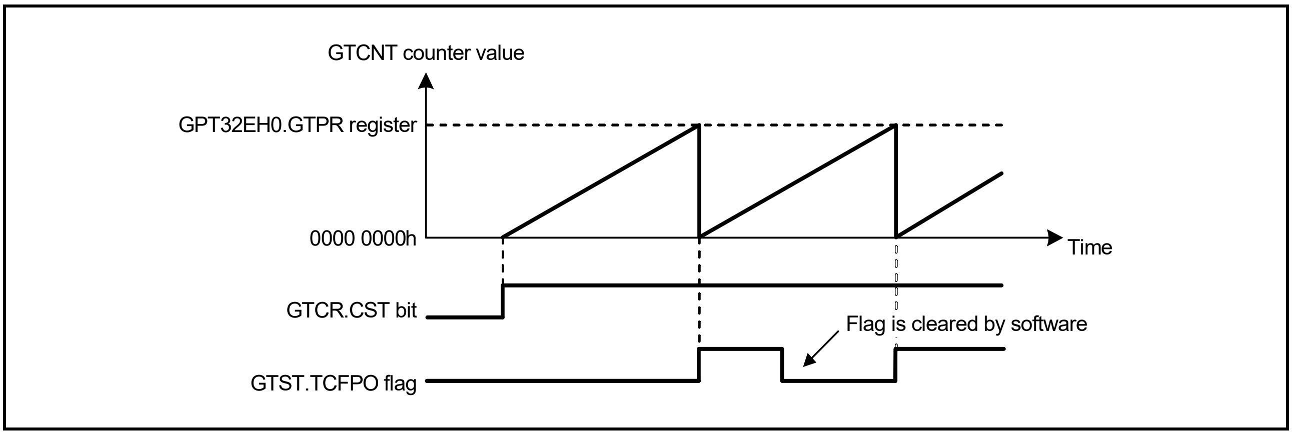
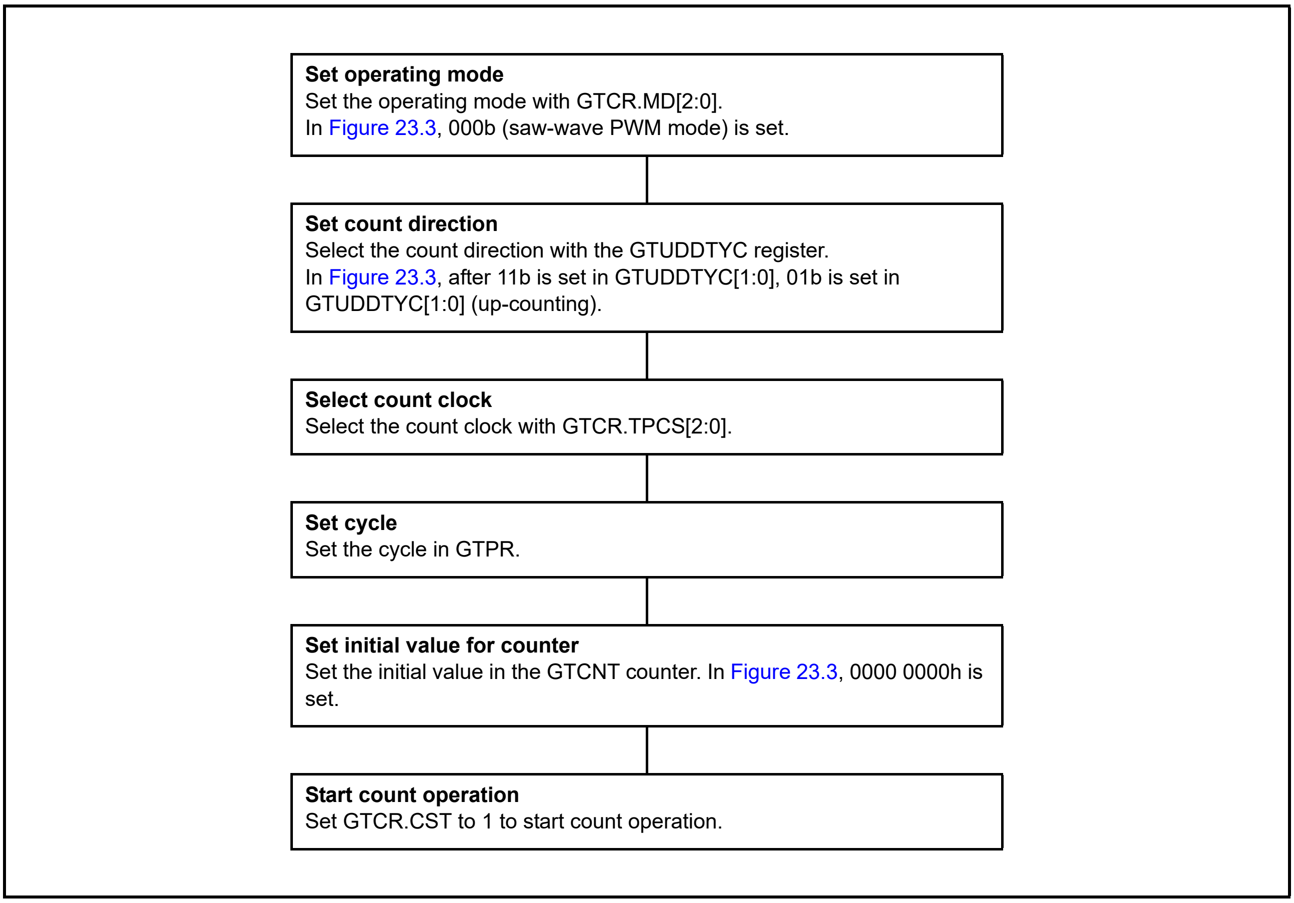


Figure 23.3 Example of periodic count operation in up-counting by the count clock

Figure 23.4 shows an example setting for periodic count operation in up-counting.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.3, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction with the GTUDDTYC register.
In Figure 23.3, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter. In Figure 23.3, 0000 0000h is set.

Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.4 Example setting for a periodic count operation in up-counting by the count clock

(3) Periodic count operation in down-counting by count clock

The GTCNT counter in each channel can perform down-counting by setting GTUDDTYC.UD with the GTUPSR and GTDNSR registers set to 0000 0000h. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1. When the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 23.5 shows an example of periodic count operation in down-counting by the count clock.

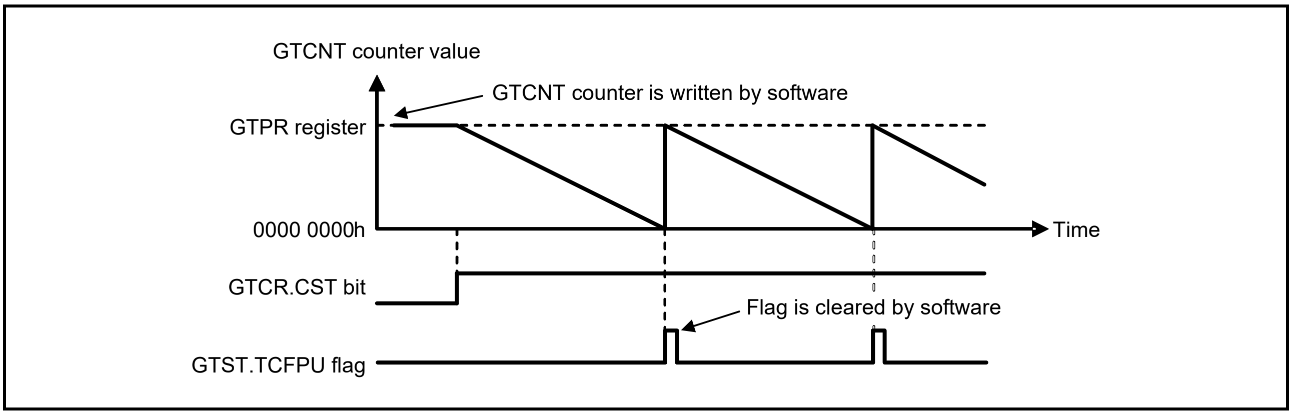
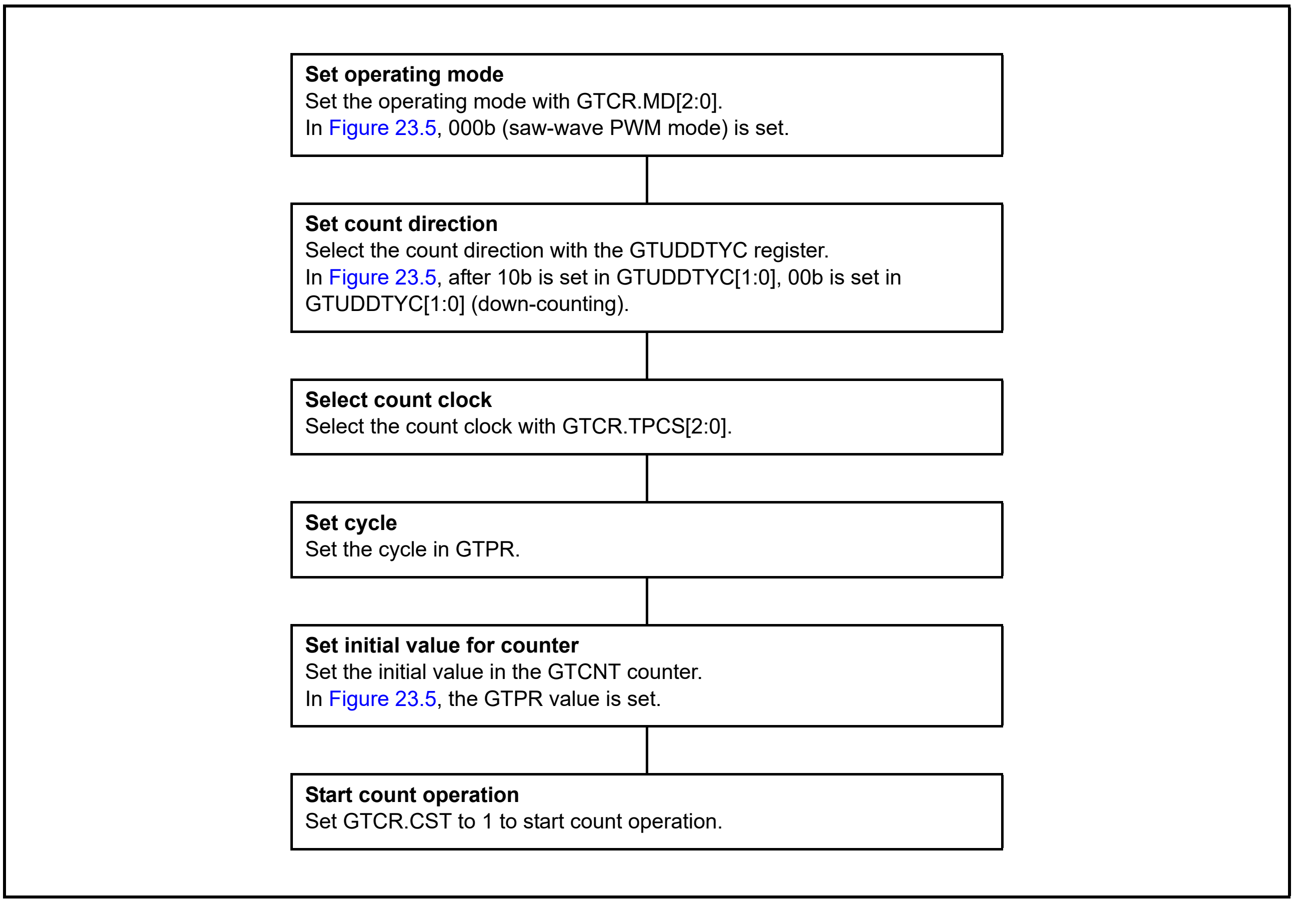


Figure 23.5 Example of periodic count operation in down-counting by the count clock

Figure 23.6 shows an example setting for periodic count operation in down-counting by the count clock.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.5, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction with the GTUDDTYC register.
In Figure 23.5, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.5, the GTPR value is set.

Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.6 Example setting for periodic count operation in down-counting by count clock

(4) Event count operation in up-counting using hardware sources

The GTCNT counter in each channel can perform up-counting using hardware sources as set in GTUPSR.

When GTUPSR is set to enable, the count clock selected in GTCR.TPCS[2:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The overflow behavior for up-counting using hardware sources is the same as for up-counting by the count clock.

When GTCR.CST bit is set to 1 to count up using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified in GTCR.TPCS[2:0] because the count operation is synchronized by the count clock selected in GTCR.TPCS[2:0]. Set GTCR.TPCS[2:0] to 000b to count up with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 23.7 shows an example of a periodic count operation in up-counting by a hardware source (rising edge of GTETRGA pin).

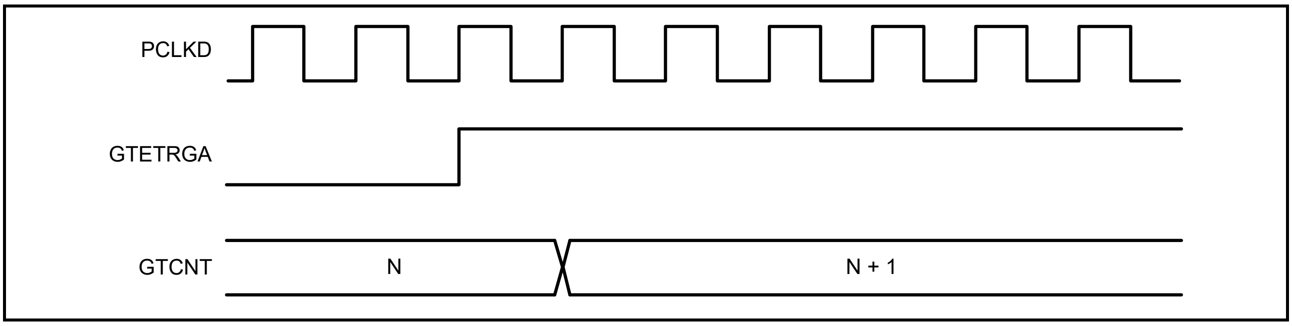
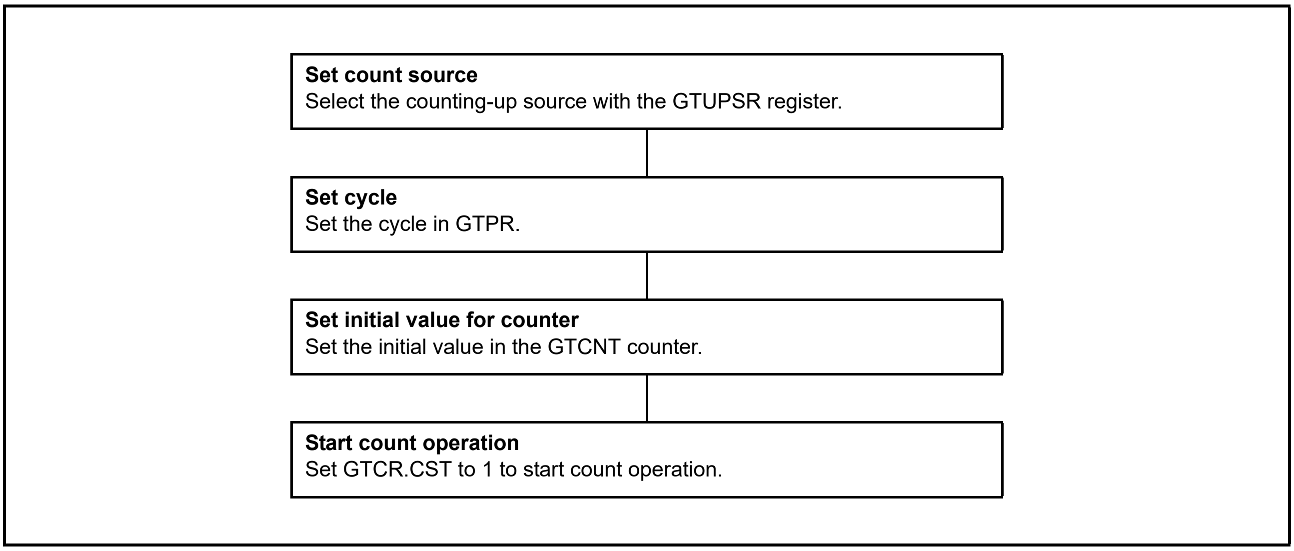


Figure 23.7 Example of periodic count operation in up-counting using hardware sources

Figure 23.8 shows an example setting for periodic count operation in up-counting by a hardware source.



Set count source
Select the counting-up source with the GTUPSR register.

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.8 Example setting for an event count operation in up-counting using hardware sources

(5) Event count operation in down-counting using hardware sources

The GTCNT counter in each channel can perform down-counting using hardware sources set in the GTDNSR register.

When GTDNSR is set to enable, the count clock selected in GTCR.TPCS[2:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, GTCNT counter value does not change. The underflow behavior for down-counting using hardware sources is the same as for down-counting by the count clock.

When GTCR.CST bit is set to 1 to count down using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified in GTCR.TPCS[2:0] because the count operation is synchronized with the count clock selected by GTCR.TPCS[2:0]. Set GTCR.TPCS[2:0] to 000b to count down with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 23.9 shows an example of a periodic count operation in down-counting by a hardware source (rising edge of GTETRGA pin).

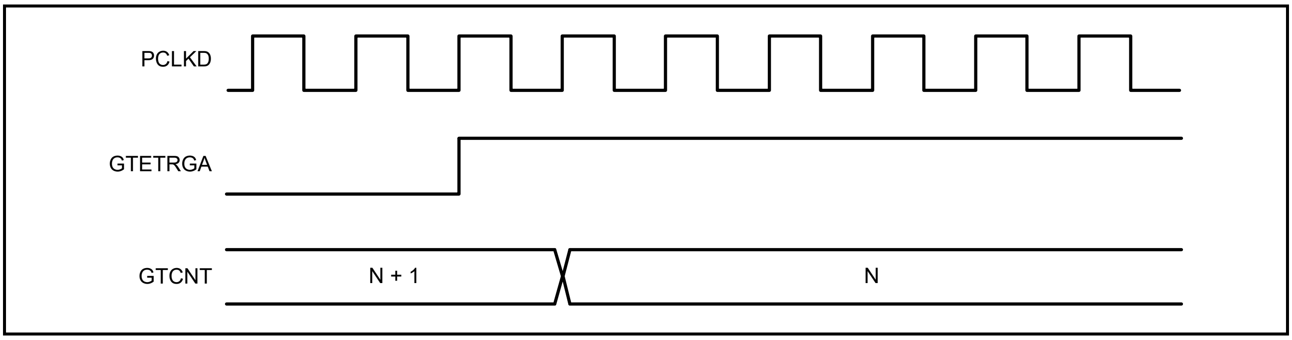
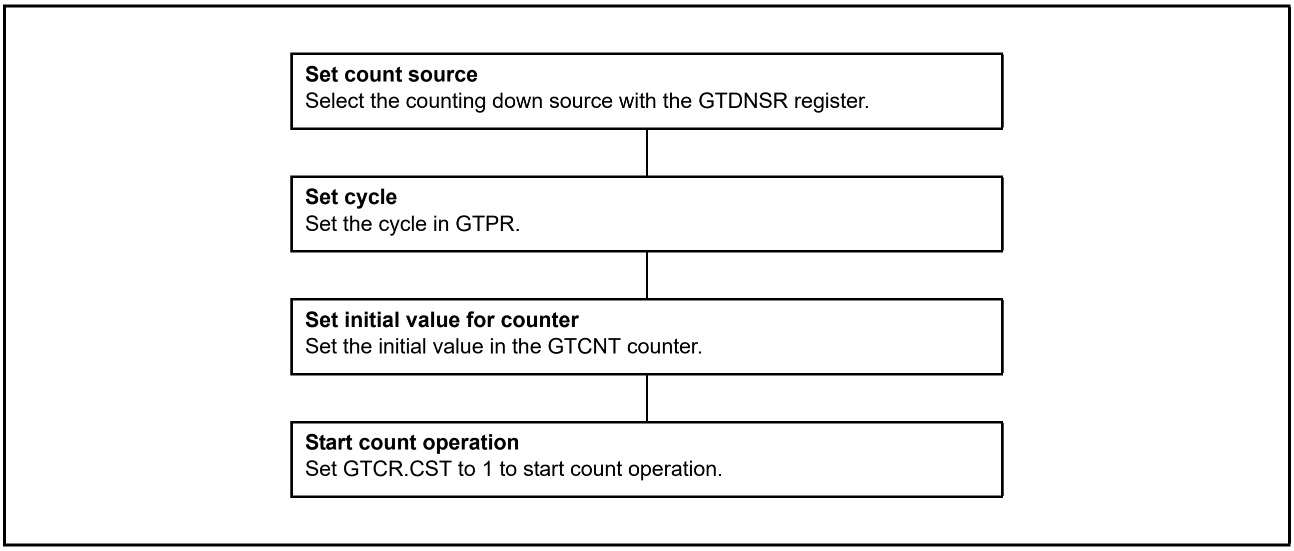


Figure 23.9 Example of event count operation in down-counting using hardware sources

Figure 23.10 shows an example setting for a periodic count operation in down-counting using a hardware source.



Set count source
Select the counting down source with the GTDNSR register.

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.10 Example setting for an event count operation in down-counting using hardware sources

(6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit is set to 1
- The hardware source selected in GTCSR register.

Writing to the GTCNT register is prohibited during count operation. The GTCNT counter can be cleared both by writing 1 to the GTCLR and by the clear request of hardware sources, whether GTCNT is counting (GTCR.CST = 1) or not (GTCR.CST = 0).

For saw waves selected by setting GTCR.MD[2:0] and the count direction flag showing down-counting (GTST.TUCF = 0), the GTCNT register is set to the value of the GTPR register when writing 1 to the GTCLR register or when clearing by hardware sources is performed. When not in saw wave mode and down-counting, the GTCNT register is set to 0 when writing 1 to the GTCLR register and when clearing by hardware sources is performed.

In event count operation when at least 1 bit in GTUPSR or GTDNSR is set to 1, after clear sources occur, both writing to GTCLR register and clearing by hardware sources are performed immediately, synchronized with PCLKD. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[2:0].

23.3.1.2 Waveform output by compare match

Compare match means that the GTCNT counter value matches the value of GTCCRA or GTCCRB. When a compare match occurs, the compare match flag is generated synchronously with the count clock, including the event count. At the same time the GPT can output low, high, or toggle output from the associated GTIOCA or GTIOCB output pin. In addition, the GTIOCA or GTIOCB pin output can be low, high, or toggle at the cycle end, which is determined by GTPR.

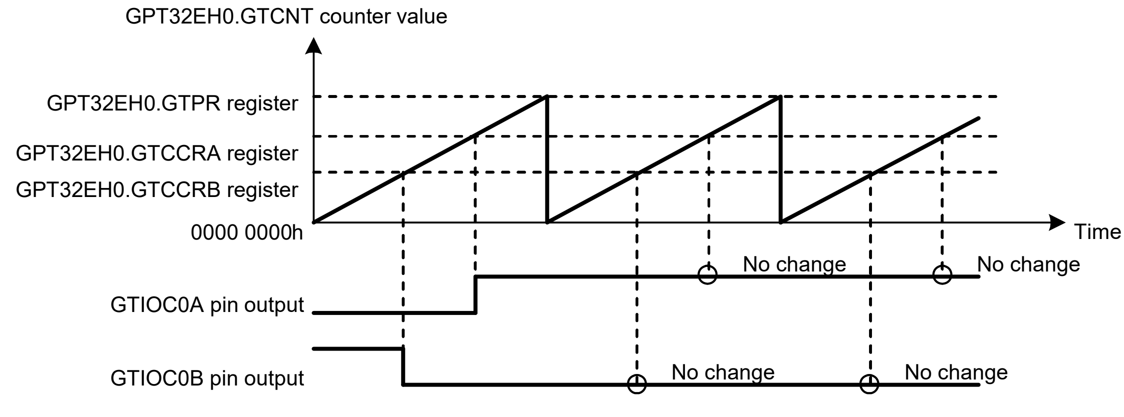
The cycle end is:

- For saw waves in up-counting — when GTCNT changes from the GTPR value to 0 (overflow)
- For saw waves in down-counting — when GTCNT changes from 0 to the GTPR value (underflow)
- For saw waves — when the GTCNT counter is cleared
- For triangle waves — when the GTCNT changes from 0 to 1 (trough).

(1) Low output and high output

Figure 23.11 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GPT32EH0.GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOC0A pin by a GPT32EH0.GTCCRA compare match, and low is output from the GTOC0B pin by a GPT32EH0.GTCCRB compare match. The pin level does not change when the specified level and pin level match.



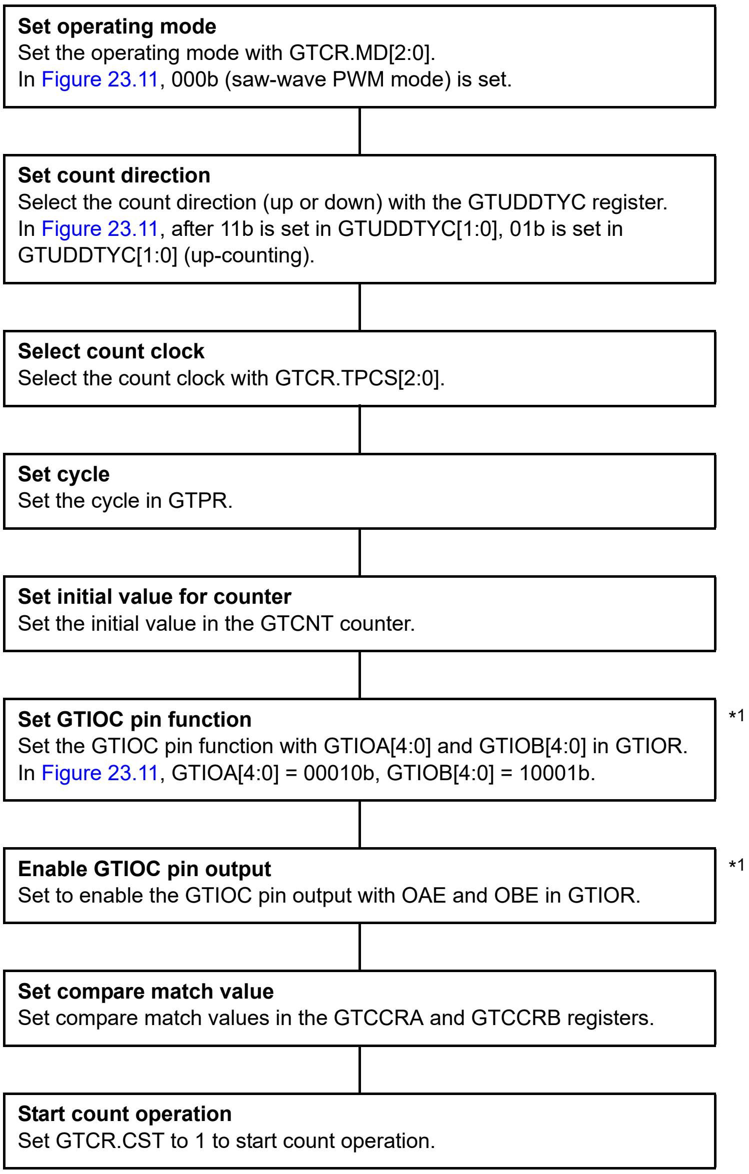
[Setting examples]

GPT32EH0.GTIOR.GTIOA[4:0] bits: Initial output is low, high output at compare match, output retained at cycle end

GPT32EH0.GTIOR.GTIOB[4:0] bits: Initial output is high, low output at compare match, output retained at cycle end

Figure 23.11 Example of low output and high output operation

Figure 23.12 shows an example setting for low output and high output operation.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.11, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.11, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function *1
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.11, GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.

Enable GTIOC pin output *1
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set compare match value
Set compare match values in the GTCCRA and GTCCRB registers.

Start count operation
Set GTCR.CST to 1 to start count operation.

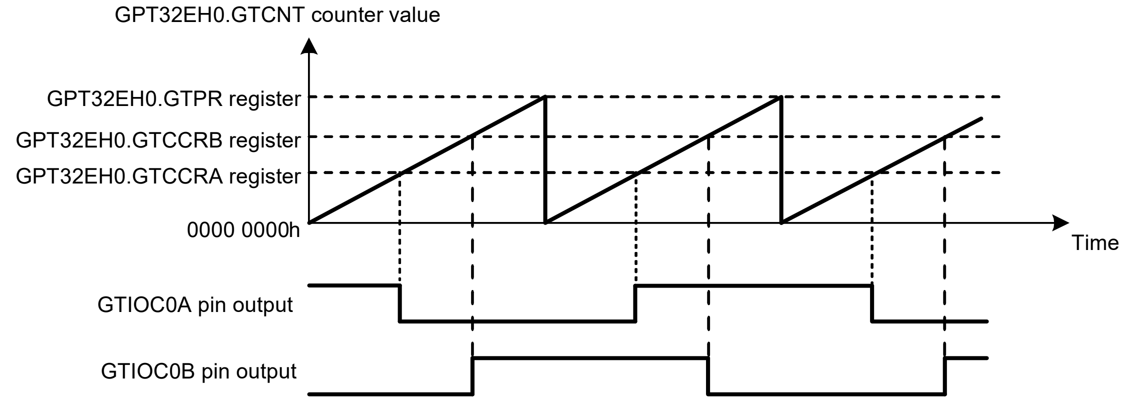
Note 1. When PWM delay generation circuit is used, reverse the setting order of GTIOC pin function and enable GTIOC pin output.

Figure 23.12 Example setting for low output and high output operation

(2) Toggled output

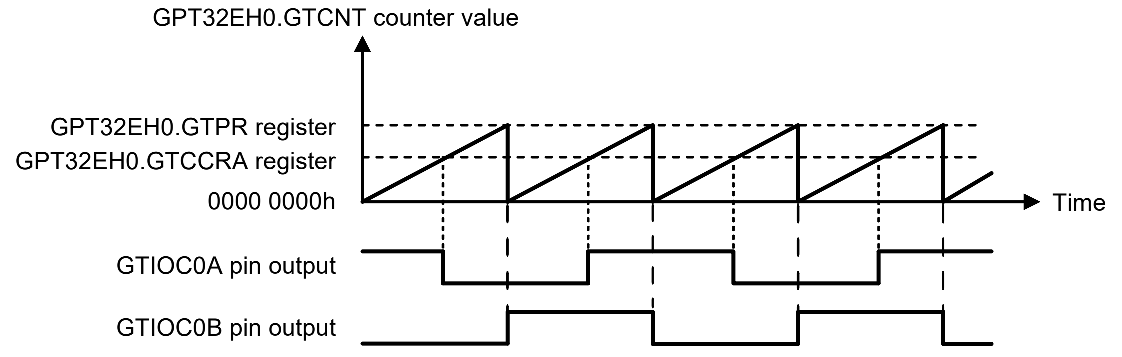
Figure 23.13 and Figure 23.14 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB. In Figure 23.13, the GPT32EH0.GTCNT counter performs up-counting, and settings are made so that the GTIOC0A pin output by a GPT32EH0.GTCCRA compare match and GTIOC0B pin output by a GPT32EH0.GTCCRB compare match are toggled.

In Figure 23.14, the GPT32EH0.GTCNT counter performs up-counting, and settings are made so that the GTIOC0A output is toggled by a compare match of GPT32EH0.GTCCRA and the GTIOC0B output is toggled at the cycle end.



[Setting examples]
GPT32EH0.GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GPT32EH0.GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at cycle end

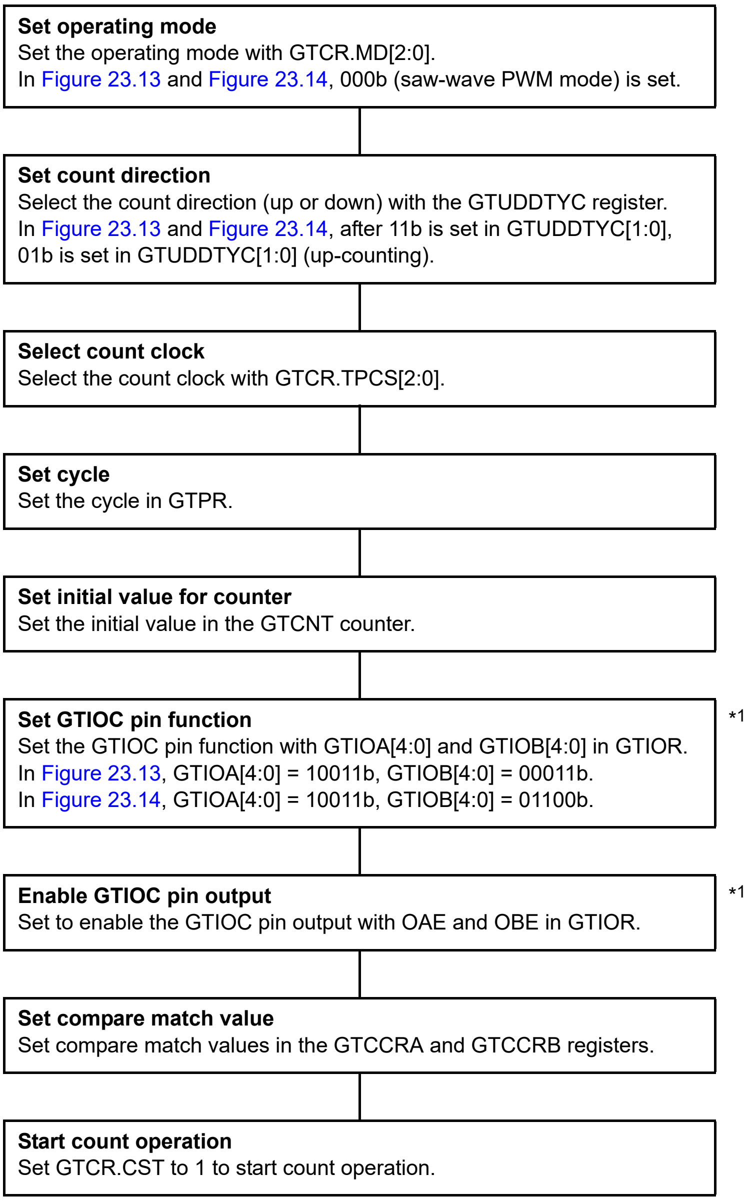
Figure 23.13 Example of toggled output operation (1)



[Setting examples]
GPT32EH0.GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GPT32EH0.GTIOR.GTIOB[4:0] bits: Initial output is low, output retained at compare match, output toggled at cycle end

Figure 23.14 Example of toggled output operation (2)

Figure 23.15 shows an example setting for toggled output operation.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.13 and Figure 23.14, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.13 and Figure 23.14, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function *1
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.13, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b.
In Figure 23.14, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.

Enable GTIOC pin output *1
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set compare match value
Set compare match values in the GTCCRA and GTCCRB registers.

Start count operation
Set GTCR.CST to 1 to start count operation.

Note 1. When PWM delay generation circuit is used, reverse the setting order of GTIOC pin function and enable GTIOC pin output.

Figure 23.15 Example setting for toggled output operation

23.3.1.3 Input capture function

The GTCNT counter value can be transferred to either GTCCRA or GTCCRB on detection of the hardware source that is set in GTICASR and GTICBSR.

Figure 23.16 shows an example of the input capture function.

In this example, the GPT32EH0.GTCNT counter performs up-counting by the count clock, and settings are made so that an input capture is performed to GTICCRA at both edges of the GTIOC0A input pin and to GTICCRB on the rising edge of the GTIOC0B input pin.

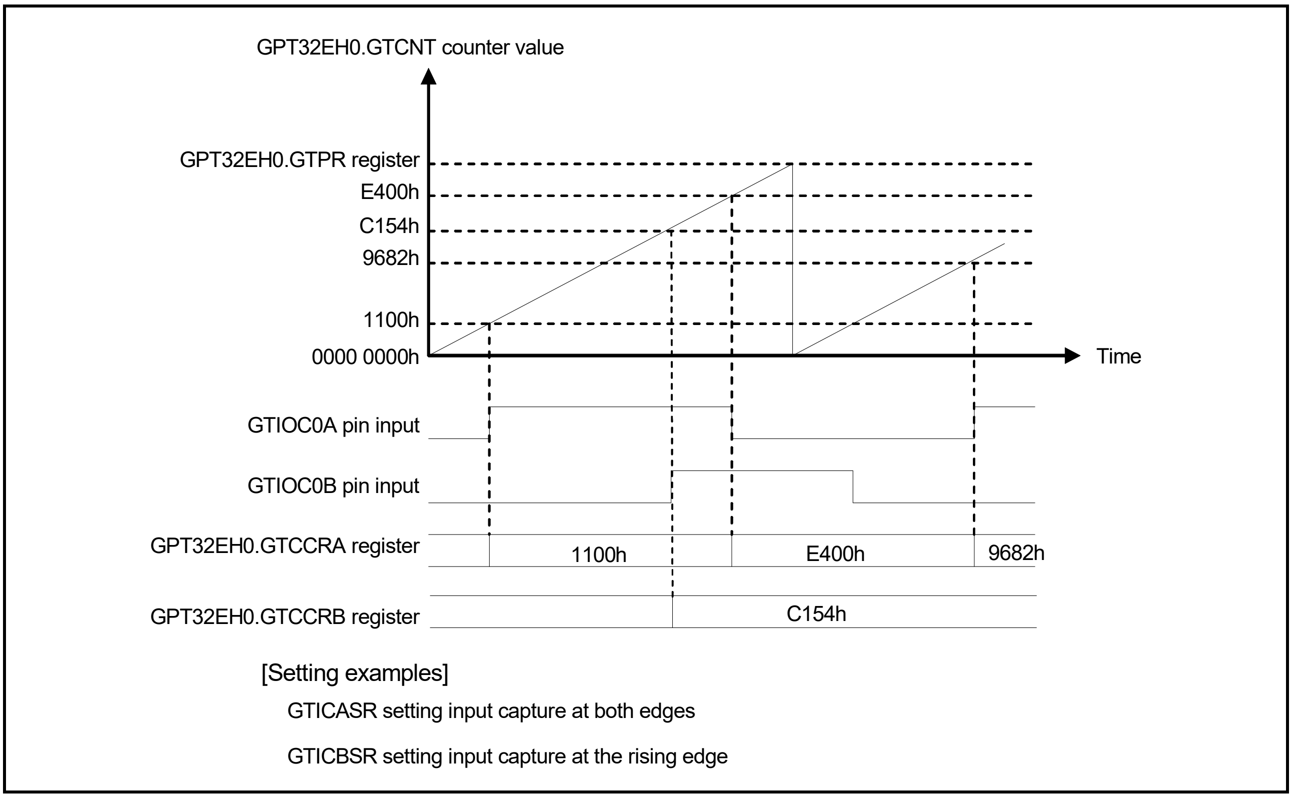
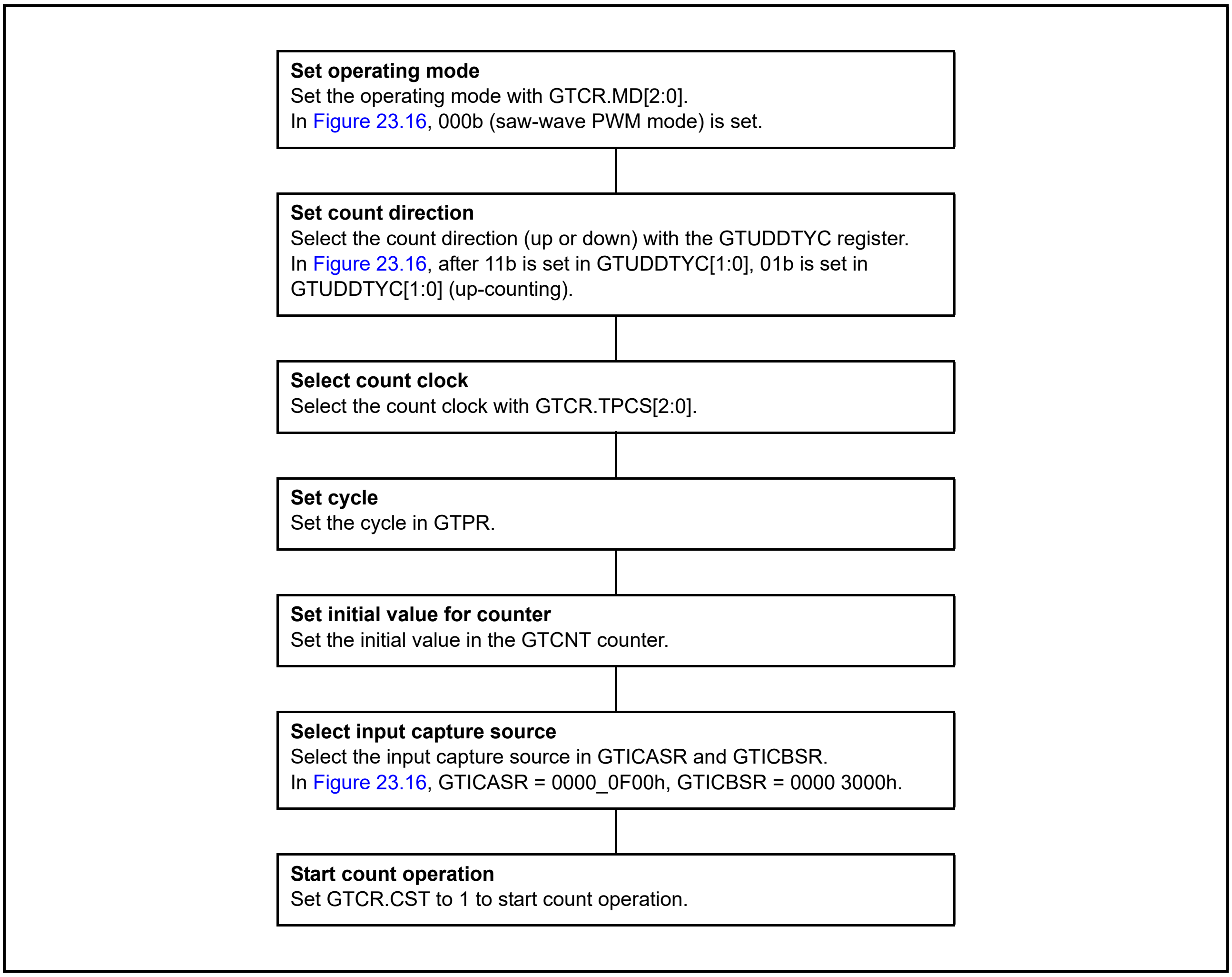


Figure 23.16 Example of input capture operation

Figure 23.17 shows an example setting for an input capture operation with count operation by the count clock.



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.16, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.16, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Select input capture source
Select the input capture source in GTICASR and GTICBSR.
In Figure 23.16, GTICASR = 0000_0F00h, GTICBSR = 0000 3000h.

Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.17 Example setting for input capture operation

23.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR, GTPBR, and GTPDBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF
- GTADTRA, GTADTBRA, and GTADTDBRA
- GTADTRB, GTADTBRB, and GTADTDBRB.

The following buffer operations can be set with GTDTCR:

- GTDVU and GTDBU
- GTDVU and GTDBD.

23.3.2.1 GTPR register buffer operation

GTPBR can function as a buffer register for GTPR, and GTPDBR can function as a buffer register for GTPBR (double-buffer register for GTPR). The buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count, and at a trough in triangle-wave mode.

In saw-wave mode or in event count, the buffer transfer is performed when the following counter clear operations occur during counting:

- Clear by hardware sources (the clear source is selected in GTCSR[23:0])
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR[n] bit is set to 1, n = channel number).

To set GTPR to function as double buffer, set GTBER.PR[1:0] to 10b or 11b. To set GTPR to not function as a buffer, set GTBER.PR[1:0] to 00b.

Figure 23.18 to Figure 23.20 show examples of GTPR buffer operation, and Figure 23.21 shows an example setting for GTPR buffer operation.

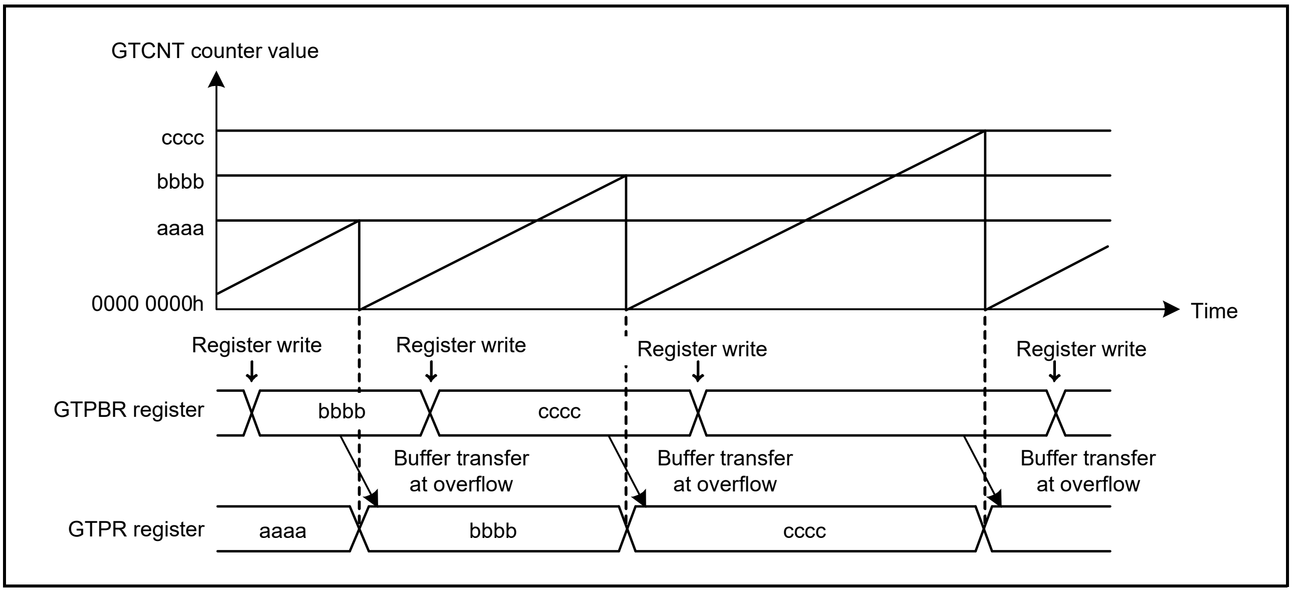


Figure 23.18 Example of GTPR buffer operation with saw waves in up-counting

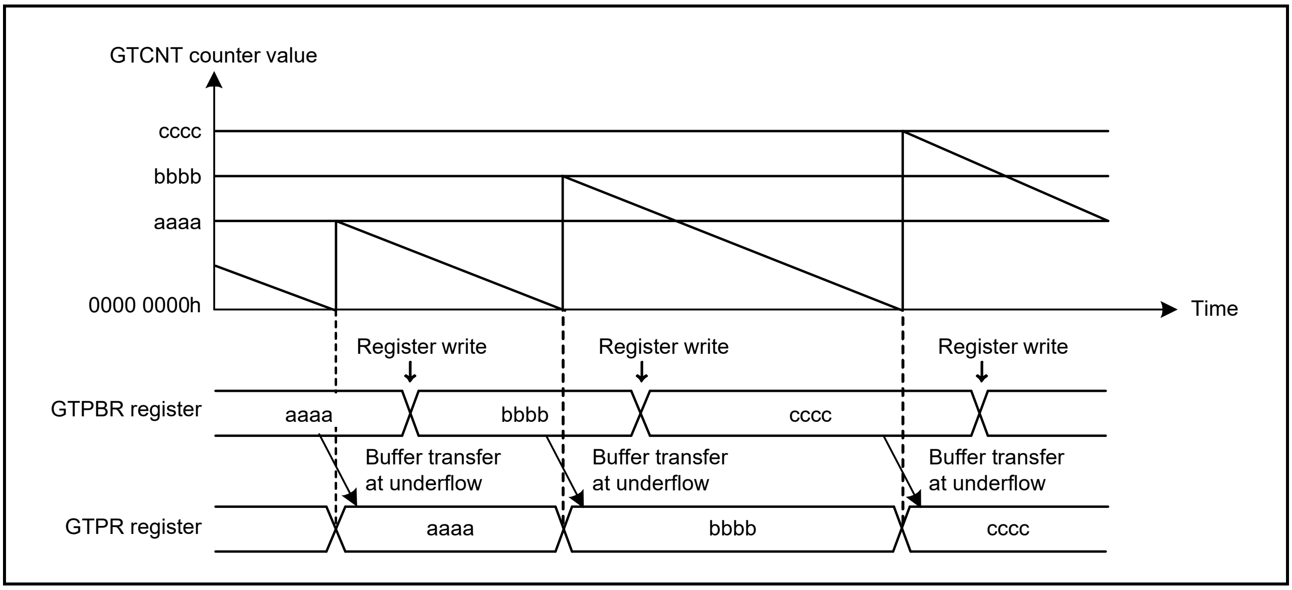


Figure 23.19 Example of GTPR buffer operation with saw waves in down-counting

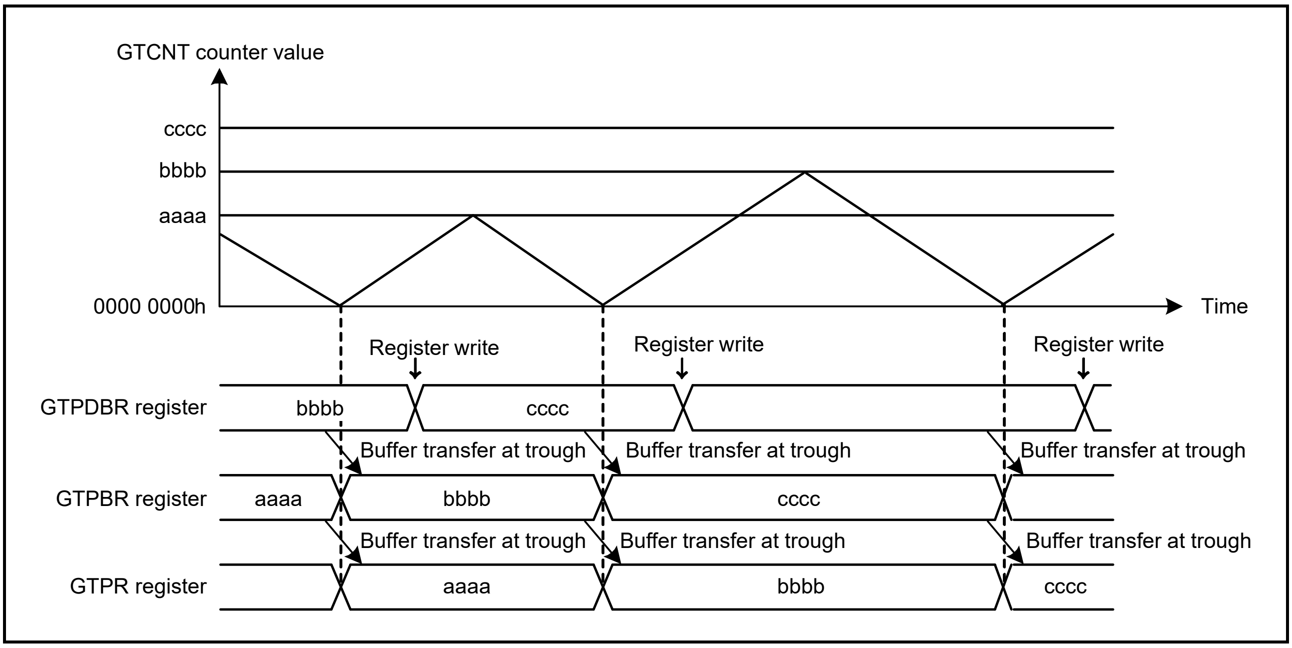
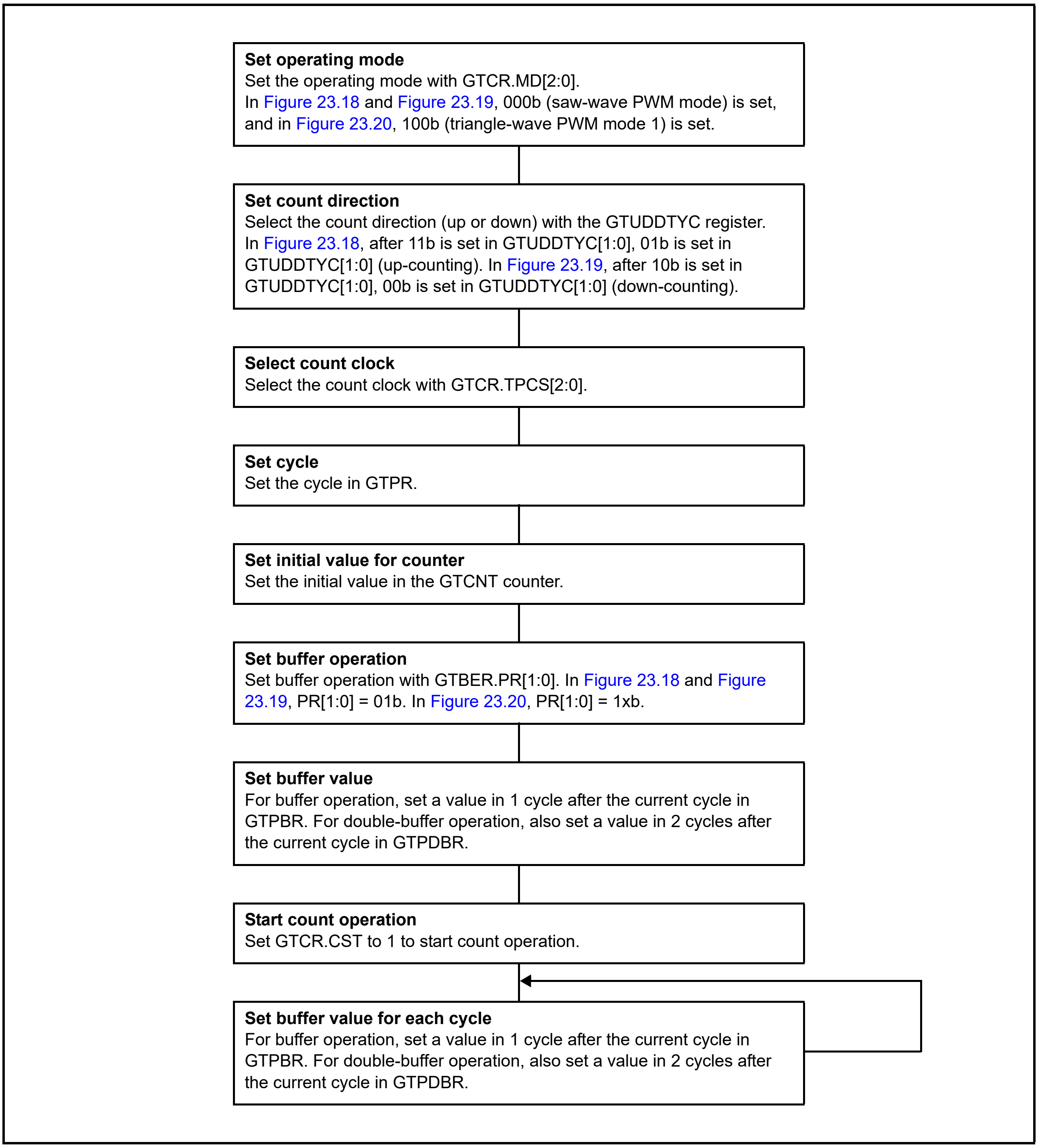


Figure 23.20 Example of GTPR double buffer operation with triangle waves



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.18 and Figure 23.19, 000b (saw-wave PWM mode) is set, and in Figure 23.20, 100b (triangle-wave PWM mode 1) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register. In Figure 23.18, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting). In Figure 23.19, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set buffer operation
Set buffer operation with GTBER.PR[1:0]. In Figure 23.18 and Figure 23.19, PR[1:0] = 01b. In Figure 23.20, PR[1:0] = 1xb.

Set buffer value
For buffer operation, set a value in 1 cycle after the current cycle in GTPBR. For double-buffer operation, also set a value in 2 cycles after the current cycle in GTPDBR.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set a value in 1 cycle after the current cycle in GTPBR. For double-buffer operation, also set a value in 2 cycles after the current cycle in GTPDBR.

Figure 23.21 Example setting for GTPR buffer operation

23.3.2.2 Buffer operation for GTCCRA and GTCCRB

GTCCRC can function as the GTCCRA buffer register and GTCCRD can function as the GTCCRC buffer register (double-buffer register for GTCCRA). Similarly, GTCCRE can function as the GTCCRB buffer register and GTCCRF can function as the GTCCRE buffer register (double-buffer register for GTCCRB).

To set GTCCRA or GTCCRB to function as a double buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 10b or 11b. For single-buffer operation, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 01b. To set GTCCRA or GTCCRB to not function as a buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 00b.

(1) When GTCCRA or GTCCRB functions as an output compare register

Buffer transfer occurs in the following situations:

- Buffer transfer by overflow or underflow
  Buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count operation. In triangle-wave mode, buffer transfer is performed at a trough (triangle-wave PWM mode 1) or a crest and trough (triangle-wave PWM mode 2).
- Buffer transfer by counter clear
  In saw-wave mode or in event count operation, during counting, buffer transfer (which is the same as an overflow during up-counting or an underflow during down-counting) is performed by the counter clear sources the same as shown in section 23.3.2.1, GTPR register buffer operation. In triangle-wave mode, buffer transfer is not performed by the counter clear.
- Forcible buffer transfer
  When GTBER.CCRSWT bit is set to 1 while the count operation is stopped, the GTCCRA and the GTCCRB register buffer transfer is performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode. Additionally, buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave 1 shot pulse mode or triangle-wave PWM mode 3.

Figure 23.22 to Figure 23.24 show examples of GTCCRA and GTCCRB buffer operation and Figure 23.25 shows an example setting for GTCCRA and GTCCRB buffer operation.

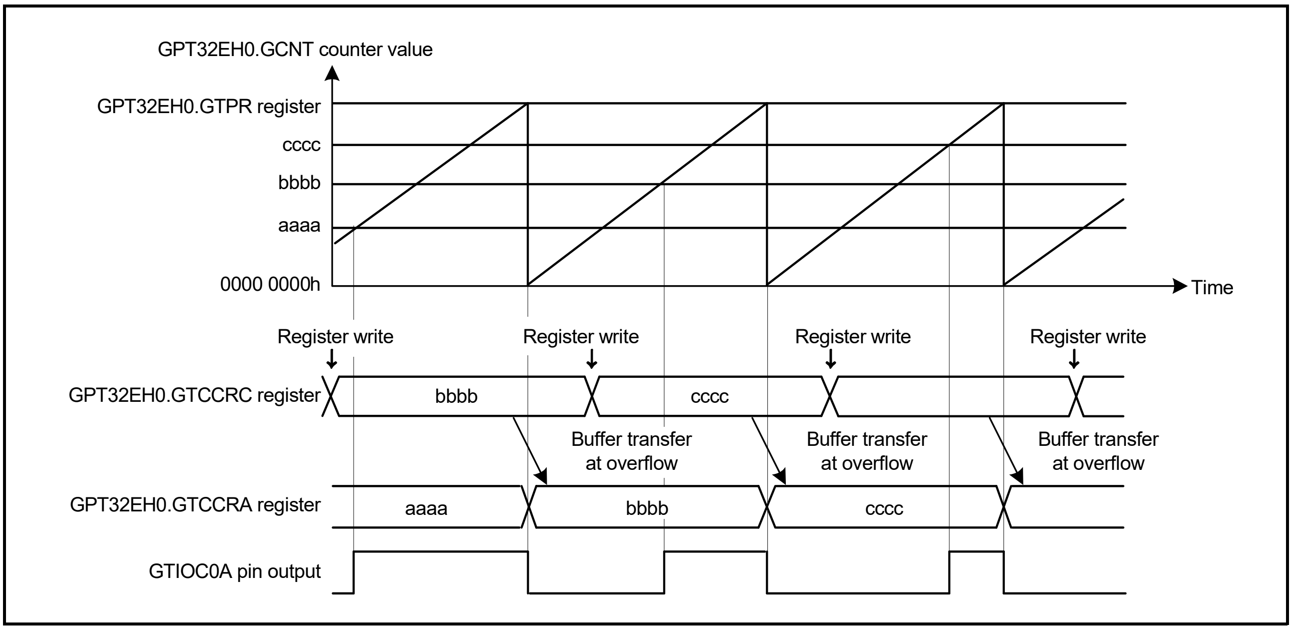


Figure 23.22 Example of GTCCRA and GTCCRB buffer operation with output compare, saw waves in up-counting, high output at GTCCRA compare match, and low output at cycle end

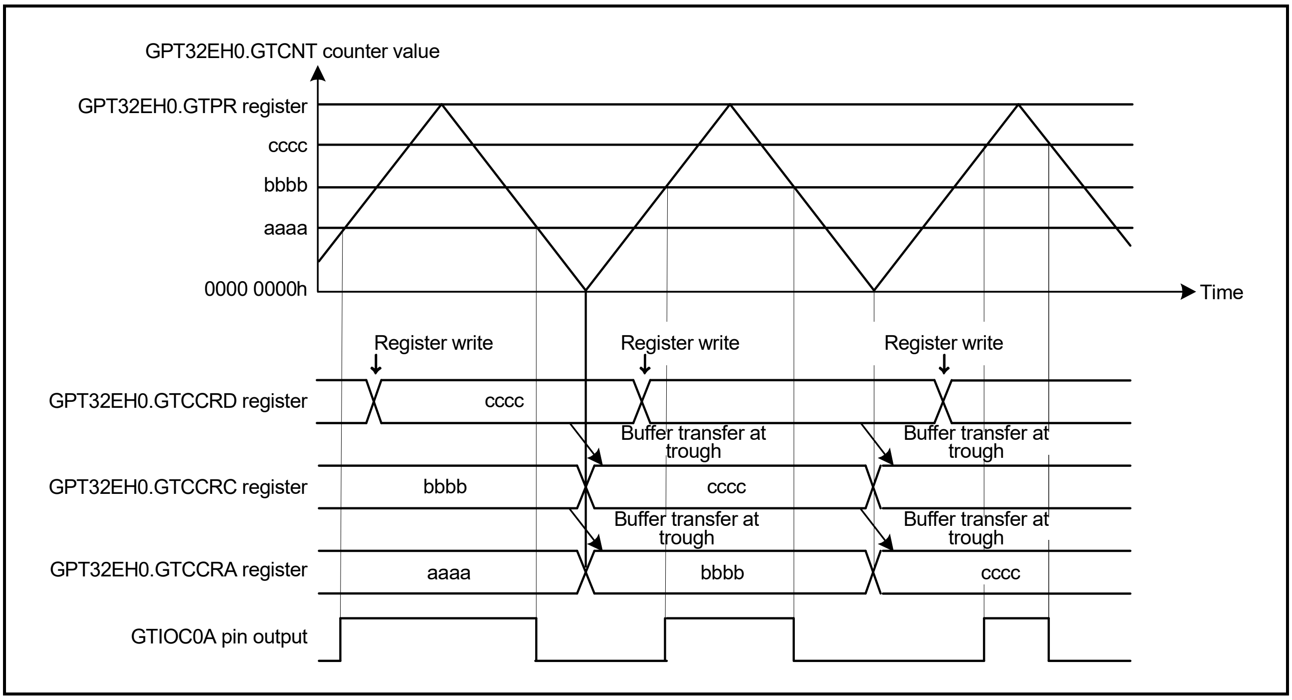


Figure 23.23 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end

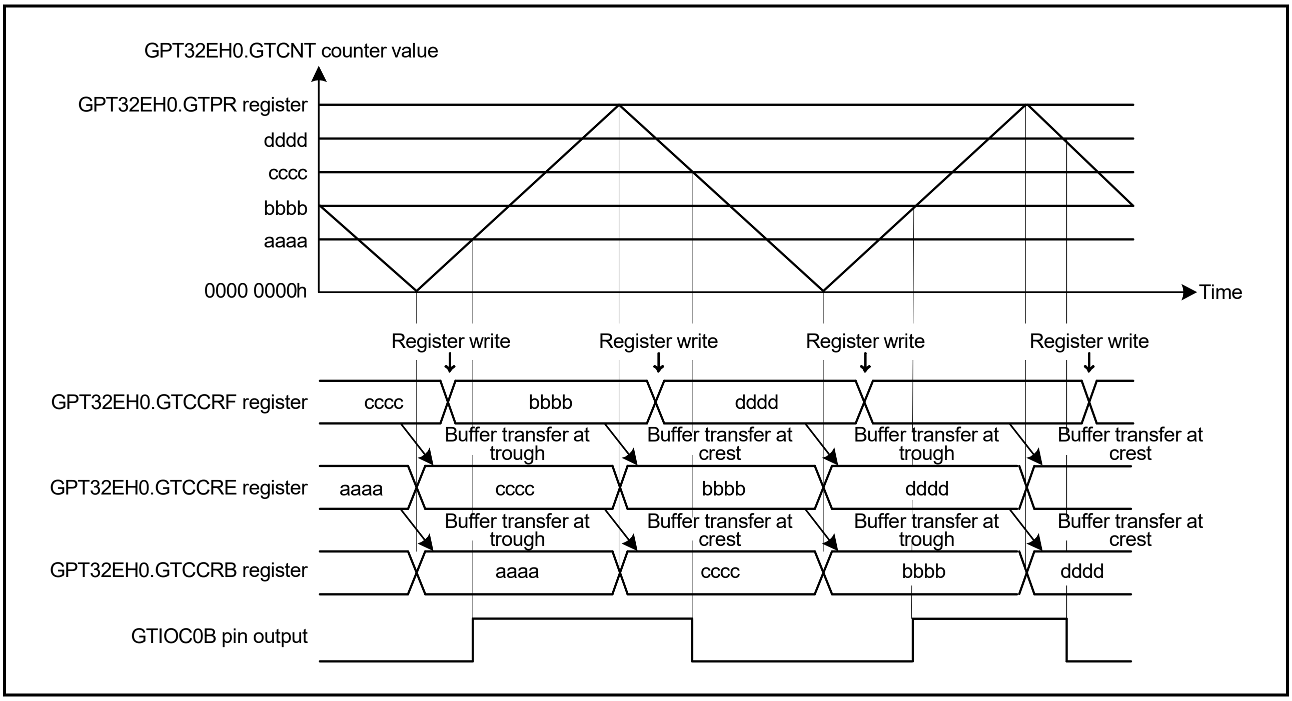
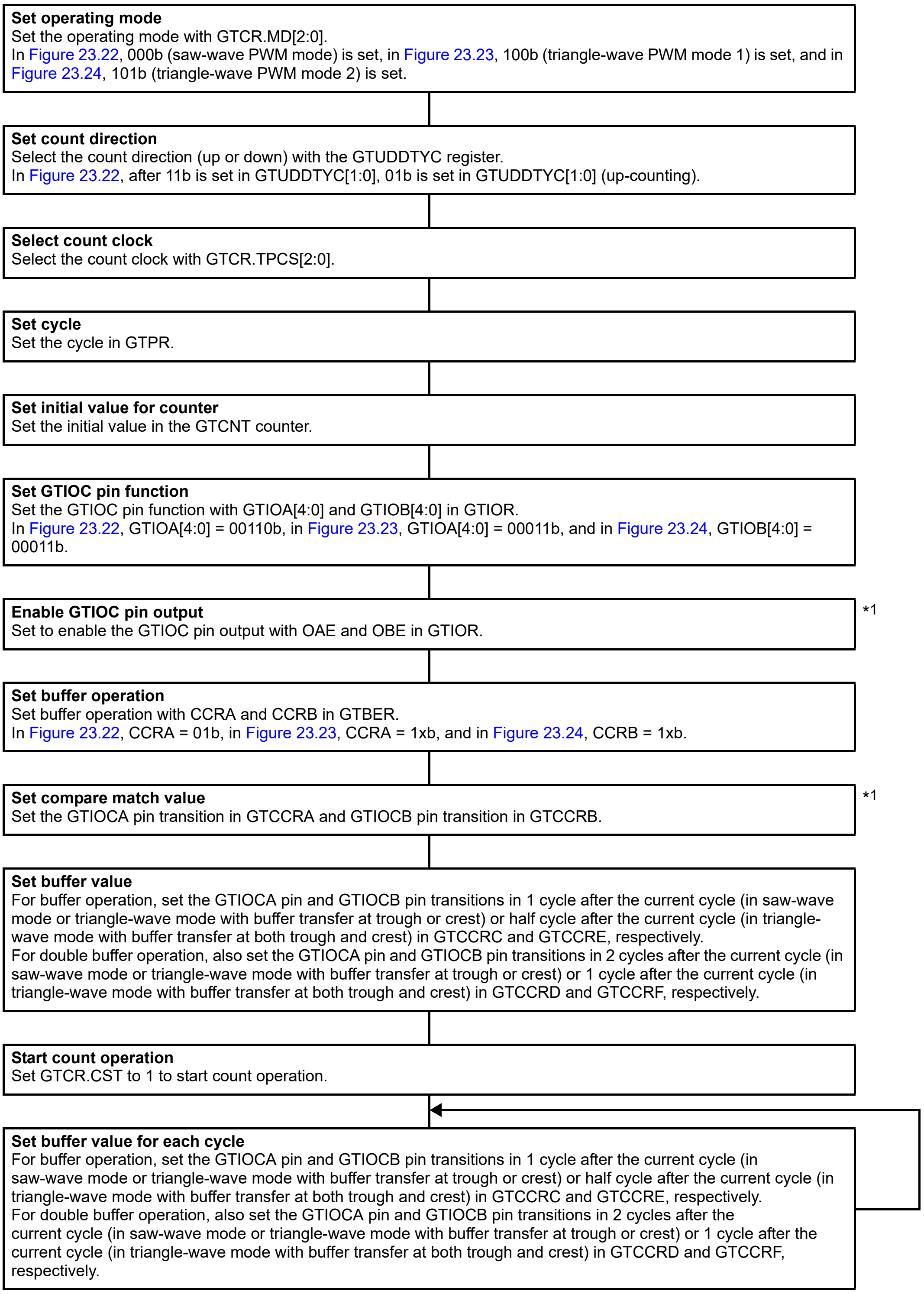


Figure 23.24 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.22, 000b (saw-wave PWM mode) is set, in Figure 23.23, 100b (triangle-wave PWM mode 1) is set, and in Figure 23.24, 101b (triangle-wave PWM mode 2) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.22, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.22, GTIOA[4:0] = 00110b, in Figure 23.23, GTIOA[4:0] = 00011b, and in Figure 23.24, GTIOB[4:0] = 00011b.

Enable GTIOC pin output *1
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.22, CCRA = 01b, in Figure 23.23, CCRA = 1xb, and in Figure 23.24, CCRB = 1xb.

Set compare match value *1
Set the GTIOCA pin transition in GTCCRA and GTIOCB pin transition in GTCCRB.

Set buffer value
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTCCRD and GTCCRF, respectively.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step for Set compare-match value.

Figure 23.25 Example setting for GTCCRA and GTCCRB buffer operation with output compare

(2) When GTCCRA or GTCCRB functions as an input capture register

When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB and the stored GTCCRA and GTCCRB register values are transferred to the buffer registers. In input capture operation, the buffer transfer is not performed by the counter clear.

Figure 23.26 and Figure 23.27 show examples of GTCCRA and GTCCRB buffer operation and Figure 23.28 shows an example setting for GTCCRA and GTCCRB buffer operation.

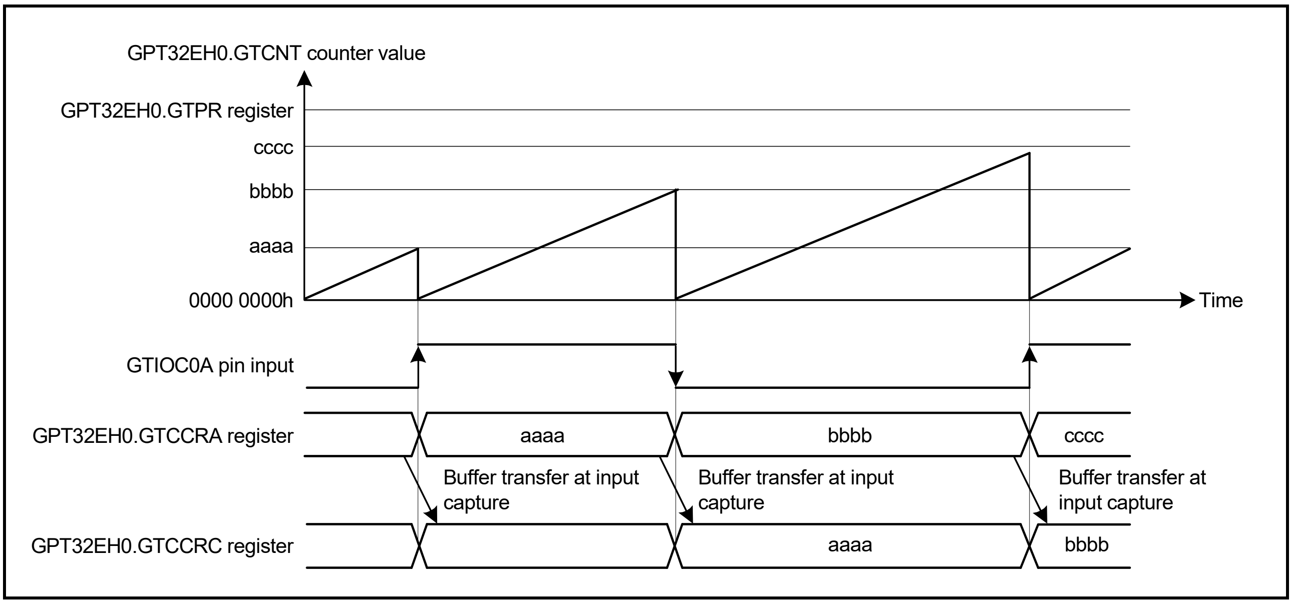


Figure 23.26 Example of GTCCRA and GTCCRB buffer operation with input capture at both edges of GTIOC0A input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0A input

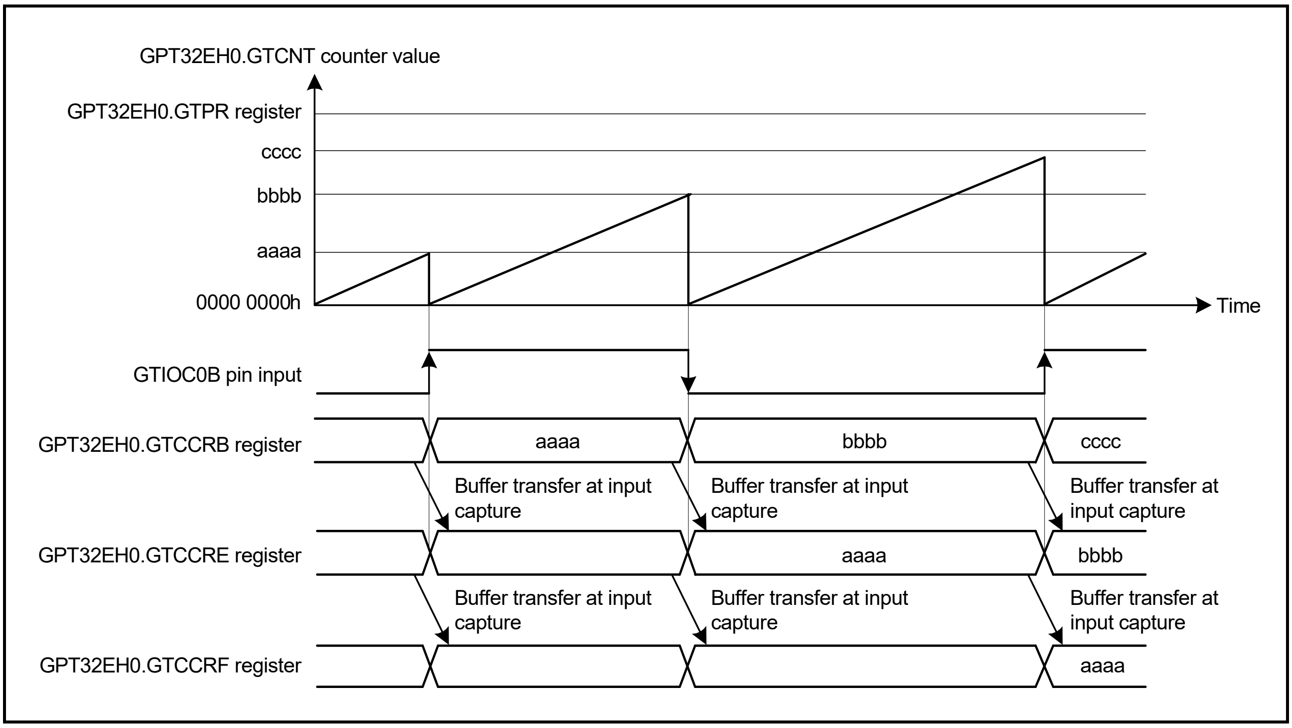
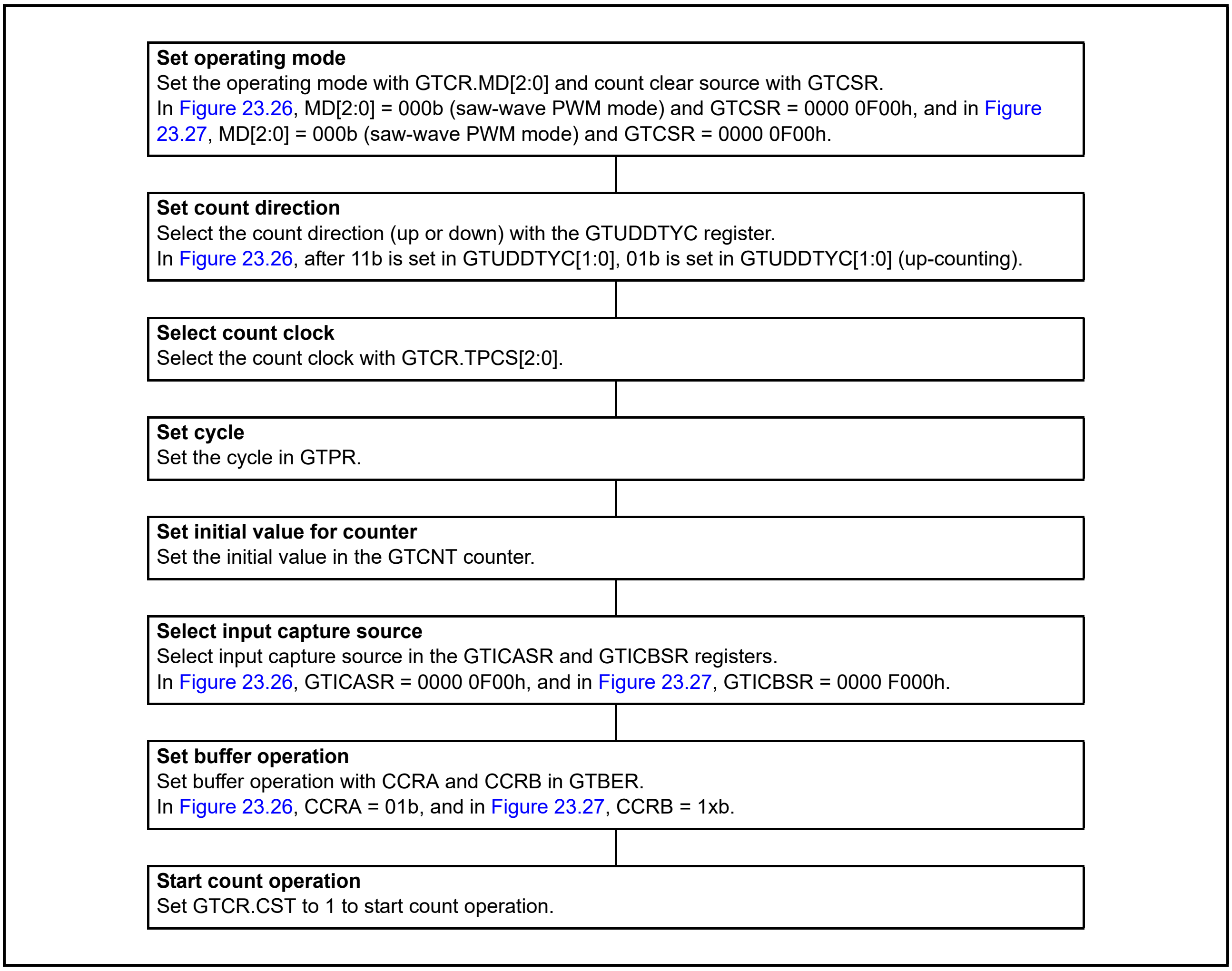


Figure 23.27 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOC0B input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOC0B input



Set operating mode
Set the operating mode with GTCR.MD[2:0] and count clear source with GTCSR.
In Figure 23.26, MD[2:0] = 000b (saw-wave PWM mode) and GTCSR = 0000 0F00h, and in Figure 23.27, MD[2:0] = 000b (saw-wave PWM mode) and GTCSR = 0000 0F00h.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.26, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Select input capture source
Select input capture source in the GTICASR and GTICBSR registers.
In Figure 23.26, GTICASR = 0000 0F00h, and in Figure 23.27, GTICBSR = 0000 F000h.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.26, CCRA = 01b, and in Figure 23.27, CCRB = 1xb.

Start count operation
Set GTCR.CST to 1 to start count operation.

Figure 23.28 Example setting for GTCCRA and GTCCRB buffer operation with input capture

23.3.2.3 Buffer operation for GTADTRA and GTADTRB

GTADTBRA can function as the GTADTRA buffer register and GTADTDBRA can function as the GTADTBRA buffer register (double-buffer register for GTADTRA). Similarly, GTADTBRB can function as the GTADTRB buffer register and GTADTDBRB can function as the GTADTBRB buffer register (double-buffer register for GTADTRB).

To set GTADTRA or GTADTRB to function as a double buffer, set GTBER.ADTDA or GTBER.ADTDB to 1. For single buffer operation, set GTBER.ADTDA or GTBER.ADTDB to 0. To set GTADTRA or GTADTRB to not function as a buffer, set GTBER.ADTTA[1:0] or GTBER.ADTTB[1:0] to 00b.

The buffer transfer timing can be set with the GTBER.ADTTA[1:0] bits. For saw waves, overflows (during up-counting) or underflows (during down-counting) can be selected. For triangle waves, crests are selected when GTBER.ADTTA[1:0] = 01b, troughs are selected when GTBER.ADTTA[1:0] = 10b, and both crests and troughs are selected when GTBER.ADTTA[1:0] = 11b.

Figure 23.29 to Figure 23.31 show examples of GTADTRA and GTADTRB buffer operation and Figure 23.32 shows an example setting for GTDTRA and GTADTRB buffer operation.

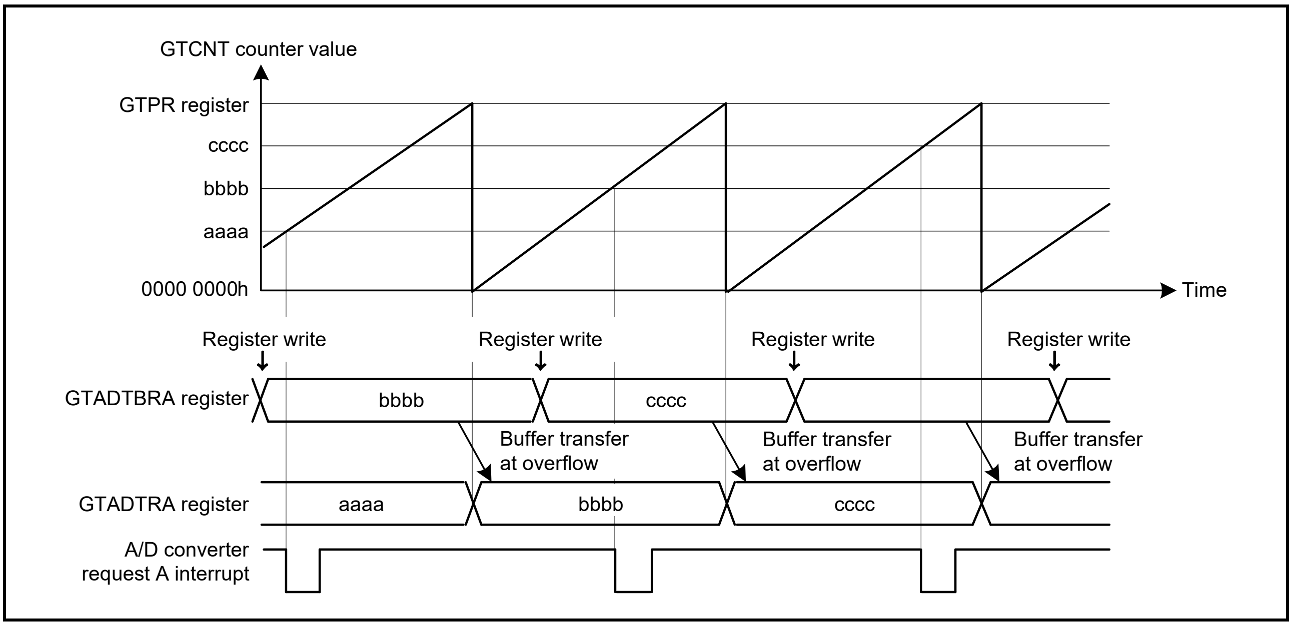


Figure 23.29 Example of GTADTRA and GTADTRB buffer operation with saw waves in up-counting and A/D converter start request interrupt generated by up-counting

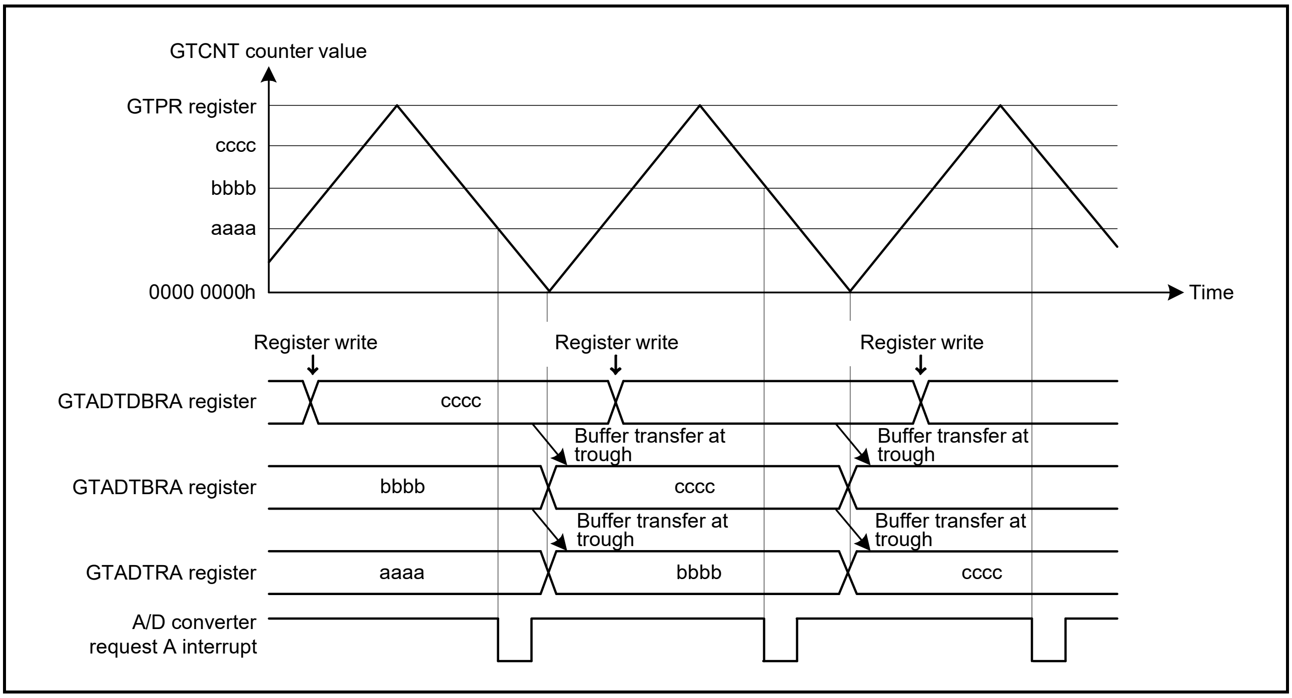


Figure 23.30 Example of GTADTRA and GTADTRB double buffer operation with triangle waves, buffer transfer at troughs, and A/D converter start request interrupt generated by down-counting

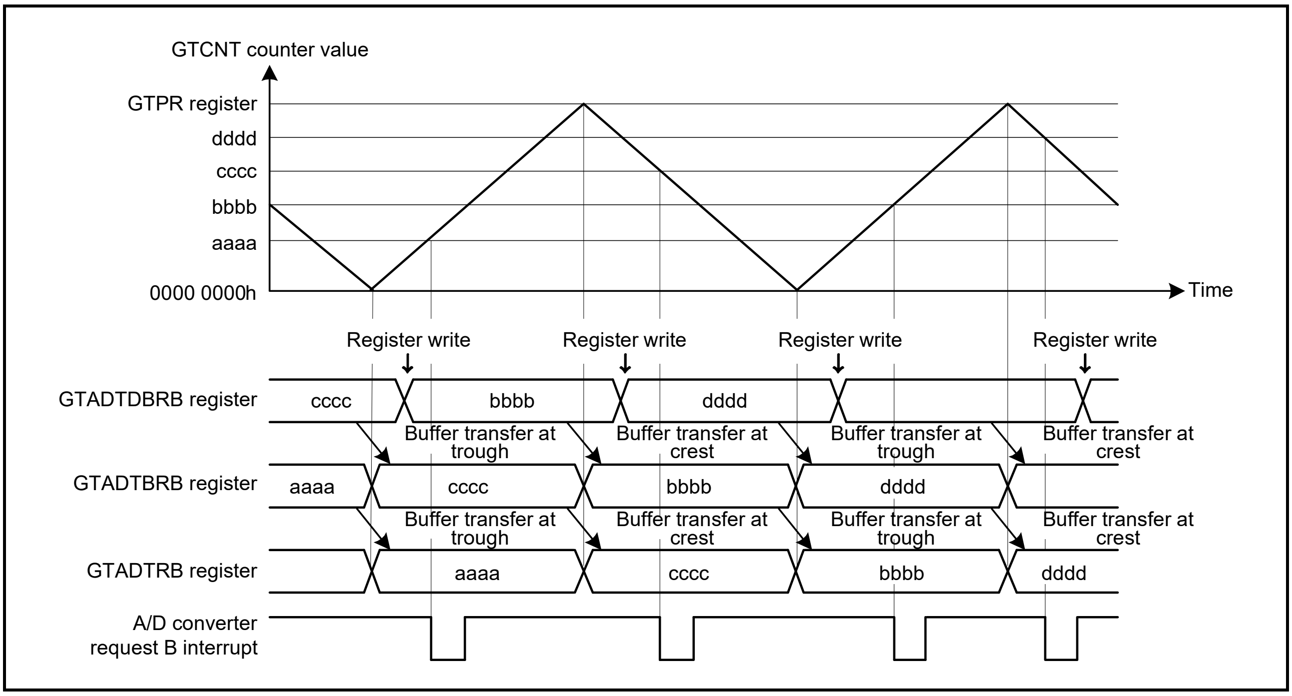
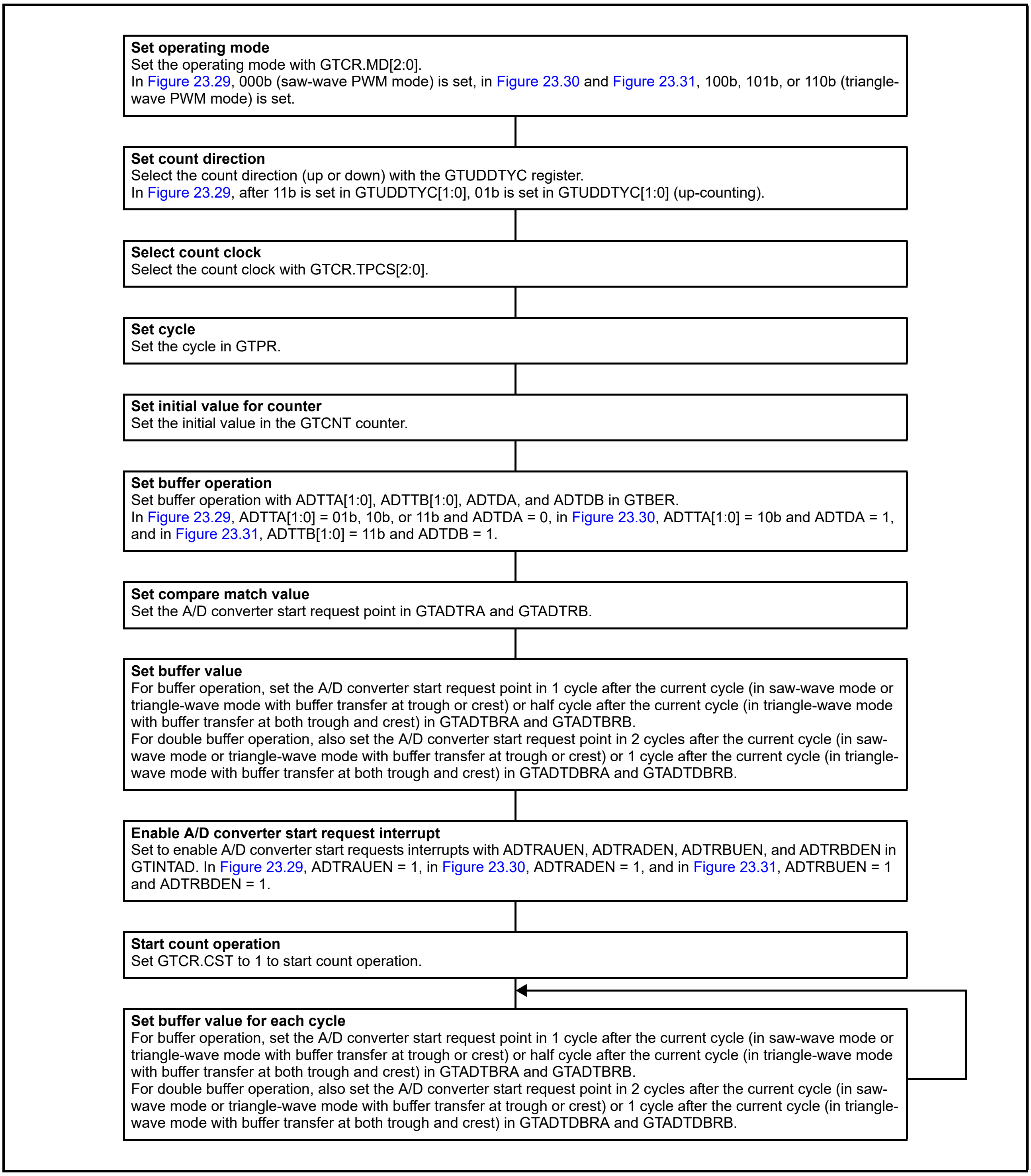


Figure 23.31 Example of GTADTRA and GTADTRB double buffer operation with triangle waves, buffer transfer at both troughs and crests, and A/D converter start request interrupt generated by both up- and down-counting



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.29, 000b (saw-wave PWM mode) is set, in Figure 23.30 and Figure 23.31, 100b, 101b, or 110b (triangle-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.29, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set buffer operation
Set buffer operation with ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB in GTBER.
In Figure 23.29, ADTTA[1:0] = 01b, 10b, or 11b and ADTDA = 0, in Figure 23.30, ADTTA[1:0] = 10b and ADTDA = 1, and in Figure 23.31, ADTTB[1:0] = 11b and ADTDB = 1.

Set compare match value
Set the A/D converter start request point in GTADTRA and GTADTRB.

Set buffer value
For buffer operation, set the A/D converter start request point in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTBRA and GTADTBRB.
For double buffer operation, also set the A/D converter start request point in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTDBRA and GTADTDBRB.

Enable A/D converter start request interrupt
Set to enable A/D converter start requests interrupts with ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN in GTINTAD. In Figure 23.29, ADTRAUEN = 1, in Figure 23.30, ADTRADEN = 1, and in Figure 23.31, ADTRBUEN = 1 and ADTRBDEN = 1.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the A/D converter start request point in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTBRA and GTADTBRB.
For double buffer operation, also set the A/D converter start request point in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTDBRA and GTADTDBRB.

Figure 23.32 Example setting for GTADTRA and GTADTRB buffer operation

23.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCA or GTIOCB pin by a compare match between the GTCNT counter and GTCCRA or GTCCRB. By setting GTDTCR, GTDVU, and GTDVD, the compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

23.3.3.1 Saw-wave PWM mode

In saw-wave PWM mode, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR. A PWM waveform is output to the GTIOCA or GTIOCB pin when a GTCCRA or GTCCRB compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

Figure 23.33 shows an example of saw-wave PWM mode operation, and Figure 23.34 shows an example setting for saw-wave PWM mode.

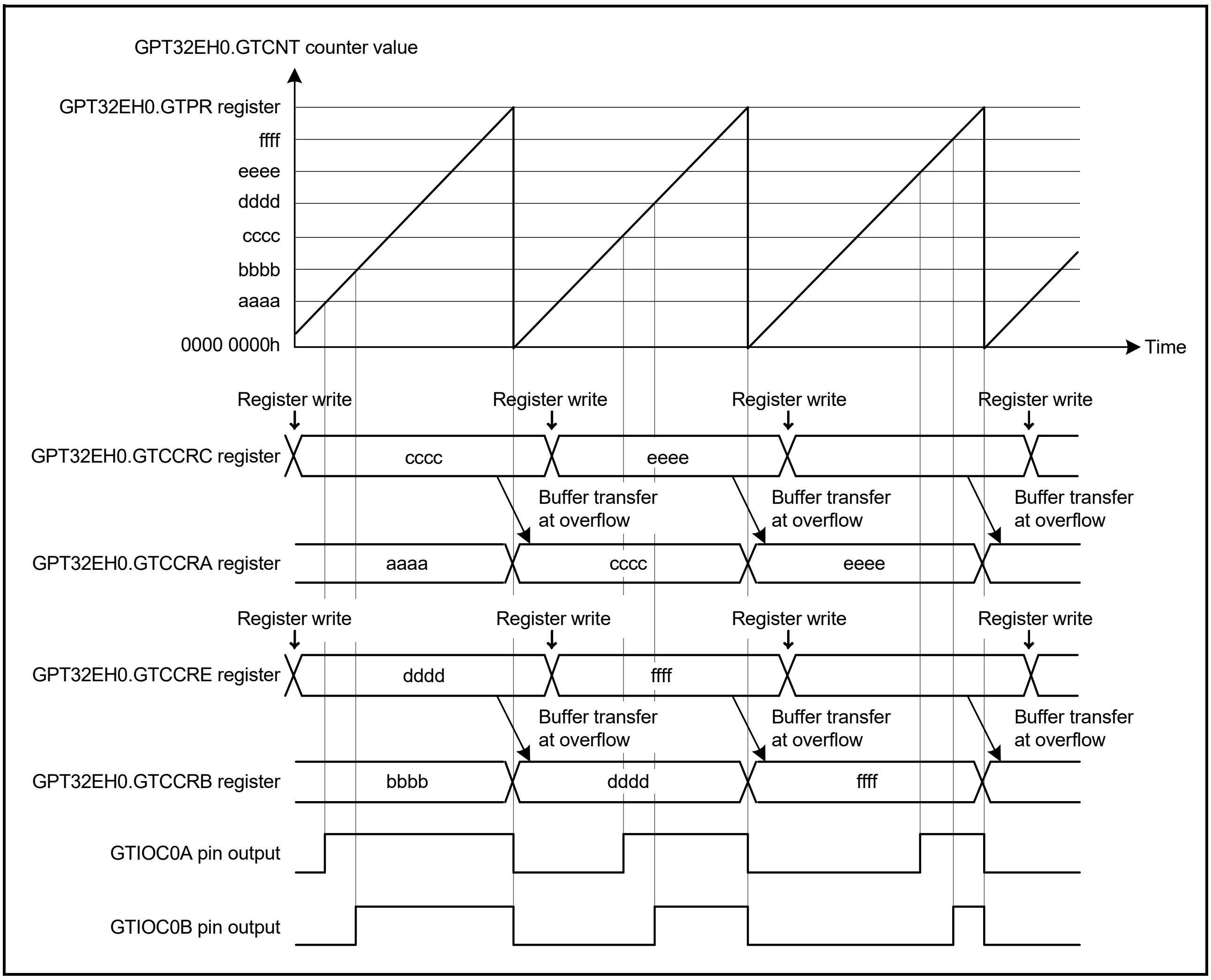
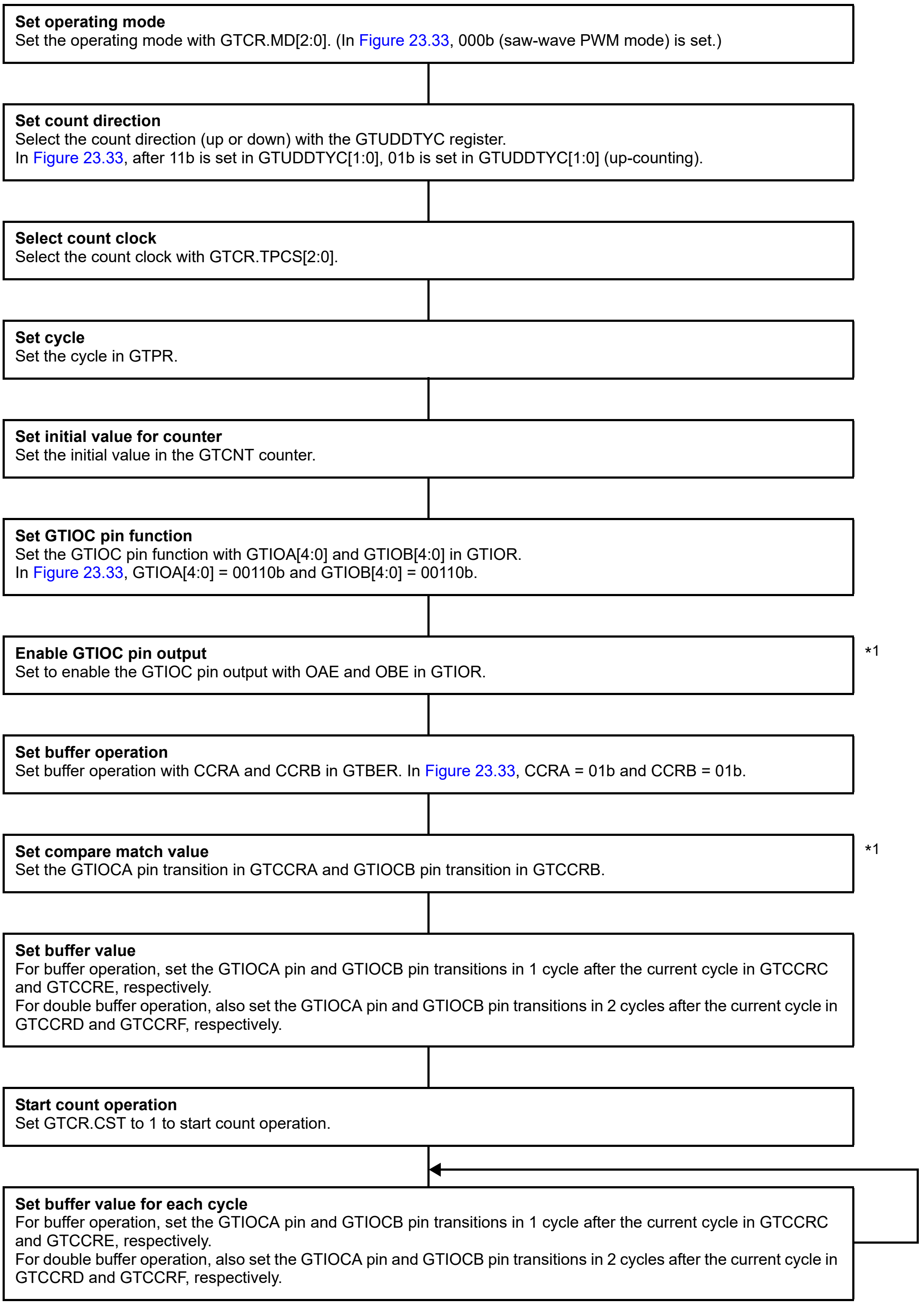


Figure 23.33 **Example of saw-wave PWM mode operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, and low output at cycle end**



Set operating mode
Set the operating mode with GTCR.MD[2:0]. (In Figure 23.33, 000b (saw-wave PWM mode) is set.)

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.33, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.33, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.

Enable GTIOC pin output *1
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER. In Figure 23.33, CCRA = 01b and CCRB = 01b.

Set compare match value *1
Set the GTIOCA pin transition in GTCCRA and GTIOCB pin transition in GTCCRB.

Set buffer value
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step for Set compare-match value.

Figure 23.34 Example setting for saw-wave PWM mode

23.3.3.2 Saw-wave one-shot pulse mode

The saw-wave one-shot pulse mode is a mode in which the cycle is set in GTPR. The GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCA or GTIOCB pin at a compare match of GTCCRA or GTCCRB with buffer operation fixed.

Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the cycle end
- GTCCRE to GTCCRB at the cycle end
- GTCCRD to temporary register A at the cycle end
- GTCCRF to temporary register B at the cycle end
- Temporary register A to GTCCRA at a GTCCRA compare match
- Temporary register B to GTCCRB at a GTCCRB compare match.

The pin output value can be selected from low output, high output, or toggle output separately for a compare match and the cycle end according to the GTIOR setting. When the GTBER.CCRSWT bit is set to 1 while the count operation is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.35 shows an example of saw-wave one-shot pulse mode operation, and Figure 23.36 shows an example setting for saw-wave one-shot pulse mode.

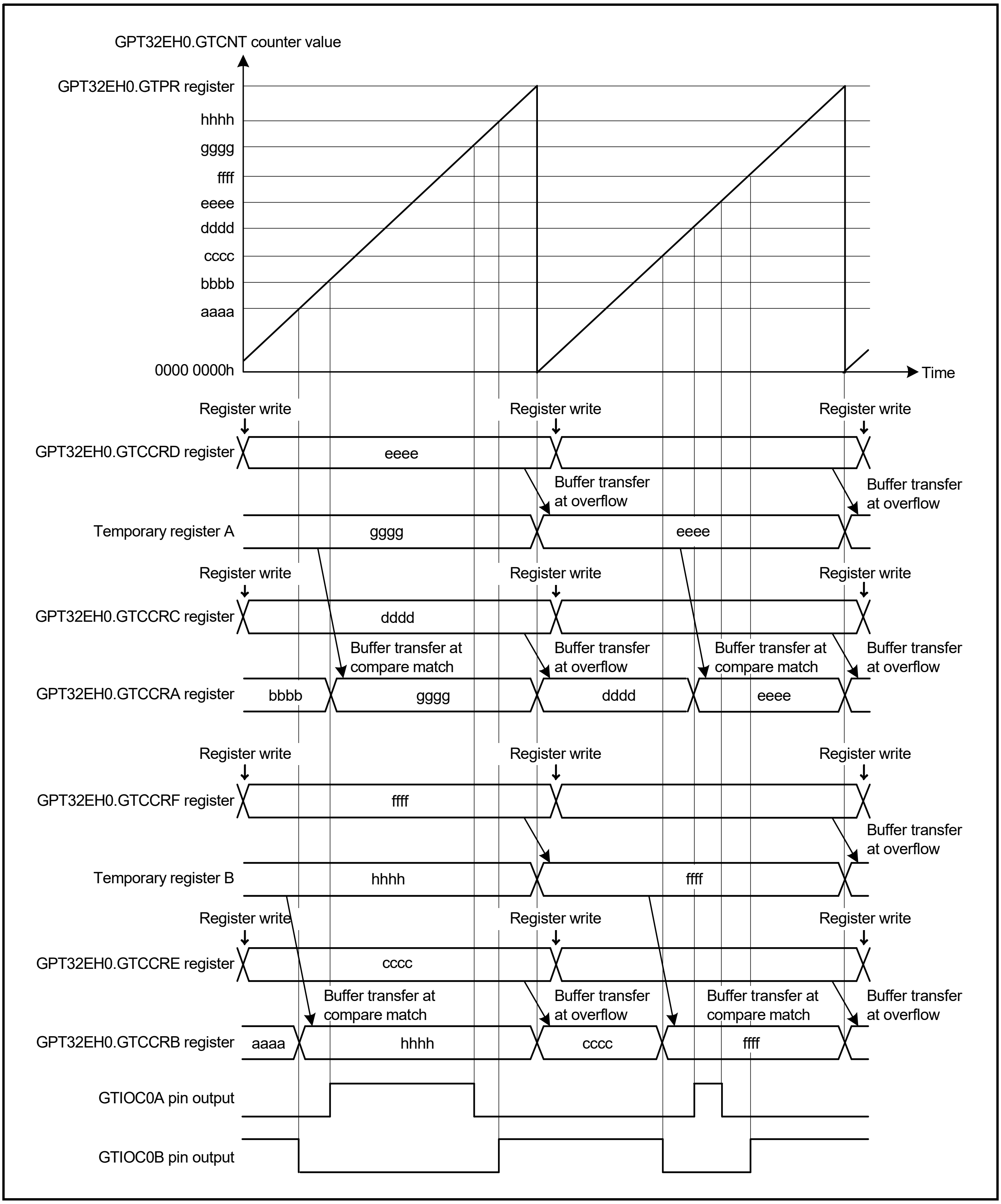
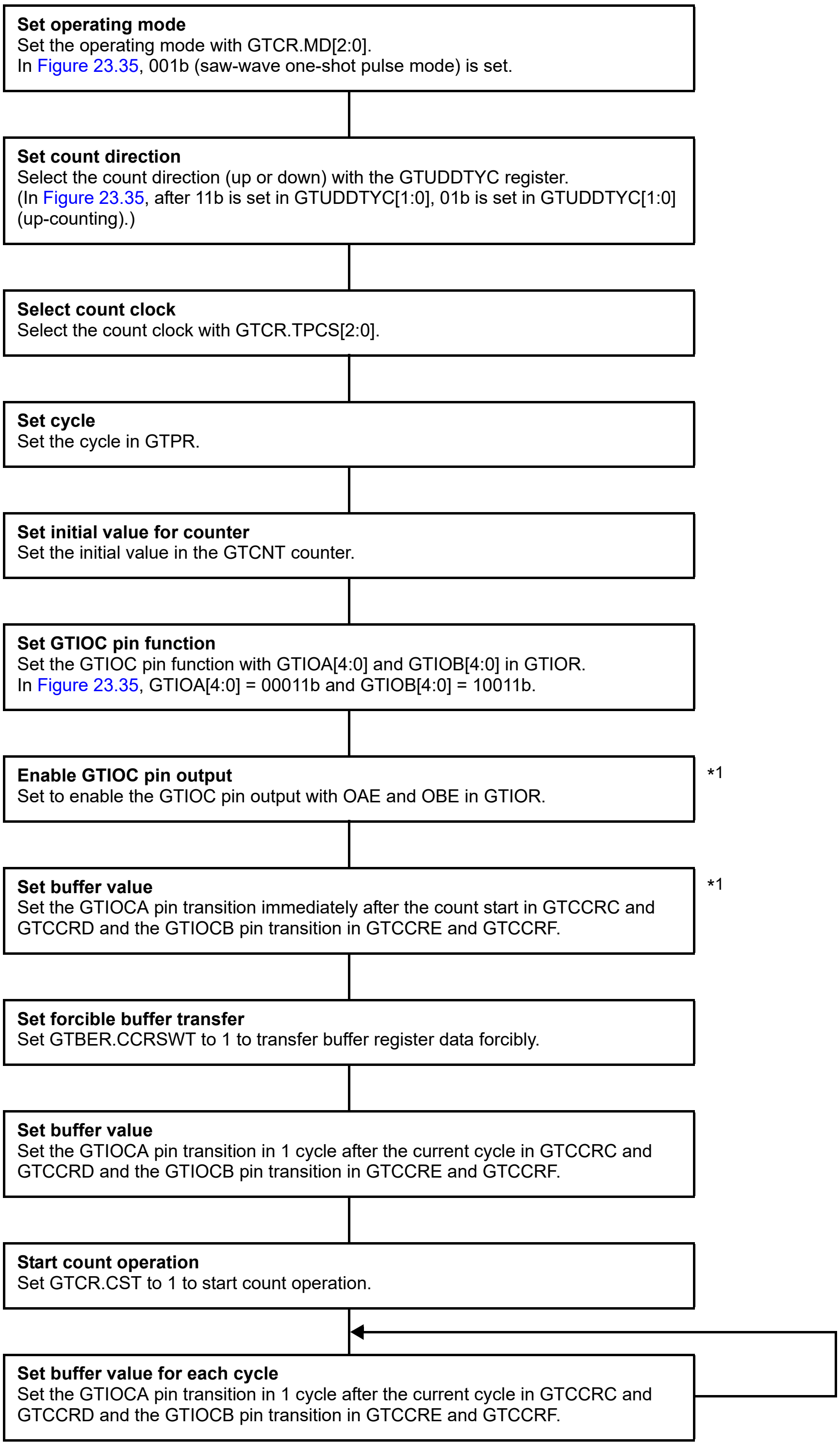


Figure 23.35 Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.35, 001b (saw-wave one-shot pulse mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
(In Figure 23.35, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).)

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.35, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output *1
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer value *1
Set the GTIOCA pin transition immediately after the count start in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Set forcible buffer transfer
Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly.

Set buffer value
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step for Set buffer value.

Figure 23.36 Example setting for saw-wave one-shot pulse mode

23.3.3.3 Triangle-wave PWM mode 1 (32-bit transfer at trough)

The triangle-wave PWM mode 1 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCA or GTIOCB pin when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end based on the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.37 shows an example of a triangle-wave PWM mode 1 operation, and Figure 23.38 shows an example setting for a triangle-wave PWM mode 1.

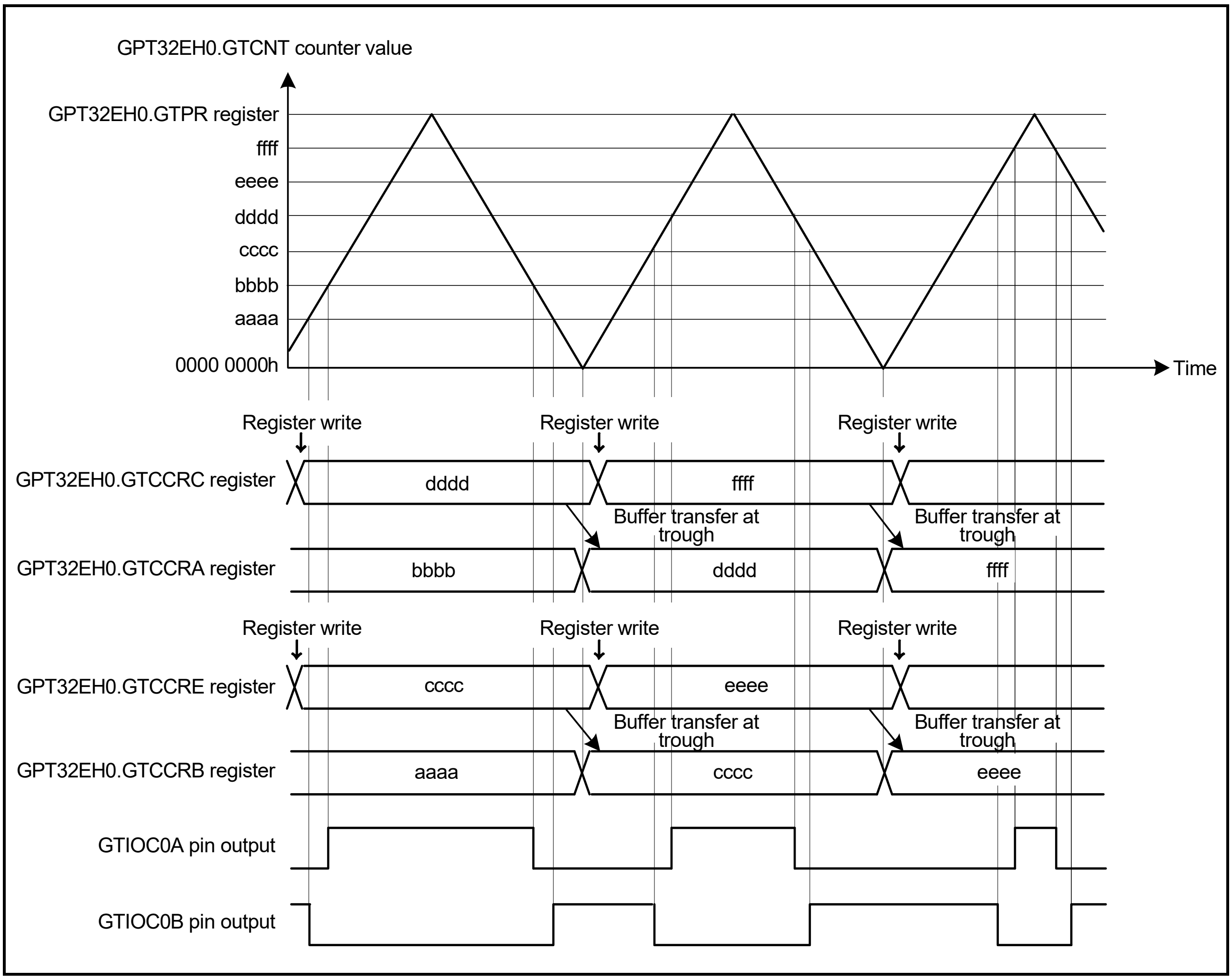
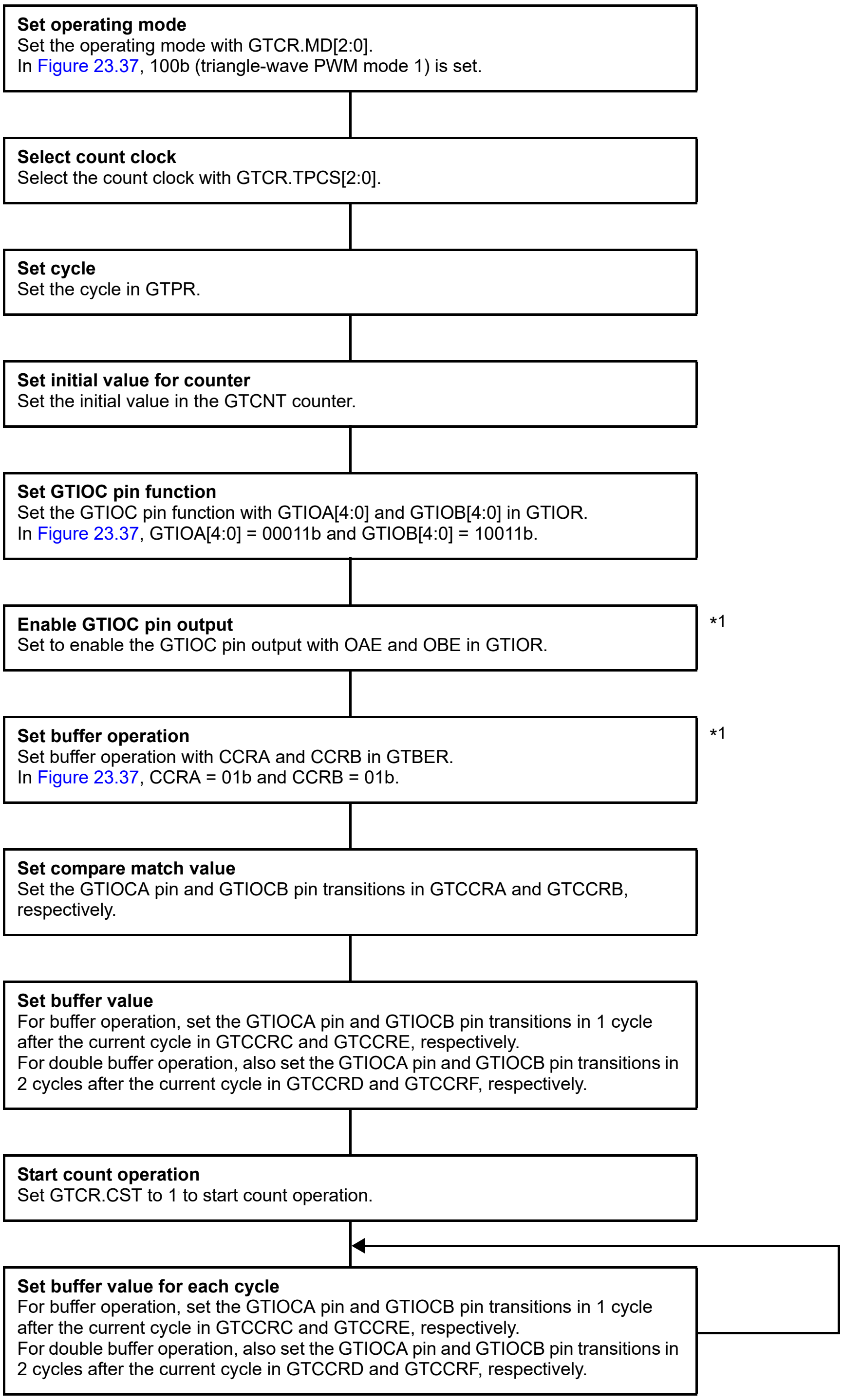


Figure 23.37 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB register compare match, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.37, 100b (triangle-wave PWM mode 1) is set.

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.37, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.
*1

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.37, CCRA = 01b and CCRB = 01b.
*1

Set compare match value
Set the GTIOCA pin and GTIOCB pin transitions in GTCCRA and GTCCRB, respectively.

Set buffer value
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 2 cycles after the current cycle in GTCCRD and GTCCRF, respectively.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step for Set buffer value.

Figure 23.38 Example setting for triangle-wave PWM mode 1

23.3.3.4 Triangle-wave PWM mode 2 (32-bit transfer at crest and trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCA or GTIOCB pin when a GTCCRA or GTCCRB compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end based on the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.39 shows an example of triangle-wave PWM mode 2 operation, and Figure 23.40 shows an example setting for triangle-wave PWM mode 2.

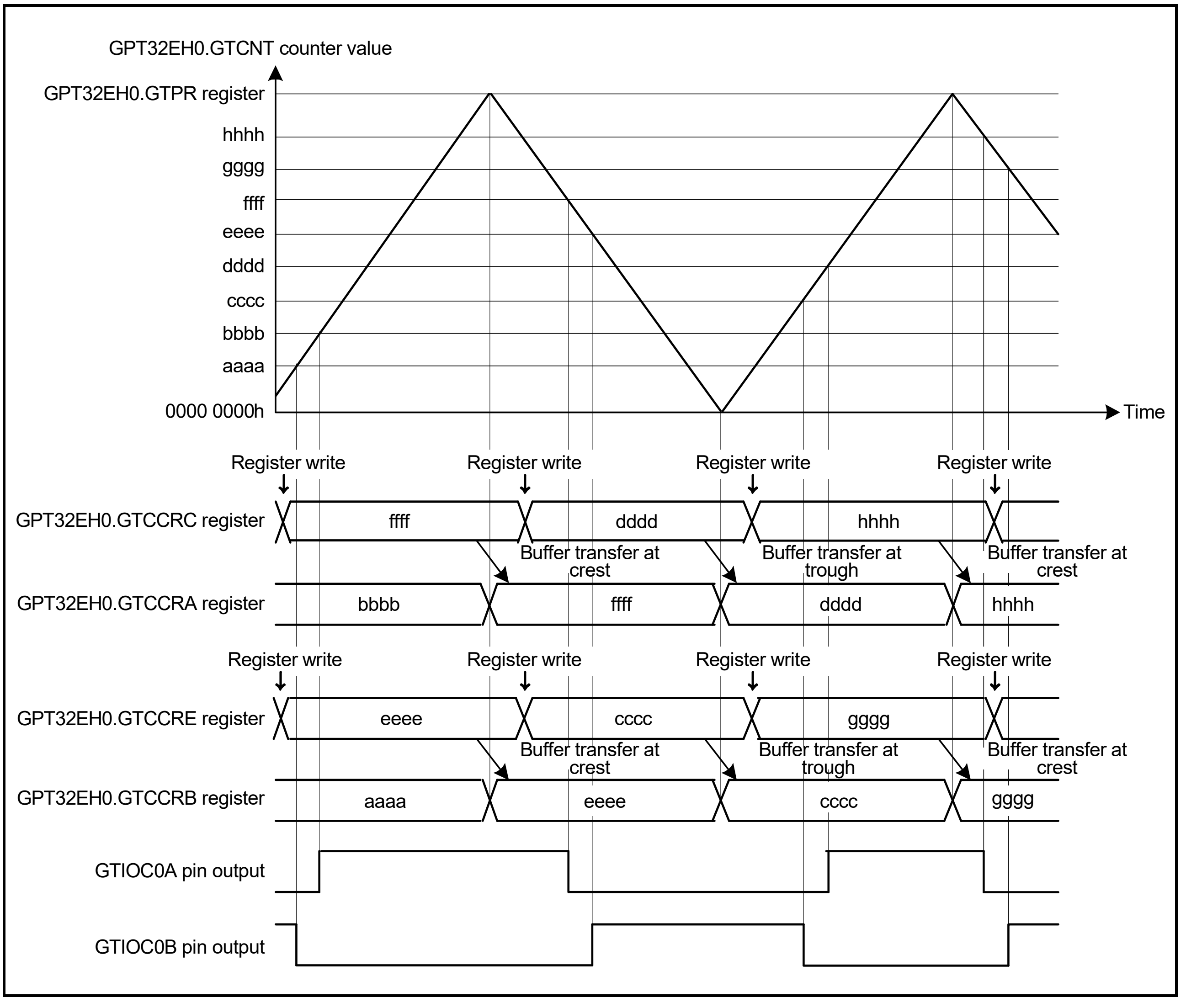
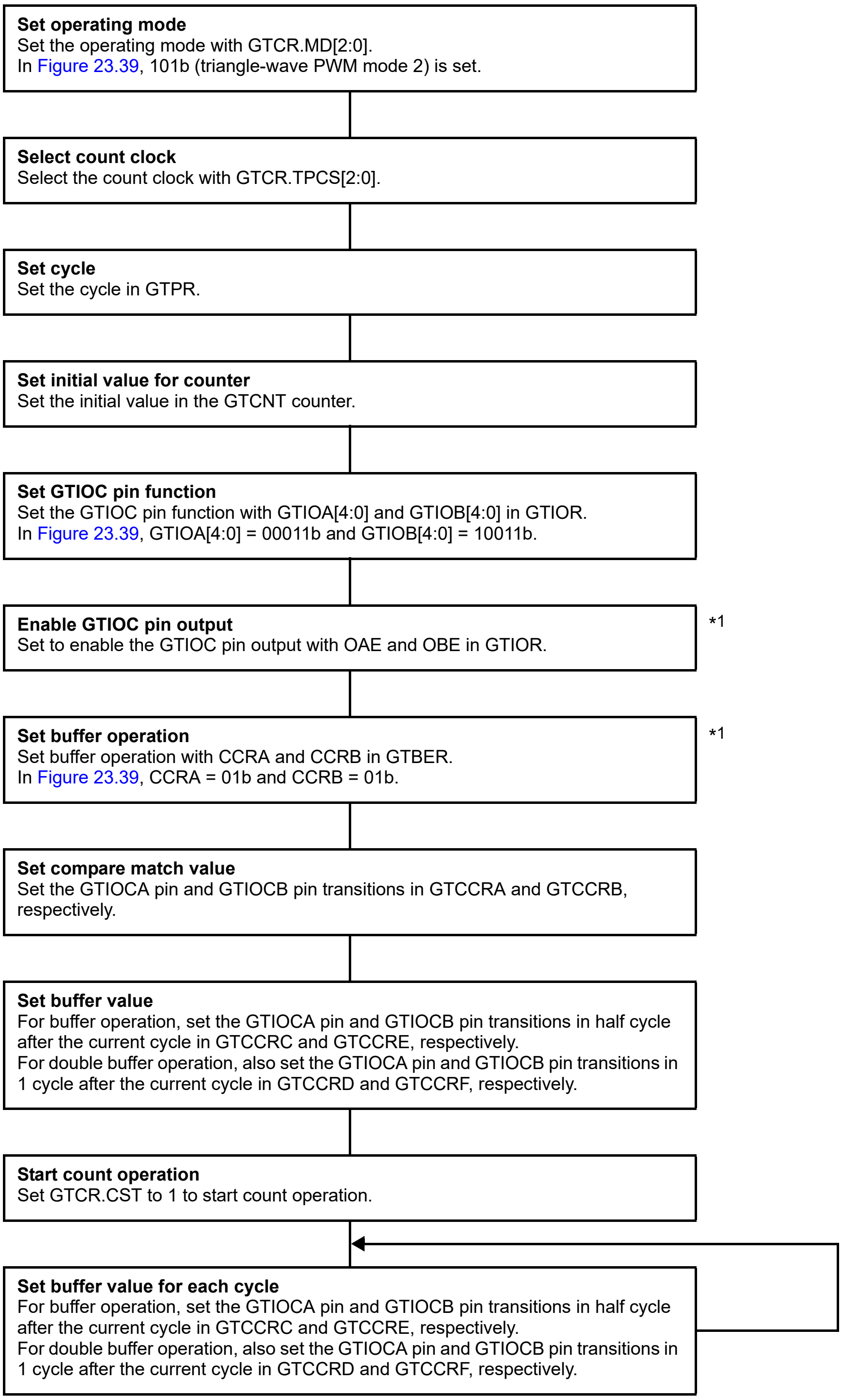


Figure 23.39 Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.39, 101b (triangle-wave PWM mode 2) is set.

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.39, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR. *1

Set buffer operation
Set buffer operation with CCRA and CCRB in GTBER.
In Figure 23.39, CCRA = 01b and CCRB = 01b. *1

Set compare match value
Set the GTIOCA pin and GTIOCB pin transitions in GTCCRA and GTCCRB, respectively.

Set buffer value
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in half cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF, respectively.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the GTIOCA pin and GTIOCB pin transitions in half cycle after the current cycle in GTCCRC and GTCCRE, respectively.
For double buffer operation, also set the GTIOCA pin and GTIOCB pin transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF, respectively.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step “Set buffer value.

Figure 23.40 Example setting for triangle-wave PWM mode 2

23.3.3.5 Triangle-wave PWM mode 3 (64-bit transfer at trough)

The triangle-wave PWM mode 3 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCA or GTIOCB pin at a compare match of GTCCRA or GTCCRB with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the trough
- GTCCRE to GTCCRB at the trough
- GTCCRD to temporary register A at the trough
- GTCCRF to temporary register B at the trough
- Temporary register A to GTCCRA at the crest
- Temporary register B to GTCCRB at the crest.

The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end based on the GTIOR setting. By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.41 shows an example of triangle-wave PWM mode 3 operation, and Figure 23.42 shows an example setting for triangle-wave PWM mode 3.

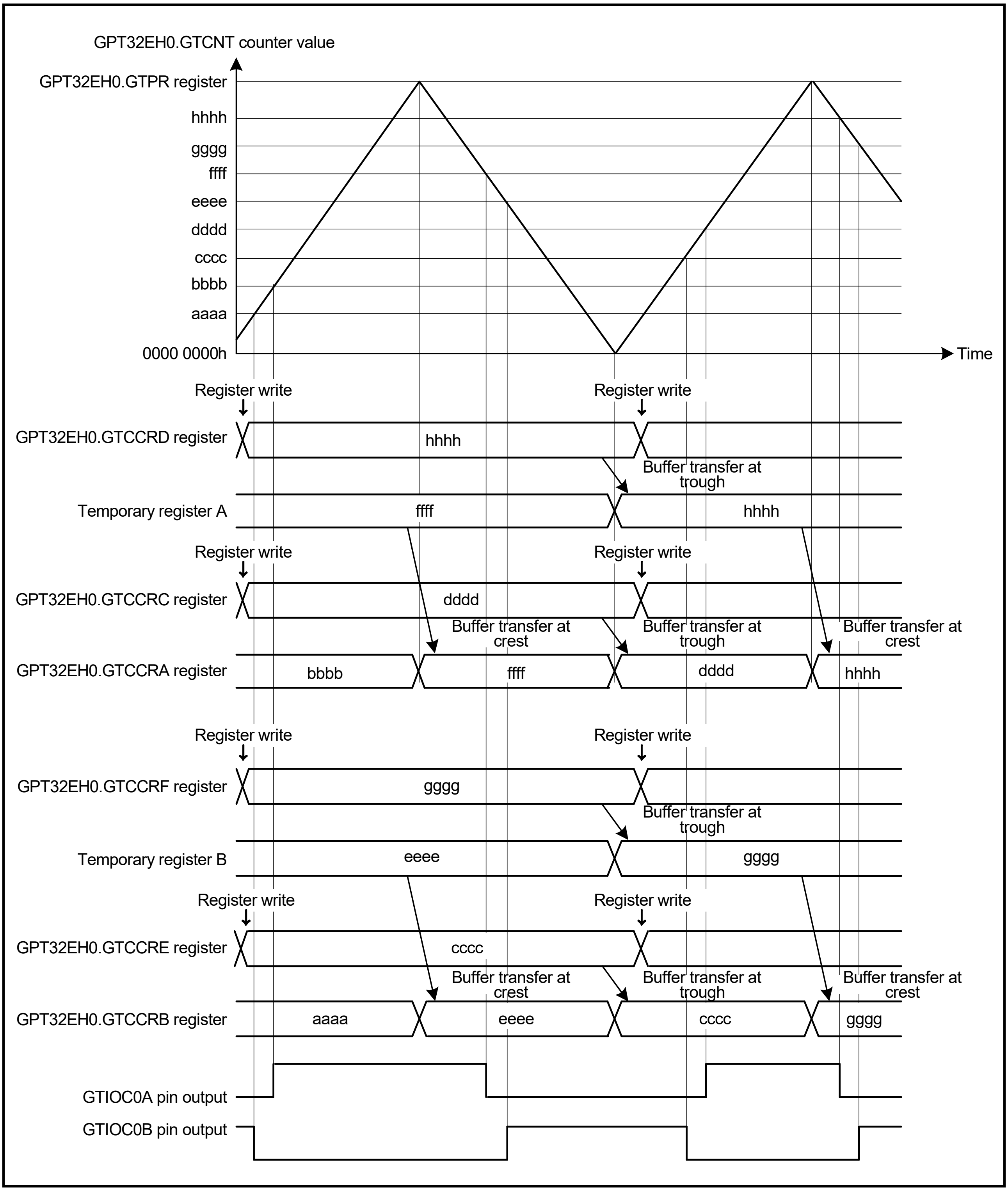
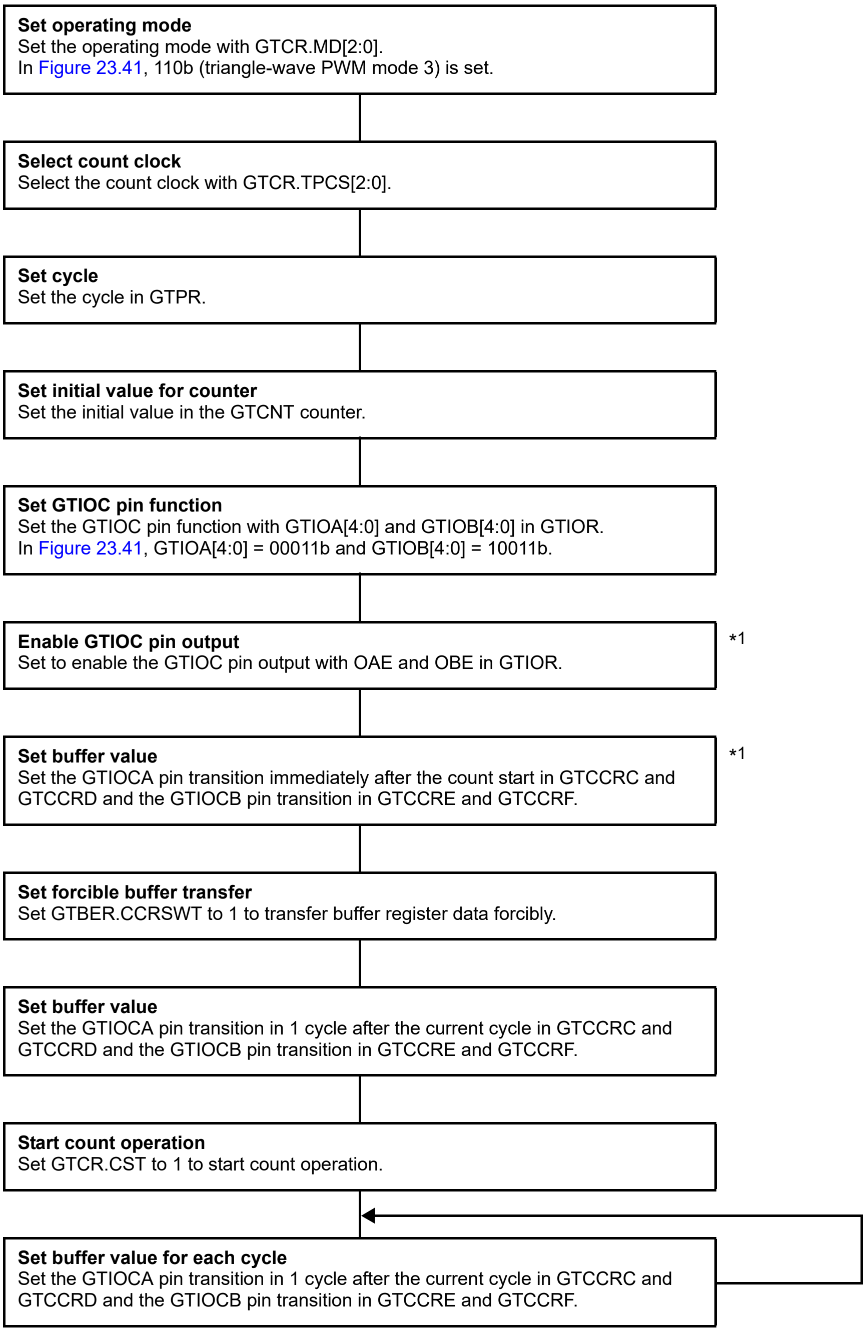


Figure 23.41 Example of triangle-wave PWM mode 3 operation with low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.41, 110b (triangle-wave PWM mode 3) is set.

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.41, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output *1
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer value *1
Set the GTIOCA pin transition immediately after the count start in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Set forcible buffer transfer
Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly.

Set buffer value
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD and the GTIOCB pin transition in GTCCRE and GTCCRF.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step for Set buffer value.

Figure 23.42 Example setting for triangle-wave PWM mode 3

23.3.4 Automatic Dead Time Setting Function

By setting GTDTCR, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (GTCCRA value) and specified dead time values (GTDVU and GTDVD values) can automatically be set to GTCCRB. The automatic dead time setting function can be used in saw-wave one-shot pulse mode and all the triangle PWM modes.

Dead time can be separately set for the first half and second half of a waveform. Dead time for the transition in the first half of a negative waveform is set in GTDVU and that in the second half is set in GTDVD. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.

GTDBU can be used as a buffer register of GTDVU, and GTDBD can be used as a buffer register of GTDVD. Buffer transfer is performed at a GTCNT overflow (during up-counting), an underflow (during down-counting), or at a GTCNT counter clear for saw waves and at the trough for triangle waves.

The compare match value set by automatic dead time setting function can be confirmed by reading from GTCCRB. Writing to GTCCRB is prohibited when the automatic dead time setting function is used.

Dead time setting beyond the cycle is prohibited. When a dead time error occurs, the compare match values for positive and negative waveforms are adjusted to generate the waveforms with the dead time as shown in Table 23.6. The adjusted value for the negative waveform is set for GTCCRB automatically. The adjusted value for the positive waveform is used as internal signal and not set for GTCCRA.

In saw-wave one-shot pulse mode, when the adjusted value is beyond the cycle or the adjusted waveform toggle points are in disorder, the complementarity of the waveforms is not guaranteed.

In triangle-wave mode, when the dead time is beyond the cycle by setting the value GTCCR = 0 or GTCCRA ≥ GTPR for GTCCRA, the output protection function keeps the level of output. For details, see section 23.8.4, Output Protection Function for GTIOC Pin Output. When the GTCCRA is GTCCRA ≥ GTPR + GTDVm, GTPR-1 is set for GTCCRB as the upper limit value. The automatic dead time value setting to GTCCRB is performed at the next clock cycle count when registers used for calculating the automatic dead time value are updated.

The way to rewrite GTDVm differs by GPT channel number.

Table 23.6 Compare match value after adjusting for dead time error

PWM output operating mode	Count direction	First half/ Second half	Condition of dead time error	Compare match value after adjusting	
				Positive waveform	Negative waveform
Saw-wave one-shot pulse mode	Up	First half	GTCCRA - GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR	GTPR - GTDVD	GTPR
	Down	First half	GTCCRA + GTDVU > GTPR	GTPR - GTDVU	GTPR
		Second half	GTCCRA - GTDVD < 0	GTDVD	0
Triangle-wave PWM mode 1/2/3	Up	First half	GTCCRA - GTDVU ≤ 0	GTDVU + 1	1
	Down	Second half	GTCCRA - GTDVD < 0	GTDVD	0

GPT32EH0 to GPT32EH3 and GPT32E4 to GPT32E7

When GTDVm buffer operation is enabled, GTDBm can be written at anytime. GTDBm is transferred to GTDVm at the cycle end.

When GTDVm buffer operation is disabled, stop the GPT using the CST bit in the GTCR register before changing GTDVm to a new value.

GPT328 to GPT3213

While GPT is running, changing the GTDVU values is prohibited. To change GTDVU to a new value, first stop the GPT using the CST bit in the GTCR register.

Figure 23.43 to Figure 23.46 show examples of automatic dead time setting function operation for GPT32EH and GPT32E. Figure 23.47 and Figure 23.48 show the setting examples.

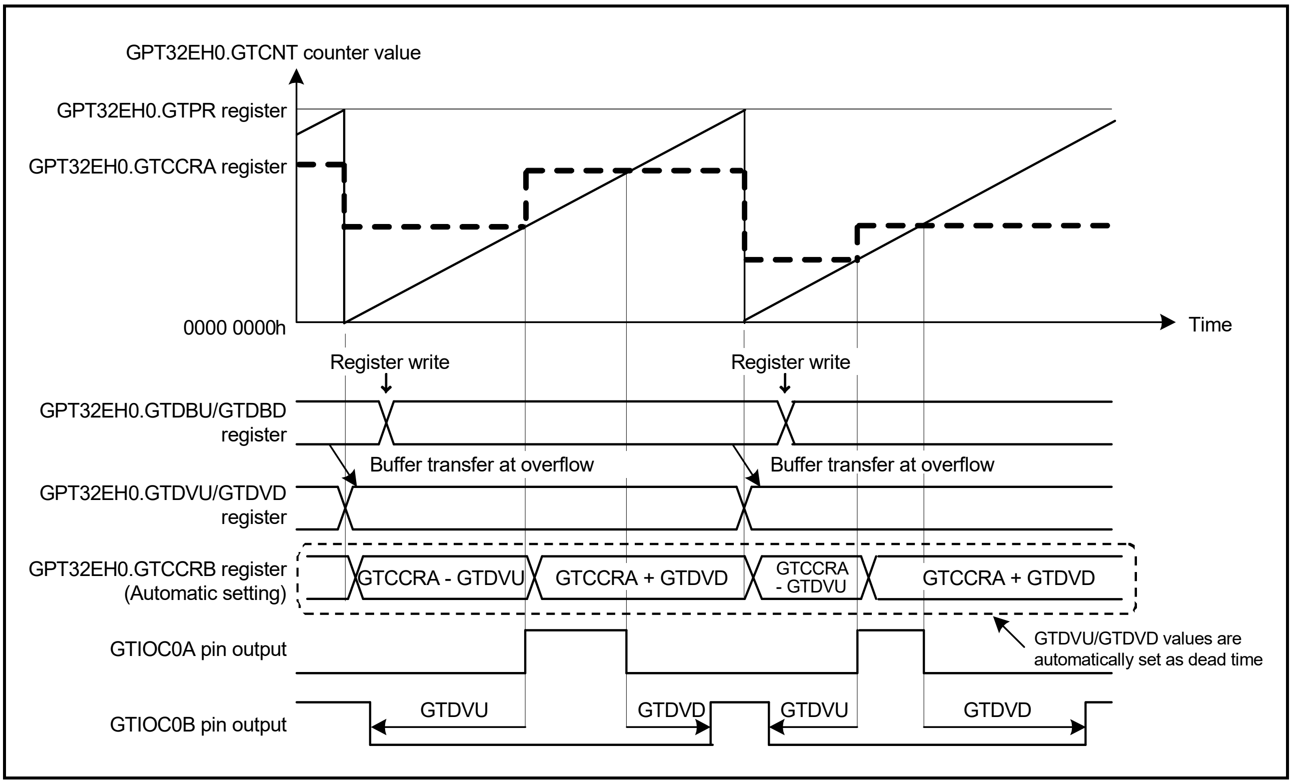


Figure 23.43 Example of automatic dead time setting function operation with saw-wave one-shot pulse mode, up-counting, GTDVU and GTDVD set to buffer operation, and active-high

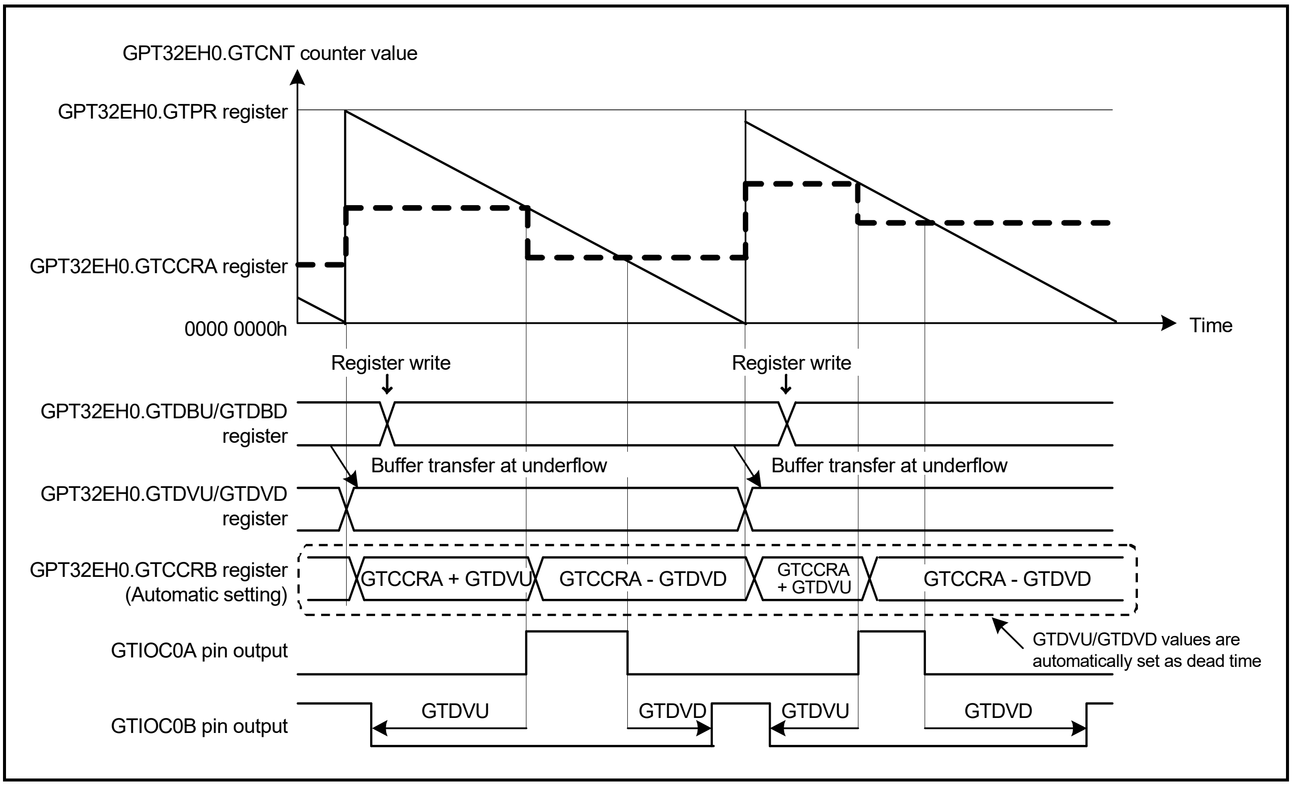


Figure 23.44 Example of automatic dead time setting function operation with saw-wave one-shot pulse mode, down-counting, GTDVU and GTDVD set to buffer operation, and active-high

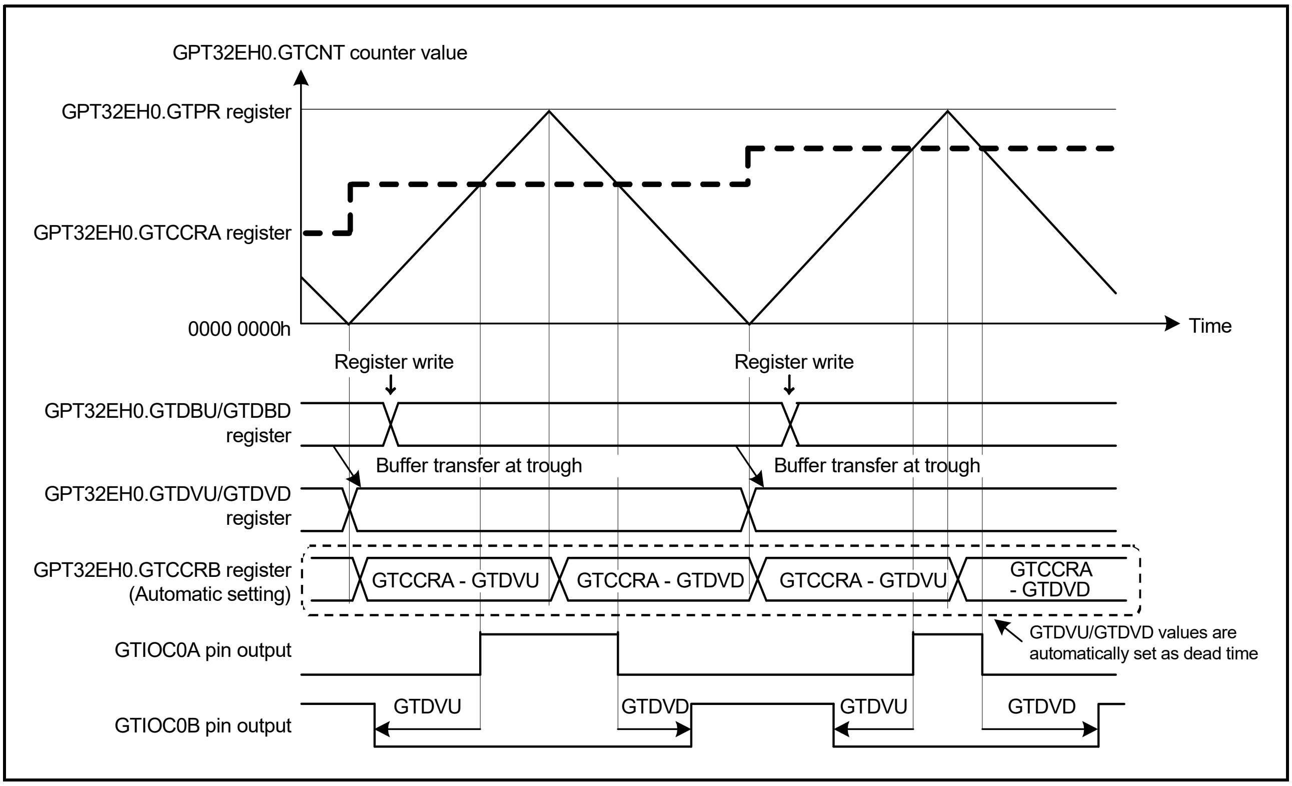


Figure 23.45 Example of automatic compare-match value setting function with dead time with triangle-wave PWM mode 1, GTDVU and GTDVD set to buffer operation, active-high

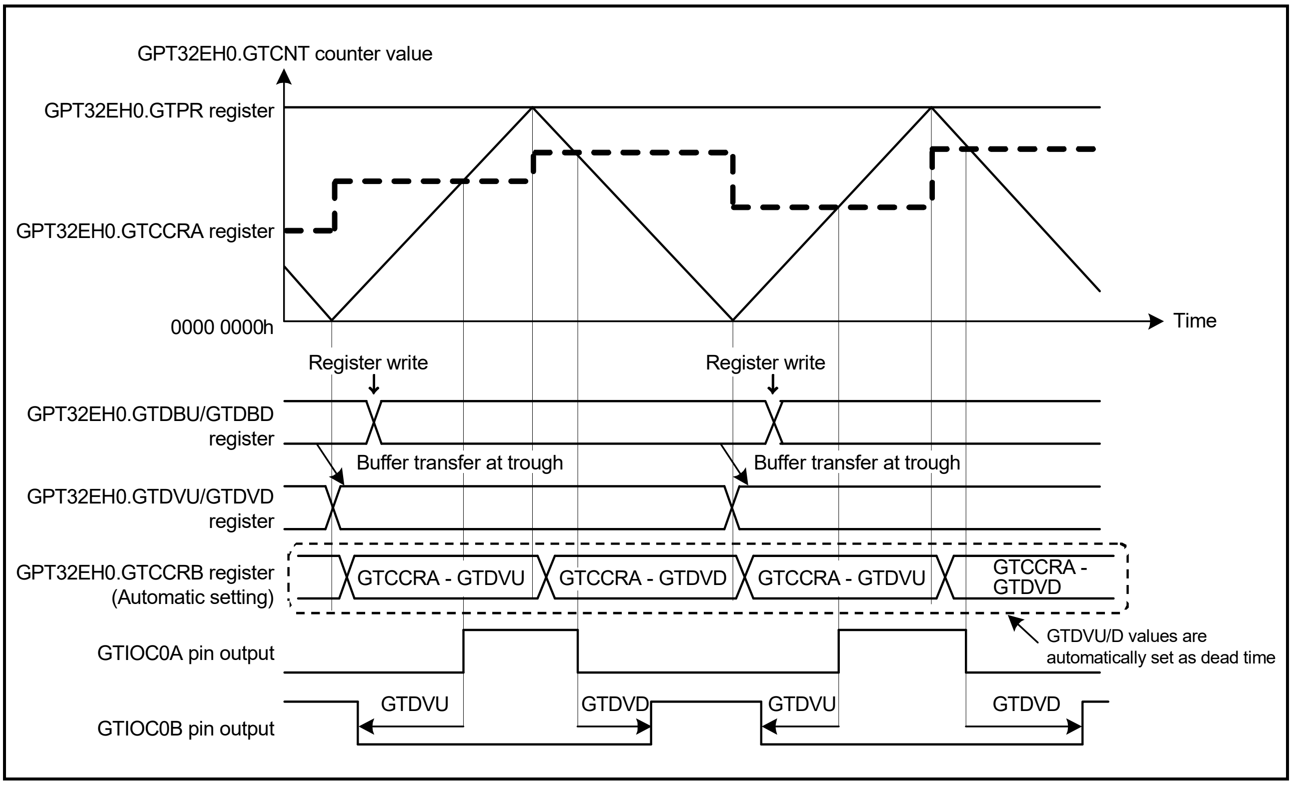
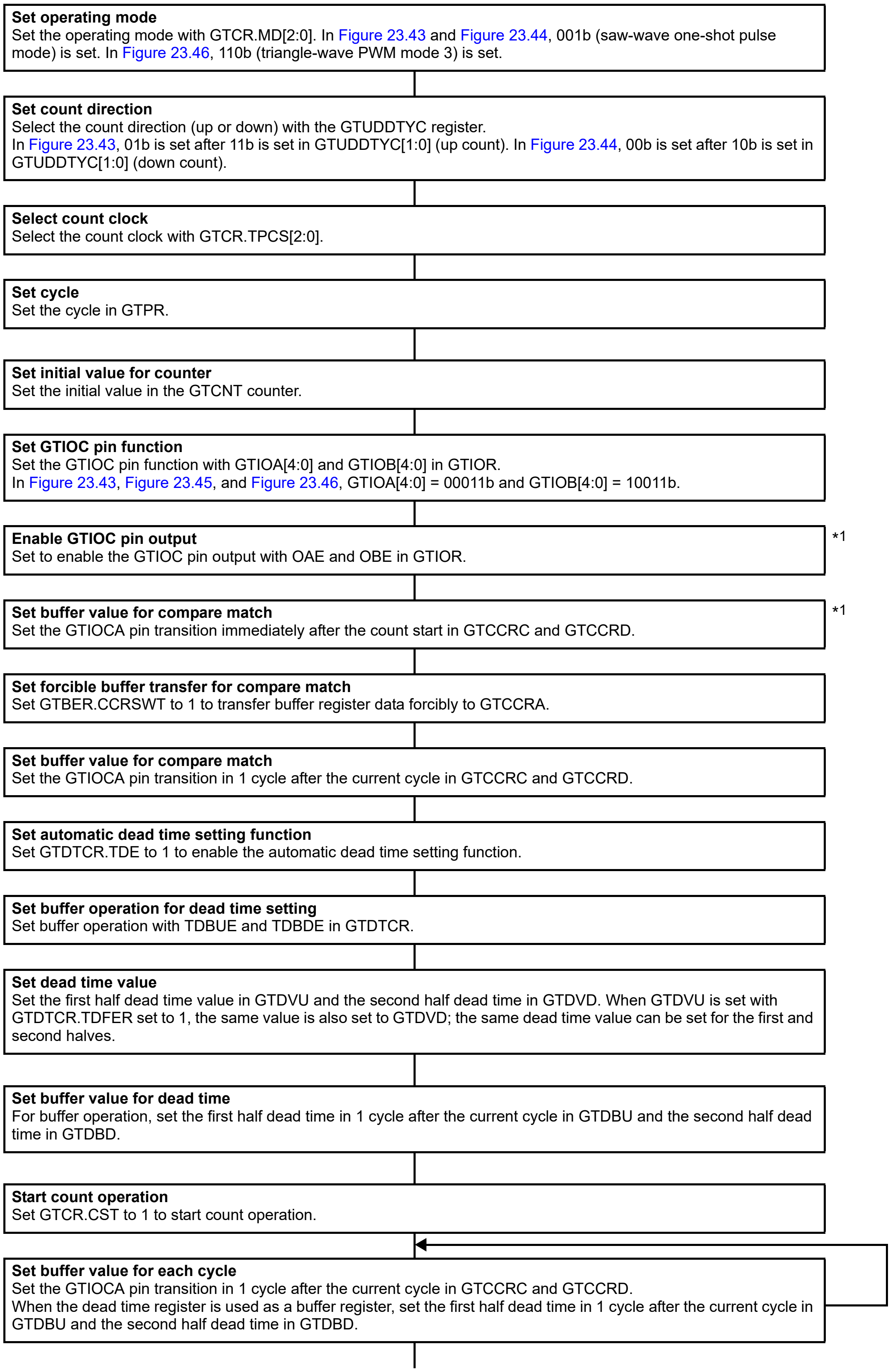


Figure 23.46 Example of automatic compare-match value setting function with dead time, with triangle-wave PWM mode 2 or 3, GTDVU and GTDVD set to buffer operation, and active-high



Set operating mode
Set the operating mode with GTCR.MD[2:0]. In Figure 23.43 and Figure 23.44, 001b (saw-wave one-shot pulse mode) is set. In Figure 23.46, 110b (triangle-wave PWM mode 3) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.43, 01b is set after 11b is set in GTUDDTYC[1:0] (up count). In Figure 23.44, 00b is set after 10b is set in GTUDDTYC[1:0] (down count).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.43, Figure 23.45, and Figure 23.46, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output *1
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer value for compare match *1
Set the GTIOCA pin transition immediately after the count start in GTCCRC and GTCCRD.

Set forcible buffer transfer for compare match
Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly to GTCCRA.

Set buffer value for compare match
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.

Set automatic dead time setting function
Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.

Set buffer operation for dead time setting
Set buffer operation with TDBUE and TDBDE in GTDTCR.

Set dead time value
Set the first half dead time value in GTDVU and the second half dead time in GTDVD. When GTDVU is set with GTDTCR.TDFER set to 1, the same value is also set to GTDVD; the same dead time value can be set for the first and second halves.

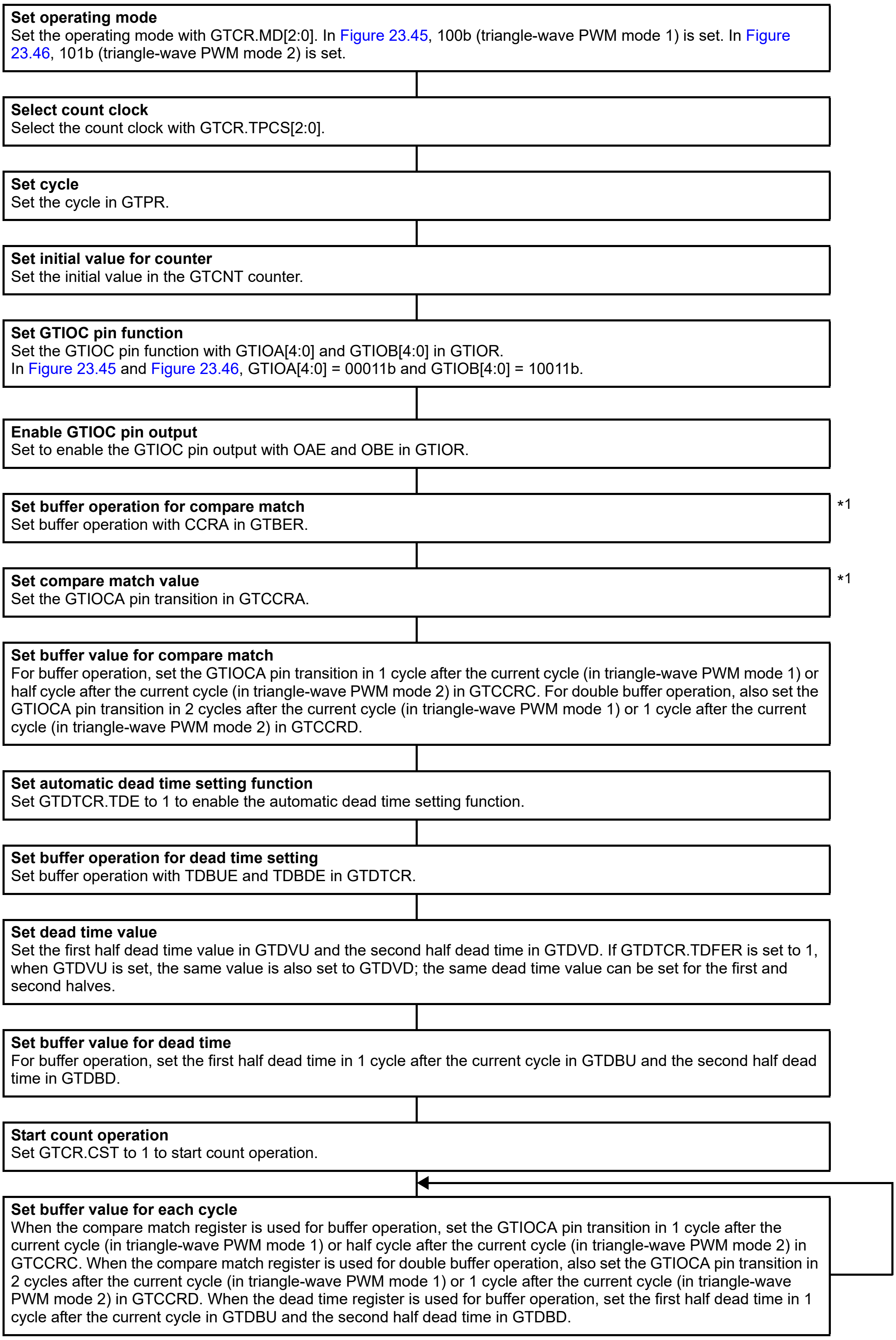
Set buffer value for dead time
For buffer operation, set the first half dead time in 1 cycle after the current cycle in GTDBU and the second half dead time in GTDBD.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
Set the GTIOCA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.
When the dead time register is used as a buffer register, set the first half dead time in 1 cycle after the current cycle in GTDBU and the second half dead time in GTDBD.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step for Set buffer value for compare match.

Figure 23.47 Example setting for automatic dead time setting function with saw-wave one-shot pulse mode, and triangle-wave PWM mode 3



Set operating mode
Set the operating mode with GTCR.MD[2:0]. In Figure 23.45, 100b (triangle-wave PWM mode 1) is set. In Figure 23.46, 101b (triangle-wave PWM mode 2) is set.

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set GTIOC pin function
Set the GTIOC pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR.
In Figure 23.45 and Figure 23.46, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.

Enable GTIOC pin output
Set to enable the GTIOC pin output with OAE and OBE in GTIOR.

Set buffer operation for compare match
Set buffer operation with CCRA in GTBER. *1

Set compare match value
Set the GTIOCA pin transition in GTCCRA. *1

Set buffer value for compare match
For buffer operation, set the GTIOCA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. For double buffer operation, also set the GTIOCA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.

Set automatic dead time setting function
Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.

Set buffer operation for dead time setting
Set buffer operation with TDBUE and TDBDE in GTDTCR.

Set dead time value
Set the first half dead time value in GTDVU and the second half dead time in GTDVD. If GTDTCR.TDFER is set to 1, when GTDVU is set, the same value is also set to GTDVD; the same dead time value can be set for the first and second halves.

Set buffer value for dead time
For buffer operation, set the first half dead time in 1 cycle after the current cycle in GTDBU and the second half dead time in GTDBD.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
When the compare match register is used for buffer operation, set the GTIOCA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. When the compare match register is used for double buffer operation, also set the GTIOCA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD. When the dead time register is used for buffer operation, set the first half dead time in 1 cycle after the current cycle in GTDBU and the second half dead time in GTDBD.

Note 1. When the PWM delay generation circuit is used, exchange the sequence of the step for Enable GTIOC pin output and the step for Set compare match value.

Figure 23.48 Example setting for automatic dead time setting function with triangle-wave PWM mode 1 or 2

23.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation is stopped and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, the count direction does not change when the UD bit in GTUDDTYC is modified during the count operation. Similarly, when the GTUDDTYC.UD bit is modified while the count operation is stopped and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected to the count cycle during up-counting and the GTPR value before the start of down-counting is reflected during down-counting.

Figure 23.49 shows an example of count direction changing function operation.

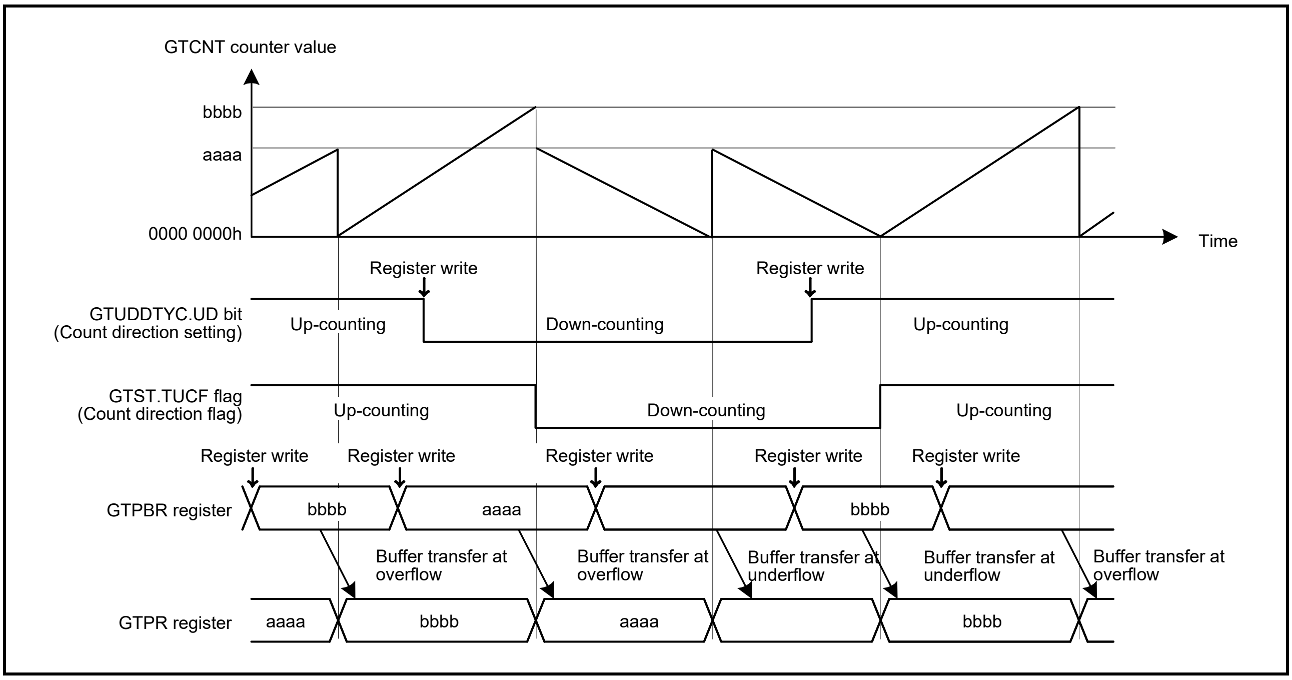


Figure 23.49 Example of count direction changing function operation during buffer operation

23.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCA pin and the GTIOCB pin are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

In saw-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an overflow or an underflow. If the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is set to 1 while the count operation is stopped, the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit value at that time is reflected at the start of counting.

In triangle-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an underflow.

If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0%/100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag
- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCB pin at cycle end is decided by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

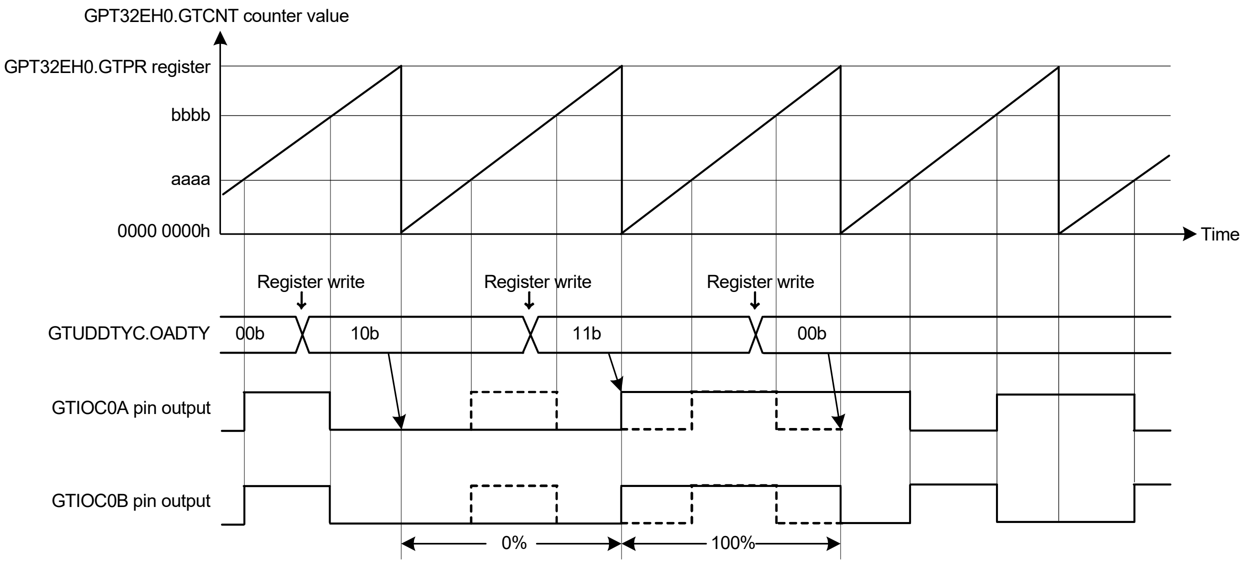
When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

GTUDDTYC.OADTYR selects the value that is the object of output retained/toggled at cycle end, when GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or when GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end). Table 23.7 shows the values of GTIOCA/GTIOCB pin output at cycle end.

Table 23.7 Output values after releasing 0% or 100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	**1**	**0**	**1**
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	-	0	0	0	0
10 (high output at cycle end)	-	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 23.50 shows an example of output duty 0% and 100% function operation.



[Setting examples]

GPT32EH0.GTIOR.GTIOA[4:0] bits: 00011b
Initial low output, output toggled at compare match, output retained at cycle end

GPT32EH0.GTUDDTYC.OADTYR bit: 0b
Applied the value of duty 0%/100% output to GTIOA[3:2] bits function after 0%/100% duty setting is released

GPT32EH0.GTIOR.GTIOB[4:0] bits: 00011b
Initial low output, output toggled at compare match, output retained at cycle end

GPT32EH0.GTUDDTYC.OBDTYR bit: 1
Applied the value of masked compare match output to GTIOB[3:2] bits function after 0%/100% duty setting is released

Figure 23.50 Example of output duty 0% and 100% functions

23.3.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by the following hardware sources:

- External trigger input
- ELC event input
- GTIOCA/GTIOCB pin input.

23.3.7.1 Hardware start operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 23.51 shows an example of a count start operation by a hardware source. Figure 23.52 shows the setting example.

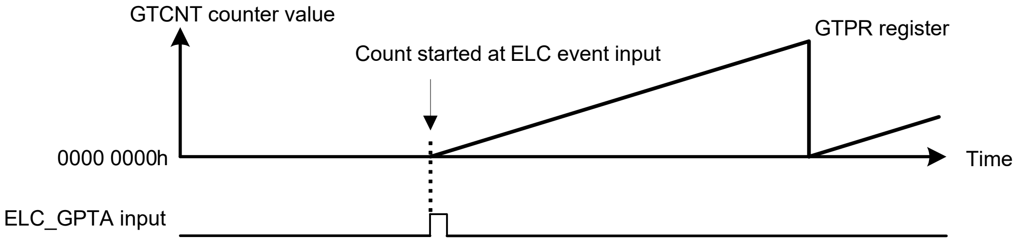


Figure 23.51 Example of count start operation by hardware source, started at the input of the signal from ELC_GPTA

Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.51, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.51, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.51, 0000 0000h is set.

Set hardware count start
Select a hardware source for starting count operation in GTSSR register.
In Figure 23.51, GTSSR.SSELCA = 1.

Set hardware source operation
Set operation of the hardware source selected by GTSSR register and start counting. In Figure 23.51, the ELC_GPTA input operation is set.

Figure 23.52 Example setting for count start operation by a hardware source

23.3.7.2 Hardware stop operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR. Figure 23.53 shows an example of a count stop operation by a hardware source. Figure 23.54 shows the setting example. In this example, the count operation stops and restarts at the edge of the ELC event input.

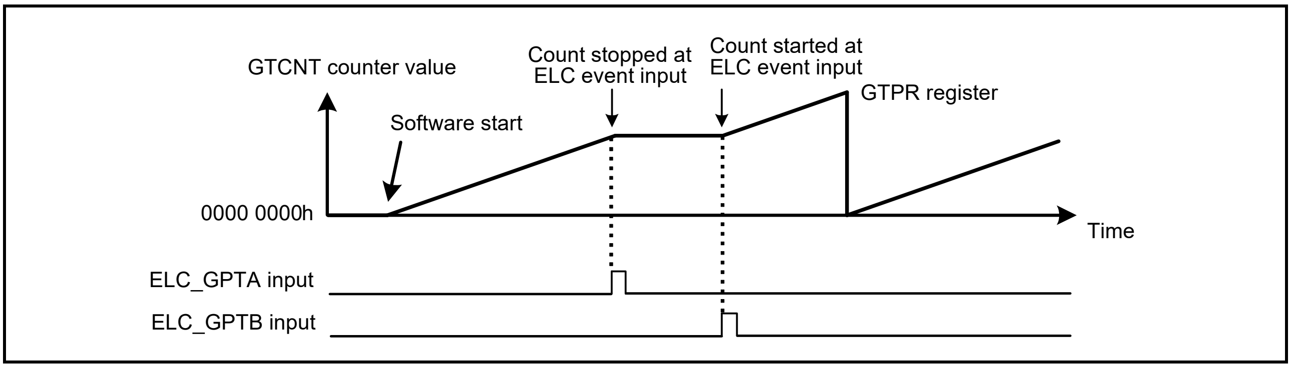
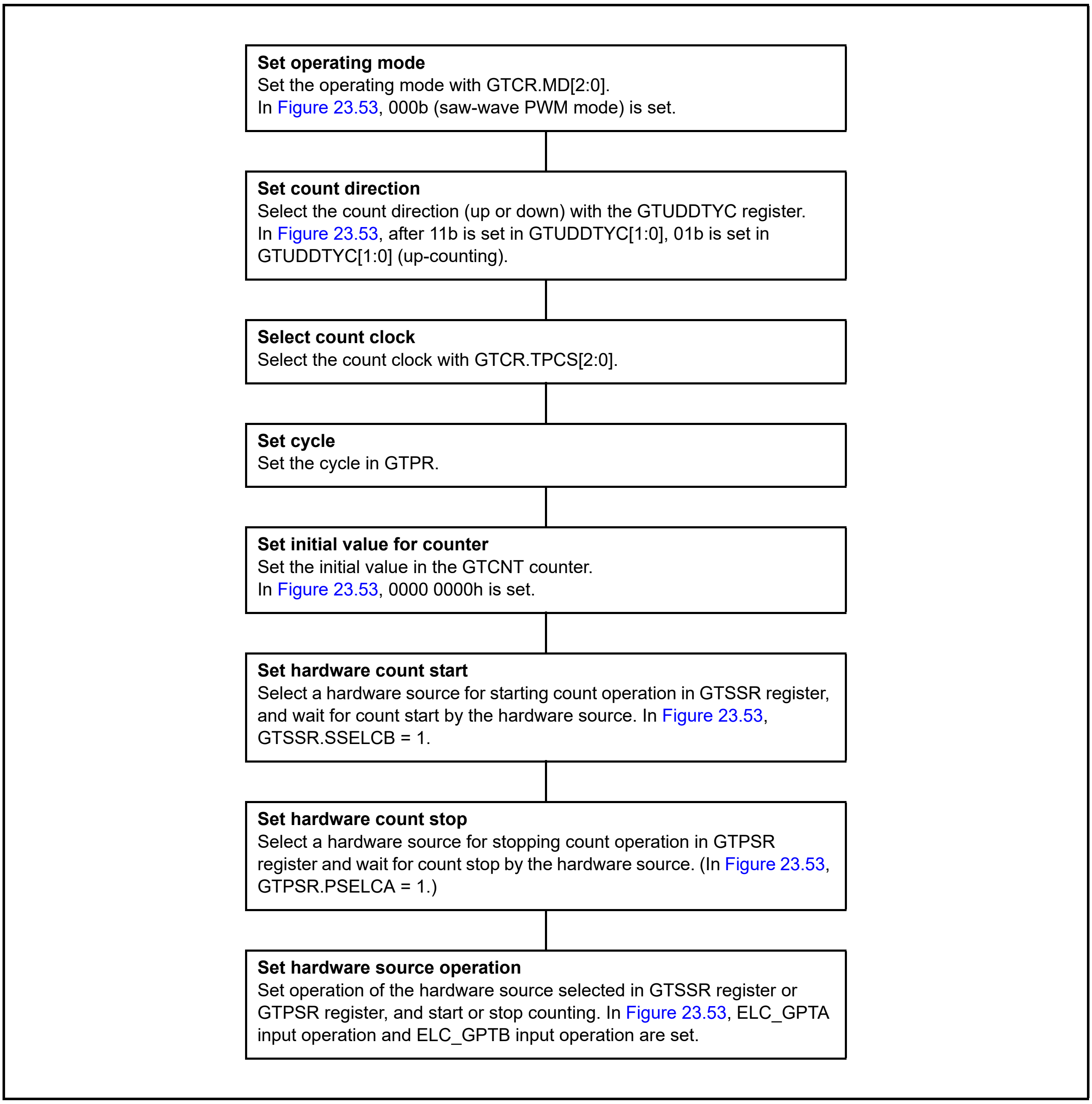


Figure 23.53 Example of count stop operation by hardware source started by software, stopped at ELC_GPTA input, and restarted at ELC_GPTB input



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.53, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.53, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.53, 0000 0000h is set.

Set hardware count start
Select a hardware source for starting count operation in GTSSR register, and wait for count start by the hardware source. In Figure 23.53, GTSSR.SSELCB = 1.

Set hardware count stop
Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. (In Figure 23.53, GTPSR.PSELCA = 1.)

Set hardware source operation
Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. In Figure 23.53, ELC_GPTA input operation and ELC_GPTB input operation are set.

Figure 23.54 Example setting for count stop operation by a hardware source

Figure 23.55 shows an example of a count start/stop operation by a hardware source. Figure 23.56 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

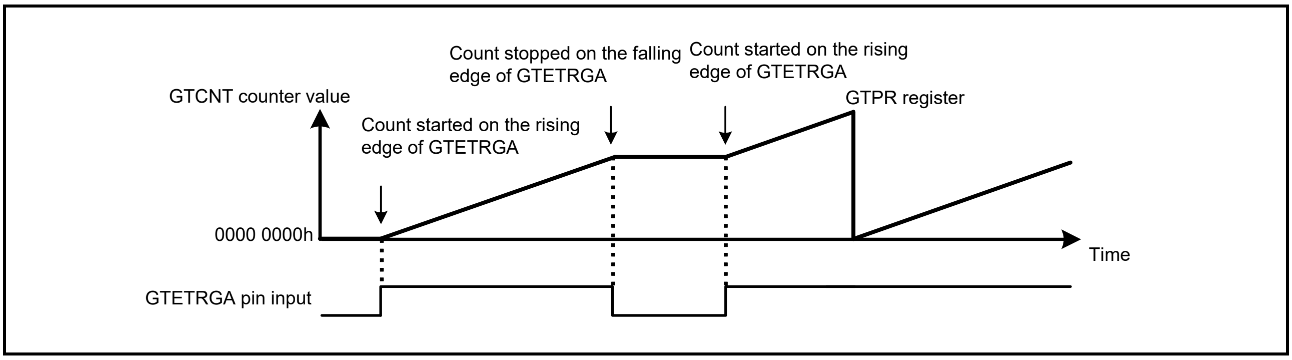
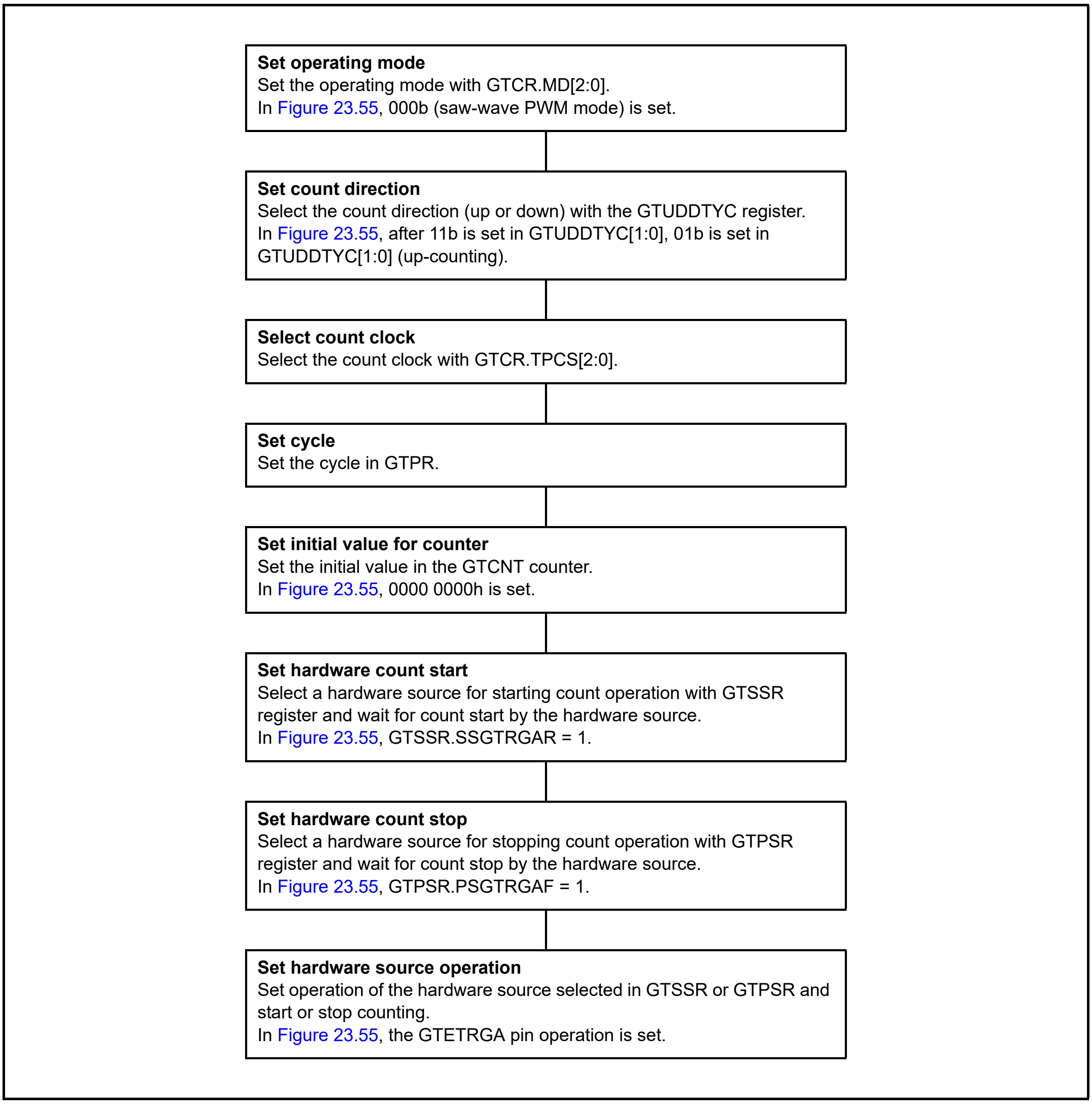


Figure 23.55 Example of count start/stop operation by hardware source, started on the rising edge of the GTETRGA pin input and stopped on the falling edge of the GTETRGA pin input



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.55, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.55, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.55, 0000 0000h is set.

Set hardware count start
Select a hardware source for starting count operation with GTSSR register and wait for count start by the hardware source.
In Figure 23.55, GTSSR.SSGTRGAR = 1.

Set hardware count stop
Select a hardware source for stopping count operation with GTPSR register and wait for count stop by the hardware source.
In Figure 23.55, GTPSR.PSGTRGAF = 1.

Set hardware source operation
Set operation of the hardware source selected in GTSSR or GTPSR and start or stop counting.
In Figure 23.55, the GTETRGA pin operation is set.

Figure 23.56 Example setting for count start/stop operation by a hardware source

23.3.7.3 Hardware clear operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 13) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 23.57 and Figure 23.58 show examples of the GTCNT counter clearing operation by a hardware source. Figure 23.59 shows the setting example. In this example, the GTCNT counter starts at the edge of the ELC_GPTA input, and the counter stops/clears at the edge of the ELC_GPTB input.

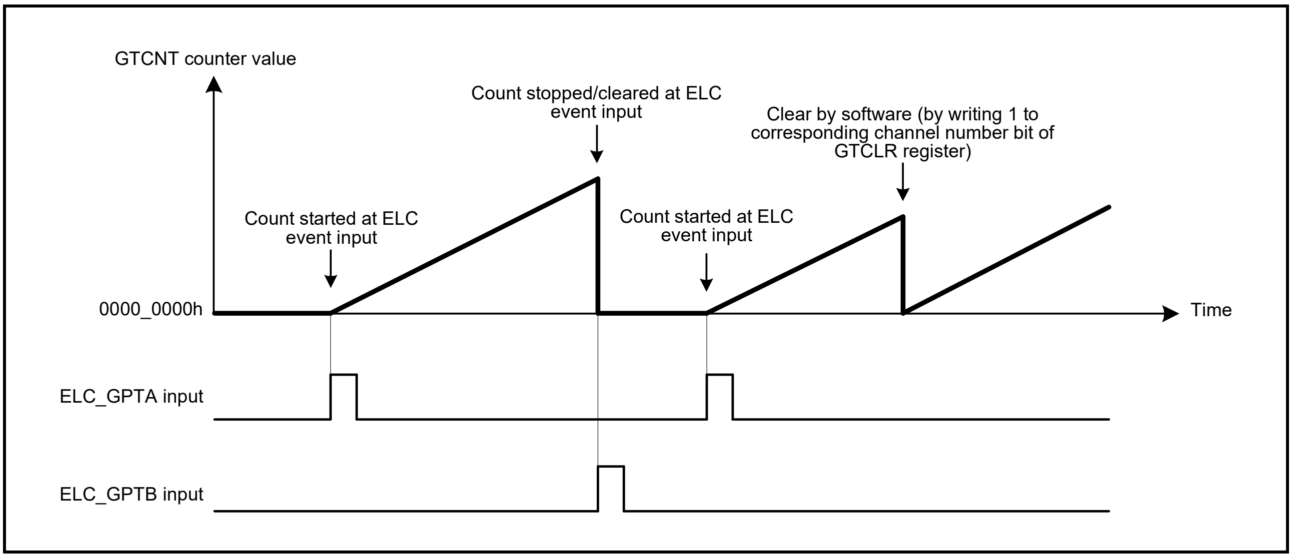


Figure 23.57 Examples of count clearing operation by hardware source with saw wave up-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input

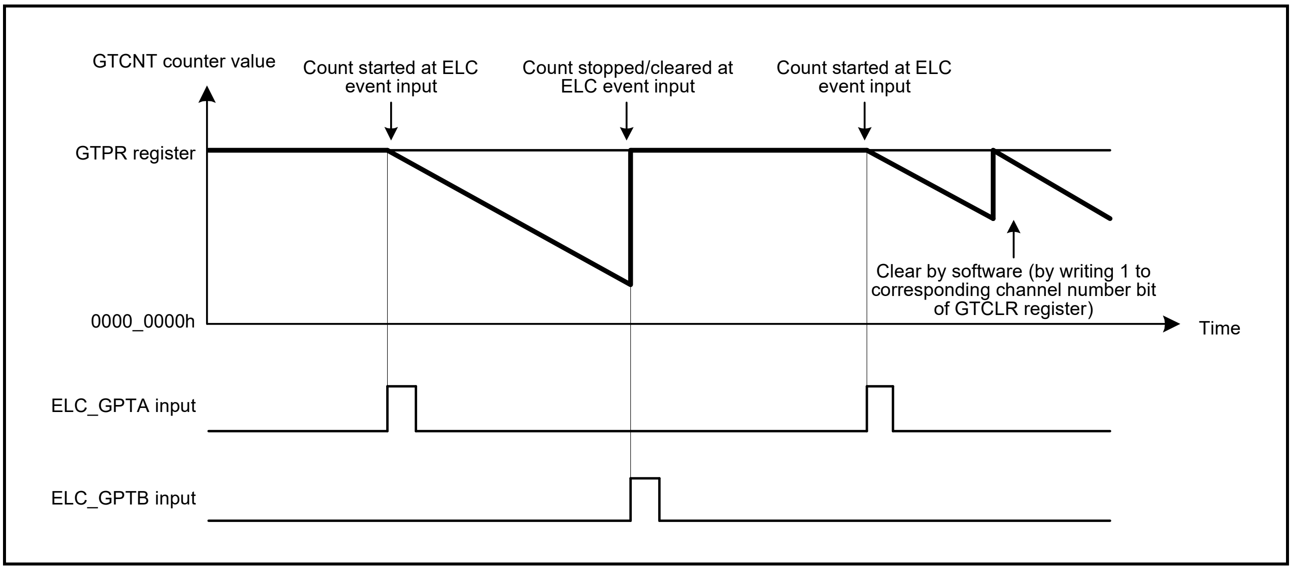
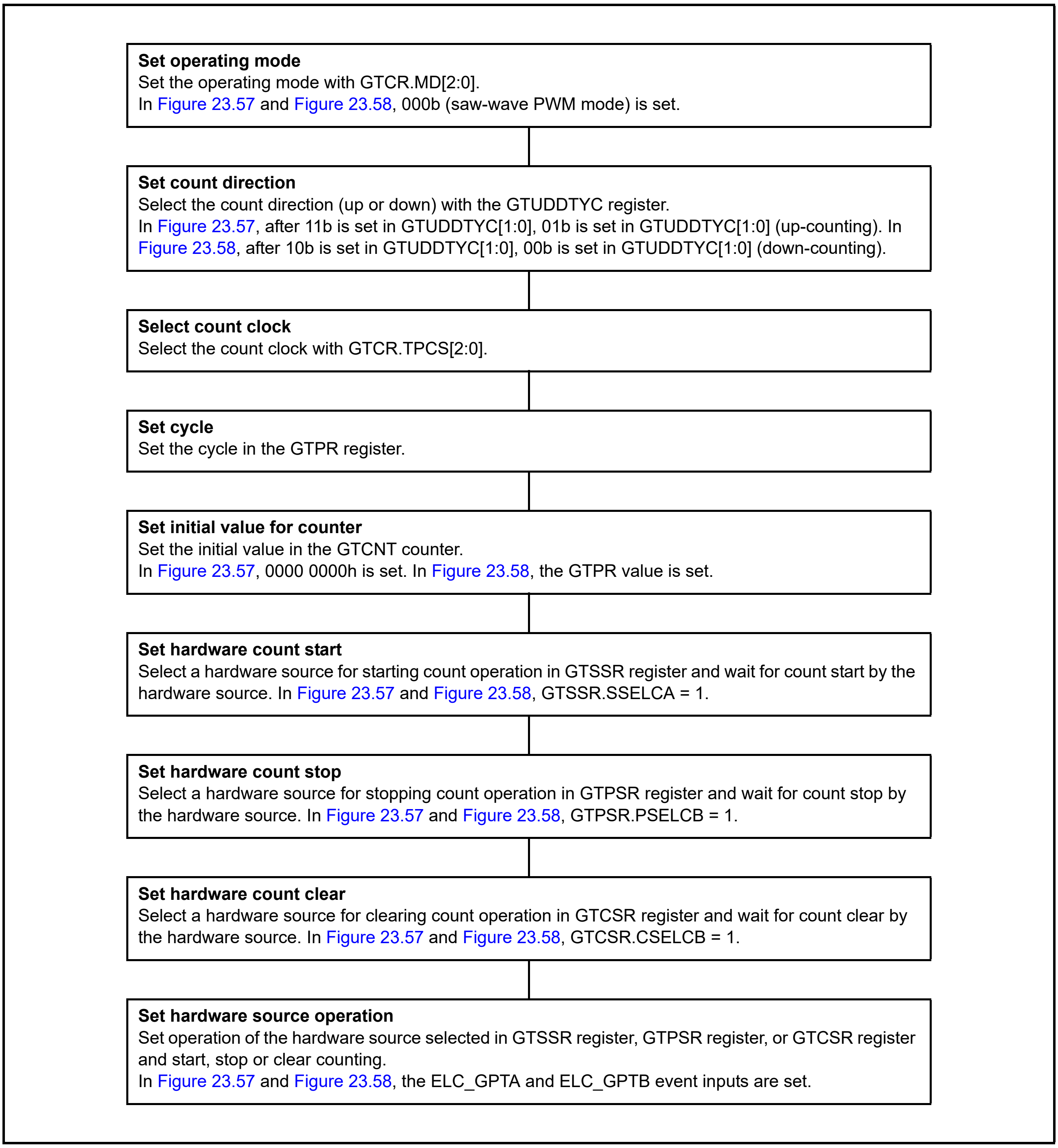


Figure 23.58 Examples of count clearing operation by hardware source with saw wave down-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.57 and Figure 23.58, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.57, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting). In Figure 23.58, after 10b is set in GTUDDTYC[1:0], 00b is set in GTUDDTYC[1:0] (down-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in the GTPR register.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.57, 0000 0000h is set. In Figure 23.58, the GTPR value is set.

Set hardware count start
Select a hardware source for starting count operation in GTSSR register and wait for count start by the hardware source. In Figure 23.57 and Figure 23.58, GTSSR.SSELCA = 1.

Set hardware count stop
Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In Figure 23.57 and Figure 23.58, GTPSR.PSELCB = 1.

Set hardware count clear
Select a hardware source for clearing count operation in GTCSR register and wait for count clear by the hardware source. In Figure 23.57 and Figure 23.58, GTCSR.CSELCB = 1.

Set hardware source operation
Set operation of the hardware source selected in GTSSR register, GTPSR register, or GTCSR register and start, stop or clear counting.
In Figure 23.57 and Figure 23.58, the ELC_GPTA and ELC_GPTB event inputs are set.

Figure 23.59 Example setting for count clearing operation by a hardware source

The GPTn_OVF/GPTn_UDF (n = 0 to 13) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 23.60 shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 13) interrupt.

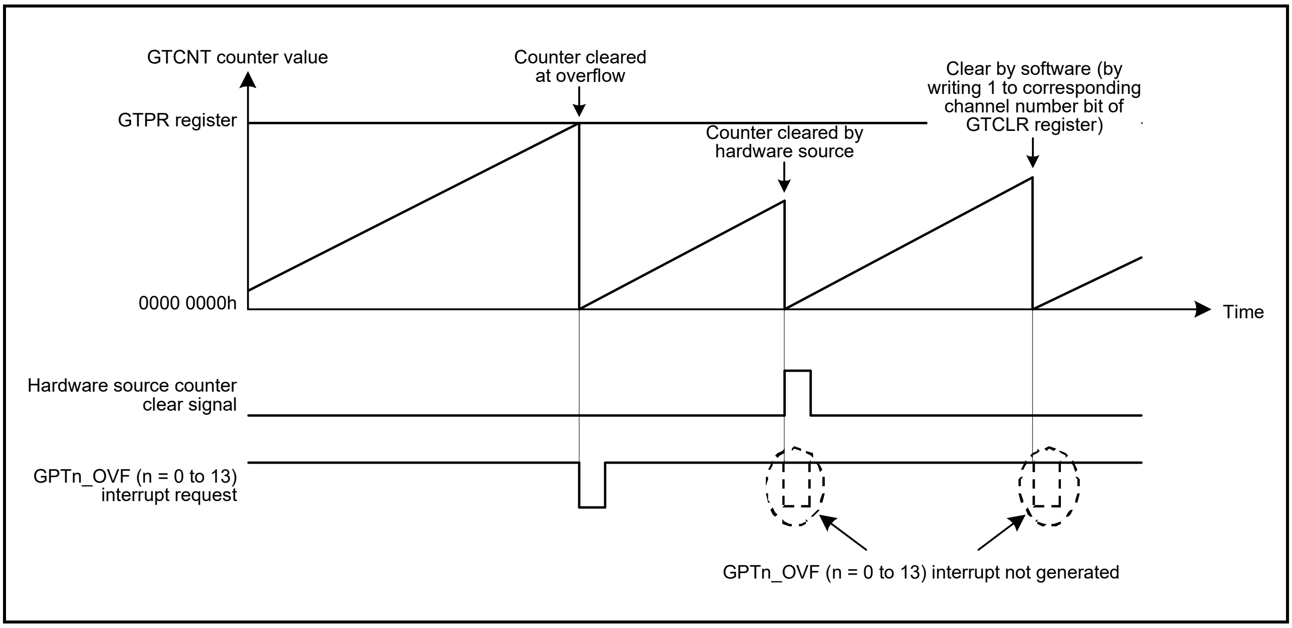


Figure 23.60 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 13) interrupt

23.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

23.3.8.1 Synchronized operation by software

The GTCNT counters can be started, stopped, and cleared on multiple channels by setting the associated GTSTR, GTSTP, or GTCLR bits simultaneously to 1.

Count start with a phase difference is possible by setting the initial value in the GTCNT counter and setting the associated GTSTR bits simultaneously to 1.

Figure 23.61 shows an example of a simultaneous start, stop, and clear by software. Figure 23.62 shows an example of phase start operation by software.

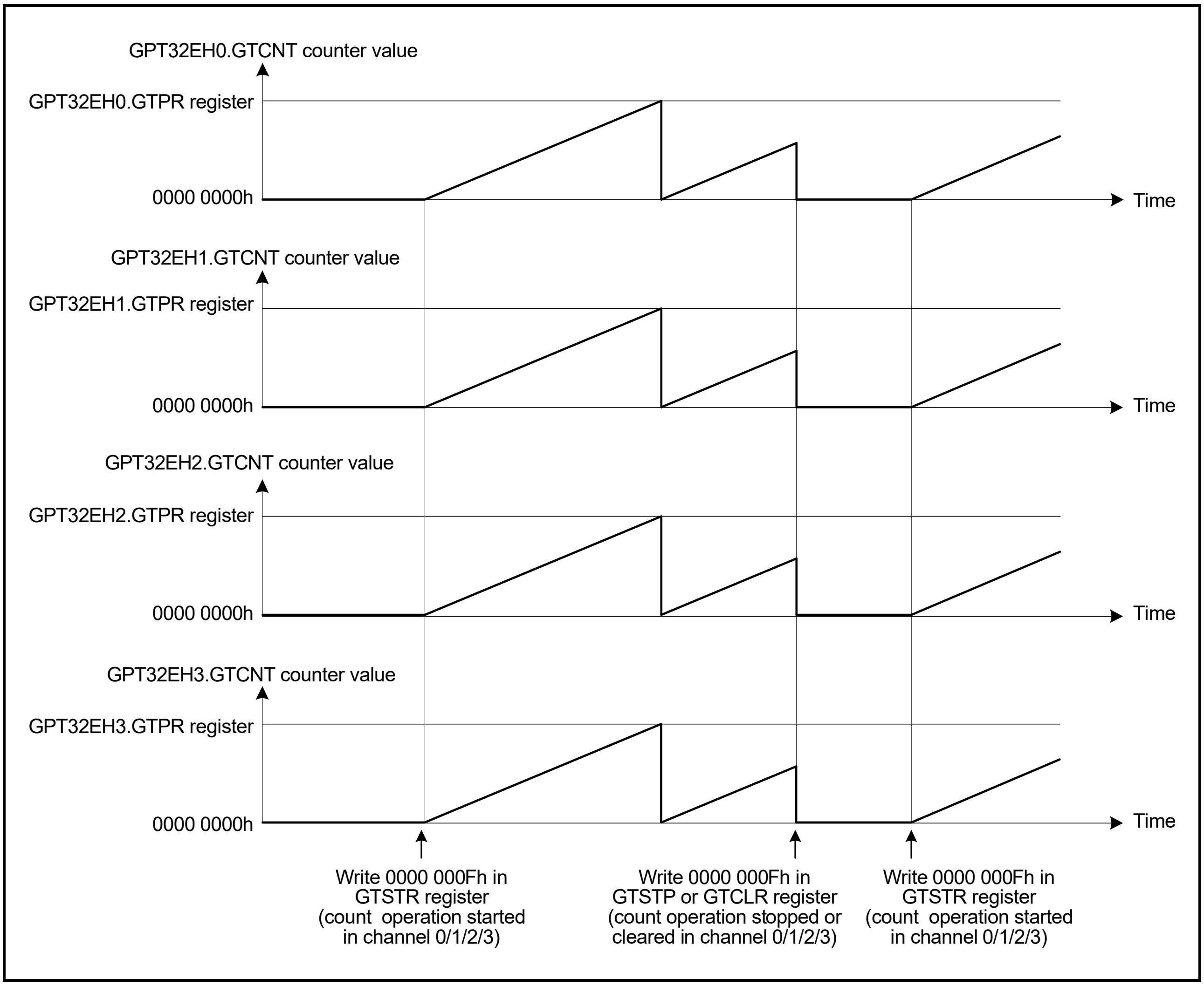


Figure 23.61 **Example of a simultaneous start, stop, and clear by software, with the same count cycle (GTPR register value)**

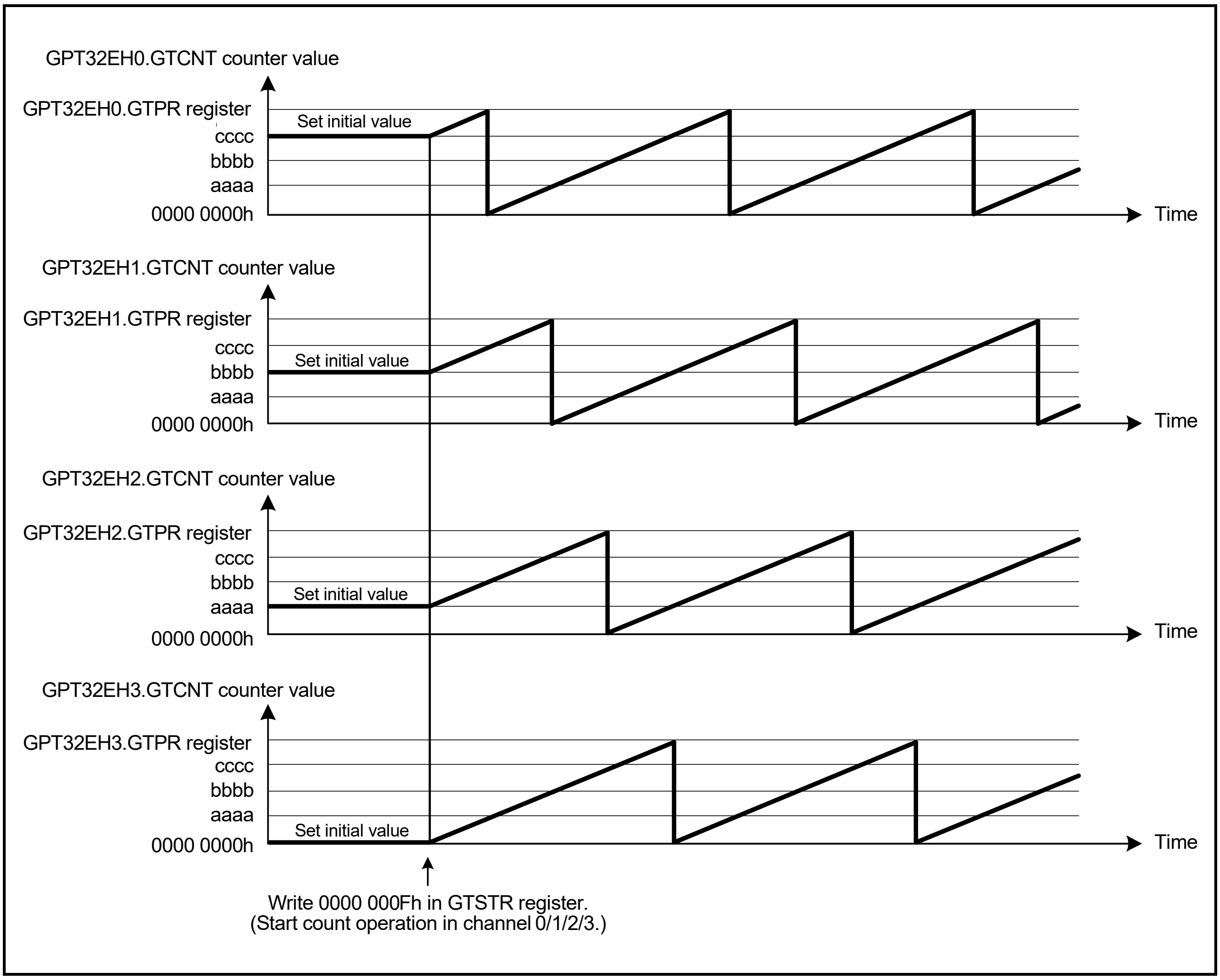


Figure 23.62 Example of software phase start with the same count cycle (GTPR register value)

23.3.8.2 Synchronized operation by hardware

The GTCNT counters can be started simultaneously by the following hardware sources:

- External trigger input
- ELC event input.

Figure 23.63 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Figure 23.64 shows the setting example.

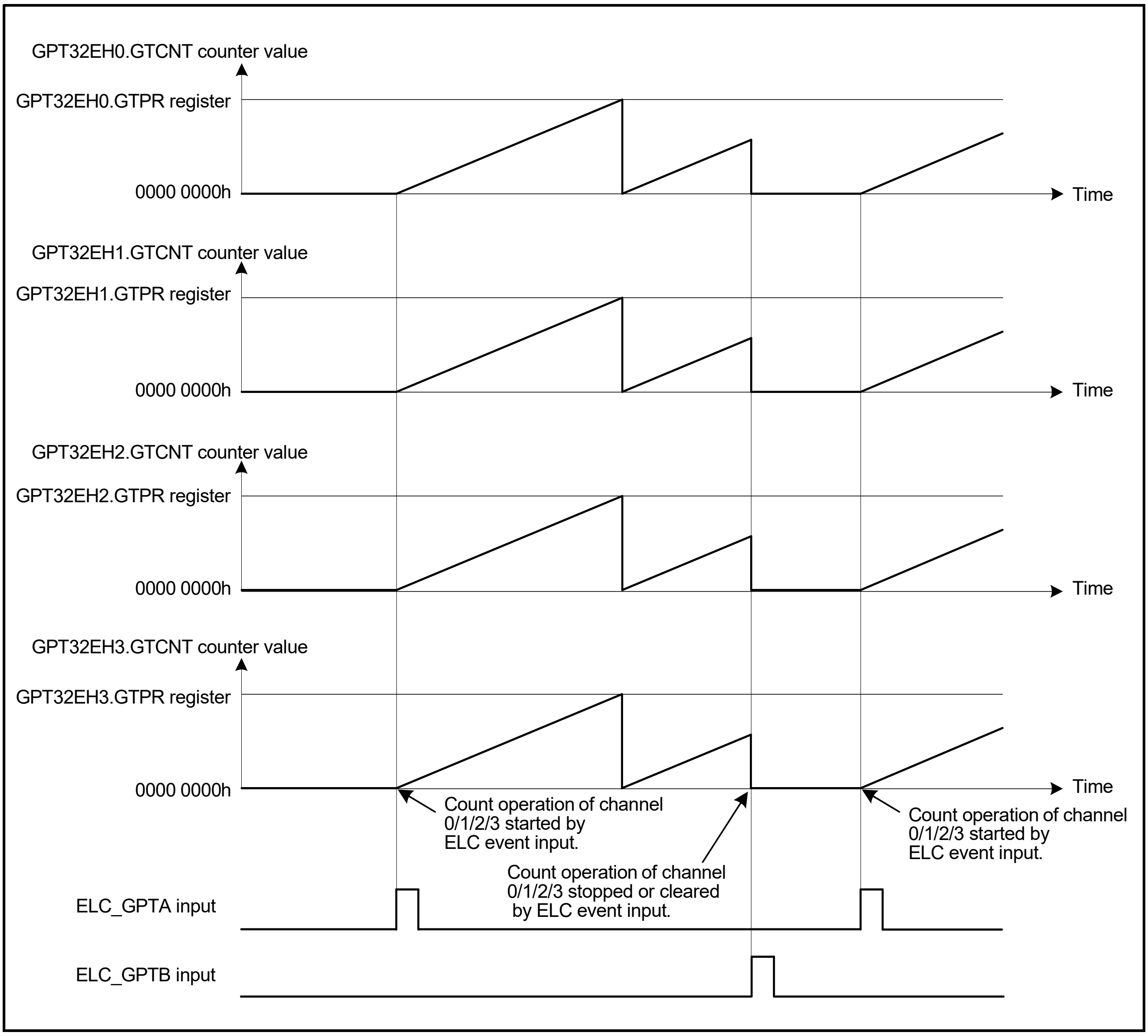
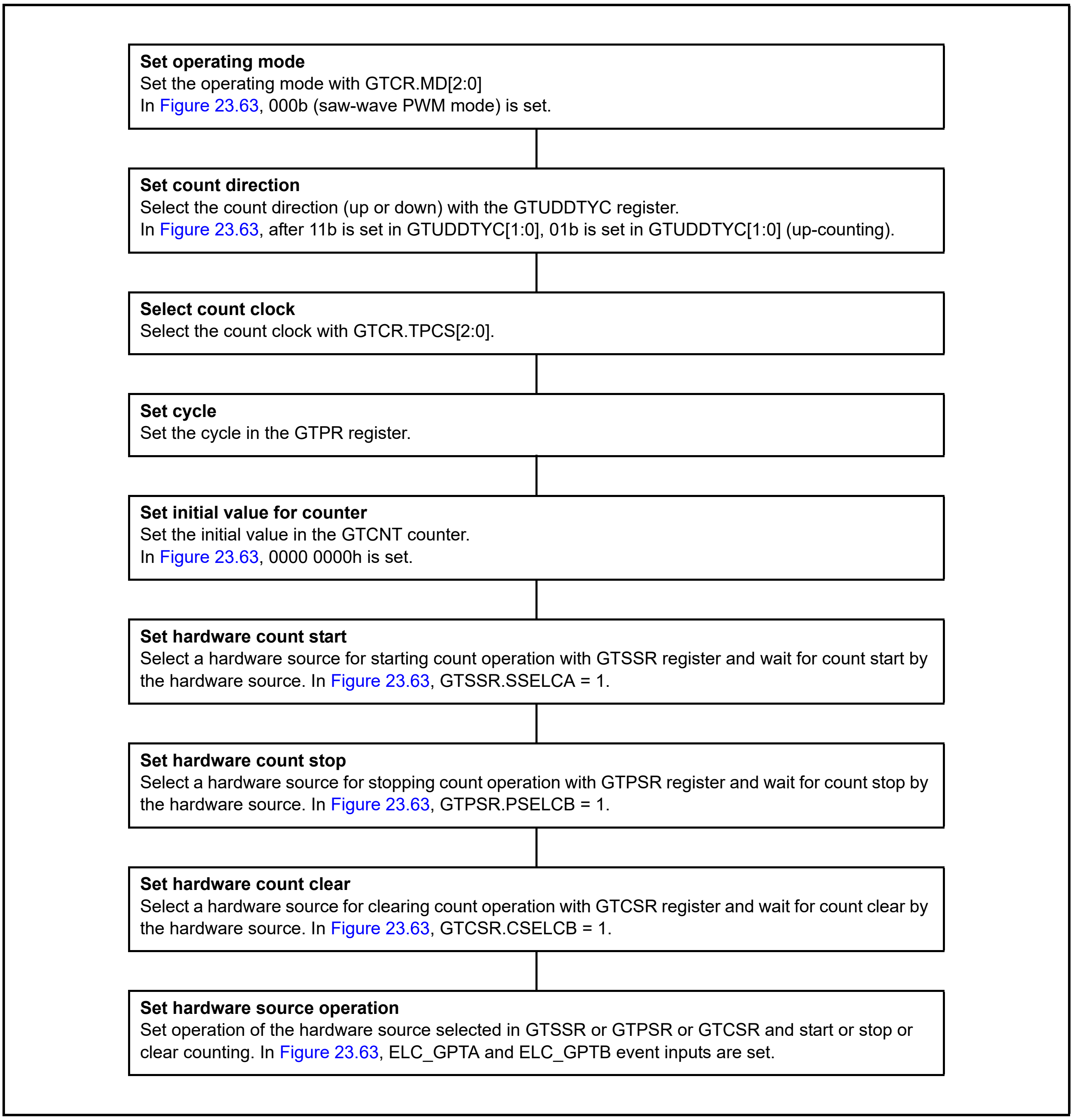


Figure 23.63 Example of a simultaneous start, stop, and clear by hardware source with the same count cycle (GTPR register value)



Set operating mode
Set the operating mode with GTCR.MD[2:0]
In Figure 23.63, 000b (saw-wave PWM mode) is set.

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 23.63, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in the GTPR register.

Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 23.63, 0000 0000h is set.

Set hardware count start
Select a hardware source for starting count operation with GTSSR register and wait for count start by the hardware source. In Figure 23.63, GTSSR.SSELCA = 1.

Set hardware count stop
Select a hardware source for stopping count operation with GTPSR register and wait for count stop by the hardware source. In Figure 23.63, GTPSR.PSELCB = 1.

Set hardware count clear
Select a hardware source for clearing count operation with GTCSR register and wait for count clear by the hardware source. In Figure 23.63, GTCSR.CSELCB = 1.

Set hardware source operation
Set operation of the hardware source selected in GTSSR or GTPSR or GTCSR and start or stop or clear counting. In Figure 23.63, ELC_GPTA and ELC_GPTB event inputs are set.

Figure 23.64 Example setting for simultaneous start by hardware source

23.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 28 phases of linked PWM waveforms for a maximum of 14 channels by multiple GPTs.

Figure 23.65 shows an example in which four channels perform synchronized operation in saw-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

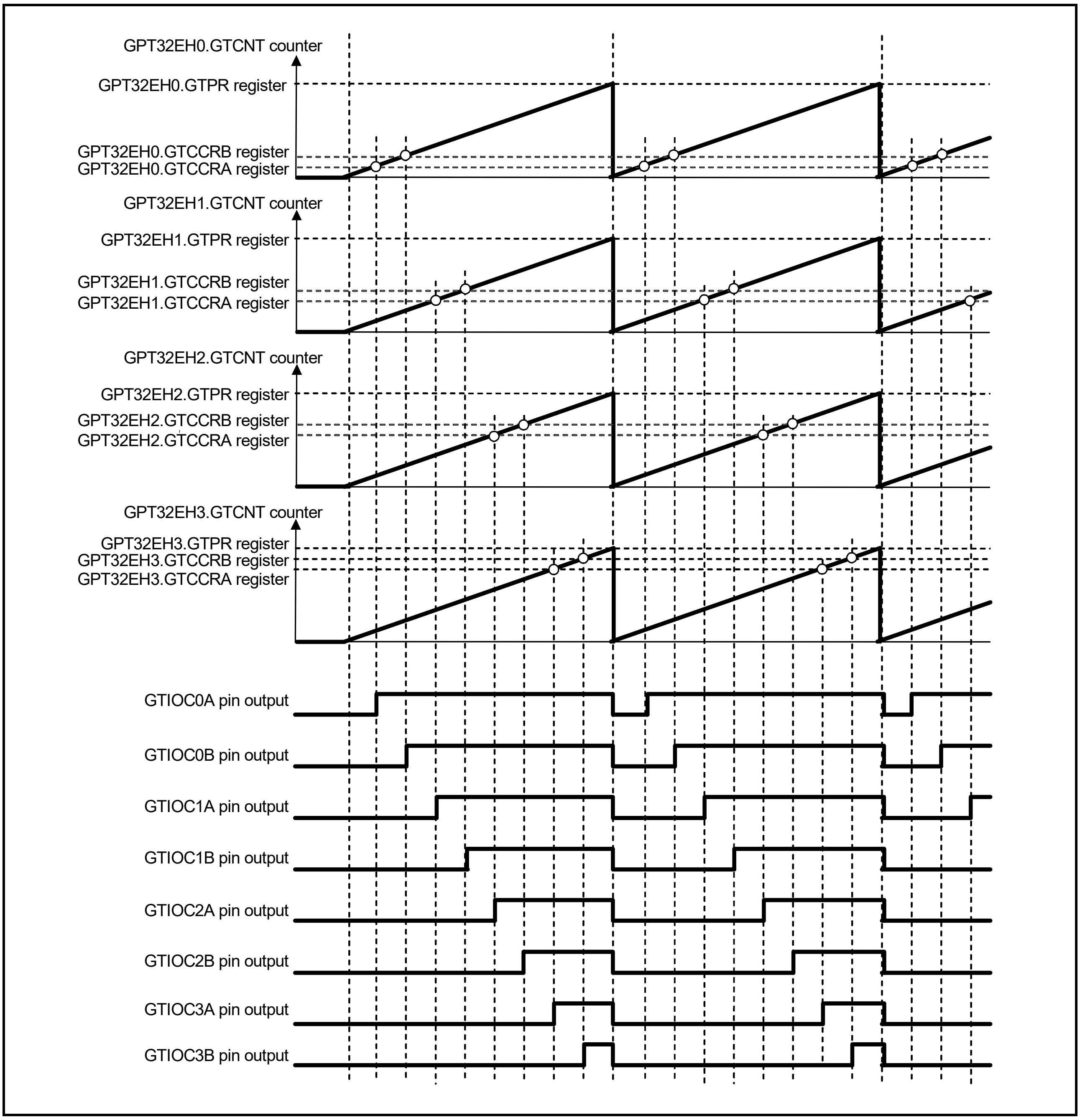


Figure 23.65 Example of synchronized PWM output

(2) 3-phase saw-wave complementary PWM output

Figure 23.66 shows an example in which three channels perform synchronized operation in saw-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOCA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

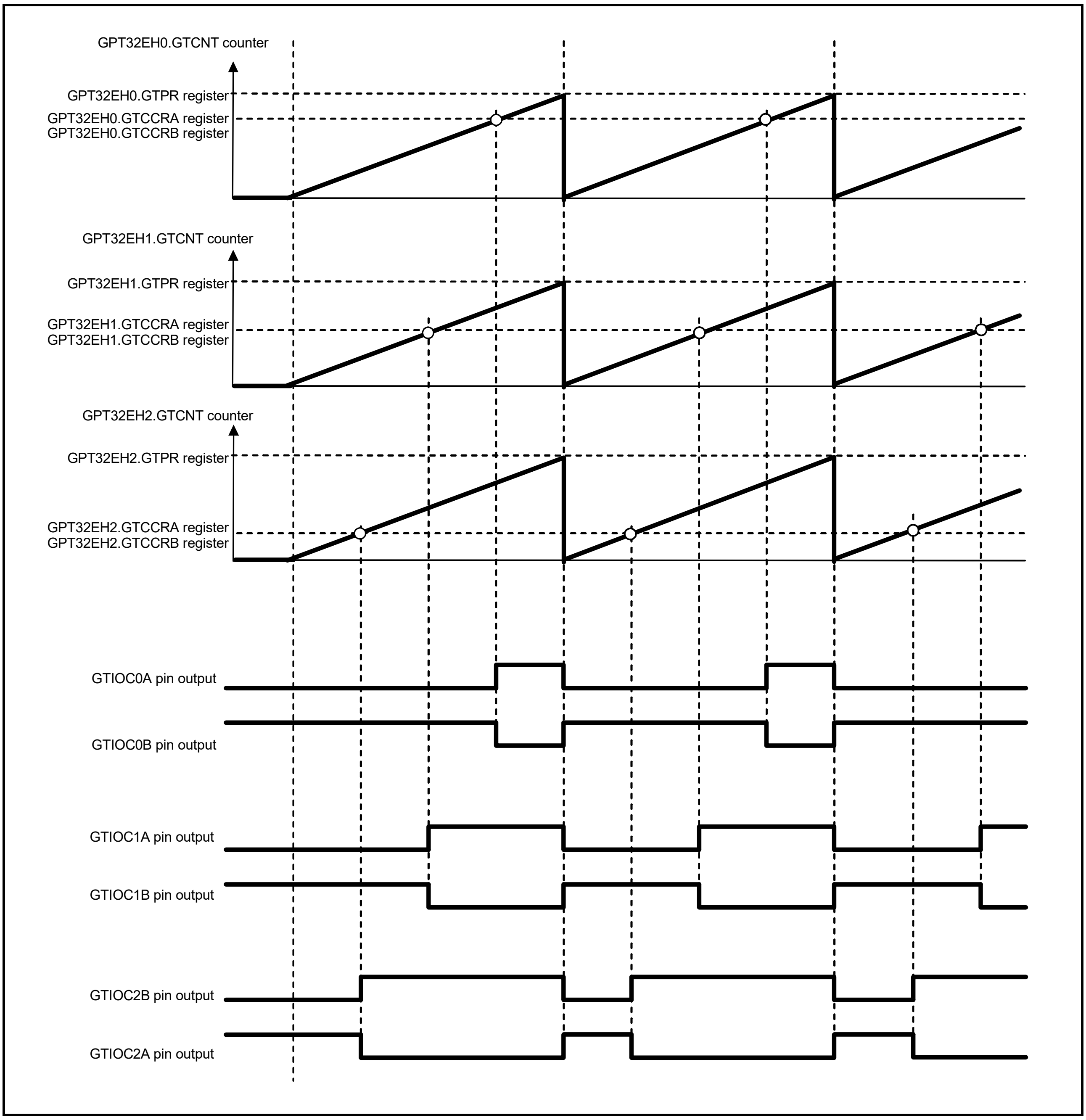


Figure 23.66 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

Figure 23.67 shows an example in which three channels perform synchronized operation in saw-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

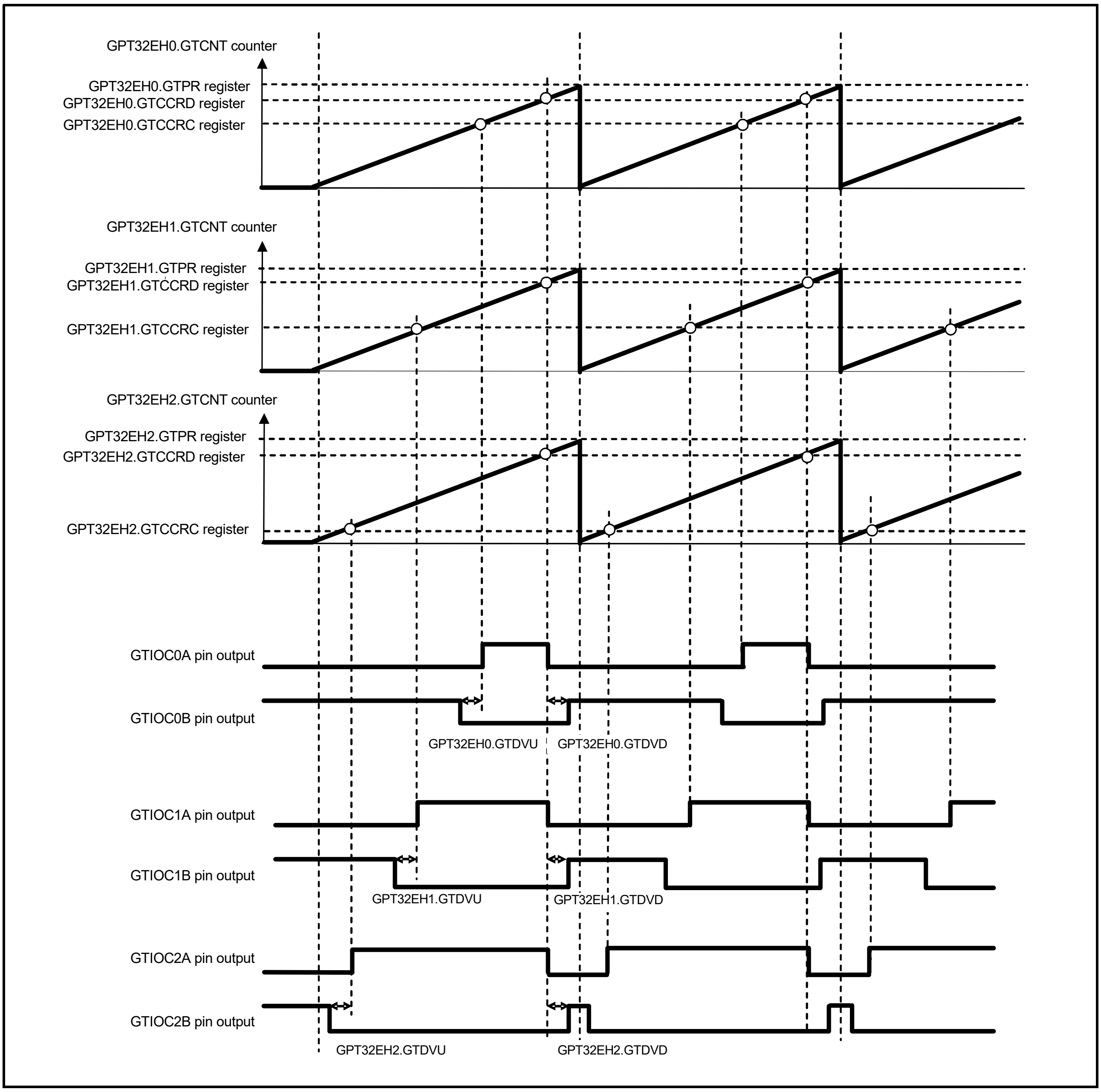


Figure 23.67 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-phase triangle-wave complementary PWM output

Figure 23.68 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

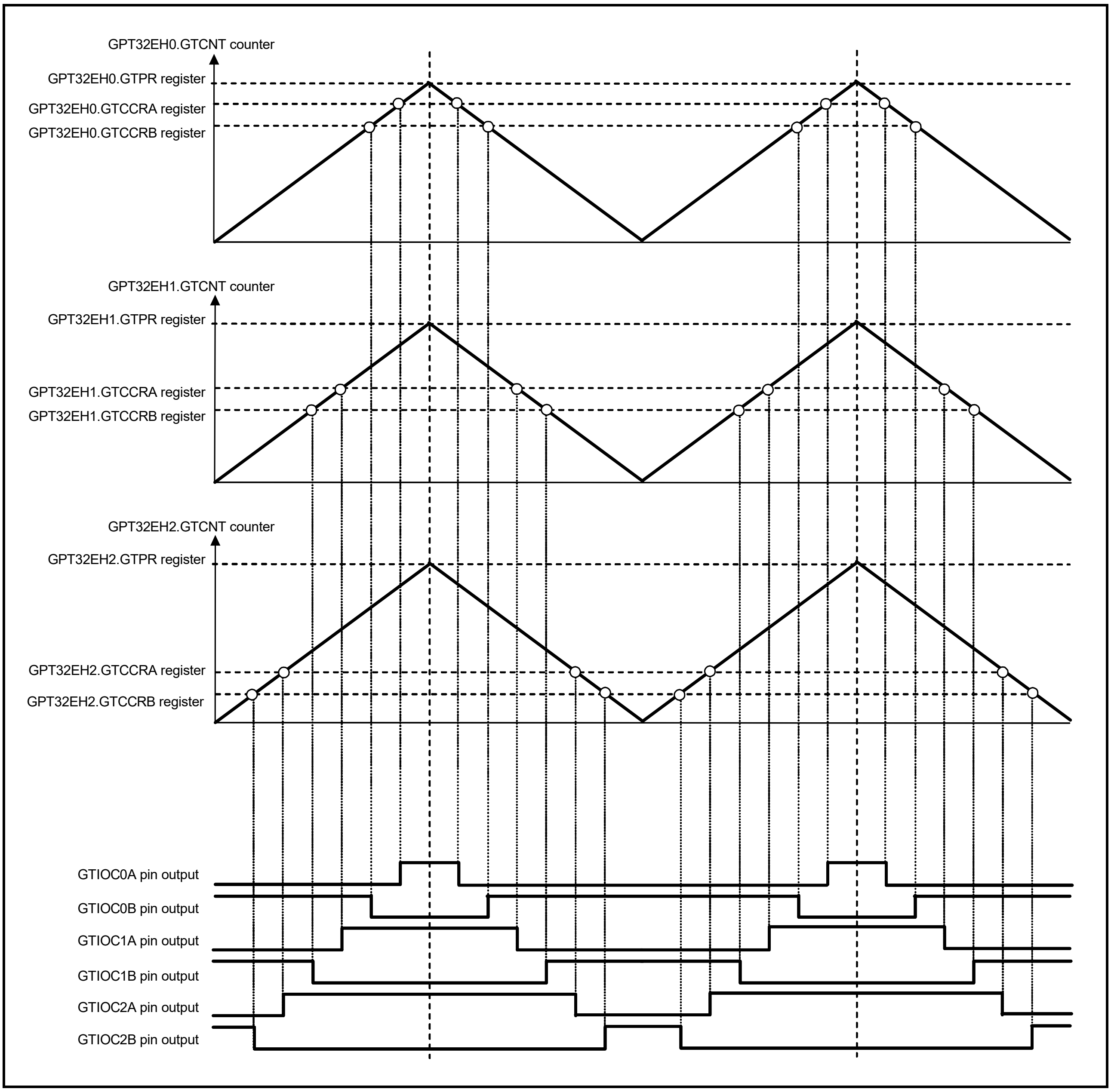


Figure 23.68 Example of 3-phase triangle-wave complementary PWM output

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

Figure 23.69 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

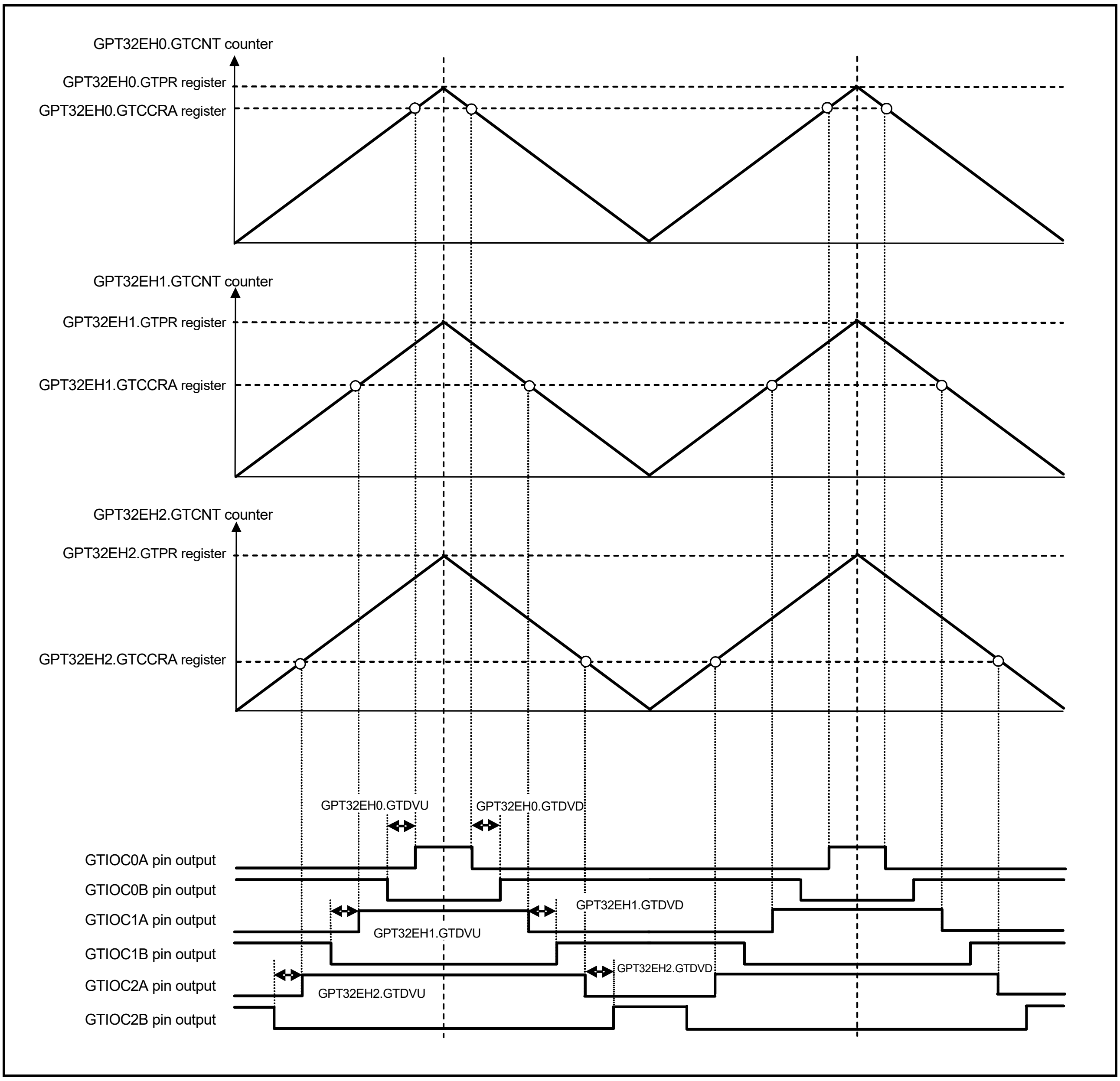


Figure 23.69 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 23.70 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

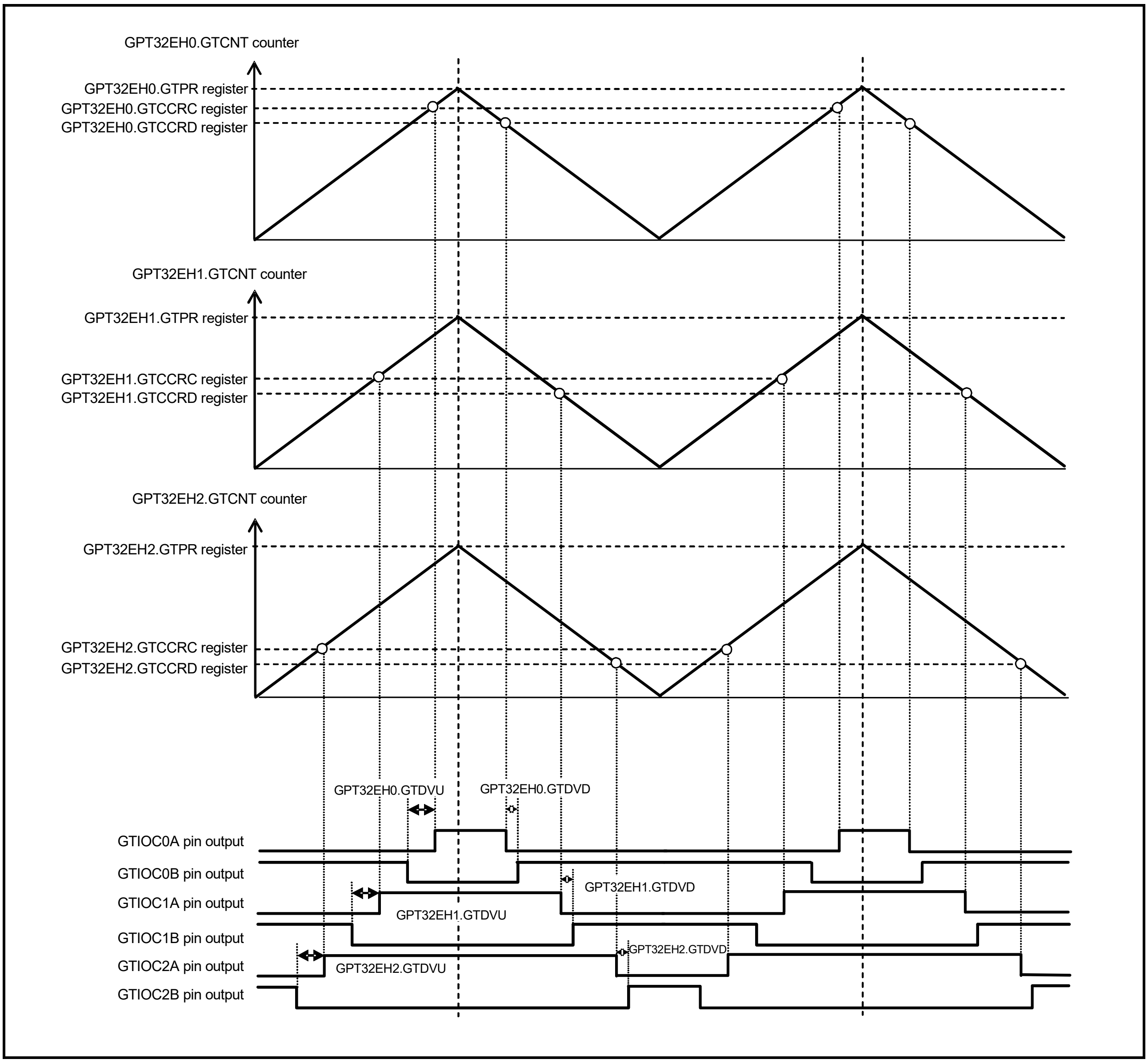


Figure 23.70 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

23.3.10 Phase Counting Function

The phase difference between the GTIOCA and GTIOCB pin inputs is detected and the associated GTCNT counts up or counts down. The detectable phase difference is available in any combination with the relationship between the edge and the level of GTIOCA and GTIOCB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see section 23.3.1.1, Counter operation.

Figure 23.71 to Figure 23.80 show phase counting modes 1 to 5. Table 23.8 to Table 23.17 show conditions of up-counting or down-counting and lists settings for the GTUPSR and GTDNSR registers.

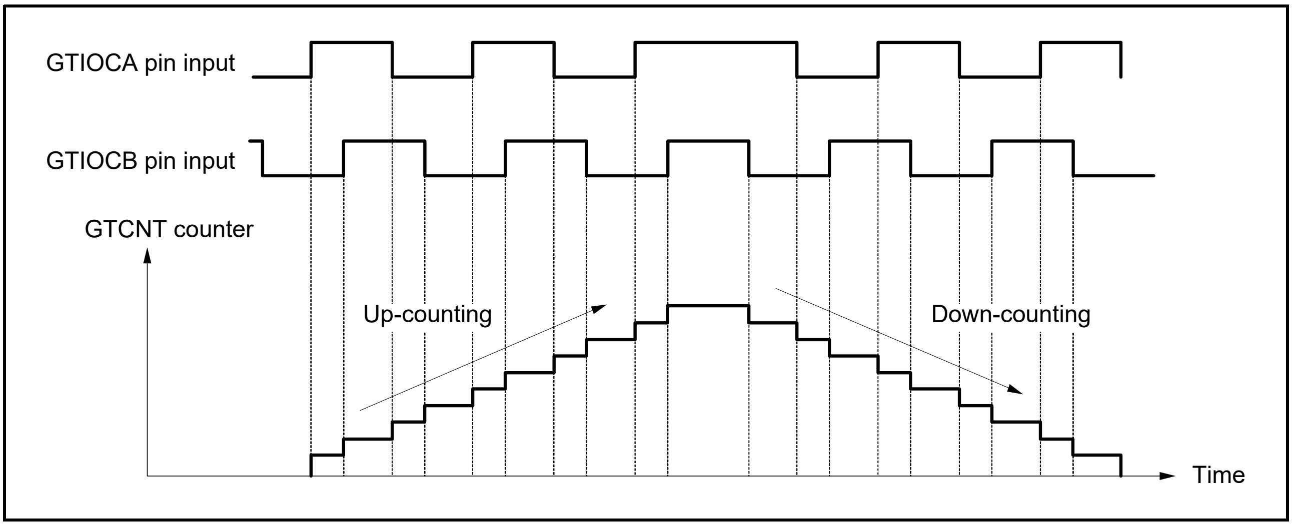


Figure 23.71 Example of phase counting mode 1

Table 23.8 Conditions of up-counting and down-counting in phase counting mode 1

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Up-counting	GTUPSR = 0000 6900h GTDNSR = 0000 9600h
low	↓		
↑	low		
↓	high		
high	↓	Down-counting	
low	↑		
↑	high		
↓	low		

↑ : Rising edge

↓ : Falling edge

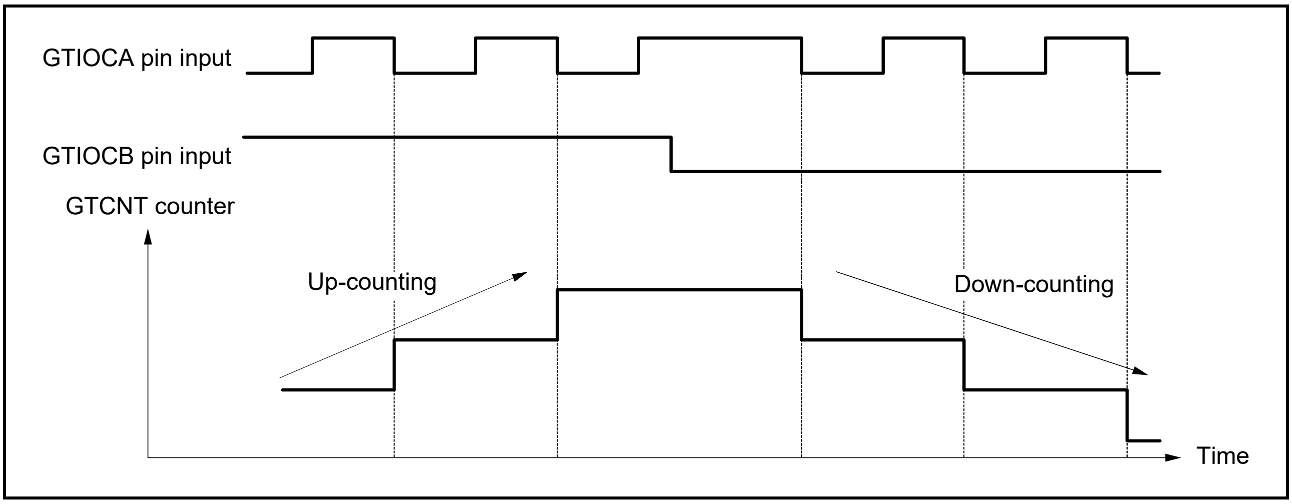


Figure 23.72 Example of phase counting mode 2 (A)

Table 23.9 Conditions of up-counting and down-counting in phase counting mode 2 (A)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Don't care	GTUPSR = 0000 0800h GTDNSR = 0000 0400h
low	↓		
↑	low		
↓	high	Up-counting	
high	↓	Don't care	
low	↑		
↑	high		
↓	low	Down-counting	

↑ : Rising edge
↓ : Falling edge

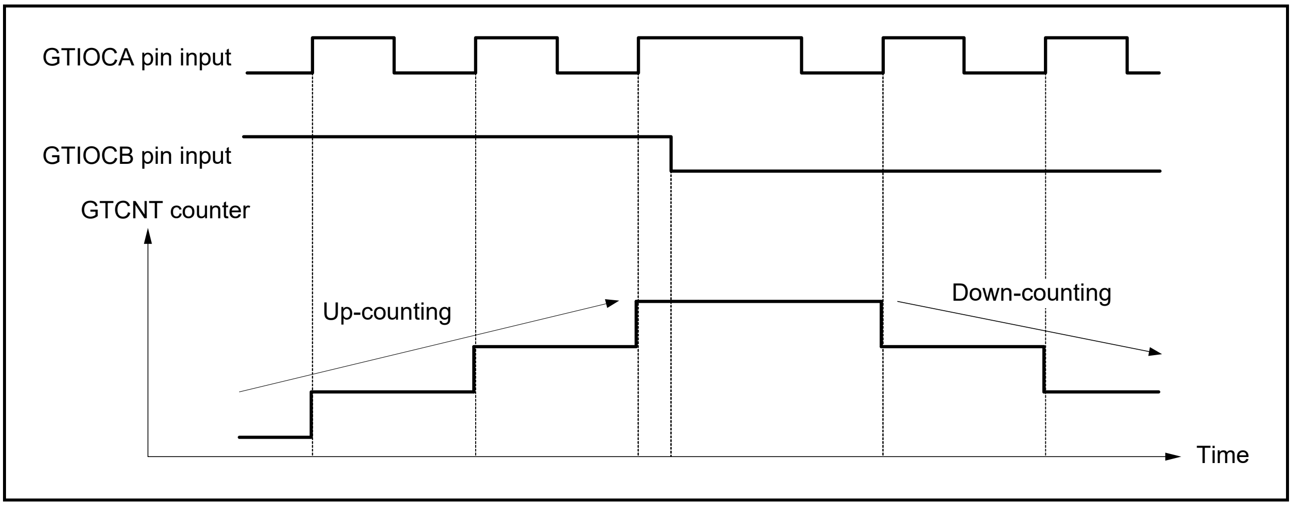


Figure 23.73 Example of phase counting mode 2 (B)

Table 23.10 Conditions of up-counting and down-counting in phase counting mode 2 (B)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Don't care	GTUPSR = 0000 0200h GTDNSR = 0000 0100h
low	↓		
↑	low	Down-counting	
↓	high	Don't care	
high	↓		
low	↑		
↑	high	Up-counting	
↓	low	Don't care	

↑ : Rising edge
↓ : Falling edge

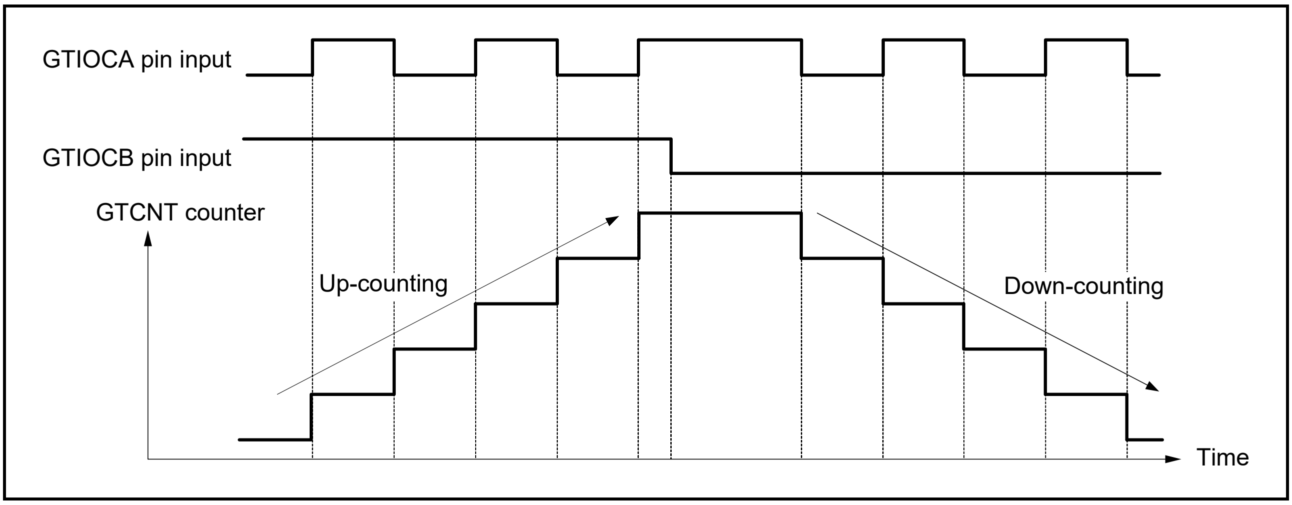


Figure 23.74 Example of phase counting mode 2 (C)

Table 23.11 Conditions of up-counting and down-counting in phase counting mode 2 (C)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Don’t care	GTUPSR = 0000 0A00h GTDNSR = 0000 0500h
low	↓		
↑	low	Down-counting	
↓	high	Don’t care	
high	↓		
low	↑		
↑	high	Up-counting	
↓	low	Down-counting	

↑ : Rising edge

↓ : Falling edge

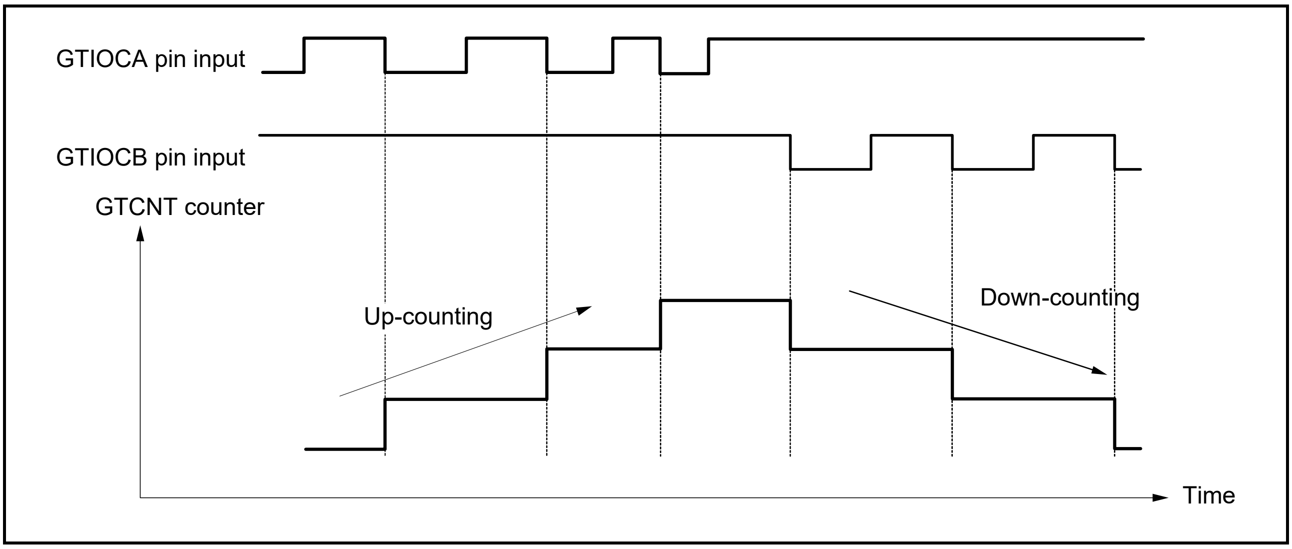


Figure 23.75 Example of phase counting mode 3 (A)

Table 23.12 Conditions of up-counting and down-counting in phase counting mode 3 (A)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Don’t care	GTUPSR = 0000 0800h GTDNSR = 0000 8000h
low	↓		
↑	low		
↓	high	Up-counting	
high	↓	Down-counting	
low	↑	Don’t care	
↑	high		
↓	low		

↑ : Rising edge

↓ : Falling edge

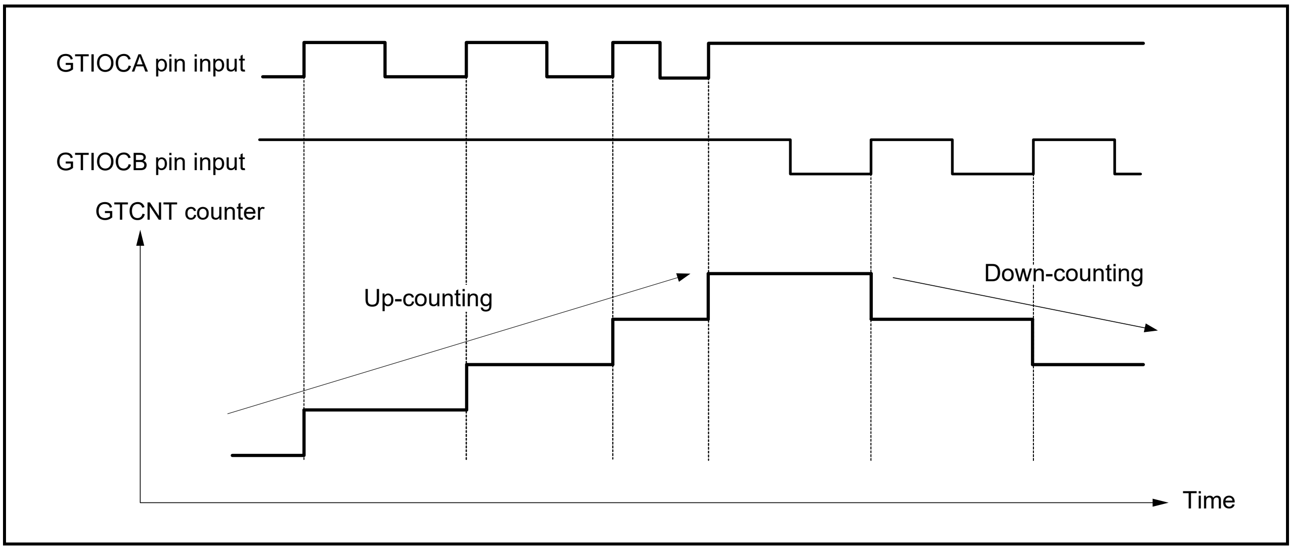


Figure 23.76 Example of phase counting mode 3 (B)

Table 23.13 Conditions of up-counting and down-counting in phase counting mode 3 (B)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Down-counting	GTUPSR = 0000 0200h GTDNSR = 0000 2000h
low	↓	Don’t care	
↑	low		
↓	high		
high	↓		
low	↑		
↑	high	Up-counting	
↓	low	Don’t care	

↑ : Rising edge

↓ : Falling edge

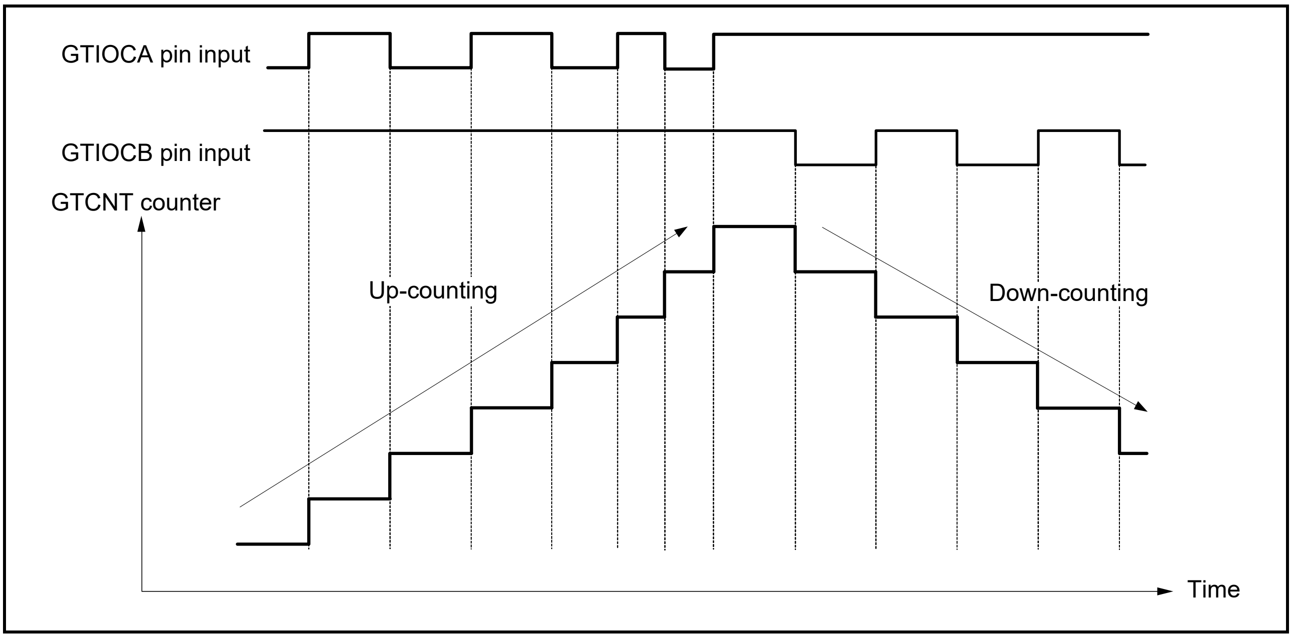


Figure 23.77 Example of phase counting mode 3 (C)

Table 23.14 Conditions of up-counting and down-counting in phase counting mode 3 (C)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Down-counting	GTUPSR = 0000 0A00h GTDNSR = 0000 A000h
low	↓	Don't care	
↑	low		
↓	high	Up-counting	
high	↓	Down-counting	
low	↑	Don't care	
↑	high	Up-counting	
↓	low	Don't care	

↑ : Rising edge

↓ : Falling edge

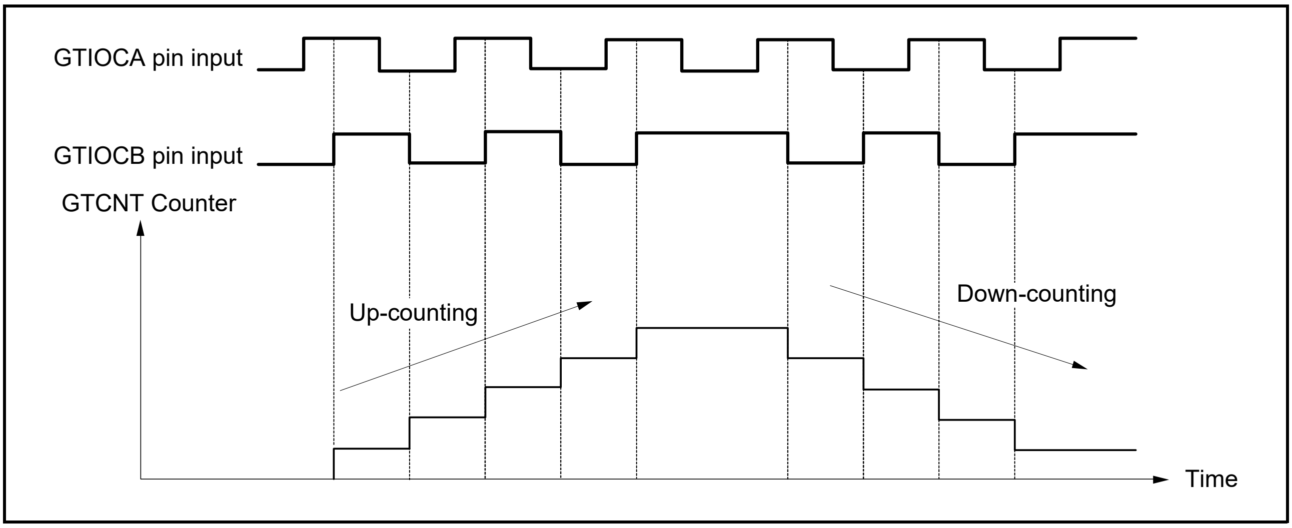


Figure 23.78 Example of phase counting mode 4

Table 23.15 Conditions of up-counting and down-counting in phase counting mode 4

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Up-counting	GTUPSR = 0000 6000h GTDNSR = 0000 9000h
low	↓		
↑	low	Don't care	
↓	high		
high	↓	Down-counting	
low	↑		
↑	high	Don't care	
↓	low		

↑ : Rising edge

↓ : Falling edge

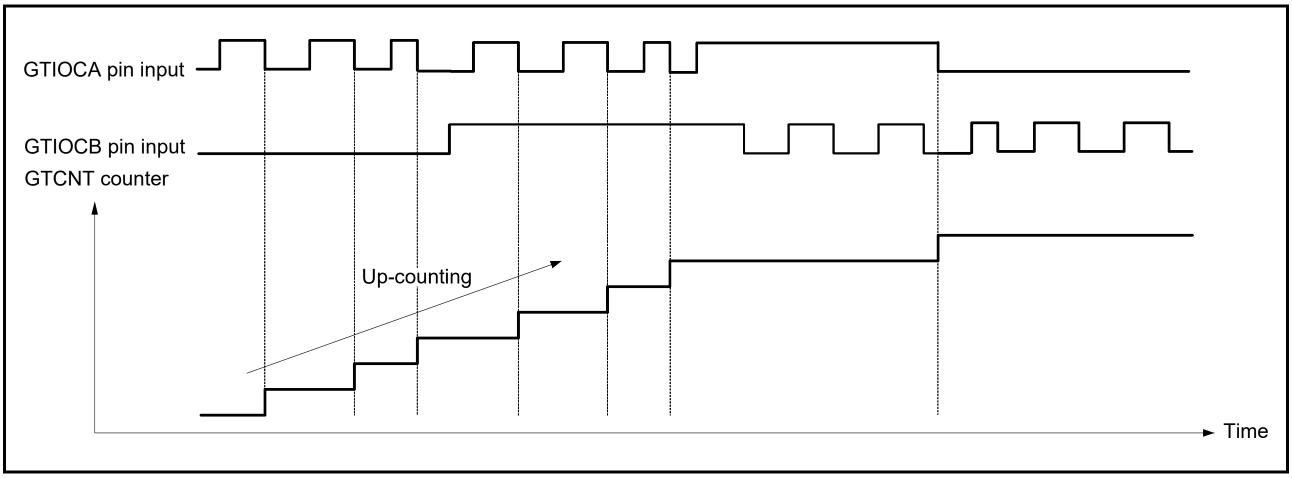


Figure 23.79 Example of phase counting mode 5 (A)

Table 23.16 Conditions of up-counting and down-counting in phase counting mode 5 (A)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Don't care	GTUPSR = 0000 0C00h GTDNSR = 0000 0000h
low	↓		
↑	low		
↓	high	Up-counting	
high	↓	Don't care	
low	↑		
↑	high		
↓	low	Up-counting	

↑ : Rising edge

↓ : Falling edge

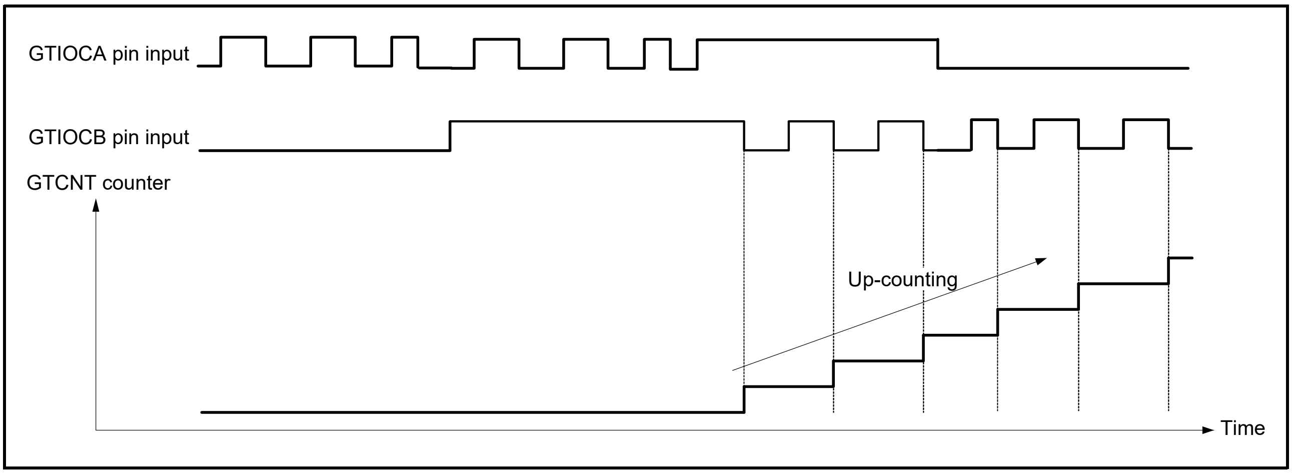


Figure 23.80 Example of phase counting mode 5 (B)

Table 23.17 Conditions of up-counting and down-counting in phase counting mode 5 (B)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
high	↑	Don’t care	GTUPSR = 0000 0C00h GTDNSR = 0000 0000h
low	↓	Up-counting	
↑	low	Don’t care	
↓	high		
high	↓	Up-counting	
low	↑	Don’t care	
↑	high		
↓	low		

↑ : Rising edge
↓ : Falling edge

23.3.11 Output Phase Switching (GPT_OPS)

GPT_OPS provides a function for easy control of brushless DC motor operation using the Output Phase Switching Control Register (OPSCR).

GPT_OPS outputs a PWM signal to be used for chopper control or level signal for each phase (U-positive phase/negative phase, V-positive phase/negative phase, W-positive phase/negative phase) of the 6-phase motor control. This function uses a soft setting value (OPSCR.UF, VF, WF) set by software or external signals detected by the Hall element, a PWM waveform of GPT32EH0.GTIOCA.

Figure 23.81 shows the GPT_OPS control flow conceptual diagram.

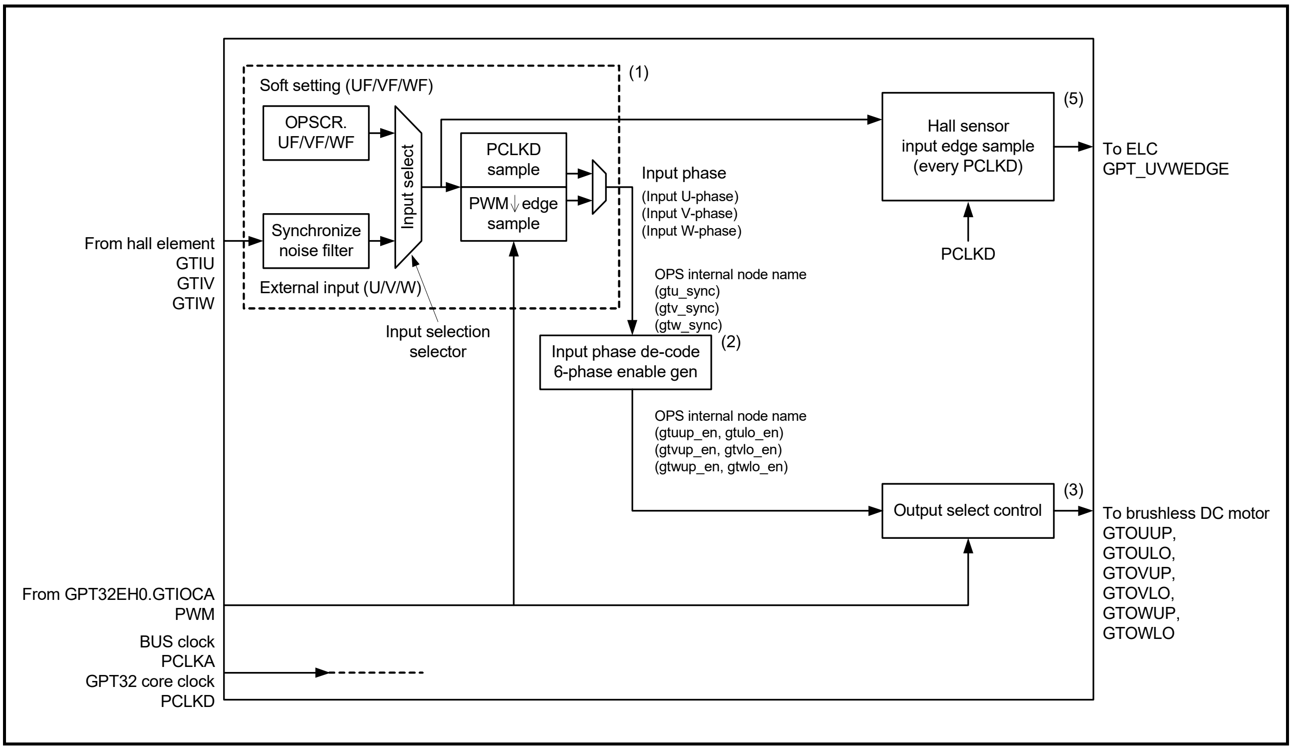
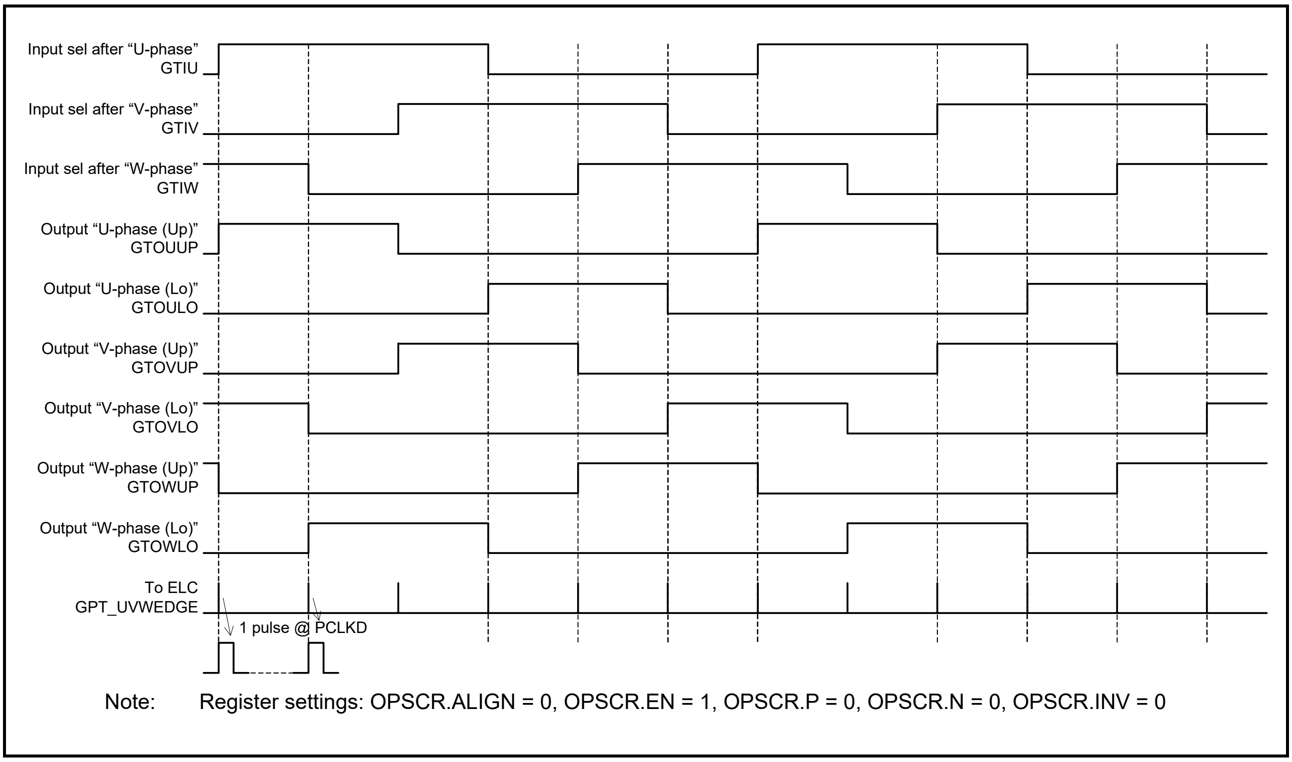


Figure 23.81 Conceptual diagram of GPT_OPS control flow

Figure 23.82 shows a 6-phase level signals output example of a GPT_OPS operation.

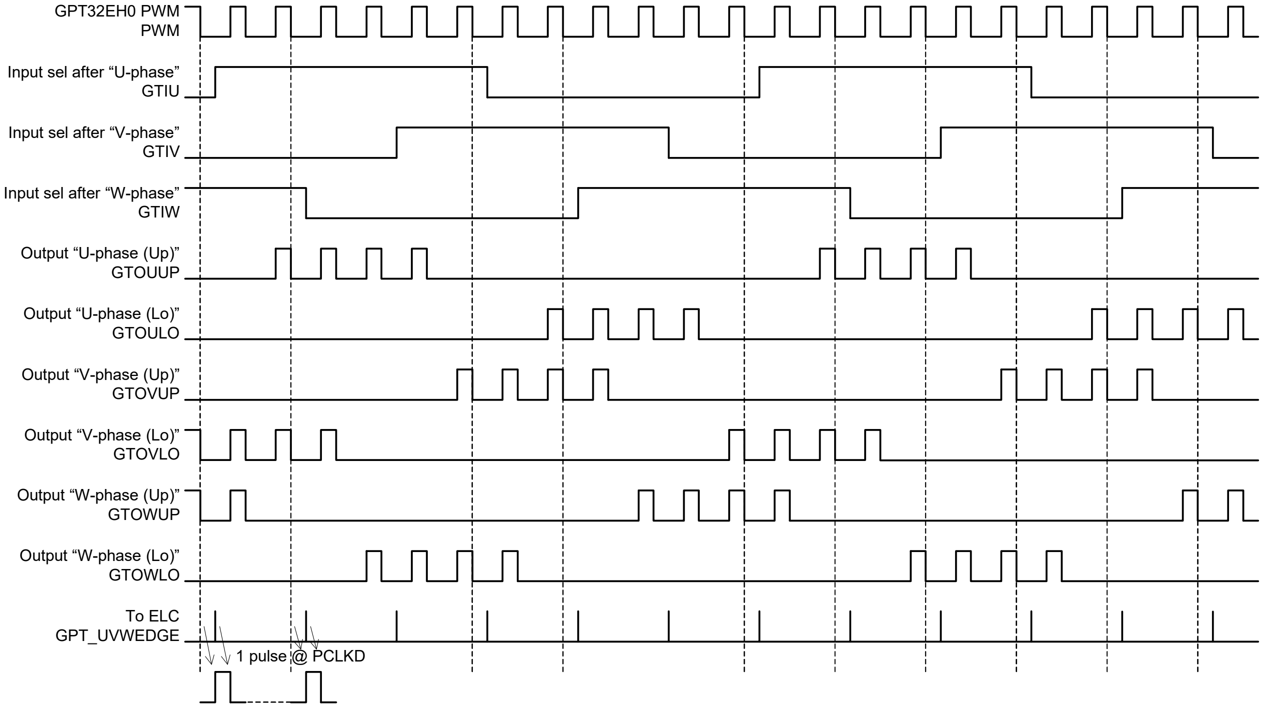
The GPT_UVWEDGE signal in Figure 23.82 is the Hall sensor input edge to ELC output.



Note: Register settings: OPSCR.ALIGN = 0, OPSCR.EN = 1, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0

Figure 23.82 Example of 6-phase level output operation

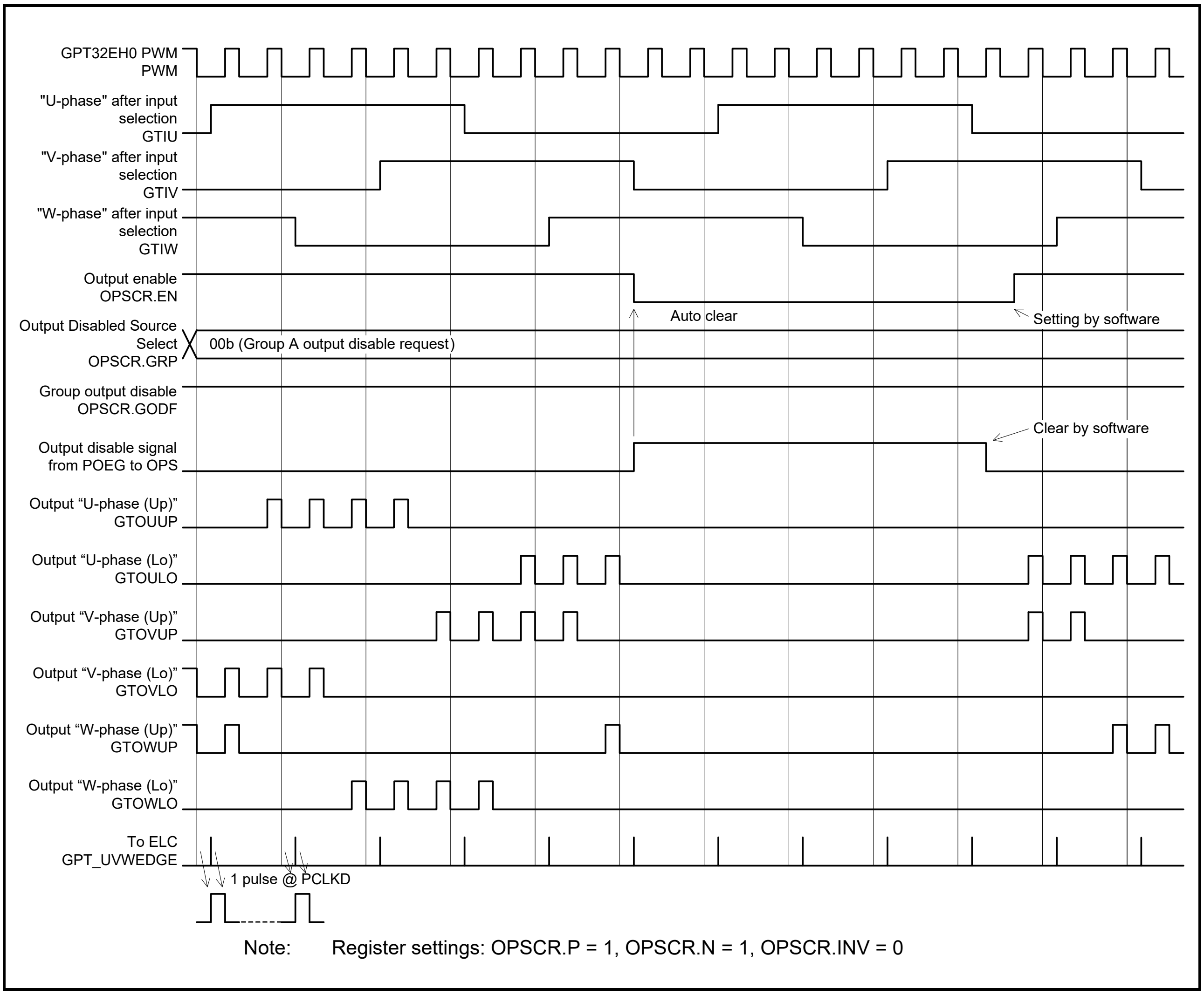
Figure 23.83 shows a 6-phase PWM output example of a GPT_OPS operation (chopper control).



Note: Register settings: OPSCR.ALIGN = 1, OPSCR.EN = 1, OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0

Figure 23.83 Example of 6-phase PWM output operation with chopper control

Figure 23.84 shows an example of output disable control (6-phase PWM output operation).



Note: Register settings: OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0

Figure 23.84 Example of group output disable control operation

23.3.11.1 Input selection and synchronization of external input signal

In the GPT_OPS control flow conceptual diagram shown in Figure 23.81, (1) is a selection of input phase from software settings and external input by the OPSCR.FB bit.

When OPSCR.FB bit = 0, select the external input. Enable the input signal after synchronization with the GPT core clock (PCLKD). After carrying out noise filtering (optional), set the external input to the input phase of PWM (PWM of GPT32EH0.GTIOCA) using falling edge sampling with OPSCR.ALIGN bit = 1.

When OPSCR.FB bit = 1, select the soft setting (OPSCR.UF, VF, WF) with the value of the input phase of PWM (PWM of GPT32EH0.GTIOCA) using falling edge sampling with OPSCR.ALIGN bit = 1.

When OPSCR.ALIGN bit = 0, GPT_OPS operates with the input phase of PCLKD synchronization with either OPSCR.FB bit = 0 or OPSCR.FB bit = 1. However, in some situations, the PWM pulse width of the output U/V/W phases (PWM output mode) of switch timing (just before or just after) is shortened.

Table 23.18 shows the input selection process and setting of associated OPSCR bits.

Table 23.18 Input selection processing method

OPSCR register		Selection of input phase sampling method (U/V/W-phase)	Synchronization input/output selection process (GPT_OPS internal node name)
FB bit	**ALIGN bit**		
0	1	External Input at PWM Falling Edge Sampling (PCLKD synchronization + falling edge sample)	"Input Phase" Input U-Phase (gtu_sync) Input V-Phase (gtv_sync) Input W-Phase (gtw_sync)
	0	External Input at PCLKD Synchronization Output (PCLKD synchronization + through mode)	
1	1	Software Settings at PWM Falling Edge Sampling (OPSCR.UF, VF, WF of falling edge sample)	
	0	Software Setting Value Selection (= OPSCR.UF/VF/WF value) (= PCLKD synchronization)	

23.3.11.2 Input sampling

The OPSCR.U, V, W bits indicate the PCLKD sampling results of the input selected by the OPSCR.FB bit.

When OPSCR.FB bit = 0 and after synchronization with the GPT core clock (PCLKD) and noise filtering (optional), OPSCR.U, V, W bits indicate the sampling results of the external input. When OPSCR.FB bit = 1, OPSCR.U, V, W bits have the value (OPSCR.UF, VF, WF) of the soft setting.

23.3.11.3 Input phase decode

In the GPT_OPS control flow conceptual diagram shown in Figure 23.81, (2) enables the 6-phase signals by decoding the input phase selected by the OPSCR.FB bit. The 6-phase enable signal is used for internal processing of GPT_OPS.

Table 23.19 shows the decode table of input phase.

Table 23.19 Decode table of input phase

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-phase	**Input V-phase**	**Input W-phase**	**U-phase (Up)**	**U-phase (Lo)**	**V-phase (Up)**	**V-phase (Lo)**	**W-phase (Up)**	**W-phase (Lo)**
(gtu_sync)	**(gtv_sync)**	**(gtw_sync)**	**(gtuup_en)**	**(gtulo_en)**	**(gtvup_en)**	**(gtvlo_en)**	**(gtwup_en)**	**(gtwlo_en)**
1	0	1	1	0	0	1	0	0
1	0	0	1	0	0	0	0	1
1	1	0	0	0	1	0	0	1
0	1	0	0	1	1	0	0	0
0	1	1	0	1	0	0	1	0
0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

23.3.11.4 Output selection control

In the GPT_OPS control flow conceptual diagram in Figure 23.81, (3) represents the selection of the output waveform by setting the OPSCR register bit.

For output selection, the following bits are relevant:

- The OPSCR.EN bit controls whether to output the 6-phase output, or to stop
- The OPSCR.P and OPSCR.N bits can select from the level signal or PWM signal (chopper output) for the output phase
- The polarity of the output phase can be set to positive logic or negative logic by the OPSCR.INV bit.

Table 23.20 and Table 23.21 show the output selection control method using the OPSCR register bit.

Table 23.20 Output selection control method (positive phase)

Enable-phase output control	Positive-phase output (P) control	Invert-phase output control	Output port name (positive phase = up) (output selection internal node allocation)	
OPSCR.EN bit	OPSCR.P bit	OPSCR.INV bit	GTOUUP GTOVUP GTOWUP	Mode
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS => 0 output
1	0	0	Level signal (gtuup_en) (gtvup_en) (gtwup_en)	Level Output Mode (Positive phase) (Positive logic)
1	0	1	Level signal (~gtuup_en) (~gtvup_en) (~gtwup_en)	Level Output Mode (Positive phase) (Negative logic)
1	1	0	PWM signal (PWM & gtuup_en) (PWM & gtvup_en) (PWM & gtwup_en)	PWM Output Mode (Positive phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtuup_en)) (~(PWM & gtvup_en)) (~(PWM & gtwup_en))	PWM Output Mode (Positive phase) (Negative logic)

Table 23.21 Output selection control method (negative phase)

Enable-phase output control	Negative-phase output (N) control	Invert-phase output control	Output port name (negative phase = Lo) (output selection internal node allocation)	
OPSCR.EN bit	OPSCR.N bit	OPSCR.INV bit	GTOULO GTOVLO GTOWLO	Mode
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS => 0 output
1	0	0	Level signal (gtulo_en) (gtvlo_en) (gtwlo_en)	Level Output Mode (Negative phase) (Positive logic)
1	0	1	Level signal (~gtulo_en) (~gtvlo_en) (~gtwlo_en)	Level Output Mode (Negative phase) (Negative logic)
1	1	0	PWM signal (PWM & gtulo_en) (PWM & gtvlo_en) (PWM & gtwlo_en)	PWM Output Mode (Negative phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtulo_en)) (~(PWM & gtvlo_en)) (~(PWM & gtwlo_en))	PWM Output Mode (Negative phase) (Negative logic)

23.3.11.5 Output selection control (group output disable function)

When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP bit is high (output disable request), the GPT_OPS output pins are changed to Hi-Z asynchronously and the OPSCR.EN bit is set to 0 by the output disable request signal synchronized with PCLKD. For the return, set the OPSCR.EN to 1 after clearing the output disable request with software.

The timing of OPSCR.EN bit cleared to 0 is 3 PCLKD cycles after generating the output disable request. To perform output disable control reliably, allow at least 4 PCLKD cycles after generating the output disable request (by clearing the output disable request flag in POEG) until the output disable request is terminated. For an example of the operation of group output disable control, see Figure 23.84.

23.3.11.6 Event Link Controller (ELC) output

In the GPT_OPS control flow conceptual diagram shown in Figure 23.81, (5) outputs the Hall sensor input signal edge to the ELC.

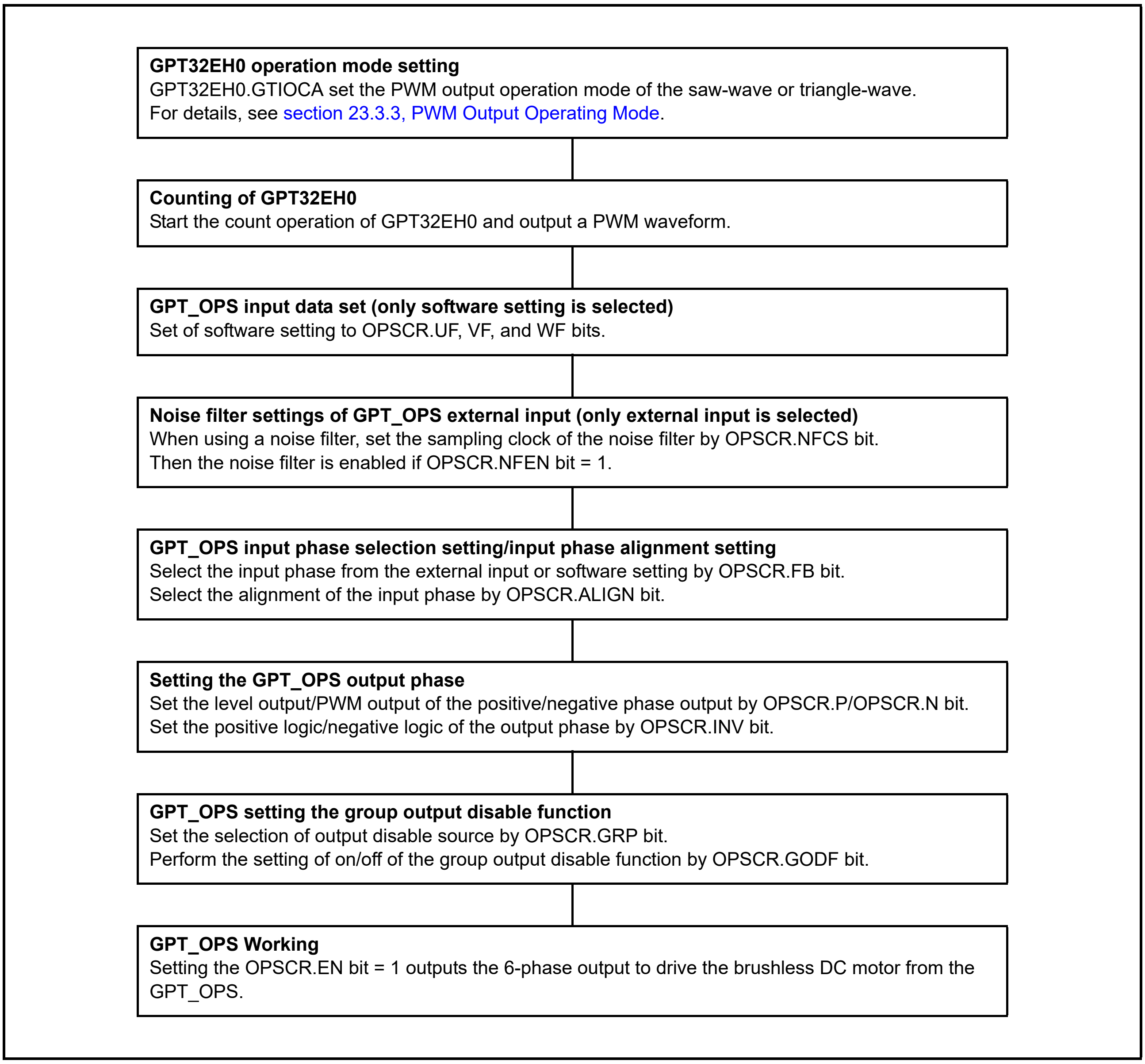
The Hall sensor input edge signal is the logical OR of the rising and falling edge signals of each U-phase/V-phase/W-phase input sampled at PCLKD. That is, if the high period of each of the U-phase/V-phase/W-phase input is short in duration, the Hall sensor edge input signal is not output at that time.

When OPSCR.FB bit = 0, the Hall sensor input edge signal is the logical OR of the edge signals of the external input phase sampled at PCLKD.

When OPSCR.FB bit = 1, the Hall sensor input edge signal is the logical OR of the edge of the soft setting (OPSCR.UF, VF, WF) sampled at PCLKD.

See Figure 23.82 to Figure 23.84 for examples of the output signal to the ELC.

23.3.11.7 GPT_OPS start operation setting flow



GPT32EH0 operation mode setting
GPT32EH0.GTIOCA set the PWM output operation mode of the saw-wave or triangle-wave.
For details, see section 23.3.3, PWM Output Operating Mode.

Counting of GPT32EH0
Start the count operation of GPT32EH0 and output a PWM waveform.

GPT_OPS input data set (only software setting is selected)
Set of software setting to OPSCR.UF, VF, and WF bits.

Noise filter settings of GPT_OPS external input (only external input is selected)
When using a noise filter, set the sampling clock of the noise filter by OPSCR.NFCS bit.
Then the noise filter is enabled if OPSCR.NFEN bit = 1.

GPT_OPS input phase selection setting/input phase alignment setting
Select the input phase from the external input or software setting by OPSCR.FB bit.
Select the alignment of the input phase by OPSCR.ALIGN bit.

Setting the GPT_OPS output phase
Set the level output/PWM output of the positive/negative phase output by OPSCR.P/OPSCR.N bit.
Set the positive logic/negative logic of the output phase by OPSCR.INV bit.

GPT_OPS setting the group output disable function
Set the selection of output disable source by OPSCR.GRP bit.
Perform the setting of on/off of the group output disable function by OPSCR.GODF bit.

GPT_OPS Working
Setting the OPSCR.EN bit = 1 outputs the 6-phase output to drive the brushless DC motor from the GPT_OPS.

Figure 23.85 Example setting of GPT_OPS start operation

23.4 Interrupt Sources

23.4.1 Overview

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTADTR compare match
- GTCNT counter overflow (GTPR compare match)/underflow.

Each interrupt source has its own status flag. When an interrupt source signal is generated, the associated status flag in GTST is set to 1. The associated status flag in GTST can be cleared by writing 0. If flag set and flag clear occur at the same time, flag clear takes priority over flag set. These flags are automatically updated by the internal state. Table 23.22 lists the GPT interrupt sources.

Table 23.22 Interrupt sources (1 of 4)

Channel	Name	Interrupt source	Interrupt flag	DMAC/DTC activation
0	GPT0_CCMPA	GPT32EH0.GTCCRA input capture/compare match	TCFA	Possible
	GPT0_CCMPB	GPT32EH0.GTCCRB input capture/compare match	TCFB	Possible
	GPT0_CMPC	GPT32EH0.GTCCRC compare match	TCFC	Possible
	GPT0_CMPD	GPT32EH0.GTCCRD compare match	TCFD	Possible
	GPT0_CMPE	GPT32EH0.GTCCRE compare match	TCFE	Possible
	GPT0_CMPF	GPT32EH0.GTCCRF compare match	TCFF	Possible
	GPT0_ADTRGA	GPT32EH0.GTADTRA compare match	ADTRAUF ADRTADF	Possible
	GPT0_ADTRGB	GPT32EH0.GTADTRB compare match	ADTRBUF ADRTBDF	Possible
	GPT0_OVF	GPT32EH0.GTCNT overflow (GPT32EH0.GTPR compare match)	TCFPO	Possible
	GPT0_UDF	GPT32EH0.GTCNT underflow	TCFPU	Possible
1	GPT1_CCMPA	GPT32EH1.GTCCRA input capture/compare match	TCFA	Possible
	GPT1_CCMPB	GPT32EH1.GTCCRB input capture/compare match	TCFB	Possible
	GPT1_CMPC	GPT32EH1.GTCCRC compare match	TCFC	Possible
	GPT1_CMPD	GPT32EH1.GTCCRD compare match	TCFD	Possible
	GPT1_CMPE	GPT32EH1.GTCCRE compare match	TCFE	Possible
	GPT1_CMPF	GPT32EH1.GTCCRF compare match	TCFF	Possible
	GPT1_ADTRGA	GPT32EH1.GTADTRA compare match	ADTRAUF ADRTADF	Possible
	GPT1_ADTRGB	GPT32EH1.GTADTRB compare match	ADTRBUF ADRTBDF	Possible
	GPT1_OVF	GPT32EH1.GTCNT overflow (GPT32EH1.GTPR compare match)	TCFPO	Possible
	GPT1_UDF	GPT32EH1.GTCNT underflow	TCFPU	Possible
2	GPT2_CCMPA	GPT32EH2.GTCCRA input capture/compare match	TCFA	Possible
	GPT2_CCMPB	GPT32EH2.GTCCRB input capture/compare match	TCFB	Possible
	GPT2_CMPC	GPT32EH2.GTCCRC compare match	TCFC	Possible
	GPT2_CMPD	GPT32EH2.GTCCRD compare match	TCFD	Possible
	GPT2_CMPE	GPT32EH2.GTCCRE compare match	TCFE	Possible
	GPT2_CMPF	GPT32EH2.GTCCRF compare match	TCFF	Possible
	GPT2_ADTRGA	GPT32EH2.GTCCRE compare match	ADTRAUF ADRTADF	Possible
	GPT2_ADTRGB	GPT32EH2.GTCCRF compare match	ADTRBUF ADRTBDF	Possible
	GPT2_OVF	GPT32EH2.GTCNT overflow (GPT32EH2.GTPR compare match)	TCFPO	Possible
	GPT2_UDF	GPT32EH2.GTCNT underflow	TCFPU	Possible

Table 23.22 Interrupt sources (2 of 4)

Channel	Name	Interrupt source	Interrupt flag	DMAC/DTC activation
3	GPT3_CCMPA	GPT32EH3.GTCCRA input capture/compare match	TCFA	Possible
	GPT3_CCMPB	GPT32EH3.GTCCRB input capture/compare match	TCFB	Possible
	GPT3_CMPC	GPT32EH3.GTCCRC compare match	TCFC	Possible
	GPT3_CMPD	GPT32EH3.GTCCRD compare match	TCFD	Possible
	GPT3_CMPE	GPT32EH3.GTCCRE compare match	TCFE	Possible
	GPT3_CMPF	GPT32EH3.GTCCRF compare match	TCFF	Possible
	GPT3_ADTRGA	GPT32EH3.GTADTRA compare match	ADTRAUF ADRTADF	Possible
	GPT3_ADTRGB	GPT32EH3.GTADTRB compare match	ADTRBUF ADRTBDF	Possible
	GPT3_OVF	GPT32EH3.GTCNT overflow (GPT32EH3.GTPR compare match)	TCFPO	Possible
	GPT3_UDF	GPT32EH3.GTCNT underflow	TCFPU	Possible
4	GPT4_CCMPA	GPT32E4.GTCCRA input capture/compare match	TCFA	Possible
	GPT4_CCMPB	GPT32E4.GTCCRB input capture/compare match	TCFB	Possible
	GPT4_CMPC	GPT32E4.GTCCRC compare match	TCFC	Possible
	GPT4_CMPD	GPT32E4.GTCCRD compare match	TCFD	Possible
	GPT4_CMPE	GPT32E4.GTCCRE compare match	TCFE	Possible
	GPT4_CMPF	GPT32E4.GTCCRF compare match	TCFF	Possible
	GPT4_ADTRGA	GPT32E4.GTADTRA compare match	ADTRAUF ADRTADF	Possible
	GPT4_ADTRGB	GPT32E4.GTADTRB compare match	ADTRBUF ADRTBDF	Possible
	GPT4_OVF	GPT32E4.GTCNT overflow (GPT32E4.GTPR compare match)	TCFPO	Possible
	GPT4_UDF	GPT32E4.GTCNT underflow	TCFPU	Possible
5	GPT5_CCMPA	GPT32E5.GTCCRA input capture/compare match	TCFA	Possible
	GPT5_CCMPB	GPT32E5.GTCCRB input capture/compare match	TCFB	Possible
	GPT5_CMPC	GPT32E5.GTCCRC compare match	TCFC	Possible
	GPT5_CMPD	GPT32E5.GTCCRD compare match	TCFD	Possible
	GPT5_CMPE	GPT32E5.GTCCRE compare match	TCFE	Possible
	GPT5_CMPF	GPT32E5.GTCCRF compare match	TCFF	Possible
	GPT5_ADTRGA	GPT32E5.GTADTRA compare match	ADTRAUF ADRTADF	Possible
	GPT5_ADTRGB	GPT32E5.GTADTRB compare match	ADTRBUF ADRTBDF	Possible
	GPT5_OVF	GPT32E5.GTCNT overflow (GPT32E5.GTPR compare match)	TCFPO	Possible
	GPT5_UDF	GPT32E5.GTCNT underflow	TCFPU	Possible

Table 23.22 Interrupt sources (3 of 4)

Channel	Name	Interrupt source	Interrupt flag	DMAC/DTC activation
6	GPT6_CCMPA	GPT32E6.GTCCRA input capture/compare match	TCFA	Possible
	GPT6_CCMPB	GPT32E6.GTCCRB input capture/compare match	TCFB	Possible
	GPT6_CMPC	GPT32E6.GTCCRC compare match	TCFC	Possible
	GPT6_CMPD	GPT32E6.GTCCRD compare match	TCFD	Possible
	GPT6_CMPE	GPT32E6.GTCCRE compare match	TCFE	Possible
	GPT6_CMPF	GPT32E6.GTCCRF compare match	TCFF	Possible
	GPT6_ADTRGA	GPT32E6.GTADTRA compare match	ADTRAUF ADRTADF	Possible
	GPT6_ADTRGB	GPT32E6.GTADTRB compare match	ADTRBUF ADRTBDF	Possible
	GPT6_OVF	GPT32E6.GTCNT overflow (GPT32E6.GTPR compare match)	TCFPO	Possible
	GPT6_UDF	GPT32E6.GTCNT underflow	TCFPU	Possible
7	GPT7_CCMPA	GPT32E7.GTCCRA input capture/compare match	TCFA	Possible
	GPT7_CCMPB	GPT32E7.GTCCRB input capture/compare match	TCFB	Possible
	GPT7_CMPC	GPT32E7.GTCCRC compare match	TCFC	Possible
	GPT7_CMPD	GPT32E7.GTCCRD compare match	TCFD	Possible
	GPT7_CMPE	GPT32E7.GTCCRE compare match	TCFE	Possible
	GPT7_CMPF	GPT32E7.GTCCRF compare match	TCFF	Possible
	GPT7_ADTRGA	GPT32E7.GTADTRA compare match	ADTRAUF ADRTADF	Possible
	GPT7_ADTRGB	GPT32E7.GTADTRB compare match	ADTRBUF ADRTBDF	Possible
	GPT7_OVF	GPT32E7.GTCNT overflow (GPT32E7.GTPR compare match)	TCFPO	Possible
	GPT7_UDF	GPT32E7.GTCNT underflow	TCFPU	Possible
8	GPT8_CCMPA	GPT328.GTCCRA input capture/compare match	TCFA	Possible
	GPT8_CCMPB	GPT328.GTCCRB input capture/compare match	TCFB	Possible
	GPT8_CMPC	GPT328.GTCCRC compare match	TCFC	Possible
	GPT8_CMPD	GPT328.GTCCRD compare match	TCFD	Possible
	GPT8_CMPE	GPT328.GTCCRE compare match	TCFE	Possible
	GPT8_CMPF	GPT328.GTCCRF compare match	TCFF	Possible
	GPT8_OVF	GPT328.GTCNT overflow (GPT328.GTPR compare match)	TCFPO	Possible
	GPT8_UDF	GPT328.GTCNT underflow	TCFPU	Possible
9	GPT9_CCMPA	GPT329.GTCCRA input capture/compare match	TCFA	Possible
	GPT9_CCMPB	GPT329.GTCCRB input capture/compare match	TCFB	Possible
	GPT9_CMPC	GPT329.GTCCRC compare match	TCFC	Possible
	GPT9_CMPD	GPT329.GTCCRD compare match	TCFD	Possible
	GPT9_CMPE	GPT329.GTCCRE compare match	TCFE	Possible
	GPT9_CMPF	GPT329.GTCCRF compare match	TCFF	Possible
	GPT9_OVF	GPT329.GTCNT overflow (GPT329.GTPR compare match)	TCFPO	Possible
	GPT9_UDF	GPT329.GTCNT underflow	TCFPU	Possible

Table 23.22 Interrupt sources (4 of 4)

Channel	Name	Interrupt source	Interrupt flag	DMAC/DTC activation
10	GPT10_CCMPA	GPT3210.GTCCRA input capture/compare match	TCFA	Possible
	GPT10_CCMPB	GPT3210.GTCCRB input capture/compare match	TCFB	Possible
	GPT10_CMPC	GPT3210.GTCCRC compare match	TCFC	Possible
	GPT10_CMPD	GPT3210.GTCCRD compare match	TCFD	Possible
	GPT10_CMPE	GPT3210.GTCCRE compare match	TCFE	Possible
	GPT10_CMPF	GPT3210.GTCCRF compare match	TCFF	Possible
	GPT10_OVF	GPT3210.GTCNT overflow (GPT3210.GTPR compare match)	TCFPO	Possible
	GPT10_UDF	GPT3210.GTCNT underflow	TCFPU	Possible
11	GPT11_CCMPA	GPT3211.GTCCRA input capture/compare match	TCFA	Possible
	GPT11_CCMPB	GPT3211.GTCCRB input capture/compare match	TCFB	Possible
	GPT11_CMPC	GPT3211.GTCCRC compare match	TCFC	Possible
	GPT11_CMPD	GPT3211.GTCCRD compare match	TCFD	Possible
	GPT11_CMPE	GPT3211.GTCCRE compare match	TCFE	Possible
	GPT11_CMPF	GPT3211.GTCCRF compare match	TCFF	Possible
	GPT11_OVF	GPT3211.GTCNT overflow (GPT3211.GTPR compare match)	TCFPO	Possible
	GPT11_UDF	GPT3211.GTCNT underflow	TCFPU	Possible
12	GPT12_CCMPA	GPT3212.GTCCRA input capture/compare match	TCFA	Possible
	GPT12_CCMPB	GPT3212.GTCCRB input capture/compare match	TCFB	Possible
	GPT12_CMPC	GPT3212.GTCCRC compare match	TCFC	Possible
	GPT12_CMPD	GPT3212.GTCCRD compare match	TCFD	Possible
	GPT12_CMPE	GPT3212.GTCCRE compare match	TCFE	Possible
	GPT12_CMPF	GPT3212.GTCCRF compare match	TCFF	Possible
	GPT12_OVF	GPT3212.GTCNT overflow (GPT3212.GTPR compare match)	TCFPO	Possible
	GPT12_UDF	GPT3212.GTCNT underflow	TCFPU	Possible
13	GPT13_CCMPA	GPT3213.GTCCRA input capture/compare match	TCFA	Possible
	GPT13_CCMPB	GPT3213.GTCCRB input capture/compare match	TCFB	Possible
	GPT13_CMPC	GPT3213.GTCCRC compare match	TCFC	Possible
	GPT13_CMPD	GPT3213.GTCCRD compare match	TCFD	Possible
	GPT13_CMPE	GPT3213.GTCCRE compare match	TCFE	Possible
	GPT13_CMPF	GPT3213.GTCCRF compare match	TCFF	Possible
	GPT13_OVF	GPT3213.GTCNT overflow (GPT3213.GTPR compare match)	TCFPO	Possible
	GPT13_UDF	GPT3213.GTCNT underflow	TCFPU	Possible

(1) GPTn_ADTRGA interrupt (n = 0 to 7)

When the GTCNT counter value matches with the GTADTRA register, an interrupt request is generated under the following conditions:

- In up-counting, the interrupt enable bit (ADTRAUEN) in the GTINTAD register is 1
- In down-counting, the interrupt enable bit (ADTRADEN) in the GTINTAD register is 1.

In event count operation, this interrupt request is not generated.

(2) GPTn_ADTRGB interrupt (n = 0 to 7)

When the GTCNT counter value matches with the GTADTRB register, an interrupt request is generated under the following conditions:

- In up-counting, the interrupt enable bit (ADTRBUEN) in the GTINTAD register is 1
- In down-counting, the interrupt enable bit (ADTRBDEN) in the GTINTAD register is 1.

In event count operation, this interrupt request is not generated.

(3) GPTn_CCMPA interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input capture signal causes transfer of the GTCNT counter value to the GTCCRA register.

(4) GPTn_CCMPB interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register
- When the GTCCRB register functions as an input capture register, the input capture signal causes transfer of the GTCNT counter value to the GTCCRB register.

(5) GPTn_CMPC interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value matches with the GTCCRC register.

A compare match is not performed and an interrupt is not requested under the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRC register).

(6) GPTn_CMPD interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value matches with the GTCCRD register.

A compare match is not performed and an interrupt is not requested under the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register).

(7) GPTn_CMPE interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value matches with the GTCCRE register.

A compare match is not performed and an interrupt is not requested under the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRE register).

(8) GPTn_CMPF interrupt (n = 0 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value matches with the GTCCRF register.

A compare match is not performed and an interrupt is not requested under the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register).

(9) GPTn_OVF interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- In saw-wave mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from GTPR to 0 during up-counting)
- In triangle-wave mode, interrupt requests are enabled at crests (GTCNT changes from GTPR to GTPR-1)
- In counting by hardware sources, overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

(10) GPTn_UDF interrupt (n = 0 to 13)

An interrupt request is generated under the following conditions:

- In saw-wave mode, interrupt requests are enabled at underflows (when the GTCNT counter value changes from 0 to GTPR during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (GTCNT changes from 0 to 1).
- In counting by hardware sources, underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

Table 23.23 Interrupt signals, interrupt permission bits, and interrupt status flags

Interrupt signal	Interrupt permission bit	Interrupt status flag
GPTn_UDF	— *1	GTST[7] (TCFPU)
GPTn_OVF		GTST[6] (TCFPO)
GPTn_ADTRGB	GTINTAD[19] (ADTRBDEN) GTINTAD[18] (ADTRBUEN)	GTST[19] (ADTRBDF) GTST[18] (ADTRBUF)
GPTn_ADTRGA	GTINTAD[17] (ADTRADEN) GTINTAD[16] (ADTRAUEN)	GTST[17] (ADTRADF) GTST[16] (ADTRAUF)
GPTn_CMPF	— *1	GTST[5] (TCFF)
GPTn_CMPE		GTST[4] (TCFE)
GPTn_CMPD		GTST[3] (TCFD)
GPTn_CMPC		GTST[2] (TCFC)
GPTn_CCMPB		GTST[1] (TCFB)
GPTn_CCMPA		GTST[0] (TCFA)

Note 1. Interrupt is always permitted.

23.4.2 DMAC/DTC Activation

The DMAC and DTC can be activated by the interrupt in each channel. For details, see section 14, Interrupt Controller Unit (ICU), and section 18, Data Transfer Controller (DTC).

23.4.3 Interrupt and A/D Conversion Request Skipping Function

By setting the GTITC register, the GTCNT counter overflow (GTPR compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF) can be skipped. Other interrupts and A/D converter start request signals can be skipped in coordination with the GPTn_OVF/GPTn_UDF skipping function.

The interrupt request skipping function only depends on the setting of GTITC register and is independent of the setting of interrupt permission bits in the GTINTAD register.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GPTn_OVF/GPTn_UDF interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. To count both troughs and crests and generate the GPTn_OVF/GPTn_UDF interrupts at troughs only or crests only in triangle-wave mode, you must set an even number of skips.

Similarly, in saw-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GPTn_OVF interrupt requests cannot be generated on either overflows or underflows only. To count both overflows and underflows with the count direction changed and generate the GPTn_OVF/GPTn_UDF interrupts on either overflows or underflows only in saw wave mode, you must first check the skipping state.

Before changing the skipping count, you must release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 23.86 to Figure 23.91 show examples of skipping function operation.

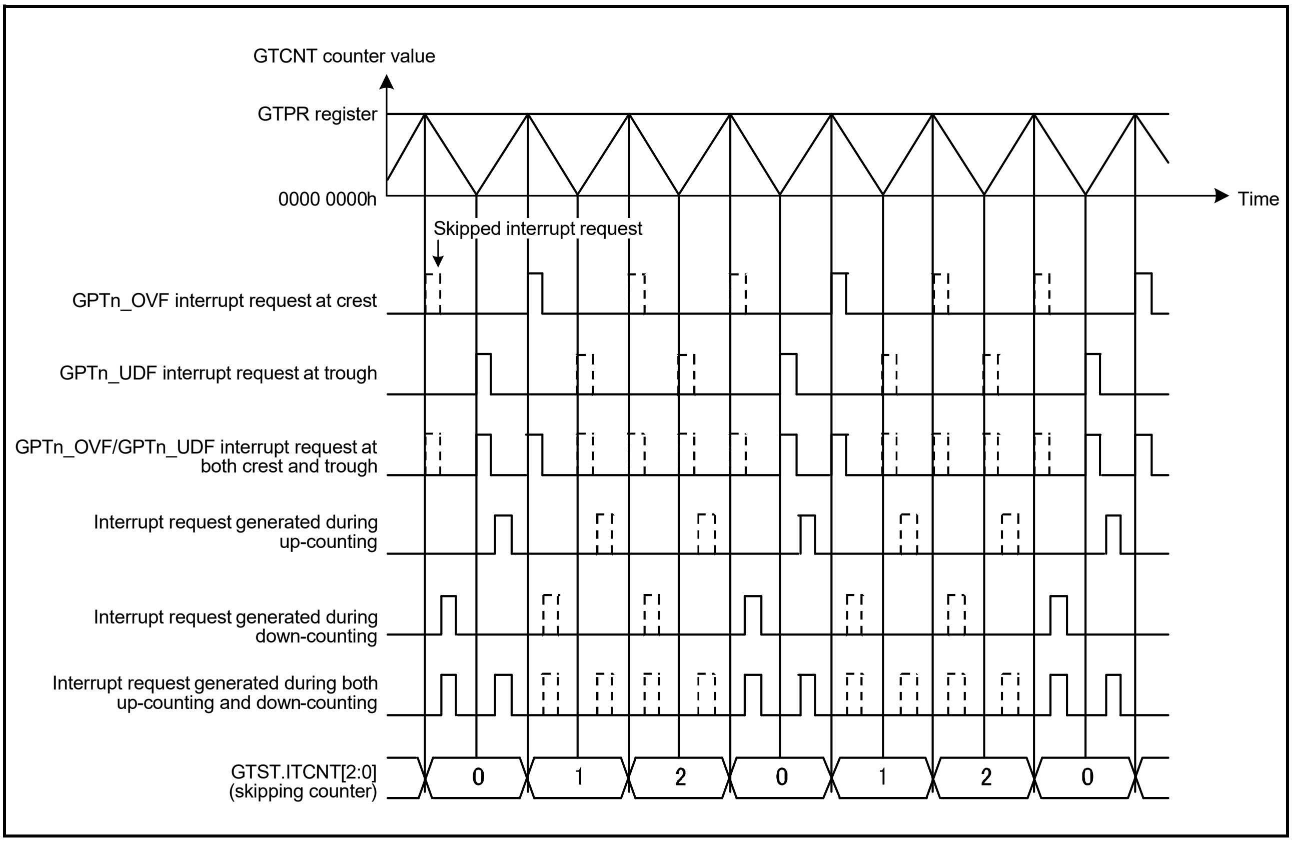


Figure 23.86 Example of interrupt skipping function operation with triangle waves, counting and skipping crests, and skipping count = 2

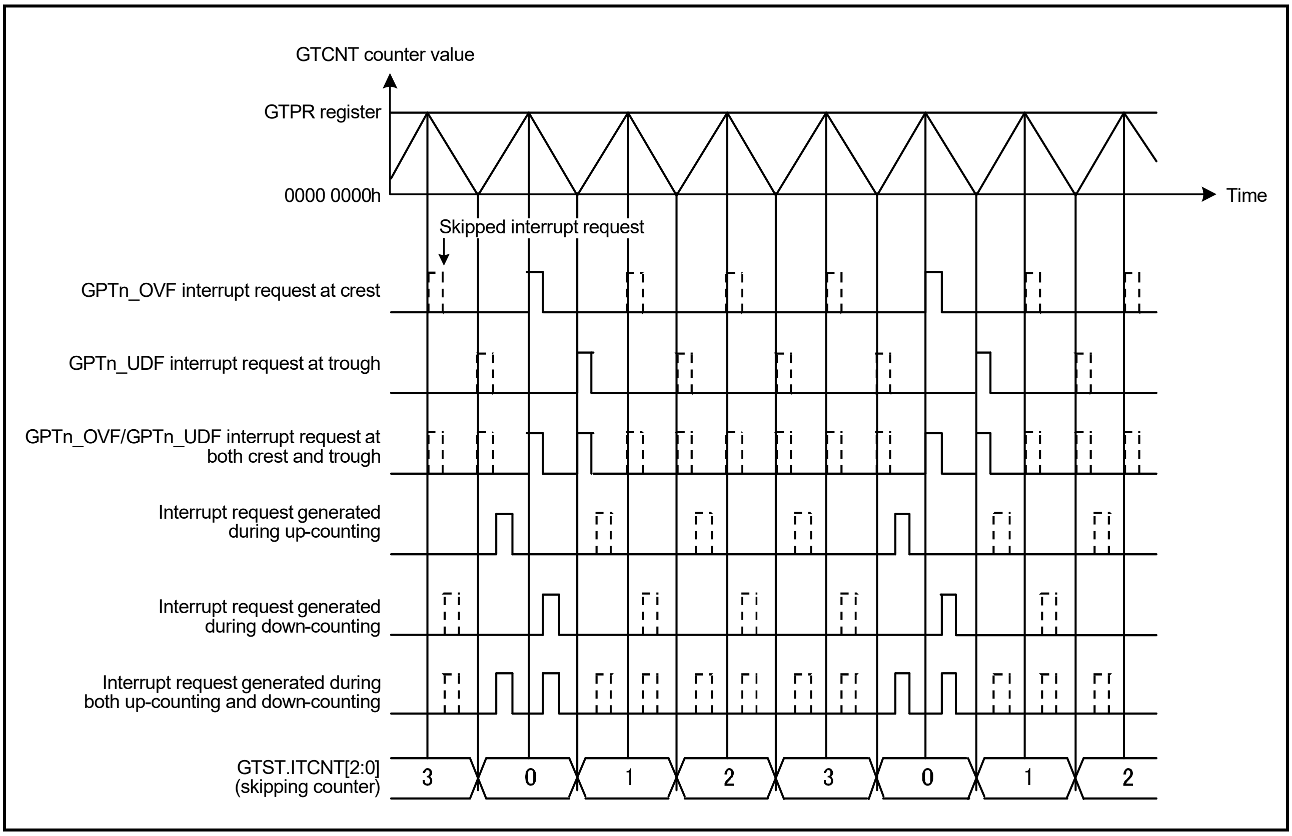


Figure 23.87 Example of interrupt skipping function operation with triangle waves, counting and skipping troughs, and skipping count = 3

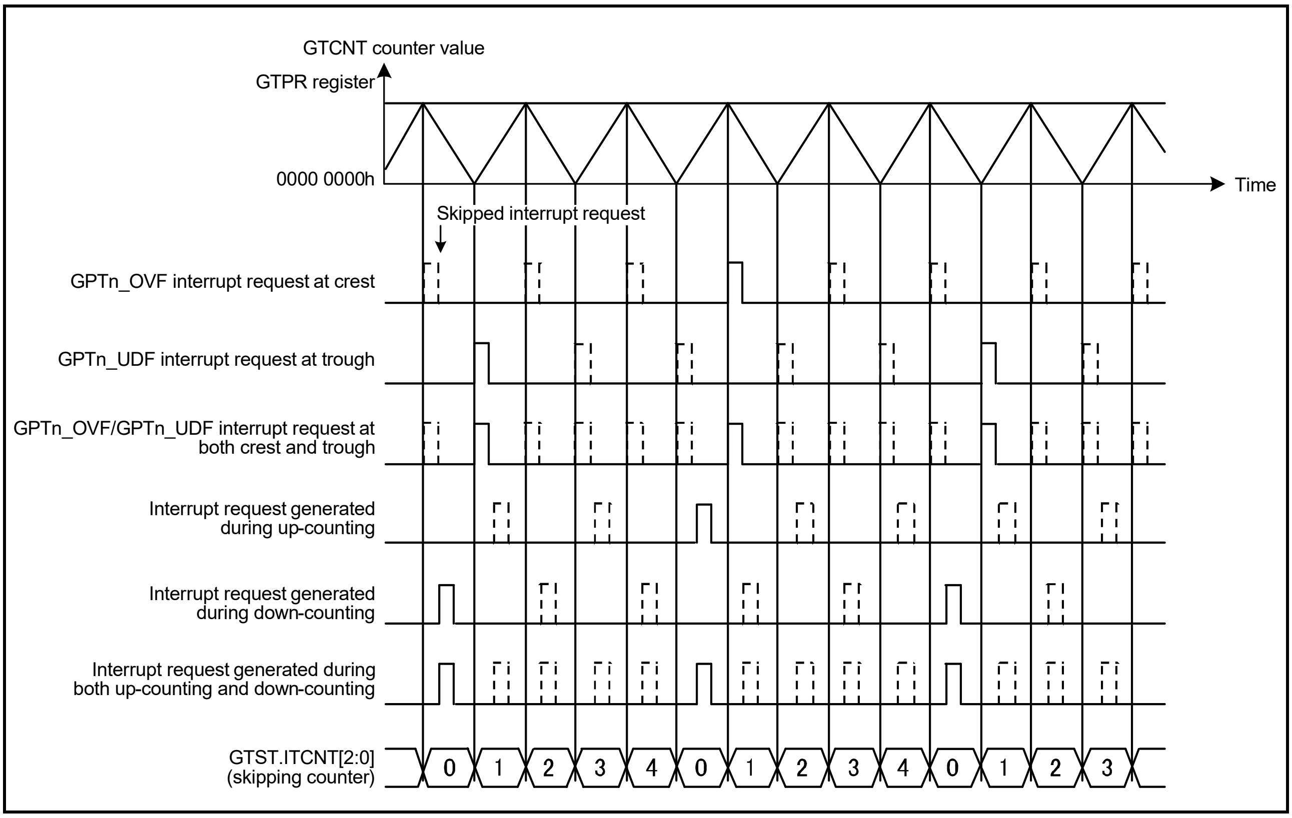


Figure 23.88 Example of interrupt skipping function operation with triangle waves, counting and skipping both troughs and crests, and skipping count = 4

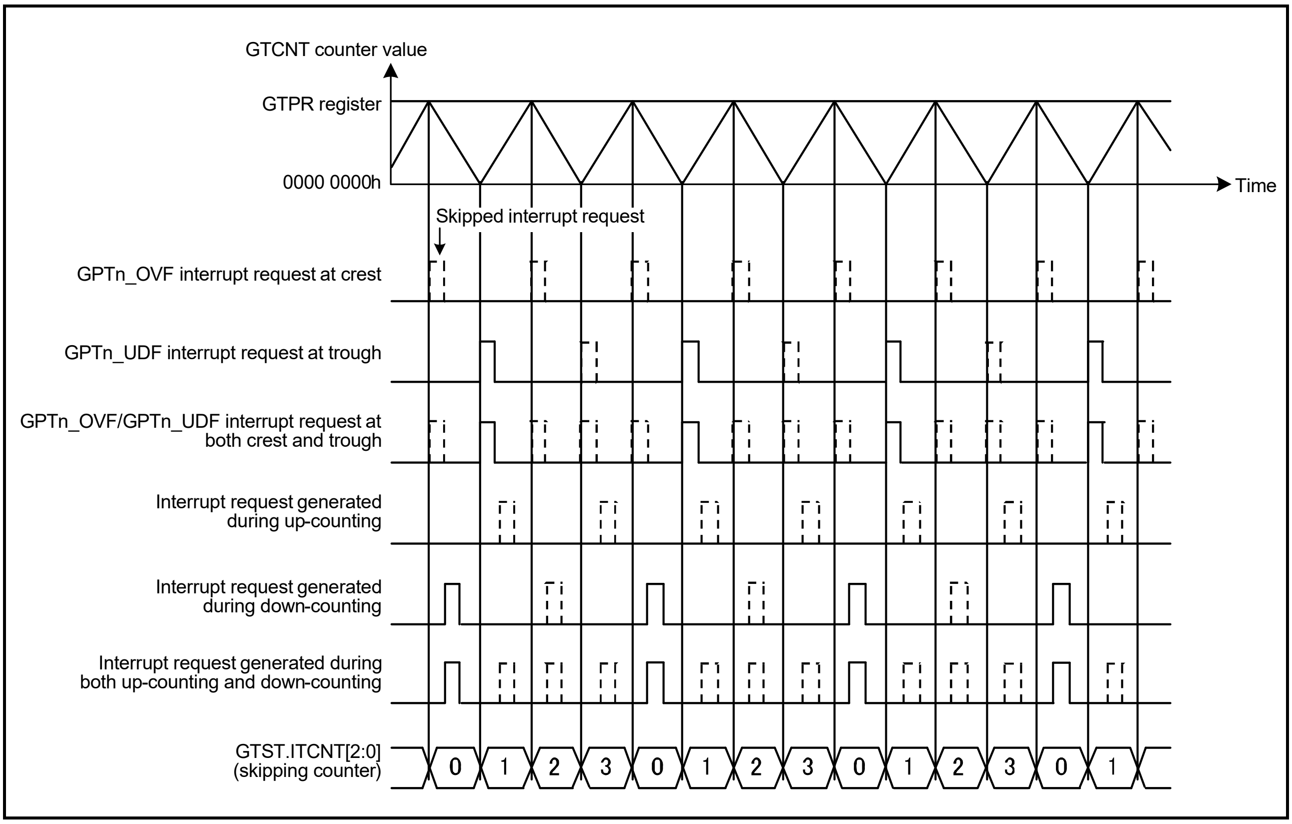


Figure 23.89 Example of interrupt skipping function operation with triangle waves, counting and skipping both troughs and crests, skipping count = 3, and skipping started at up-counting

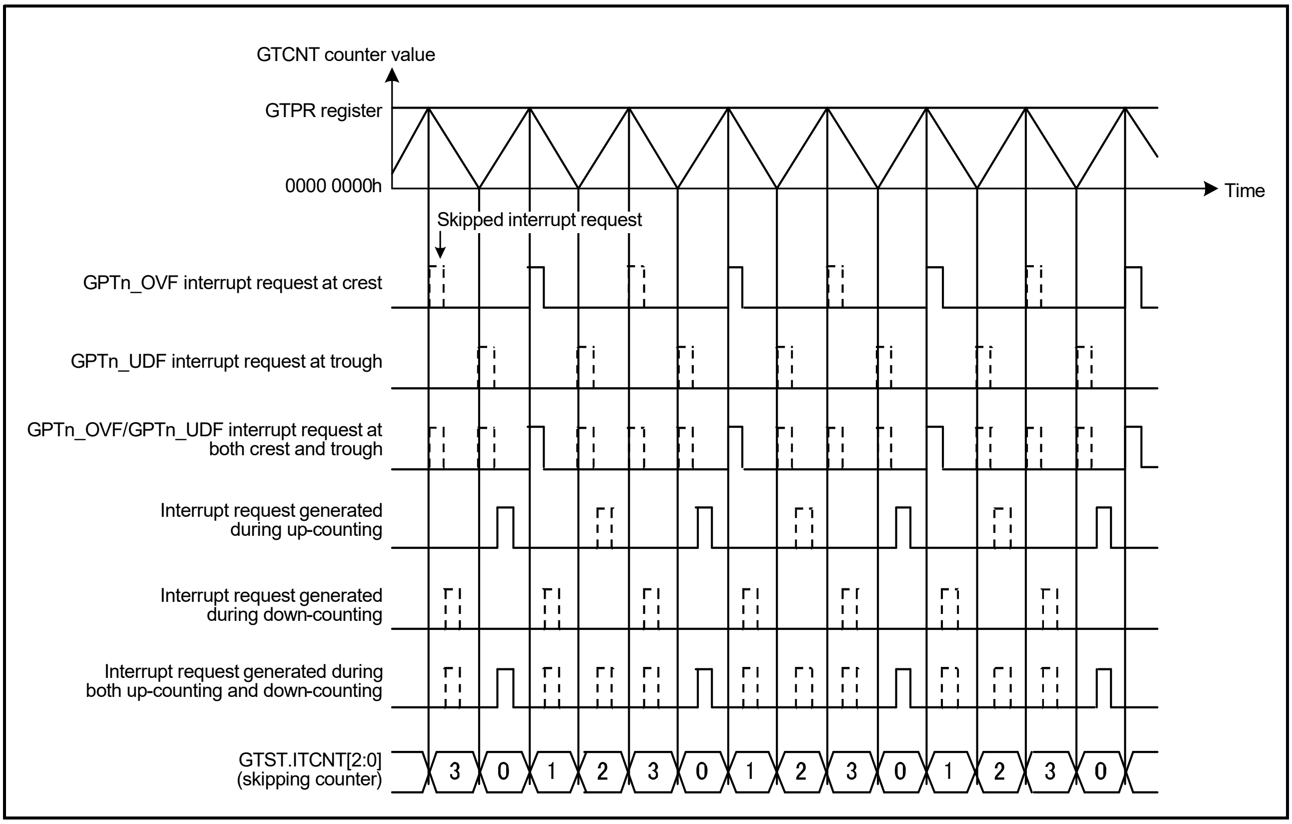


Figure 23.90 Example of interrupt skipping function operation with triangle waves, counting and skipping both troughs and crests, skipping count = 3, and skipping started at down-counting

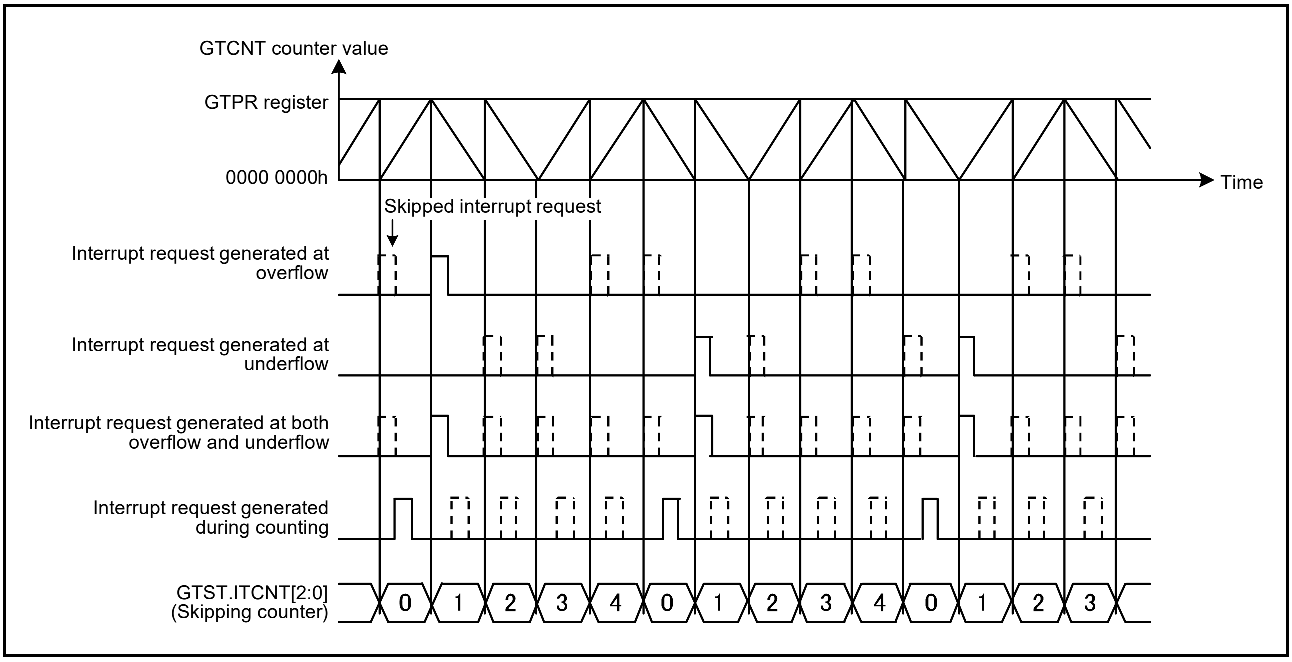


Figure 23.91 Example of interrupt skipping function operation with saw waves, operation with count direction changed, counting and skipping both overflows and underflows, and skipping count = 4

23.5 A/D Converter Start Request

An A/D converter start request can be issued at a compare match between the GTCNT counter and GTADTRA or GTADTRB, and up-counting only, down-counting only, or both up-counting and down-counting can be specified.

In event count operation, A/D converter start requests interrupt cannot be generated. An A/D converter start request does not direct output to the A/D converter module but results in output to ELC as event signals.

GTADTRA and GTADTRB each have two buffer registers. Buffer operation with GTADTRA combined with GTADTBRA and GTADTDBRA, and buffer operation with GTADTRB combined with GTADTBRB and GTADTDBRB can be performed.

Figure 23.92 shows an example of A/D converter start request operation, and Figure 23.93 shows an example setting for A/D converter start request operation.

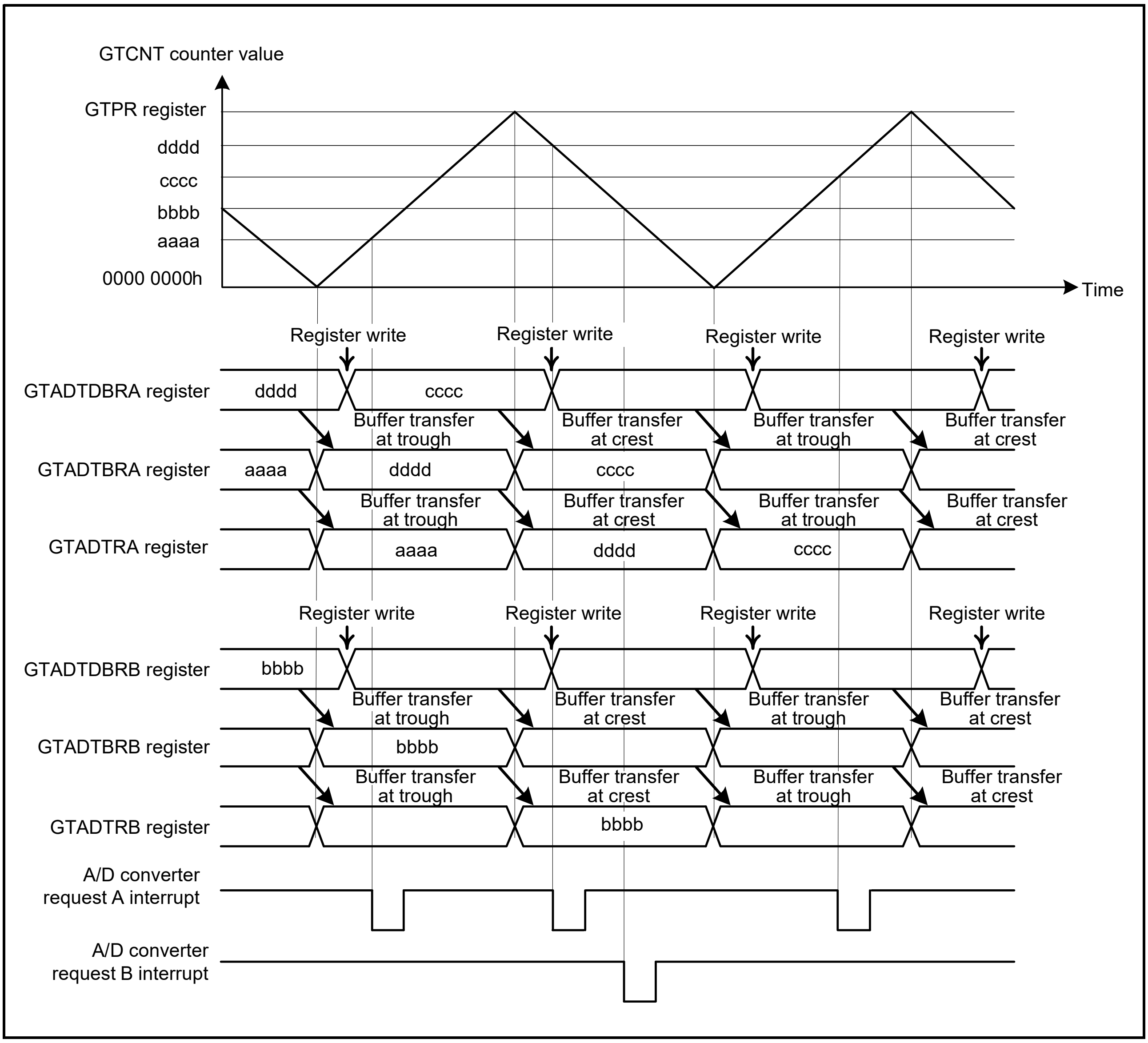
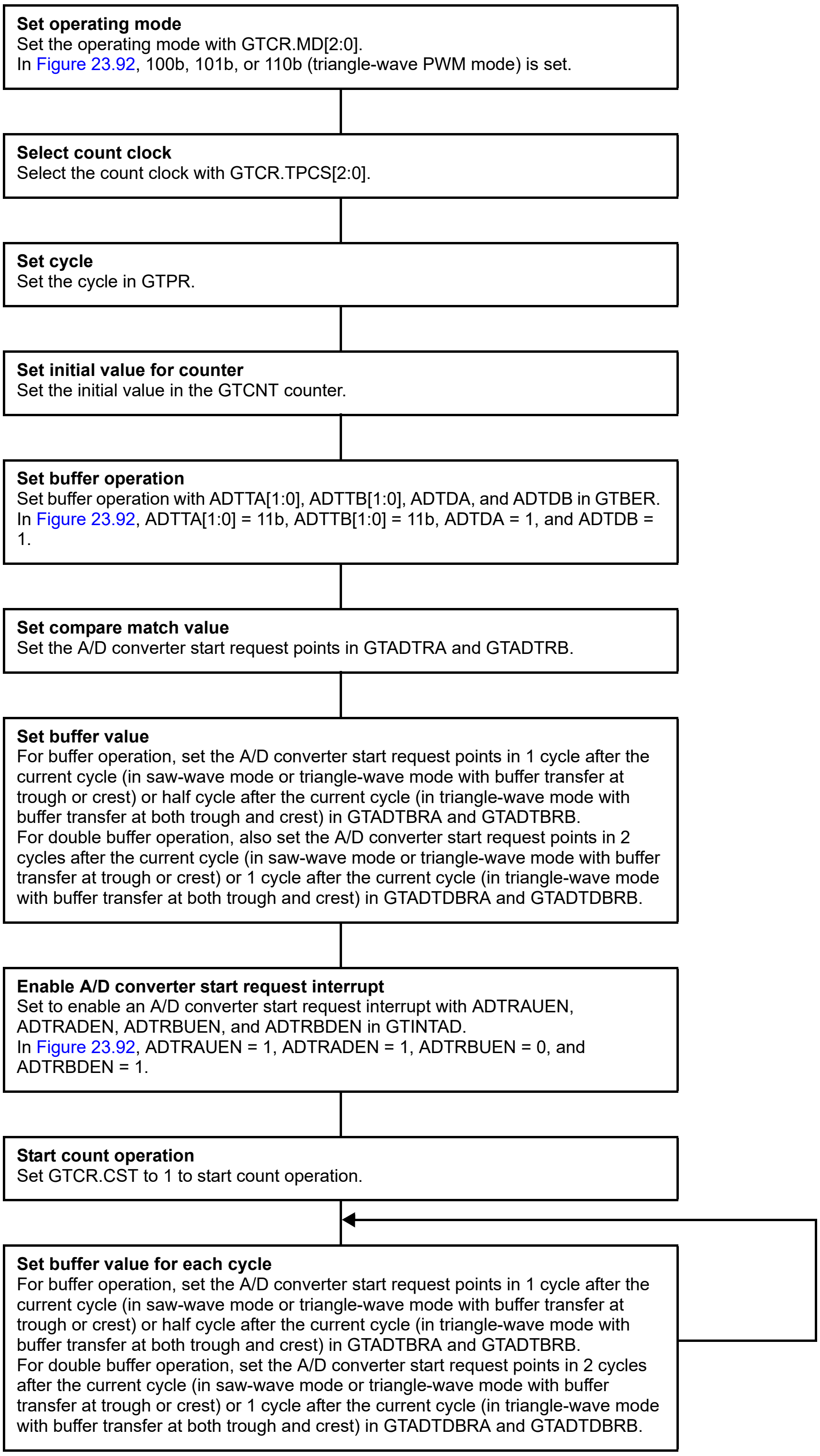


Figure 23.92 Example of A/D converter start request timing operation with triangle waves, double buffer operation, buffer transfer at both troughs and crests, A/D converter start request interrupt by GTADTRA at both up-counting and down-counting, and A/D converter start request interrupt by GTADTRB at down-counting



Set operating mode
Set the operating mode with GTCR.MD[2:0].
In Figure 23.92, 100b, 101b, or 110b (triangle-wave PWM mode) is set.

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.

Set buffer operation
Set buffer operation with ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB in GTBER.
In Figure 23.92, ADTTA[1:0] = 11b, ADTTB[1:0] = 11b, ADTDA = 1, and ADTDB = 1.

Set compare match value
Set the A/D converter start request points in GTADTRA and GTADTRB.

Set buffer value
For buffer operation, set the A/D converter start request points in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTBRA and GTADTBRB.
For double buffer operation, also set the A/D converter start request points in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTDBRA and GTADTDBRB.

Enable A/D converter start request interrupt
Set to enable an A/D converter start request interrupt with ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN in GTINTAD.
In Figure 23.92, ADTRAUEN = 1, ADTRADEN = 1, ADTRBUEN = 0, and ADTRBDEN = 1.

Start count operation
Set GTCR.CST to 1 to start count operation.

Set buffer value for each cycle
For buffer operation, set the A/D converter start request points in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTBRA and GTADTBRB.
For double buffer operation, set the A/D converter start request points in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in GTADTDBRA and GTADTDBRB.

Figure 23.93 Example setting for A/D converter start request timing operation

23.6 Operations Linked by the ELC

23.6.1 Event Signal Output to the ELC

The GPT can perform operation linked with another module set in advance when its interrupt request signal is used as an event signal by the ELC.

A/D converter start requests can be enabled and disabled individually with each up-counting and down-counting for both interrupts and events output to ELC by enable bits of the interrupt request.

The GPT has the following ELC event signals:

- Generating of compare match A interrupt (GPTn_CCMPA (n = 0 to 13))
- Generating of compare match B interrupt (GPTn_CCMPB (n = 0 to 13))
- Generating of compare match C interrupt (GPTn_CMPC (n = 0 to 13))
- Generating of compare match D interrupt (GPTn_CMPD (n = 0 to 13))
- Generating of compare match E interrupt (GPTn_CMPE (n = 0 to 13))
- Generating of compare match F interrupt (GPTn_CMPF (n = 0 to 13))
- Generating of overflow interrupt (GPTn_OVF (n = 0 to 13))
- Generating of underflow interrupt (GPTn_UDF (n = 0 to 13))
- A/D converter start request A interrupt (GPTn_ADTRGA (n = 0 to 7))
- A/D converter start request B interrupt (GPTn_ADTRGB (n = 0 to 7)).

23.6.2 Event Signal Inputs from the ELC

The GPT can perform the following operations in response to a maximum of eight events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down counting
- Input capture.

See section 23.3, Operation for detail on hardware resources.

23.7 Noise Filter Function

Each pin for use in input capture and Hall sensor input to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses whose length is less than three sampling cycles.

The noise filter functionality includes enabling and disabling the noise filter for each pin and setting of the sampling clock for each channel.

Figure 23.94 shows the timing of noise filtering.

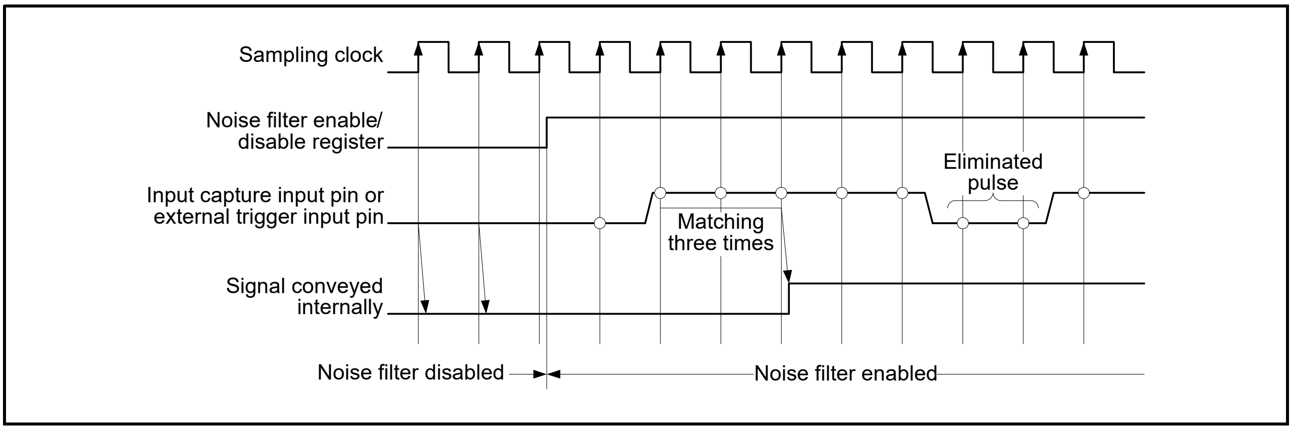


Figure 23.94 Timing of noise filtering

If noise filtering is enabled, the input capture operation or external trigger operation performs on the edges of the noise filtered signal after a delay of a sampling interval × 3 + PCLKD. This is caused by the noise filtering for the input capture input or external trigger operation.

23.8 Protection Function

23.8.1 Write-Protection for Registers

To prevent registers from being accidentally modified, registers can be write-protected in channel units by setting GTWP.WP. Write-protection can be set for the following registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD,GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR.

23.8.2 Disabling of Buffer Operation

If the timing of buffer register write is delayed in relative to the timing for the buffer transfer, buffer operation can be suspended with the GTBER.BD setting. Buffer transfer can be temporarily disabled even when a buffer transfer condition is generated during a buffer register write. This can be done by setting the associated GTBER.BD bit to 1 (buffer operation disabled) before a buffer register write and clearing the bit to 0 (buffer operation enabled) after completion of writing to all buffer registers. Figure 23.95 shows an example of operation for disabling buffer operation.

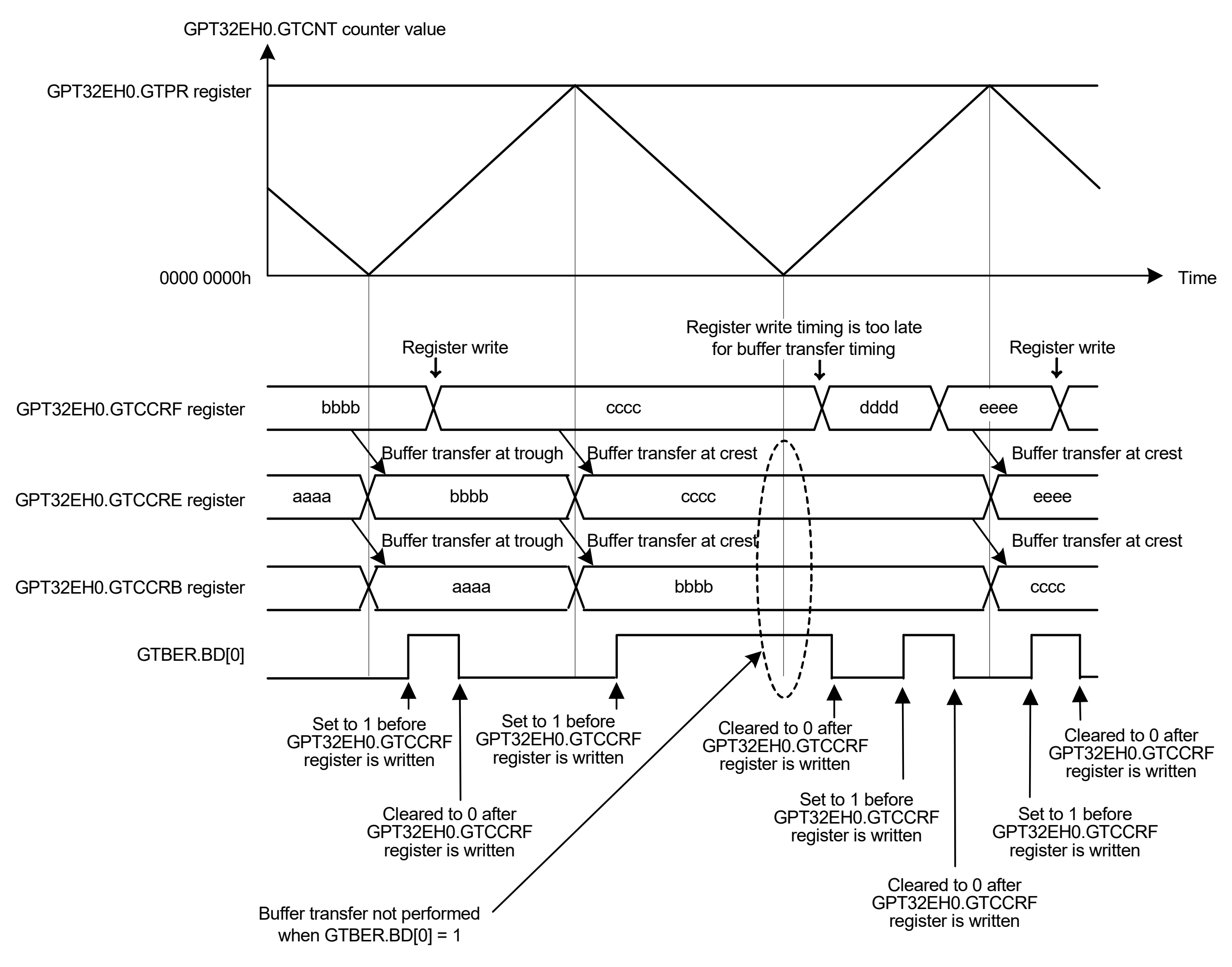


Figure 23.95 **Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests**

23.8.3 GTIOC Pin Output Negate Control

For protection from system failure, the output disable control that changes the GTIOC pin output value forcibly is provided for GTIOC pin output by the request of output disable from POEG.

When dead time error occurs or the GTIOCA pin output value is the same as the GTIOCB pin output value, output protection is required. The GPT detects this condition and generates output disable requests to POEG based on the settings in the output disable request permission bits, such as GTINTAD.GRPDTE, GTINTAD.GRPABH, and GTINTAD.GRPABL. After the POEG receives output disable requests from each channel and calculates external input using an OR operation, the POEG generates output disable requests to GPT.

One output disable signal (representing the shared output disable request signal of the GTIOCA pin and the GTIOCB pin) out of four output disable requests generated by the POEG is selected by setting GTINTAD.GRP[1:0]. The status of the selected disable output request is monitored by reading the GTST.ODF bit. The output level during output disable is based on the GTIOR.OADF[1:0] setting for the GTIOCA pin and the GTIOR.OBDF[1:0] setting for the GTIOCB pin.

The change to the output disable state is performed asynchronously by generating the output disable request from the POEG. The release of the output disable state is performed at end of cycle by terminating the output disable request. The timing of release of the output disable state is a minimum of 3 PCLKD cycles after terminating the output disable request. To perform output disable control reliably, allow at least 4 PCLKD cycles after generating the output disable request (by clearing the output disable request flag in POEG) until the output disable request is terminated.

When event count is performed or when the output disable state is to be released immediately without waiting for an end of cycle, GTIOR.OADF[1:0] must be set to 00b (for GTIOCA pin) or GTIOR.OBDF[1:0] must be set to 00b (for GTIOCB pin).

Figure 23.96 shows an example of the GTIOC pin output disable control operation.

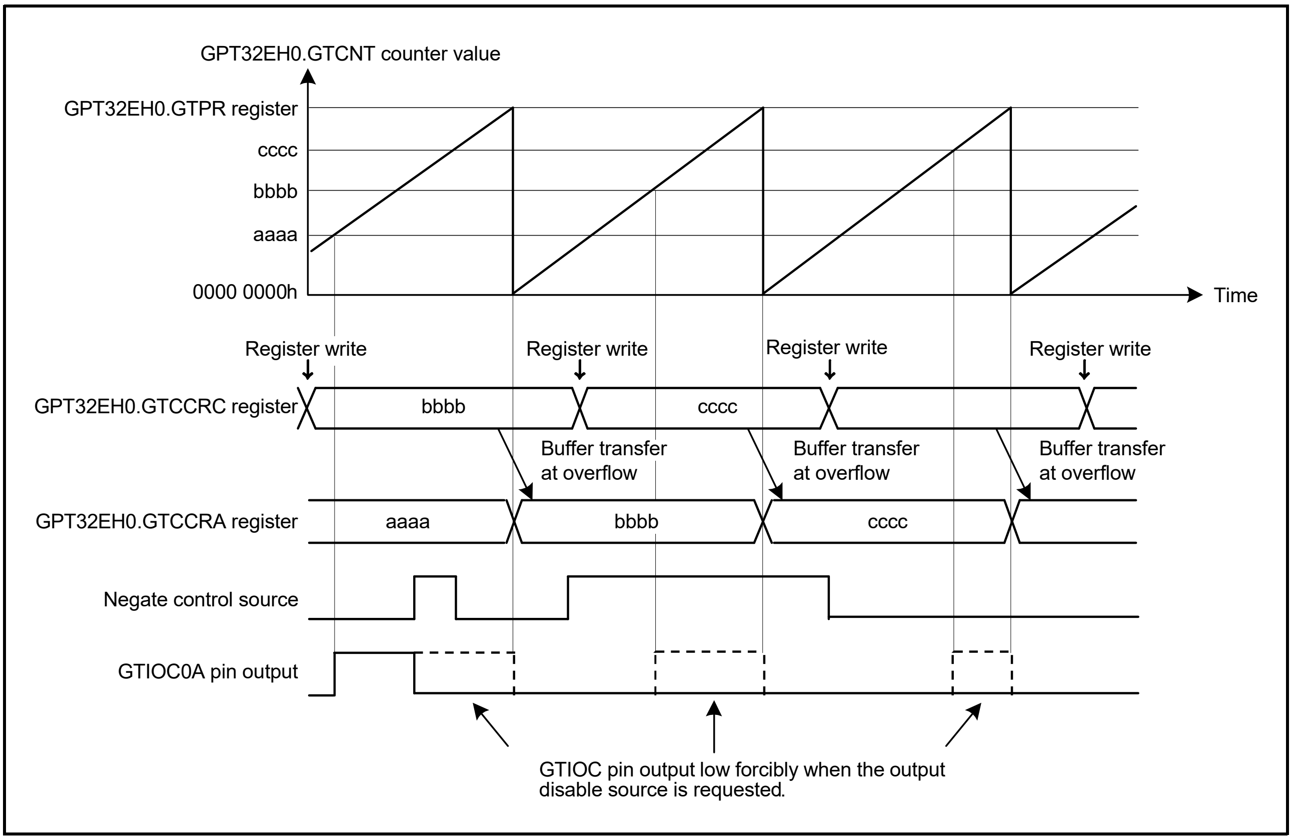


Figure 23.96 **Example of GTIOC pin output disable control operation with saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable**

23.8.4 Output Protection Function for GTIOC Pin Output

In preparation for incorrect settings of the GTCCRA register (settings outside the range of 0 < GTCCRA < GTPR), the output protection function for the GTIOC pin output (disabling function) is activated when the automatic dead time setting (GTDTCR.TDE = 1) is made in triangle-wave mode. The status of the output protection function can be read from GTSOS.SOS[1:0].

Figure 23.97 shows the output protection function state transition.

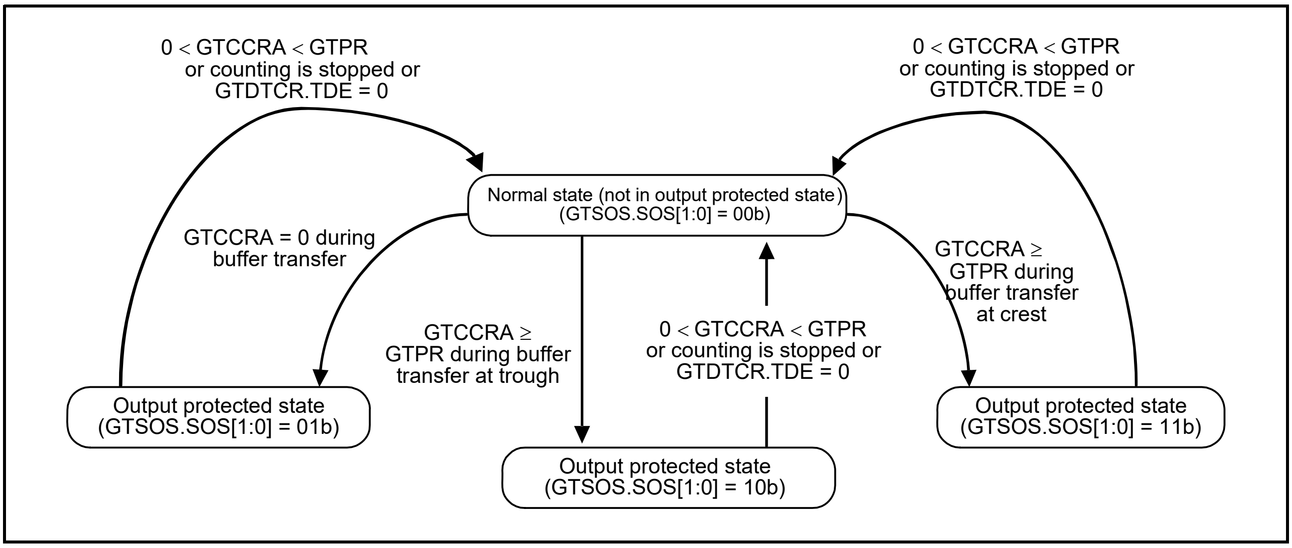
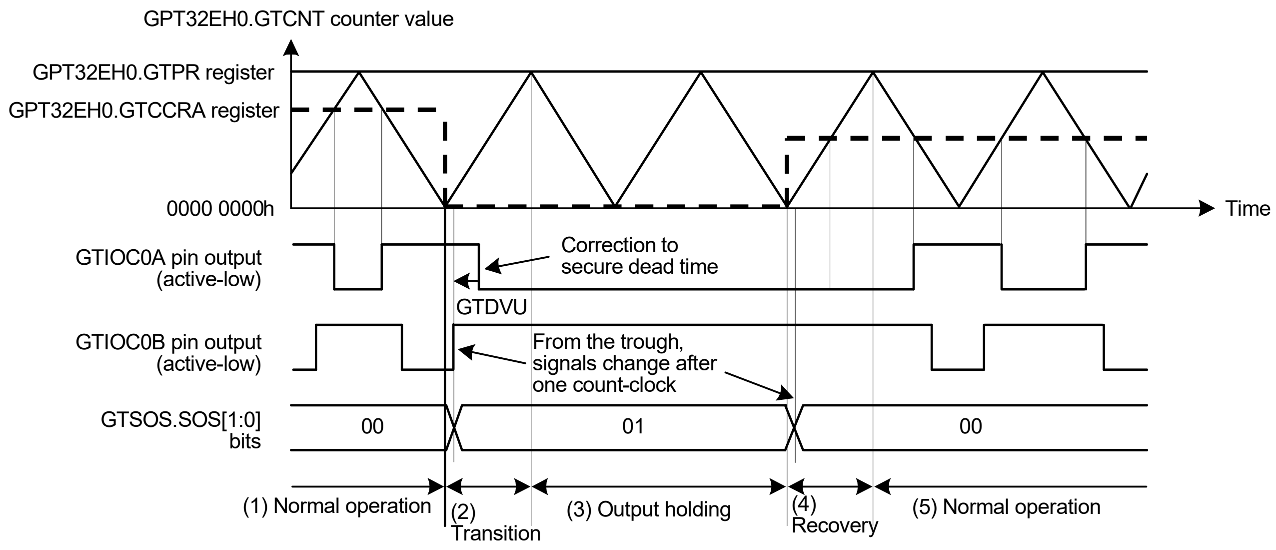


Figure 23.97 **Output protection function**

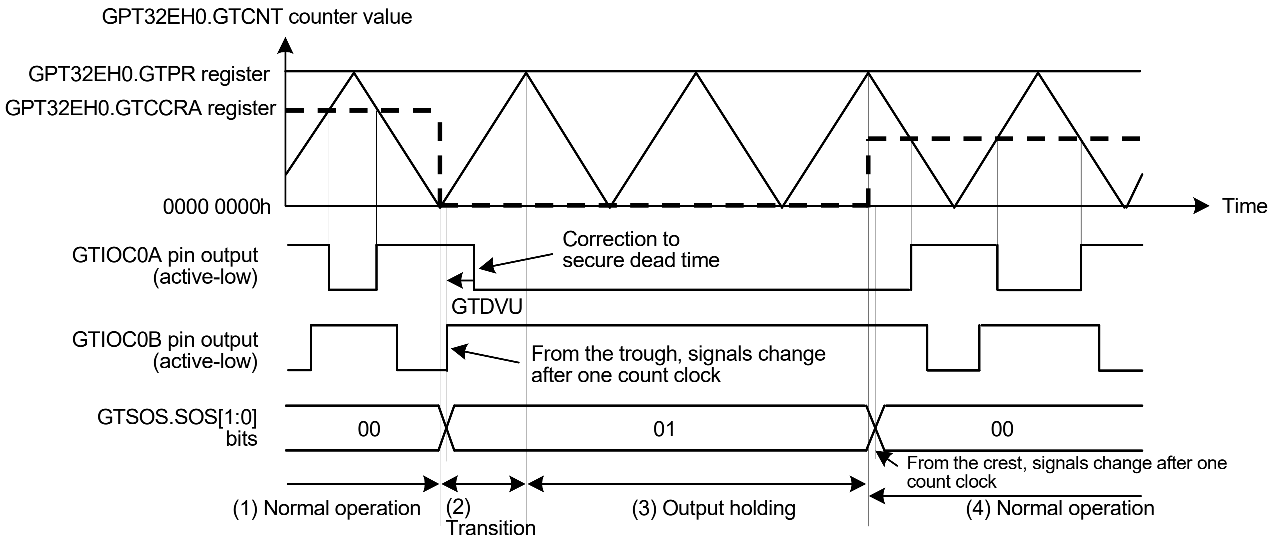
23.8.4.1 Output protection function when the GTCCRA register is set to 0 during buffer transfer

Figure 23.98 and Figure 23.99 show examples of output protection function operation when the GTCCRA register is set to 0 during buffer transfer at troughs, and Figure 23.100 and Figure 23.101 show examples when the GTCCRA register is set to 0 during buffer transfer at crests.



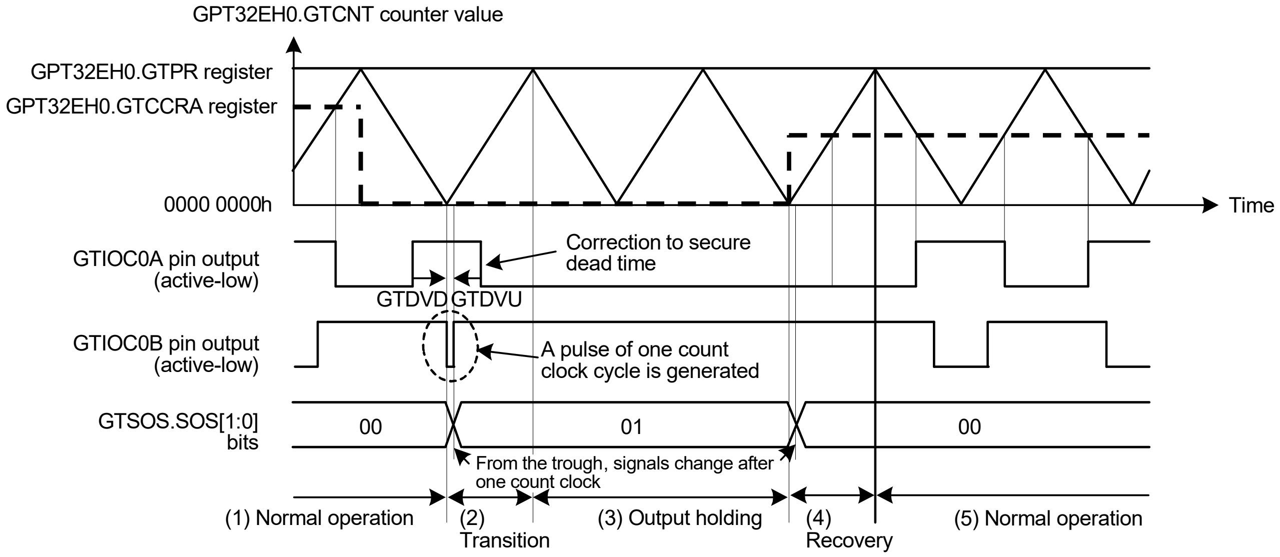
(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Transition period:	When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest
(3) Output holding period:	The output is held while GTCCRA register = 0.
(4) Recovery period:	When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period:	Toggle output due to a compare match is performed normally.

Figure 23.98 **Example of output protection operation when GTCCRA is set to 0 during buffer transfer at troughs, with 0 < GTCCRA < GTPR restored during buffer transfer at troughs, and active-low**



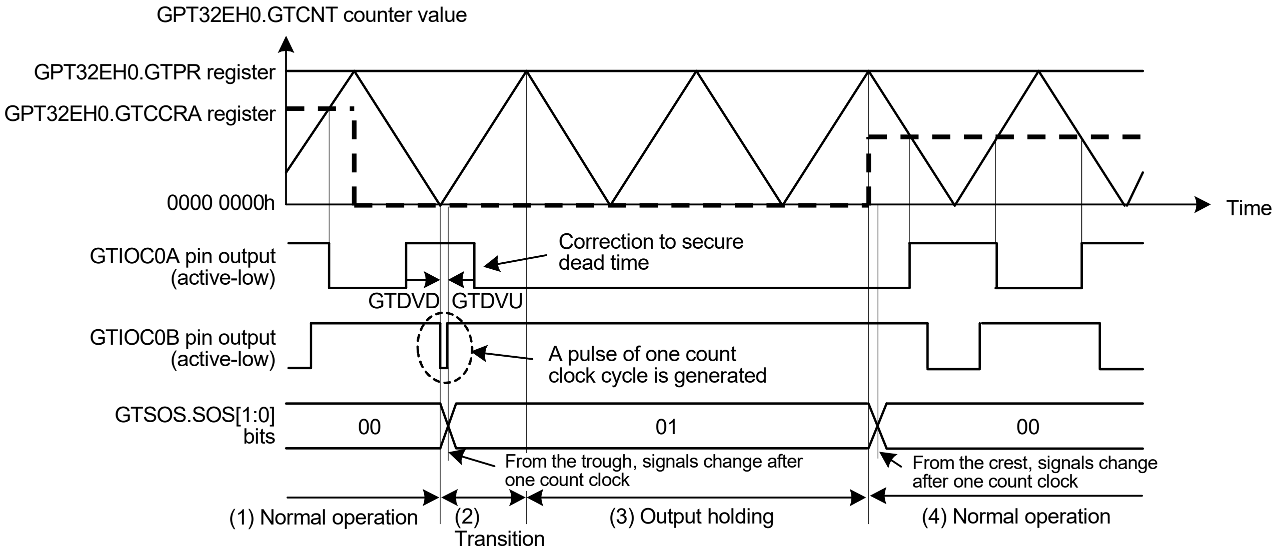
(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Transition period:	When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest
(3) Output holding period:	The output is held while GTCCRA register = 0.
(4) Normal operation period:	When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 23.99 Example of output protection operation when GTCCRA is set to 0 during buffer transfer at troughs, with 0 < GTCCRA < GTPR restored during buffer transfer at crests, and active-low



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Transition period:	When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest
(3) Output holding period:	The output is held while GTCCRA register = 0.
(4) Recovery period:	When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period:	Toggle output due to a compare match is performed normally.

Figure 23.100 Example of output protection operation when GTCCRA is set to 0 during buffer transfer at crests, with 0 < GTCCRA < GTPR restored during buffer transfer at troughs, and active-low

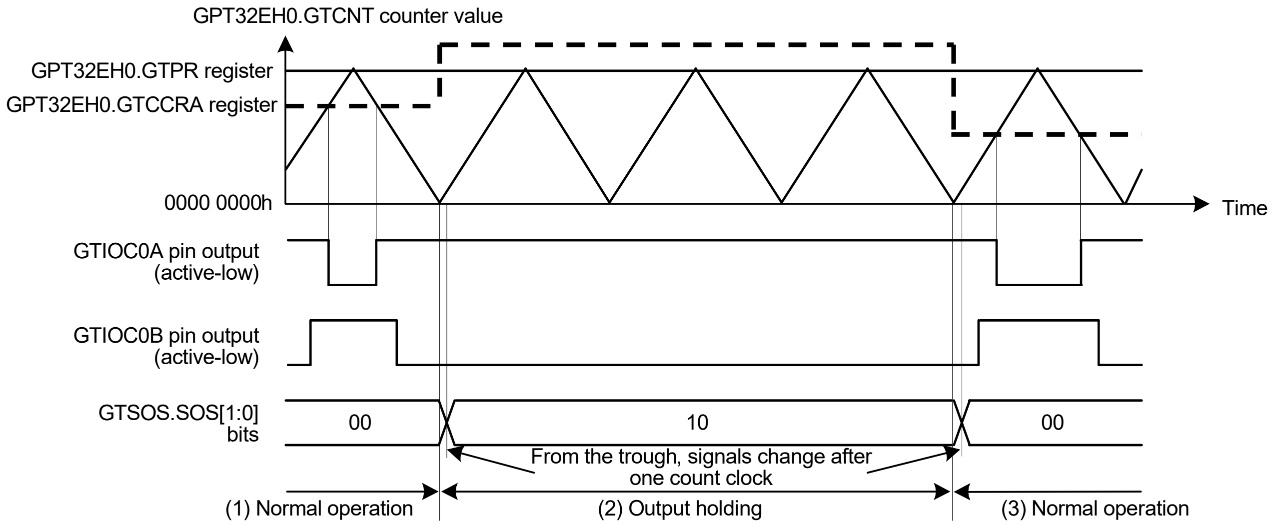


(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Transition period:	When GTCCRA register = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest
(3) Output holding period:	The output is held while GTCCRA register = 0.
(4) Normal operation period:	When 0 < GTCCRA register < GTPR register Is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 23.101 Example of output protection operation when GTCCRA is set to 0 during buffer transfer at crests, with 0 < GTCCRA < GTPR restored during buffer transfer at crests, and active-low

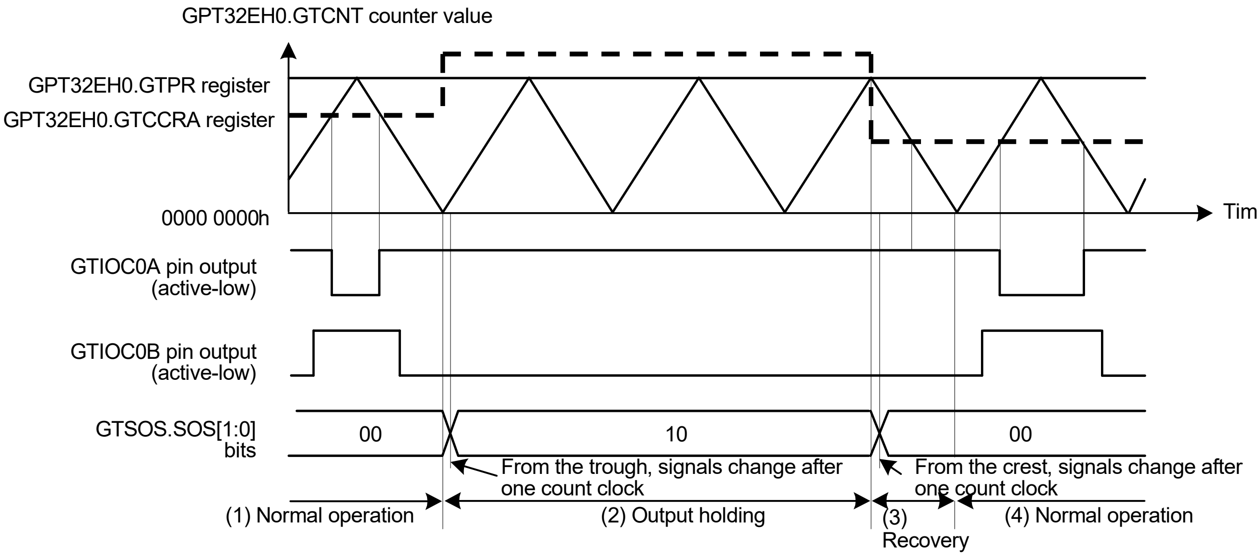
23.8.4.2 Output protection function when GTCCRA ≥ GTPR is set during buffer transfer at troughs

Figure 23.102 and Figure 23.103 show examples of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs.



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Normal operation period:	When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 23.102 Example of output protection operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs, with 0 < GTCCRA < GTPR restored during buffer transfer at troughs, and active-low

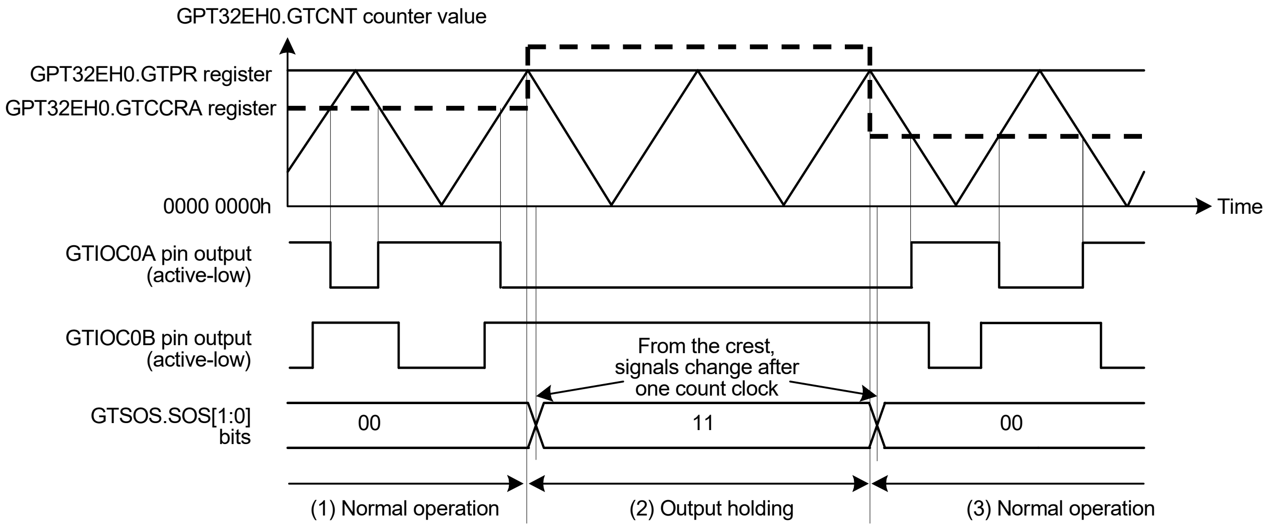


(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Recovery period:	When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period:	Toggle output due to a compare match is performed normally.

Figure 23.103 Example of output protection operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs, with 0 < GTCCRA < GTPR restored during buffer transfer at crests, and active-low

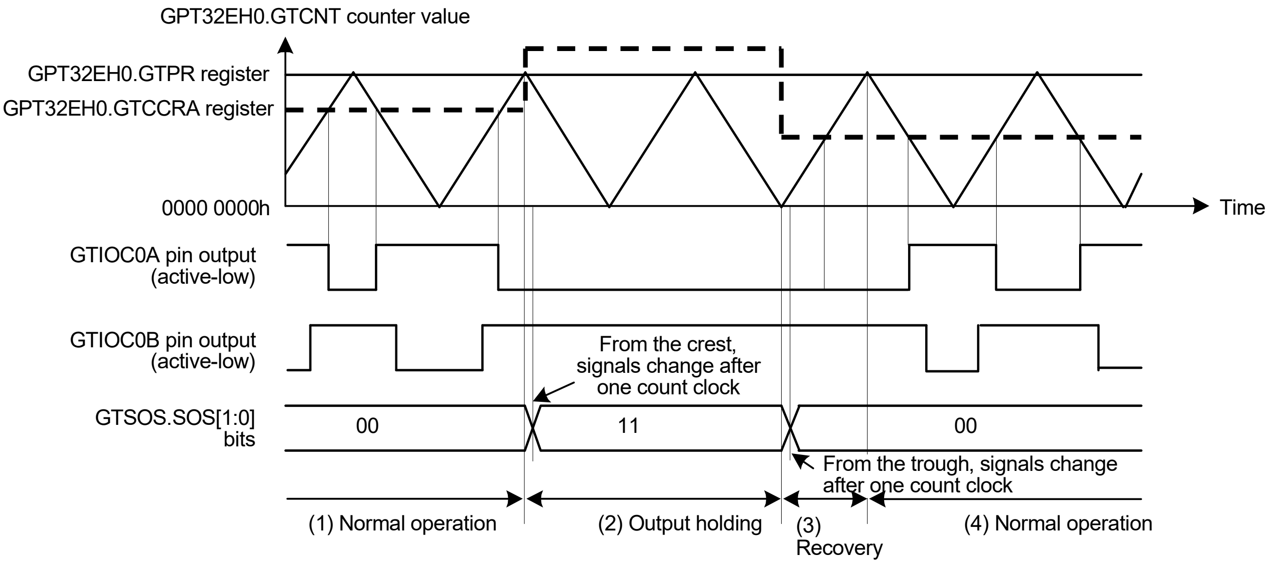
23.8.4.3 Output protection function when GTCCRA ≥ GTPR is set during buffer transfer at crests

Figure 23.104 and Figure 23.105 show examples of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests.



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA register ≥ GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 11b, and the output is held while GTCCRA register ≥ GTPR register.
(3) Normal operation period:	When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 23.104 Example of output protection operation when GTCCRA ≥ GTPR is set during buffer transfer at crests, with 0 < GTCCRA < GTPR restored during buffer transfer at crests, and active-low



(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA ≥ GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period:	When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest
(4) Normal operation period:	Toggle output due to a compare match is performed normally.

Figure 23.105 Example of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests, with 0 < GTCCRA < GTPR restored during buffer transfer at troughs, and active-low

23.8.4.4 Restricted specification of output protection function

The value of the GTCCRA register must be set within the range of (0 < GTCCRA < GTPR) at count start. If an incorrect value is set in the GTCCRA register during counting (a setting outside the range of 0 < GTCCRA < GTPR), the output protection function deactivates the level of one of the positive and negative outputs.

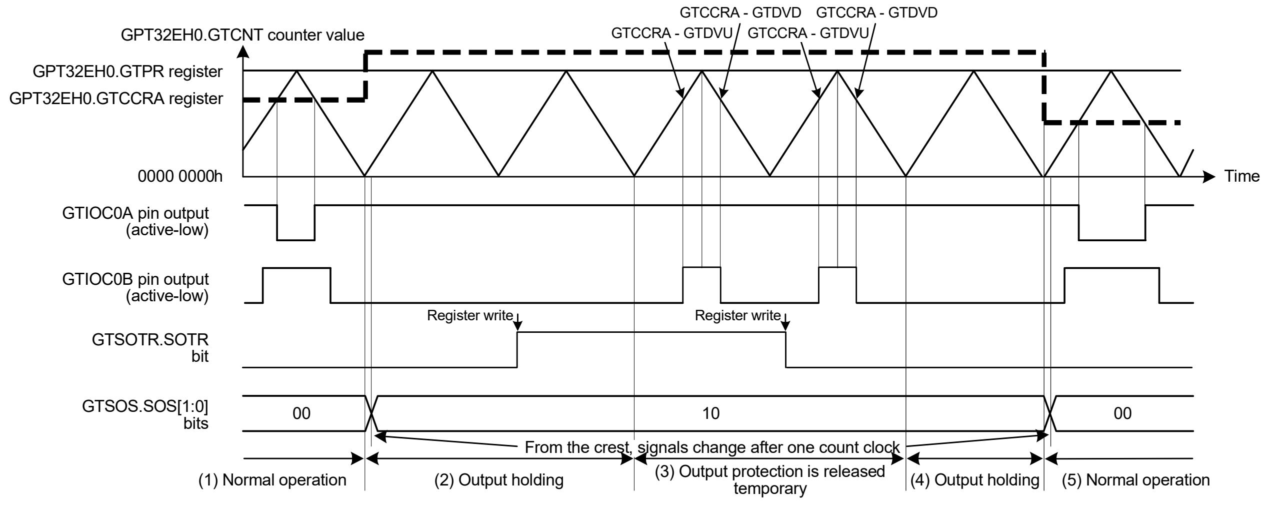
The function does not operate correctly if the following conditions are not satisfied:

- GTCCRA is 0 < GTCCRA < GTPR when counting starts
- The register conditions must be GTCCRA < GTPR + GTDVD - 1 during buffer transfer at crests
- When GTCCRA is greater than or equal to GTPR during buffer transfer at troughs, the register conditions must be GTCCRA > GTDVU + 1.

23.8.4.5 Temporary cancellation of output protection function

When the GTSOTR.SOTR bit is set to 1 with GTSOS.SOS[1:0] bits equal to 10b (showing output protection state by GTCCRA ≥ GTPR during buffer transfer at troughs), the output protection function for GTIOCB pin is temporarily canceled. GTSOS.SOS[1:0] bits retain the value of 10b even when the output protection function is canceled. When the SOTR bit is set to 0, the output protection function for GTIOCB pin resumes.

Figure 23.106 shows examples of temporary cancellation of output protection function operation when the GTCCRA ≥ GTPR is set during buffer transfer at troughs.



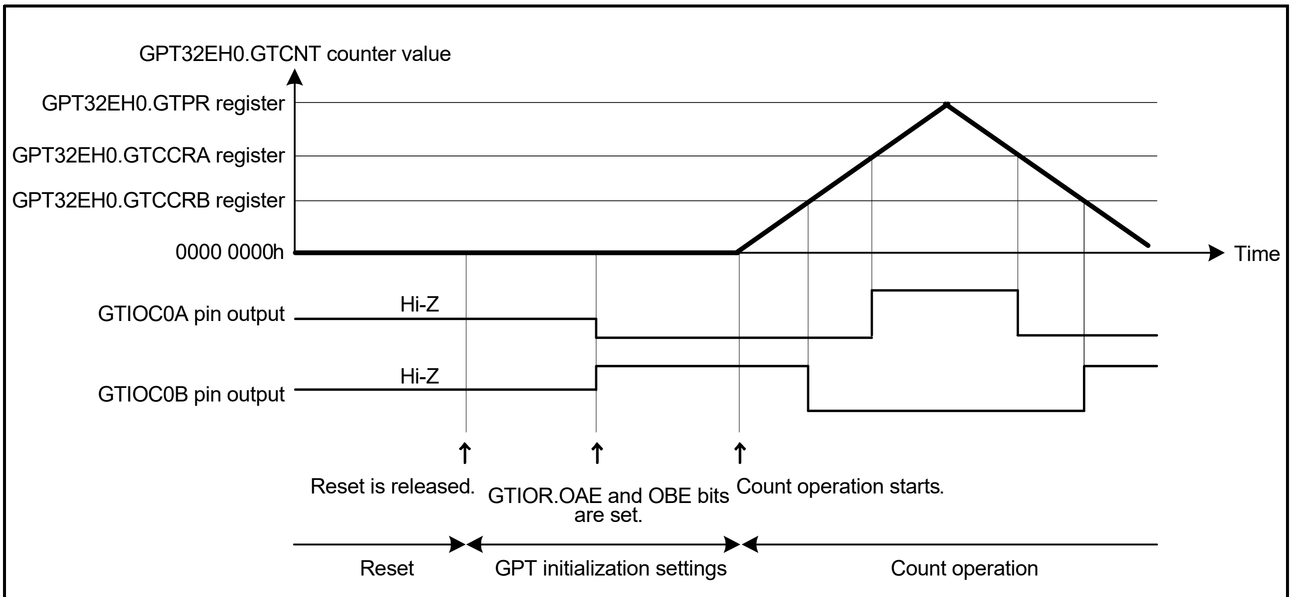
(1) Normal operation period:	Toggle output due to a compare match is performed normally.
(2) Output holding period:	When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA register ≥ GTPR register. If GTSOTR.SOTR is set to 1 during the output holding period, the GTIOCB output is holding until the first trough after the setting.
(3) Output protection releasing period:	After GTSOTR.SOTR is set to 1, the GTIOCB output is performed normally at the first trough after the setting. If GTSOTR.SOTR is set to 0 during the output protection releasing period, the output is holding at the first trough after the setting.
(4) Output holding period:	Until 0 < GTCCRA register < GTPR register is detected in the trough, the output is holding. When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to
(5) Normal operation period:	GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 23.106 Example of temporary cancellation of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs, with 0 < GTCCRA < GTPR restored during buffer transfer at troughs, and active-low

23.9 Initialization Method of Output Pins

23.9.1 Pin Settings after Reset

The GPT registers are initialized at reset. Start counting after selecting the port pin function with the PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial low output, output retained at cycle end, output toggled at compare match
GTIOR.GTIOB[4:0] bits: Initial high output, output retained at cycle end, output toggled at compare match

Figure 23.107 Example of pin settings after reset

23.9.2 Pin Initialization Caused by Error during Operation

If an error occurs during GPT operation, the following four types of pin processing can be performed before pin initialization:

- Set the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values in OADFLT and OBDFLT in GTIOR, and output the arbitrary values on count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR, and PmnPFS registers of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0 and the control bit associated with the pin in the PmnPFS.PMR to 0 to allow arbitrary values to be output from the pin set as a general output port when an error occurs.
- Drive the output to a high impedance state using the POEG function.

When the automatic dead time setting is made, clear the GTDTCR.TDE bit to 0 after counting stops. When counting stops, only the values of registers that are changed by a GPT external source change. If counting resumes, operation continues from where it stopped. If counting stops, registers must be initialized before counting starts.

23.10 Usage Notes

23.10.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

23.10.2 GTCCRn Settings during Compare Match Operation (n = A to F)

(1) When automatic dead time setting is made in triangle-wave PWM mode

The GTCCRA register must satisfy the following conditions: GTDVU < GTCCRA, GTDVD < GTCCRA, and GTCCRA < GTPR.

When the setting of GTCCRA = 0 or GTCCRA ≥ GTPR is made during count operation, the output protection function is activated.

However, the function does not operate correctly if the following conditions are not satisfied:

- GTCCRA is 0 < GTCCRA < GTPR when counting starts
- The register conditions must be GTCCRA < GTPR + GTDVD - 1 during buffer transfer at crests
- When GTCCRA is greater than or equal to GTPR during buffer transfer at troughs, the register conditions must be GTCCRA > GTDVU + 1.

For details, see section 23.8.4, Output Protection Function for GTIOC Pin Output.

(2) When automatic dead time setting is not made in triangle-wave PWM mode

The GTCCRA register must be set within the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. When GTCCRA > GTPR, no compare match occurs.

Similarly, GTCCRB must be set within the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB = 0 or GTCCRB = GTPR is satisfied. When GTCCRB > GTPR, no compare match occurs.

(3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following constraints. If the constraints are not satisfied, correct output waveforms with secured dead time might not be obtained:

- In up-counting: GTCCRC < GTCCRD, GTCCRC > GTDVU, GTCCRD < GTPR - GTDVD
- In down-counting: GTCCRC > GTCCRD, GTCCRC < GTPR - GTDVU, GTCCRD > GTDVD.

(4) When automatic dead time setting is not made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following constraints. If the constraints are not satisfied, two compare matches do not occur and pulse output cannot be performed:

- In up-counting: 0 < GTCCRC < GTCCRD < GTPR
- In down-counting: GTPR > GTCCRC > GTCCRD > 0.

Similarly, GTCCRE and GTCCRF must be set to satisfy the following constraints. If the constraints are not satisfied, two compare matches do not occur and pulse output cannot be performed:

- In up-counting: 0 < GTCCRE < GTCCRF < GTPR
- In down-counting: GTPR > GTCCRE > GTCCRF > 0.

(5) In saw-wave PWM mode

The GTCCRA register must be set with the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. If GTCCRA > GTPR is set, no compare match occurs.

Similarly, GTCCRB must be set with the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB = 0 or GTCCRB = GTPR is satisfied. If GTCCRB > GTPR is set, no compare match occurs.

23.10.3 Setting Range for the GTCNT Counter

The GTCNT counter register must be set with the range of $0 \leq \text{GTCNT} \leq \text{GTPR}$.

23.10.4 Starting and Stopping the GTCNT Counter

The control timing of starting and stopping the GTCNT counter by the GTCR.CST bit synchronizes the count clock that is selected in GTCR.TPCS[2:0]. When GTCR.CST is updated, the GTCNT counter starts/stops after a count clock selected in GTCR.TPCS[2:0]. Therefore, an event generated before the GTCNT counter actually starts is ignored. On the other hand, there might be cases where an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

23.10.5 Priority Order of Each Event

(1) GTCNT register

Table 23.24 shows a priority order of events updating GTCNT register.

Table 23.24 Priority order of sources updating GTCNT

Source updating GTCNT	Priority order
Writing by CPU (Writing to GTCNT/GTCLR)	High ↑
Clear by hardware sources set in GTCSR	
Count up or down by hardware sources set in GTUPSR/GTDNSR	
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

(2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), writing by CPU has priority over starting/stopping by hardware sources.

When there is a conflict between starting by hardware sources set in the GTSSR register and stopping by hardware sources set in GTPSR register, the GTCR.CST bit value does not change. Where there is a conflict between updating the GTCR.CST bit and reading by the CPU, pre-update data is read.

(3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to GTCCRm registers, writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. Where there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

(4) GTPR registers

When there is a conflict between buffer transfer operation and writing to the GTPR register, writing to GTPR register has priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

(5) GTADTRm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to the GTADTRm registers, writing to the GTADTRm registers has priority over buffer transfer operation. Where there is a conflict between updating GTADTRm registers and reading by the CPU, pre-update data is read.

(6) GTDVm registers (m = U, D)

When there is a conflict between buffer transfer operation and writing to GTDVm registers, writing to GTDVm registers has priority over buffer transfer operation. When there is a conflict between updating GTDVm registers and reading by the CPU, pre-update data is read.

24. PWM Delay Generation Circuit

24.1 Overview

The MCU has 4 channel delay circuits that can connect to the General PWM Timer (GPT). Table 24.1 lists the specifications for the PWM Delay Generation Circuit, Figure 24.1 shows a block diagram, and Table 24.2 lists the I/O pins.

Table 24.1 Specifications of the PWM Delay Generation Circuit

Parameter	Specifications
Function	The circuit can control the timing with which signals on the two PWM output pins for channel 0/1/2/3 rise and fall to an accuracy of up to 1/32 times the period of the GPT clock (PCLKD).

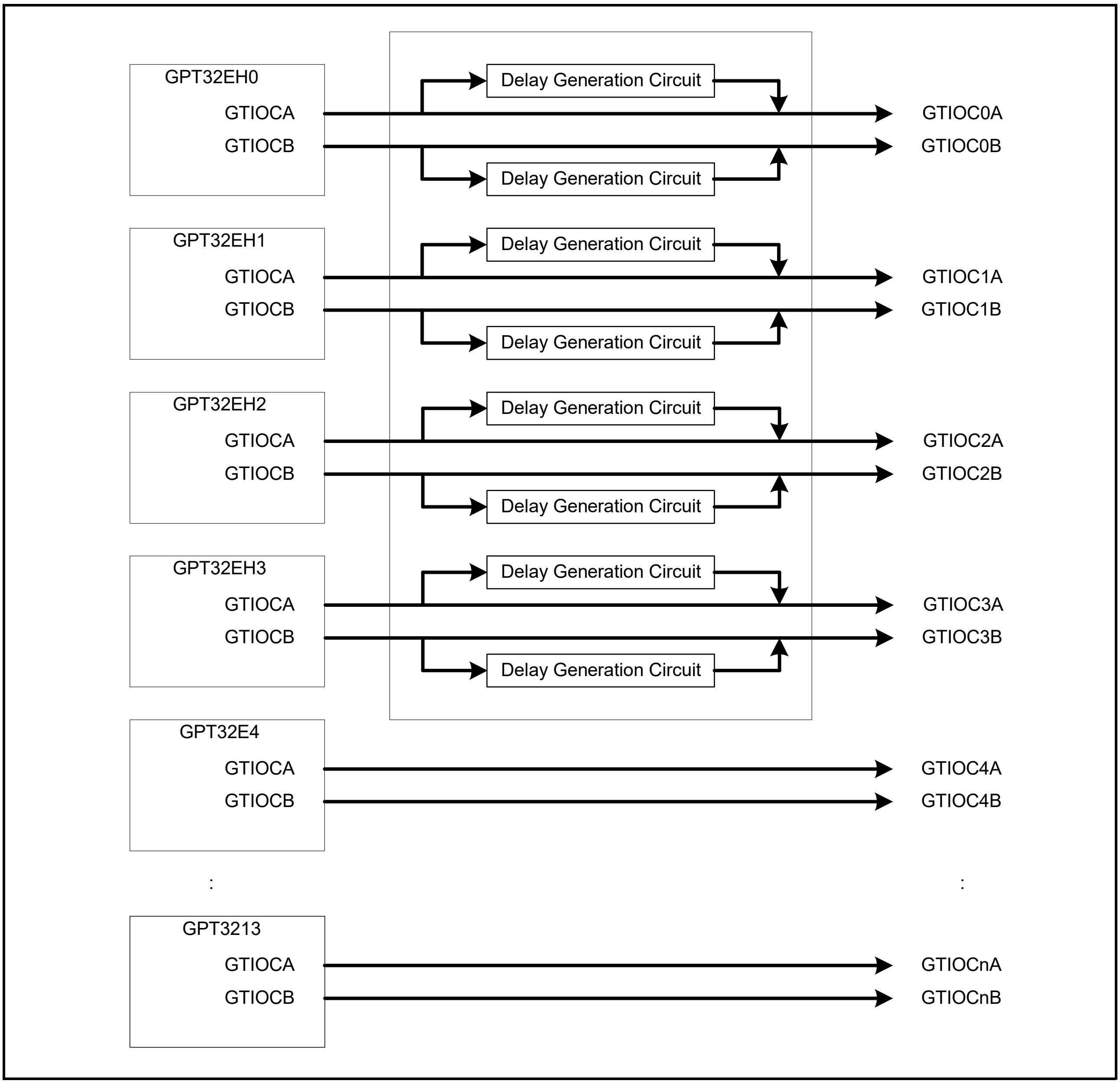


Figure 24.1 PWM delay generation circuit block diagram

Table 24.2 PWM delay generation circuit I/O pins

I/O pin	I/O	Function
GTIOC0A	Output	Delayed output of GTIOCA pin of GPT channel 0
GTIOC0B	Output	Delayed output of GTIOCB pin of GPT channel 0
GTIOC1A	Output	Delayed output of GTIOCA pin of GPT channel 1
GTIOC1B	Output	Delayed output of GTIOCB pin of GPT channel 1
GTIOC2A	Output	Delayed output of GTIOCA pin of GPT channel 2
GTIOC2B	Output	Delayed output of GTIOCB pin of GPT channel 2
GTIOC3A	Output	Delayed output of GTIOCA pin of GPT channel 3
GTIOC3B	Output	Delayed output of GTIOCB pin of GPT channel 3

24.2 Register Descriptions

24.2.1 PWM Output Delay Control Register (GTDLYCR)

Address(es): GPT_ODC.GTDLYCR 4007 B000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DLYRST	DLLEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DLLEN	DLL Operation Enable	0: DLL operation disabled 1: DLL operation enabled.	R/W
b1	DLYRST	PWM Delay Generation Circuit Reset	0: Normal operation 1: Reset.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDLYCR register controls the PWM delay generation circuit, which applies delays to the PWM outputs. GTDLYCR register can be written when register write protection is disabled (GPT32EH0.GTWP.WP = 0).

DLLEN bit (DLL Operation Enable)

The DLLEN bit selects whether the on-chip DLL in the PWM delay generation circuit is activated or not.

DLYRST bit (PWM Delay Generation Circuit Reset)

The DLYRST bit resets the internal state of the PWM delay generation circuit.

24.2.2 PWM Output Delay Control Register 2 (GTDLYCR2)

Address(es): GPT_ODC.GTDLYCR2 4007 B002h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	DLYEN3	DLYEN2	DLYEN1	DLYEN0	—	—	—	—	DLYBS3	DLYBS2	DLYBS1	DLYBS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DLYBS0	PWM Delay Generation Circuit bypass for channel 0	0: Delay generation circuit of channel 0 bypassed 1: Delay generation circuit of channel 0 not bypassed.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	DLYBS1	PWM Delay Generation Circuit bypass for channel 1	0: Delay generation circuit of channel 1 bypassed 1: Delay generation circuit of channel 1 not bypassed.	R/W
b2	DLYBS2	PWM Delay Generation Circuit bypass for channel 2	0: Delay generation circuit of channel 2 bypassed 1: Delay generation circuit of channel 2 not bypassed.	R/W
b3	DLYBS3	PWM Delay Generation Circuit bypass for channel 3	0: Delay generation circuit of channel 3 bypassed 1: Delay generation circuit of channel 3 not bypassed.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DLYEN0	PWM Delay Generation Circuit enable for channel 0	0: Delay generation circuit of channel 0 enabled 1: Delay generation circuit of channel 0 disabled.	R/W
b9	DLYEN1	PWM Delay Generation Circuit enable for channel 1	0: Delay generation circuit of channel 1 enabled 1: Delay generation circuit of channel 1 disabled.	R/W
b10	DLYEN2	PWM Delay Generation Circuit enable for channel 2	0: Delay generation circuit of channel 2 enabled 1: Delay generation circuit of channel 2 disabled.	R/W
b11	DLYEN3	PWM Delay Generation Circuit enable for channel 3	0: Delay generation circuit of channel 3 enabled 1: Delay generation circuit of channel 3 disabled.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDLYCR2 register controls each channel of PWM delay generation circuit. GTDLYCR2 can be written when register write protection is disabled (GPT32EH0.GTWP.WP = 0).

DLYBSn (n = 0 to 3) bit (PWM Delay Generation Circuit Bypass for channel n)

The DLYBSn bit selects whether delays are applied to PWM output signals from the GTIOCnA and GTIOCnB pins (n = 0 to 3) by the PWM delay generation circuit or whether the circuit is bypassed.

A signal delayed in the PWM delay generation circuit is output 3 cycles of GPT operation clock (PCLKD) later than if it bypasses the PWM delay generation circuit.

DLYENn (n = 0 to 3) bit (PWM Delay Generation Circuit Enable for channel n)

The DLYENn bit selects whether channel n (n = 0 to 3) of PWM delay generation circuit is power on or off. If channel n of the PWM delay generation circuit is not used, set to 1.

24.2.3 GTIOCnA Rising Output Delay Register (GTDLYRnA) (n = 0 to 3)

Address(es): GPT_ODC.GTDLYR0A 4007 B018h, GPT_ODC.GTDLYR1A 4007 B01Ch,
GPT_ODC.GTDLYR2A 4007 B020h, GPT_ODC.GTDLYR3A 4007 B024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	DLY[4:0]	GTIOCnA Output Rising Edge Delay Setting	b4 b0 0 0 0 0 0: Delay on rising edges is not applied 0 0 0 0 1: Delay of 1/32 times PCLKD period applied 0 0 0 1 0: Delay of 2/32 times PCLKD period applied 0 0 0 1 1: Delay of 3/ 32 times PCLKD period applied 0 0 1 0 0: Delay of 4/ 32 times PCLKD period applied 0 0 1 0 1: Delay of 5/ 32 times PCLKD period applied 0 0 1 1 0: Delay of 6/ 32 times PCLKD period applied 0 0 1 1 1: Delay of 7/ 32 times PCLKD period applied 0 1 0 0 0: Delay of 8/ 32 times PCLKD period applied 0 1 0 0 1: Delay of 9/ 32 times PCLKD period applied 0 1 0 1 0: Delay of 10/ 32 times PCLKD period applied 0 1 0 1 1: Delay of 11/ 32 times PCLKD period applied 0 1 1 0 0: Delay of 12/ 32 times PCLKD period applied 0 1 1 0 1: Delay of 13/ 32 times PCLKD period applied 0 1 1 1 0: Delay of 14/ 32 times PCLKD period applied 0 1 1 1 1: Delay of 15/ 32 times PCLKD period applied 1 0 0 0 0: Delay of 16/ 32 times PCLKD period applied 1 0 0 0 1: Delay of 17/ 32 times PCLKD period applied 1 0 0 1 0: Delay of 18/ 32 times PCLKD period applied 1 0 0 1 1: Delay of 19/ 32 times PCLKD period applied 1 0 1 0 0: Delay of 20/ 32 times PCLKD period applied 1 0 1 0 1: Delay of 21/ 32 times PCLKD period applied 1 0 1 1 0: Delay of 22/ 32 times PCLKD period applied 1 0 1 1 1: Delay of 23/ 32 times PCLKD period applied 1 1 0 0 0: Delay of 24/ 32 times PCLKD period applied 1 1 0 0 1: Delay of 25/ 32 times PCLKD period applied 1 1 0 1 0: Delay of 26/ 32 times PCLKD period applied 1 1 0 1 1: Delay of 27/ 32 times PCLKD period applied 1 1 1 0 0: Delay of 28/ 32 times PCLKD period applied 1 1 1 0 1: Delay of 29/ 32 times PCLKD period applied 1 1 1 1 0: Delay of 30/ 32 times PCLKD period applied 1 1 1 1 1: Delay of 31/ 32 times PCLKD period applied.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDLYRnA register sets a delay to be applied to rising edges of output signals on the GTIOCnA pin. On the timing for the transfer of settings, see section 24.3.2, Timing for Transfer of GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYRnA can be written when register write protection is disabled (GPT32EHn.GTWP.WP = 0).

24.2.4 GTIOCnA Falling Output Delay Register (GTDLYFnA) (n = 0 to 3)

Address(es): GPT_ODC.GTDLYF0A 4007 B028h, GPT_ODC.GTDLYF1A 4007 B02Ch, GPT_ODC.GTDLYF2A 4007 B030h, GPT_ODC.GTDLYF3A 4007 B034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	DLY[4:0]	GTIOCnA Output Falling Edge Delay Setting	b4 b0 0 0 0 0 0: Delay on falling edges is not applied 0 0 0 0 1: Delay of 1/32 times PCLKD period applied 0 0 0 1 0: Delay of 2/32 times PCLKD period applied 0 0 0 1 1: Delay of 3/ 32 times PCLKD period applied 0 0 1 0 0: Delay of 4/ 32 times PCLKD period applied 0 0 1 0 1: Delay of 5/ 32 times PCLKD period applied 0 0 1 1 0: Delay of 6/ 32 times PCLKD period applied 0 0 1 1 1: Delay of 7/ 32 times PCLKD period applied 0 1 0 0 0: Delay of 8/ 32 times PCLKD period applied 0 1 0 0 1: Delay of 9/ 32 times PCLKD period applied 0 1 0 1 0: Delay of 10/ 32 times PCLKD period applied 0 1 0 1 1: Delay of 11/ 32 times PCLKD period applied 0 1 1 0 0: Delay of 12/ 32 times PCLKD period applied 0 1 1 0 1: Delay of 13/ 32 times PCLKD period applied 0 1 1 1 0: Delay of 14/ 32 times PCLKD period applied 0 1 1 1 1: Delay of 15/ 32 times PCLKD period applied 1 0 0 0 0: Delay of 16/ 32 times PCLKD period applied 1 0 0 0 1: Delay of 17/ 32 times PCLKD period applied 1 0 0 1 0: Delay of 18/ 32 times PCLKD period applied 1 0 0 1 1: Delay of 19/ 32 times PCLKD period applied 1 0 1 0 0: Delay of 20/ 32 times PCLKD period applied 1 0 1 0 1: Delay of 21/ 32 times PCLKD period applied 1 0 1 1 0: Delay of 22/ 32 times PCLKD period applied 1 0 1 1 1: Delay of 23/ 32 times PCLKD period applied 1 1 0 0 0: Delay of 24/ 32 times PCLKD period applied 1 1 0 0 1: Delay of 25/ 32 times PCLKD period applied 1 1 0 1 0: Delay of 26/ 32 times PCLKD period applied 1 1 0 1 1: Delay of 27/ 32 times PCLKD period applied 1 1 1 0 0: Delay of 28/ 32 times PCLKD period applied 1 1 1 0 1: Delay of 29/ 32 times PCLKD period applied 1 1 1 1 0: Delay of 30/ 32 times PCLKD period applied 1 1 1 1 1: Delay of 31/ 32 times PCLKD period applied.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDLYFnA register sets a delay to be applied to falling edges of output signals on the GTIOCnA pin. On the timing for the transfer of settings, see section 24.3.2, Timing for Transfer of GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYFnA can be written when register write protection is disabled (GPT32EHn.GTWP.WP = 0).

24.2.5 GTIOCnB Rising Output Delay Register (GTDLYRnB) (n = 0 to 3)

Address(es): GPT_ODC.GTDLYR0B 4007 B01Ah, GPT_ODC.GTDLYR1B 4007 B01Eh,
GPT_ODC.GTDLYR2B 4007 B022h, GPT_ODC.GTDLYR3B 4007 B026h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	DLY[4:0]	GTIOCnB Output Rising Edge Delay Setting	b4 b0 0 0 0 0 0: Do not apply delay on rising edges is not applied 0 0 0 0 1: Delay of 1/32 times PCLKD period applied 0 0 0 1 0: Delay of 2/32 times PCLKD period applied 0 0 0 1 1: Delay of 3/ 32 times PCLKD period applied 0 0 1 0 0: Delay of 4/ 32 times PCLKD period applied 0 0 1 0 1: Delay of 5/ 32 times PCLKD period applied 0 0 1 1 0: Delay of 6/ 32 times PCLKD period applied 0 0 1 1 1: Delay of 7/ 32 times PCLKD period applied 0 1 0 0 0: Delay of 8/ 32 times PCLKD period applied 0 1 0 0 1: Delay of 9/ 32 times PCLKD period applied 0 1 0 1 0: Delay of 10/ 32 times PCLKD period applied 0 1 0 1 1: Delay of 11/ 32 times PCLKD period applied 0 1 1 0 0: Delay of 12/ 32 times PCLKD period applied 0 1 1 0 1: Delay of 13/ 32 times PCLKD period applied 0 1 1 1 0: Delay of 14/ 32 times PCLKD period applied 0 1 1 1 1: Delay of 15/ 32 times PCLKD period applied 1 0 0 0 0: Delay of 16/ 32 times PCLKD period applied 1 0 0 0 1: Delay of 17/ 32 times PCLKD period applied 1 0 0 1 0: Delay of 18/ 32 times PCLKD period applied 1 0 0 1 1: Delay of 19/ 32 times PCLKD period applied 1 0 1 0 0: Delay of 20/ 32 times PCLKD period applied 1 0 1 0 1: Delay of 21/ 32 times PCLKD period applied 1 0 1 1 0: Delay of 22/ 32 times PCLKD period applied 1 0 1 1 1: Delay of 23/ 32 times PCLKD period applied 1 1 0 0 0: Delay of 24/ 32 times PCLKD period applied 1 1 0 0 1: Delay of 25/ 32 times PCLKD period applied 1 1 0 1 0: Delay of 26/ 32 times PCLKD period applied 1 1 0 1 1: Delay of 27/ 32 times PCLKD period applied 1 1 1 0 0: Delay of 28/ 32 times PCLKD period applied 1 1 1 0 1: Delay of 29/ 32 times PCLKD period applied 1 1 1 1 0: Delay of 30/ 32 times PCLKD period applied 1 1 1 1 1: Delay of 31/ 32 times PCLKD period applied.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDLYRnB register sets a delay to be applied to rising edges of output signals on the GTIOCnB pin. On the timing for the transfer of settings, see section 24.3.2, Timing for Transfer of GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYRnB can be written when register write protection is disabled (GPT32EHn.GTWP.WP = 0).

24.2.6 GTIOCnB Falling Output Delay Register (GTDLYFnB) (n = 0 to 3)

Address(es): GPT_ODC.GTDLYF0B 4007 B02Ah, GPT_ODC.GTDLYF1B 4007 B02Eh,
GPT_ODC.GTDLYF2B 4007 B032h, GPT_ODC.GTDLYF3B 4007 B036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	DLY[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	DLY[4:0]	GTIOCnB Output Falling Edge Delay Setting	b4 b0 0 0 0 0 0: Delay on falling edges is not applied 0 0 0 0 1: Delay of 1/32 times PCLKD period applied 0 0 0 1 0: Delay of 2/32 times PCLKD period applied 0 0 0 1 1: Delay of 3/ 32 times PCLKD period applied 0 0 1 0 0: Delay of 4/ 32 times PCLKD period applied 0 0 1 0 1: Delay of 5/ 32 times PCLKD period applied 0 0 1 1 0: Delay of 6/ 32 times PCLKD period applied 0 0 1 1 1: Delay of 7/ 32 times PCLKD period applied 0 1 0 0 0: Delay of 8/ 32 times PCLKD period applied 0 1 0 0 1: Delay of 9/ 32 times PCLKD period applied 0 1 0 1 0: Delay of 10/ 32 times PCLKD period applied 0 1 0 1 1: Delay of 11/ 32 times PCLKD period applied 0 1 1 0 0: Delay of 12/ 32 times PCLKD period applied 0 1 1 0 1: Delay of 13/ 32 times PCLKD period applied 0 1 1 1 0: Delay of 14/ 32 times PCLKD period applied 0 1 1 1 1: Delay of 15/ 32 times PCLKD period applied 1 0 0 0 0: Delay of 16/ 32 times PCLKD period applied 1 0 0 0 1: Delay of 17/ 32 times PCLKD period applied 1 0 0 1 0: Delay of 18/ 32 times PCLKD period applied 1 0 0 1 1: Delay of 19/ 32 times PCLKD period applied 1 0 1 0 0: Delay of 20/ 32 times PCLKD period applied 1 0 1 0 1: Delay of 21/ 32 times PCLKD period applied 1 0 1 1 0: Delay of 22/ 32 times PCLKD period applied 1 0 1 1 1: Delay of 23/ 32 times PCLKD period applied 1 1 0 0 0: Delay of 24/ 32 times PCLKD period applied 1 1 0 0 1: Delay of 25/ 32 times PCLKD period applied 1 1 0 1 0: Delay of 26/ 32 times PCLKD period applied 1 1 0 1 1: Delay of 27/ 32 times PCLKD period applied 1 1 1 0 0: Delay of 28/ 32 times PCLKD period applied 1 1 1 0 1: Delay of 29/ 32 times PCLKD period applied 1 1 1 1 0: Delay of 30/ 32 times PCLKD period applied 1 1 1 1 1: Delay of 31/ 32 times PCLKD period applied.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTDLYFnB register sets a delay to be applied to falling edges of output signals on the GTIOCnB pin. On the timing for the transfer of settings, see section 24.3.2, Timing for Transfer of GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB Register Settings.

GTDLYFnB can be written when register write protection is disabled (GPT32EHn.GTWP.WP = 0).

24.3 Operation

24.3.1 Adjustments to the Timing of Rising and Falling Edges in PWM Waveforms

The timing of rising and falling edges in PWM waveforms which are output from the GTIOCnA and GTIOCnB pins, where n = channel number, can be delayed to an accuracy of 1/32 of the GPT operating clock (PCLKD) period.

If the timing of rising or falling edges in PWM waveforms output from the GTIOCnA and GTIOCnB pins must be adjusted, initialize the PWM generation circuit as shown in the procedure in Figure 24.2.

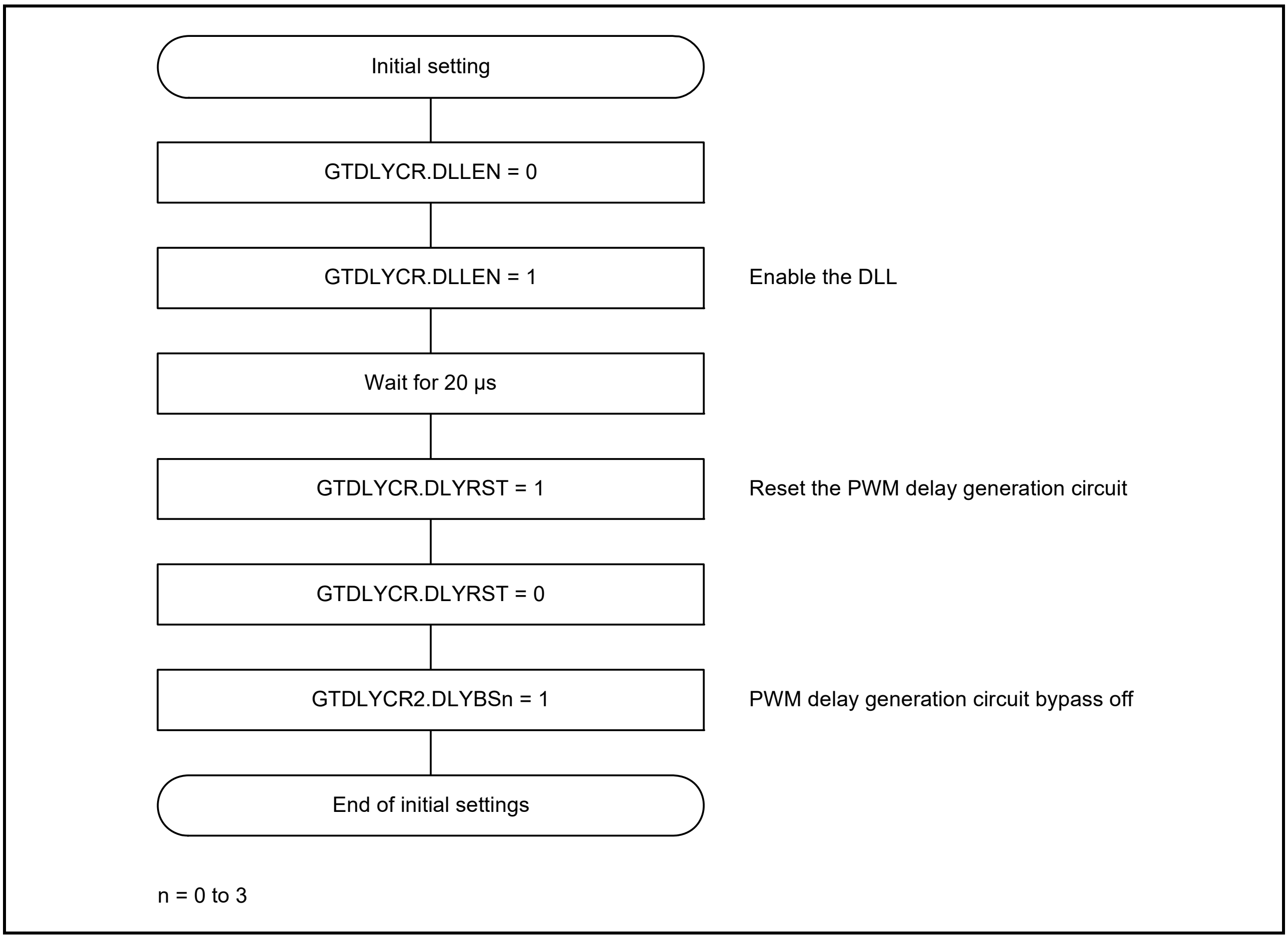


Figure 24.2 Example of initialization flow for the PWM delay generation circuit

In the PWM delay generation circuit, delay can be applied to rising and falling edges of the PWM output to an accuracy of 1/32 of the period of the GPT operation clock (PCLKD). This is described in section 23.3.3, PWM Output Operating Mode. Delays associated with the settings are reflected in the PWM output with the timing described in section 24.3.2, Timing for Transfer of GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB Register Settings. Table 24.3 shows the association between the GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB registers and the PWM outputs.

Table 24.3 Association between PWM output pins and delay setting registers

PWM output pin	Rising-edge delay setting register	Falling-edge delay setting register
GTIOC0A	GTDLYR0A	GTDLYF0A
GTIOC0B	GTDLYR0B	GTDLYF0B
GTIOC1A	GTDLYR1A	GTDLYF1A
GTIOC1B	GTDLYR1B	GTDLYF1B
GTIOC2A	GTDLYR2A	GTDLYF2A
GTIOC2B	GTDLYR2B	GTDLYF2B
GTIOC3A	GTDLYR3A	GTDLYF3A
GTIOC3B	GTDLYR3B	GTDLYF3B

When the PWM delay generation circuit is in use, the timing with which a PWM output signal rises and falls can be controlled to an accuracy of 1/32 of the period of the GPT operation clock (PCLKD). When this option is not in use, the period of the PWM output waveform is controlled to an accuracy of one period of the input clock for the timer counter, which is PCLKD. With the PWM delay generation circuit, the output can be controlled to an accuracy 32 times better. Additionally, the delay settings also control the periods at high and low level for the PWM waveform to the given accuracy. PWM delay generation circuit channels can be individually enabled or disabled.

24.3.2 Timing for Transfer of GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB Register Settings

Settings for the GTDLYRnA, GTLDYRnB, GTDLYFnA, and GTDLYFnB registers are initially transferred to temporary registers, and then reflected in the delay on the GTIOCnA and GTIOCnB (n = 0 to 3) outputs. Transfer of the settings takes place on overflows (in up-counting) or underflows (in down-counting) for saw waves, and in the troughs of triangle waves.

Figure 24.3 and Figure 24.4 show examples of the operation of the GTDLYR0A and GTDLYF0A registers.

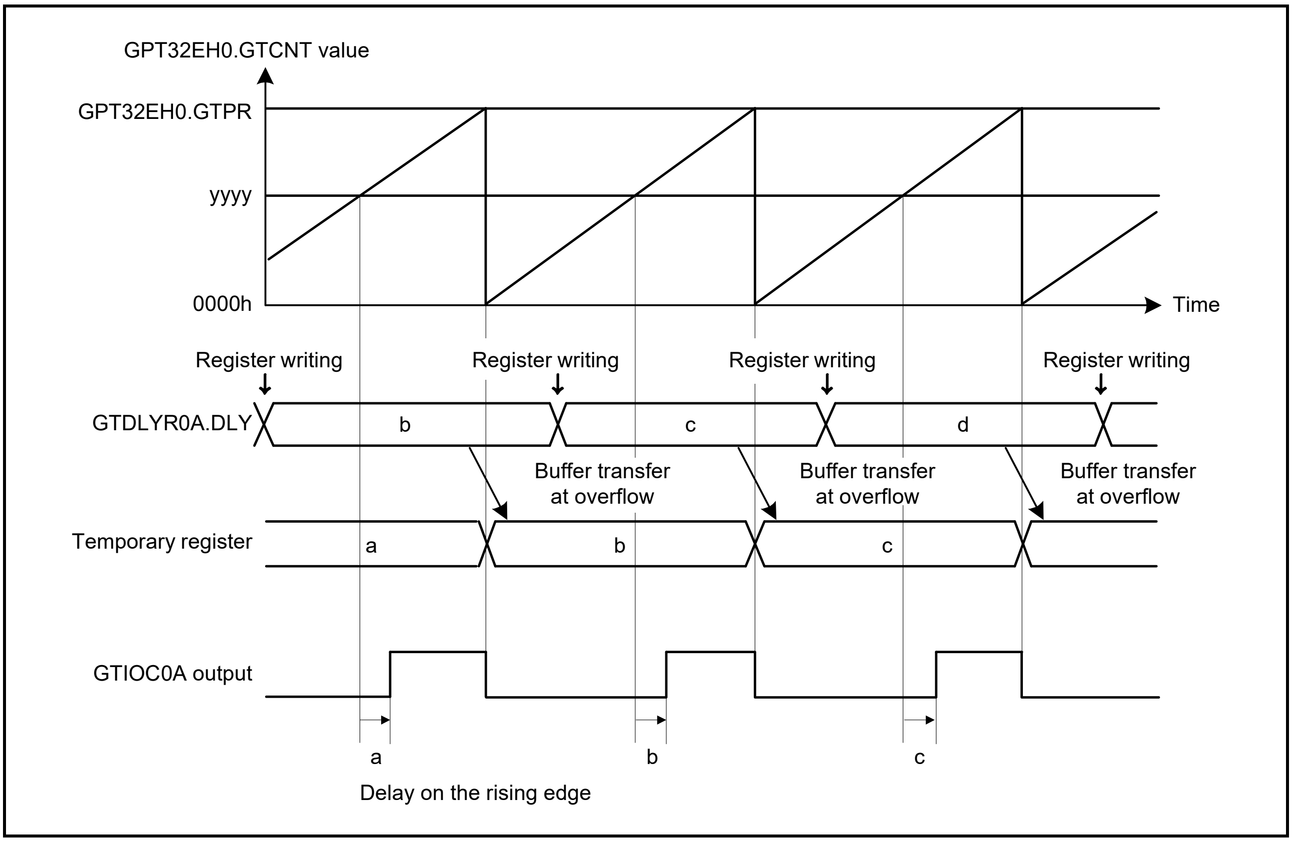


Figure 24.3 Example of GTLDYR0A register operation with PWM saw-wave generation

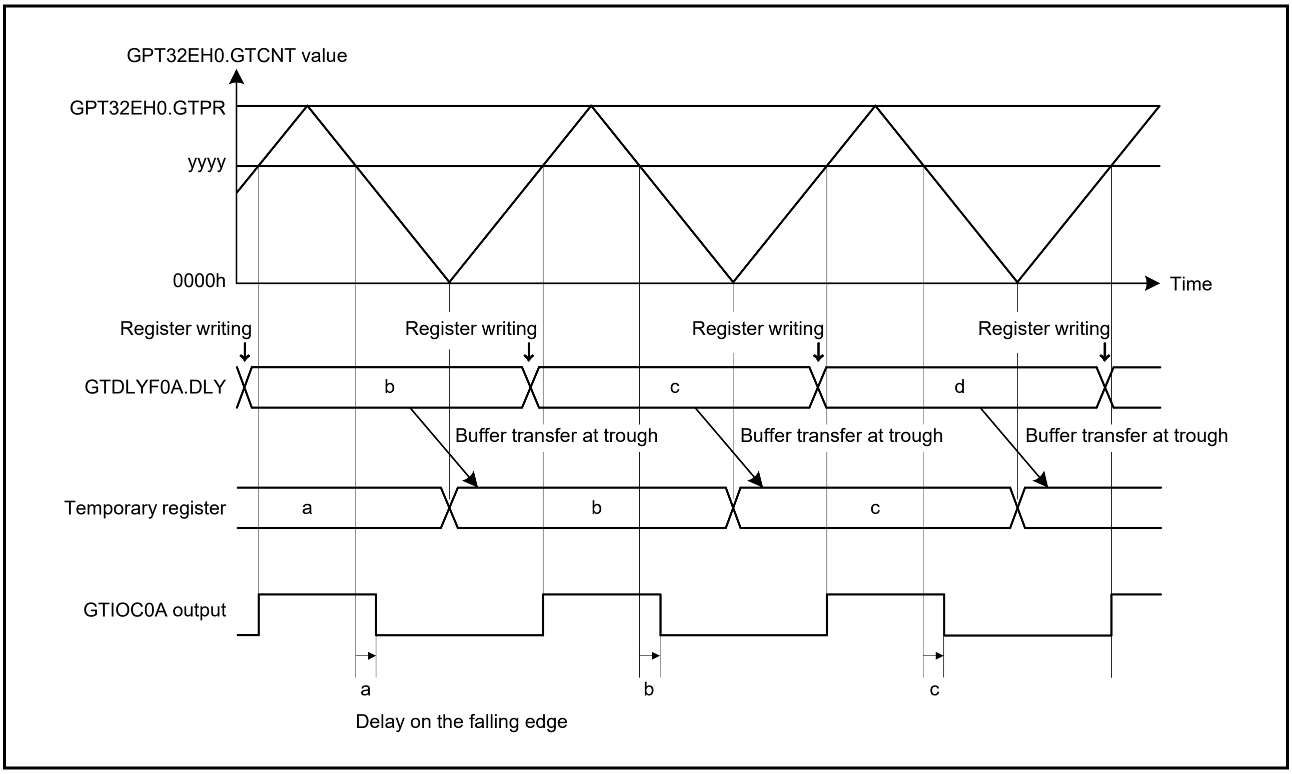


Figure 24.4 Example of GTLDYF0A register operation with PWM triangle-wave generation

24.4 Usage Notes

24.4.1 Settings for the Module-Stop Function

The Module Stop Control Register D (MSTPCRD) can enable or disable operation of the PWM delay generation circuit. The PWM delay generation circuit is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

24.4.2 Notes on Delay Settings for PWM Delay Generation Circuit

When the PWM delay generation circuit generates delays for a PWM output waveform and the waveform is toggled in response to compare-matches, do not change the settings for delay while the compare-match value is within the ranges listed in Table 24.4. This constraint applies to the GTDLYFnA, GTDLYRnA, GTDLFnB, and GTDLYRnB registers.

Table 24.4 Constraints on delay settings

Mode	Direction of counting	Compare-match value
Saw-wave mode	Up	GTPR - 2 or above
	Down	2 or below
Triangle-wave mode	Down	2 or below

Figure 24.5 shows an example of how the constraints apply to the timing of setting GTDLYFnA in saw-wave waveform one-shot pulse mode (counting up). Do not change the value set in GTDLYFnA while GTCCRD ≥ GTPR - 2.

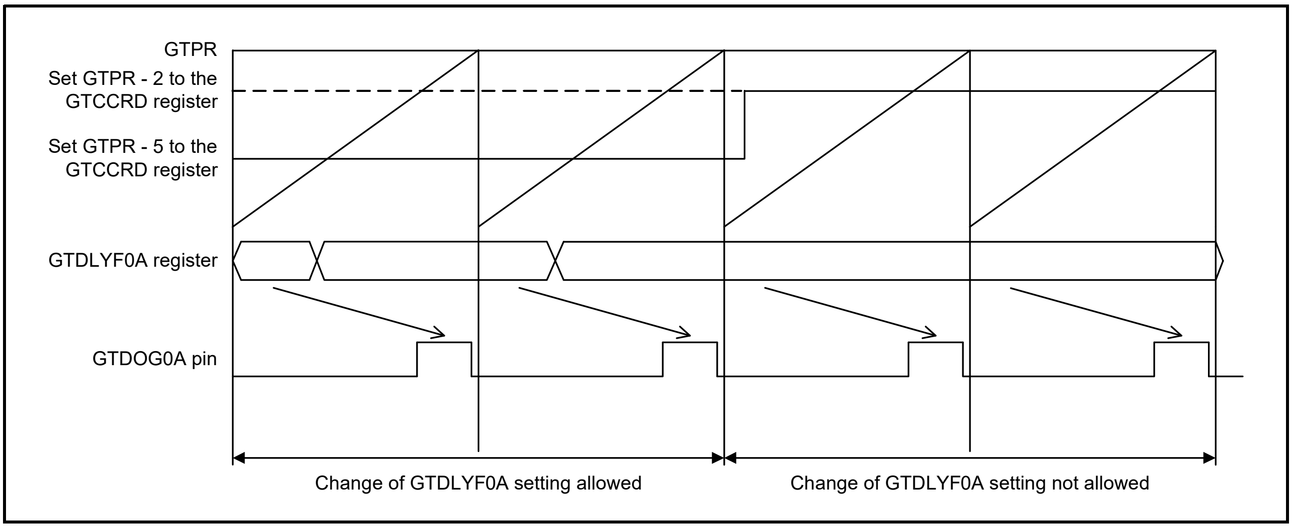


Figure 24.5 Constraints on the timing of GTDLYF0A register settings

Changing the values in the GTDLYFnA, GTDLYRnA, GTDLYFnB, and GTDLYRnB registers during periods where changes to settings are not allowed, might lead to faulty output waveforms such as shifts in the timing of output waveform transitions from the expected values.

25. Asynchronous General-Purpose Timer (AGT)

25.1 Overview

The Asynchronous General-Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events.

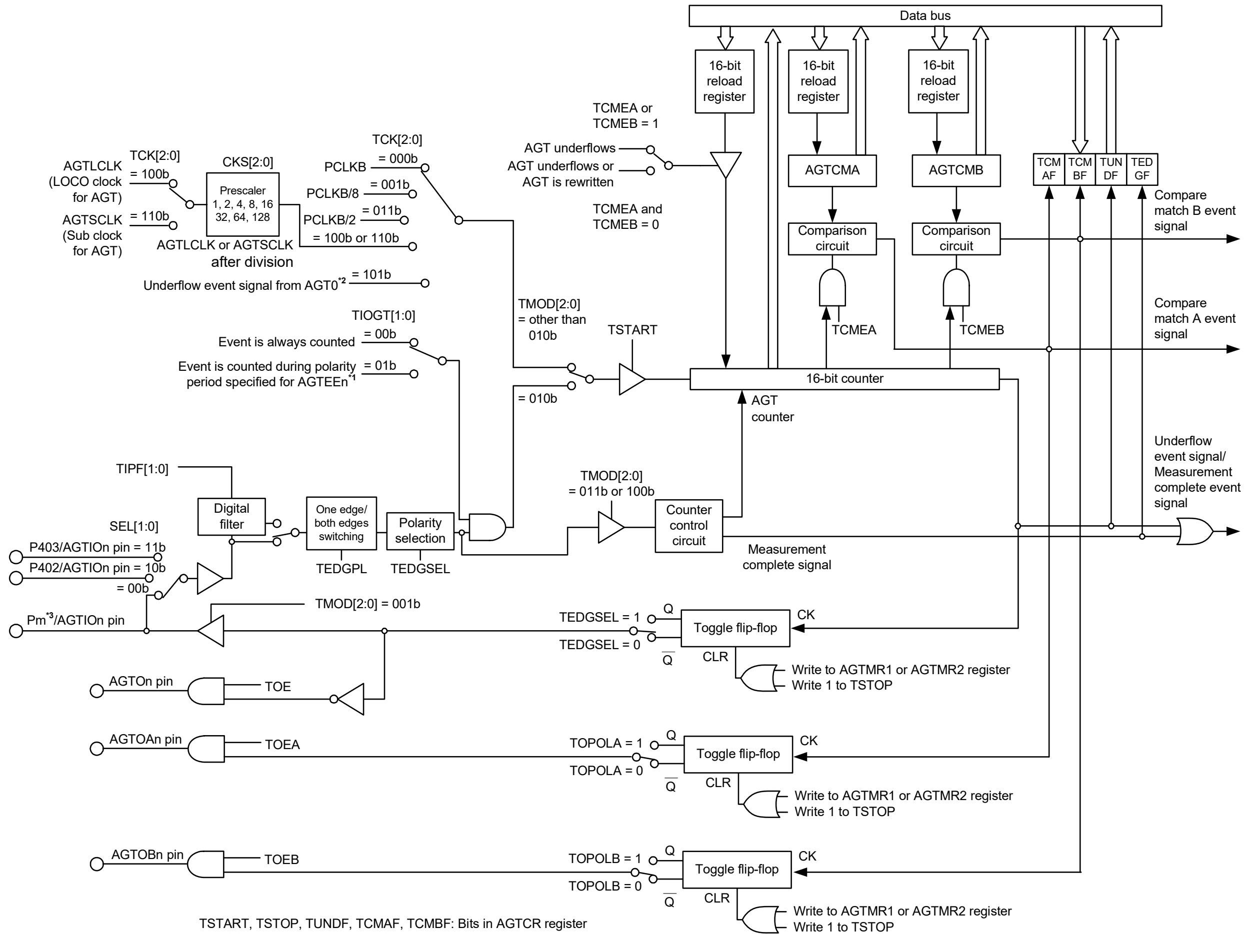
This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated in the same address, and can be accessed with the AGT register.

Table 25.1 lists the AGT specifications, Figure 25.1 shows a block diagram, and Table 25.2 lists the I/O pins.

Table 25.1 AGT specifications

Parameter		Specifications
Operating modes	Timer mode	The count source is counted
	Pulse output mode	The count source is counted and the output is inverted at each timer underflow
	Event counter mode	An external event is counted
	Pulse width measurement mode	An external pulse width is measured
	Pulse period measurement mode	An external pulse period is measured
Count source (Operating clock)[*2]		PCLKB, PCLKB/2, PCLKB/8, AGTLCLK, AGTLCLK/2, AGTLCLK/4, AGTLCLK/8, AGTLCLK/16, AGTLCLK/32, AGTLCLK/64, AGTLCLK/128, AGTSCLK, AGTSCLK/2, AGTSCLK/4, AGTSCLK/8, AGTSCLK/16, AGTSCLK/32, AGTSCLK/64, AGTSCLK/128, or underflow signal of AGT0[*1] selectable.
Interrupt/Event link function (Output)		• Underflow event signal or measurement complete event signal – When the counter underflows – When the measurement of the active width of the external input (AGTIO) is complete in pulse width measurement mode – When the set edge of the external input (AGTIO) is input in pulse period measurement mode • Compare match A event signal – When the values of AGT and AGTCMA matched (Compare match A function enabled) • Compare match B event signal – When the values of AGT and AGTCMB matched (Compare match B function enabled).
Selectable functions		• Compare match function One or two of the compare match A and B registers is selectable.

Note 1. AGT0 cannot use the AGT0 underflow signal. AGT1 connects directly with the underflow event signal from the AGT0 timer.
Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.



TSTART, TSTOP, TUNDF, TCMAF, TCMBF: Bits in AGTCR register
TEDGSEL, TOE, TIPF[1:0], TIOGT[1:0]: Bits in AGTIOC register
TMOD[2:0], TEDGPL, TCK[2:0], Bits in AGTMR1 register
TCMEA, TOEA, TOPOLA, TCMEB, TOEB, TOPOLB: Bits in AGTCMSR register
SEL[1:0]: Bits in AGTIOSEL register
CKS[2:0] : Bit in AGTMR2 register

Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGT0 cannot use AGT underflow event. AGT1 uses the underflow of AGT0.
Note 3. m = 100, 301, 407, and 705 (AGT0), m = 204, 400, and 901 (AGT1).

Figure 25.1 AGT block diagram

Table 25.2 AGT I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input for AGT
AGTIOn*1	Input*1/output	External event input and pulse output for AGT
AGTOn	Output	Pulse output for AGT
AGTOAn	Output	Output compare match A output for AGT
AGTOBn	Output	Output compare match B output for AGT

Note: Channel number (n = 0, 1).
Note 1. AGTIO can also be used in Deep Software Standby mode.

AGTIO can be controlled by the VBTICTLR register.

For more information, see section 12.2.2, VBATT Input Control Register (VBTICTLR) and section 20.5.5, I/O Buffer Specification.

25.2 Register Descriptions

25.2.1 AGT Counter Register (AGT)

Address(es): AGT0.AGT 4008 4000h, AGT1.AGT 4008 4100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Description	Setting range	R/W
b15 to b0	16-bit counter and reload register [*1], [*2]	0000h to FFFFh	R/W

Note 1. When 1 is written to the TSTOP bit in the AGTCR register, the 16-bit counter is forcibly stopped and set to FFFFh.

Note 2. When the TCK[2:0] bit setting in the AGTMR1 register is other than 001b (PCLKB/8) or 011b (PCLKB/2), if the AGT register is set to 0000h, a request signal to the ICU, the DTC, and the ELC is generated once immediately after the count starts. The AGTOn and AGTIOn outputs are toggled.
When the AGT register is set to 0000h in event counter mode, regardless of the value of bits TCK[2:0], a request signal to the ICU, the DTC, and the ELC is generated once immediately after the count starts.
In addition, the AGTOn output toggles even during a period other than the specified count period. When the AGT register is set to 0001h or more, a request signal is generated each time AGT underflows.

AGT is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter change according to the TSTART bit in the AGTCR register and TCMEA/TCMEB bit in the AGTCMSR register. For details, see section 25.3.1, Reload Register and Counter Rewrite Operation. The AGT register can be set with a 16-bit memory manipulation instruction.

25.2.2 AGT Compare Match A Register (AGTCMA)

Address(es): AGT0.AGTCMA 4008 4002h, AGT1.AGTCMA 4008 4102h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Description	Setting range	R/W
b15 to b0	16-bit compare match A data is stored.[*1]	0000h to FFFFh	R/W

Note 1. Set the AGTCMA register to FFFFh when the compare match A is not used.

The AGTCMA register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see section 25.3.2, Reload Register and Compare Register A/B Rewrite Operation. The AGTCMA register can be set by a 16-bit memory manipulation instruction.

25.2.3 AGT Compare Match B Register (AGTCMB)

Address(es): AGT0.AGTCMB 4008 4004h, AGT1.AGTCMB 4008 4104h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Description	Setting range	R/W
b15 to b0	16-bit compare match B data is stored.*1	0000h to FFFFh	R/W

Note 1. Set the AGTCMB register to FFFFh when compare match B is not used.

The AGTCMB register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register B change in accordance with the TSTART bit in the AGTCR register. For details, see section 25.3.2, Reload Register and Compare Register A/B Rewrite Operation. The AGTCMB register can be set by a 16-bit memory manipulation instruction.

25.2.4 AGT Control Register (AGTCR)

Address(es): AGT0.AGTCR 4008 4008h, AGT1.AGTCR 4008 4108h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCMBF	TCMAF	TUNDF	TEDGF	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TSTART	AGT Count Start*2	0: Count stops 1: Count starts.	R/W
b1	TCSTF	AGT Count Status Flag*2	0: Count stopped 1: Count in progress.	R
b2	TSTOP	AGT Count Forced Stop*1	0: Writing is invalid 1: The count is forcibly stopped.	W
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	TEDGF	Active Edge Judgment Flag	0: No active edge received 1: Active edge received.	R/(W)*3
b5	TUNDF	Underflow Flag	0: No underflow 1: Underflow.	R/(W)*3
b6	TCMAF	Compare Match A Flag	0: No match 1: Match.	R/(W)*3
b7	TCMBF	Compare Match B Flag	0: No match 1: Match.	R/(W)*3

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, the TSTART and TCSTF bits are initialized at the same time. The pulse output level is also initialized. The read value is 0.
Note 2. For information on using the TSTART and TCSTF bits, see section 25.4.1, Count Operation Start and Stop Control.
Note 3. Only 0 can be written to clear the flag.

TSTART bit (AGT Count Start)

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When this bit is set to 1, the TCSTF bit is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF bit is set to 0 (count stopped) in synchronization with the count source. For details, see section 25.4.1, Count Operation Start and Stop Control.

TCSTF flag (AGT Count Status Flag)

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF flag is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF flag is set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit.

TSTOP bit (AGT Count Forced Stop)

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active Edge Judgment Flag)

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input (AGTIO) is complete in pulse width measurement mode.
- When the set edge of the external input (AGTIO) is input in pulse period measurement mode

[Clearing condition]

- When 0 is written to this flag by a program.

TUNDF flag (Underflow Flag)

The TUNDF flag indicates that the counter underflowed.

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

TCMAF flag (Compare Match A Flag)

The TCMAF flag indicates that compare match A was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMA register.

[Clearing condition]

- When 0 is written to this flag by software.

TCMBF flag (Compare Match B Flag)

The TCMBF flag indicates that compare match B was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMB register.

[Clearing condition]

- When 0 is written to this flag by software.

25.2.5 AGT Mode Register 1 (AGTMR1)

Address(es): AGT0.AGTMR1 4008 4009h, AGT1.AGTMR1 4008 4109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TCK[2:0]			TEDGPL	TMOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	TMOD[2:0]	Operating Mode*3	b2 b0 0 0 0: Timer mode 0 0 1: Pulse output mode 0 1 0: Event counter mode 0 1 1: Pulse width measurement mode 1 0 0: Pulse period measurement mode. Other settings are prohibited.	R/W
b3	TEDGPL	Edge Polarity*4	0: Single-edge 1: Both-edge.	R/W
b6 to b4	TCK[2:0]	Count Source*1, *2, *5	b6 b4 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified in CKS[2:0] bits in AGTMR2 register 1 0 1: Underflow event signal from AGT0*6 1 1 0: Divided clock AGTSCLK specified in CKS[2:0] bits in AGTMR2 register. Other settings are prohibited.	R/W
b7	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note: Write access to the AGTMR1 register initializes the output from the AGTOn, AGTIOn, AGTOAn and AGTOBn pins of the AGT (n = 0, 1). For details on the output level at initialization, see the description of section 25.2.7, AGT I/O Control Register (AGTIOC).

Note 1. When event counter mode is selected, the external input (AGTIOn) is selected as the count source regardless of the setting of TCK[2:0] bits.

Note 2. Do not switch count sources during count operation. Only switch count sources when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped).

Note 3. The operating mode can only be changed when the count is stopped while both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped). Do not change the operating mode during count operation.

Note 4. The TEDGPL bit is enabled only in event counter mode.

Note 5. When running AGT in Software Standby and Deep Software Standby modes, set AGTSCLK or AGTLCLK (TCK[2:0] = 100b or 110b) as the count source.

Note 6. AGT0 cannot use AGT0 underflow (setting prohibited). AGT1 uses the underflow of AGT0.

25.2.6 AGT Mode Register 2 (AGTMR2)

Address(es): AGT0.AGTMR2 4008 400Ah, AGT1.AGTMR2 4008 410Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CKS[2:0]	AGTSCLK/AGTLCLK Count Source Clock Frequency Division Ratio *1, *2, *3	b2 b0 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	LPM	Low Power Mode	0: Normal mode 1: Low-power mode.	R/W

Note 1. Do not rewrite CKS[2:0] during count operation. Only rewrite the CKS[2:0] bits when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped).

Note 2. When count source is AGTSCLK/AGTLCLK, the switch of CKS[2:0] is valid.

Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] are set to 000b, and wait for 1 cycle of the count source.

LPM bit (Low Power Mode)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set 1 to operate in low power. When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- AGT: Read AGT register twice. Only the second reading of data is valid.
- AGT, AGTCMA, AGTCMB, and AGTCR: Allow at least 2 cycles of the count source clock when writing to the register.

25.2.7 AGT I/O Control Register (AGTIOC)

Address(es): AGT0.AGTIOC 4008 400Ch, AGT1.AGTIOC 4008 410Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDGSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TEDGSEL	I/O Polarity Switch	Function varies depending on the operating mode. See Table 25.3 and Table 25.4. The TEDGSEL bit switches the AGTO output polarity and the AGTIO input/output edge and polarity. In pulse output mode, it only controls the polarity of AGTOn output and AGTIOn output. AGTOn output and AGTIOn output are initialized when the AGTMR1 register is written and the TSTOP bit in the AGTCR register is written with 1.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	TOE	AGTOn Output Enable	0: AGTOn output disabled 1: AGTOn output enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TIPF[1:0]	Input Filter*3	b5 b4 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32. These bits specify the sampling frequency of the filter for the AGTIOn input. If the input to the AGTIOn pin is sampled and the value matches three successive times, that value is taken as the input value.	R/W
b7, b6	TIOGT[1:0]	Count Control*1,*2,*4	b7 b6 0 0: Event is always counted 0 1: Event is counted during polarity period specified for AGTEEn. Other settings are prohibited.	R/W

Note 1. When AGTEEn pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.
Note 2. Bits TIOGT[1:0] are enabled only in event counter mode.
Note 3. When event counter mode operation is performed during Software Standby and Deep Software Standby modes, the digital filter function cannot be used.
Note 4. When using in Deep Software Standby mode, set TIOGT[1:0] = 00b (event is always counted).

Table 25.3 AGTIOn I/O edge and polarity switching

Operating mode	Function
Timer mode	Not used
Pulse output mode	0: Output is started at high (initialization level: high) 1: Output is started at low (initialization level: low).
Event counter mode	0: Count on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

Table 25.4 AGTOn output polarity switching

Operating mode	Function
All modes	0: Output is started at low (initialization level: low) 1: Output is started at high (initialization level: high).

25.2.8 AGT Event Pin Select Register (AGTISR)

Address(es): AGT0.AGTISR 4008 400Dh, AGT1.AGTISR 4008 410Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	EEPS	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	EEPS	AGTEEn Polarity Selection	0: An event is counted during the low-level period 1: An event is counted during the high-level period.	R/W

Bit	Symbol	Bit name	Description	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

25.2.9 AGT Compare Match Function Select Register (AGTCMSR)

Address(es): AGT0.AGTCMSR 4008 400Eh, AGT1.AGTCMSR 4008 410Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TCMEA	Compare Match A Register Enable*1, *2	0: Compare match A register disabled 1: Compare match A register enabled.	R/W
b1	TOEA	AGTOAn Output Enable*1, *2	0: AGTOAn output disabled 1: AGTOAn output enabled.	R/W
b2	TOPOLA	AGTOAn Polarity Select*1, *2	0: AGTOAn output is started on low 1: AGTOAn output is started on high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	TCMEB	Compare Match B Register Enable*1, *2	0: Compare match B register disabled 1: Compare match B register enabled.	R/W
b5	TOEB	AGTOBn Output Enable*1, *2	0: AGTOBn output disabled 1: AGTOBn output enabled.	R/W
b6	TOPOLB	AGTOBn Polarity Select*1, *2	0: AGTOBn output is started on low 1: AGTOBn output is started on high.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Do not rewrite the AGTCMSR register during a count operation. Only rewrite the AGTCMSR register when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count is stopped).

Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.

25.2.10 AGT Pin Select Register (AGTIOSEL)

Address(es): AGT0.AGTIOSEL 4008 400Fh, AGT1.AGTIOSEL 4008 410Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TIES	—	—	SEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	SEL[1:0]	AGTIO Pin Select*1,*3	b1 b0 0 0: Select Pm*2/AGTIO as AGTIO Pm/AGTIO can not be used as AGTIO input pin in Deep Software Standby mode. 0 1: Setting prohibited 1 0: Select P402/AGTIO as AGTIO P402/AGTIO can be used as AGTIO input pin in Deep Software Standby mode. P402/AGTIOn is input only. It cannot be used for output. 1 1: Select P403/AGTIO as AGTIO. P403/AGTIO can be used as AGTIO input pin in Deep Software Standby mode. P403/AGTIOn is input only. It cannot be used for output.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b4	TIES	AGTIO Input Enable	0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. P402/AGTIO and P403/AGTIO can be used as external event input pins for the AGT in Deep Software Standby mode. Pm*2/AGTIO cannot be used as external event input pins for the AGT in Deep Software Standby mode. P402/AGTIO and P403/AGTIO are input only.
When Pm/AGTIO is selected, you must set the Port mn Pin Function Select (PmnPFS) register. See section 20, I/O Ports.
Note 2. m = 100, 301, 407, and 705 (AGT0), m = 204, 400, and 901 (AGT1).
Note 3. When P402/AGTIO and P403/AGTIO are selected, you must set the VBTICTLR register. See section 12, Battery Backup Function.

The AGTIOSEL register sets the AGTIO pin when using the AGTIO in Deep Software Standby mode and Software Standby mode. The AGTIOSEL register can be set with an 8-bit memory manipulation instruction.

SEL[1:0] bits (AGTIO Pin Select*1,*3)

The SEL[1:0] bits select the AGTIO pin function.

TIES bit (AGTIO Input Enable)

The TIES bit enables or disables an external event input.

25.3 Operation

25.3.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA bit and TCMEB bit are 0 (compare match A/B registers are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or TCMEB bit is 1 (compare match A register or compare match B register is valid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the underflow of the counter.

Figure 25.2 and Figure 25.3 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.

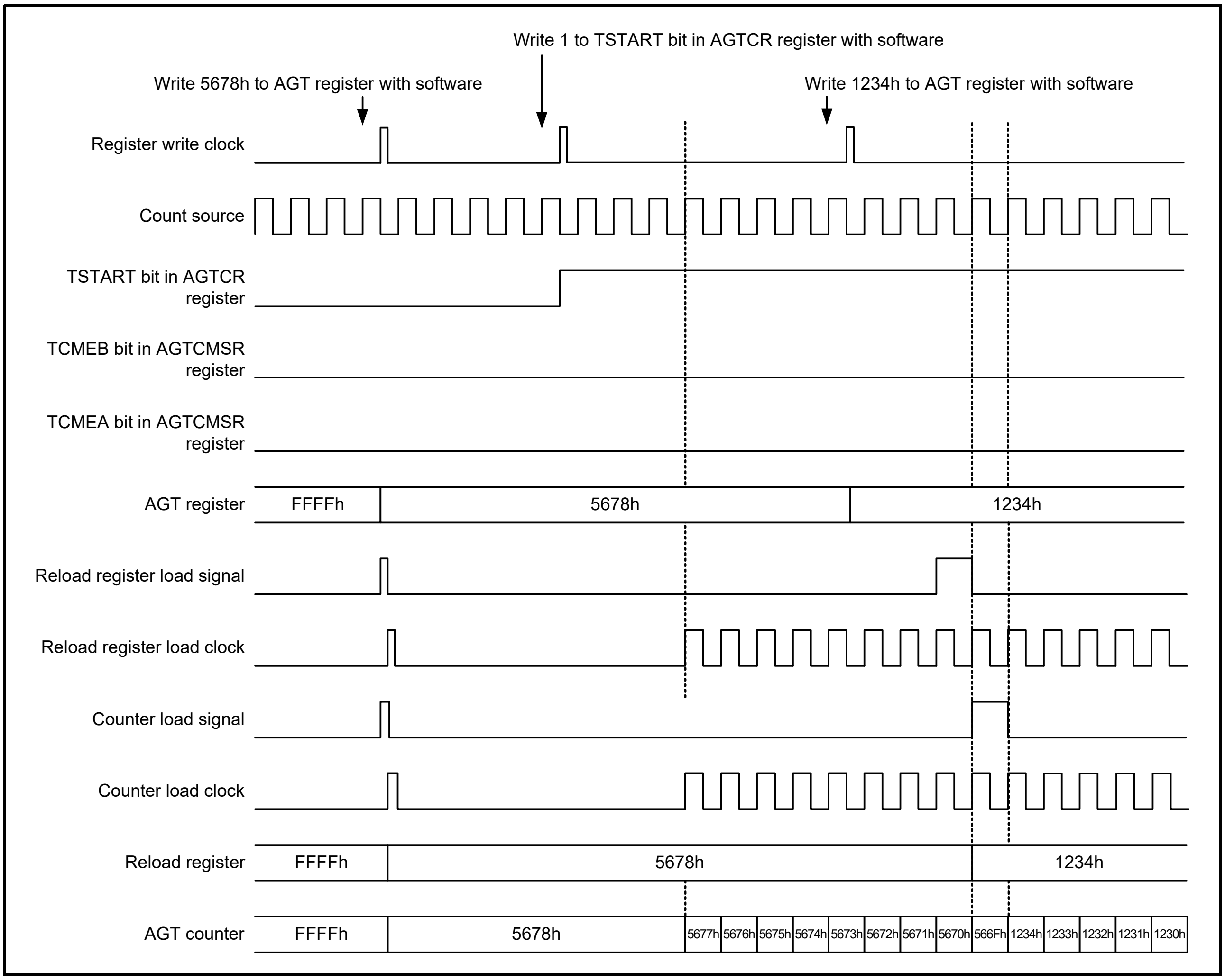


Figure 25.2 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when compare match register A or B is invalid

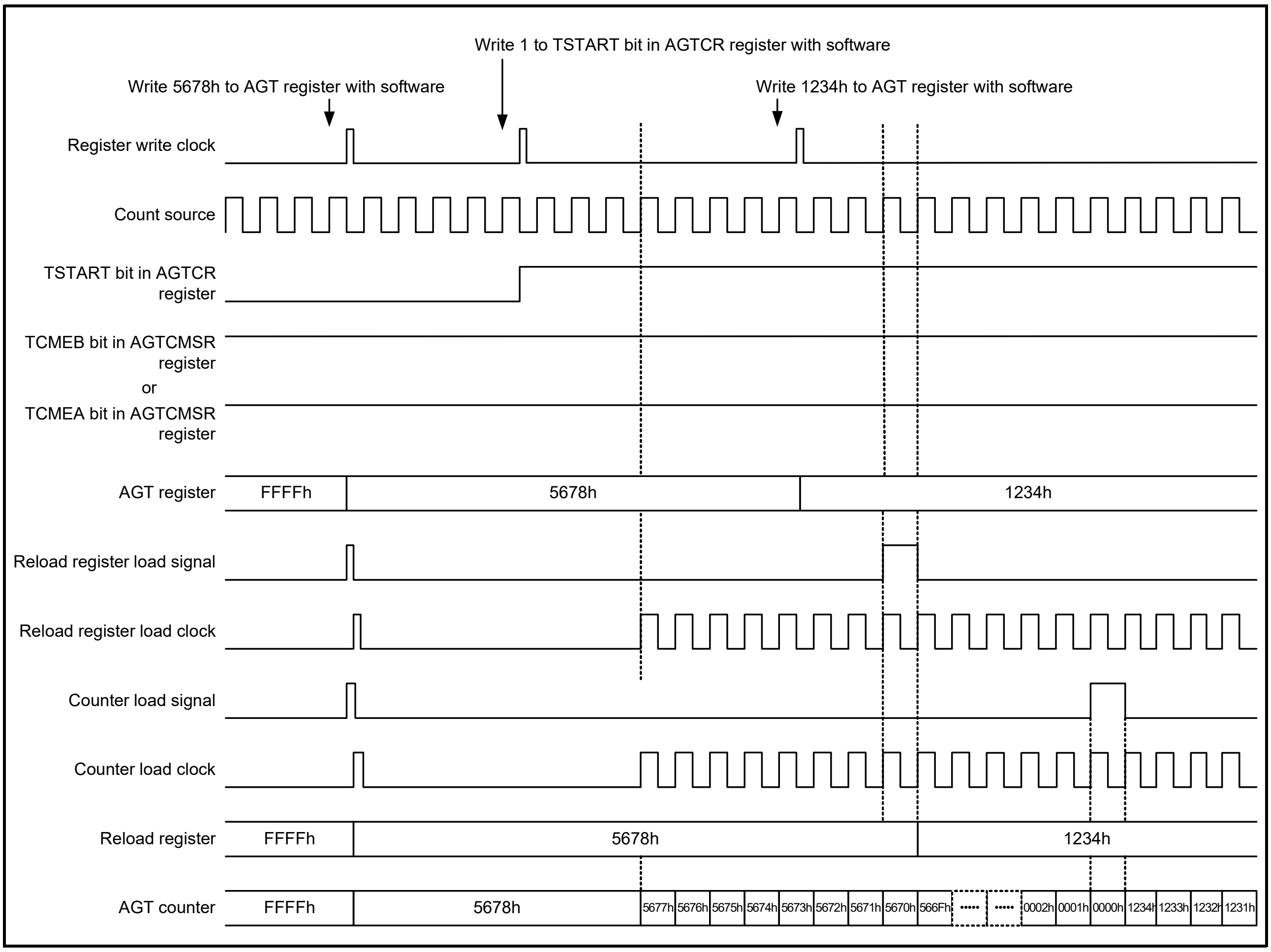


Figure 25.3 **Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when compare match register A or B is valid**

25.3.2 Reload Register and Compare Register A/B Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to compare register A/B depends on the value of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and compare register A/B. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the compare register in synchronization with the underflow of the counter.

Figure 25.4 shows the timing of rewrite operation with TSTART bit value for compare register A. Compare register B is of the same timing as compare register A.

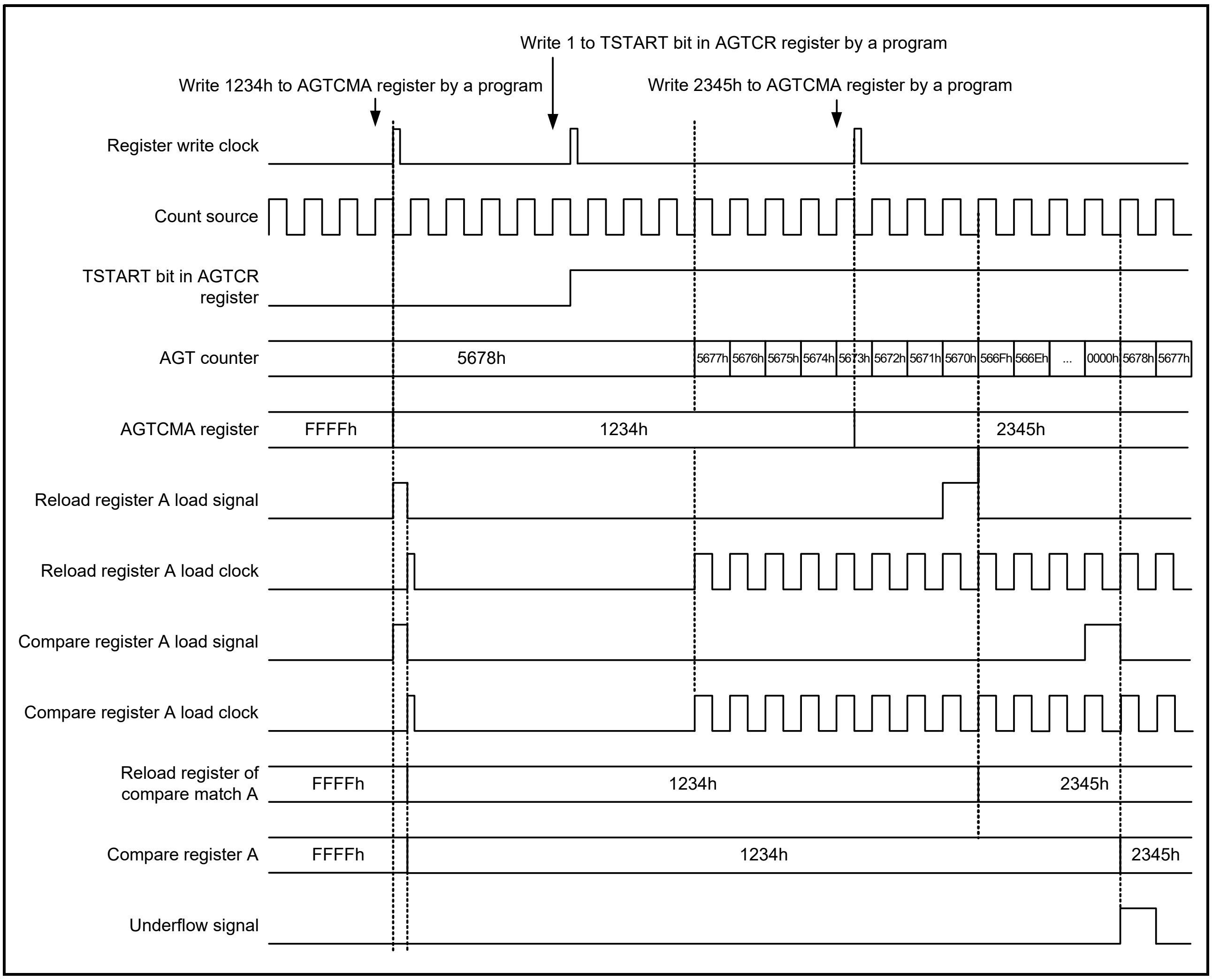


Figure 25.4 Timing of rewrite operation with TSTART bit value for compare register A

25.3.3 Timer Mode

In timer mode, the AGT counter is decremented by the count source selected in bits TCK[2:0] in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0000h and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 25.5 shows the operation example in timer mode.

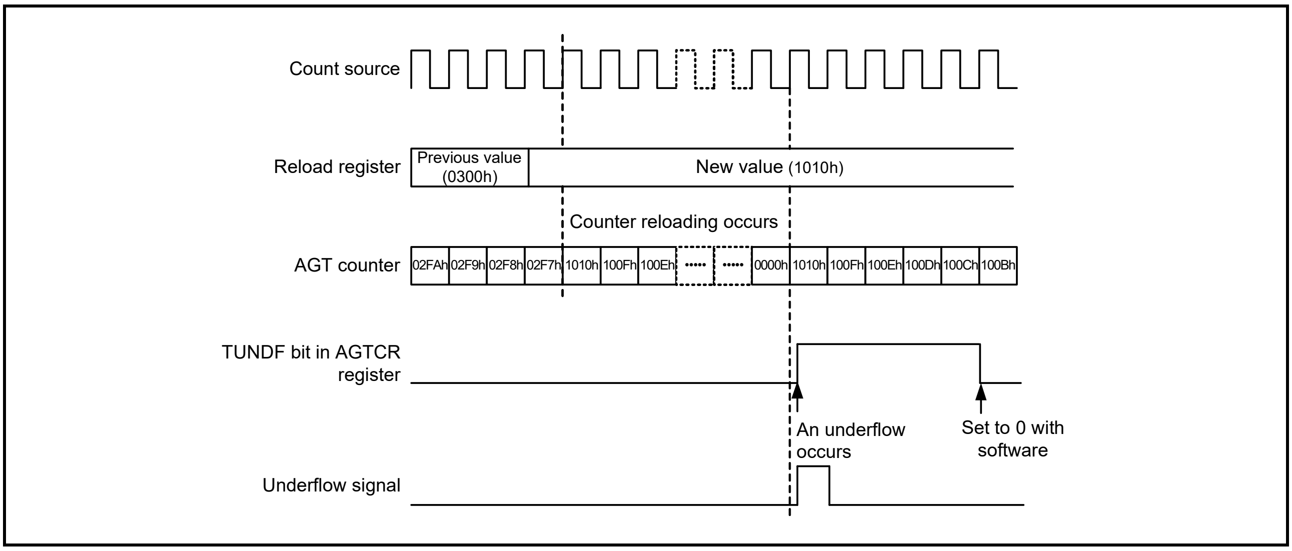


Figure 25.5 **Operation example in timer mode**

25.3.4 Pulse Output Mode

In pulse output mode, the counter is decremented by the count source selected in TCK[2:0] bits in the AGTMR1 register, and the output level of pins AGTIOn and AGTOn pin inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0000h and the next count source is input, an underflow occurs and an interrupt request is generated. In addition, a pulse can be output from the AGTIOn and AGTOn pins. The output level is inverted each time an underflow occurs. The pulse output from the AGTOn pin can be stopped with the TOE bit in the AGTIOC register. The output level can be selected with the TEDGSEL bit in the AGTIOC register.

Figure 25.6 shows the operation example in pulse output mode.

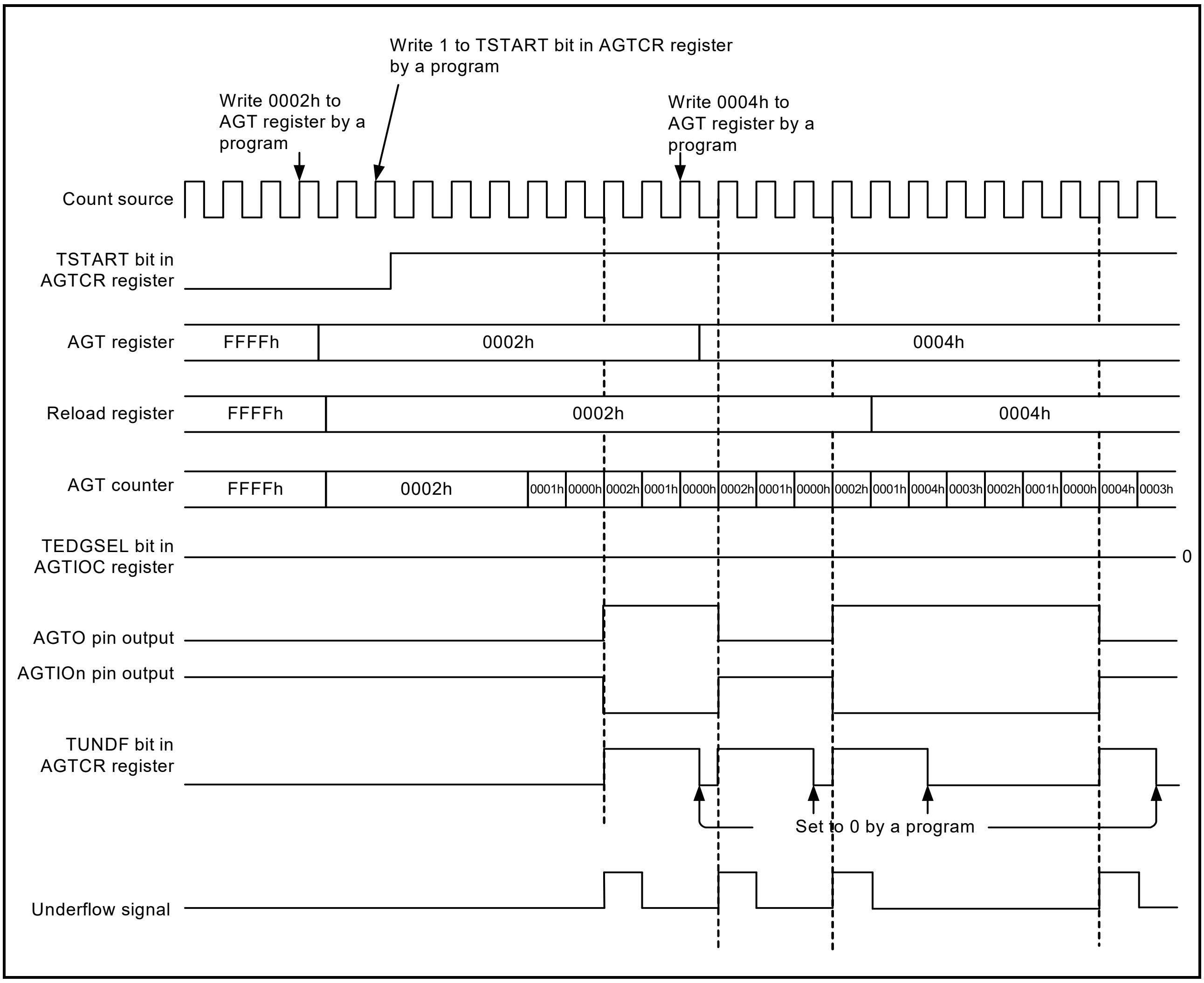


Figure 25.6 Operation example in pulse output mode

25.3.5 Event Counter Mode

In event counter mode, the counter is decremented by an external event signal input to the AGTIOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC and AGTISR registers. In addition, the filter function for the AGTIOn input can be specified with bits TIPF[1:0] in the AGTIOC register. The output from the AGTOn pin can be toggled even in event counter mode.

Figure 25.7 shows the operation example in event counter mode.

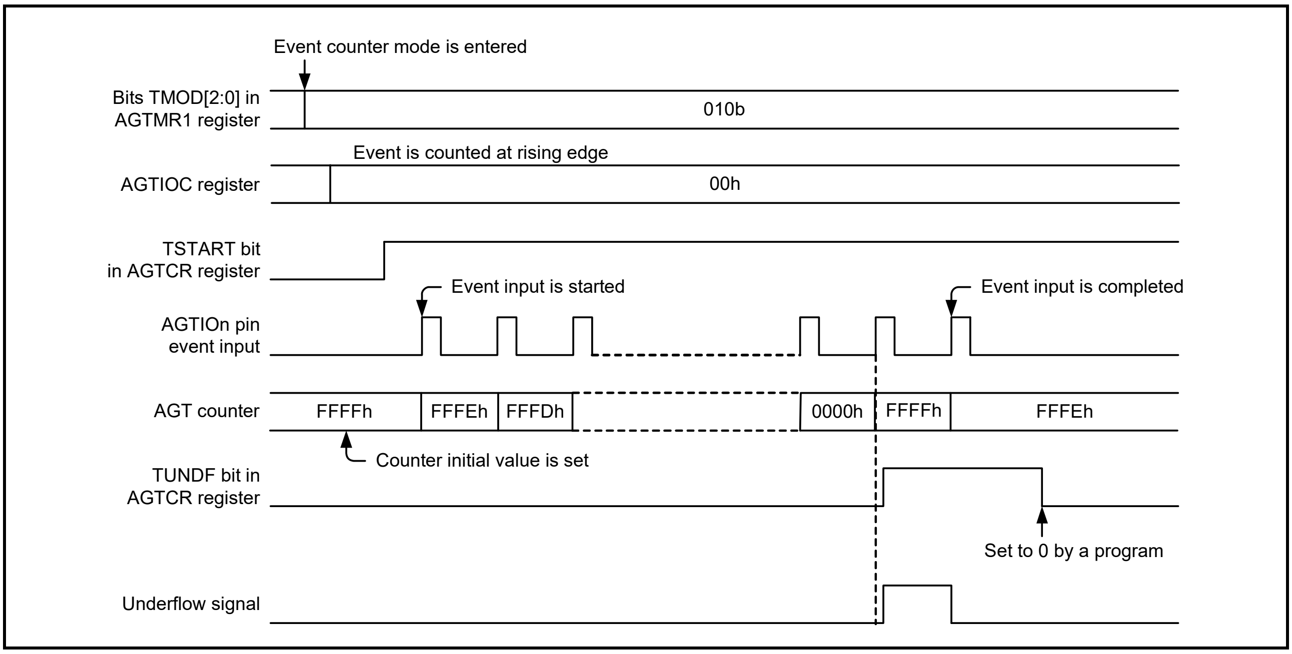


Figure 25.7 Operation example 1 in event counter mode

Figure 25.8 shows an operation example for counting during the specified period in event counter mode (bits TIOGT[1:0] in the AGTIOC register are set to 01b).

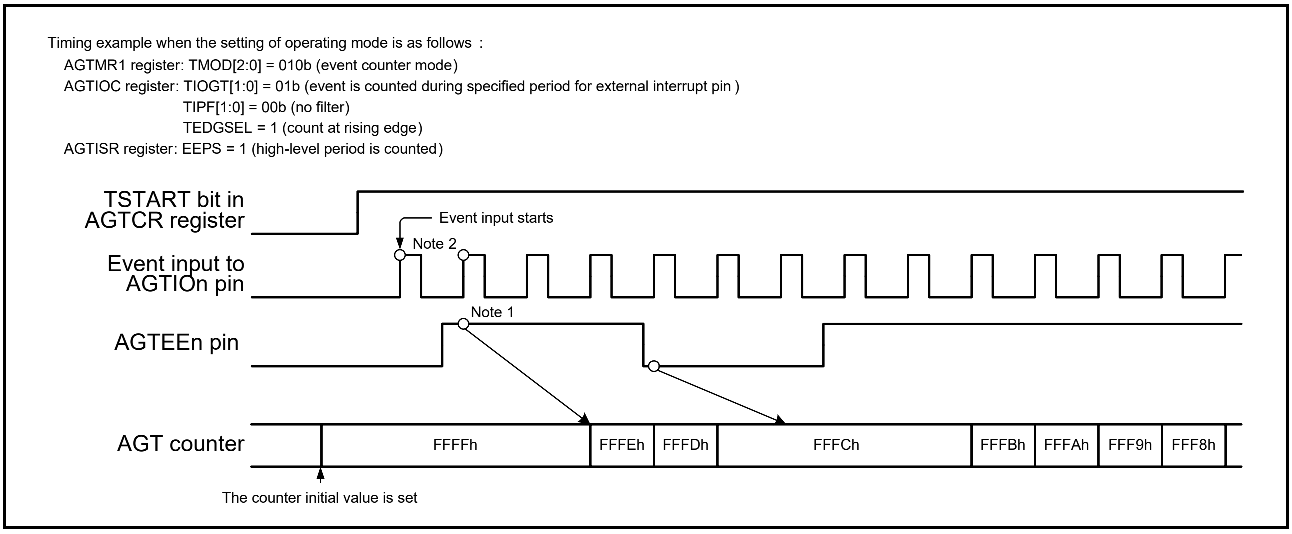

Timing example when the setting of operating mode is as follows :
AGTMR1 register: TMOD[2:0] = 010b (event counter mode)
AGTIOC register: TIOGT[1:0] = 01b (event is counted during specified period for external interrupt pin)
TIPF[1:0] = 00b (no filter)
TEDGSEL = 1 (count at rising edge)
AGTISR register: EEPS = 1 (high-level period is counted)

Figure 25.8 Operation example 2 in event counter mode

Note 1. To control synchronization, there is a delay of 2 cycles of the count source until count operation is affected. It is also possible that the count start timing is shifted by 1 cycle because of the phase difference between the AGTEEn and the sampling clock.

Note 2. Count operation can be performed for 2 cycles of the count source immediately after the count starts, depending on the previous state before the count stops.
To disable the count for 2 cycles immediately after the count starts, write 1 to the TSTOP bit in the AGTCR register to initialize the internal circuit, and then complete the operation settings before starting the count operation.

25.3.6 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the AGTIOn pin is measured. When the level specified in the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the counter is decremented by the count source selected by TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTIOn pin ends, the counter is stopped, the TEDGF bit in the AGTCR register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF bit in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 25.9 shows the operation example in pulse width measurement mode.

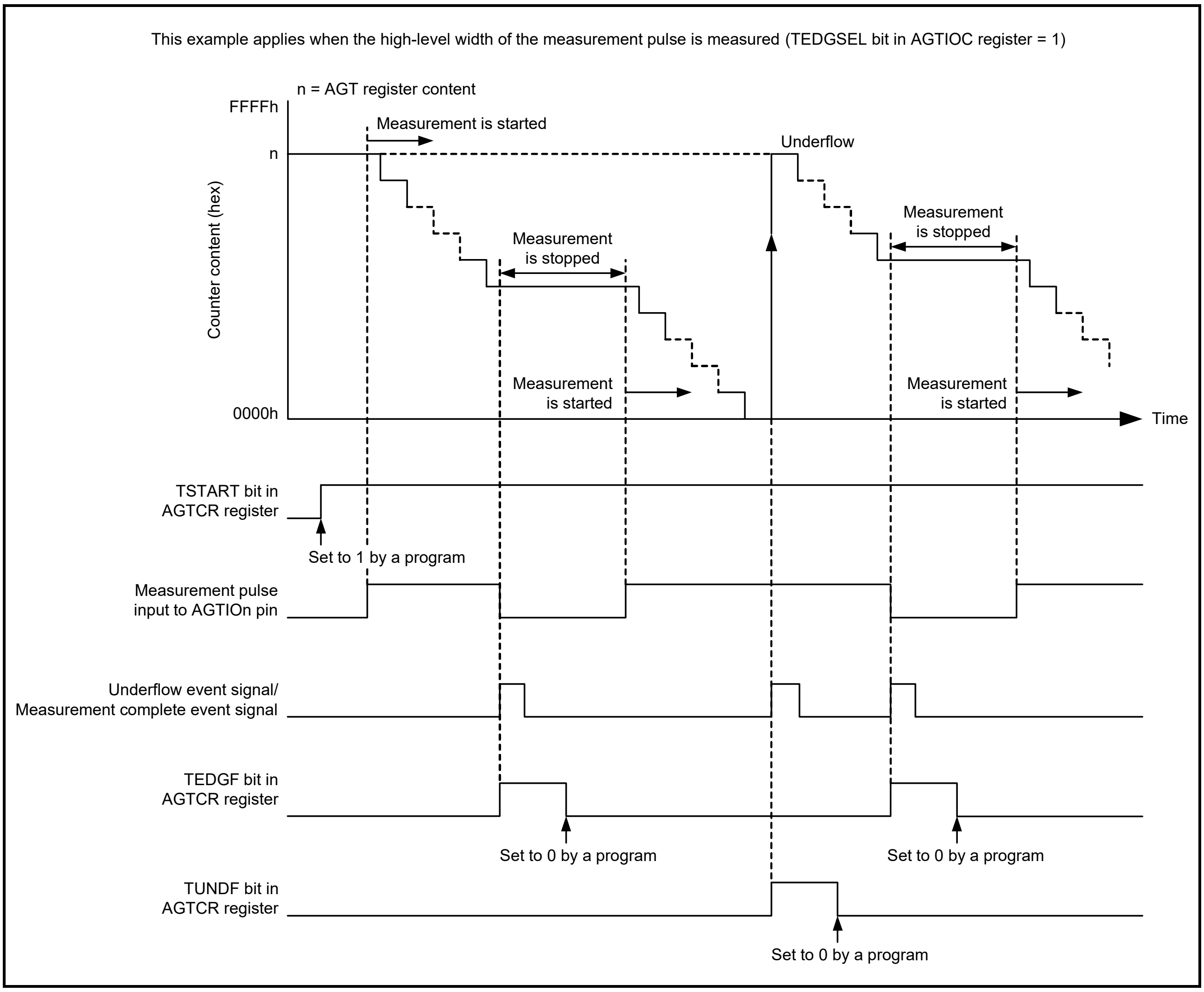


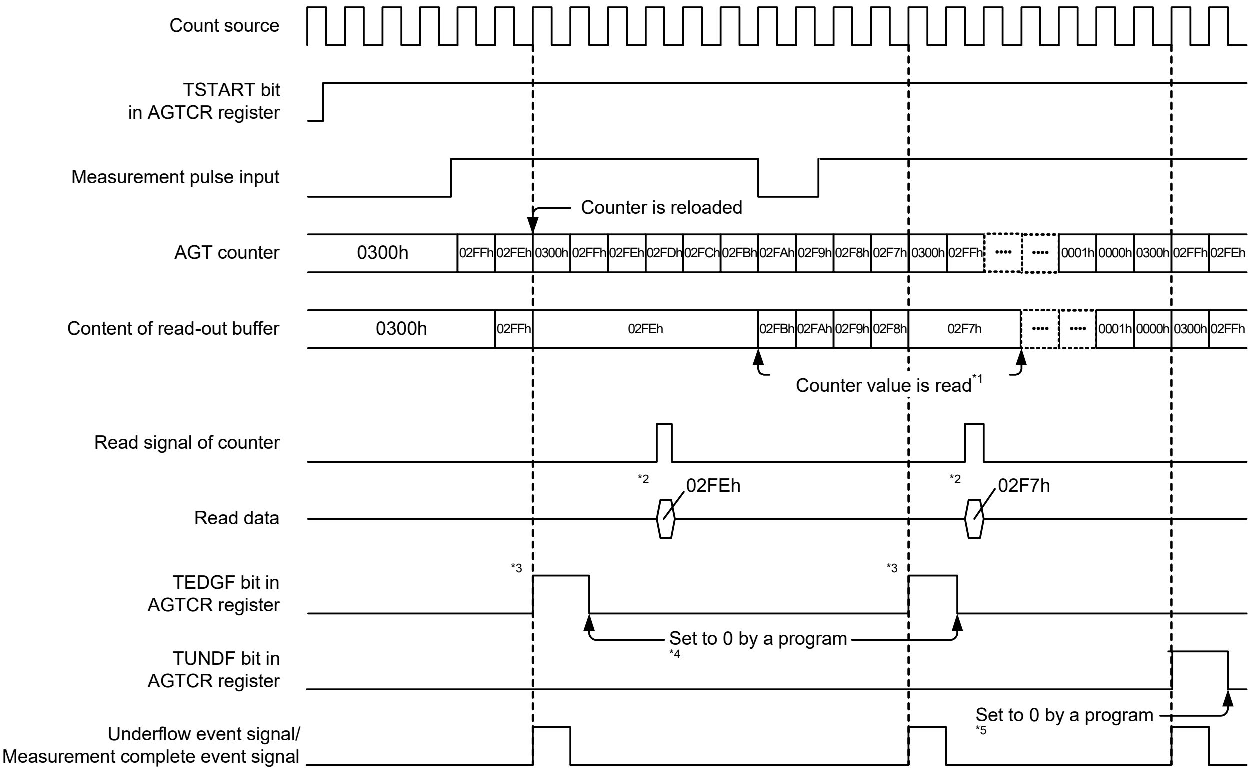
Figure 25.9 Operation example in pulse width measurement mode

25.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTIOn pin is measured. The counter is decremented by the count source selected with bits TCK[2:0] in the AGTMR1 register. When a pulse with the level specified in the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF bit in the AGTCR register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (AGT register) is read at this time and the difference from the reload value (see section 25.4.5, How to Calculate Event Number, Pulse Width, and Pulse Period) is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF bit in the AGTCR register is set to 1 (underflow) and an interrupt request is generated.

Figure 25.10 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source are measured. Also, the low-level and high-level widths must both be longer than the period of the count source. If a pulse period shorter than these conditions is input, the input might be ignored.



This example applies when the initial value of the AGT register is set to 0300h, the TEDGSEL bit in the AGTIOC register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. Reading from the AGT register must be performed during the period from when the TEDGF bit is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the AGT register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.

Note 2. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.

Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF bit in the AGTCR register is set to 1 (active edge received).

Note 4. To set to 0 with software, write 0 to the TEDGF bit in the AGTCR register using an 8-bit memory manipulation instruction.

Note 5. To set to 0 with software, write 0 to the TUNDF bit in the AGTCR register using an 8-bit memory manipulation instruction.

Figure 25.10 Operation example in pulse period measurement mode

25.3.8 Compare Match Function

The compare match function detects matches between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA bit or the TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected in bits TCK[2:0] in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF bit in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See section 25.3.1, Reload Register and Counter Rewrite Operation for details. In addition, the output level of the AGTOAn and AGTOBn pins is inverted by the match and by the underflow. The output level can be selected with the TOPOLA or TOPOLB bit in the AGTCMSR register.

Figure 25.11 shows the operation example in compare match mode.

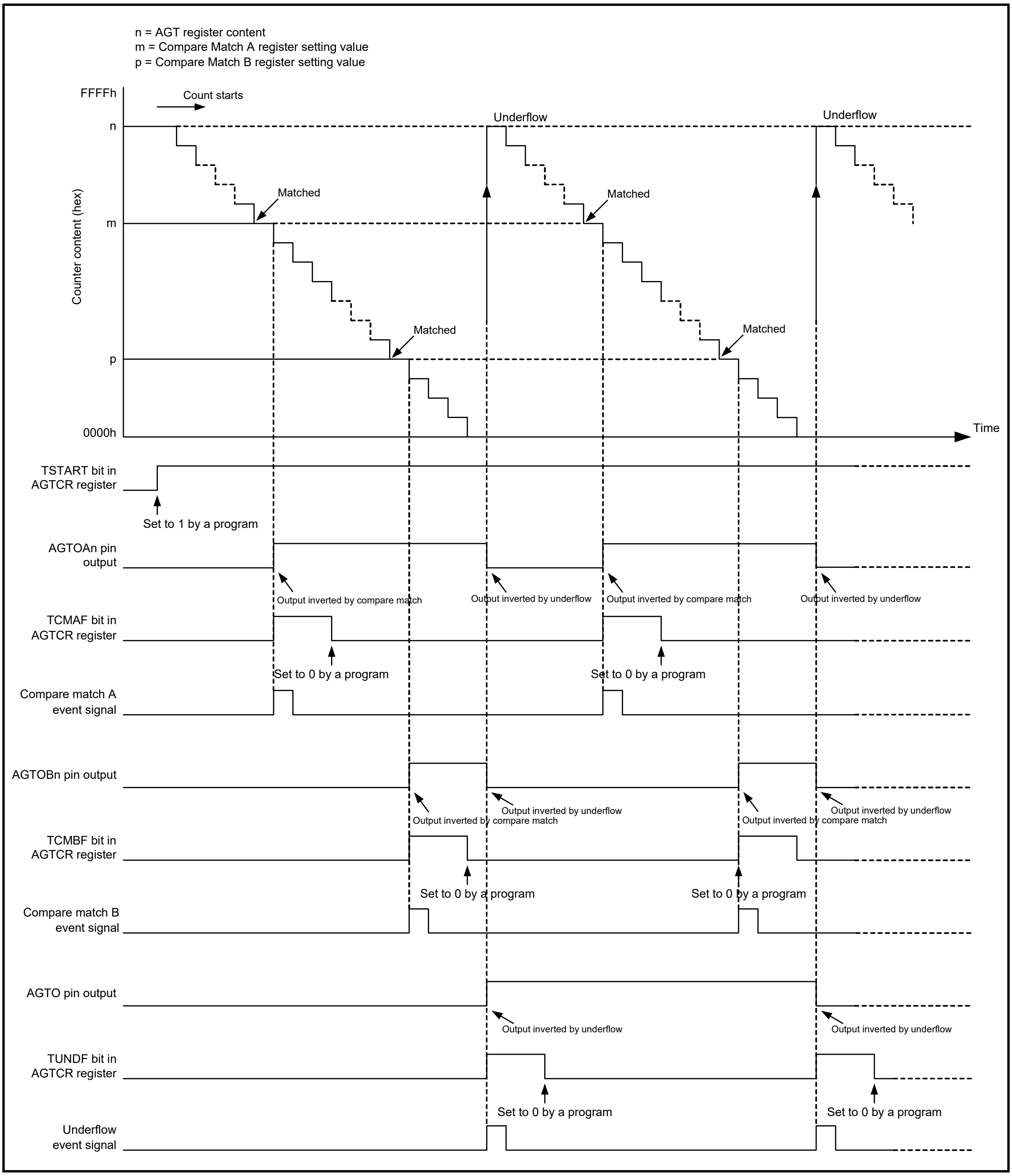


Figure 25.11 Operation example in compare match mode (TOPOLA = 0, TOPOLB = 0)

25.3.9 Output Settings for Each Mode

Table 25.5 to Table 25.8 list the states of pins AGTOn, AGTIOn, AGTOAn, and AGTOBn in each mode.

Table 25.5 AGTOn pin setting

Operating mode	AGTIOC register		AGTOn pin output
	TOE bit	TEDGSEL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 25.6 AGTIOn pin setting

Operating mode	AGTIOC register	AGTIOn pin I/O
	TEDGSEL bit	
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

Table 25.7 AGTOAn pin setting

Operating mode	AGTCMSR register		AGTOAn pin output
	TOEA bit	TOPOLA bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

Table 25.8 AGTOBn pin setting (1 of 2)

Operating mode	AGTCMSR register		AGTOBn pin output
	TOEB bit	TOPOLB bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)

Table 25.8 AGTOBn pin setting (2 of 2)

	AGTCMSR register		
Operating mode	TOEB bit	TOPOLB bit	AGTOBn pin output
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

25.3.10 Standby Mode

The AGT can operate in Software Standby and Deep Software Standby modes. Set it to Software Standby mode or Deep Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

Table 25.9 and Table 25.10 show the settings that can be used in Software Standby and Deep Software Standby modes.

Table 25.9 Usable settings for AGT0 in Software Standby and Deep Software Standby modes

Operating mode	TCK[2:0] bits of AGTMR1 register	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse output mode	100b or 110b	AGTLCLK or AGTSCLK	—
Event counter mode	- (Invalid)	AGTIOn	—
Pulse width measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse period measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—

Table 25.10 Usable settings AGT1 in Software Standby and Deep Software Standby modes

Operating mode	TCK[2:0] bits of AGTMR1 register	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b or 101b [*1]	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare match A/B
Pulse output mode	100b or 110b or 101b [*1]	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare match A/B
Event counter mode	- (Invalid)	AGTIOn	• Underflow • Compare match A/B
Pulse width measurement mode	100b or 110b or 101b [*1]	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge
Pulse period measurement mode	100b or 110b or 101b [*1]	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge

Note: Release of Software Standby mode or Deep Software Standby mode is only AGT1.

Note 1. Only when AGT0 operates in Table 25.9.

25.3.11 Interrupt Sources

The AGT has three interrupt sources described in Table 25.11.

Table 25.11 AGT interrupt sources

Name	Interrupt source	DMAC/DTC activation
AGTn_AGTI	• When the counter underflows • When measurement of the active width of the external input (AGTIO) is completed in pulse width measurement mode • When the set edge of the external input (AGTIO) is input in pulse period measurement mode.	Possible
AGTn_AGTCMAI	When the values of AGT and AGTCMA match	Possible
AGTn_AGTCMBI	When the values of AGT and AGTCMB match	Possible

Note: Channel number (n = 0 or 1).

25.3.12 Event Signal Output to ELC

The AGT uses the Event Link Controller (ELC) to perform a link operation to a specified module using the interrupt request signal as the event signal. The AGT outputs compare match A, compare match B, and underflow/measurement complete signals as event signals. For details, see section 19, Event Link Controller (ELC).

25.4 Usage Notes

25.4.1 Count Operation Start and Stop Control

- When the operating mode (see Table 25.1) is set to other than the event counter mode, or the count source is set to other than AGT0 underflow (TCK[2:0] = 101b)
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGT*1 other than the TCSTF flag until this bit is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 3 cycles of the count source. When the TCSTF flag is set to 0, the count stops. Do not access the registers associated with AGT*1 other than the TCSTF flag until this bit is set to 0.
  - Clear the interrupt register before changing the TSTART bit from 0 to 1. See section 14, Interrupt Controller Unit (ICU) for details.

Note 1. Registers associated with AGT: AGT, AGTCMA, AGTCMB, AGTCR, AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR.

- When the operating mode (see Table 25.1) is set to event counter mode, or the count source is set to AGT0 underflow (TCK[2:0] = 101b)
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF bit in the AGTCR register remains 0 (count stops) for 2 cycles of the PCLKB. Do not access the registers associated with AGT*1 other than the TCSTF bit until this bit is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for 2 cycles of the PCLKB. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with AGT*1 other than the TCSTF bit until this bit is set to 0.
  - Clear the interrupt register before changing the TSTART bit from 0 to 1. See section 14, Interrupt Controller Unit (ICU) for details.

Note 1. Registers associated with AGT: AGT, AGTCMA, AGTCMB, AGTCR, AGTMR1, AGTMR2, AGTIOC, AGTISR and AGTCMSR.

25.4.2 Access to Counter Register

When the TSTART and TCSTF bits in the AGTCR register are both 1 (count starts), allow at least 3 cycles of the count source clock between writes when writing to the AGT register successively.

25.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, AGTCMSR and AGTIOC) can be changed only when the count is stopped with both the TSTART and TCSTF bits set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with AGT operating mode are changed, the values of bits TEDGF, TUNDF, TCMAF and TCMBF are undefined. Before starting the count, write 0 to the following bits:

- TEDGF (no active edge received)
- TUNDF (no underflow)
- TCMAF (no match)
- TCMBF (no match).

25.4.4 Digital Filter

When using the digital filter, do not start the timer operation for 5 cycles of the digital filter clock after setting bits TIPF[1:0] and when the TEDGSEL bit in the AGTIOC register changes.

25.4.5 How to Calculate Event Number, Pulse Width, and Pulse Period

- In event counter mode, event number is expressed mathematically as follows:
  Event number = initial value of counter [AGT register] - counter value of active event end
- In pulse width measurement mode, pulse width is expressed mathematically as follows:
  Pulse width = counter value of stopping measurement - counter value of next stopping measurement
- In pulse period measurement mode, input pulse period is expressed mathematically as follows:
  Period of input pulse = (initial value of counter [AGT register] - reading value of the read-out buffer) + 1

25.4.6 When Count Is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the AGTCR register, do not access the following I/O registers for 1 cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

25.4.7 When Selecting AGT0 Underflow as the Count Source

Operate the AGT according to the procedures described in this section when selecting the underflow signal of AGT as the count source.

(1) Procedure for starting operation

1. Set AGT0 and AGT1.
2. Start the count operation of AGT1.
3. Start the count operation of AGT0.

(2) Procedure for stopping operation

1. Stop the count operation of AGT0.
2. Stop the count operation of AGT1.
3. Stop the count source clock of AGT1 (write 000b in the AGT1.AGTMR1.TCK[2:0] bits).

25.4.8 Reset of I/O Register

The I/O register of the AGT is not initialized by some types of resets. For details, see section 6, Resets.

25.4.9 When Selecting PCLKB, PCLKB/8, or PCLKB/2 as the Count Source

When a reset is generated, the operation of the AGT cannot be guaranteed. Set the registers associated with AGT again.

25.4.10 When Selecting AGTSCLK or AGTLCLK as the Count Source

The MSTPD2 bit in the MSTPCRD register must be set to 1 except when accessing the AGT1 registers. The MSTPD3 bit in the MSTPCRD register must be set to 1 except when accessing the AGT0 registers. When a reset occurs while MSTPD2 or MSTPD3 bit is 0, the operation of AGT1 or AGT0 cannot be guaranteed. Set the registers associated with AGT again.

25.4.11 When Switching Source Clock

When switching a clock source by changing SCKCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. Therefore, when using the AGTIO, AGTEE, or both input as external event input, the clock source should not be switched. If switching the clock source while using the external event input, extend the input pulse width by 4 clock cycles of the switched source clock cycles.

26. Realtime Clock (RTC)

26.1 Overview

The RTC has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar.

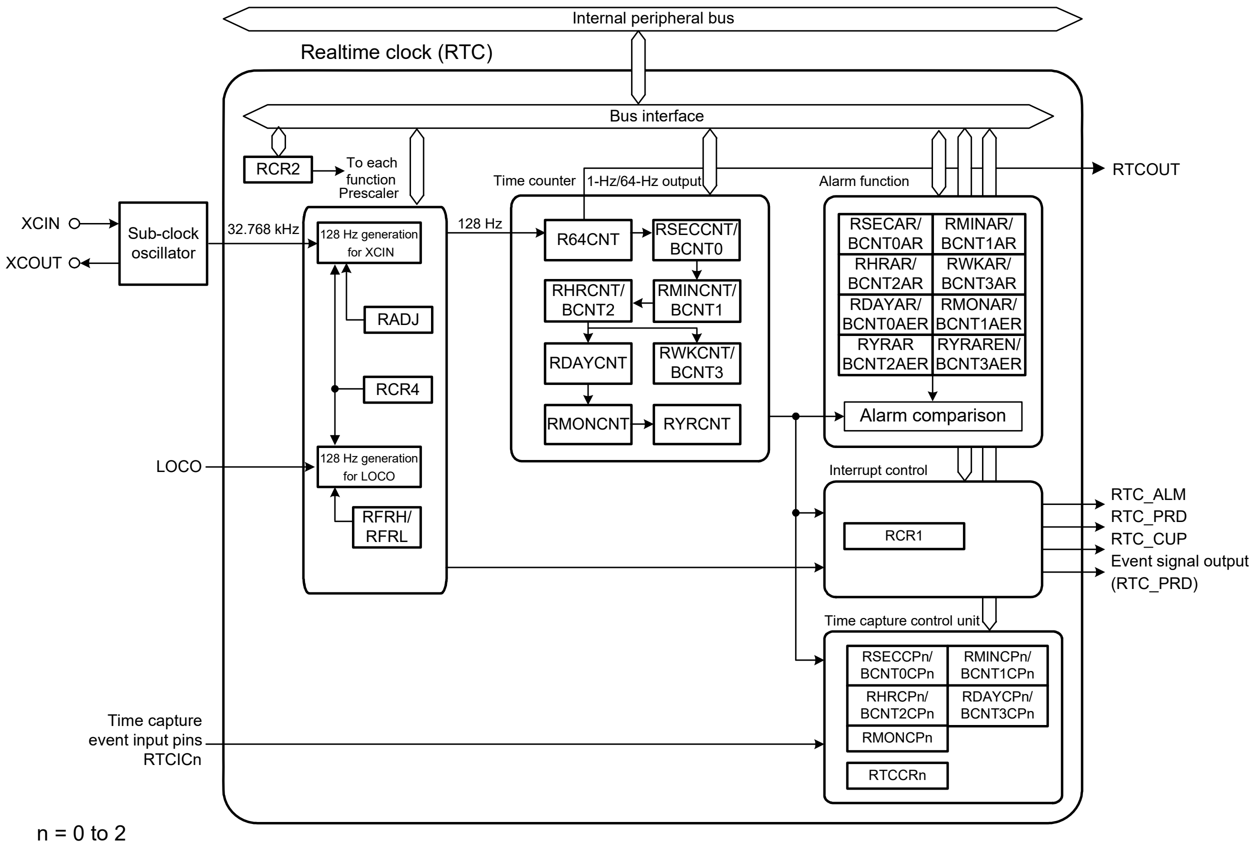
The sub-clock oscillator or LOCO can be selected as the count source of the time counters. The RTC uses a 128-Hz clock acquired by dividing the count source by a prescaler. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted by 1/128 second.

Table 26.1 lists the RTC specifications, Figure 26.1 shows a block diagram, and Table 26.2 lists the I/O pins.

Table 26.1 RTC specifications

Parameter	Specifications
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock oscillator (XCIN) or LOCO
Clock and calendar functions	• Calendar count mode Year, month, date, day of week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to 1 minute) Automatic adjustment function for leap years • Binary count mode Count seconds in 32 bits, binary display • Shared by both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz) Clock error correction function Clock (1-Hz/64-Hz) output.
Interrupts	• Alarm interrupt (RTC_ALM) As an alarm interrupt condition, selectable for comparison with the following: Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected Binary count mode: Each bit of the 32-bit binary counter • Periodic interrupt (RTC_PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period • Carry interrupt (RTC_CUP) An interrupt is generated at either of the following conditions: - When a carry from the 64-Hz counter to the second counter is generated - When the 64-Hz counter is changed and the R64CNT register is read at the same time • Return from Software Standby mode or Deep Software Standby mode can be performed by an alarm interrupt or periodic interrupt.
Time capture function	• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or the 32-bit binary counter value is captured.
Event link function	Periodic event output (RTC_PRD)

Note 1. The frequency of the peripheral module clock (PCLKB) must be ≥ the frequency of the count source clock.



R64CNT: 64-Hz counter
RSECCNT/BCNT0: Second Counter/Binary Counter 0
RMINCNT/BCNT1: Minute Counter/Binary Counter 1
RHRCNT/BCNT2: Hour Counter/Binary Counter 2
RWKCNT/BCNT3: Day-of-week Counter/Binary Counter 3
RDAYCNT: Date Counter
RMONCNT: Month Counter
RYRCNT: Year Counter

RCR1: Rtc Control Register 1
RCR2: Rtc Control Register 2
RCR4: Rtc Control Register 4
RADJ: Time Error Adjustment Register
RSECAR/BCNT0AR: Second Alarm Register/binary Counter 0 Alarm Register

RFRH/RFRL: Frequency Register
RMINAR/BCNT1AR: Minute Alarm Register/Binary Counter 1 Alarm Register
RHRAR/BCNT2AR: Hour Alarm Register/Binary Counter 2 Alarm Register
RWKAR/BCNT3AR: Day-of-week Alarm Register/Binary Counter 3 Alarm Register
RDAYAR/BCNT0AER: Date Alarm Register/Binary Counter 0 Alarm Enable Register
RMONAR/BCNT1AER: Month Alarm Register/Binary Counter 1 Alarm Enable Register
RYRAR/BCNT2AER: Year Alarm Register/Binary Counter 2 Alarm Enable Register
RYRAREN/BCNT3AER: Year Alarm Enable Register/Binary Counter 3 Alarm Enable Register
RTCCRn: Time Capture Control Register n
RSECCPn/BCNT0CPn: Second Capture Register n/bcnt0 Capture Register n
RMINCPn/BCNT1CPn: Minute Capture Register n/bcnt1 Capture Register n
RHRCPn/BCNT2CPn: Hour Capture Register n/bcnt2 Capture Register n
RDAYCPn/BCNT3CPn: Date Capture Register n/bcnt3 Capture Register n
RMONCPn: Month Capture Register n

Figure 26.1 RTC block diagram

Table 26.2 RTC pin configuration

Pin name	I/O	Function
XCIN	Input	Connect a 32.768-kHz crystal to these pins
XCOUT	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in Deep Software Standby mode
RTCIC0	Input	Time capture event input pins. RTCIC0 to RTCIC2 can be controlled by the VBTICTLR register. For more information, see section 12, Battery Backup Function and section 20, I/O Ports.
RTCIC1	Input	
RTCIC2	Input	

26.2 Register Descriptions

Write or read from the RTC registers as described in section 26.6.5, Notes on Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations, for example while the RCR2.START bit is 1, the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate.

Note: A reset generated while writing to a register might destroy the register value. In addition, do not allow the chip to enter Software Standby mode or Deep Software Standby mode immediately after setting any of these registers. For details, see section 26.6.4, Transitions to Low Power Modes after Setting Registers.

26.2.1 64-Hz Counter (R64CNT)

Address(es): RTC.R64CNT 4004 4000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	F64HZ	64 Hz	Indicates the state between 1 Hz and 64 Hz of the sub-second digit	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0.	R

The R64CNT counter is used in both calendar count mode and in binary count mode. The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock. The state in the sub-second range can be confirmed by reading this counter.

This counter is cleared to 00h by an RTC software reset or an execution of a 30-second adjustment. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode

Address(es): RTC.RSECCNT 4004 4002h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Count	Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W

Bit	Symbol	Bit name	Description	R/W
b6 to b4	SEC10[2:0]	10-Second Count	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RSECCNT counter sets and counts the BCD-coded second value. It counts the carries generated once per second in the 64-Hz counter.

The setting range is decimal 00 to 59. The RTC does not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation using the START bit in RCR2.

To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode

Address(es): RTC.BCNT0 4004 4002h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT0 is a read/write 32-bit binary counter b7 to b0 that performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode

Address(es): RTC.RMINCNT 4004 4004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RMINCNT counter sets and counts the BCD-coded minute value. It counts the carries generated once every minute in the second counter.

A value from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode

Address(es): RTC.BCNT1 4004 4004h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1 is a read/write 32-bit binary counter b15 to b8 that performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode

Address(es): RTC.RHRCNT 4004 4006h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	10-Hour Count	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	AM/PM select for time counter setting. 0: AM 1: PM.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RHRCNT counter sets and counts the BCD-coded hour value. It counts the carries generated once per hour in the minute counter. The specifiable time differs based on the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 — from 00 to 11 (in BCD)
- When the RCR2.HR24 bit is 1 — from 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode

Address(es): RTC.BCNT2 4004 4006h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT2 is a read/write 32-bit binary counter b23 to b16 that performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation using the START bit in

RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode

Address(es): RTC.RWKCNT 4004 4008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited.	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W

The RWKCNT counter sets and counts in the coded day-of-week value. It counts the carries generated once per day in the hour counter. A value from 0 through 6 can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode

Address(es): RTC.BCNT3 4004 4008h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3 is a read/write 32-bit binary counter b31 to b24 that performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.6 Day Counter (RDAYCNT)

Address(es): RTC.RDAYCNT 4004 400Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Count	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DATE10[1:0]	10-Day Count	Counts from 0 to 3 once per carry from the ones place	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RDAYCNT counter is used in calendar count mode to set and count the BCD-coded date value. It counts the carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year. Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. When specifying a value, the range of specifiable days depends on the month and whether the year is a leap year. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.7 Month Counter (RMONCNT)

Address(es): RTC.RMONCNT 4004 400Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Count	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	10-Month Count	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMONCNT counter is used in calendar count mode to set and count the BCD-coded month value. It counts the carries generated once per month in the date counter.

A value from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.8 Year Counter (RYRCNT)

Address(es): RTC.RYRCNT 4004 400Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	YR1[3:0]	1-Year Count	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	10-Year Count	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRCNT counter is used in calendar count mode to set and count the BCD-coded year value. It counts the carries generated once per year in the month counter.

A value from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time.

26.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

(1) In calendar count mode

Address(es): RTC.RSECAR 4004 4010h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds.	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds.	R/W
b7	ENB	ENB	0: The register value is not compared with the RSECCNT counter value 1: The register value is compared with the RSECCNT counter value.	R/W

RSECAR is an alarm register associated with the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

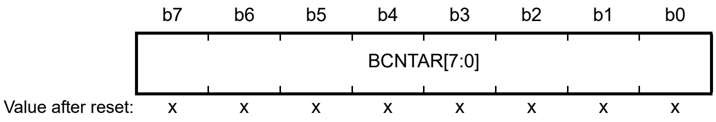
- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR

- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RSECAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is cleared to 00h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT0AR 4004 4010h



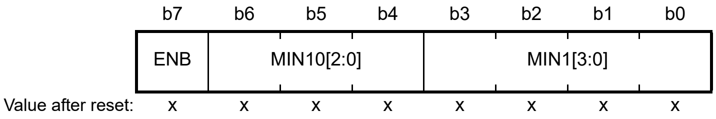
x: Undefined

BCNT0AR is a read/write alarm register associated with the 32-bit binary counter b7 to b0. This register is cleared to 00h by an RTC software reset.

26.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode

Address(es): RTC.RMINAR 4004 4012h



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	0: The register value is not compared with the RMINCNT counter value 1: The register value is compared with the RMINCNT counter value.	R/W

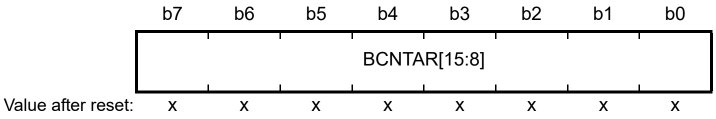
RMINAR is an alarm register associated with the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RMINAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is cleared to 00h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT1AR 4004 4012h



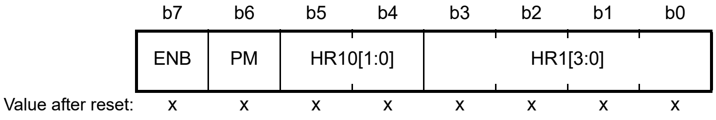
x: Undefined

BCNT1AR is a read/write alarm register associated with the 32-bit binary counter from b15 to b8. This register is cleared to 00h by an RTC software reset.

26.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode

Address(es): RTC.RHRAR 4004 4014h



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HR10[1:0]	10 Hours	Value for the tens place of hours	R/W
b6	PM	PM	Time alarm setting: 0: AM 1: PM.	R/W
b7	ENB	ENB	0:The register value is not compared with the RHRCNT counter value 1: The register value is compared with the RHRCNT counter value.	R/W

RHRAR is an alarm register associated with the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 — From 00 to 11 (in BCD)
- When the RCR2.HR24 bit is 1 — From 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. When the RCR2.HR24 bit is 0, be sure to set the PM bit. When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect. This register is cleared to 00h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT2AR 4004 4014h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT2AR is a read/write alarm register associated with the 32-bit binary counter b23 to b16. This register is cleared to 00h by an RTC software reset.

26.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode

Address(es): RTC.RWKAR 4004 4016h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited.	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RWKCNT counter value 1: The register value is compared with the RWKCNT counter value.	R/W

RWKAR is an alarm register associated with the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values all match, the IR flag associated with the RTC_ALM interrupt is set to 1. RWKAR values from 0 through 6 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is cleared to 00h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT3AR 4004 4016h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3AR is a read/write alarm register associated with the 32-bit binary counter b31 to b24. This register is cleared to 00h by an RTC software reset.

26.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode

Address(es): RTC.RDAYAR 4004 4018h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RDAYCNT counter value 1: The register value is compared with the RDAYCNT counter value.	R/W

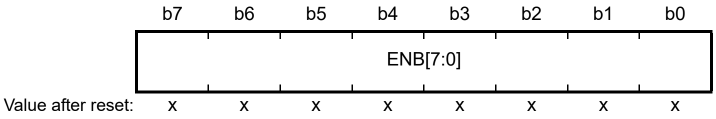
RDAYAR is an alarm register associated with the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RDAYAR values from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is cleared to 00h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT0AER 4004 4018h



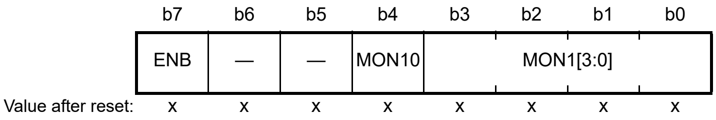
x: Undefined

BCNT0AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b7 to b0. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]) and, when all match, the IR flag associated with the RTC_ALM interrupt becomes 1. This register is cleared to 00h by an RTC software reset.

26.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

(1) In calendar count mode

Address(es): RTC.RMONAR 4004 401Ah



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RMONCNT counter value 1: The register value is compared with the RMONCNT counter value.	R/W

RMONAR is an alarm register associated with the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. RMONAR values from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is cleared to 00h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT1AER 4004 401Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b15 to b8. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]) and, when all match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is cleared to 00h by an RTC software reset.

26.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode

Address(es): RTC.RYRAR 4004 401Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RYRAR is an alarm register associated with the BCD-coded year counter RYRCNT. RYRAR values from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is cleared to 0000h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT2AER 4004 401Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

BCNT2AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b23 to b16. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]) and, when all match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is cleared to 0000h by an RTC software reset.

26.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode

Address(es): RTC.RYRAREN 4004 401Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value 1: The register value is compared with the RYRCNT counter value.	R/W

When the ENB bit in RYRAREN is set to 1, the RYRAR value is compared with the RYRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is cleared to 00h by an RTC software reset.

(2) In binary count mode

Address(es): RTC.BCNT3AER 4004 401Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b31 to b24. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]) and, when all match, the IR flag associated with the RTC_ALM interrupt is set to 1. This register is cleared to 00h by an RTC software reset.

26.2.17 RTC Control Register 1 (RCR1)

Address(es): RTC.RCR1 4004 4022h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled 1: An alarm interrupt request is enabled.	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled 1: A periodic interrupt request is enabled.	R/W
b3	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1 Hz 1: RTCOUT outputs 64 Hz.	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	b7 b4 0 1 1 0: Generate periodic interrupt every 1/256 second*1 0 1 1 1: Generate periodic interrupt every 1/128 second 1 0 0 0: Generate periodic interrupt every 1/64 second 1 0 0 1: Generate periodic interrupt every 1/32 second 1 0 1 0: Generate periodic interrupt every 1/16 second 1 0 1 1: Generate periodic interrupt every 1/8 second 1 1 0 0: Generate periodic interrupt every 1/4 second 1 1 0 1: Generate periodic interrupt every 1/2 second 1 1 1 0: Generate periodic interrupt every 1 second 1 1 1 1: Generate periodic interrupt every 2 seconds. Other settings: No periodic interrupts are generated.	R/W

Note 1. When LOCO is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0110b, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and in binary count mode. Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits are updated before proceeding.

AIE bit (Alarm Interrupt Enable)

The AIE bit enables or disables alarm interrupt requests.

If the times indicated in the counters and alarm settings match in Deep Software Standby mode, the MCU returns from the mode regardless of the AIE bit value.

CIE bit (Carry Interrupt Enable)

The CIE bit enables or disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE bit (Periodic Interrupt Enable)

The PIE bit enables or disabled a periodic interrupt.

If the periods indicated in the counters and PES[3:0] settings match in Deep Software Standby mode, the MCU returns from the mode regardless of the PIE bit value.

RTCOS bit (RTCOUT Output Select)

The RTCOS bit selects the RTCOUT output period. The RTCOS bit must be rewritten while the count operation is stopped (the RCR2.START bit is 0) and the RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. For details on controlling the I/O ports, see section 20.5.1, Procedure for Specifying the Pin Functions.

PES[3:0] bits (Periodic Interrupt Select)

The PES[3:0] bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified in these bits.

26.2.18 RTC Control Register 2 (RCR2)

(1) In calendar count mode

Address(es): RTC.RCR2 4004 4024h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	START	Start	0: Prescaler and time counter are stopped 1: Prescaler and time counter operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Invalid (writing 0 has no effect) 1: The prescaler and the target registers for RTC software reset *1 are initialized. • In reading 0: Normal time operation in progress, or RTC software reset has completed 1: RTC software reset in progress.	R/W
b2	ADJ30	30-Second Adjustment	• In writing 0: Invalid (writing 0 has no effect) 1: 30-second adjustment is executed. • In reading 0: Normal time operation in progress, or 30-second adjustment has completed 1: 30-second adjustment in progress.	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output is disabled 1: RTCOUT output is enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*2	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*2	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.	R/W
b6	HR24	Hours Mode	0: RTC operates in 12-hour mode 1: RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select	0: Calendar count mode 1: Binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRy, RSECCPy/BCNT0CPy, RMINCPy/BCNT1CPy, RHRCPy/BCNT2CPy, RDAYCPy/BCNT3CPy, RMONCPy, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When LOCO is selected, the setting of this bit is disabled.

The RCR2 register is related to hours mode, automatic adjustment function, enabling RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START bit (Start)

The START bit stops or restarts the prescaler or time counter operation. This bit is updated in synchronization with the next cycle of the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software.

When 1 is written to the RESET bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. Check that this bit is 0 before proceeding.

ADJ30 bit (30-Second Adjustment)

The ADJ30 bit is for 30-second adjustment.

When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.

The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment is complete. If 1 is written to the ADJ30 bit, check that the bit is 0 before proceeding. When the 30-second adjustment is performed, the prescaler and R64CNT are also reset. The ADJ30 bit is cleared to 0 by an RTC software reset.

RTCOE bit (RTCOUT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.

When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

AADJE bit (Automatic Adjustment Enable*2)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is cleared to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select*2)

The AADJP bit selects the automatic-adjustment period.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.

The AADJP bit is cleared to 0 by an RTC software reset.

HR24 bit (Hours Mode)

The HR24 bit specifies whether the RTC operates in 12- or 24-hour mode.

Use the START bit to stop counting before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

CNTMD bit (Count Mode Select)

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is complete.

For details on initial settings, see section 26.3.1, Outline of Initial Settings of Registers after Power On.

(2) In binary count mode

Address(es): RTC.RCR2 4004 4024h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	—	AADJP	AADJE	RTCOE	—	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	START	Start	0: The 32-bit binary counter, 64-Hz counter, and prescaler are stopped 1: The 32-bit binary counter, 64-Hz counter, and prescaler are in normal operation.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Invalid (writing 0 has no effect) 1: The prescaler and the target registers for RTC software reset*1 are initialized. • In reading 0: Normal time operation in progress, or RTC software reset has completed 1: RTC software reset in progress.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output is disabled 1: RTCOUT output is enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*2	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*2	0: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 32 seconds 1: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 8 seconds.	R/W
b6	—	Reserved	This bit is undefined. The write value should be 0.	R/W
b7	CNTMD	Count Mode Select	0: Calendar count mode 1: Binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRy, RSECCPy/BCNT0CPy, RMINCPy/BCNT1CPy, RHRCPy/BCNT2CPy, RDAYCPy/BCNT3CPy, RMONCPy, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When LOCO is selected, the setting of this bit is disabled.

START bit (Start)

The START bit stops or restarts the prescaler or counter (clock) operation. The bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software.

When 1 is written to this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is 0 before proceeding.

RTCOE bit (RTCOUT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When an RTCOUT signal is to be output from an external pin, enable the port control in addition to setting this bit.

AADJE bit (Automatic Adjustment Enable)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is cleared to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

The AADJP bit selects the automatic-adjustment period.

Correction period can be selected from 32 second units or 8 second units in binary count mode.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit. The AADJP bit is cleared to 0 by an RTC software reset.

CNTMD bit (Count Mode Select)

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is complete.

For details on initial settings, see section 26.3.1, Outline of Initial Settings of Registers after Power On.

26.2.19 RTC Control Register 4 (RCR4)

Address(es): RTC.RCR4 4004 4028h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RCKSEL
Value after reset:	0	0	0	0	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RCKSEL	Count Source Select	0: Sub-clock oscillator is selected 1: LOCO is selected.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR4 register selects the count source and is used in both calendar count mode and binary count mode.

When the RCKSEL bit is set to 0, the time is counted with the sub-clock oscillator. When the bit is set to 1, the time is counted with LOCO.

RCKSEL bit (Count Source Select)

The RCKSEL bit selects the count source from the sub-clock oscillator and LOCO.

The count source must be selected only once before specifying the initial settings of the RTC registers at power on.

26.2.20 Frequency Register (RFRH/RFRL)

Address(es): RTC.RFRH.4004 402Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RFC16	Reserved	Write 0 before writing to the RFRL register after a cold start	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): RTC.RFRL.4004 402Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFC[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b15 to b0	RFC[15:0]	Frequency Comparison Value	Write 00FFh to this register when using the LOCO	R/W

RFRL is a register for controlling the prescaler when LOCO is selected.

The RTC time counter operates on a 128-Hz clock signal as the base clock. Therefore, when LOCO is selected, LOCO is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency comparison value in the RFC[15:0] bits to generate a 128-Hz clock from the LOCO frequency. Before writing to RFC[15:0] after a cold start, write 0000h to the RFRH register.

A value from 0007h through 01FFh can be specified as the frequency comparison value. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2. The operating frequency of the peripheral module clock and the LOCO should be such that the peripheral module clock is ≥ to the LOCO.

Calculation method of frequency comparison value:

RFC[15:0] = (LOCO clock frequency) / 128 - 1

When the LOCO frequency is 32.768 kHz, the RFRL register should be set to 00FFh.

26.2.21 Time Error Adjustment Register (RADJ)

Address(es): RTC.RADJ 4004 402Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	PMADJ[1:0]		ADJ[5:0]					
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler	R/W
b7, b6	PMADJ[1:0]	Plus-Minus	b7 b6 0 0: Adjustment is performed 0 1: Adjustment is performed by the addition to the prescaler 1 0: Adjustment is performed by the subtraction from the prescaler 1 1: Setting prohibited.	R/W

Adjustment is performed by the addition to or subtraction from the prescaler. If the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ. If the RCR2.AADJE bit is 1, adjustment is performed in the interval specified in the automatic adjustment period select (RCR2.AADJP) bit.

The current adjustment by software (disabling automatic adjustment) might be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting, then specify the next adjustment value.

RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits are updated before continuing with more processing. This register is cleared to 00h by an RTC software reset. The setting of this register is enabled only when the sub-clock oscillator is selected. When LOCO is selected, adjustment is not performed.

ADJ[5:0] bits (Adjustment Value)

The ADJ[5:0] bits specify the adjustment value (number of sub-clock cycles) from the prescaler.

PMADJ[1:0] bits (Plus-Minus)

The PMADJ[1:0] bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

26.2.22 Time Capture Control Register y (RTCCRy) (y = 0 to 2)

Address(es): RTC.RTCCR0 4004 4040h, RTC.RTCCR1 4004 4042h, RTC.RTCCR2 4004 4044h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected 0 1: Rising edge is detected 1 0: Falling edge is detected 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status	0: No event is detected 1: An event is detected.*1	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: Turn noise filter off 0 1: Setting prohibited 1 0: Turn noise filter on (count source) 1 1: Turn noise filter on (count source by divided by 32).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is disabled as the time capture event input 1: The RTCICn pin is enabled as the time capture event input (n = 0 to 2).	R/W

Note 1. Indicates that an event is detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRy register is used both in calendar count mode and in binary count mode. RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

RTCCRy is updated in synchronization with the count source. When RTCCRy is modified, check that all the bits except the TCST bit are updated before continuing with more processing. This register is cleared to 00h by an RTC software reset. When RTCICn is used as the time capture pin, VBTICTLR.VCHnIEN (n = 0 to 2) must be set to 1. For more information, see section 12, Battery Backup Function.

TCCT[1:0] bits (Time Capture Control)

The TCCT[1:0] bits control the edge detection of the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2. The detection edge is selectable. The TCCT[1:0] bits must be set while the VBTICTLR.VCHnIEN bit is 1.

TCST bit (Time Capture Status)

The TCST bit indicates that an event on the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2, was detected. When the TCST bit is 0, no event is detected. When the TCST bit is 1, this bit indicates that an event was detected on the associated pin and the capture register is valid. When multiple events are detected, the capture time for the first event is retained.

If an event is detected while the count operation is stopped (the RCR2.START bit is 0), the captured value is not guaranteed. In this case, set the TCST bit to 0 to delete the captured value. Writing 0 sets the TCST bit to 0. Writing any value other than 0 has no effect.

Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected). The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit is updated before continuing with additional processing.

TCNF[1:0] bits (Time Capture Noise Filter Control)

The TCNF[1:0] bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.

Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for 3 cycles of the specified sampling period, and then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the VBTICTLR.VCHnIEN bit is 1.

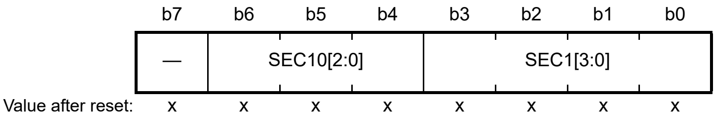
TCEN bit (Time Capture Event Input Pin Enable)

The TCEN bit enables or disables the time capture event input pins RTCIC0, RTCIC1, and RTCIC2. When the functions of the time capture event input pins are multiplexed, set VBTICTLR first. If the TCEN bit is set to 0, also set the TCCT[1:0] bits to 00b.

26.2.23 Second Capture Register y (RSECCPy) (y = 0 to 2)/BCNT0 Capture Register y (BCNT0CPy) (y = 0 to 2)

(1) In calendar count mode

Address(es): RTC.RSECCP0 4004 4052h, RTC.RSECCP1 4004 4062h, RTC.RSECCP2 4004 4072h



x: Undefined

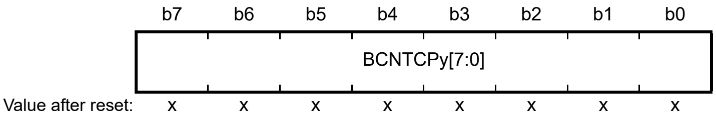
Bit	Symbol	Bit name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Capture	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	10-Second Capture	Capture value for the tens place of seconds	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset	R

RSECCPy is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively. This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode

Address(es): RTC.BCNT0CP0 4004 4052h, RTC.BCNT0CP1 4004 4062h, RTC.BCNT0CP2 4004 4072h



x: Undefined

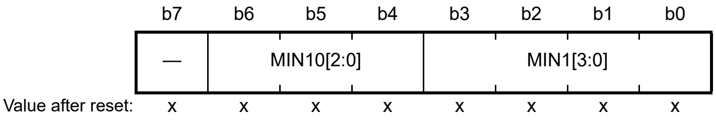
BCNT0CPy is a read-only register that captures the BCNT0 value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT0CP0, BCNT0CP1, and BCNT0CP2 registers, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

26.2.24 Minute Capture Register y (RMINCPy) (y = 0 to 2)/BCNT1 Capture Register y (BCNT1CPy) (y = 0 to 2)

(1) In calendar count mode

Address(es): RTC.RMINCP0 4004 4054h, RTC.RMINCP1 4004 4064h, RTC.RMINCP2 4004 4074h



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Capture	Capture value for the ones place of minutes	R

Bit	Symbol	Bit name	Description	R/W
b6 to b4	MIN10[2:0]	10-Minute Capture	Capture value for the tens place of minutes	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset	R

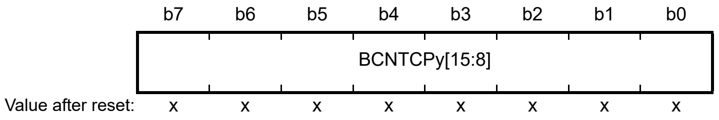
RMINCPy is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode

Address(es): RTC.BCNT1CP0 4004 4054h, RTC.BCNT1CP1 4004 4064h, RTC.BCNT1CP2 4004 4074h



x: Undefined

BCNT1CPy is a read-only register that captures the BCNT1 value when a time capture event is detected.

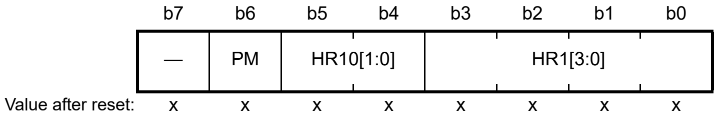
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT1CP0, BCNT1CP1, and BCNT1CP2 registers, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

26.2.25 Hour Capture Register y (RHRCPy) (y = 0 to 2)/BCNT2 Capture Register y (BCNT2CPy) (y = 0 to 2)

(1) In calendar count mode

Address(es): RTC.RHRCP0 4004 4056h, RTC.RHRCP1 4004 4066h, RTC.RHRCP2 4004 4076h



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Capture	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	10-Hour Capture	Capture value for the tens place of hours	R
b6	PM	PM	0: AM 1: PM.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RHRCPy is a read-only register that captures the RHRCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively. The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode

Address(es): RTC.BCNT2CP0 4004 4056h, RTC.BCNT2CP1 4004 4066h, RTC.BCNT2CP2 4004 4076h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPy[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT2CPy is a read-only register that captures the BCNT2 value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT2CP0, BCNT2CP1, and BCNT2CP2 registers, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

26.2.26 Date Capture Register y (RDAYCPy) (y = 0 to 2)/BCNT3 Capture Register y (BCNT3CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RDAYCP0 4004 405Ah, RTC.RDAYCP1 4004 406Ah, RTC.RDAYCP2 4004 407Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Capture	Capture value for the ones place of days	R
b5, b4	DATE10[1:0]	10-Day Capture	Capture value for the tens place of days	R
b7, b6	—	Reserved	These bits are read as 0 after an RTC software reset	R

RDAYCPy is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode

Address(es): RTC.BCNT3CP0 4004 405Ah, RTC.BCNT3CP1 4004 406Ah, RTC.BCNT3CP2 4004 407Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPy[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3CPy is a read-only register that captures the BCNT3 value when a time capture event is detected.

The event detection times detected by the RTCTC0, RTCTC1, and RTCTC2 pins are stored in the BCNT3CP0, BCNT3CP1, and BCNT3CP2 registers, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

26.2.27 Month Capture Register y (RMONCPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RMONCP0 4004 405Ch, RTC.RMONCP1 4004 406Ch, RTC.RMONCP2 4004 407Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Capture	Capture value for the ones place of months	R
b4	MON10	10-Month Capture	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0	R

RMONCPy is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is cleared to 00h by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

26.3 Operation

26.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, perform the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register.

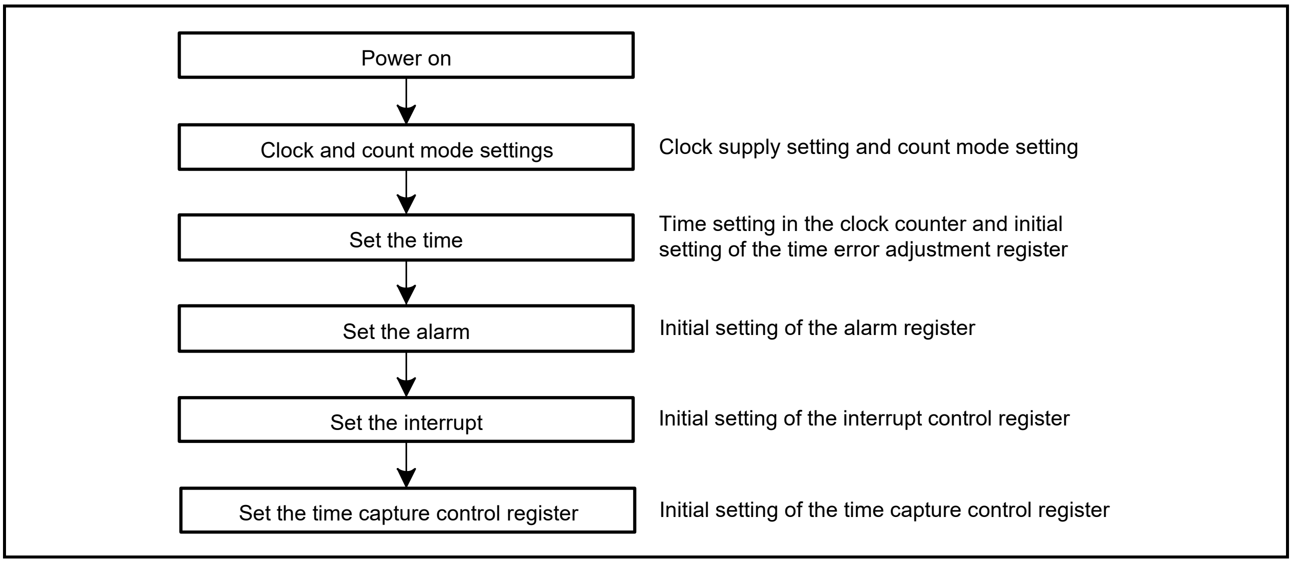
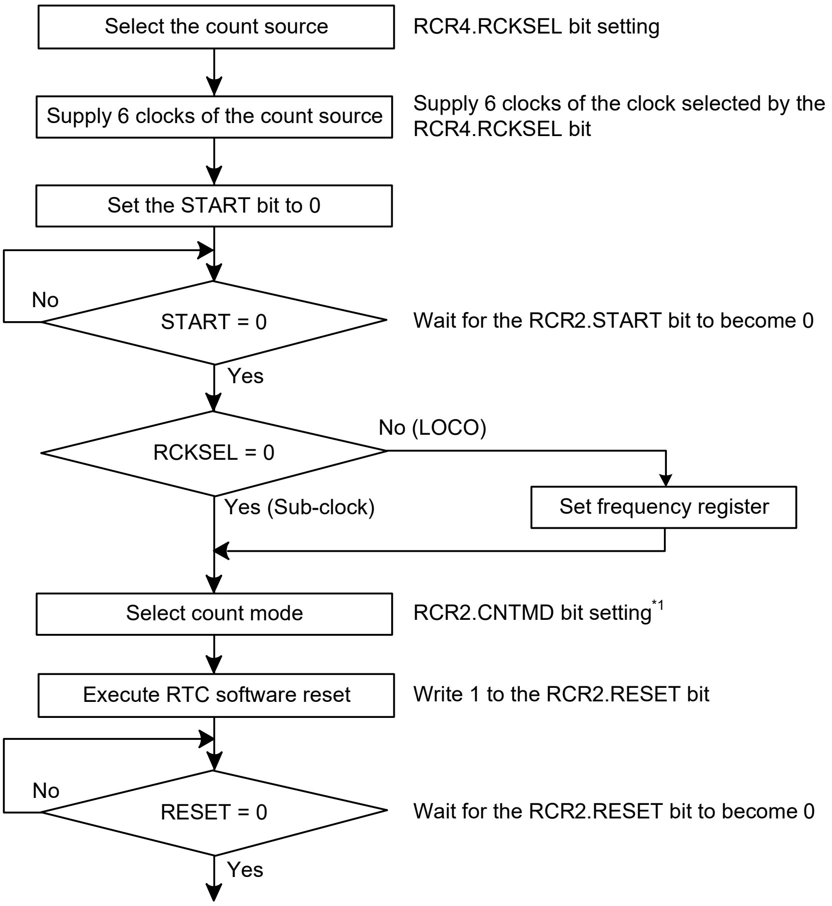


Figure 26.2 Outline of initial settings after a power on

26.3.2 Clock and Count Mode Setting Procedure

Figure 26.3 shows how to set the clock and the count mode.

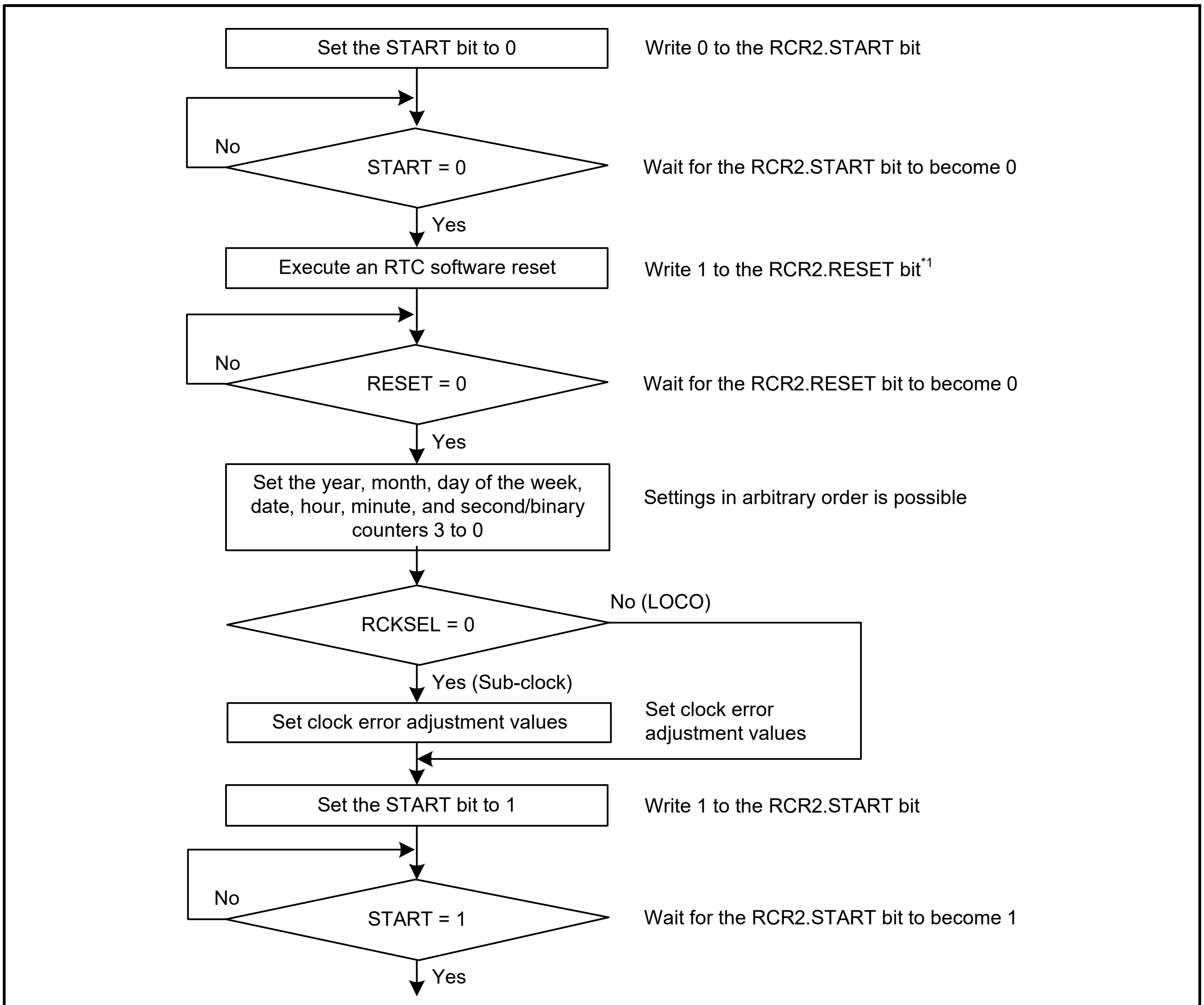


Note 1. This step is not required if the count mode is set concurrently by setting the START bit to 0. A value associated with the count mode setting must be written to the RCR2.CNTMD bit.

Figure 26.3 Clock and count mode setting procedure

26.3.3 Setting the Time

Figure 26.4 shows how to set the time.



Note 1. This step is not required for the time-setting procedure because an RTC software reset is executed in the clock setting procedure of the initial settings for the power supply.

Figure 26.4 Setting the time

26.3.4 30-Second Adjustment

Figure 26.5 shows how to execute a 30-second adjustment.

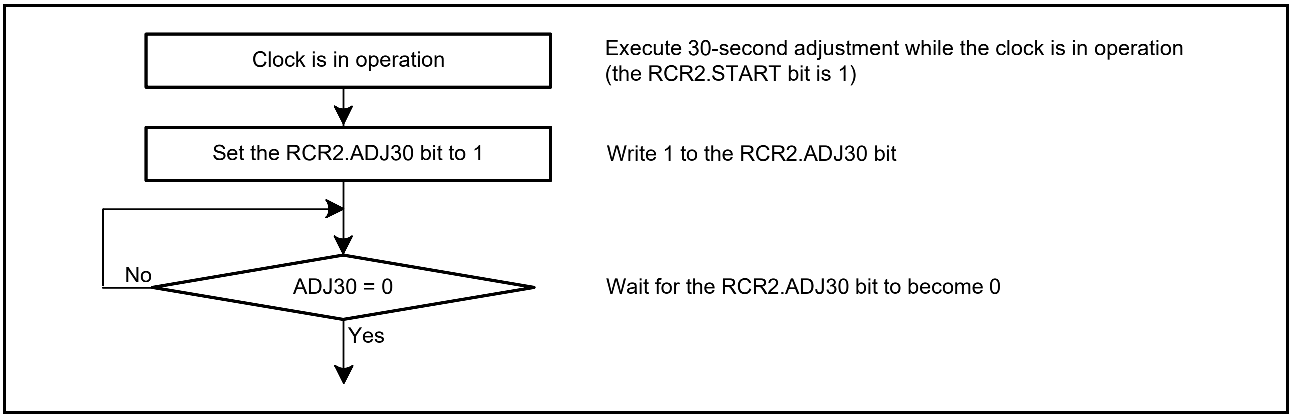
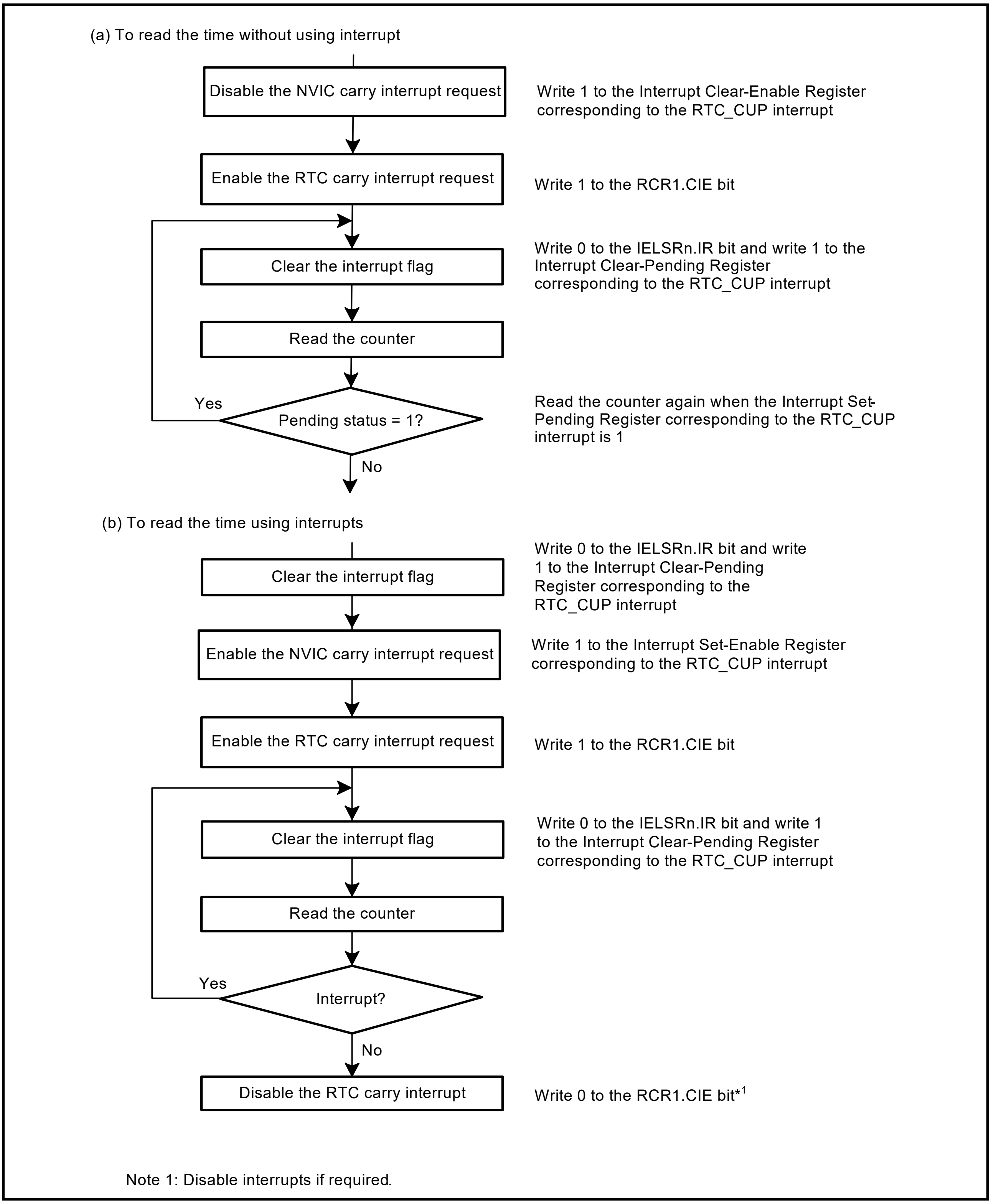


Figure 26.5 30-second adjustment

26.3.5 Reading 64-Hz Counter and Time

Figure 26.6 shows how to read a 64-Hz counter and time.



Note 1: Disable interrupts if required.

Figure 26.6 Reading time

If a carry occurs while the 64-Hz counter and time are being read, the correct time is not obtained, therefore they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 26.6, and the procedure using carry interrupts is shown in (b). To keep the program simple, Renesas recommends using method (a) in most cases.

26.3.6 Alarm Function

Figure 26.7 shows how to use the alarm function.

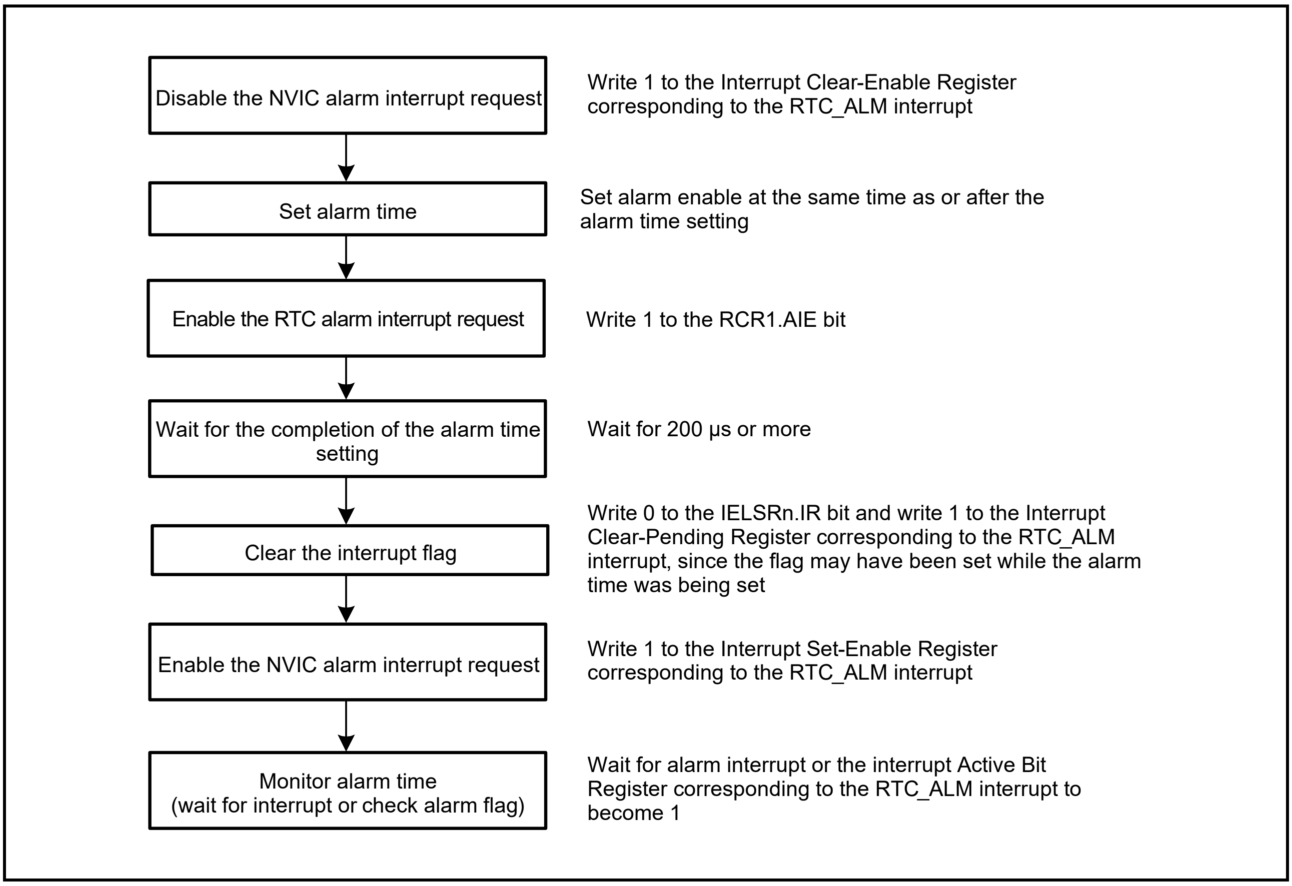


Figure 26.7 Using the alarm function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register associated with the target bit of the alarm, and set the alarm time in the alarm register. For bits that are not the target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IELSRn.IR bit and Interrupt Set-Pending/Clear-Pending Register associated with the RTC_ALM interrupt are set to 1. Alarm detection can be confirmed by reading the interrupt Set-Pending Register associated with the RTC_ALM interrupt, but an interrupt should be used in most cases. If 1 is set in the Interrupt Set-Enable Register associated with the RTC_ALM interrupt, an alarm interrupt is generated in the event of the alarm, enabling the alarm to be detected.

Writing 0 sets the IELSRn.IR bit associated with the RTC_ALM interrupt to 0. If interrupt is enabled, the Interrupt Set-Pending/Clear-Pending Register associated with the RTC_ALM interrupt is cleared automatically after exiting the interrupt handler. Otherwise, write 1 to the Interrupt Clear-Pending Register associated with the RTC_ALM interrupt to clear it.

When the counter and the alarm time match in a low power state, the MCU returns from the low power state. In Deep Software Standby mode, the MCU returns from the Deep Software Standby mode even when the alarm interrupt request is disabled.

26.3.7 Procedure for Disabling Alarm Interrupt

Figure 26.8 shows the procedure for disabling the enabled alarm interrupt request.

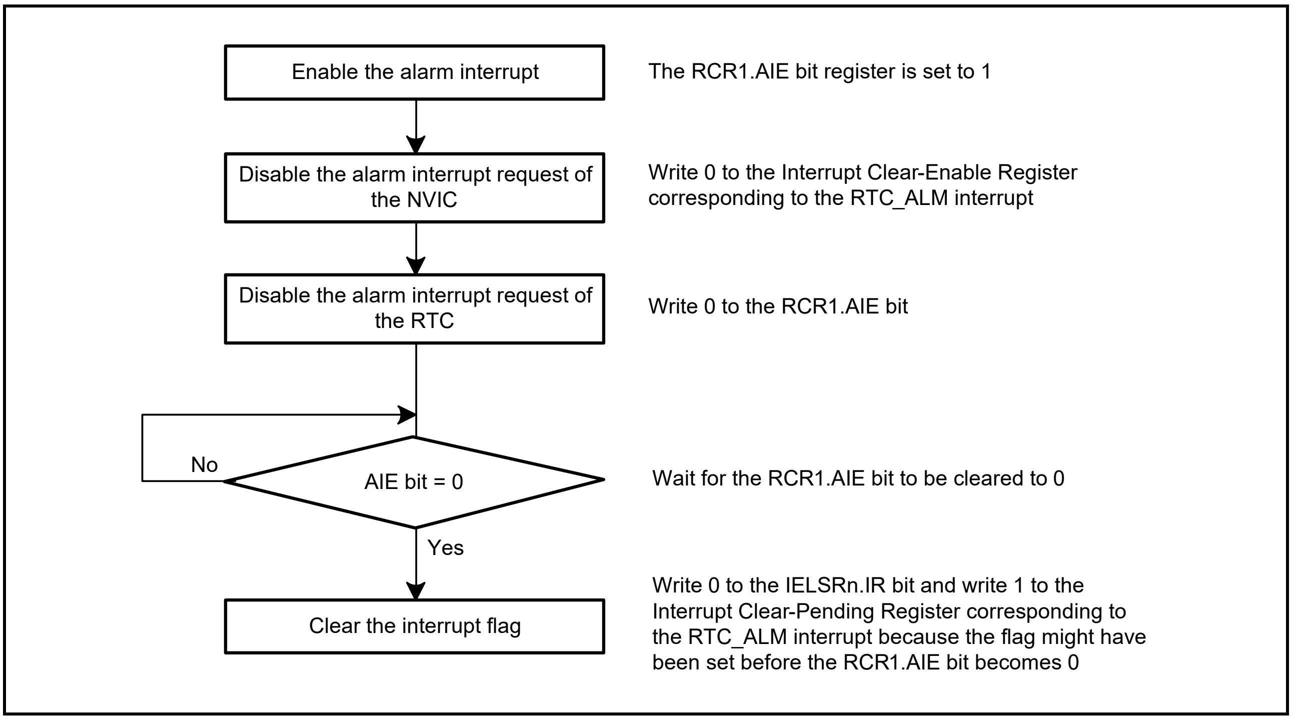


Figure 26.8 Procedure for disabling alarm interrupt request

26.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors, running fast or slow, in the time caused by variation in the precision of oscillation by the sub-clock oscillator. Because 32,768 cycles of the sub-clock oscillator constitute 1 second of operation when the sub-clock oscillator is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low.

The time error adjustment functions include:

- Automatic adjustment
- Adjustment by software.

Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

26.3.8.1 Automatic adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1. Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJP bit elapses.

(1) Example 1: Sub-clock oscillator running at 32.769 kHz

(a) Adjustment procedure

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by 1 clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 60 (3Ch).

(2) Example 2: Sub-clock oscillator running at 32.766 kHz

(a) Adjustment procedure

When the sub-clock oscillator is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by 2 clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 20 (14h).

(3) Example 3: Sub-clock oscillator running at 32.764 kHz

(a) Adjustment procedure

At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Because the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for 4 clock cycles per second. In 8 seconds, the delay is 32 clock cycles, therefore correction can be made by advancing the clock 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 32 (20h).

26.3.8.2 Adjustment by software

Enable adjustment by software by setting the RCR2.AADJE bit to 0. Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register on execution of a write instruction to the RADJ register.

(1) Example 1: Sub-clock oscillator running at 32.769 kHz

(a) Adjustment procedure

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by 1 cycle every second.

(b) Register settings

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 1 (01h).
  This is written to the RADJ register once per 1-second interrupt.

26.3.8.3 Procedure for changing the mode of adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

To change adjustment by software to automatic adjustment:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
3. Use the RCR2.AADJP bit to select the period of adjustment.
4. In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

To change automatic adjustment to adjustment by software:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
3. Proceed with the adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the wanted time. After that, the time is adjusted every time a value is written to the RADJ register.

26.3.8.4 Procedure for stopping adjustment

Stop the adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

26.3.8.5 Capturing the time

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Set VBTICTLR.VCHnIEN (n = 0 to 2) to 1 to enable the RTCICn input. Operation when the noise filter is off is shown in Figure 26.9 and operation when the noise filter is on is shown in Figure 26.10.

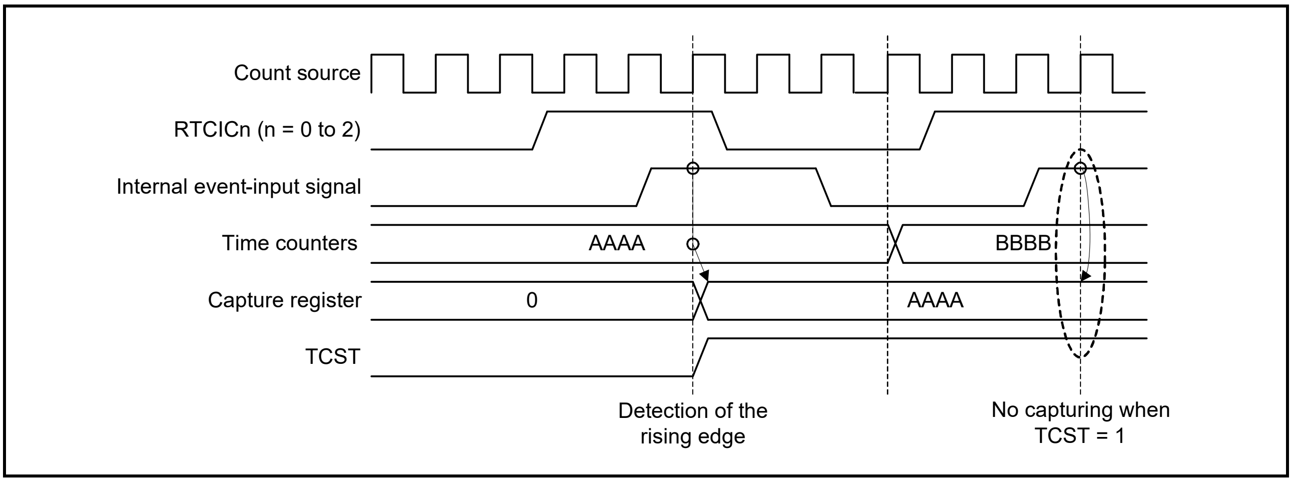


Figure 26.9 Timing of a time capture operation with the filter off

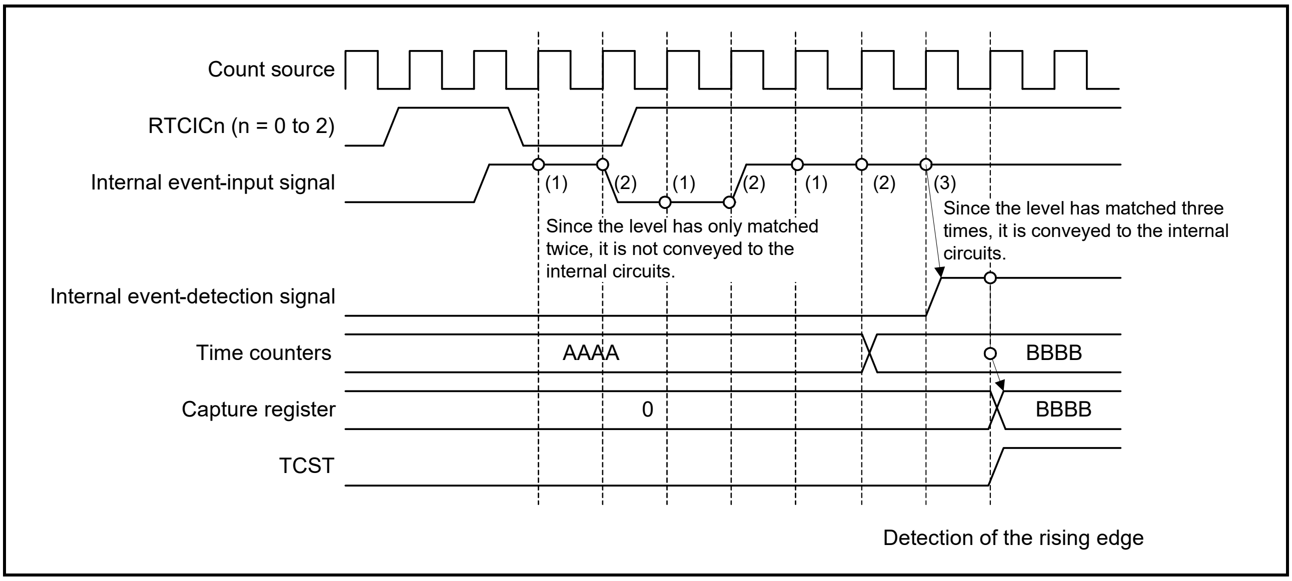


Figure 26.10 Timing of a time capture operation with the filter on

26.4 Interrupt Sources

The RTC has three interrupt sources and are listed in Table 26.3.

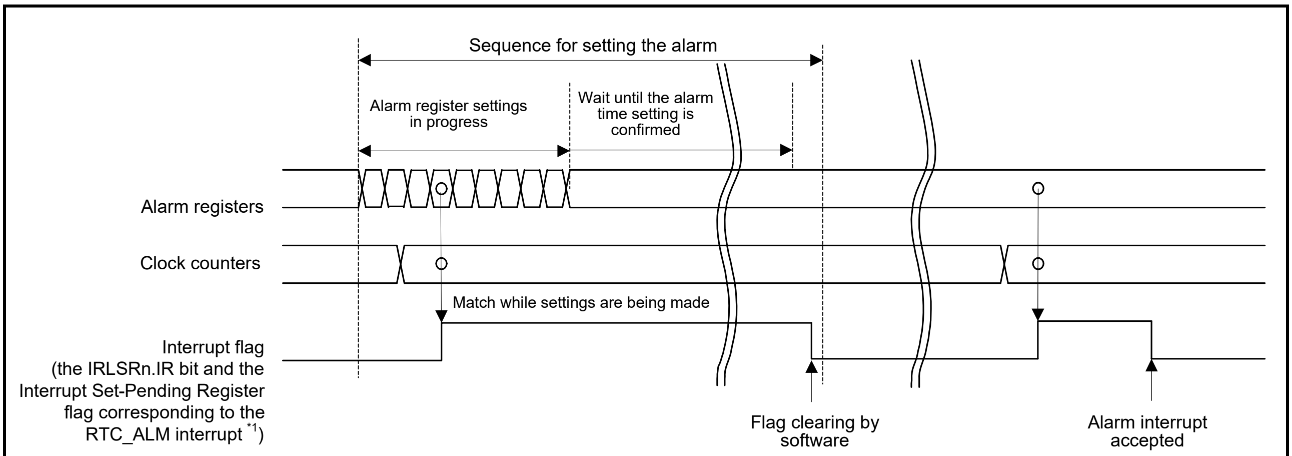
Table 26.3 RTC interrupt sources

Name	Interrupt source
RTC_ALM	Alarm interrupt
RTC_PRD	Periodic interrupt
RTC_CUP	Carry interrupt

(1) Alarm interrupt (RTC_ALM)

This interrupt is generated based on the result of comparison between the alarm registers and RTC counters. For details, see section 26.3.6, Alarm Function.

Because there is a possibility that the interrupt flag might be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and clear the IELSRn.IR bit and the interrupt Set-Pending Register associated with the RTC_ALM interrupt to 0 again after modifying values of the alarm registers. After the interrupt flag for the alarm interrupt is set to 1 and the state is returned to non-matching of the alarm registers and clock counters, the flag is not set again until there is another match or the values of the alarm registers are modified again.



Note 1. See section 14, Interrupt Controller Unit (ICU) for details on the associated interrupt vector number.

Figure 26.11 Timing diagram for the alarm interrupt (RTC_ALM)

(2) Periodic interrupt (RTC_PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (RTC_CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

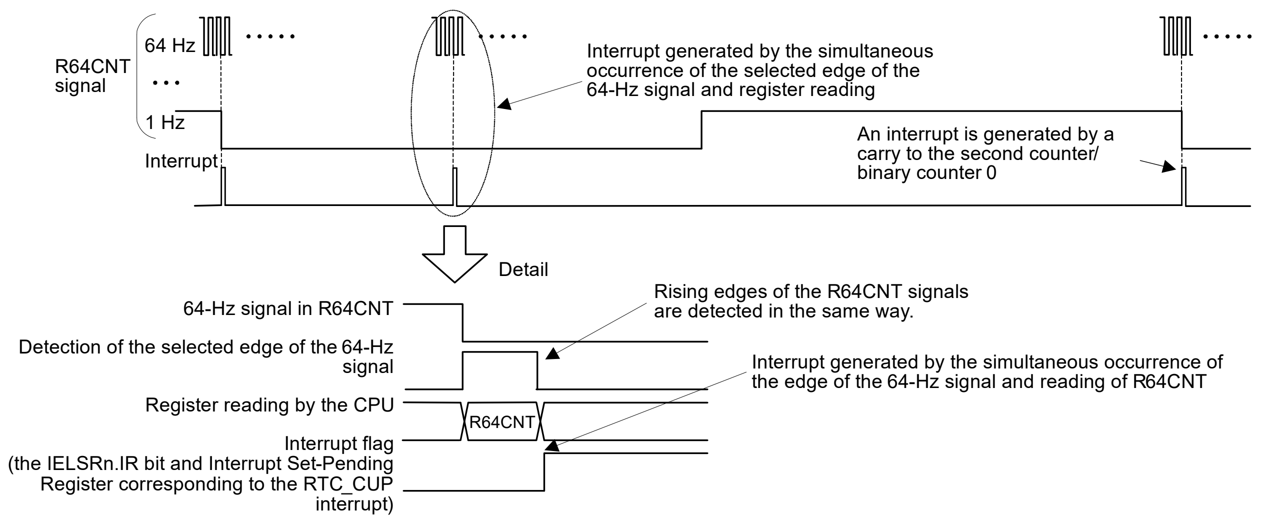


Figure 26.12 Timing diagram for the carry interrupt (RTC_CUP)

26.5 Event Link Output

The RTC generates periodic event output (RTC_PRD) event signal for the Event Link Controller (ELC) that can be used to initiate operations by other modules selected in advance.

The periodic event signal is output at the interval selected from 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by setting the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is used, only set the ELC after setting the RTC, for example initialization and time settings. Setting the RTC after the ELC can lead to output of unexpected event signals.

26.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the associated enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal through the ELC when an interrupt source is generated, regardless of the setting of the associated interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during Software Standby mode or Deep Software Standby mode, the periodic event signals for the ELC are not output.

26.6 Usage Notes

26.6.1 Register Writing during Counting

The following registers must not be written to during counting, that is, while the RCR2.START bit is 1:

- RSECCNT/BCNT0
- RMINCNT/BCNT1
- RHRCNT/BCNT2
- RDAYCNT
- RWKCNT/BCNT3
- RMONCNT
- RYRCNT
- RCR1.RTCOS
- RCR2.RTCOE
- RCR2.HR24
- RFRL.

The counter must be stopped before writing to any of these registers.

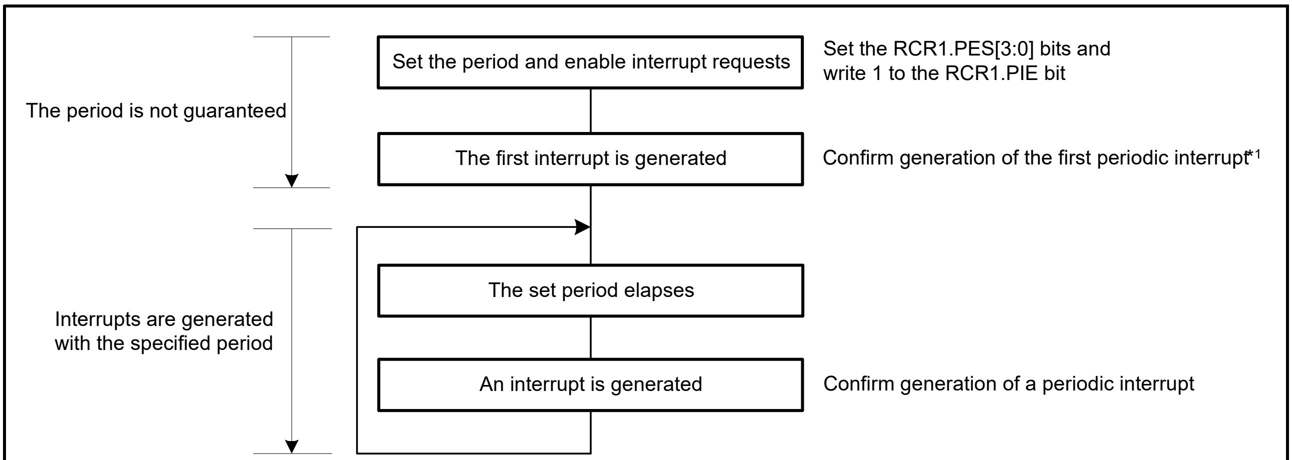
26.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 26.13.

The generation and period of the periodic interrupt can be changed by setting the RCR1.PES[3:0] bits. However, because the prescaler R64CNT and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting the RCR1.PES[3:0] bits. In addition, any of the following operation can affect the interrupt period:

- Stopping/restarting or resetting counter operation
- Reset by RTC software
- 30-second adjustment by changing the RCR2 value.

When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted based on the adjustment value.



Note 1. When an interrupt generation period changes while the periodic interrupt is used, an interrupt might be generated at the completion of the setting. If the interrupt is generated immediately after the setting, the period is not guaranteed for two interrupts including the current interrupt

Figure 26.13 Using the periodic interrupt function

26.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted based on the adjustment value.

26.6.4 Transitions to Low Power Modes after Setting Registers

A transition to a low power state (Software Standby mode, Deep Software Standby mode, or battery backup) during a write to an RTC register might corrupt the value in the register. After setting the register, confirm that the setting is in place before initiating a transition to a low power state.

26.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter after writing to the counter register, follow the procedure in section 26.3.5, Reading 64-Hz Counter and Time
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR4 register, or frequency register is reflected when four read operations are performed after writing
- The values written to the RCR1.CIE, RCR1.RTCOS, and RCR2.RTCOE bits can be read immediately after writing
- To read the value from the timer counter after return from a reset, Software Standby mode, Deep Software Standby mode, or the battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit is 1)
- After a reset is generated, write to the RTC register after 6 cycles of the count source clock elapse.

26.6.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop the counting operation, then restart it from the initial setting. For details on the initial setting, see section 26.3.1, Outline of Initial Settings of Registers after Power On.

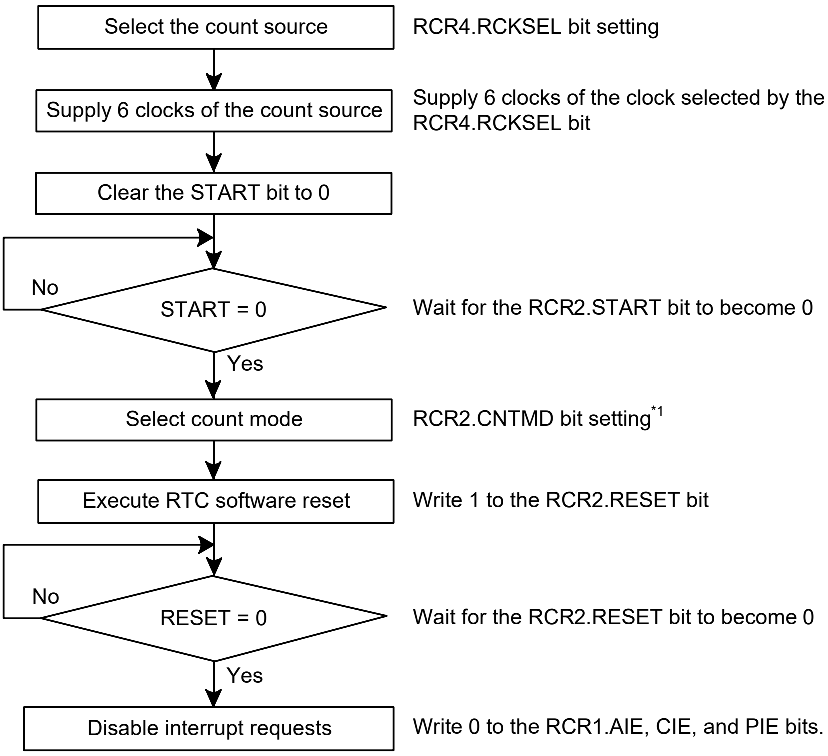
26.6.7 Initialization Procedure when the RTC Is Not To Be Used

Registers in the RTC are not initialized by a reset. Depending on the initial state, the generation of an unintentional interrupt request or operation of the counter might lead to increased power consumption.

For applications that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 26.14.

Alternatively, when the sub-clock oscillator is not used as the system clock or realtime clock, the counter can be stopped by writing 0 (subclock oscillator is selected) to the RCR4.RCKSEL bit and stopping the sub-clock oscillator. To stop the sub-clock oscillator, write 1 to the SOSCCR.SOSTP bit.

For details on the setting of the SOSCCR.SOSTP bit, see section 9, Clock Generation Circuit.



Note 1. This step is not necessary if the count mode is set concurrently with setting the START bit to 0.

Figure 26.14 Initialization procedure

26.6.8 When Switching Source Clock

When switching a clock source by changing SCKCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. If the RTC periodical interrupt or RTC periodical event output was generated at this time, the interrupt or event is invalid.

27. Watchdog Timer (WDT)

27.1 Overview

The Watchdog Timer (WDT) is a 14-bit down-counter and can be used to reset the MCU when the counter underflows because the system has run out of control and become unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt. The refresh-permitted period can be set to refresh the counter and to detect when the system runs out of control.

Table 27.1 lists the WDT specifications and Figure 27.1 shows a block diagram.

Table 27.1 WDT specifications

Parameter	Specifications
Count source	Peripheral clock (PCLKB)
Clock division ratio	Division by 4, 64, 128, 512, 2,048, or 8,192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto start mode: Counting automatically starts after a reset, or after an underflow or refresh error occurs • Register start mode: Counting is started with a refresh by writing to the WDTRR register.
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated.
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
WDT reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Reading of the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	• Down-counter underflow event output • Refresh error event output.
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output.

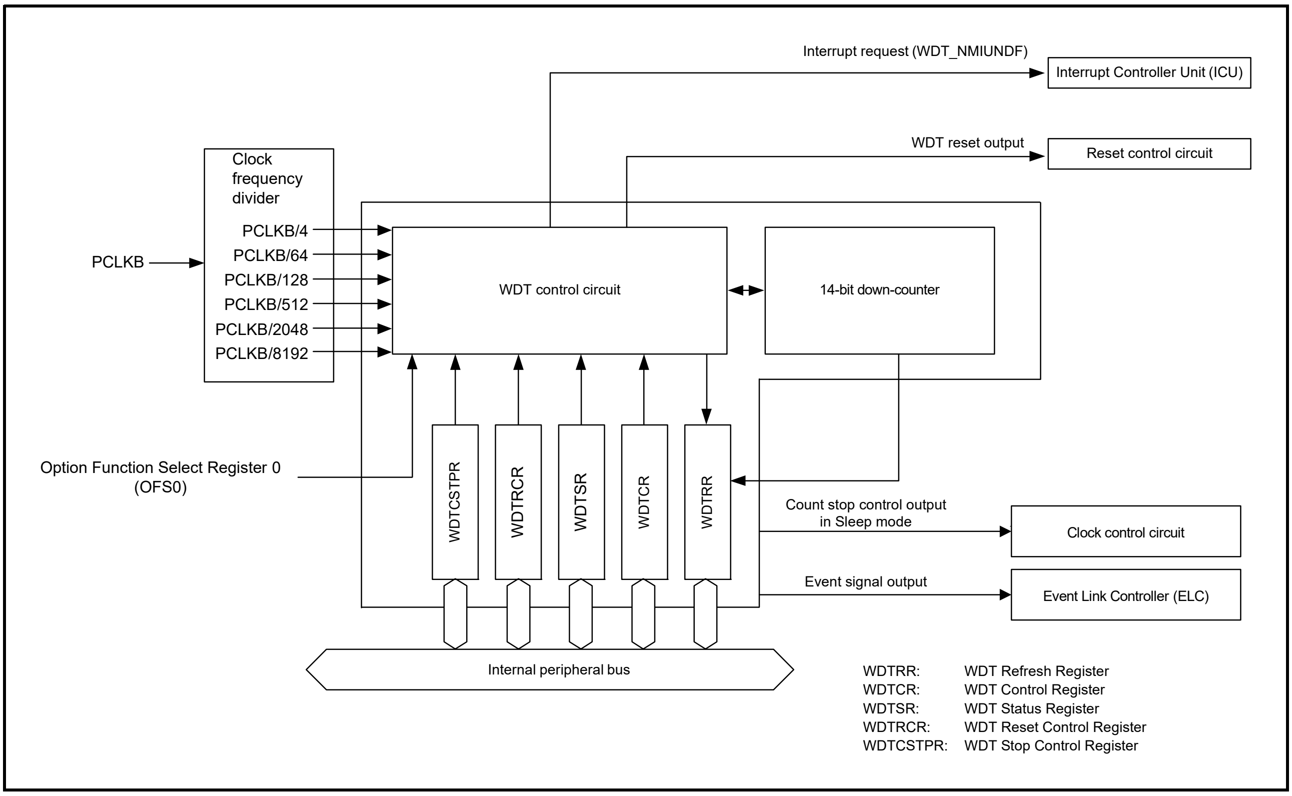


Figure 27.1 WDT block diagram

27.2 Register Descriptions

27.2.1 WDT Refresh Register (WDTRR)

Address(es): WDT.WDTRR 4004 4200h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

The WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 00h and then writing FFh to WDTRR (refresh operation) within the refresh-permitted period.

After the down-counter is refreshed, it starts counting down from the value selected in the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]) in Option Function Select Register 0 in auto start mode. In register start mode, counting down starts from the value selected in the Timeout Period Select bits (WDTCR.TOPS[1:0]) in the WDT Control Register.

When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh. For details on the refresh operation, see section 27.3.3, Refresh Operation.

27.2.2 WDT Control Register (WDTCR)

Address(es): WDT.WDTCR 4004 4202h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 1,024 cycles (03FFh) 0 1: 4,096 cycles (0FFFh) 1 0: 8,192 cycles (1FFFh) 1 1: 16,384 cycles (3FFFh).	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	CKS[3:0]	Clock Division Ratio Select	b7 b4 0 0 0 1: PCLKB/4 0 1 0 0: PCLKB/64 1 1 1 1: PCLKB/128 0 1 1 0: PCLKB/512 0 1 1 1: PCLKB/2048 1 0 0 0: PCLKB/8192. Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (do not specify window end position).	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (do not specify window start position).	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Some constraints apply to writes to the WDTCR register. For details, see section 27.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the settings in the WDTCR register are disabled, and the settings in Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made for the OFS0 register. For details, see section 27.3.7, Associations between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1,024, 4,096, 8,192, and 16,384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of PCLKB cycles until the counter underflows.

Table 27.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 27.2 Timeout period settings

CKS[3:0] bits				TOPS[1:0] bits				
b7	b6	b5	b4	b1	b0	Clock division ratio	Timeout period (number of cycles)	PCLKB clock cycles
0	0	0	1	0	0	PCLKB/4	1,024	4,096
				0	1		4,096	16,384
				1	0		8,192	32,768
				1	1		16,384	65,536
0	1	0	0	0	0	PCLKB/64	1,024	65,536
				0	1		4,096	262,144
				1	0		8,192	524,288
				1	1		16,384	1,048,576
1	1	1	1	0	0	PCLKB/128	1,024	131,072
				0	1		4,096	524,288
				1	0		8,192	1,048,576
				1	1		16,384	2,097,152
0	1	1	0	0	0	PCLKB/512	1,024	524,288
				0	1		4,096	2,097,152
				1	0		8,192	4,194,304
				1	1		16,384	8,388,608
0	1	1	1	0	0	PCLKB/2048	1,024	2,097,152
				0	1		4,096	8,388,608
				1	0		8,192	16,777,216
				1	1		16,384	33,554,432
1	0	0	0	0	0	PCLKB/8192	1,024	8,388,608
				0	1		4,096	33,554,432
				1	0		8,192	67,108,864
				1	1		16,384	134,217,728

CKS[3:0] bits (Clock Division Ratio Select)

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the peripheral clock (PCLKB) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, this allows the WDT to be configured to a count period between 4,096 and 134,217,728 cycles of the PCLKB clock.

RPES[1:0] bits (Window End Position Select)

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the value for the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

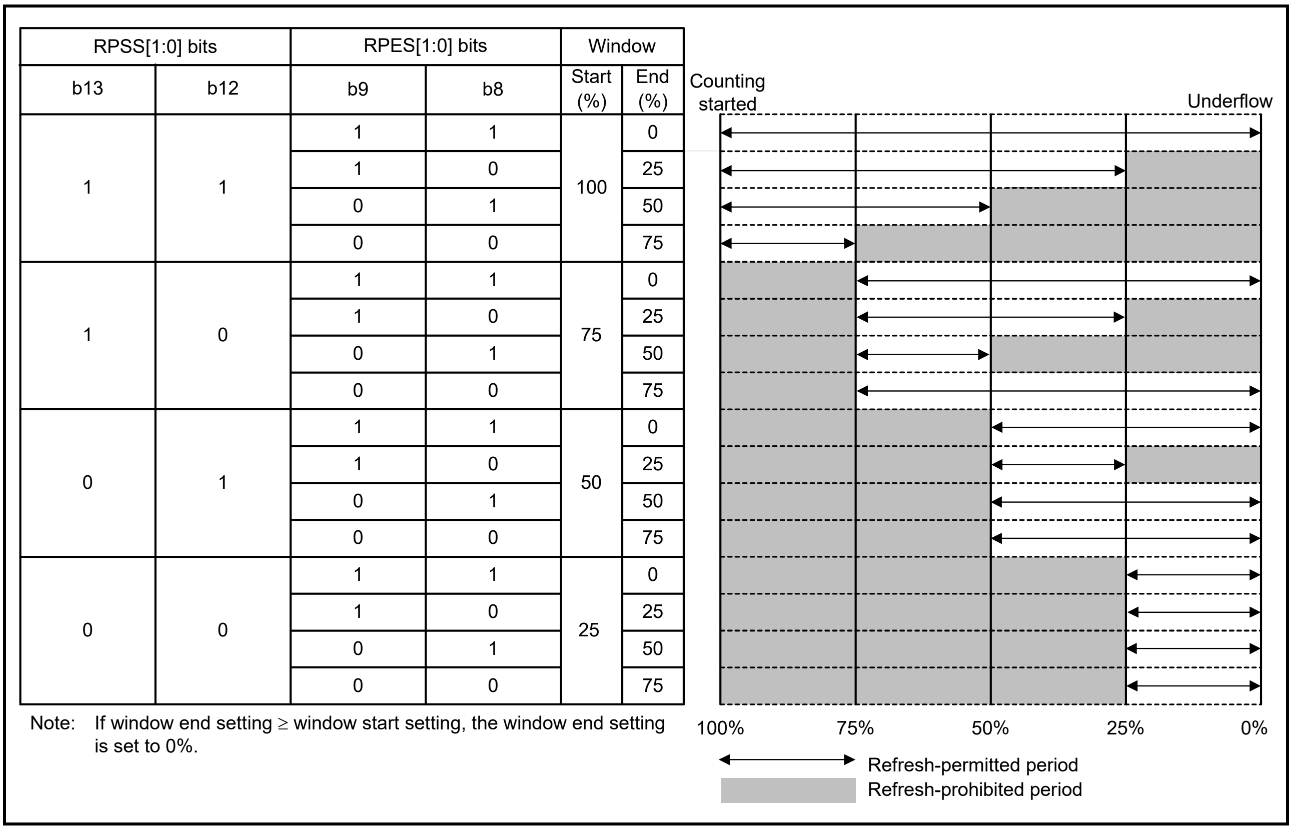
RPSS[1:0] bits (Window Start Position Select)

The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window end position is set to 0%.

Table 27.3 lists the counter values for the window start and end positions, and Figure 27.2 shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 27.3 Relationship between the timeout period and window start and end counter values

TOPS[1:0] bits		Timeout period		Window start and end counter value			
		Cycles	Counter value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh



RPSS[1:0] bits		RPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

Figure 27.2 RPSS[1:0] and RPES[1:0] bit settings and refresh-permitted period

27.2.3 WDT Status Register (WDTSR)

Address(es): WDT.WDTSR 4004 4204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred.	R(/W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred.	R(/W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] bits (Down-Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the down-counter. A value of 1 indicates that the down-counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N+1) PCLKB cycles. In addition, clearing of the flag is ignored for (N+1) PCLKB cycles after an underflow. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0001b, N = 4
- When WDTCR.CKS[3:0] = 0100b, N = 64
- When WDTCR.CKS[3:0] = 1111b, N = 128
- When WDTCR.CKS[3:0] = 0110b, N = 512
- When WDTCR.CKS[3:0] = 0111b, N = 2048
- When WDTCR.CKS[3:0] = 1000b, N = 8192.

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N+1) PCLKB cycles. In addition, clearing of the flag is ignored for (N+1) PCLKB cycles after a refresh error. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0001b, N = 4
- When WDTCR.CKS[3:0] = 0100b, N = 64
- When WDTCR.CKS[3:0] = 1111b, N = 128
- When WDTCR.CKS[3:0] = 0110b, N = 512
- When WDTCR.CKS[3:0] = 0111b, N = 2048
- When WDTCR.CKS[3:0] = 1000b, N = 8192

27.2.4 WDT Reset Control Register (WDTRCR)

Address(es): WDT.WDTRCR 4004 4206h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request or interrupt request output is enabled 1: Reset output is enabled.	R/W

Some constraints apply to writes to the WDTRCR register. For details, see section 27.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTRCR register settings are disabled, and the settings in Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made for the OFS0 register. For details, see section

27.3.7, Associations between Option Function Select Register 0 (OFS0) and WDT Registers.

27.2.5 WDT Count Stop Control Register (WDTCSTPR)

Address(es): WDT.WDTCSTPR 4004 4208h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SLCSTP	Sleep-Mode Count Stop Control	0: Count stop is disabled 1: Count is stopped when transition to Sleep mode.	R/W

The WDTCSTPR register controls whether to stop the WDT counter in a low power mode. Some constraints apply to writes to the WDTCSTPR register. For details, see section 27.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTCSTPR register settings are disabled, and the settings in Option Function Select register 0 (OFS0) are enabled. The settings for the WDTCSTPR register can also be made for the OFS0 register. For details, see section 27.3.7, Associations between Option Function Select Register 0 (OFS0) and WDT Registers.

SLCSTP bit (Sleep-Mode Count Stop Control)

The SLCSTP bit selects whether to stop counting when transition to Sleep mode.

27.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see section 27.3.7, Associations between Option Function Select Register 0 (OFS0) and WDT Registers.

27.3 Operation

27.3.1 Count Operation in Each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting is started with a refresh by writing to the register.

In auto start mode, counting automatically starts after release from the reset state in accordance with the settings in Option Function Select Register 0 (OFS0) in the flash.

In register start mode, counting starts with a refresh by writing to the register after the respective registers are set after a release from the reset state.

Select auto start mode or register start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register. When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled, and the settings in the OFS0 register are enabled. When the register start mode is selected, the OFS0 register settings are disabled, and the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

27.3.1.1 Register start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) is 1, register start mode is selected and the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

After the reset state is released, set the following to Sleep mode in the WDTCSTPR register:

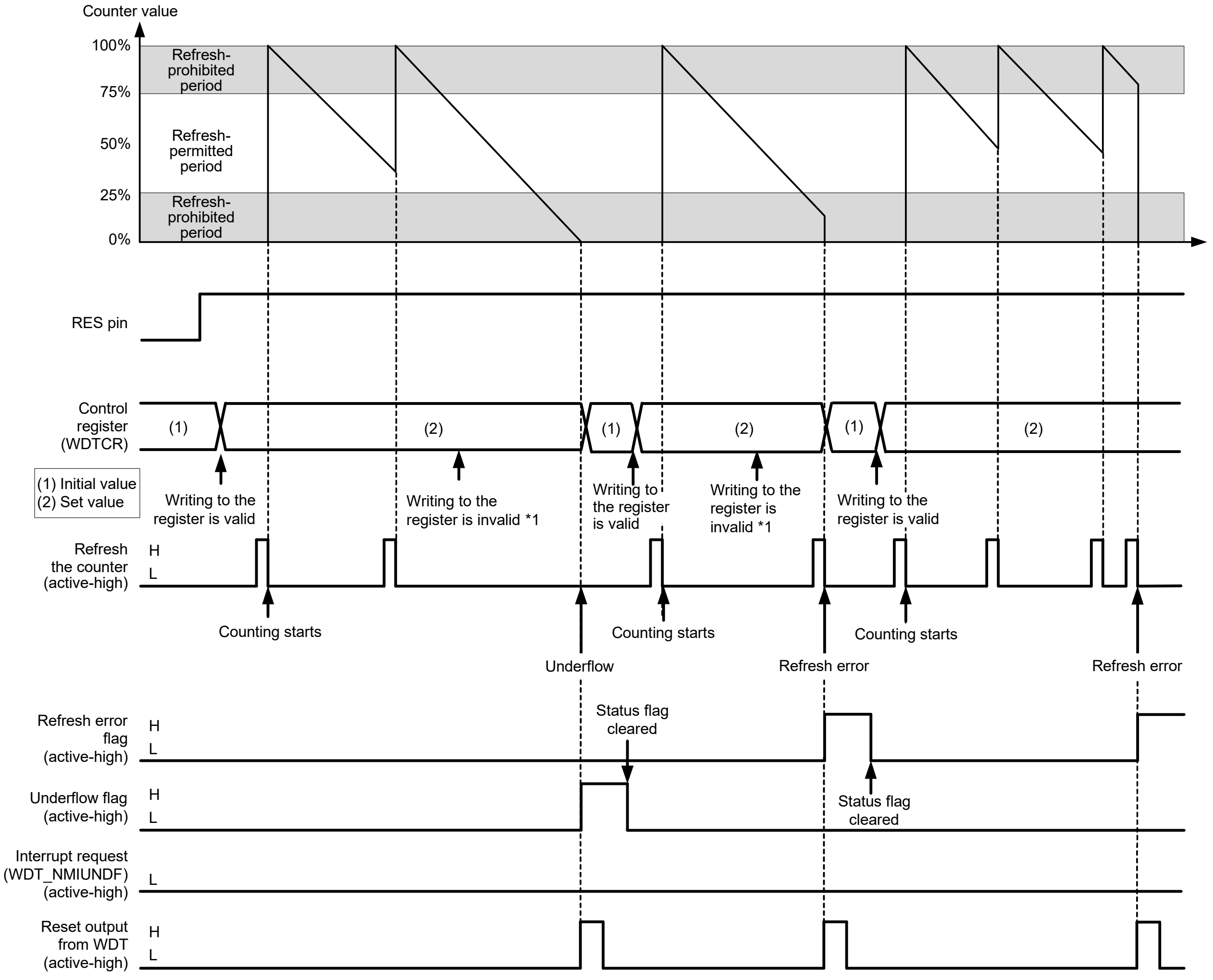
- Clock division ratio
- Window start and end positions
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to Sleep mode in the WDTCSTPR register.

Refresh the down-counter to start counting down from the value set in the Timeout Period Select bits (WDTCR.TOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The WDT does not output the reset signal as long as counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs a reset signal or a non-maskable interrupt request/interrupt request (WDT_NMIUNDF). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 27.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b).



Note 1. See section 27.3.2, Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

Figure 27.3 Operation example in register start mode

27.3.1.2 Auto start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) in the Option Function Select Register 0 (OFS0) is 0, auto start mode is selected. The WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled while the settings in the OFS0 register are enabled.

Within the reset state, the following values in Option Function Select register 0 (OFS0) are set in the WDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control on transition to Sleep mode.

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The WDT does not output the reset signal as long as the counting continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to a runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT asserts the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 27.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b).

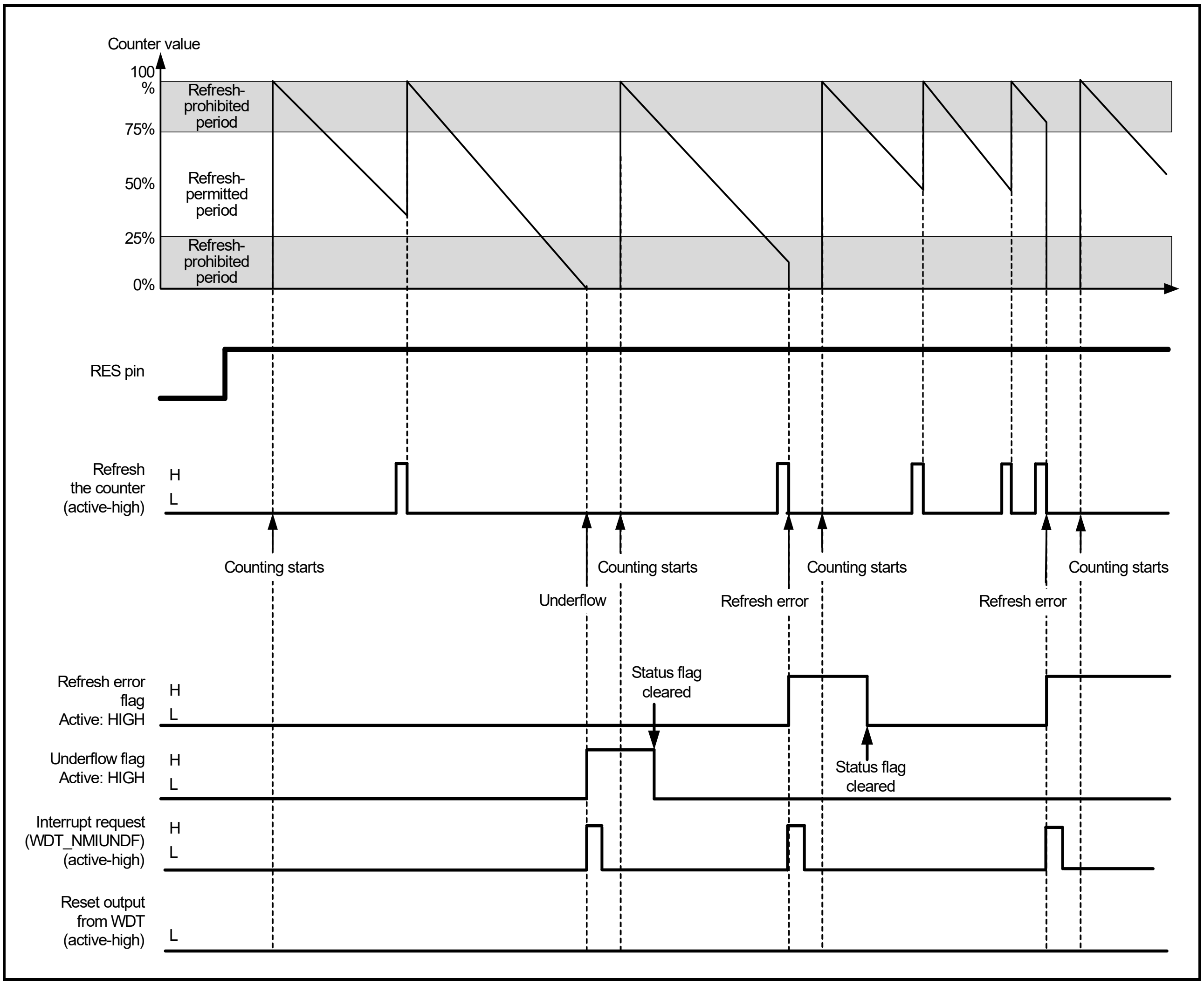


Figure 27.4 Operation example in auto start mode

27.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSTPR) is possible once between the release from the reset state and the first refresh operation.

After a refresh (counting starts) or a write to WDTCR, WDTRCR or WDTCSTPR, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSTPR against subsequent write attempts. This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 27.5 shows control waveforms produced in response to writing to the WDTCR.

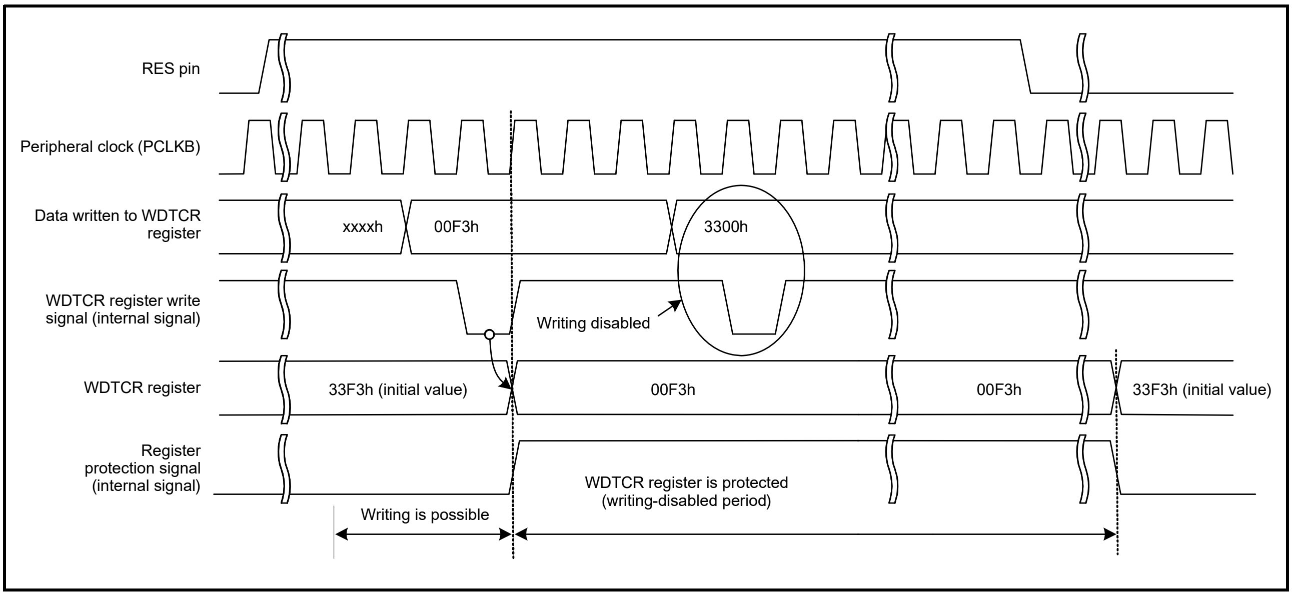


Figure 27.5 Control waveforms produced in response to writes to the WDTCR register

27.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and FFh to the WDT Refresh Register (WDTRR). If a value other than FFh is written after 00h, the down-counter is not refreshed. After an invalid value is written, correct refreshing resumes by writing 00h and FFh to the WDTRR register.

When a register other than WDTRR is accessed or WDTRR is read between writing 00h and writing FFh to WDTRR, correct refreshing is performed.

Writing to refresh the counter must be performed within the refresh-permitted period and whether this is done is determined by writing FFh. For this reason, correct refreshing is performed even when 00h is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

- 00h → FFh
- 00h (n-1th time) → 00h (nth time) → FFh
- 00h → access to another register or read from WDTRR → FFh.

[Example write sequences that are invalid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh.

After FFh is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing FFh to WDTRR 4 count cycles before the down-counter underflows.

Figure 27.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLKB/64.

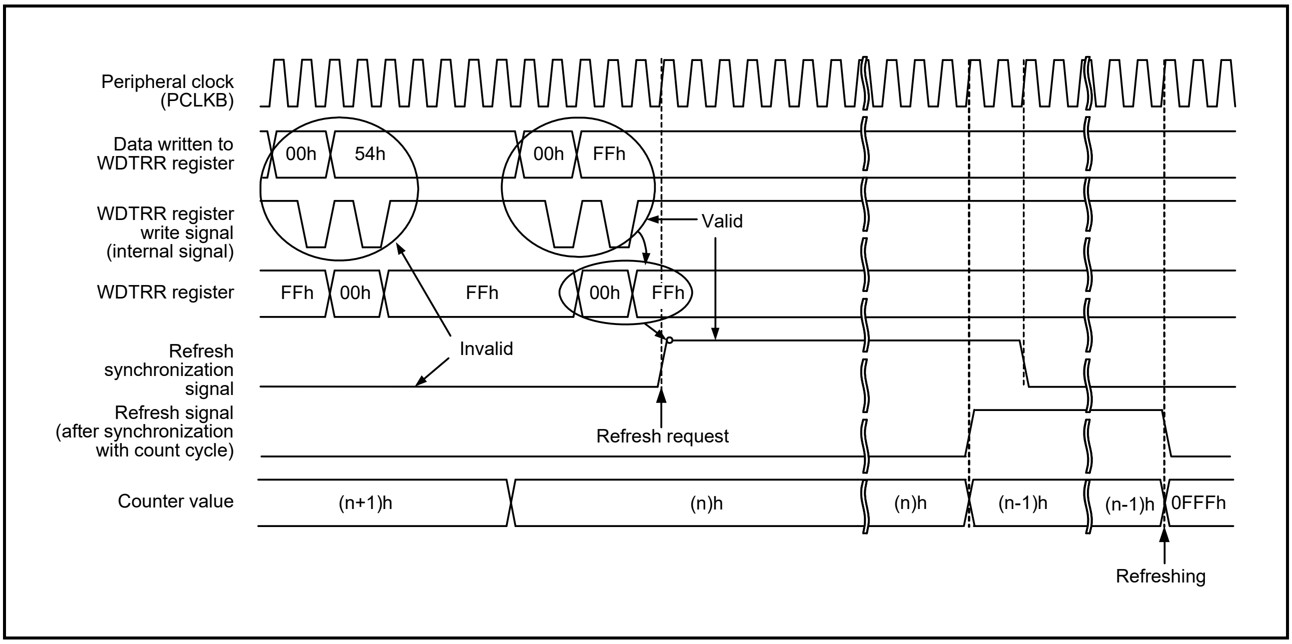


Figure 27.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0100b and WDTCR.TOPS[1:0] = 01b

27.3.4 Reset Output

When the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

27.3.5 Interrupt Sources

When the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in Option Function Select Register 0 (OFS0) is set to 0 in auto start mode, an interrupt signal (WDT_NMIUNDF) is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 14, Interrupt Controller Unit (ICU).

Table 27.4 WDT interrupt sources

Name	Interrupt source	DTC activation	DMAC activation
WDT_NMIUNDF	• Down-counter underflow • Refresh error	Not possible	Not possible

27.3.6 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value.

Figure 27.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLKB/64.

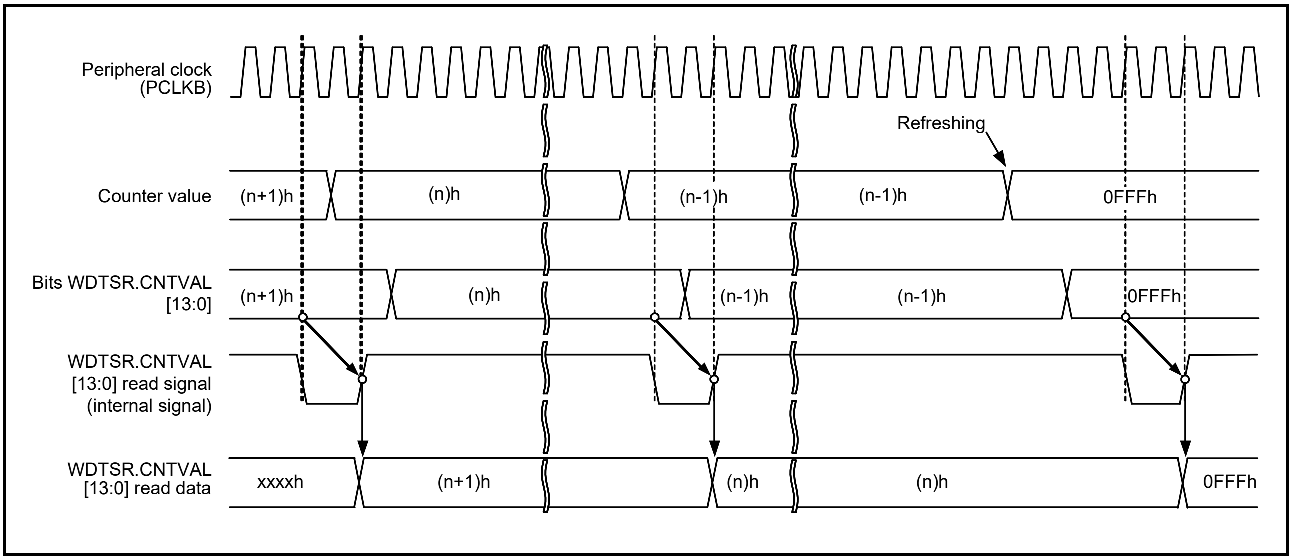


Figure 27.7 **Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0100b and WDTCR.TOPS[1:0] = 01b**

27.3.7 Associations between Option Function Select Register 0 (OFS0) and WDT Registers

Table 27.5 lists the associations between Option Function Select register 0 (OFS0), used in auto start mode, and the registers used in register start mode. Do not change the OFS0 register settings during WDT operation. For details on Option Function Select register 0 (OFS0), see section 7, Option Function Select Register 0 (OFS0).

Table 27.5 **Association between Option Function Select register 0 (OFS0) and the WDT registers**

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period select	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio select	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position select	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position select	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output select	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS
Count stop	Sleep-mode count stop control	OFS0.WDTSTPCTL	WDTCSTPR.SLCSTP

27.4 Link Operation by ELC

The WDT is capable of a link operation for the previously specified module when interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow or refresh error.

An event signal is output regardless of the setting in the WDTRCR.RSTIRQS bit in register start mode or the OFS0.WDTRSTIRQS bit in auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error Flag (WDTSR.REFEF) or Underflow Flag (WDTSR.UNDFF) is 1. For details, see section 19, Event Link Controller (ELC).

27.5 Usage Notes

27.5.1 Restrictions on the ICU Event Link Setting Register n (IELSRn) Setting

Setting 47h to the ICU Event Link Setting Register n (IELSRn.IELS[8:0] bits) is prohibited when enabling the WDT reset assertion (OFS0.WDTRSTIRQS = 1 or WDTRCR.RSTIRQS = 1) or when enabling the event link operation (47h is set to ELSRm.ELS[8:0]).

28. Independent Watchdog Timer (IWDT)

28.1 Overview

The Independent Watchdog Timer (IWDT) is a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT can be used to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates using an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a failsafe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT as follows:

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by PCLKB)
- IWDT does not support the register start mode
- When transitioning to a low power mode (excluding Deep Software Standby mode), the OFS0.IWDTSTPCTL bit can be used to select whether to stop the counter or not.

Table 28.1 lists the IWDT specifications and Figure 28.1 shows a block diagram.

Table 28.1 IWDT specifications

Parameter	Specifications
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	• Counting automatically starts after a reset
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated (and counting restarts automatically)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
IWDT reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the IWDTSR register
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output
Auto start mode	Configurable to the following triggers: • Clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Timeout period of the IWDT (OFS0.IWDTTOPS[1:0] bits) • Window start position in the IWDT (OFS0.IWDTRPSS[1:0] bits) • Window end position in the IWDT (OFS0.IWDTRPES[1:0] bits) • Reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Down-count stop function on transition to Sleep mode, Software Standby mode, or Snooze mode (OFS0.IWDTSTPCTL bit)

Note 1. This must satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

To use the IWDT, you must supply the IWDT-dedicated clock (IWDTCLK). The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

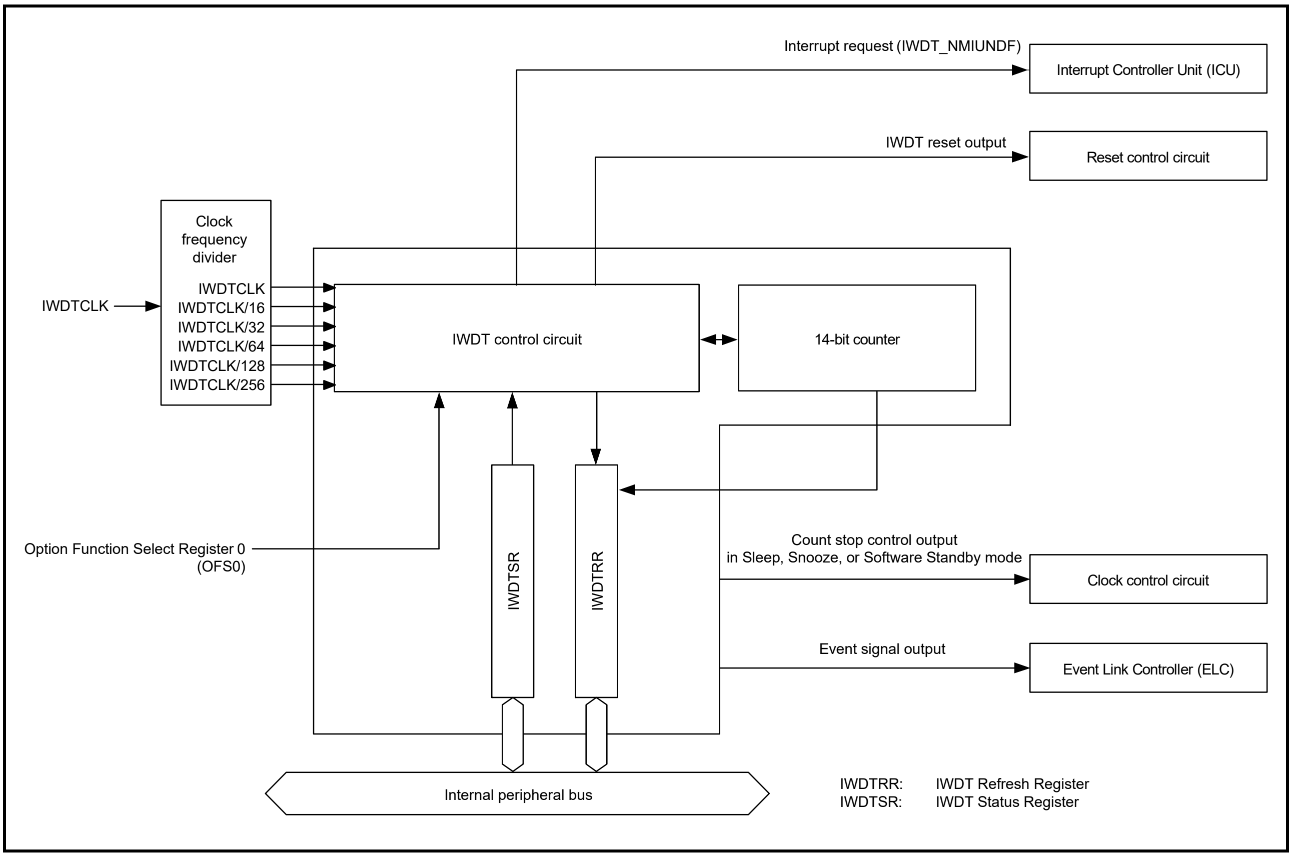


Figure 28.1 IWDT block diagram

28.2 Register Descriptions

28.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 4004 4400h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the down-counter of the IWDT. The down-counter of the IWDT is refreshed by writing 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period. After the counter is refreshed, it starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]) in Option Function Select Register 0 (OFS0).

When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh. For details on the refresh operation, see section 28.3.2, Refresh Operation.

28.2.2 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 4004 4404h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred.	R/(W)*1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred.	R/(W)*1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] bits (Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the counter. A value of 1 indicates that the down-counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N+2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of the flag is ignored for (N+2) IWDTCLK cycles after an underflow. N is specified in the IWDTCKS[3:0] bits as follows:

- When IWDTCKS[3:0] = 0000b, N = 1
- When IWDTCKS[3:0] = 0010b, N = 16
- When IWDTCKS[3:0] = 0011b, N = 32
- When IWDTCKS[3:0] = 0100b, N = 64
- When IWDTCKS[3:0] = 1111b, N = 128
- When IWDTCKS[3:0] = 0101b, N = 256

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N+2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of the flag is ignored for (N+2) IWDTCLK cycles after a refresh error. N is specified in the IWDTCKS[3:0] bits as follows:

- When IWDTCKS[3:0] = 0000b, N = 1
- When IWDTCKS[3:0] = 0010b, N = 16
- When IWDTCKS[3:0] = 0011b, N = 32
- When IWDTCKS[3:0] = 0100b, N = 64
- When IWDTCKS[3:0] = 1111b, N = 128
- When IWDTCKS[3:0] = 0101b, N = 256.

28.2.3 Option Function Select Register 0 (OFS0)

For information on Option Function Select Register 0 (OFS0), see section 7.2.1, Option Function Select Register 0 (OFS0).

IWDTTOPS[1:0] bits (IWDT Timeout Period Select)

The IWDTTOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 128, 512, 1024, or 2048 cycles, taking the divided clock specified in the IWDTCKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits determines the number of IWDTCLK cycles until the counter underflows.

Table 28.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit settings, the timeout period, and the number of IWDTCLK cycles.

Table 28.2 Timeout period settings

IWDTCKS[3:0] bits				IWDTTOPS[1:0] bits				
b7	b6	b5	b4	b1	b0	Clock division ratio	Timeout period (number of cycles)	IWDTCLK cycles
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1,024	1,024
				1	1		2,048	2,048
0	0	1	0	0	0	IWDTCLK/16	128	2,048
				0	1		512	8,192
				1	0		1,024	16,384
				1	1		2,048	32,768
0	0	1	1	0	0	IWDTCLK/32	128	40,96
				0	1		512	16,384
				1	0		1,024	32,768
				1	1		2,048	65,536
0	1	0	0	0	0	IWDTCLK/64	128	8,192
				0	1		512	32,768
				1	0		1,024	65,536
				1	1		2,048	131,072
1	1	1	1	0	0	IWDTCLK/128	128	16,384
				0	1		512	65,536
				1	0		1,024	131,072
				1	1		2,048	262,144
0	1	0	1	0	0	IWDTCLK/256	128	32,768
				0	1		512	131,072
				1	0		1,024	262,144
				1	1		2,048	5242,88

IWDTCKS[3:0] bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

The IWDTCKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the IWDT-dedicated clock (IWDTCLK) divided by 1, 16, 32, 64, 128, and 256. Combined with the IWDTTOPS[1:0] bit setting, this allows the IWDT to be configured to a count period between 128 and 524288 IWDTCLK cycles.

IWDTRPES[1:0] bits (IWDT Window End Position Select)

The IWDTRPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

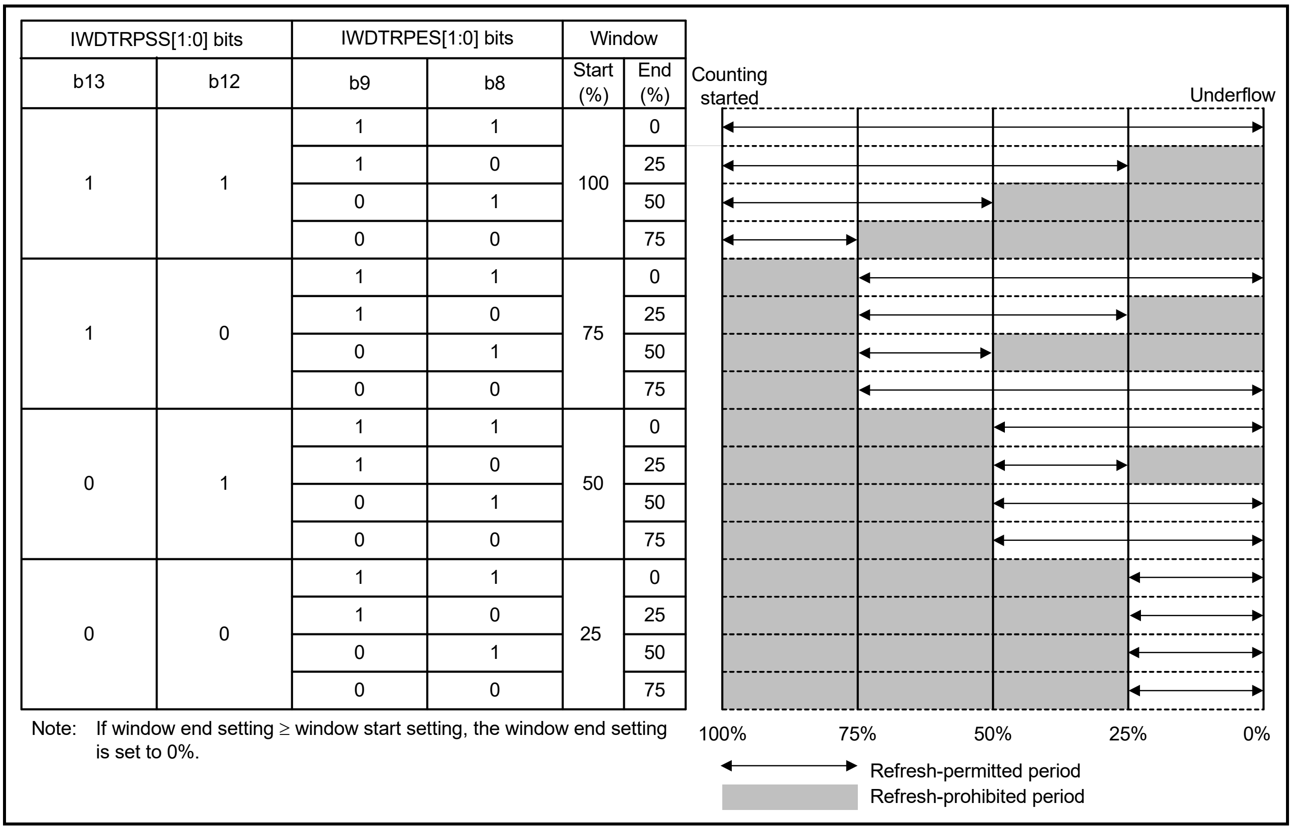
IWDTRPSS[1:0] bits (IWDT Window Start Position Select)

The IWDTRPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the window end position. If the window start position is less than or equal to the window end position, the window end position is set to 0%.

Table 28.3 lists the counter values for the window start and end positions, and Figure 28.2 shows the refresh-permitted period set in the IWDTRPSS[1:0], IWDTRPES[1:0], and IWDTTOPS[1:0] bits.

Table 28.3 Relationship between the timeout period and window start and end counter values

IWDTTOPS[1:0] bits		Timeout period		Window start and end counter value			
b1	b0	Cycles	Counter value	100%	75%	50%	25%
0	0	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1,024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2,048	07FFh	07FFh	05FFh	03FFh	01FFh



IWDTRPSS[1:0] bits		IWDTRPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

Figure 28.2 IWDTRPSS[1:0] and IWDTRPES[1:0] bit settings and refresh-permitted period

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit specifies the behavior when an underflow or a refresh error occurs. Setting 1 selects reset output. Setting 0 selects non-maskable interrupt or interrupt.

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit controls whether to stop counting on transition to Sleep, Snooze, or Software Standby mode.

28.3 Operation

28.3.1 Auto Start Mode

When the IWDT Start Mode Select bit (OFS0.IWDTSTRT) is 0, auto start mode is selected. Otherwise, the IWDT is disabled.

Within the reset state, the following values in Option Function Select Register 0 (OFS0) are set in the IWDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control on transition to the low power modes.

When the reset state is released, the down-counter automatically starts counting down from the value set in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]).

After that, as long as the program continues normal operation and the counter is refreshed within the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The IWDT does not output the reset signal as long as this procedure continues. However, if the counter underflows because the program crashes, or because a refresh error occurs when an attempt is made to refresh outside the refresh-permitted period, the IWDT asserts the reset signal or non-maskable interrupt request/interrupt request (IWDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle, and restarts the count. Reset output or interrupt request output can be selected in the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the IWDT Underflow/Refresh Error Interrupt Enable bit (NMIER.IWDTEN).

Figure 28.3 shows an example of operation under the following conditions:

- Auto start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b).

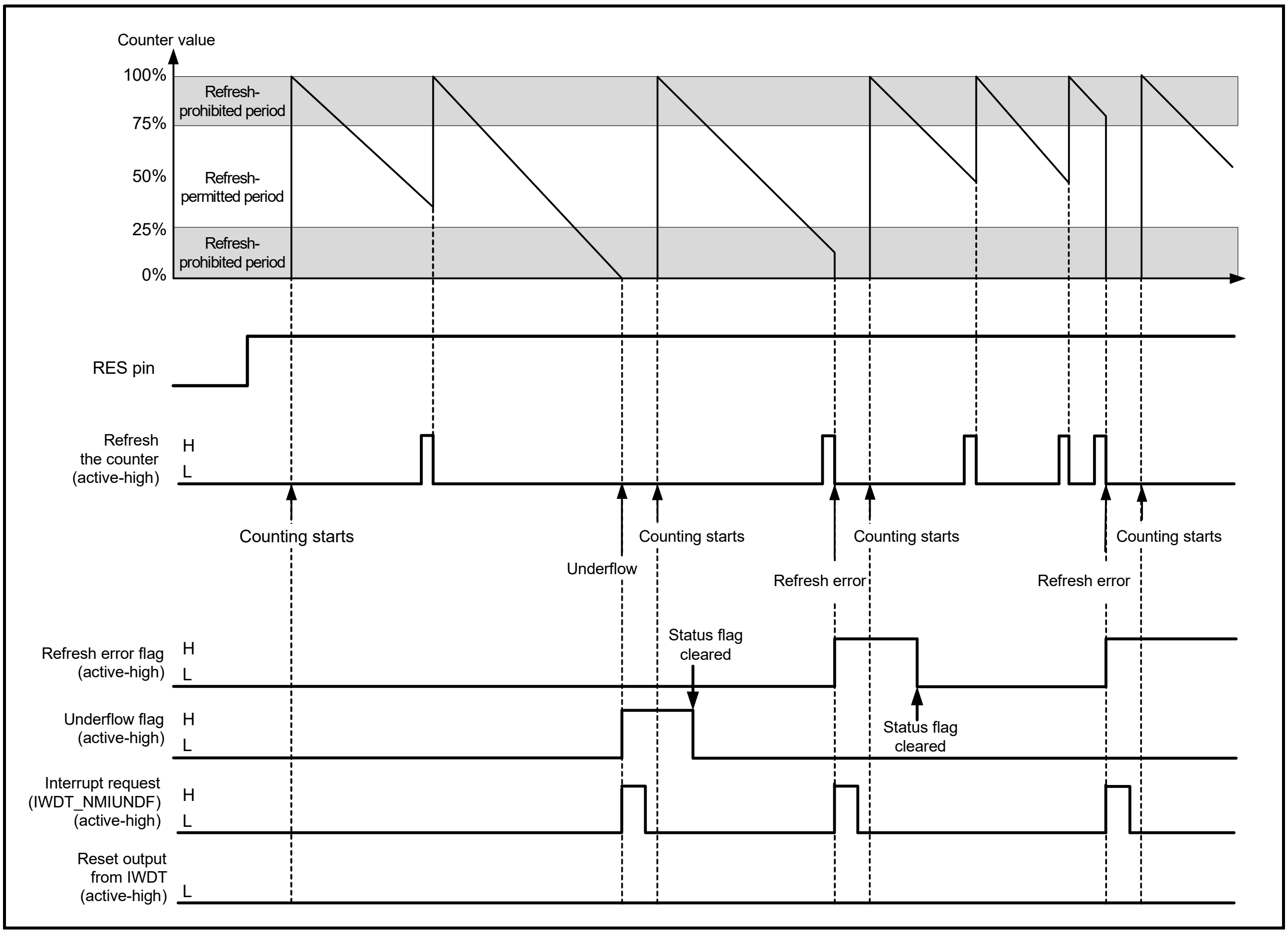


Figure 28.3 Operation example in auto start mode

28.3.2 Refresh Operation

The down-counter is refreshed by writing the values 00h and FFh to the IWDT Refresh Register (IWDTRR). If a value other than FFh is written after 00h, the down-counter is not refreshed. If an invalid value is written, correct refreshing resumes on a write of 00h and FFh to the IWDTRR register.

When writes are made in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied. Writes of 00h (n-1th time) → 00h (nth time) → FFh are valid, and the refresh is performed correctly. Even when the first value written before 00h is not 00h, correct refreshing is performed as long as the operation contains the write sequence of 00h → FFh.

Correct refreshing is also performed when a register other than IWDTRR is accessed or IWDTRR is read between writing 00h and writing FFh to IWDTRR. Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by the FFh write. For this reason, correct refreshing is performed even when 00h is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

- 00h → FFh
- 00h (n-1th time) → 00h (nth time) → FFh
- 00h → access to another register or read from IWDTRR → FFh.

[Example write sequences that are invalid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh.

After FFh is written to the IWDT Refresh Register (IWDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting (the IWDT-dedicated clock frequency division ratio select bits (OFS0.IWDTCKS[3:0]) determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up 1 counting cycle). To meet this requirement, complete writing FFh to IWDTRR 4 count cycles before the end of the refresh-permitted period or a counter underflow. The value of the counter can be checked in the counter bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to IWDTRR before 1FFFh is reached (at 2002h, for example), refreshing occurs if FFh is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits reaches 1FFFh.
- When the window end position is set to 1FFFh, refreshing occurs if 2003h (four count cycles before 1FFFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after a write of 00h → FFh to IWDTRR.
- When the refresh-permitted period continues until count 0000h, refreshing can be performed immediately before an underflow. In this case, if 0003h (four count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after a write of 00h → FFh to IWDTRR, no underflow occurs and refreshing is performed.

Figure 28.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

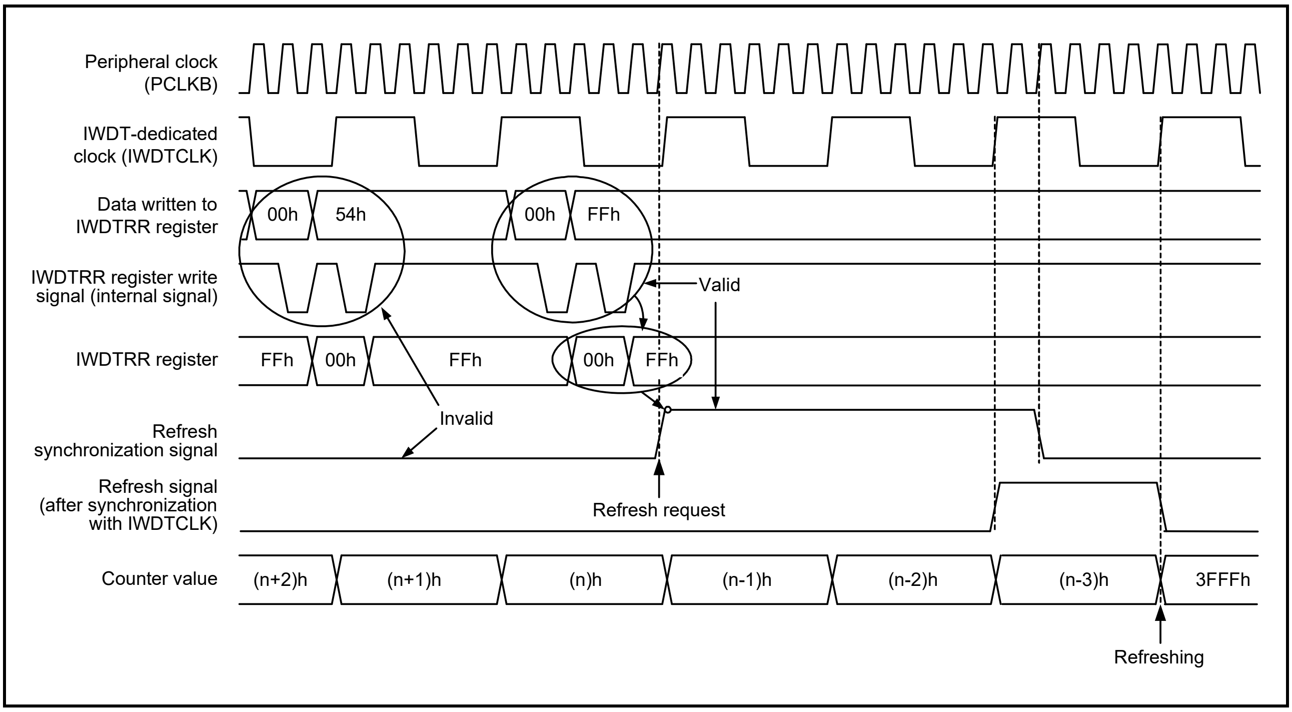


Figure 28.4 IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b and OFS0.IWDTTOPS[1:0] = 11b

28.3.3 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDFF) flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT. After a release from the reset state or interrupt request generation, read the IWDTSR.REFEF and UNDFF flags to check for the reset or interrupt source. For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving the status flags unchanged does not affect operation. If the flags are not cleared on the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written. After 0 is written to each flag, up to 3 IWDTCLK cycles and 2 PCLKB cycles are required before the value is reflected.

28.3.4 Reset Output

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS) in Option Function Select Register 0 (OFS0) is set to 1, a reset signal is output when an underflow in the down-counter or a refresh error occurs. Counting down starts automatically after the reset output.

28.3.5 Interrupt Sources

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS) in Option Function Select Register 0 (OFS0) is set to 0, an interrupt signal (IWDT_NMIUNDF) is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 14, Interrupt Controller Unit (ICU).

Table 28.4 IWDT interrupt source

Name	Interrupt source	DTC activation	DMAC activation
IWDT_NMIUNDF	• Down-counter underflow • Refresh error	Not possible	Not possible

28.3.6 Reading the Down-Counter Value

Because the counter is the IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT Status register. Check these bits to obtain the counter value indirectly. Reading the counter value requires multiple PCLKB clock cycles (up to four clock cycles), and the read counter value might differ from the actual counter value by a value of one count.

Figure 28.5 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

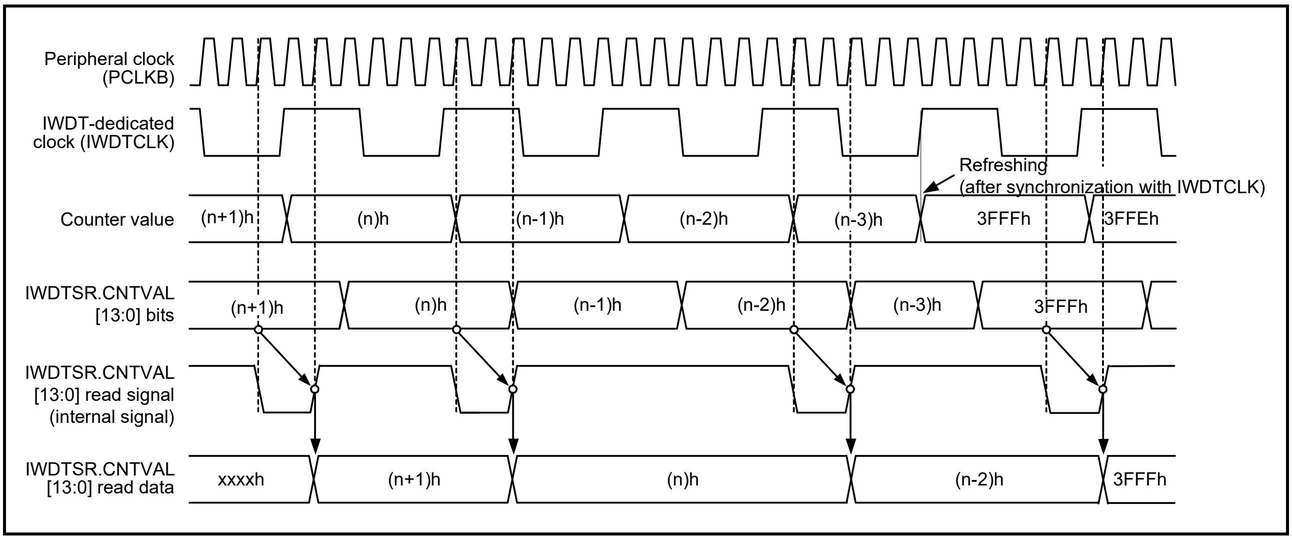


Figure 28.5 Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b and OFS0.IWDTTOPS[1:0] = 11b

28.4 Output to the Event Link Controller (ELC)

The IWDT is capable of link operation for a specified module when the interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow or refresh error.

An event signal is output regardless of the setting in the OFS0.WDTRSTIRQS bit. An event signal can also be output when the next interrupt source is generated while the Refresh Error Flag (IWDTSR.REFEF) or Underflow Flag (IWDTSR.UNDFF) is 1. For details, see section 19, Event Link Controller (ELC).

28.5 Usage Notes

28.5.1 Refresh Operations

While configuring the refresh time, consider variations in the range of errors given the accuracy of PCLKB and IWDTCLK. Set values that ensure refreshing is possible.

28.5.2 Constraints on the Clock Division Ratio Setting

Satisfy the following required frequency of the peripheral module clock (PCLKB):

PCLKB ≥ 4 × (the frequency of the count clock source after division).

29. Ethernet MAC Controller (ETHERC)

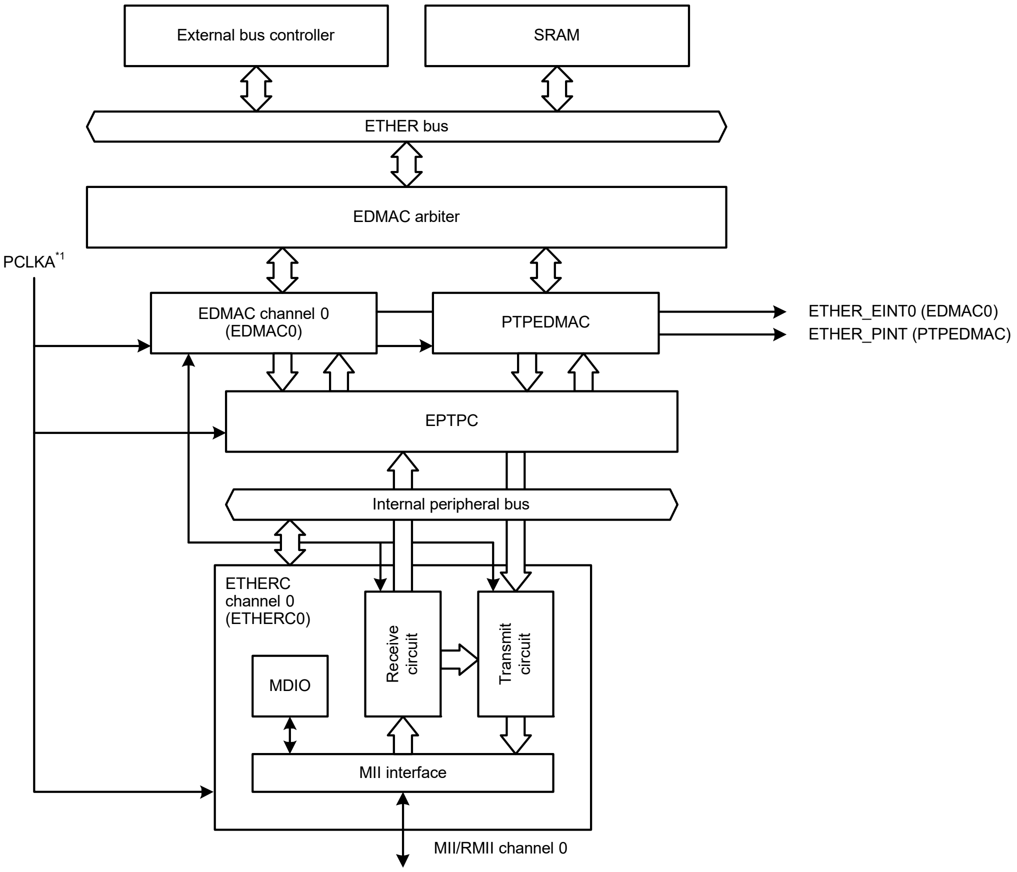
29.1 Overview

The MCU provides a one-channel Ethernet Controller (ETHERC) compliant with the Ethernet or IEEE802.3 Media Access Control (MAC) layer protocol. ETHERC channel has one channel of the MAC layer interface. Connecting the MCU to the physical layer LSI (PHY-LSI) allows transmission and reception of frames compliant with the Ethernet/ IEEE802.3 standard. The ETHERC is connected through the Ethernet PTP Controller (EPTPC) to the Ethernet DMA Controller (EDMAC), so data can be transferred without using the CPU. When the EPTPC is not used, bypass the EPTPC by setting the bypass registers in the EPTPC. See section 30.2.79, Bypass 1588 Module Register (BYPASS).

Table 29.1 lists the ETHERC specifications, Figure 29.1 shows the configuration, and Table 29.2 lists the I/O pins. Figure 29.2 and Figure 29.3 show examples connections of the MCU to an external PHY-LSI.

Table 29.1 ETHERC specifications

Parameter	Specifications
Number of channels	One channel
Protocol	Flow control compliant with IEEE802.3x
Data transmission/reception	Frames compliant with the Ethernet/IEEE802.3 standard can be transmitted and received
Bit rate	Supports 10 Mbps and 100 Mbps
Operation modes	Supports full-duplex and half-duplex modes
Interfaces	Media Independent Interface (MII), Reduced Media Independent Interface (RMII), compliant with the IEEE802.3u standard
Functions	Magic Packet^{TM} detection, Wake-on-LAN (WOL) signal output



Note 1. When using the ETHERC, set the clock to ICLK = PCLKA.
12.5 MHz ≤ PCLKA ≤ 120 MHz.

Figure 29.1 ETHERC configuration

Table 29.2 ETHERC I/O pins

Operating mode	Pin name	I/O	Description
MII	ET0_TX_CLK *1	Input	Transmit clock Timing reference signal for outputting the ET0_TX_EN, ET0_ETXD3 to ET0_ETXD0, and ET0_TX_ER signals.
	ET0_RX_CLK *1	Input	Receive clock Timing reference signal for inputting the ET0_RX_DV, ET0_ERXD3 to ET0_ERXD0, and ET0_RX_ER signals.
	ET0_TX_EN *1	Output	Transmit data valid This signal indicates that valid transmit data was output on pins ET0_ETXD3 to ET0_ETXD0.
	ET0_ETXD3 to ET0_ETXD0 *1	Output	4-bit transmit data
	ET0_TX_ER *1	Output	Transmit error This signal notifies the PHY-LSI that an error occurred during transmission.
	ET0_RX_DV *1	Input	Receive data valid This signal indicates that valid receive data is on pins ET0_ERXD3 to ET0_ERXD0.
	ET0_ERXD3 to ET0_ERXD0 *1	Input	4-bit receive data
	ET0_RX_ER *1	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC.
	ET0_CRS *1	Input	Carrier sense
	ET0_COL *1	Input	Collision detection signal
	ET0_MDC *1	Output	Management data clock Reference clock signal for transfer of information on the ET0_MDIO pin.
	ET0_MDIO *1	I/O	Management data Input/Output Bidirectional data signal for exchanging management data with the PHY-LSI.
	ET0_LINKSTA	Input	Link status input from the PHY-LSI
	ET0_EXOUT	Output	General output pin
	ET0_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received.
RMII	REF50CK0 *2	Input	Reference clock Timing reference signal for the RMII0_TXD_EN, RMII0_TXD1 to RMII0_TXD0, RMII0_CRS_DV, RMII0_RXD1 to RMII0_RXD0, and RMII0_RX_ER pins.
	RMII0_TXD_EN *2	Output	Transmit data valid This signal indicates that valid transmit data was output on the RMII0_TXD1 and RMII0_TXD0 pins.
	RMII0_TXD1 to RMII0_TXD0 *2	Output	2-bit transmit data
	RMII0_CRS_DV *2	Input	Carrier sense/receive data valid This signal indicates that valid receive data is on the RMII0_RXD1 and RMII0_RXD0 pins.
	RMII0_RXD1 to RMII0_RXD0 *2	Input	2-bit receive data
	RMII0_RX_ER *2	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC. See the note in section 29.5.2, Input to RMII0_RX_ER Pin while RMII Is Selected.
	ET0_MDC *2	Output	Management data clock Reference clock signal for transfer of information on the ET0_MDIO pin
	ET0_MDIO *2	I/O	Management data Input/Output Bidirectional data signal for exchanging management data with the PHY-LSI.
	ET0_LINKSTA	Input	Link status input from the PHY-LSI.
	ET0_EXOUT	Output	General output pin
	ET0_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received.

Note 1. MII signal compliant with IEEE802.3u.
Note 2. RMII signal compliant with IEEE802.3u.

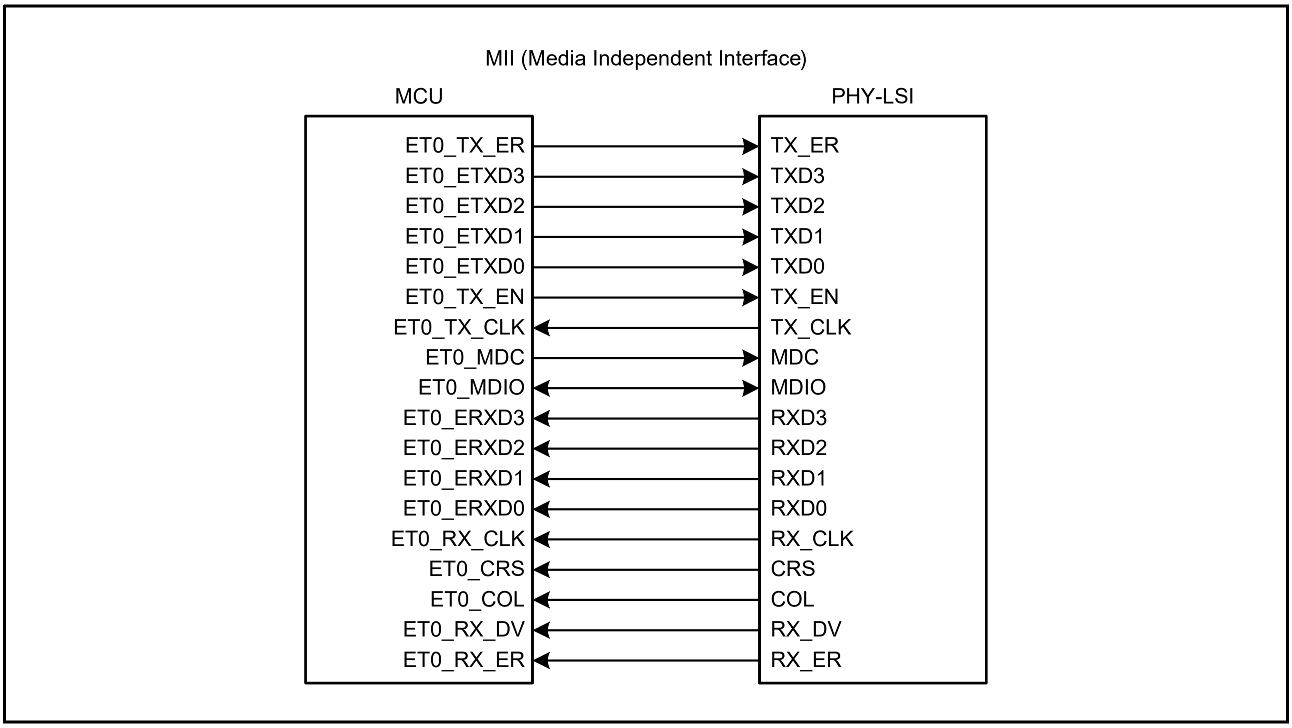


Figure 29.2 Example of connection with PHY-LSI for MII

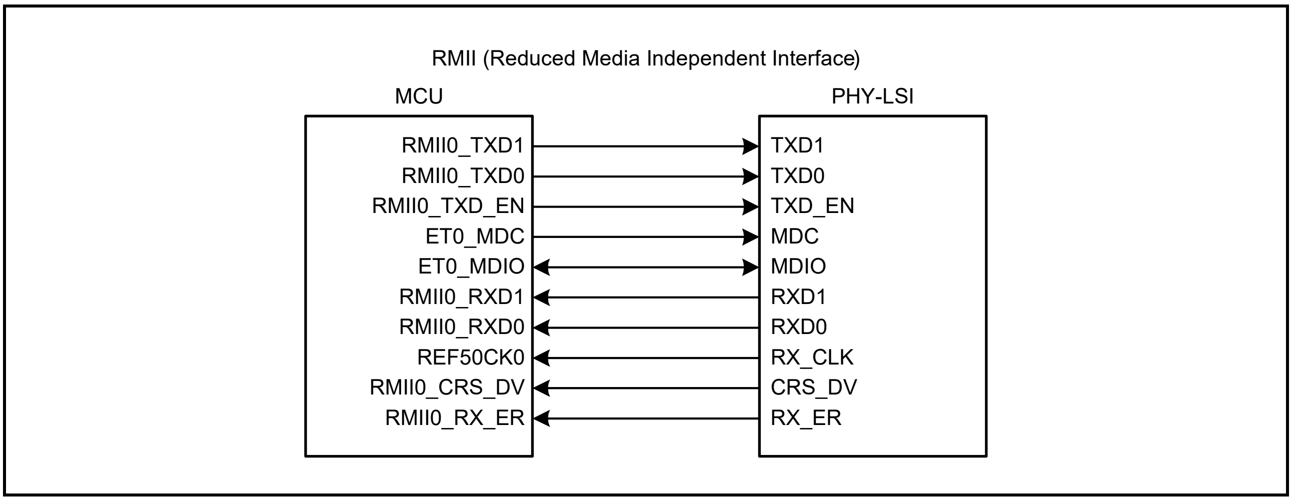


Figure 29.3 Example of connection with PHY-LSI for RMII

29.2 Register Descriptions

29.2.1 ETHERC Mode Register (ECMR)

Address(es): ETHERC0.ECMR 4006 4100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PRCEF	—	—	MPDE	—	—	RE	TE	—	ILB	RTM	DM	PRM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PRM	Promiscuous Mode	0: Disable promiscuous mode 1: Enable promiscuous mode.	R/W
b1	DM	Duplex Mode	0: Half-duplex mode 1: Full-duplex mode.	R/W
b2	RTM	Bit Rate	0: 10 Mbps 1: 100 Mbps.	R/W
b3	ILB	Internal Loopback Mode	0: Perform normal data transmission or reception 1: Loop data back in the ETHERC when full-duplex mode is selected.	R/W
b4	—	Reserved	The read value is 0. The write value should be 0.	R/W
b5	TE	Transmission Enable	0: Disable transmit function 1: Enable transmit function.	R/W
b6	RE	Reception Enable	0: Disable receive function 1: Enable receive function.	R/W
b8, b7	—	Reserved	The read value is 0. The write value should be 0.	R/W
b9	MPDE	Magic Packet Detection Enable	0: Disable Magic Packet detection 1: Enable Magic Packet detection.	R/W
b11, b10	—	Reserved	The read value is 0. The write value should be 0.	R/W
b12	PRCEF	CRC Error Frame Receive Mode	0: Notify EDMAC of a CRC error 1: Do not notify EDMAC of a CRC error.	R/W
b15 to b13	—	Reserved	The read value is 0. The write value should be 0.	R/W
b16	TXF	Transmit Flow Control Operating Mode	0: Disable automatic PAUSE frame transmission (PAUSE frame is not automatically transmitted) 1: Enable automatic PAUSE frame transmission (PAUSE frame is automatically transmitted as required).	R/W
b17	RXF	Receive Flow Control Operating Mode	0: Disable PAUSE frame detection 1: Enable PAUSE frame detection.	R/W
b18	PFR	PAUSE Frame Receive Mode	0: Do not transfer PAUSE frame to the EDMAC 1: Transfer PAUSE frame to the EDMAC.	R/W
b19	ZPF	0 Time PAUSE Frame Enable	0: Do not use PAUSE frames that contain a pause_time parameter of 0 1: Use PAUSE frames that contains a pause_time parameter of 0.	R/W
b20	TPC	PAUSE Frame Transmit	0: Transmit PAUSE frame even during a PAUSE period 1: Do not transmit PAUSE frame during a PAUSE period.	R/W
b31 to b21	—	Reserved	The read value is 0. The write value should be 0.	R/W

The ECMR register controls ETHERC operation. Except for the TE and RE bits, set the bits in this register during initialization after a reset. When rewriting this register outside the initialization process, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

PRM bit (Promiscuous Mode)

When the PRM bit is set to 1, the ETHERC operates in promiscuous mode, where all Ethernet frames are received. In promiscuous mode, the ETHERC receives all valid frames regardless of whether the address matches the destination or broadcast address and regardless of the multicast bit setting.

RTM bit (Bit Rate)

The RTM bit sets the bit rate when the RMII is selected.

ILB bit (Internal Loopback Mode)

When the ILB bit is set to 1, transmit frames can be looped back in the MCU. Set the DM bit to 1 (full-duplex mode) to perform a loopback test.

TE bit (Transmission Enable)

When the TE bit is set to 1, the ETHERC transmit function is enabled. When the TE bit is set to 0, the transmit function is disabled after the frame being processed is completely transmitted.

RE bit (Reception Enable)

When the RE bit is set to 1, the ETHERC receive function is enabled. When the RE bit is set to 0, the receive function is disabled after the frame being processed is completely received.

PRCEF bit (CRC Error Frame Receive Mode)

When the PRCEF bit is set to 1, the EDMAC is not notified that a CRC error has occurred even when the error is detected in a receive frame. Accordingly, the EDMAC0.EESR.CERF flag and RFS0 bit in receive descriptor 0 (RD0) do not become 1.

ZPF bit (0 Time PAUSE Frame Enable)

When the ZPF bit is 1, a PAUSE frame with a pause_time parameter of 0 is transmitted when a PAUSE frame transmit request is canceled before the PAUSE time of the previously transmitted PAUSE frame has elapsed. After the PAUSE frame containing the pause_time parameter of 0 is received, the ETHERC is ready for transmission.

When the ZPF bit is 0, even if the PAUSE frame transmit request from the receive FIFO is canceled, the next PAUSE frame is not transmitted until the PAUSE time of the previously transmitted PAUSE frame has elapsed. When a PAUSE frame containing a pause_time parameter of 0 is received, it is discarded.

29.2.2 Receive Frame Maximum Length Register (RFLR)

Address(es): ETHERC0.RFLR 4006 4108h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RFL[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	RFL[11:0]	Receive Frame Maximum Length	The set value becomes the maximum frame length. The minimum value that can be set is 1,518 bytes, and the maximum value that can be set is 2,048 bytes. Values less than 1,518 bytes are regarded as 1,518 bytes, and values larger than 2,048 bytes are regarded as 2,048 bytes.	R/W
b31 to b12	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RFLR register specifies the maximum frame length that can be received by the MCU. Set the length in bytes. Do not rewrite this register while the ECMR.RE bit is 1 (receive function enabled).

RFL[11:0] bits (Receive Frame Maximum Length)

The RFL[11:0] bits set the frame length to be checked. The frame length is the number of bytes in a field, extending from the destination address to the frame check sequence [FCS] of the received frame. When this length exceeds the RFL[11:0] bit value, the EDMAC is notified of a frame-too-long error, and the excess data is discarded.

29.2.3 ETHERC Status Register (ECSR)

Address(es): ETHERC0.ECSR 4006 4110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	BFR	PSRTO	—	LCHNG	MPD	ICD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ICD	False Carrier Detect Flag	0: PHY-LSI has not detected a false carrier on the line 1: PHY-LSI detected a false carrier on the line.	R/W *1
b1	MPD	Magic Packet Detect Flag	0: Magic Packet not detected 1: Magic Packet detected.	R/W *1
b2	LCHNG	Link Signal Change Flag	0: Change in the ET0_LINKSTA signal not detected 1: Change in the ET0_LINKSTA signal detected (high to low, or low to high).	R/W *1
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	PSRTO	PAUSE Frame Retransmit Over Flag	0: PAUSE frame retransmit count has not reached the upper limit 1: PAUSE frame retransmit count reached the upper limit.	R/W *1
b5	BFR	Continuous Broadcast Frame Reception Flag	0: Continuous reception of broadcast frames not detected 1: Continuous reception of broadcast frames detected.	R/W *1
b31 to b6	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. Write 1 to clear the flag.

The ECSR register indicates the status of the ETHERC. When any flag in the ECSR register is set to 1 while the associated bit in the ECSIPR register is 1 (interrupt enabled), the EDMAC0.EESR.ECI flag is set to 1.

ICD flag (False Carrier Detect Flag)

The ICD flag indicates that the PHY-LSI has detected a false carrier on the line. The flag is set to 1 when a receive error signal shown in Figure 29.11 is received from the PHY-LSI. The information might not be correct when signals input from the PHY-LSI change faster than software recognizes the change. Check the timing of the PHY-LSI.

LCHNG flag (Link Signal Change Flag)

The LCHNG flag indicates that the ET0_LINKSTA signal input from the PHY-LSI has changed from high to low, or from low to high. Check the PSR.LMON flag for the current link status. See section 29.5.1, Preventing the LCHNG Flag from Erroneously Setting to 1 for more information.

PSRTO flag (PAUSE Frame Retransmit Over Flag)

The PSRTO flag indicates that the number of retransmissions reached the value set in the TPAUSER register when retransmitting a PAUSE frame while automatic PAUSE frame transmission is enabled.

29.2.4 ETHERC Interrupt Enable Register (ECSIPR)

Address(es): ETHERC0.ECSIPR 4006 4118h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	BFSIPR	PSRTOIP	—	LCHNGIP	MPDIP	ICDIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ICDIP	False Carrier Detect Interrupt Enable	0: Disable interrupt notification 1: Enable interrupt notification.	R/W
b1	MPDIP	Magic Packet Detect Interrupt Enable	0: Disable interrupt notification 1: Enable interrupt notification.	R/W
b2	LCHNGIP	LINK Signal Change Interrupt Enable	0: Disable interrupt notification 1: Enable interrupt notification.	R/W
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	PSRTOIP	PAUSE Frame Retransmit Over Interrupt Enable	0: Disable interrupt notification 1: Enable interrupt notification.	R/W
b5	BFSIPR	Continuous Broadcast Frame Reception Interrupt Enable	0: Disable interrupt notification 1: Enable interrupt notification.	R/W
b31 to b6	—	Reserved	The read value is 0. The write value should be 0.	R/W

The ECSIPR register selects whether to notify the EDMAC of the status indicated in the ECSR register. Each bit is associated with the flag with the same bit number in the ECSR register.

29.2.5 PHY Interface Register (PIR)

Address(es): ETHERC0.PIR 4006 4120h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MDC	MII/RMII Management Data Clock	This value is output from the ET0_MDC pin to supply the management data clock to the MII or RMII.	R/W
b1	MMD	MII/RMII Management Mode	0: Read 1: Write.	R/W
b2	MDO	MII/RMII Management Data-Out	This value is output from the ET0_MDIO pin when the MMD bit is 1 (write), and not when MMD is 0 (read).	R/W
b3	MDI	MII/RMII Management Data-In	This bit indicates the level of the ET0_MDIO pin. The write value should be 0.	R
b31 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The PIR register accesses registers in the PHY-LSI through the MII or RMII. The management clock and management data are controlled by software. See section 29.3.4, Accessing the MII and RMII Registers for details on accessing the MII and RMII registers.

29.2.6 PHY Status Register (PSR)

Address(es): ETHERC0.PSR 4006 4128h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Bit name	Description	R/W
b0	LMON	ET0_LINKSTA Pin Status Flag	The link status can be read by connecting the link signal output from the PHY-LSI to the ET0_LINKSTA pin. For details on the polarity, see the specifications of the connected PHY-LSI.	R
b31 to b1	—	Reserved	The read value is 0.	R

The PSR register monitors interface signals from the PHY-LSI.

29.2.7 Random Number Generation Counter Upper Limit Setting Register (RDMLR)

Address(es): ETHERC0.RDMLR 4006 4140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	RMD[19:16]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b19 to b0	RMD[19:0]	Random Number Generation Counter	00000h: Normal operation 00001h to FFFFFh: Setting prohibited.	R/W
b31 to b20	—	Reserved	The read value is 0. The write value should be 0.	R/W

The RDMLR register specifies the maximum value for the counter used in the random number generator. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled).

29.2.8 Interpacket Gap Register (IPGR)

Address(es): ETHERC0.IPGR 4006 4150h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IPG[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	IPG[4:0]	Interpacket Gap	00h: 16 bit times 01h: 20 bit times : : 14h: 96 bit times (initial value) : : 1Fh: 140 bit times.	R/W
b31 to b5	—	Reserved	The read value is 0. The write value should be 0.	R/W

The IPGR register specifies the interpacket gap (IPG) value. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). See section 29.3.6, Adjusting Transmission Efficiency by Changing the IPG for details on the IPG.

29.2.9 Automatic PAUSE Frame Register (APR)

Address(es): ETHERC0.APR 4006 4154h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	AP[15:0]	Automatic PAUSE Time Setting	These bits set the value of the pause_time parameter for PAUSE frames that are automatically transmitted. Transmission is not performed until the set value multiplied by 512 bit times has elapsed.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The APR register specifies the PAUSE time for PAUSE frames that are automatically transmitted. The value set in the APR register is used for the pause_time parameter of the PAUSE frame. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled).

29.2.10 Manual PAUSE Frame Register (MPR)

Address(es): ETHERC0.MPR 4006 4158h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MP[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Bit name	Description	R/W
b15 to b0	MP[15:0]	Manual PAUSE Time Setting	These bits set the value of the pause_time parameter for PAUSE frames that are manually transmitted. Transmission is not performed until the set value multiplied by 512 bit times has elapsed. The read value is undefined.	W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	W

The MPR register specifies the PAUSE time for PAUSE frames that are manually transmitted. The value set in the MPR register is used for the pause_time parameter of the PAUSE frame. When a value is set to this register, a PAUSE frame is transmitted. Rewrite this register while the ECMR.TE bit is 1 (transmit function enabled).

29.2.11 Received PAUSE Frame Counter (RFCF)

Address(es): ETHERC0.RFCF 4006 4160h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RPAUSE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	RPAUSE[7:0]	Received PAUSE Frame Count	Number of received PAUSE frames.	R
b31 to b8	—	Reserved	The read value is 0.	R

The RFCF register is a counter that indicates the number of received PAUSE frames. The counter is reset after this register is read.

29.2.12 PAUSE Frame Retransmit Count Setting Register (TPAUSER)

Address(es): ETHERC0.TPAUSER 4006 4164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TPAUSE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	TPAUSE[15:0]	Automatic PAUSE Frame Retransmit Setting	0000h: Number of retransmissions is unlimited 0001h: Maximum number of retransmissions is 1 : : FFFFh: Maximum number of retransmissions is 65,535.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The TPAUSER register selects the maximum number of times a PAUSE frame is automatically transmitted. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled).

29.2.13 PAUSE Frame Retransmit Counter (TPAUSECR)

Address(es): ETHERC0.TPAUSECR 4006 4168h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TXP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	TXP[7:0]	PAUSE Frame Retransmit Count	Number of times a PAUSE frame was retransmitted.	R
b31 to b8	—	Reserved	The read value is 0.	R

The TPAUSECR register is a counter that indicates the number of times a PAUSE frame was automatically retransmitted. The counter is reset after this register is read.

29.2.14 Broadcast Frame Receive Count Setting Register (BCFRR)

Address(es): ETHERC0.BCFRR 4006 416Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BCF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	BCF[15:0]	Broadcast Frame Continuous Receive Count Setting	0000h: Number of receptions is unlimited 0001h: Receive 1 frame. : : FFFFh: Receive 65,535 frames.	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The BCFRR register specifies the number of times broadcast frames can be received continuously. When the number of received frames exceeds the BCF[15:0] bit value, the excess broadcast frames are discarded. Do not rewrite this register while the EMCR.RE bit is 1 (receive function enabled).

29.2.15 MAC Address Upper Bit Register (MAHR)

Address(es): ETHERC0.MAHR 4006 41C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MAHR[31:0]	MAC Address Upper Bit	See the description following this table	R/W

The MAHR register specifies the upper 32 bits ([47:16]) of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0123 4567h.

Set the MAHR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). When rewriting this register, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

29.2.16 MAC Address Lower Bit Register (MALR)

Address(es): ETHERC0.MALR 4006 41C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MALR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	MALR[15:0]	MAC Address Lower Bit	These bits set the lower 16 bits of the MAC address	R/W
b31 to b16	—	Reserved	The read value is 0. The write value should be 0.	R/W

The MALR register specifies the lower 16 bits of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0000 89ABh.

Set the MALR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). When rewriting this register, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

29.2.17 Transmit Retry Over Counter Register (TROCR)

Address(es): ETHERC0.TROCR 4006 41D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	TROCR[31:0]	Transmit Retry Over Counter	See the description following this table	R/W

The TROCR register is a counter that indicates the number of frames that failed to be retransmitted. The register is incremented by 1 when a frame fails to be retransmitted 15 times. The counter stops when the register value becomes FFFF FFFFh. Writing any value to the TROCR register clears the counter value to 0.

29.2.18 Late Collision Detect Counter Register (CDCR)

Address(es): ETHERC0.CDCR 4006 41D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	CDCR[31:0]	Late Collision Detect Counter	See the description following this table	R/W

The CDCR register is a counter that indicates the number of late collisions that are detected after transmission starts. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the CDCR register clears the counter value to 0.

29.2.19 Lost Carrier Counter Register (LCCR)

Address(es): ETHERC0.LCCR 4006 41D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LCCR[31:0]	Lost Carrier Counter	See the description following this table	R/W

The LCCR register is a counter that indicates the number of times a loss of carrier is detected during frame transmission. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the LCCR register clears the counter value to 0.

29.2.20 Carrier Not Detect Counter Register (CNDCR)

Address(es): ETHERC0.CNDCR 4006 41DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	CNDCR[31:0]	Carrier Not Detect Counter	See the description following this table	R/W

The CNDCR register is a counter that indicates the number of times a carrier is not detected during preamble transmission. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the CNDCR register clears the counter value to 0.

29.2.21 CRC Error Frame Receive Counter Register (CEFCR)

Address(es): ETHERC0.CEFCR 4006 41E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	CEFCR[31:0]	CRC Error Frame Receive Counter	See the description following this table	R/W

The CEFCR register is a counter that indicates the number of received frames in which a CRC error was detected. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the CEFCR register clears the counter value to 0.

29.2.22 Frame Receive Error Counter Register (FRECR)

Address(es): ETHERC0.FRECR 4006 41E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	FRECR[31:0]	Frame Receive Error Counter	See the description following this table	R/W

The FRECR register is a counter that indicates the number of times a frame receive error has occurred. The PHY-LSI notifies the ETHERC of the frame receive error using the ET0_RX_ER pin. The FRECR register increments each time the ET0_RX_ER pin goes high. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the FRECR register clears the counter value to 0.

29.2.23 Too-Short Frame Receive Counter Register (TSFRCR)

Address(es): ETHERC0.TSFRCR 4006 41ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSFRCR[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSFRCR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	TSFRCR[31:0]	Too-Short Frame Receive Counter	See the description following this table	R/W

The TSFRCR register is a counter that indicates the number of times a short frame that is shorter than 64 bytes was received. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the TSFRCR register clears the counter value to 0.

29.2.24 Too-Long Frame Receive Counter Register (TLFRCR)

Address(es): ETHERC0.TLFRCR 4006 41F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TLFRCR[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TLFRCR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	TLFRCR[31:0]	Too-Long Frame Receive Counter	See the description following this table	R/W

The TLFRCR register is a counter that indicates the number of times a long frame that is longer than the RFLR register value was received. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the TLFRCR register clears the counter value to 0.

Note: The TLFRCR register does not increment when a frame is received with an alignment error. In this case, the RFCR register increments.

29.2.25 Received Alignment Error Frame Counter Register (RFCR)

Address(es): ETHERC0.RFCR 4006 41F4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFCR[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFCR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	RFCR[31:0]	Received Alignment Error Frame Counter	See the description following this table	R/W

The RFCR register is a counter that indicates the number of times a frame was received with an alignment error, meaning that it is not an integral number of octets. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the RFCR register clears the counter value to 0.

29.2.26 Multicast Address Frame Receive Counter Register (MAFCR)

Address(es): ETHERC0.MAFCR 4006 41F8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MAFCR[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAFCR[16:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MAFCR[31:0]	Multicast Address Frame Receive Counter	See the description following this table	R/W

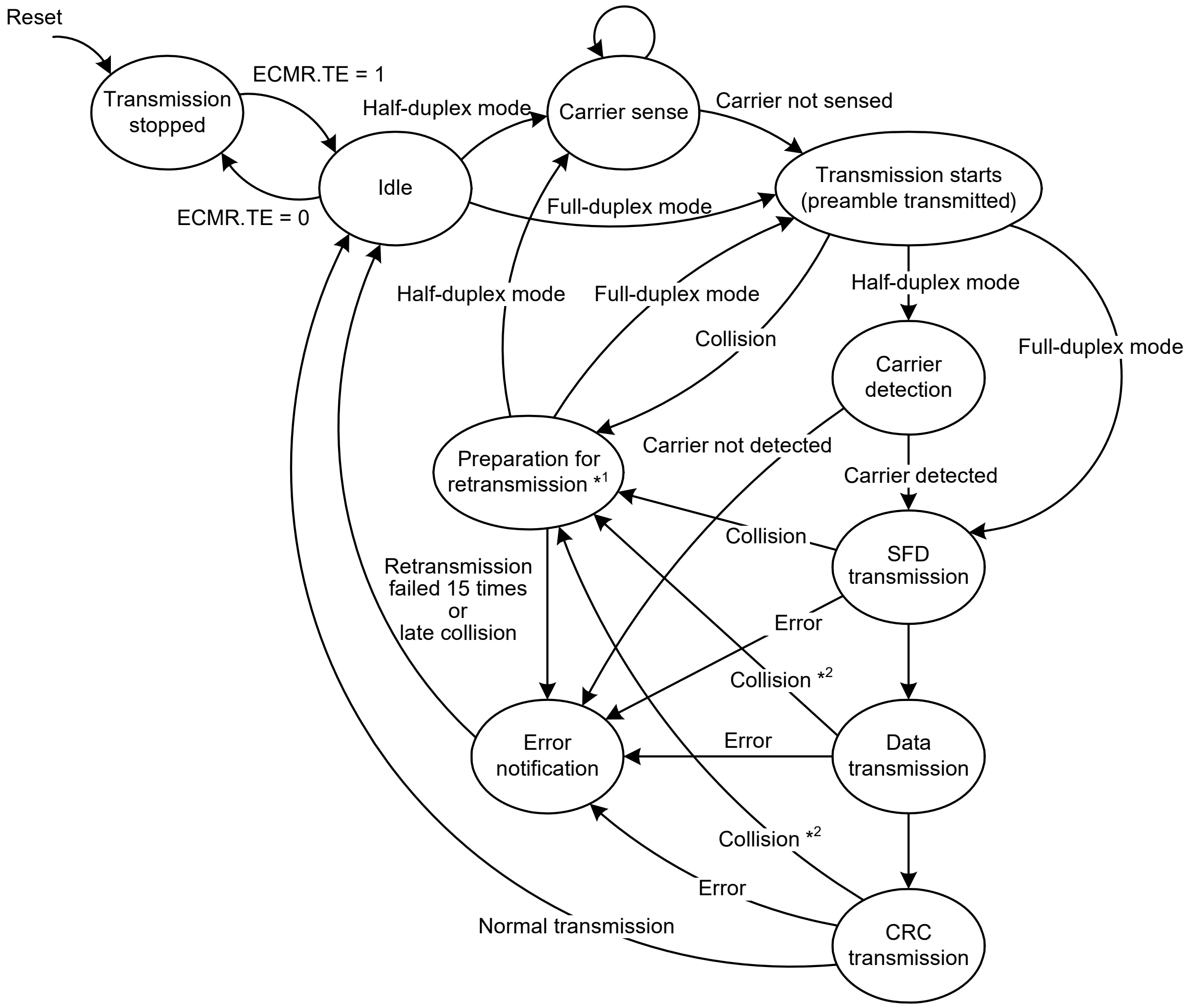
The MAFCR register is a counter that indicates the number of times a frame with the multicast address set was received. When the register value becomes FFFF FFFFh, the counter stops. Writing any value to the MAFCR register clears the counter value to 0.

29.3 Operation

This section provides an overview of the ETHERC operations. The ETHERC supports flow control compliant with IEEE802.3x, and can transmit and receive PAUSE frames. When using the ETHERC, set the clock to ICLK = PCLKA beforehand.

29.3.1 Transmission

The ETHERC transmitter assembles transmit data into a frame and outputs it to the MII or RMII when a transmit request is received from the EDMAC. The frame transmitted through the MII or RMII is transmitted on the line by the PHY-LSI. Figure 29.4 shows the state transitions of the ETHERC transmitter.



Note 1. Preparation for retransmission after a collision is detected includes transmitting a jam signal and adjusting transmission time intervals using the backoff algorithm.

Note 2. Preparation for retransmission is performed only when a collision is detected within 512 bit times after preamble transmission starts. When a collision is detected after 512 bit times, only a jam signal is transmitted and processing using the backoff algorithm is not performed.

Figure 29.4 ETHERC transmitter state transitions

The ETHERC transmitter state transitions are as follows:

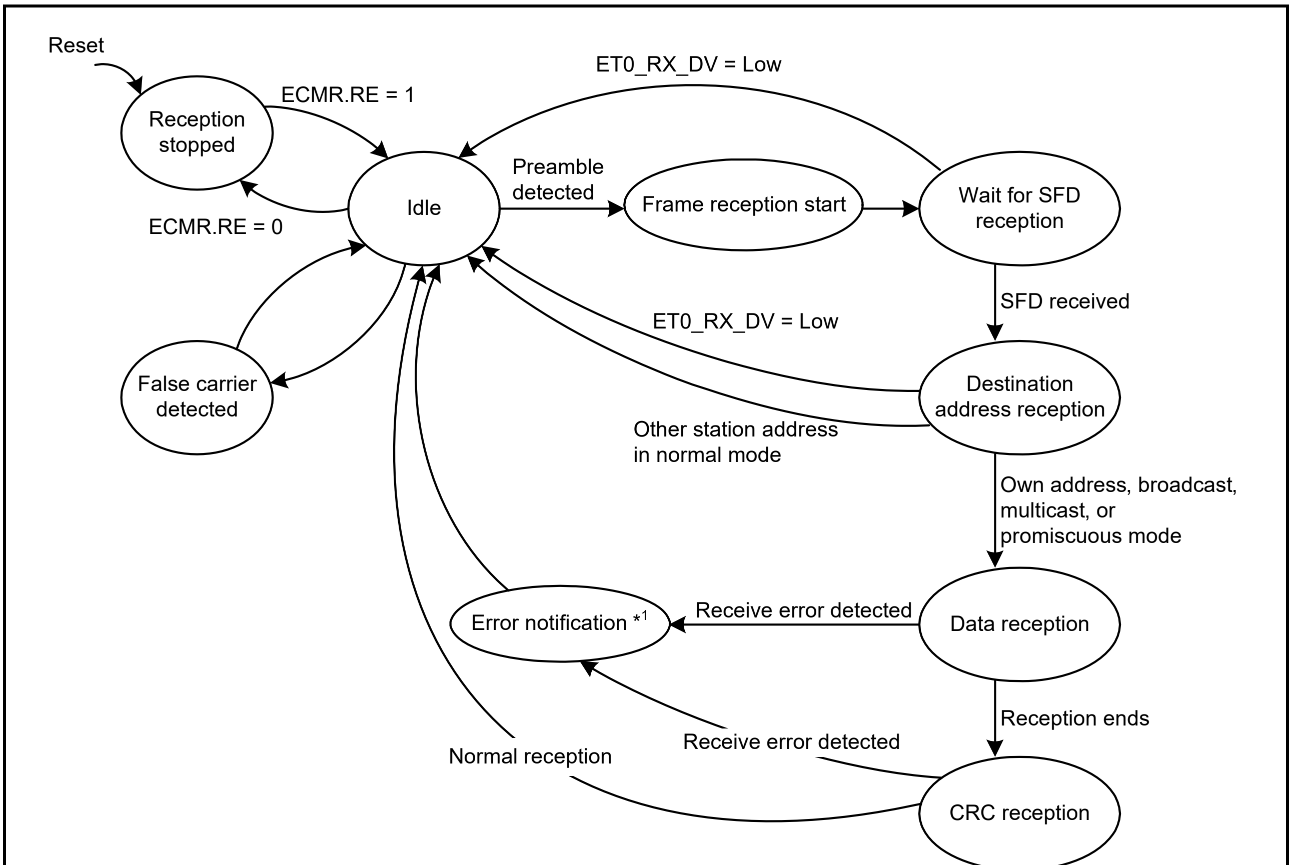
1. When the ECMR.TE bit is set to 1, the ETHERC enters the transmit idle state.
2. When a transmit request is received from the EDMAC, the ETHERC enters the carrier sense state. The ETHERC waits for the interpacket gap and then transmits a preamble to the MII or RMII. When full-duplex mode is selected,

carrier sensing is not required, so the ETHERC transmits a preamble immediately after receiving a transmit request from the EDMAC.

3. The ETHERC transmits the Start Frame Delimiter (SFD), transmit data, and CRC sequentially. When the transmission completes successfully, the ETHERC notifies the EDMAC of successful completion, and the EDMAC sets the EDMAC0.EESR.TC flag to 1. When a late collision or loss of carrier is detected during data transmission, the ETHERC stops the transmission and notifies the EDMAC of the error.
4. After the time specified as the interpacket gap has elapsed, the ETHERC enters the idle state and continues transmission when transmit data remains.

29.3.2 Reception

The ETHERC receiver separates the frame input from the MII or RMII into the preamble, SFD, receive data, and CRC, and transmits only the receive data (destination address, source address, type/length, data/LLC). Figure 29.5 shows the state transitions of the ETHERC receiver.



Note 1. Data in error frames is also transferred to the receive buffer.

Figure 29.5 ETHERC receiver state transitions

The ETHERC receiver state transitions are as follows:

1. When the ECMR.RE bit is set to 1, the ETHERC enters the receive idle state.
2. When the SFD following the preamble of the receive packet is detected, the ETHERC starts reception. If the received SFD is invalid, the ETHERC discards the frame.
3. In normal mode, the ETHERC starts data reception when the destination address of the receive frame is the address of the MCU or the receive frame is a broadcast or multicast frame. In promiscuous mode, the ETHERC starts data reception regardless of the receive frame type.
4. After receiving data from the MII or RMII, the ETHERC performs a CRC check. The ETHERC notifies the EDMAC of the CRC check result. After the received data is transferred to the receive buffer, the CRC check result

is written back to the receive descriptor as status. The result is also reflected in the EDMAC0.EESR.CERF flag.

5. When the ECMR.RE bit is 1 after one frame is received, the ETHERC prepares to receive the next frame.

29.3.3 Frame Timing

29.3.3.1 MII frame timing

Figure 29.6 to Figure 29.11 show the MII frame timing.

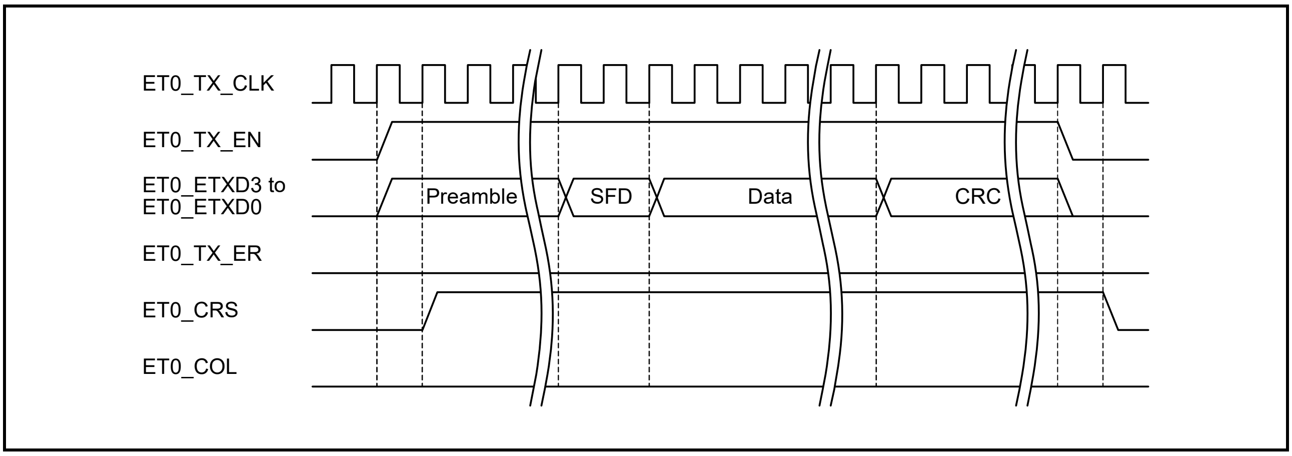


Figure 29.6 MII frame transmit timing during normal transmission

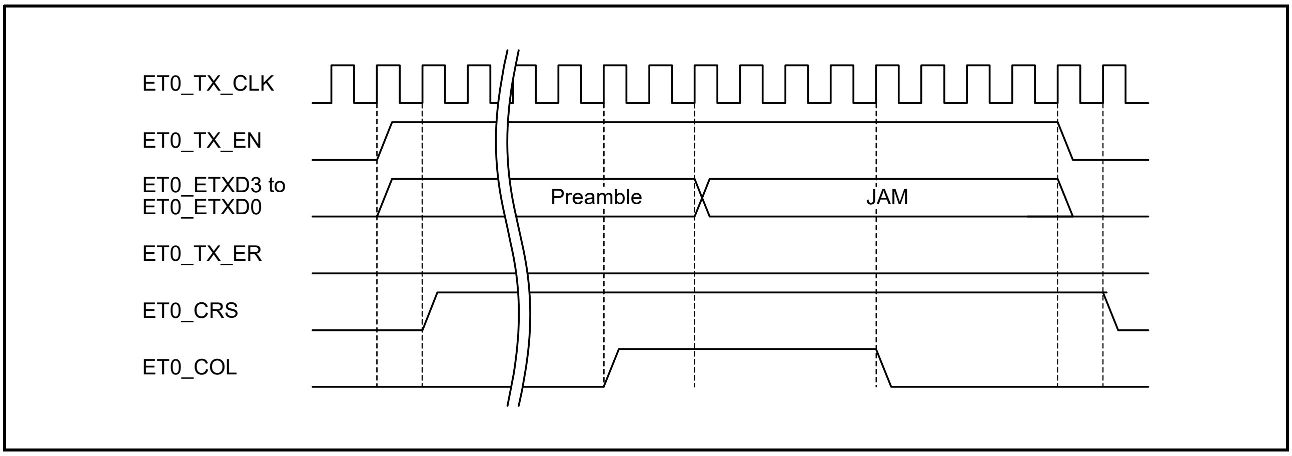


Figure 29.7 MII frame transmit timing when a collision occurs

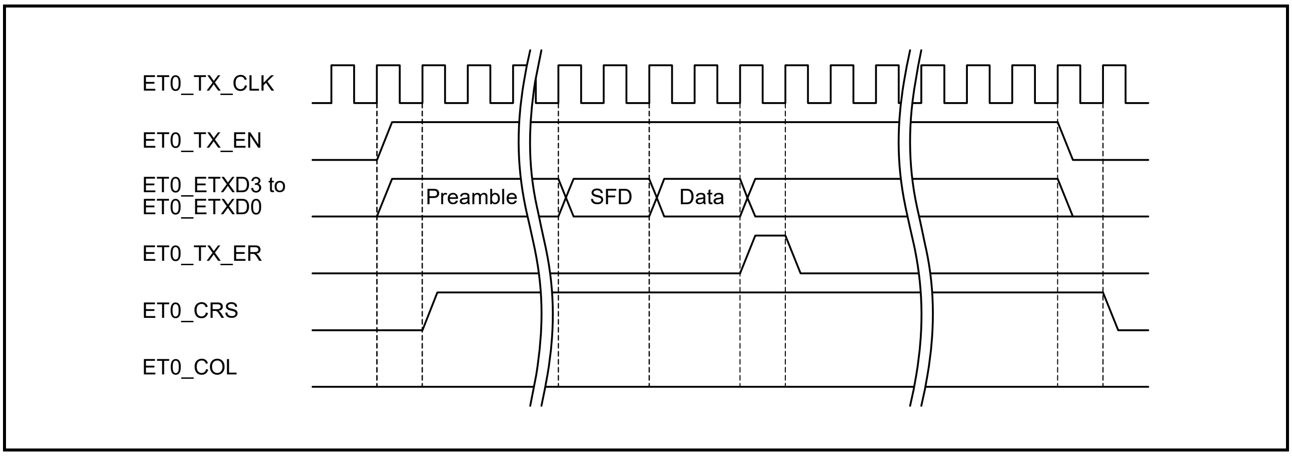


Figure 29.8 MII frame transmit timing when a transmit error occurs

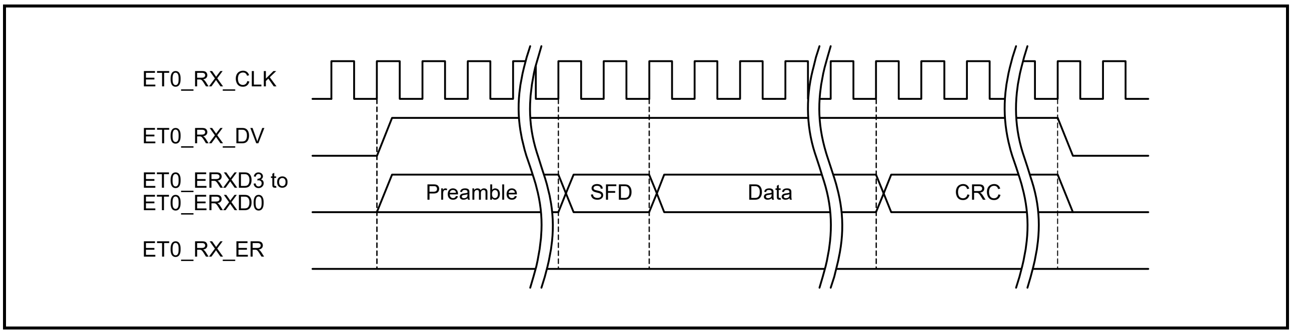


Figure 29.9 MII frame receive timing during normal reception

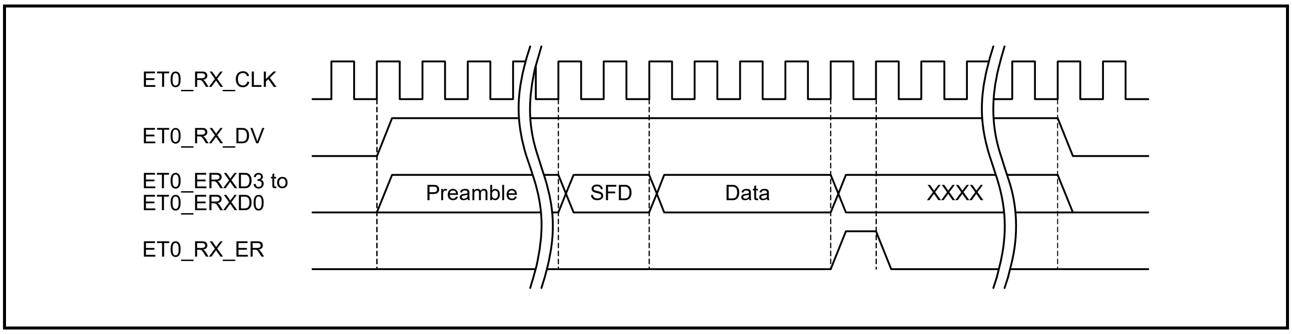


Figure 29.10 MII frame receive timing for receive error notification

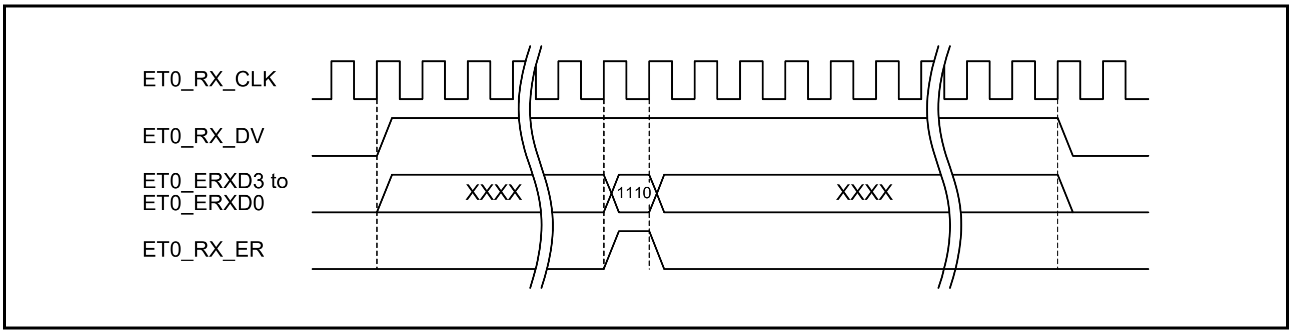


Figure 29.11 MII frame receive timing for false carrier notification

29.3.3.2 RMII frame timing

Figure 29.12 to Figure 29.14 show the RMII frame timing.

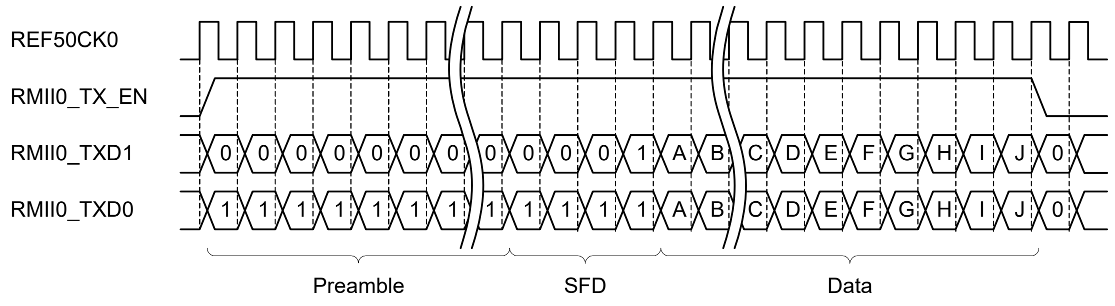


Figure 29.12 RMII frame transmit timing during normal transmission

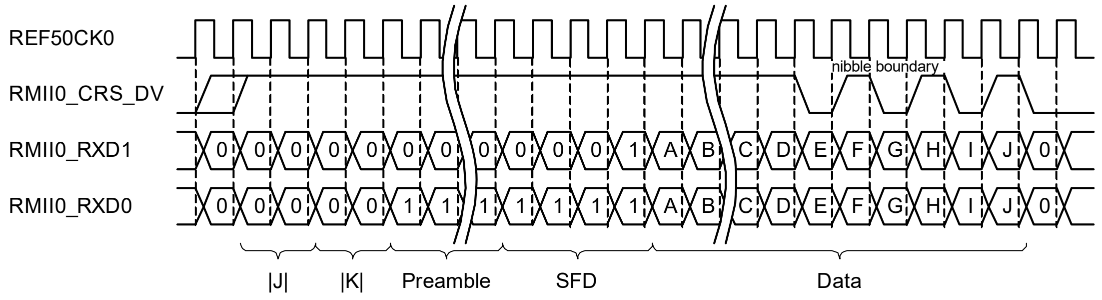


Figure 29.13 RMII frame receive timing during normal reception

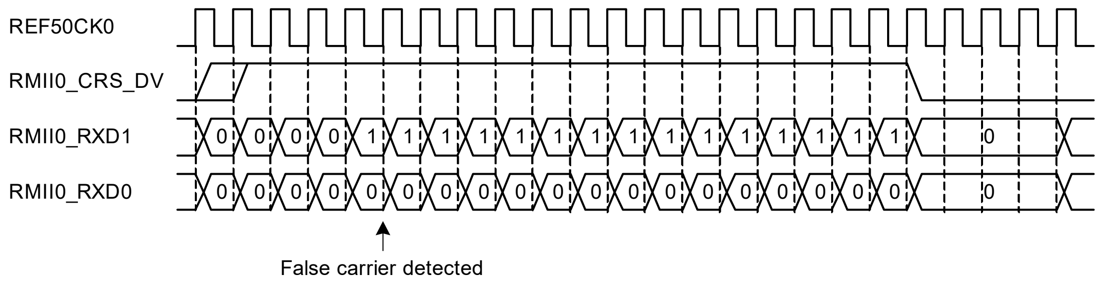


Figure 29.14 RMII frame receive timing when a false carrier is detected

29.3.4 Accessing the MII and RMII Registers

Use the PIR register to access the MII and RMII registers in the PHY-LSI. Serial data in the MII and RMII management frame format is transmitted and received through the ET0_MDC and ET0_MDIO pins controlled by software.

29.3.4.1 MII and RMII management frame format

Table 29.3 lists the MII and RMII management frame formats.

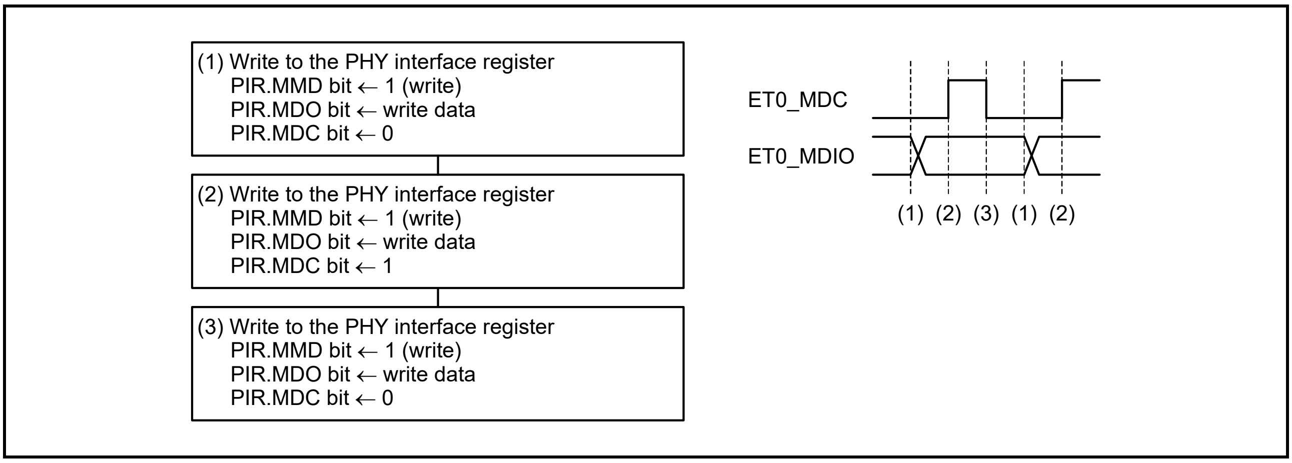
Table 29.3 MII and RMII management frame formats

Access type	MII and RMII management frame								
	Parameter	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
	Number of bits	32	2	2	5	5	2	16	1
Read		1...1	01	10	00001	RRRRR	Z0	DDDDDDDDDDDDDDDD	Z
Write		1...1	01	01	00001	RRRRR	10	DDDDDDDDDDDDDDDD	Z

Note: PRE (preamble): Send 32 consecutive 1s.
ST (start of frame): Send 01b.
OP (operation code): Send 10b for read or 01b for write.
PHYAD (PHY address): Up to 32 PHY-LSIs can be connected to one MAC. PHY-LSIs are selected with these 5 bits. When the PHY-LSI address is 1, send 00001b.
REGAD (register address): One register is selected from up to 32 registers in the PHY-LSI. When the register address is 1, send 00001b.
TA (turnaround): Use 2-bit turnaround time to avoid contention between the register address and data during a read operation. Send 10b during a write operation. Release the bus for 1 bit during a read operation (Z is output).
(This is indicated as Z0 because 0 is output from the PHY-LSI on the next clock cycle.)
DATA (data): 16-bit data. Sequentially send or receive starting from the MSB.
IDLE (IDLE condition): Wait time before inputting the next MII or RMII management format. Release the bus during a write operation (Z is output). No control is required, because a bus was already released during a read operation.

29.3.4.2 MII and RMII register access procedure

Access to the MII and RMII registers includes writing data in 1-bit units, reading data in 1-bit units, and releasing the bus. Figure 29.15 to Figure 29.18 show examples of the MII and RMII register access timing. The access timing differs with the PHY-LSI type.



(1) Write to the PHY interface register
PIR.MMD bit ← 1 (write)
PIR.MDO bit ← write data
PIR.MDC bit ← 0

(2) Write to the PHY interface register
PIR.MMD bit ← 1 (write)
PIR.MDO bit ← write data
PIR.MDC bit ← 1

(3) Write to the PHY interface register
PIR.MMD bit ← 1 (write)
PIR.MDO bit ← write data
PIR.MDC bit ← 0

Figure 29.15 1-bit data write flow

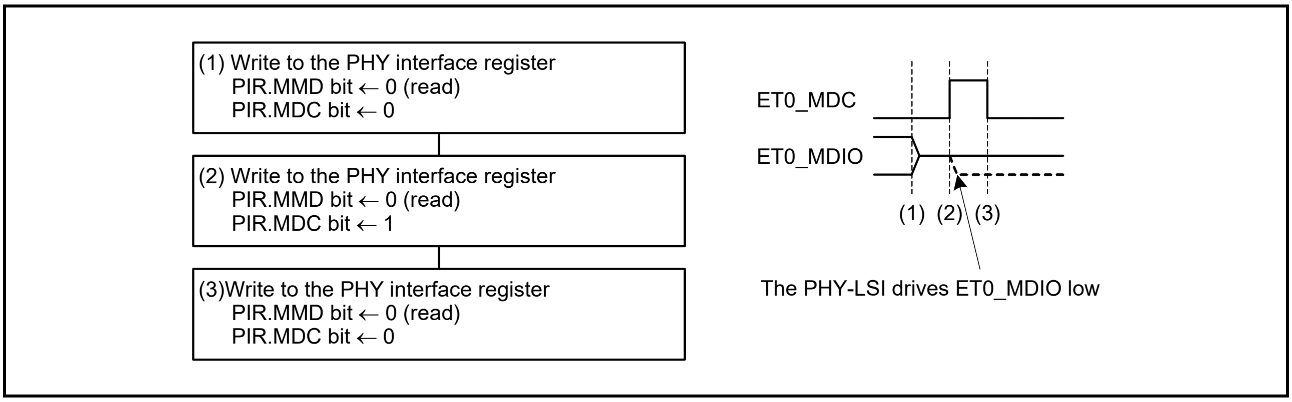


Figure 29.16 Bus release flow, with TA in read operation in Table 29.3

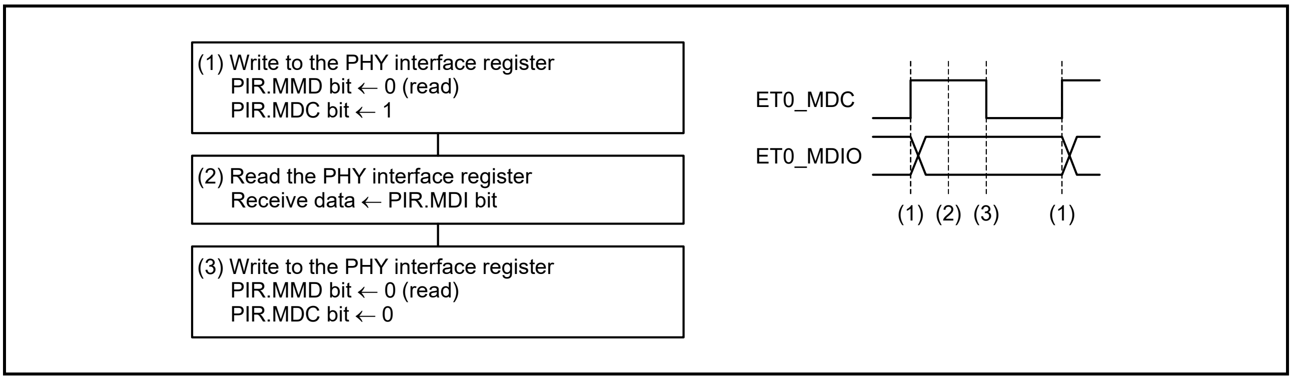


Figure 29.17 1-bit data read flow

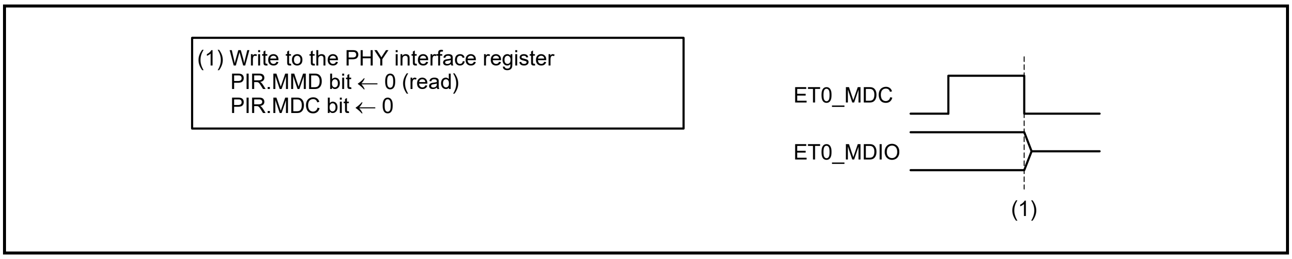


Figure 29.18 Bus release flow, with IDLE in write operation in Table 29.3

29.3.5 Magic Packet Detection

The ETHERC supports Wake-on-LAN (WOL). WOL is a function to detect a Magic Packet transmitted from a host device or other device and wake the MCU from a low power mode such as Sleep. When the ETHERC detects a Magic Packet, it outputs high on the ET0_WOL pin. Write 1 to the EDMAC0.EDMR.SWR bit to drive the ET0_WOL pin low.

Because a Magic Packet is transmitted in broadcast mode, it is received regardless of the destination MAC address selected in the format. The ETHERC outputs high on the ET0_WOL pin only when the destination MAC address matches its own MAC address. See the technical documentation provided by Advanced Micro Devices, Inc., for details on the Magic Packet.

To use WOL in the MCU, use the procedure in the following example:

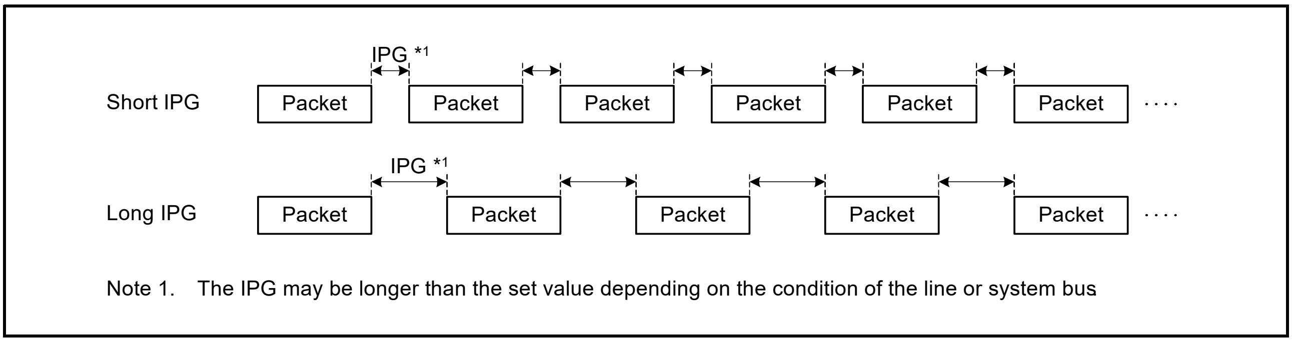
1. Configure the ICU to disable ETHER_EINT0 interrupt requests.
2. Set the ECMR.MPDE bit to 1 to enable Magic Packet detection, and set the ECMR.RE bit to 1 to enable reception.
3. Set the ECSIPR.MPDIP bit to 1 to enable notification of Magic Packet detection interrupts.
4. Set the EDMAC0.EESIPR.ECIIP bit to 1 to enable ETHERC status register source interrupts.
5. Configure the ICU to enable ETHER_EINT0 interrupt requests.
6. Change the CPU operating mode to Sleep mode or place unused peripherals in the module-stop state, as required.
7. When a Magic Packet is detected, an interrupt request is sent to the CPU. High is output on the ET0_WOL pin to notify peripheral devices that the Magic Packet was detected.

29.3.5.1 Constraints on Magic Packet detection

The ETHERC receives packets, including broadcast packets, even when waiting to receive a Magic Packet. This means that receive data might already be stored in the receive FIFO of the EDMAC when a Magic Packet is detected. Also, flags in the ECSR and EDMAC0.EESR registers might have changed. When returning to normal operation after detecting a Magic Packet, set the EDMAC0.EDMR.SWR bit to 1 to reset the ETHERC and EDMAC.

29.3.6 Adjusting Transmission Efficiency by Changing the IPG

The IPG is a non-transmit period between transmit frames. The ETHERC can change the value of the IPG to increase or decrease transmission efficiency based on the value set in the IPGR register. Typical values are specified in the IEEE802.3 standard. When changing the setting, confirm that all devices in the same network operate normally.



Note 1. The IPG may be longer than the set value depending on the condition of the line or system bus

Figure 29.19 Differences in transmission efficiency based on changes in the IPG

29.3.7 Flow Control

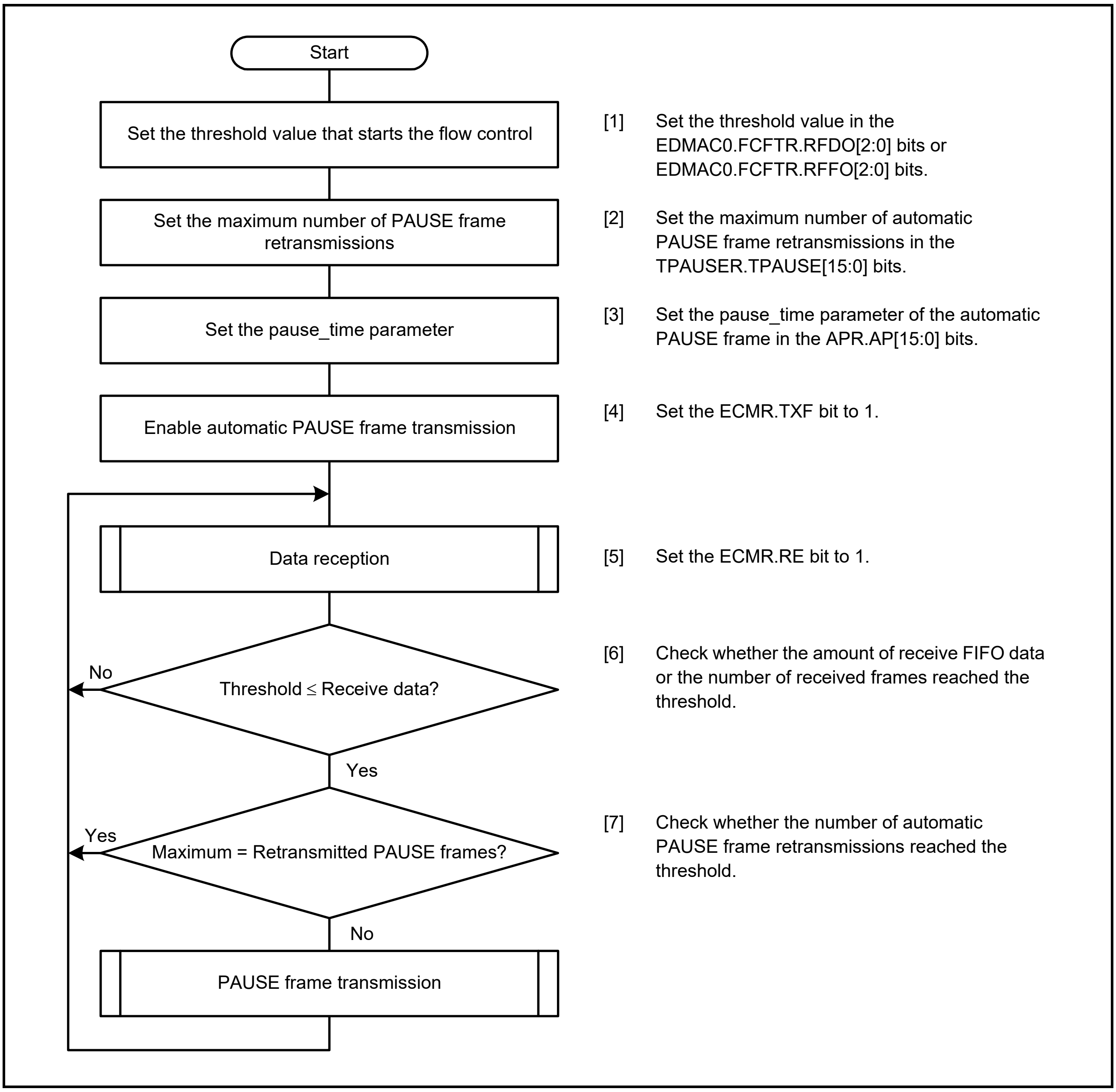
The ETHERC can perform flow control compliant with IEEE802.3x in full-duplex mode, and the receiver and transmitter can be set independently. PAUSE frames can be transmitted automatically or manually.

29.3.7.1 Automatic PAUSE frame transmission

When the ECMR.TXF bit is set to 1, automatic PAUSE frame transmission is enabled. A PAUSE frame is automatically transmitted by a PAUSE frame transmit request from the EDMAC. The APR.AP[15:0] bit value is used for the pause_time parameter of the PAUSE frame.

When a PAUSE frame is transmitted, if the EDMAC is still requesting PAUSE frame transmission after the PAUSE time elapses, a PAUSE frame is transmitted again. The maximum number of PAUSE frame retransmissions can be set in the TPAUSER.TPAUSE[15:0] bits. If the maximum number of retransmissions is reached, subsequent PAUSE frames are not transmitted.

Figure 29.20 shows the procedure for setting up automatic PAUSE frame transmission.



[1] Set the threshold value in the EDMAC0.FCFTR.RFDO[2:0] bits or EDMAC0.FCFTR.RFFO[2:0] bits.

[2] Set the maximum number of automatic PAUSE frame retransmissions in the TPAUSER.TPAUSE[15:0] bits.

[3] Set the pause_time parameter of the automatic PAUSE frame in the APR.AP[15:0] bits.

[4] Set the ECMR.TXF bit to 1.

[5] Set the ECMR.RE bit to 1.

[6] Check whether the amount of receive FIFO data or the number of received frames reached the threshold.

[7] Check whether the number of automatic PAUSE frame retransmissions reached the threshold.

Figure 29.20 Example procedure for setting up automatic PAUSE frame transmission

29.3.7.2 Manual PAUSE frame transmission

A PAUSE frame can be manually transmitted at any time. When the software writes the pause_time parameter of the PAUSE frame to the MPR.MP[15:0] bits, the ETHERC transmits a PAUSE frame once. To transmit a PAUSE frame more than once, write to the MPR.MP[15:0] bits for each transmission.

29.3.7.3 PAUSE frame reception

When the ECMR.RXF bit is set to 1, PAUSE frame detection is enabled. After a PAUSE frame is received, the ETHERC completes transmission of the current frame and waits for the PAUSE time of the received PAUSE frame to elapse before it can transmit the next frame. The ETHERC also increments the RFCF.RPAUSE[7:0] bit value.

However, while waiting for the PAUSE time to elapse, if a PAUSE frame that contains a pause_time parameter of 0 is received and the ECMR.ZPF bit is 1, the ETHERC becomes ready to transmit immediately.

29.4 Interrupts

When a flag in the ECSR register sets to 1 and the associated bit in the ECSIPR register is 1, the ETHERC notifies the EDMAC of the interrupt source status. After receiving the notification, the EDMAC sets the EDMAC0.EESR.ECI flag to 1. When the EDMAC0.EESIPR.ECIIP bit is 1, the EDMAC sends an ETHER_EINT0 interrupt request to the CPU. For details, see section 31, Ethernet DMA Controller (EDMAC).

29.5 Usage Notes

29.5.1 Preventing the LCHNG Flag from Erroneously Setting to 1

The ECSR.LCHNG flag might set to 1 even when the input level of the ET0_LINKSTA pin remains the same. In this case, high is input to the ET0_LINKSTA pin when setting the PFS.PmnPFS register to assign the ET0_LINKSTA signal to a port or when releasing the ETHERC and EDMAC software reset using the EDMAC0.EDMR.SWR bit. The ECSR.LCHNG flag sets to 1 because the ET0_LINKSTA signal in the ETHERC is fixed low regardless of the input level to the external pin if the MPC does not assign the ET0_LINKSTA signal or during an ETHERC and EDMAC software reset.

To avoid erroneously generating a link signal change interrupt, clear the ECSR.LCHNG flag, and then set the ECSIPR.LCHNGIP bit to 1.

29.5.2 Input to RMII0_RX_ER Pin while RMII Is Selected

When the width of a reception error signal received from the PHY-LSI is only 1 cycle of the REF50CK0 clock (50 MHz) while the RMII is selected, the signal is not recognized as an error signal.

29.5.3 Processing when Erroneous Frame Is Detected

If an erroneous frame is detected due to a corrupted frame or noise in the external circuit when the ETHERC and EPTPC are receiving data, subsequent normal frames might not be received properly.

Reset the EDMAC, ETHERC, and EPTPC after an erroneous frame is detected. Then, wait for the required number of cycles before setting communications again.

When set to bypass EPTPC, you do not need to reset the processing. See section 30.2.79, Bypass 1588 Module Register (BYPASS).

(1) Detecting an erroneous frame

An erroneous frame can be detected by reading the INFABT flag in the SYNFP Status Register (SYSR) of EPTPCn. Even when the EPTPCn is not used but only the EDMACn and ETHERCn are used to receive and transmit standard Ethernet frames, read the INFABT flag to detect an erroneous frame (n = 0).

(2) Resetting after detection of an erroneous frame

When the EPTPCn.SYSR.INFABT flag becomes 1, reset EPTPCn, EDMACn, and ETHERCn according to the channel. Then wait for the required number of cycles before setting the registers. Even when the EPTPCn is not used but only the EDMACn and ETHERCn are used to receive and transmit standard Ethernet frames, reset the EPTPCn and the registers. In this case, you do not need to reset PTPEDMAC. The following steps show the resetting procedure where n = 0:

1. Set the EPTPC_CFG.PTRSTR.RESET bit to 1 (reset the EPTPCn through software).
2. Set the EDMACn.EDMR.SWR bit to 1 (reset the EDMACn and ETHERCn through software).
3. Wait for at least 64 cycles of the peripheral module clock (PCLKA). This step is necessary to initialize EDMACn and ETHERCn. Use a software loop or timer to wait for at least 64 PCLKA cycles.
4. Set the EPTPC_CFG.PTRSTR.RESET bit to 0 (release the EPTPCn reset).
5. Reset communications.
6. Set the EDMACn, ETHERCn, PTPEDMAC, and EPTPCn registers to enable communications.

29.5.4 Collision Occurrence in Half-Duplex Mode

Transmission might start and communication might collide within 21 clock cycles (50 MHz) from reception in half-duplex mode.

30. Ethernet PTP Controller (EPTPC)

30.1 Overview

The MCU provides an on-chip Precision Time Protocol (PTP) module for the Ethernet Controller (EPTPC). The module applies the PTP as defined in version 2 of the IEEE 1588-2008 standard to handle timing and synchronization between devices. The EPTPC is composed of a Synchronization Frame Processing unit (SYNFP0) and a Statistical Time Correction Algorithm unit (STCA).

The EPTPC is used in combination with the on-chip Ethernet Controller (ETHERC) and the DMA Controller for the PTP Ethernet Controller (PTPEDMAC). When the EPTPC is not used, you can bypass it by setting the bypass registers in the EPTPC. See section 30.2.79, Bypass 1588 Module Register (BYPASS).

Table 30.1 lists the EPTPC specifications, and Figure 30.1 shows the configuration.

Table 30.1 EPTPC specifications

Parameter	Specifications
Protocol	Compliant with the Precision Time Protocol (PTP) defined in IEEE 1588.
Synchronization Frame Processing unit (SYNFP0)	• Transmission and reception of PTP messages as a master or slave • Support for clock device: - Ordinary clock (OC) • Calculation of meanPathDelay and offsetFromMaster as defined in IEEE 1588 • Capable of generating a master clock • Hardware filtering of received multicast packets with a MAC address • Capable of hardware filtering with the type of PTP message • Support for PTP message frames in layer 4 (IPv4 and UDP) and layer 2 (Ethernet frames) • Can be used as a normal Ethernet port when time synchronization is not in use
Statistical Time Correction Algorithm unit (STCA)	• Frequency of the clock supplied to the Statistical Time Correction Algorithm unit is selectable as 20, 25, 50, or 100 MHz • In slave operation, the synchronized state can be indicated by the offsetFromMaster value staying below a threshold specified in advance or calculated statistically from collected positive and negative gradient values (worst-10 acquisition) • Local clock counter holds corrected time information obtained from a master clock • STCA clock can be used as the clock source for generating pulse signals from pulse output timer m (m = 0 to 5) • Peripheral modules such as GPT can be started or stopped on the edge of pulses synchronized with the master clock in response to interrupt requests by the pulse output timer or the output of event signals to the ELC
Interrupt sources	ETHER_MINT interrupt: • Requested when the state of the individual modules is changed • Requested on rising edges of the pulse signal generated by the pulse output timer. ETHER_IPLS interrupt: • Requested on rising or falling edges of the pulse signal generated by the previously selected pulse output timer group • Can be requested on every edge or only once
Event linking	• Event signal is output to the ELC on a rising or falling edge of the pulse signal generated by the pulse output timer • Event signal can be output on every edge or only once

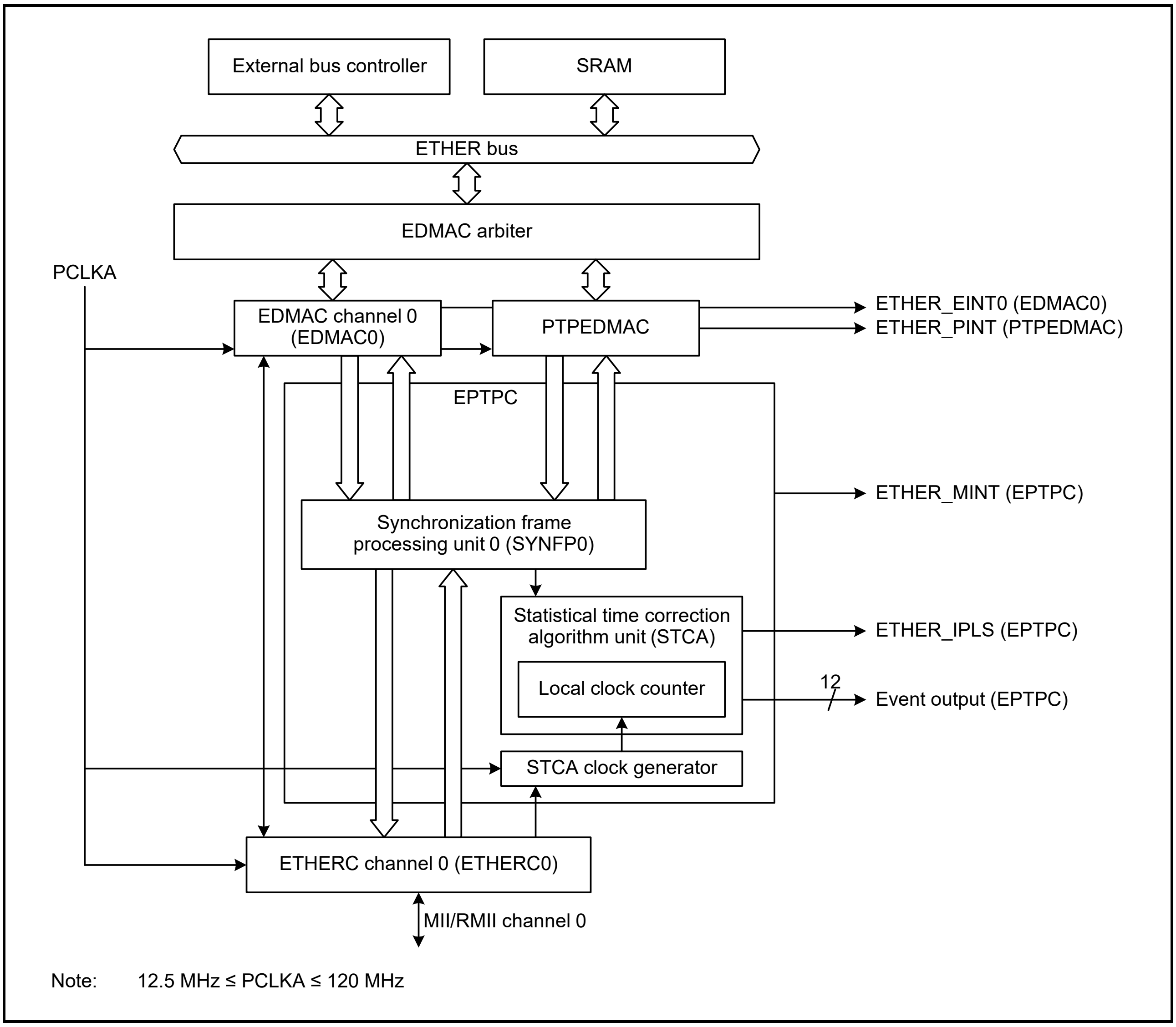


Figure 30.1 EPTPC configuration

In this section, individual channels might not be mentioned in the overall descriptions of modules that have multiple channels. Table 30.2 lists examples of the notation.

Table 30.2 Notation examples

Module name	Channel	Meaning
SYNFP module	One channel	Synchronization processing unit 0 (SYNFP0)
Pulse output timer m	m = 0 to 5	Pulse output timer channels 0 to 5

30.1.1 Combination of Clock Device and Ethernet Port

The EPTPC supports operation as one type of clock device:

- Ordinary clock (OC)

In addition, it supports both end-to-end (E2E) and peer-to-peer (P2P) operation. Table 30.3 lists the available combinations for usage of Ethernet ports 0.

Table 30.3 Combination of clock devices and Ethernet ports

Clock device	Ethernet port 0	
No control by EPTPC	PTP packets are not handled	
Ordinary clock (OC) Only Ethernet port 0 is used for handling PTP packets	Master	End-to-end (E2E)
		Peer-to-peer (P2P)
	Slave	E2E
		P2P

30.1.2 Frame Format of PTP Messages

The frame format of PTP messages can be selected from the four types by setting the FORM0 and FORM1 bits in the SYNFP Frame Format Setting Register (SYFORMR). Figure 30.2 shows the PTP message formats for transmission and reception by the EPTPC.

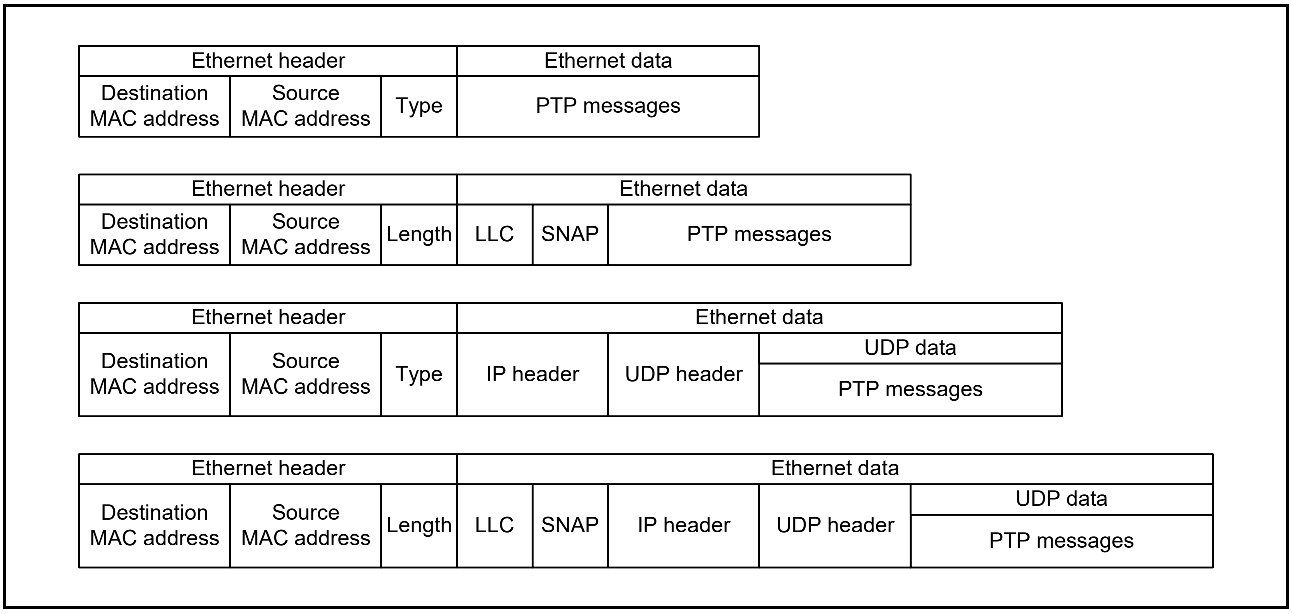


Figure 30.2 Frame format of PTP messages

The EPTPC module is capable of transmitting PTP messages. When it sends a PTP message, multicast addresses as defined in IEEE 1588 are normally specified as the destination MAC address and IP address, depending on the type of PTP message to be sent. In addition, when a PTP message is encapsulated for use with UDP, the port number must also be specified in accordance with the message type, as stipulated in IEEE 1588.

Table 30.4 provides a summary of the information required to specify the Ethernet frame format for PTP messages.

Table 30.4 PTP message types for multicast and information for specifying the Ethernet frame format

PTP message type			IEEE802.3 frame format (SYFORMR.FORM0 bit = 1)		Ethernet II frame format (SYFORMR.FORM0 bit = 0)		UDP port number*1
			MAC address	IP address	MAC address	Ethertype	
PTP-primary	Event messages	Sync	01-00-5E-00-01-81	224.0.1.129	01-1B-19-00-00-00	88F7h	319
		Delay_Req					
PTP-pdelay		Pdelay_Req	01-00-5E-00-00-6B	224.0.0.107	01-80-C2-00-00-0E		
		Pdelay_Resp					
	General messages	Pdelay_Resp_Follow_Up					320
PTP-primary		Announce	01-00-5E-00-01-81	224.0.1.129	01-1B-19-00-00-00		
		Follow_Up					
		Delay_Resp					
		Signaling					
		Management					

Note 1. The port number must be specified only when a PTP message is encapsulated for use with UDP, when the SYFORMR.FORM1 bit = 1.

30.1.3 PTP Message Type and Processing Details

Table 30.5 and Table 30.6 give details on EPTPC processing for receiving and transmitting PTP messages.

Table 30.5 Processing of PTP messages received by the EPTPC

Message type	Message	The EPTPC...
Event	Sync	Calculates the value of offsetFromMaster if twoStepFlag in flagField is FALSE
	Delay_Req	Responds to Delay_Resp
	Pdelay_Req	Responds to Pdelay_Resp
	Pdelay_Resp	Calculates the value of meanPathDelay if twoStepFlag in flagField is FALSE
General	Announce	—
	Follow_Up	Calculates the value of offsetFromMaster if twoStepFlag in flagField of the most recently received Sync message was TRUE and the value of meanPathDelay is fixed
	Delay_Resp	Calculates the value of meanPathDelay
	Pdelay_Resp_Follow_Up	Calculates the value of meanPathDelay if twoStepFlag in flagField of the most recently received Pdelay_Resp message was TRUE
	Management	—
	Signaling	—

Table 30.6 Processing of PTP messages to be transmitted by the EPTPC (1 of 2)

Message type	Message	The EPTPC...
Event	Sync	Transmits sync messages at the fixed interval specified in the SYTLIR.SYNC[7:0] bits
	Delay_Req	Proceeds with transmission with an interval from 0 to twice the interval set in the SYTLIR.DREQ[7:0] bits and determined by a random number
	Pdelay_Req	Transmits Pdelay_Req messages at the fixed interval specified in the SYTLIR.DREQ[7:0] bits
	Pdelay_Resp	Transmits responses to Pdelay_Req

Table 30.6 Processing of PTP messages to be transmitted by the EPTPC (2 of 2)

Message type	Message	The EPTPC...
General	Announce	Transmits Announce messages at the fixed interval specified in the SYTLIR.ANCE[7:0] bits
	Follow_Up	—
	Delay_Resp	Transmits responses to Delay_Req
	Pdelay_Resp_Follow_Up	—
	Management	—
	Signaling	—

30.2 Register Descriptions

30.2.1 ETHER_MINT Interrupt Source Status Register (MIESR)

Address(es): EPTPC.MIESR 4006 5000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SY0	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ST	STCA Status Flag	0: No change in the state of the STCA unit 1: A change in the state of the STCA unit.	R
b1	SY0	SYNFP0 Status Flag	0: No change in the state of the SYNFP0 unit 1: A change in the state of the SYNFP0 unit.	R
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	CYC0	Pulse Output Timer 0 Rising Edge Detection Flag	0: Rising edge not detected on the periodic pulse signal from pulse output timer 0 1: Rising edge detected on the periodic pulse signal from pulse output timer 0.	R/W*1
b17	CYC1	Pulse Output Timer 1 Rising Edge Detection Flag	0: Rising edge not detected on the periodic pulse signal from pulse output timer 1 1: A Rising edge detected on the periodic pulse signal from pulse output timer 1.	R/W*1
b18	CYC2	Pulse Output Timer 2 Rising Edge Detection Flag	0: Rising edge not detected on the periodic pulse signal from pulse output timer 2 1: A Rising edge detected on the periodic pulse signal from pulse output timer 2.	R/W*1
b19	CYC3	Pulse Output Timer 3 Rising Edge Detection Flag	0: Rising edge not detected on the periodic pulse signal from pulse output timer 3 1: A Rising edge detected on the periodic pulse signal from pulse output timer 3.	R/W*1
b20	CYC4	Pulse Output Timer 4 Rising Edge Detection Flag	0: Rising edge not detected on the periodic pulse signal from pulse output timer 4 1: A Rising edge detected on the periodic pulse signal from pulse output timer 4.	R/W*1
b21	CYC5	Pulse Output Timer 5 Rising Edge Detection Flag	0: Rising edge not detected on the periodic pulse signal from pulse output timer 5 1: A Rising edge detected on the periodic pulse signal from pulse output timer 5.	R/W*1

Bit	Symbol	Bit name	Description	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 clears the flag. Writing 0 does not affect the flag value.

The MIESR register indicates changes in the states of the STCA and SYNFP0 units, which act as ETHER_MINT interrupt sources, and enables the detection of rising edges on pulse output timers m (m = 0 to 5). For more the ETHER_MINT interrupt, see section 30.4, Interrupts.

ST flag (STCA Status Flag)

The ST flag indicates changes in the state of the STCA unit.

[Setting condition]

- A change in the state of a flag in the STSR register for which notification is enabled in the STIPR register.

[Clearing conditions]

When any of the following conditions is met:

- The flags in the STSR register are all 0
- The bits in the STIPR register are all 0
- A bit is set to 1 in the STIPR register, but the associated flag in the STSR register is 0.

SY0 flag (SYNFP0 Status Flag)

The SY0 flag indicates changes in the state of the SYNFP0 unit.

[Setting condition]

- A change in the state of a flag in the SYSR register for which notification is enabled in the SYIPR register.

[Clearing conditions]

When any of the following conditions is met:

- The flags in the SYSR register are all 0
- The bits in the SYIPR register are all 0
- A bit is set to 1 in the SYIPR register, but the associated flag in the SYSR register is 0.

CYCm flag (Pulse Output Timer m Rising Edge Detection Flag)

The CYCm flag indicates detection of a rising edge on the periodic pulse signal produced by the associated pulse output timer m (m = 0 to 5).

[Setting condition]

- Detection of a rising edge on the periodic pulse signal produced by a pulse output timer for which notification is enabled in the MITSELR register.

[Clearing condition]

- 1 is written to this flag.
  After the flag is cleared to 0, it is set to 1 again on detection of a rising edge on the periodic pulse signal from the associated pulse output timer.

30.2.2 ETHER_MINT Interrupt Request Enable Register (MIEIPR)

Address(es): EPTPC.MIEIPR 4006 5004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	CYC5	CYC4	CYC3	CYC2	CYC1	CYC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SY0	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ST	STCA Status Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests by the STCA status flag 1: Enable generation of ETHER_MINT interrupt requests by the STCA status flag.	R/W
b1	SY0	SYNFP0 Status Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests by the SYNFP0 status flag 1: Enable generation of ETHER_MINT interrupt requests by the SYNFP0 status flag.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	CYC0	Pulse Output Timer 0 Rising Edge Detection Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 0 1: Enable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 0.	R/W
b17	CYC1	Pulse Output Timer 1 Rising Edge Detection Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 1 1: Enable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 1.	R/W
b18	CYC2	Pulse Output Timer 2 Rising Edge Detection Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 2 1: Enable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 2.	R/W
b19	CYC3	Pulse Output Timer 3 Rising Edge Detection Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 3 1: Enable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 3.	R/W
b20	CYC4	Pulse Output Timer 4 Rising Edge Detection Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 4 1: Enable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 4.	R/W
b21	CYC5	Pulse Output Timer 5 Rising Edge Detection Interrupt Request Enable	0: Disable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 5 1: Enable generation of ETHER_MINT interrupt requests on detection of a rising edge of pulse output timer 5.	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MIEIPR register enables or disables the generation of ETHER_MINT interrupt requests when ETHER_MINT interrupt source conditions are satisfied.

30.2.3 ELC Output/ETHER_IPLS Interrupt Request Permission Register (ELIPPR)

Address(es): EPTPC.ELIPPR 4006 5010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PLSN	—	—	—	—	—	—	—	PLSP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CYCN5	CYCN4	CYCN3	CYCN2	CYCN1	CYCN0	—	—	CYCP5	CYCP4	CYCP3	CYCP2	CYCP1	CYCP0
Value after reset:	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	CYCP0	Pulse Output Timer 0 Rising Edge Detection Event Output Enable	0: Do not output rising edges of pulse output timer 0 to the ELC as event signals 1: Output rising edges of pulse output timer 0 to the ELC as event signals.	R/W
b1	CYCP1	Pulse Output Timer 1 Rising Edge Detection Event Output Enable	0: Do not output rising edges of pulse output timer 1 to the ELC as event signals 1: Output rising edges of pulse output timer 1 to the ELC as event signals.	R/W
b2	CYCP2	Pulse Output Timer 2 Rising Edge Detection Event Output Enable	0: Do not output rising edges of pulse output timer 2 to the ELC as event signals 1: Output rising edges of pulse output timer 2 to the ELC as event signals.	R/W
b3	CYCP3	Pulse Output Timer 3 Rising Edge Detection Event Output Enable	0: Do not output rising edges of pulse output timer 3 to the ELC as event signals 1: Output rising edges of pulse output timer 3 to the ELC as event signals.	R/W
b4	CYCP4	Pulse Output Timer 4 Rising Edge Detection Event Output Enable	0: Do not output rising edges of pulse output timer 4 to the ELC as event signals 1: Output rising edges of pulse output timer 4 to the ELC as event signals.	R/W
b5	CYCP5	Pulse Output Timer 5 Rising Edge Detection Event Output Enable	0: Do not output rising edges of pulse output timer 5 to the ELC as event signals 1: Rising edges of the signal from pulse output timer 5 to the ELC as event signals.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CYCN0	Pulse Output Timer 0 Falling Edge Detection Event Output Enable	0: Do not output falling edges of pulse output timer 0 to the ELC as event signals 1: Output falling edges of pulse output timer 0 to the ELC as event signals.	R/W
b9	CYCN1	Pulse Output Timer 1 Falling Edge Detection Event Output Enable	0: Do not output falling edges of pulse output timer 1 to the ELC as event signals 1: Output falling edges of pulse output timer 1 to the ELC as event signals.	R/W
b10	CYCN2	Pulse Output Timer 2 Falling Edge Detection Event Output Enable	0: Do not output falling edges of pulse output timer 2 to the ELC as event signals 1: Output falling edges of pulse output timer 2 to the ELC as event signals.	R/W
b11	CYCN3	Pulse Output Timer 3 Falling Edge Detection Event Output Enable	0: Do not output falling edges of pulse output timer 3 to the ELC as event signals 1: Output falling edges of pulse output timer 3 to the ELC as event signals.	R/W

Bit	Symbol	Bit name	Description	R/W
b12	CYCN4	Pulse Output Timer 4 Falling Edge Detection Event Output Enable	0: Do not output falling edges of pulse output timer 4 to the ELC as event signals 1: Output falling edges of pulse output timer 4 to the ELC as event signals.	R/W
b13	CYCN5	Pulse Output Timer 5 Falling Edge Detection Event Output Enable	0: Do not output falling edges of pulse output timer 5 to the ELC as event signals 1: Output falling edges of pulse output timer 5 to the ELC as event signals.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PLSP	Pulse Output Timer Rising Edge Detection ETHER_IPLS Interrupt Request Enable	0: Disable ETHER_IPLS interrupt requests triggered by rising edges of signals from the selected pulse output timer 1: Enable ETHER_IPLS interrupt requests triggered by rising edges of signals from the selected pulse output timer.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	PLSN	Pulse Output Timer Falling Edge Detection ETHER_IPLS Interrupt Request Enable	0: Disable ETHER_IPLS interrupt requests triggered by falling edges of signals from the selected pulse output timer 1: Enable ETHER_IPLS interrupt requests triggered by falling edges of signals from the selected pulse output timer.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ELIPPR register determines whether rising and falling edges of the periodic pulse signals produced by pulse output timers m are output as event signals to the ELC. The register also enables or disables ETHER_IPLS interrupts triggered by rising or falling edges of signals from the pulse output timer selected in the IPTSELR register.

Peripheral modules such as the GPT can be controlled with the clock synchronized by the PTP by using the ELC linking function to set a periodic pulse generated by pulse output timer m as a trigger for operations of the peripheral module.

The ELIPACR register can be used to set up the one-time-only output of event signals to the ELC or of ETHER_IPLS interrupt requests. For more on the ETHER_IPLS interrupt, see section 30.4, Interrupts.

30.2.4 ELC Output/ETHER_IPLS Interrupt Permission Automatic Clearing Register (ELIPACR)

Address(es): EPTPC.ELIPACR 4006 5014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	PLSN	—	—	—	—	—	—	—	PLSP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CYCN5	CYCN4	CYCN3	CYCN2	CYCN1	CYCN0	—	—	CYCP5	CYCP4	CYCP3	CYCP2	CYCP1	CYCP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CYCP0	ELIPPR.CYCP0 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of rising edges of pulse output timer 0 1: Enable automatic clearing of enable bit for output of rising edges of pulse output timer 0.	R/W
b1	CYCP1	ELIPPR.CYCP1 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of rising edges of pulse output timer 1 1: Enable automatic clearing of enable bit for output of rising edges of pulse output timer 1.	R/W
b2	CYCP2	ELIPPR.CYCP2 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of rising edges of pulse output timer 2 1: Enable automatic clearing of enable bit for output of rising edges of pulse output timer 2.	R/W

Bit	Symbol	Bit name	Description	R/W
b3	CYCP3	ELIPPR.CYCP3 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of rising edges of pulse output timer 3 1: Enable automatic clearing of enable bit for output of rising edges of pulse output timer 3.	R/W
b4	CYCP4	ELIPPR.CYCP4 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of rising edges of pulse output timer 4 1: Enable automatic clearing of enable bit for output of rising edges of pulse output timer 4.	R/W
b5	CYCP5	ELIPPR.CYCP5 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of rising edges of pulse output timer 5 1: Enable automatic clearing of enable bit for output of rising edges of pulse output timer 5.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CYCN0	ELIPPR.CYCN0 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of falling edges of pulse output timer 0 1: Enable automatic clearing of enable bit for output of falling edges of pulse output timer 0.	R/W
b9	CYCN1	ELIPPR.CYCN1 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of falling edges of pulse output timer 1 1: Enable automatic clearing of enable bit for output of falling edges of pulse output timer 1.	R/W
b10	CYCN2	ELIPPR.CYCN2 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of falling edges of pulse output timer 2 1: Enable automatic clearing of enable bit for output of falling edges of pulse output timer 2.	R/W
b11	CYCN3	ELIPPR.CYCN3 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of falling edges of pulse output timer 3 1: Enable automatic clearing of enable bit for output of falling edges of pulse output timer 3.	R/W
b12	CYCN4	ELIPPR.CYCN4 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of falling edges of pulse output timer 4 1: Enable automatic clearing of enable bit for output of falling edges of pulse output timer 4.	R/W
b13	CYCN5	ELIPPR.CYCN5 Bit Automatic Clearing	0: Disable automatic clearing of enable bit for output of falling edges of pulse output timer 5 1: Enable automatic clearing of enable bit for output of falling edges of pulse output timer 5.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	PLSP	ELIPPR.PLSP Bit Automatic Clearing	0: Disable automatic clearing of enable bit for ETHER_IPLS interrupt requests on detection of rising edges of the pulse output timer 1: Enable automatic clearing of enable bit for ETHER_IPLS interrupt requests on detection of rising edges of the pulse output timer.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	PLSN	ELIPPR.PLSN Bit Automatic Clearing	0: Disable automatic clearing of enable bit for ETHER_IPLS interrupt requests on detection of falling edges of the pulse output timer 1: Enable automatic clearing of enable bit for ETHER_IPLS interrupt requests on detection of falling edges of the pulse output timer.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ELIPACR register enables one-time output of each event to the ELC or each ETHER_IPLS interrupt request triggered by detecting edges of the periodic pulses of pulse output timer m. Normally, an event is output to the ELC or an ETHER_IPLS interrupt request is generated on each edge of the periodic pulses of pulse output timer m while the associated bit in the ELIPPR register is 1 (enabled). When a bit in the ELIPPR register is 1 while the associated bit in the ELIPACR register is also 1, the bit in the ELIPPR register automatically clears to 0 when the event signal for the ELC or ETHER_IPLS interrupt request is generated. For more on the ETHER_IPLS interrupt, see section 30.4, Interrupts.

30.2.5 STCA Status Register (STSR)

Address(es): EPTPC.STSR 4006 5040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	W10D	SYNTOUT	—	SYNCOUT	SYNC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SYNC	Synchronized State Detection Flag	0: Synchronization not detected 1: Synchronization detected.	R/W*1
b1	SYNCOUT	Synchronization Loss Detection Flag	0: Loss of synchronization not detected 1: Loss of synchronization detected.	R/W*1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	SYNTOUT	Sync Message Reception Timeout Detection Flag	0: Sync message reception timeout not detected 1: Sync message reception timeout detected.	R/W*1
b4	W10D	Worst 10 Acquisition Completion Flag	0: Ten worst values not acquired yet 1: Ten worst values acquired.	R/W*1
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When the SYNSTARTR.STR bit is 0, the value of the associated flag stays the same.

Note 1. Writing 1 clears the flag. Writing 0 does not affect the flag value.

The STSR register indicates the state of the STCA module.

SYNC flag (Synchronized State Detection Flag)

The SYNC flag indicates that synchronization has occurred more than the number of times specified in the STMR.SYTH[3:0] bits in succession when the STMR.ALEN0 bit is 1. When the STMR.ALEN0 bit is 0, the SYNC flag is not set to 1 even if synchronization has occurred more than the specified number of times in succession.

SYNCOUT flag (Synchronization Loss Detection Flag)

The SYNCOUT flag indicates that loss of synchronization has occurred more than the number of times specified in the STMR.DVTH[3:0] bits in succession when the STMR.ALEN0 bit is 1. Because the time is not synchronized immediately after time synchronization is started (when the SYNSTARTR.STR bit is set to 1), SYNCOUT is set to 1 regardless of the STMR.ALEN0 bit setting. When using the SYNTOUT flag, set the SYNTOUT flag to 0 immediately after starting time synchronization.

When the STMR.ALEN0 bit is 0, the SYNCOUT flag is not set to 1 even if loss of synchronization occurs more than the specified number of times in succession after time synchronization starts and the SYNTOUT flag is immediately set to 0.

SYNTOUT flag (Sync Message Reception Timeout Detection Flag)

The SYNTOUT flag indicates that a Sync message was not received during the period specified in the SYNTOR register when the STMR.ALEN1 bit is 1. The SYNTOUT flag is set to 1 immediately after time synchronization is started (when the SYNSTARTR.STR bit is set to 1) when no Sync message is received after the EPTPC starts. When using the SYNTOUT flag, set the SYNTOUT flag to 0 immediately after starting time synchronization.

W10D flag (Worst 10 Acquisition Completion Flag)

The W10D flag indicates that acquisition of the worst 10 values is complete.

30.2.6 STCA Status Notification Enable Register (STIPR)

Address(es): EPTPC.STIPR 4006 5044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	W10D	SYNTOUT	—	SYNCOUT	SYNC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SYNC	SYNC Status Notification Enable	0: Disable notification of the STSR.SYNC state 1: Enable notification of the STSR.SYNC state.	R/W
b1	SYNCOUT	SYNCOUT Status Notification Enable	0: Disable notification of the STSR.SYNCOUT state 1: Enable notification of the STSR.SYNCOUT state.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	SYNTOUT	SYNTOUT Status Notification Enable	0: Disable notification of the STSR.SYNTOUT state 1: Enable notification of the STSR.SYNTOUT state.	R/W
b4	W10D	W10D Status Notification Enable	0: Disable notification of the STSR.W10D state 1: Enable notification of the STSR.W10D state.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The STIPR register specifies whether the MIESR.ST flag does or does not reflect changes in the state of the STCA module.

30.2.7 STCA Clock Frequency Setting Register (STCFR)

Address(es): EPTPC.STCFR 4006 5050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	STCF[1:0]	STCA Clock Frequency	b1 b0 0 0: 20 MHz 0 1: 25 MHz 1 0: 50 MHz 1 1: 100 MHz.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings during operations.

The STCFR register specifies the frequency of the clock source for the STCA module (STCA clock). The setting in this register must be set to the same frequency as that selected in the STCSELR register.

STCF[1:0] bits (STCA Clock Frequency)

The STCF[1:0] bits select the frequency of the STCA clock. To enable synchronous control in compliance with IEEE 1588, the STCA clock frequency must be specified as 20, 25, 50, or 100 MHz. Operation is not guaranteed if the frequency selected in these bits differs from the clock frequency actually input to the STCA module.

30.2.8 STCA Operating Mode Register (STMR)

Address(es): EPTPC.STMR 4006 5054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	ALEN1	ALEN0	—	—	—	—	DVTH[3:0]				SYTH[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	W10S	—	CMOD	—	—	—	—	—	WINT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	WINT[7:0]	Worst 10 Acquisition Time	00h: Do not acquire the worst 10 values 01h: Sync message reception: 1 time : FFh: Sync message reception: 255 times.	R/W
b12 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	CMOD	Time Synchronization Correction Mode	0: Mode 1 1: Mode 2.	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	W10S	Worst 10 Acquisition Control Select	0: Start measurement by hardware and use the value acquired in the PW10VR or MW10R register as the filtering limit 1: Start measurement using the GETW10R.GW10 bit and use the value set in the PLIMITR or MLIMITR register as the filtering limit.	R/W
b19 to b16	SYTH[3:0]	Synchronized State Detection Threshold Setting	0h: None *1 1h: 1 time : Fh: 15 times.	R/W
b23 to b20	DVTH[3:0]	Synchronization Loss Detection Threshold Setting	0h: None *2 1h: 1 time : Fh: 15 times.	R/W
b27 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	ALEN0	Alarm Detection Enable 0	0: Disable STSR.SYNC or SYNCOUT flag from setting to 1 on detection of synchronization or loss of synchronization 1: Enable STSR.SYNC or SYNCOUT flag to set to 1 on detection of synchronization or loss of synchronization.	R/W
b29	ALEN1	Alarm Detection Enable 1	0: Disable STSR.SYNTOUT flag from setting to 1 on detection of the Sync message reception timeout interrupt 1: Enable STSR.SYNTOUT flag to set to 1 on detection of the Sync message reception timeout interrupt.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The STSR.SYNC flag is not set to 1 regardless of the ALEN0 bit setting.
Note 2. The STSR.SYNCOUT flag is not set to 1 regardless of the ALEN0 bit setting.

The STMR register specifies the operating mode of the STCA module.

WINT[7:0] bits (Worst 10 Acquisition Time)

The WINT[7:0] bits set the time for acquiring the worst 10 gradients (the number of times Sync messages are received). Renesas recommends setting the number of Sync message receptions to 32 or more in most cases.

CMOD bit (Time Synchronization Correction Mode)

Mode 1 or mode 2 can be selected in the CMOD bit to correct the local time information when the EPTPC operates as a slave clock. Select the appropriate mode for your system configuration. Table 30.7 provides a summary of the two correction modes.

Table 30.7 Correction mode features

Correction mode	Function	Features	Notes
Mode 1	Mode for correcting the counter every Sync message reception by using the current offsetFromMaster. Operation is in mode 1 after the start of correction, then shifts to the specified mode.	The time information of the master clock is set as the local time information at a specific time.	Synchronization is not guaranteed if calculating offsetFromMaster is not possible, for example if packets are temporarily being discarded because of a failure of communications.
Mode 2	In mode 2, the gradient value calculated from offsetFromMaster (worst-10 control) is retained and used in correcting the local time information so that it approximates the time information of the master clock.	Even when calculating offsetFromMaster is not possible, a certain level of synchronization can be guaranteed in this mode, because the counter is still corrected from the gradient information.	Establishing synchronization takes longer.

W10S bit (Worst 10 Acquisition Control Select)

The W10S bit selects the value used for measuring and filtering the worst 10 gradients. When this bit is set to 0, the values acquired in the PW10VRU, PW10VRM, and PW10VRL registers and the MW10RU, MW10RM, and MW10RL registers are used as the limit for the filter. When the bit is set to 1, the values set in the PLIMITRU, PLIMITRM, and PLIMITRL registers and the MLIMITRU, MLIMITRM, and MLIMITRL registers are used as the limit for the filter.

SYTH[3:0] bits (Synchronized State Detection Threshold Setting)

The SYTH[3:0] bits specify the number of consecutive times that a value should fall within the thresholds set in registers SYNTDBRU and SYNTDBRL, to be considered as a synchronized state. When the ALEN0 bit is 1, the STSR.SYNCOUT flag becomes 1.

DVTH[3:0] bits (Synchronization Loss Detection Threshold Setting)

The DVTH[3:0] bits specify a value for the number of consecutive times the offsetFromMaster value must exceed the specified thresholds for the STCA module to detect loss of synchronization. The thresholds are specified in the SYNTDARU and SYNTDARL registers. When the ALEN0 bit is 1, the STSR.SYNCOUT flag is set to 1 on loss of synchronization detection.

ALEN0 bit (Alarm Detection Enable 0)

When the ALEN0 bit is 1, the STSR.SYNC or SYNCOUT flag is set to 1 on detection of synchronization or loss of synchronization. When this bit is 0, the SYNC or SYNCOUT flag is not set to 1 even if synchronization or loss of synchronization is detected.

ALEN1 bit (Alarm Detection Enable 1)

When the ALEN1 bit is 1, the STSR.SYNTOUT flag is set to 1 if a Sync message is not received within the time specified in the SYNTOR register. When this bit is 0, the SYNTOUT flag is not set to 1 even if a reception timeout occurs.

30.2.9 Sync Message Reception Timeout Register (SYNTOR)

Address(es): EPTPC.SYNTOR 4006 5058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	If no Sync message is received within 1024 × n (ns), where n is the SYNTOR setting, a timeout for reception of Sync messages occurs, and the STSR.SYNTOUT flag is set to 1.	R/W

The SYNTOR register specifies the timeout period for reception of Sync messages. The timeout period is 1024 times the SYNTOR setting, in nanoseconds. If no Sync message is received within the period specified in these bits, a timeout is detected. When the SYNTOR register is 0, the STSR.SYNTOUT flag is not set to 1.

30.2.10 ETHER_IPLS Interrupt Request Timer Select Register (IPTSELR)

Address(es): EPTPC.IPTSELR 4006 5060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	IPTSEL5	IPTSEL4	IPTSEL3	IPTSEL2	IPTSEL1	IPTSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IPTSEL0	Pulse Output Timer 0 Select	0: Pulse output timer 0 not selected as a source for ETHER_IPLS interrupt requests 1: Pulse output timer 0 selected as a source for ETHER_IPLS interrupt requests.	R/W
b1	IPTSEL1	Pulse Output Timer 1 Select	0: Pulse output timer 1 not selected as a source for ETHER_IPLS interrupt requests 1: Pulse output timer 1 selected as a source for ETHER_IPLS interrupt requests.	R/W
b2	IPTSEL2	Pulse Output Timer 2 Select	0: Pulse output timer 2 not selected as a source for ETHER_IPLS interrupt requests 1: Pulse output timer 2 selected as a source for ETHER_IPLS interrupt requests.	R/W
b3	IPTSEL3	Pulse Output Timer 3 Select	0: Pulse output timer 3 not selected as a source for ETHER_IPLS interrupt requests 1: Pulse output timer 3 selected as a source for ETHER_IPLS interrupt requests.	R/W
b4	IPTSEL4	Pulse Output Timer 4 Select	0: Pulse output timer 4 not selected as a source for ETHER_IPLS interrupt requests 1: Pulse output timer 4 selected as a source for ETHER_IPLS interrupt requests.	R/W

Bit	Symbol	Bit name	Description	R/W
b5	IPTSEL5	Pulse Output Timer 5 Select	0: Pulse output timer 5 not selected as a source for ETHER_IPLS interrupt requests 1: Pulse output timer 5 selected as a source for ETHER_IPLS interrupt requests.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The IPTSELR register selects the pulse output timers that generate ETHER_IPLS interrupt requests. Each pulse output timer m (m = 0 to 5) takes the clock signal from the STCA as its clock source and produces pulses with a specified period and duty cycle. An ETHER_IPLS interrupt is requested on rising edges if the ELIPPR.PLSP bit is set to 1 and on falling edges if the PLSN bit in the same register is set to 1. When multiple channels are selected in this register, the interrupt request signal becomes the logical OR of the interrupt requests from the selected channels. For more on the ETHER_IPLS interrupt, see section 30.4, Interrupts.

30.2.11 ETHER_MINT Interrupt Request Timer Select Register (MITSELR)

Address(es): EPTPC.MITSELR 4006 5064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	MINTEN5	MINTEN4	MINTEN3	MINTEN2	MINTEN1	MINTEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MINTEN0	Pulse Output Timer 0 ETHER_MINT Interrupt Output Enable	0: Do not reflect rising edges of pulse output timer 0 on MIESR.CYC0 flag as ETHER_MINT interrupt source 1: Reflect rising edges of pulse output timer 0 on MIESR.CYC0 flag as ETHER_MINT interrupt source.	R/W
b1	MINTEN1	Pulse Output Timer 1 ETHER_MINT Interrupt Output Enable	0: Do not reflect rising edges of pulse output timer 1 on MIESR.CYC1 flag as ETHER_MINT interrupt source 1: Reflect rising edges of pulse output timer 1 on MIESR.CYC1 flag as ETHER_MINT interrupt source.	R/W
b2	MINTEN2	Pulse Output Timer 2 ETHER_MINT Interrupt Output Enable	0: Do not reflect rising edges of pulse output timer 2 on MIESR.CYC2 flag as ETHER_MINT interrupt source 1: Reflect rising edges of pulse output timer 2 on MIESR.CYC2 flag as ETHER_MINT interrupt source.	R/W
b3	MINTEN3	Pulse Output Timer 3 ETHER_MINT Interrupt Output Enable	0: Do not reflect rising edges of pulse output timer 3 on MIESR.CYC3 flag as ETHER_MINT interrupt source 1: Reflect rising edges of pulse output timer 3 on MIESR.CYC3 flag as ETHER_MINT interrupt source.	R/W
b4	MINTEN4	Pulse Output Timer 4 ETHER_MINT Interrupt Output Enable	0: Do not reflect rising edges of pulse output timer 4 on MIESR.CYC4 flag as ETHER_MINT interrupt source 1: Reflect rising edges of pulse output timer 4 on MIESR.CYC4 flag as ETHER_MINT interrupt source.	R/W
b5	MINTEN5	Pulse Output Timer 5 ETHER_MINT Interrupt Output Enable	0: Do not reflect rising edges of pulse output timer 5 on MIESR.CYC5 flag as ETHER_MINT interrupt source 1: Reflect rising edges of pulse output timer 5 on MIESR.CYC5 flag as ETHER_MINT interrupt source.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MITSELR register selects pulse output timers that generate ETHER_MINT interrupt requests. Each pulse output timer m (m = 0 to 5) takes the clock signal from the STCA as its clock source and produces pulses with a specified period and duty cycle. An ETHER_MINT interrupt is requested on rising edges of the pulse signal from the associated pulse output timer m if the setting of the MIEIPR.CYCm bit is 1. For more on the ETHER_MINT interrupt, see section 30.4, Interrupts.

30.2.12 ELC Output Timer Select Register (ELTSELR)

Address(es): EPTPC.ELTSELR 4006 5068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ELTDIS5	ELTDIS4	ELTDIS3	ELTDIS2	ELTDIS1	ELTDIS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ELTDIS0	Pulse Output Timer 0 Event Generation Disable	0: Use pulse output timer 0 for generating event signals for the ELC 1: Do not use pulse output timer 0 for generating event signals for the ELC.	R/W
b1	ELTDIS1	Pulse Output Timer 1 Event Generation Disable	0: Use pulse output timer 1 for generating event signals for the ELC 1: Do not use pulse output timer 1 for generating event signals for the ELC.	R/W
b2	ELTDIS2	Pulse Output Timer 2 Event Generation Disable	0: Use pulse output timer 2 for generating event signals for the ELC 1: Do not use pulse output timer 2 for generating event signals for the ELC.	R/W
b3	ELTDIS3	Pulse Output Timer 3 Event Generation Disable	0: Use pulse output timer 3 for generating event signals for the ELC 1: Do not use pulse output timer 3 for generating event signals for the ELC.	R/W
b4	ELTDIS4	Pulse Output Timer 4 Event Generation Disable	0: Use pulse output timer 4 for generating event signals for the ELC 1: Do not use pulse output timer 4 for generating event signals for the ELC.	R/W
b5	ELTDIS5	Pulse Output Timer 5 Event Generation Disable	0: Use pulse output timer 5 for generating event signals for the ELC 1: Do not use pulse output timer 5 for generating event signals for the ELC.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ELTSELR register selects the pulse output timers that output event signals to the ELC. Each pulse output timer m (m = 0 to 5) takes the clock signal from the STCA as its clock source and produces pulses with a specified period and duty cycle. An event signal is output to the ELC on rising edges if the ELIPPR.CYCPm bit is set to 1 and on falling edges if the CYCNm bit in the same register is set to 1. For more on output of event signals to the ELC, see section 30.4, Interrupts.

30.2.13 Time Synchronization Channel Select Register (STCHSELR)

Address(es): EPTPC.STCHSELR 4006 506Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SYSEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SYSEL	Timer Information Input Select	0: Use time information from the SYNFP0 module 1: Setting prohibited.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The STCHSELR register selects the time information input to the STCA module.

30.2.14 Slave Time Synchronization Start Register (SYNSTARTR)

Address(es): EPTPC.SYNSTARTR 4006 5080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	STR	Slave Time Synchronization Control	0: Stop slave time synchronization 1: Start slave time synchronization.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYNSTARTR register starts or stops time synchronization. This register is used when the EPTPC is operating as a slave node.

30.2.15 Local Clock Counter Initial Value Load Directive Register (LCIVLDR)

Address(es): EPTPC.LCIVLDR 4006 5084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	LOAD	Local Clock Counter Initial Value Load Directive	0: Do not load initial value to the local clock counter 1: Load initial value to the local clock counter.	W*1
b31 to b1	—	Reserved	The write value should be 0.	W

Note 1. Do not change the value of this bit while the SYNSTARTR.STR bit is 1.

The LCIVLDR register specifies the value in the LCIVRU, LCIVRM, and LCIVRL registers as the initial value of the local clock counter.

30.2.16 Synchronization Loss Detection Threshold Register (SYNTDARU, SYNTDARL)

Address(es): EPTPC.SYNTDARU 4006 5090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the upper-order 32 bits of the threshold for detection of loss of synchronization.	R/W

Address(es): EPTPC.SYNTDARL 4006 5094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the threshold for detection of loss of synchronization.	R/W

The SYNTDARU and SYNTDARL registers specify the threshold value for offsetFromMaster to be used in determining loss of synchronization. When setting a threshold value, write the upper-order 32 bits to SYNTDARU and the lower-order 32 bits to SYNTDARL, in that order and in consecutive operations. If the offsetFromMaster value exceeds the value specified in SYNTDARU and SYNTDARL, a loss of synchronization is detected. Set the value in SYNTDARU and SYNTDARL in nanoseconds. SYNTDARU and SYNTDARL are not used when the device is operating as a master clock.

30.2.17 Synchronization Detection Threshold Register (SYNTDBRU, SYNTDBRL)

Address(es): EPTPC.SYNTDBRU 4006 5098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the upper-order 32 bits of the threshold for detection of synchronization.	R/W

Address(es): EPTPC.SYNTDBRL 4006 509Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the threshold for detection of synchronization.	R/W

The SYNTDBRU and SYNTDBRL registers specify the threshold value for offsetFromMaster to be used in determining synchronization. When setting a threshold value, write the upper-order 32 bits to SYNTDBRU and the lower-order 32 bits to SYNTDBRL, in that order and in consecutive operations. If the offsetFromMaster value is less than the value specified in SYNTDBRU and SYNTDBRL, synchronization is detected. Set the value in SYNTDBRU and SYNTDBRL in nanoseconds. SYNTDBRU and SYNTDBRL are not used when the device is operating as a master clock.

30.2.18 Local Clock Counter Initial Value Register (LCIVRU, LCIVRM, LCIVRL)

Address(es): EPTPC.LCIVRU 4006 50B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	These bits specify the upper-order 16 bits of the integer portion of the initial value for the local clock counter.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC.LCIVRM 4006 50B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the integer portion of the initial value for the clock counter.	R/W

Address(es): EPTPC.LCIVRL 4006 50B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the fractional portion of the initial value of the clock counter in nanoseconds.	R/W

The LCIVRU, LCIVRM, and LCIVRL registers specify the initial value in seconds of the clock counter. When setting an initial value, write the upper-order 16 bits of the integer portion to LCIVRU, the lower-order 32 bits of the integer portion to LCIVRM, and the fractional portion in nanoseconds to LCIVRL, in that order and in consecutive operations.

The value in these registers can be used as the initial value of the local clock counter. When setting these register values in the local clock counter, set the LCIVLDR.LOAD bit to 1.

(1) Example

When 2.000000025 (s) is set as the initial value, write the following values to the registers:

- LCIVRU: 0000_0000h
- LCIVRM: 0000_0002h
- LCIVRL: 0000_0019h.

30.2.19 Worst 10 Acquisition Directive Register (GETW10R)

Address(es): EPTPC.GETW10R 4006 5124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GW10
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	GW10	Worst 10 Acquisition Directive	0: Do not acquire the worst 10 values 1: Start acquiring the worst 10 values.	R/W*1
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not set this bit to 1 while the STMR.W10S bit is 0.

Software uses the GETW10R register to start calculation of gradient values for use in selecting the worst 10 values. A gradient value is the amount by which the timer counter of a slave is incremented when a given interval elapses. Setting the GW10 bit to 1 while the value of the STMR.W10S bit is 1 selects calculation of a gradient value by the EPTPC each time it receives a Sync message. Gradient values are calculated the number of times specified in the STMR.WINT[7:0] bits. The GW10 bit clears to 0 on completion of this number of calculations. The GETW10R register is not used when the device is operating as a master clock.

30.2.20 Positive Gradient Limit Register (PLIMITRU, PLIMITRM, PLIMITRL)

Address(es): EPTPC.PLIMITRU 4006 5128h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b30 to b0	—	—	These bits specify the upper-order 31 bits of the limit for the positive gradient.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Address(es): EPTPC.PLIMITRM 4006 512Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the middle-order 32 bits of the limit for the positive gradient.	R/W

Address(es): EPTPC.PLIMITRL 4006 5130h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the limit for the positive gradient.	R/W

The PLIMITRU, PLIMITRM, and PLIMITRL registers specify an upper limit on the gradient (= positive gradient) used in time synchronization. When setting the upper limit, write to the registers consecutively in the order of PLIMITRU, PLIMITRM, PLIMITRL. Gradients that exceed the value specified in these registers are not used in time synchronization. These registers are not used when the device is operating as a master clock. The registers are valid while the STMR.CMOD and W10S bits are 1.

Use the following expression to calculate the gradient value to be set in the registers:

PLIMITRU, PLIMITRM, and PLIMITRL register values = A (s)/T (s) × 2^{32}

A: Time (s) by which the slave local clock counter advances during the interval between received Sync messages

T: Actual time (s) between received Sync messages

For example, if the interval between Sync messages is 0.5 seconds and the local clock counter advances by 0.7 seconds during that time, and this is to be set as the limit, then the setting for PLIMITR = 0.7/0.5 × 2^{32} = 6 012 954 214 = 1 6666 6666h, and the settings for the individual registers are as follows:

- PLIMITRU = 0000_0000h
- PLIMITRM = 0000_0001h
- PLIMITRL = 6666_6666h.

The minimum setting depends on the STCA clock frequency as the clock source for counting by the local clock counter. For example, if the STCA clock frequency is 50 MHz, then the minimum allowable setting for PLIMITRU, PLIMITRM, and PLIMITRL = (1/50 (MHz)) (s)/0.5 (s) × 2^{32} = 172 = ACh, and the settings for the individual registers are as follows:

- PLIMITRU = 0000_0000h
- PLIMITRM = 0000_0000h
- PLIMITRL = 0000_00ACh.

The gradient limit values to be set are valid when time synchronization correction mode is mode 2 (STMR.CMOD is 1) and the gradient is controlled by software (STMR.W10S is 1).

30.2.21 Negative Gradient Limit Register (MLIMITRU, MLIMITRM, MLIMITRL)

Address(es): EPTPC.MLIMITRU 4006 5134h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b30 to b0	—	—	These bits specify the upper-order 31 bits of the limit for the negative gradient.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Address(es): EPTPC.MLIMITRM 4006 5138h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the middle-order 32 bits of the limit for the negative gradient.	R/W

Address(es): EPTPC.MLIMITRL 4006 513Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the limit for the negative gradient.	R/W

The MLIMITRU, MLIMITRM, and MLIMITRL registers specify a lower limit on the gradient (= negative gradient) used in time synchronization. Use a two’s complement value to set the lower limit. When setting the lower limit, write to the registers consecutively in the order of MLIMITRU, MLIMITRM, MLIMITRL. Gradients that are less than the value specified in these registers are not used in time synchronization. These registers are not used when the device is operating as a master clock. The registers are valid while the STMR.CMOD and W10S bits are 1.

The procedure for setting the value, and the minimum value that can be set, are the same as for the PLIMITRU, PLIMITRM, and PLIMITRL registers.

30.2.22 Statistical Information Retention Control Register (GETINFOR)

Address(es): EPTPC.GETINFOR 4006 5140h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INFO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	INFO	Information Retention Control	When written: 0: No effect 1: Information is retained. When read: 0: Information retention is complete 1: Processing for information retention is in progress. After information fetching is directed, values of some statistical information read before completion of information fetching are not guaranteed.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GETINFOR register controls retention of the following statistical information:

- LCCVRU, LCCVRM, and LCCVRL registers
- PW10VRU, PW10VRM, and PW10VRL registers
- MW10RU, MW10RM, and MW10RL registers.

The only value that is writable to the INFO bit is 1. When setting a value in the PW10VRU, PW10VRM, and PW10VRL registers, or the MW10RU, MW10RM, and MW10RL registers, only set the INFO bit to 1 while the STMR.W10S bit is 1. If the INFO bit is set to 1 before acquisition of the worst 10 values is complete, the information retained in the PW10VRU, PW10VRM, and PW10VRL registers and MW10RU, MW10RM, and MW10RL registers is not guaranteed to be correct. Use the GETW10R.GW10 bit to confirm that acquisition is completed before setting the INFO bit to 1. The INFO bit automatically returns to 0 on completion of information fetching.

30.2.23 Local Clock Counter (LCCVRU, LCCVRM, LCCVRL)

Address(es): EPTPC.LCCVRU 4006 5170h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	These bits indicate the upper-order 16 bits of the integer portion of the value of the local clock counter.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

Address(es): EPTPC.LCCVRM 4006 5174h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the lower-order 32 bits of the integer portion of the value in the local clock counter.	R

Address(es): EPTPC.LCCVRL 4006 5178h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the fractional portion of the value in the local clock counter (in nanoseconds).	R

The LCCVRU, LCCVRM, and LCCVRL registers indicate the value of the local clock counter. When the GETINFOR.INFO bit is set to 1, the value of the local clock counter at that time is stored in these registers as follows:

- LCCVRU: Upper-order 16 bits of the integer portion in seconds
- LCCVRM: Lower-order 32 bits of the integer portion in seconds
- LCCVRL: Fractional portion in nanoseconds.

For example, if the local time information is 14:25, 44 seconds, 10 milliseconds, 23 microseconds, and 39 nanoseconds, the registers have $14 \times 3600 + 25 \times 60 + 44 = 51944$ (s) = 0000_0000_CAE8h as the setting of the upper-order 48 bits and $10 \times 10^6 + 23 \times 10^3 + 39 = 10023039$ (ns) = 0098_F07Fh as the setting of the lower-order 32 bits.

30.2.24 Positive Gradient Worst 10 Value Register (PW10VRU, PW10VRM, PW10VRL)

Address(es): EPTPC.PW10VRU 4006 5210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the upper-order 32 bits of the positive gradient value.	R

Address(es): EPTPC.PW10VRM 4006 5214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the middle-order 32 bits of the positive gradient value.	R

Address(es): EPTPC.PW10VRL 4006 5218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the lower-order 32 bits of the positive gradient value.	R

The PW10VRU, PW10VRM, and PW10VRL registers indicate the worst 10 of the positive gradient values. When the GETINFOR.INFO bit is set to 1, the worst 10 values at that time are stored in these registers. The format of the worst 10 gradients stored in the registers is the same as for the PLIMITRU, PLIMITRM, and PLIMITRL registers. See the PLIMITR register descriptions. The PW10VRU, PW10VRM, and PW10VRL registers are not used when the device is used as a master clock.

30.2.25 Negative Gradient Worst 10 Value Register (MW10RU, MW10RM, MW10RL)

Address(es): EPTPC.MW10RU 4006 52D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the upper-order 32 bits of the negative gradient value.	R

Address(es): EPTPC.MW10RM 4006 52D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the middle-order 32 bits of the negative gradient value.	R

Address(es): EPTPC.MW10RL 4006 52D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the lower-order 32 bits of the negative gradient value.	R

The MW10RU, MW10RM, and MW10RL registers indicate the worst 10 of the negative gradient values. When the GETINFOR.INFO bit is set to 1, the worst 10 value at that time is stored in these registers. The format of the worst 10 gradients stored in the registers is the same as for the MLIMITRU, MLIMITRM, and MLIMITRL registers. See the MLIMITR register descriptions. The MW10RU, MW10RM, and MW10RL registers are not used when the device is used as a master clock.

30.2.26 Timer Start Time Setting Register m (TMSTTRUm, TMSTTRLm) (m = 0 to 5)

Address(es): EPTPC.TMSTTRU0 4006 5300h, EPTPC.TMSTTRU1 4006 5310h, EPTPC.TMSTTRU2 4006 5320h, EPTPC.TMSTTRU3 4006 5330h, EPTPC.TMSTTRU4 4006 5340h, EPTPC.TMSTTRU5 4006 5350h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the upper-order 32 bits of the start time of the pulse output timer in nanoseconds.	R/W

Address(es): EPTPC.TMSTTRL0 4006 5304h, EPTPC.TMSTTRL1 4006 5314h, EPTPC.TMSTTRL2 4006 5324h, EPTPC.TMSTTRL3 4006 5334h EPTPC.TMSTTRL4 4006 5344h, EPTPC.TMSTTRL5 4006 5354h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the start time of the pulse output timer in nanoseconds.	R/W

The TMSTTRUm and TMSTTRLm register specify the start time of pulse output timer m. Set the start time of pulse output timer m (64 bits) in nanoseconds. Although the setting is in nanoseconds, the start time of pulse output timer m depends on the resolution of the STCA clock. For example, if the STCA clock is running at 50 MHz, 1 cycle takes 20 ns, so the time at which the timer starts might differ from the time set in these registers by up to 20 ns. When writing to the registers, write values consecutively in the order of TMSTTRUm, TMSTTRLm, while the TMSTARTR.ENm bit is 0. The format for setting times in these registers differs from that described in section 30.2.23, Local Clock Counter (LCCVRU, LCCVRM, LCCVRL).

30.2.27 Timer Cycle Setting Registers m (TMCYCRm) (m = 0 to 5)

Address(es): EPTPC.TMCYCR0 4006 5308h, EPTPC.TMCYCR1 4006 5318h, EPTPC.TMCYCR2 4006 5328h, EPTPC.TMCYCR3 4006 5338h
EPTPC.TMCYCR4 4006 5348h, EPTPC.TMCYCR5 4006 5358h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b29 to b0	—	—	These bits specify the cycle of the pulse output timer in nanoseconds. Set a value that is equivalent to at least 4 cycles of the STCA clock.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMCYCRm registers specify the period of the output signal generated by the associated pulse output timer m. Set a value in nanoseconds that is equivalent to at least 4 cycles of the STCA clock while the value of the TMSTARTR.ENm bit is 0. Although the setting is in nanoseconds, the period of the output signal generated by pulse output timer m and the time at which the timer starts depend on the period of the STCA clock. For example, if the STCA clock is running at 50 MHz, 1 cycle takes 20 ns, so the clock source for counting by pulse output timer m might differ from the period set in these registers by up to 19 ns. The SYNFP module handles calculations to correct this difference.

For example, if the setting for the timer period is 81 ns and the STCA clock is running at 50 MHz, the only available settings close to the actual timer period are for 80 or 100 ns. By setting the timer period in the SYNFP module to 80 ns for 19 and to 100 ns for 1 of every 20 cycles, the average period can be adjusted to 81 ns.

$$(80\ (\text{ns}) \times 19 + 100\ (\text{ns}) \times 1)/20 = 81\ (\text{ns})$$

The minimum value that can be set in a TMCYCRm register is 4 cycles of the STCA clock. For example, if the STCA clock is running at 50 MHz, the minimum setting corresponds to 80 ns. Timer operation is not guaranteed if a value set in one of these registers is less than this value.

30.2.28 Timer Pulse Width Setting Register m (TMPLSRm) (m = 0 to 5)

Address(es): EPTPC.TMPLSR0 4006 530Ch, EPTPC.TMPLSR1 4006 531Ch, EPTPC.TMPLSR2 4006 532Ch, EPTPC.TMPLSR3 4006 533Ch,
EPTPC.TMPLSR4 4006 534Ch, EPTPC.TMPLSR5 4006 535Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b28 to b0	—	—	These bits specify the high-level width of the pulse signal from the timer in nanoseconds. Set a value that is equivalent to at least 2 cycles of the STCA clock.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMPLSRm registers specify the high-level width of the output signal generated by the associated pulse output timer m. When the TMSTARTR.ENm bit is 0, set a value corresponding to a time no shorter than 2 cycles of the STCA clock in nanoseconds. Although the setting is in nanoseconds, the high-level width of the signal from the timer depends on the period of the STCA clock. The method for correcting the high-level width of the signal from the timer is the same as that for correcting the timer periods set in the TMCYCRm register.

The upper-order 3 bits of the TMPLSRm register are reserved. These bits are read as 000b. When writing, write 000b to these bits.

30.2.29 Timer Start Register (TMSTARTR)

Address(es): EPTPC.TMSTARTR 4006 537Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EN0	Pulse Output Timer 0 Start	0: Stop pulse output timer 0 1: Start pulse output timer 0.	R/W
b1	EN1	Pulse Output Timer 1 Start	0: Stop pulse output timer 1 1: Start pulse output timer 1.	R/W
b2	EN2	Pulse Output Timer 2 Start	0: Stop pulse output timer 2 1: Start pulse output timer 2.	R/W
b3	EN3	Pulse Output Timer 3 Start	0: Stop pulse output timer 3 1: Start pulse output timer 3.	R/W
b4	EN4	Pulse Output Timer 4 Start	0: Stop pulse output timer 4 1: Start pulse output timer 4.	R/W
b5	EN5	Pulse Output Timer 5 Start	0: Stop pulse output timer 5 1: Start pulse output timer 5.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMSTARTR register starts and stops the pulse output timers.

30.2.30 SYNFP Status Register (SYSR)

Address(es): EPTPC0.SYSR 4006 5800h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GENDN	RESDN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	INFABT	—	RECLP	—	—	—	—	—	DRQOVR	INTDEV	DRPTO	—	MPDUD	INTCHG	OFMUD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OFMUD	offsetFromMaster Value Update Flag	0: offsetFromMaster value not updated 1: offsetFromMaster value updated.	R/W*1
b1	INTCHG	Receive logMessageInterval Value Change Detection Flag	0: Received logMessageInterval value did not change 1: Received logMessageInterval value changed.	R/W*1
b2	MPDUD	meanPathDelay Value Update Flag	0: meanPathDelay value not updated 1: meanPathDelay value updated.	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DRPTO	Delay_Resp/Pdelay_Resp Reception Timeout Detection Flag	0: No Delay_Resp/Pdelay_Resp timeout occurred 1: Delay_Resp/Pdelay_Resp timeout occurred.	R/W*1
b5	INTDEV	Receive logMessageInterval Value Out-of-Range Flag	0: Received logMessageInterval value is within the range 1: Received logMessageInterval value is out of the range.	R/W*1
b6	DRQOVR	Delay_Req Reception FIFO Overflow Detection Flag	0: Received Delay_Req did not cause the reception FIFO to overflow 1: Received Delay_Req caused the reception FIFO to overflow.	R/W*1
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	RECLP	Loop Reception Detection Flag	0: Received message did not return through a loop 1: Received message returned through a loop.	R/W*1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	INFABT	Control Information Abnormality Detection Flag	0: No abnormality in control information 1: Abnormality in control information.	R/W*1
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b16	RESDN	Response Stop Completion Detection Flag	0: Stopping responses not completed 1: Stopping responses completed.	R/W*1
b17	GENDN	Generation Stop Completion Detection Flag	0: Stopping generation not completed 1: Stopping generation completed.	R/W*1
b23 to b18	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 clears the flag. Writing 0 does not affect the flag value.

The SYSR register indicates the state of the SYNFP module.

OFMUD flag (offsetFromMaster Value Update Flag)

The OFMUD flag indicates that the value of offsetFromMaster was updated.

INTCHG flag (Receive logMessageInterval Value Change Detection Flag)

The INTCHG flag indicates that the logMessageInterval value of the Delay_Resp, Sync or Announce message differs from the previously received value.

MPDUD flag (meanPathDelay Value Update Flag)

The MPDUD flag indicates that the value of meanPathDelay was updated.

DRPTO flag (Delay_Resp/Pdelay_Resp Reception Timeout Detection Flag)

The DRPTO flag indicates that a Delay_Resp or Pdelay_Resp message was not received within the period set in the RSTOUTR register.

INTDEV flag (Receive logMessageInterval Value Out-of-Range Flag)

The INTDEV flag indicates that a Delay_Resp message was received with a logMessageInterval value outside the range, -7 to +6.

DRQOVR flag (Delay_Req Reception FIFO Overflow Detection Flag)

The DRQOVR flag indicates that the FIFO buffer for storing information from received Delay_Req messages holds 32 or more entries.

RECLP flag (Loop Reception Detection Flag)

The RECLP flag indicates that the value of the sourcePortIdentity field in a received PTP message matches the local PortIdentity as set in the SYCIDRU, SYCIDRL, and SYPNUMR registers.

INFABT flag (Control Information Abnormality Detection Flag)

The INFABT flag indicates that the control information includes a mismatch. If an erroneous frame is detected because of a corrupted frame or noise in the external circuit when the ETHERC and EPTPC are receiving data, subsequent normal frames might not be received properly.

Reset the EDMAC, ETHERC, and EPTPC after an erroneous frame is detected. Then, wait for the required number of cycles before setting communications again.

- Detecting an erroneous frame

  To detect an erroneous frame, read the INFABT flag in the SYNFP Status Register (SYSR) of EPTPC0. An INFABT flag is provided for EPTPC0.
  When the EPTPC0 is not used and only the EDMAC0 and ETHERC0 are used to receive and transmit standard Ethernet frames, read the INFABT flag to detect an erroneous frame.

- Resetting after detection of an erroneous flag

  When the EPTPC0.SYSR.INFABT flag is set to 1, reset the EPTPC0 and ETHERC0, and then wait for the required number of cycles before setting the registers.
  When the EPTPC0 is not used and only the EDMAC0 and ETHERC0 are used to receive and transmit standard Ethernet frames, reset the EPTPC0 and the registers. In this case, resetting PTPEDMAC is not required.
  To reset the EPTPC0 and the registers:

  a. Set the EPTPC_CFG.PTRSTR.RESET bit to 1 (reset the EPTPC0 by software).

  b. Set the EDMAC0.EDMR.SWR bit to 1 (reset the EDMAC0 and ETHERC0 by software).

  c. Use a software loop or timer to wait for at least 64 cycles of the peripheral module clock, PCLKA. This step is necessary to initialize EDMAC0 and ETHERC0.

  d. Set the EPTPC_CFG.PTRSTR.RESET bit to 0 (release the EPTPC0 reset).

  e. Reset communications by setting the EDMAC0, ETHERC0, PTPEDMAC, and EPTPC0 registers to enable communications.

RESDN flag (Response Stop Completion Detection Flag)

The RESDN flag indicates the end of processing for transmission of a Delay_Resp or Pdelay_Resp as response messages when the handling of a received Delay_Req or Pdelay_Req by the SYNFP module is disabled in the SYRFL1R or SYRVLDR register.

GENDN flag (Generation Stop Completion Detection Flag)

The GENDN flag indicates the end of processing for transmission of messages of a type disabled in the SYTRENR or SYRVLDR register.

30.2.31 SYNFP Status Notification Enable Register (SYIPR)

Address(es): EPTPC0.SYIPR 4006 5804h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GENDN	RESDN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	INFABT	—	RECLP	—	—	—	—	—	DRQOVR	INTDEV	DRPTO	—	MPDUD	INTCHG	OFMUD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OFMUD	SYSR.OFMUD Status Notification Enable	0: Disable notification of the SYSR.OFMUD state 1: Enable notification of the SYSR.OFMUD state.	R/W
b1	INTCHG	SYSR.INTCHG Status Notification Enable	0: Disable notification of the SYSR.INTCHG state 1: Enable notification of the SYSR.INTCHG state.	R/W
b2	MPDUD	SYSR.MPDUD Status Notification Enable	0: Disable notification of the SYSR.MPDUD state 1: Enable notification of the SYSR.MPDUD state.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DRPTO	SYSR.DRPTO Status Notification Enable	0: Disable notification of the SYSR.DRPTO state 1: Enable notification of the SYSR.DRPTO state.	R/W
b5	INTDEV	SYSR.INTDEV Status Notification Enable	0: Disable notification of the SYSR.INTDEV state 1: Enable notification of the SYSR.INTDEV state.	R/W
b6	DRQOVR	SYSR.DRQOVR Status Notification Enable	0: Disable notification of the SYSR.DRQOVR state 1: Enable notification of the SYSR.DRQOVR state.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	RECLP	SYSR.RECLP Status Notification Enable	0: Disable notification of the SYSR.RECLP state 1: Enable notification of the SYSR.RECLP state.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	INFABT	SYSR.INFABT Status Notification Enable	0: Disable notification of the SYSR.INFABT state 1: Enable notification of the SYSR.INFABT state.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b16	RESDN	SYSR.RESDN Status Notification Enable	0: Disable notification of the SYSR.RESDN state 1: Enable notification of the SYSR.RESDN state.	R/W
b17	GENDN	SYSR.GENDN Status Notification Enable	0: Disable notification of the SYSR.GENDN state 1: Enable notification of the SYSR.GENDN state.	R/W
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYIPR register specifies whether the MIESR.SY0 flag reflects changes in the state of the SYNFP0 module.

30.2.32 SYNFP MAC Address Registers (SYMACRU, SYMACRL)

Address(es): EPTPC0.SYMACRU 4006 5810h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the upper-order 24 bits of the local MAC address.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC0.SYMACRL 4006 5814h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the lower-order 24 bits of the local MAC address.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYMACRU and SYMACRL registers specify the local MAC address for Ethernet ports 0. Set these registers before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.33 SYNFP LLC-CTL Value Register (SYLLCCTLR)

Address(es): EPTPC0.SYLLCCTLR 4006 5818h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CTL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTL[7:0]	LLC-CTL Field	These bits specify the value used for the control field in the LLC sublayer when generating IEEE802.3 frames.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYLLCCTLR register specifies the control field (LLC-CTL) value of LLC frames generated by the SYNFP module. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.34 SYNFP Local IP Address Register (SYIPADDRR)

Address(es): EPTPC0.SYIPADDRR 4006 581Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the local IP address.	R/W

The SYIPADDRR register specifies the local IP address for Ethernet port 0. Set the SYIPADDRR register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.35 SYNFP Specification Version Setting Register (SYSPVRR)

Address(es): EPTPC0.SYSPVRR 4006 5840h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TRSP[3:0]				VER[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	VER[3:0]	versionPTP Field Value	These bits specify the versionPTP field value of the PTP v2 header. When a message is received, this value is compared with the versionPTP field of the received frame. In generating messages, the value is used for the versionPTP field of the frame to be transmitted. Set these bits to 0010b (PTP v2).	R/W
b7 to b4	TRSP[3:0]	transportSpecific Field Value	These bits specify the transportSpecific field value of the PTP v2 header. When a message is received, this value is compared with the transportSpecific field of the received frame. In generating messages, the value is used for the transportSpecific field of the frame to be transmitted. Set these bits to 0000b (IEEE 1588).	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYSPVRR register specifies the transportSpecific and versionPTP field values of the PTP v2 message header. Do not change the settings while reception or transmission of PTP messages is enabled.

30.2.36 SYNFP Domain Number Setting Register (SYDOMR)

Address(es): EPTPC0.SYDOMR 4006 5844h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DNUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DNUM[7:0]	domainNumber Field Value Setting	These bits specify the domainNumber field value of the PTP v2 header. When a message is received, this value is compared with the domainNumber field of the received frame as a condition for PTP reception processing. In generating messages, the value is used for the domainNumber field of the frame to be transmitted.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYDOMR register specifies the domainNumber field value of the PTP v2 message header. Do not change the settings while reception or transmission of PTP messages is enabled.

30.2.37 Announce Message Flag Field Setting Register (ANFR)

Address(es): EPTPC0.ANFR 4006 5850h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	—	FLAG8	—	—	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FLAG0	leap61	This bit specifies the logical value of the leap61 member of timePropertiesDS. 0: Set leap61 to FALSE 1: Set leap61 to TRUE.	R/W
b1	FLAG1	leap59	This bit specifies the logical value of the leap59 member of timePropertiesDS. 0: Set leap59 to FALSE 1: Set leap59 to TRUE.	R/W
b2	FLAG2	currentUtcOffsetValid	This bit specifies the logical value of the currentUtcOffsetValid member of timePropertiesDS. 0: Set currentUtcOffsetValid to FALSE 1: Set currentUtcOffsetValid to TRUE.	R/W
b3	FLAG3	ptpTimescale	This bit specifies the logical value of the ptpTimescale member of timePropertiesDS. 0: Set ptpTimescale to FALSE 1: Set ptpTimescale to TRUE.	R/W
b4	FLAG4	timeTraceable	This bit specifies the logical value of the timeTraceable member of timePropertiesDS. 0: Set timeTraceable to FALSE 1: Set timeTraceable to TRUE.	R/W
b5	FLAG5	frequencyTraceable	This bit specifies the logical value of the frequencyTraceable member of timePropertiesDS. 0: Set frequencyTraceable to FALSE 1: Set frequencyTraceable to TRUE.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FLAG8	alternateMasterFlag	0: Set alternateMasterFlag to FALSE 1: Set alternateMasterFlag to TRUE.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	FLAG10	unicastFlag	0: Set unicastFlag to FALSE 1: Set unicastFlag to TRUE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: Set PTP profile Specific 1 to FALSE 1: Set PTP profile Specific 1 to TRUE.	R/W
b14	FLAG14	PTP profile Specific 2	0: Set PTP profile Specific 2 to FALSE 1: Set PTP profile Specific 2 to TRUE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ANFR register specifies the flagField section of the header when the SYNFP module is to generate an Announce message. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

30.2.38 Sync Message Flag Field Setting Register (SYNFR)

Address(es): EPTPC0.SYNFR 4006 5854h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	FLAG9	FLAG8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FLAG8	alternateMasterFlag	0: Set alternateMasterFlag to FALSE 1: Set alternateMasterFlag to TRUE.	R/W
b9	FLAG9	twoStepFlag	Set this bit to 0 (FALSE).	R/W
b10	FLAG10	unicastFlag	0: Set unicastFlag to FALSE 1: Set unicastFlag to TRUE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: Set PTP profile Specific 1 to FALSE 1: Set PTP profile Specific 1 to TRUE.	R/W
b14	FLAG14	PTP profile Specific 2	0: Set PTP profile Specific 2 to FALSE 1: Set PTP profile Specific 2 to TRUE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYNFR register specifies the flagField section of the header when the SYNFP module is to generate a Sync message. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

30.2.39 Delay_Req Message Flag Field Setting Register (DYRQFR)

Address(es): EPTPC0.DYRQFR 4006 5858h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	FLAG10	unicastFlag	0: Set unicastFlag to FALSE 1: Set unicastFlag to TRUE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: Set PTP profile Specific 1 to FALSE. 1: Set PTP profile Specific 1 to TRUE.	R/W

Bit	Symbol	Bit name	Description	R/W
b14	FLAG14	PTP profile Specific 2	0: Set PTP profile Specific 2 to FALSE. 1: Set PTP profile Specific 2 to TRUE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DYRQFR register specifies the flagField section of the header when the SYNFP module is to generate a Delay_Req or Pdelay_Req message. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

30.2.40 Delay_Resp Message Flag Field Setting Register (DYRPFR)

Address(es): EPTPC0.DYRPFR 4006 585Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FLAG14	FLAG13	—	—	FLAG10	FLAG9	FLAG8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	FLAG8	alternateMasterFlag*1	0: Set alternateMasterFlag to FALSE 1: Set alternateMasterFlag to TRUE.	R/W
b9	FLAG9	twoStepFlag*2	Set this bit to 0 (FALSE).	R/W
b10	FLAG10	unicastFlag	0: Set unicastFlag to FALSE 1: Set unicastFlag to TRUE.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FLAG13	PTP profile Specific 1	0: Set PTP profile Specific 1 to FALSE 1: Set PTP profile Specific 1 to TRUE.	R/W
b14	FLAG14	PTP profile Specific 2	0: Set PTP profile Specific 2 to FALSE 1: Set PTP profile Specific 2 to TRUE.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is reserved for Pdelay_Resp messages. Set the bit to 0.
Note 2. This bit is reserved for Delay_Resp messages.

The DYRPFR register specifies the flagField section of the header when the SYNFP module is to generate a Delay_Resp or Pdelay_Resp message. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1. Do not change the settings in this register while transmission of Delay_Resp or Pdelay_Resp messages is enabled. After disabling this transmission processing, do not change the settings in this register until the SYSR.RESDN flag sets to 1.

30.2.41 SYNFP Local Clock ID Register (SYCIDRU, SYCIDRL)

Address(es): EPTPC0.SYCIDRU 4006 5860h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the upper-order 32 bits of the clock-ID of the local port.	R/W

Address(es): EPTPC0.SYCIDRL 4006 5864h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the clock-ID of the local port.	R/W

The SYCIDR register specifies the clock-ID of the local port. The clock-ID is used for the clockIdentity section in the sourcePortIdentity field of the header when the SYNFP module is to generate a PTP message. When a PTP message is received, the value in these registers is compared with the clockIdentity section in the sourcePortIdentity field of the PTP message to determine whether the message is one that was transmitted by your application. Renesas recommends making this setting the same as the value of portDS.portIdentity.clockIdentity in most cases.

Do not change the settings in these registers while reception or transmission of PTP messages is enabled.

30.2.42 SYNFP Local Port Number Register (SYPNUMR)

Address(es): EPTPC0.SYPNUMR 4006 5868h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PNUM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	PNUM[15:0]	Local Port Number Setting	These bits specify the port number of the local port.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYPNUMR register specifies the port number of the local port. This register is used for the portNumber section in the sourcePortIdentity field of the header when the SYNFP module is to generate a PTP message. When a PTP message is received, the value in this register is compared with the portNumber section in the sourcePortIdentity field of the PTP message to determine whether the message is one that was transmitted by the local device. Renesas recommends making this setting the same as the value of portDS.portIdentity.portNumber in most cases.

Do not change the settings in this register while reception or transmission of PTP messages is enabled.

30.2.43 SYNFP Register Value Load Directive Register (SYRVLDR)

Address(es): EPTPC0.SYRVLDR 4006 5880h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ANUP	STUP	BMUP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BMUP	BMC Update	When this bit is set to 1, the SYNFP module simultaneously reflects values of registers storing the information that identifies MasterClock.	W
b1	STUP	State Update	When this bit is set to 1, the SYNFP module simultaneously reflects register values for PTP message reception and transmission.	W
b2	ANUP	Announce Message Generation Information Update	When this bit is set to 1, the Announce message generation block simultaneously reflects register values required for generating Announce messages.	W
b31 to b3	—	Reserved	The write value should be 0.	W

The SYRVLDR register simultaneously updates multiple register values in the SYNFP module.

BMUP bit (BMC Update)

When the BMUP bit is set to 1, the SYNFP module simultaneously reflects the values of the following registers that store the information that identifies MasterClock:

- MTCIDU and MTCIDL registers
- MTPID register.

STUP bit (State Update)

When the STUP bit is set to 1, the SYNFP module simultaneously reflects the values of the following registers and bits for PTP message reception and transmission:

- SYNFR register
- DYRQFR register
- SYTLIR.DREQ[7:0] bits
- RSTOUTR register
- SYRFL1R register
- SYRFL2R register
- SYTRENR register.

ANUP bit (Announce Message Generation Information Update)

When the ANUP bit is set to 1, the Announce message generation block simultaneously reflects the values of the following registers and bits required for generating Announce messages:

- ANFR register
- SYTLIR.ANCE[7:0] bits
- GMPR register
- GMCQR register
- GMIDRU and GMIDRL registers
- CUOTSR register
- SRR register.

30.2.44 SYNFP Reception Filter Register 1 (SYRFL1R)

Address(es): EPTPC0.SYRFL1R 4006 5890h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	PDFUP2	—	PDFUP0	—	PDRP2	—	PDRP0	—	PDRQ2	—	PDRQ0	—	DRP2	—	DRP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	DRQ2	—	DRQ0	—	FUP2	—	FUP0	—	SYNC2	—	SYNC0	—	—	—	ANCE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ANCE0	Announce Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SYNC0	Sync Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	SYNC2	Sync Message Processing	0: Do not process messages in the SYNFP 1: Process messages in the SYNFP.	R/W

Bit	Symbol	Bit name	Description	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	FUP0	Follow_Up Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	FUP2	Follow_Up Message Processing	0: Do not process messages in the SYNFP 1: Process messages in the SYNFP.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DRQ0	Delay_Req Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	DRQ2	Delay_Req Message Processing	0: Do not process messages in the SYNFP 1: Process messages in the SYNFP.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b16	DRP0	Delay_Resp Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b17	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18	DRP2	Delay_Resp Message Processing	0: Do not process messages in the SYNFP 1: Process messages in the SYNFP.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b20	PDRQ0	Pdelay_Req Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b21	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22	PDRQ2	Pdelay_Req Message Processing	0: Do not process messages in the SYNFP 1: Process messages in the SYNFP.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b24	PDRP0	Pdelay_Resp Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26	PDRP2	Pdelay_Resp Message Processing	0: Do not process messages in the SYNFP 1: Process messages in the SYNFP.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b28	PDFUP0	Pdelay_Resp_Follow_Up Message Processing	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b29	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30	PDFUP2	Pdelay_Resp_Follow_Up Message Processing	0: Do not process messages in the SYNFP 1: Process messages in the SYNFP.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The SYRFL1R register specifies filtering for the reception of PTP messages. Multiple bits corresponding to different types of messages can be set to 1. Setting all bits for a type of message to 0 leads to all messages of the given type being discarded. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

30.2.45 SYNFP Reception Filter Register 2 (SYRFL2R)

Address(es): EPTPC0.SYRFL2R 4006 5894h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	ILL0	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	SIG0	—	—	—	MAN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MAN0	Management Message Processing Setting	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SIG0	Signaling Message Processing Setting	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b27 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	ILL0	Illegal Message Processing Setting*1	0: Do not transfer messages to the PTPEDMAC 1: Transfer messages to the PTPEDMAC.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. PTP messages other than PTP v2 messages and messages of undefined type are handled as illegal messages.

The SYRFL2R register specifies filtering for the reception of PTP messages. Multiple bits corresponding to different types of messages can be set to 1. Setting all bits for a type of message to 0 leads to all messages of the given type being discarded. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

30.2.46 SYNFP Transmission Enable Register (SYTRENR)

Address(es): EPTPC0.SYTRENR 4006 5898h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PDRQ	—	—	—	DRQ	—	—	—	SYNC	—	—	—	ANCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ANCE	Announce Message Transmission Enable	0: Do not transmit Announce messages 1: Transmit Announce messages.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SYNC	Sync Message Transmission Enable	0: Do not transmit Sync messages 1: Transmit Sync messages.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DRQ	Delay_Req Message Transmission Enable	0: Do not transmit Delay_Req messages 1: Transmit Delay_Req messages.	R/W

Bit	Symbol	Bit name	Description	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	PDRQ	Pdelay_Req Message Transmission Enable	0: Do not transmit Pdelay_Req messages 1: Transmit Pdelay_Req messages.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYTRENR register enables or disables transmission of PTP messages. Do not set the PDRQ and DRQ bits to 1 at the same time. Operation is not guaranteed when both bits are set to 1. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

30.2.47 Master Clock ID Register (MTCIDU, MTCIDL)

Address(es): EPTPC0.MTCIDU 4006 58A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the upper-order 32 bits of the clock-ID of the master clock.	R/W

Address(es): EPTPC0.MTCIDL 4006 58A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the clock-ID of the master clock.	R/W

The MTCIDU and MTCIDL registers specify the clock-ID of the master clock for synchronization. The value specified in these registers is only reflected in the SYNFP module after the SYRVLDR.BMUP bit is set to 1.

30.2.48 Master Clock Port Number Register (MTPID)

Address(es): EPTPC0.MTPID 4006 58A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PNUM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	PNUM[15:0]	Master Clock Port Number Setting	These bits specify the port number of the master clock.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MTPID register specifies the port number of the master clock for synchronization. The value specified in this register is only reflected in the SYNFP module after the SYRVLDR.BMUP bit is set to 1. In normal usage, set the value of parentDS.parentPortIdentity.portNumber in this register.

30.2.49 SYNFP Transmission Interval Setting Register (SYTLIR)

Address(es): EPTPC0.SYTLIR 4006 58C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DREQ[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC[7:0]								ANCE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b7 to b0	ANCE[7:0]	Announce Message Transmission Interval Setting	These bits set the interval for the transmission of Announce messages.	R/W
b15 to b8	SYNC[7:0]	Sync Message Transmission Interval Setting	These bits set the interval for the transmission of Sync messages. The setting is also placed in the logMessageInterval field of transmitted Sync messages.	R/W
b23 to b16	DREQ[7:0]	Delay_Req Transmission Interval Average Value/ Pdelay_Req Transmission Interval Setting	The bits set the average interval for the transmission of Delay_Req messages and the interval for the transmission of Pdelay_Req messages.The setting is also placed in the logMessageInterval field of Delay_Resp messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYTLIR register specifies the interval for the transmission of messages generated by the SYNFP module. The setting is an integer logarithm in base 2 (log2(x)) and determines a value x in seconds. In other words, the interval for transmission is 2^n (s), where n is the setting. The available settings are from -7 (F9h) to +6 (06h).

Examples:

- If the setting is 06h, then the interval for transmission is $2^6 = 64$ (s)
- If the setting is 00h, then the interval for transmission is $2^0 = 1$ (s)

- If the setting is FFh, then the interval for transmission is $2^{-1} = 0.5$ (s) = 500 (ms)
- If the setting is F9h, then the interval for transmission is $2^{-7} = 0.0078125$ (s) = 7.8125 (ms).

The value specified in the ANCE[7:0] bits is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1. The values specified in the DREQ[7:0] and SYNC[7:0] bits are only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

30.2.50 SYNFP Received logMessageInterval Value Indication Register (SYRLIR)

Address(es): EPTPC0.SYRLIR 4006 58C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DRESP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC[7:0]								ANCE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	ANCE[7:0]	Announce Message logMessageInterval Field Indication Flag	These bits indicate the logMessageInterval field value of a received Announce message.	R
b15 to b8	SYNC[7:0]	Sync Message logMessageInterval Field Indication Flag	These bits indicate the logMessageInterval field value of a received Sync message.	R
b23 to b16	DRESP[7:0]	Delay_Resp Message logMessageInterval Field Indication Flag	These bits indicate the logMessageInterval field value of a received Delay_Resp message.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

The SYRLIR register indicates the logMessageInterval field values of received PTP messages.

30.2.51 offsetFromMaster Value Register (OFMRU, OFMRL)

Address(es): EPTPC0.OFMRU 4006 58C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the upper-order 32 bits of the calculated offsetFromMaster value.	R

Address(es): EPTPC0.OFMRL 4006 58CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the lower-order 32 bits of the calculated offsetFromMaster value.	R

The OFMRU and OFMRL registers indicate the calculated offsetFromMaster value. The value is expressed as a two’s complement in nanoseconds. The numeric representation differs from that of the offsetFromMaster member of the current data set (currentDS), as shown in the following note. For reads, access the registers in the order of OFMRU, OFMRL.

Note 1. The value of currentDS.offsetFromMaster is multiplied by 2^{16}. Example: 2.5 (ns) = 0000_0000_0002_8000h

30.2.52 meanPathDelay Value Register (MPDRU, MPDRL)

Address(es): EPTPC0.MPDRU 4006 58D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the upper-order 32 bits of the calculated meanPathDelay value.	R

Address(es): EPTPC0.MPDRL 4006 58D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits indicate the lower-order 32 bits of the calculated meanPathDelay value.	R

The MPDRU and MPDRL registers indicate the calculated meanPathDelay value. The value is expressed as a two's complement in nanoseconds. The numeric representation differs from that of the meanPathDelay member of the current data set (currentDS), as shown in the following note. For reads, access the registers in the order of MPDRU, MPDRL.

Note 1. The value of currentDS.meanPathDelay is multiplied by 2^{16}. Example: 2.5 (ns) = 0000_0000_0002_8000h

30.2.53 grandmasterPriority Field Setting Register (GMPR)

Address(es): EPTPC0.GMPR 4006 58E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	GMPR1[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	GMPR2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	GMPR2[7:0]	grandmasterPriority2 Field Value Setting	These bits specify the value of the grandmasterPriority2 fields of Announce messages.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	GMPR1[7:0]	grandmasterPriority1 Field Value Setting	These bits specify the value of the grandmasterPriority1 fields of Announce messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GMPR register specifies the grandmasterPriority1 and grandmasterPriority2 field values of Announce messages generated by the SYNFP module. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

30.2.54 grandmasterClockQuality Field Setting Register (GMCQR)

Address(es): EPTPC0.GMCQR 4006 58E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the value of the grandmasterClockQuality fields of Announce messages. The associations between bits and the grandmasterClockQuality fields is as follows: b31 to b24: clockClass b23 to b16: clockAccuracy b15 to b0: offsetScaledLogVariance.	R/W

The GMCQR register specifies the grandmasterClockQuality field value of Announce messages generated by the SYNFP module. The value specified in the GMCQR register is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

30.2.55 grandmasterIdentity Field Setting Register (GMIDRU, GMIDRL)

Address(es): EPTPC0.GMIDRU 4006 58E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the upper-order 32 bits of the value of the grandmasterIdentity fields of Announce messages.	R/W

Address(es): EPTPC0.GMIDRL 4006 58ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the value of the grandmasterIdentity fields of Announce messages.	R/W

The GMIDRU and GMIDRL registers specify the grandmasterIdentity field value of Announce messages generated by the SYNFP module. The value specified in these registers is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

30.2.56 currentUtcOffset/timeSource Field Setting Register (CUOTSR)

Address(es): EPTPC0.CUOTSR 4006 58F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CUTO[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TSRC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	TSRC[7:0]	timeSource Field Setting	These bits specify the value of the timeSource fields of Announce messages.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	CUTO[15:0]	currentUtcOffset Field Setting	These bits specify the value of the currentUtcOffset fields of Announce messages.	R/W

The CUOTSR register specifies the currentUtcOffset and timeSource field values of Announce messages generated by the SYNFP module. The values specified in this register are only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

30.2.57 stepsRemoved Field Setting Register (SRR)

Address(es): EPTPC0.SRR 4006 58F4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SRMV[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	SRMV[15:0]	stepsRemoved Field Value Setting	These bits specify the value of the stepsRemoved fields of Announce messages.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SRR register specifies the stepsRemoved field value of Announce messages generated by the SYNFP module. The value specified in this register is only reflected in the SYNFP module after the SYRVLDR.ANUP bit is set to 1.

30.2.58 PTP-primary Message Destination MAC Address Setting Register (PPMACRU, PPMACRL)

Address(es): EPTPC0.PPMACRU 4006 5900h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the upper-order 24 bits of the destination MAC address for PTP-primary messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC0.PPMACRL 4006 5904h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the lower-order 24 bits of the destination MAC address for PTP-primary messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PPMACRU and PPMACRL registers specify the destination MAC address for PTP-primary messages. In normal usage, set 01:1B:19:00:00:00 in these registers. The value is used in the destination MAC address field when generating an Ethernet frame for a PTP-primary message. It is also used as a determining condition for received frames carrying PTP messages. Set these registers before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.59 PTP-pdelay Message MAC Address Setting Register (PDMACRU, PDMACRL)

Address(es): EPTPC0.PDMACRU 4006 5908h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the upper-order 24 bits of the destination MAC address for PTP-pdelay messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC0.PDMACRL 4006 590Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the lower-order 24 bits of the destination MAC address for PTP-pdelay messages.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PDMACRU and PDMACRL registers specify the destination MAC address for PTP-pdelay messages. In normal usage, set 01:80:C2:00:00:0E in these registers. This value is used in the destination MAC address field when generating frames carrying PTP-pdelay messages in the Ethernet format. It is also used as a determining condition for received frames carrying PTP messages. Set these registers before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.60 PTP Message Ethertype Setting Register (PETYPER)

Address(es): EPTPC0.PETYPER 4006 5910h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[15:0]															
Value after reset:	1	0	0	0	1	0	0	0	1	1	1	1	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b15 to b0	TYPE[15:0]	PTP Message Ethertype Value Setting	These bits specify the Ethertype field value for frames in the Ethernet II format.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PETYPER register specifies the Ethertype field for frames carrying the PTP messages. In normal usage, set 0000_88F7h in this register. This value is used in the Ethertype field when generating frames carrying PTP messages in the Ethernet II format. It is also used as a determining condition for received frames carrying PTP messages. Set these registers before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.61 PTP-primary Message Destination IP Address Setting Register (PPIPR)

Address(es): EPTPC0.PPIPR 4006 5920h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the destination IP address for PTP-primary messages.	R/W

The PPIPR register specifies the destination IP address for PTP messages. In normal usage, set E000_0181h (224.0.1.129) in this register. This value is used in the destination IP address field when generating frames carrying PTP-primary messages in the IPv4 format. The lower-order 23 bits are also used in the destination MAC address field for Ethernet frames. The value is also used as a determining condition for received frames carrying PTP messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.62 PTP-pdelay Message Destination IP Address Setting Register (PDIPR)

Address(es): EPTPC0.PDIPR 4006 5924h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the destination IP address for PTP-pdelay messages.	R/W

The PDIPR register specifies the destination IP address for PTP-pdelay messages. In normal usage, set E000_006Bh (224.0.0.107) in this register. The value is used in the destination IP address field when generating frames carrying PTP-pdelay messages in the IPv4 format. The lower-order 23 bits are also used in the destination MAC address field for Ethernet frames. The value is also used as a determining condition for received frames carrying PTP messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.63 PTP Event Message TOS Setting Register (PETOSR)

Address(es): EPTPC0.PETOSR 4006 5928h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	EVTO[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	EVTO[7:0]	PTP Event Message TOS Field Value Setting	These bits specify the value of the TOS field within the IPv4 headers of PTP event messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PETOSR register specifies the TOS (type of service) field value within the IPv4 headers of PTP event messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.64 PTP general Message TOS Setting Register (PGTOSR)

Address(es): EPTPC0.PGTOSR 4006 592Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	GETO[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	GETO[7:0]	PTP general Message TOS Field Value Setting	These bits specify the value of the TOS field within the IPv4 headers of PTP general messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PGTOSR register specifies the TOS (type of service) field value within the IPv4 headers of PTP general messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.65 PTP-primary Message TTL Setting Register (PPTTLR)

Address(es): EPTPC0.PPTTLR 4006 5930h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PRTL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	PRTL[7:0]	PTP-primary Message TTL Field Value Setting	These bits specify the value of the TTL field within the IPv4 headers of PTP-primary messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PPTTLR register specifies the TTL (time to live) field value within the IPv4 headers of PTP-primary messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.66 PTP-pdelay Message TTL Setting Register (PDTTLR)

Address(es): EPTPC0.PDTTLR 4006 5934h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PDTL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b7 to b0	PDTL[7:0]	PTP-pdelay Message TTL Field Value	These bits specify the value of the TTL field within the IPv4 headers of PTP-pdelay messages.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PDTTLR register specifies the TTL field value within the IPv4 headers of PTP-pdelay messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.67 PTP Event Message UDP Destination Port Number Setting Register (PEUDPR)

Address(es): EPTPC0.PEUDPR 4006 5938h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EVUPT[15:0]															
Value after reset:	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b15 to b0	EVUPT[15:0]	PTP Event Message Destination Port Number Setting	These bits specify the value of the destination port number field within the UDP headers of PTP event messages.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PEUDPR register specifies the destination port number field value within the UDP headers of PTP event messages. In normal usage, set 013Fh (319) in this register. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.68 PTP general Message UDP Destination Port Number Setting Register (PGUDPR)

Address(es): EPTPC0.PGUDPR 4006 593Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GEUPT[15:0]															
Value after reset:	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	GEUPT[15:0]	PTP general Message Destination Port Number	These bits specify the value of the destination port number field within the UDP headers of PTP general messages.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PGUDPR register specifies the destination port number field value within the UDP headers of PTP general messages. In normal usage, set 0140h (320) in this register. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.69 Frame Reception Filter Setting Register (FFLTR)

Address(es): EPTPC0.FFLTR 4006 5940h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EXTPRM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ENB	PRT	SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SEL	Receive MAC Address Select*1	These bits select how filtering is handled when multicast frames other than PTP messages are received. b2 b0 0 0 0: Disable filtering (receive all multicast frames) 0 0 1: Disable filtering (receive all multicast frames) 0 1 0: Disable filtering (receive all multicast frames) 0 1 1: Disable filtering (receive all multicast frames) 1 0 0: Do not receive multicast frames 1 0 1: Do not receive multicast frames 1 1 0: Only receive multicast frames matching the MAC address setting in FMAC0RU and FMAC0RL 1 1 1: Only receive multicast frames matching the MAC address setting in FMAC1RU and FMAC1RL.	R/W
b1	PRT	Frame Reception Enable*1		R/W
b2	ENB	Reception Filter Enable*1		R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EXTPRM	Extended Promiscuous Mode Setting	0: Normal operation (receive unicast frames addressed to the EPTPC, filter PTP frames, filter multicast frames, and receive all broadcast frames) 1: Extended promiscuous mode (receive all frames).	R/W

Bit	Symbol	Bit name	Description	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

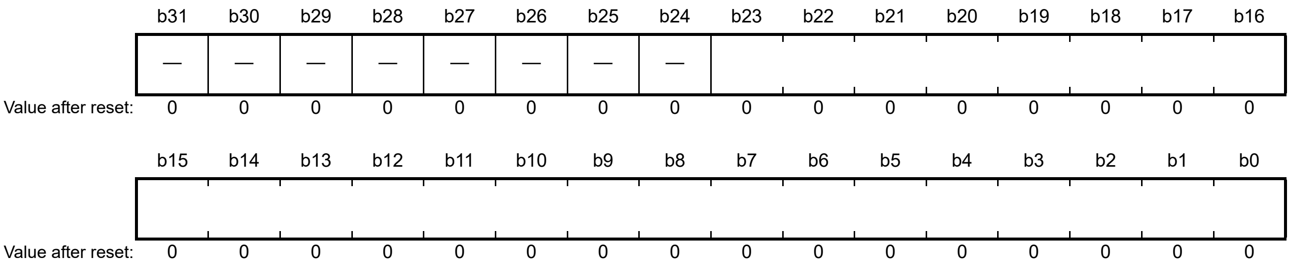
Note 1. The setting in these bits is only valid when the EXTPRM bit is 0.

The FFLTR register switches extended promiscuous mode on or off and selects how filtering is handled when multicast frames other than PTP messages are received. To enable the filter for the reception of multicast frames other than PTP messages, set the ENB, PRT, and SEL bits to 110b or 111b. Frames passed by the filter are then transferred by EDMAC0.

Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

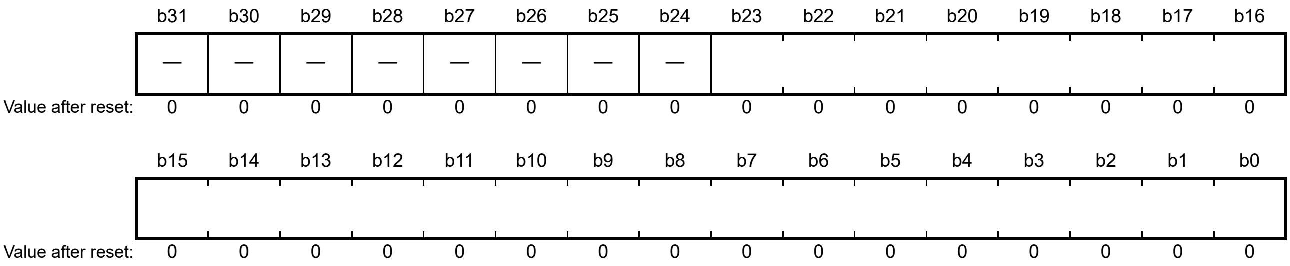
30.2.70 Frame Reception Filter MAC Address 0 Setting Register (FMAC0RU, FMAC0RL)

Address(es): EPTPC0.FMAC0RU 4006 5960h



Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the upper-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC0.FMAC0RL 4006 5964h



Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the lower-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The FMAC0RU and FMAC0RL registers specify the MAC address for filtering during the reception of multicast frames other than PTP messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.71 Frame Reception Filter MAC Address 1 Setting Register (FMAC1RU, FMAC1RL)

Address(es): EPTPC0.FMAC1RU 4006 5968h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the upper-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC0.FMAC1RL 4006 596Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits specify the lower-order 24 bits of the destination MAC address for received multicast frames.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The FMAC1RU and FMAC1RL registers specify the MAC address for filtering during the reception of multicast frames other than PTP messages. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.72 Asymmetric Delay Setting Register (DASYMRU, DASYMRL)

Address(es): EPTPC0.DASYMRU 4006 59C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	These bits specify the upper-order 16 bits of the asymmetric delay value. Set them to 0000h in this MCU.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): EPTPC0.DASYMRL 4006 59C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the lower-order 32 bits of the asymmetric delay value. Set them to 0000_0000h in this MCU.	R/W

The DASYMRU and DASYMRL registers specify the asymmetric delay value (delayAsymmetry). Set the registers DASYMRU and DASYMRL to 0000_0000h in this MCU.

30.2.73 Timestamp Latency Setting Register (TSLATR)

Address(es): EPTPC0.TSLATR 4006 59C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	INGP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EGP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	EGP[15:0]	Output Port Timestamp Latency Setting	These bits specify the timestamp latency (ns) for the output ports.	R/W
b31 to b16	INGP[15:0]	Input Port Timestamp Latency Setting	These bits specify the timestamp latency (ns) for the input ports.	R/W

The TSLATR register specifies the amount of latency in timestamp acquisition in nanoseconds. Do not change the settings while reception or transmission of PTP messages is enabled.

EGP[15:0] bits (Output Port Timestamp Latency Setting)

Set the EGP[15:0] bits to the fixed values listed in Table 30.8 for the target system. The timestamp latency differs with the link transfer rate (100 or 10 Mbps) and the frequency of the STCA clock (20, 25, 50, or 100 MHz).

Table 30.8 EGP[15:0] bit settings (ns)

Link transfer rate		STCA clock frequency			
		20 MHz	25 MHz	50 MHz	100 MHz
MII	100 Mbps	590	625	695	730
	10 Mbps	7430	7465	7535	7570
RMII	100 Mbps	770	805	875	910
	10 Mbps	9230	9265	9335	9370

INGP[15:0] bits (Input Port Timestamp Latency Setting)

Set the INGP[15:0] bits to the fixed values listed in Table 30.9 for the target system. The timestamp latency differs with the link transfer rate (100 or 10 Mbps) and the frequency of the STCA clock (20, 25, 50, or 100 MHz).

Table 30.9 INGP[15:0] bit settings (ns)

Link transfer rate		STCA clock frequency			
		20 MHz	25 MHz	50 MHz	100 MHz
MII	100 Mbps	980	945	875	840
	10 Mbps	8180	8145	8075	8015
RMII	100 Mbps	1060	1025	955	920
	10 Mbps	8980	8945	8875	8815

30.2.74 SYNFP Operation Setting Register (SYCONFR)

Address(es): EPTPC0.SYCONFR 4006 59CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FILDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SBDIS	—	—	—	—	TCYC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	TCYC[7:0]	PTP Message Transmission Interval Setting	These bits specify the time from the completion of one transmission to the start of the next in transmission clock cycles. A value n in these bits means that a transmission interval of n cycles is secured. No interval is secured if the setting is 00h. Recommended setting: 28h (40 cycles).	R/W
b11 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	SBDIS	Sync Message Transmission Bandwidth Securing Disable	0: Enable securing of the bandwidth for the transmission of SYNC messages (give lower priority to transfers by the EDMAC) 1: Disable securing of the bandwidth for the transmission of SYNC messages (give higher priority to transfers by the EDMAC).	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	FILDIS	Receive Message domainNumber Filter Disable	0: Include comparison with the domainNumber field in the filtering conditions for the reception of PTP messages 1: Do not include comparison with the domainNumber field in the filtering conditions for the reception of PTP messages.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYCONFR register controls operation of the SYNFP module. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

TCYC[7:0] bits (PTP Message Transmission Interval Setting)

The TCYC[7:0] bits specify a wait time between packets to secure a fixed transmission delay. The setting defines the interval from input of the transmission completed signal from the ETHERC to output of the next transmission request as a number of cycles of the transmission clock, which runs at 2.5 MHz if the link transfer rate is 10 Mbps and 25 MHz if the rate is 100 Mbps.

SBDIS bit (Sync Message Transmission Bandwidth Securing Disable)

The SBDIS bit disables securing of bandwidth to increase accuracy of the interval for the transmission of SYNC messages.

FILDIS bit (Receive Message domainNumber Filter Disable)

The FLDIS bit selects whether or not to include comparison with the domainNumber field in the filtering conditions for the reception of PTP messages.

30.2.75 SYNFP Frame Format Setting Register (SYFORMR)

Address(es): EPTPC0.SYFORMR 4006 59D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FORM1	FORM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FORM0	Ethernet Frame Format Setting	0: Ethernet II frame format 1: IEEE802.3 frame format.	R/W
b1	FORM1	Ethernet/UDP Encapsulation	0: PTP directly over Ethernet 1: PTP over UDP/IPv4.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SYFORMR register specifies the format for frame generation by the SYNFP module. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

30.2.76 Response Message Reception Timeout Register (RSTOUTR)

Address(es): EPTPC0.RSTOUTR 4006 59D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	Response Message Reception Timeout Time Setting	If no response message is received within n × 1024 (ns), where n is the setting in these bits, a timeout is detected.	R/W

The RSTOUTR register specifies the time for detection of a timeout during the reception of PTP response messages (Delay_Resp and Pdelay_Resp). If no Delay_Resp or Pdelay_Resp message is received within the time specified in this register after transmission of a Delay_Req or Pdelay_Req message, the SYSR.DRPTO flag is set to 1. The value specified in this register is only reflected in the SYNFP module after the SYRVLDR.STUP bit is set to 1.

30.2.77 PTP Reset Register (PTRSTR)

Address(es): EPTPC_CFG.PTRSTR 4006 4500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RESET
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RESET	EPTPC Software Reset	0: Do not reset the EPTPC 1: Reset the EPTPC.*1	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not access the EPTPC-related registers other than this register while a software reset is being issued.

The PTRSTR register resets the EPTPC. It takes 64 cycles of the peripheral module clock (PCLKA) until initialization of the EPTPC is complete. After the RESET bit is set to 1, wait for 64 PCLKA cycles before clearing its value to 0.

30.2.78 STCA Clock Select Register (STCSELR)

Address(es): EPTPC_CFG.STCSELR 4006 4504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCLKSEL[2:0]			—	—	—	—	—	SCLKDIV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SCLKDIV[2:0]	PCLKA Clock Frequency Division	b2 b0 0 0 1: 1 0 1 0: 1/2 0 1 1: 1/3 1 0 0: 1/4 1 0 1: 1/5 1 1 0: 1/6. Other settings are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	SCLKSEL[2:0]	STCA Clock Select	b10 b8 0 0 0: Use PCLKA clock divided by 1 to 6 0 1 0: Input clock from the REF50CK0 pin. Other settings are prohibited.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The STCSELR register selects the STCA clock signal for the EPTPC. Set this register before starting the EDMAC, ETHERC, or PTPEDMAC. Do not change the settings while the EPTPC is operating.

SCLKDIV[2:0] bits (PCLKA Clock Frequency Division)

The SCLKDIV[2:0] bits select the division ratio of PCLKA. When the setting of the SCLKSEL[2:0] bits is 000b, the frequency-divided PCLKA is used as the STCA clock signal.

SCLKSEL[2:0] bits (STCA Clock Select)

The SCLKSEL[2:0] bits select the STCA clock signal for use in the EPTPC.

30.2.79 Bypass 1588 Module Register (BYPASS)

Address(es): EPTPC_CFG.BYPASS 4006 4508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BYPASS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BYPASS0	Bypass1588 module for Ether 0ch	0: Use 1588 module for Ether channel 0 1: Bypass 1588 module for Ether channel 0.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Do not access the BYPASS register while the Ether module is in operation. When the EPTPC is not used, bypass it by setting the BYPASS register.

30.3 Operation

After release from the reset state, the EPTPC is set to not receive (analyze) or transmit (generate) PTP messages, so it has no effect on the transmission or reception of frames by the ETHERC and EDMAC at that time. The EPTPC registers must be configured to transmit and receive PTP messages for the ETHERC and EDMAC to be able to use packet filtering by MAC address in the SYNFP module.

Figure 30.3 shows a block diagram of the modules involved in frame transfer.

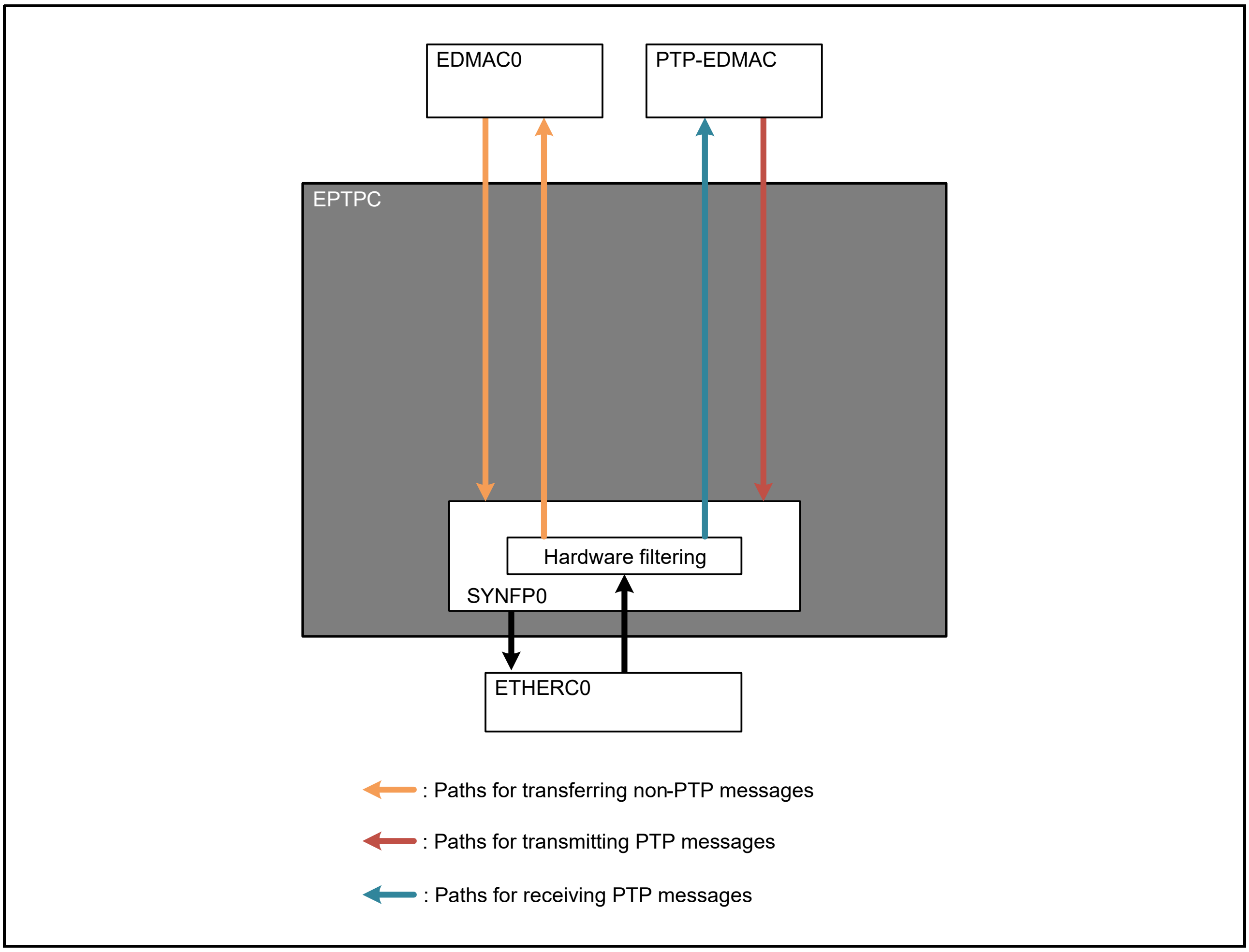


Figure 30.3 **Block diagram of the modules involved in frame transfers**

30.3.1 Transmission and Reception of Non-PTP Messages

The EPTPC operates in extended promiscuous mode when the FFLTR.EXTPRM bit setting is 1. In this mode, all frames received by the Ethernet ports are transferred to the EDMAC without filtering. The EPTPC operates in normal mode when the FFLTR.EXTPRM bit setting is 0. In this mode, the SYNFP module applies its hardware filtering function to filter frames received by the Ethernet ports.

The EPTPC and EDMAC transfer received unicast frames if they are for the given node.

Operation when multicast frames are received can be selected from the following: frames are transferred to the EDMAC, frames are not transferred to the EDMAC, or frames are transferred to the EDMAC only when the address matches the specified MAC address.

The EPTPC transfers received broadcast frames to the EDMAC for the receiving Ethernet port.

30.3.2 Paths for the Transfer of Non-PTP Messages

Messages received through the Ethernet port are transferred to the EDMAC. Figure 30.4 is a diagram of paths for the transmission and reception of non-PTP messages.

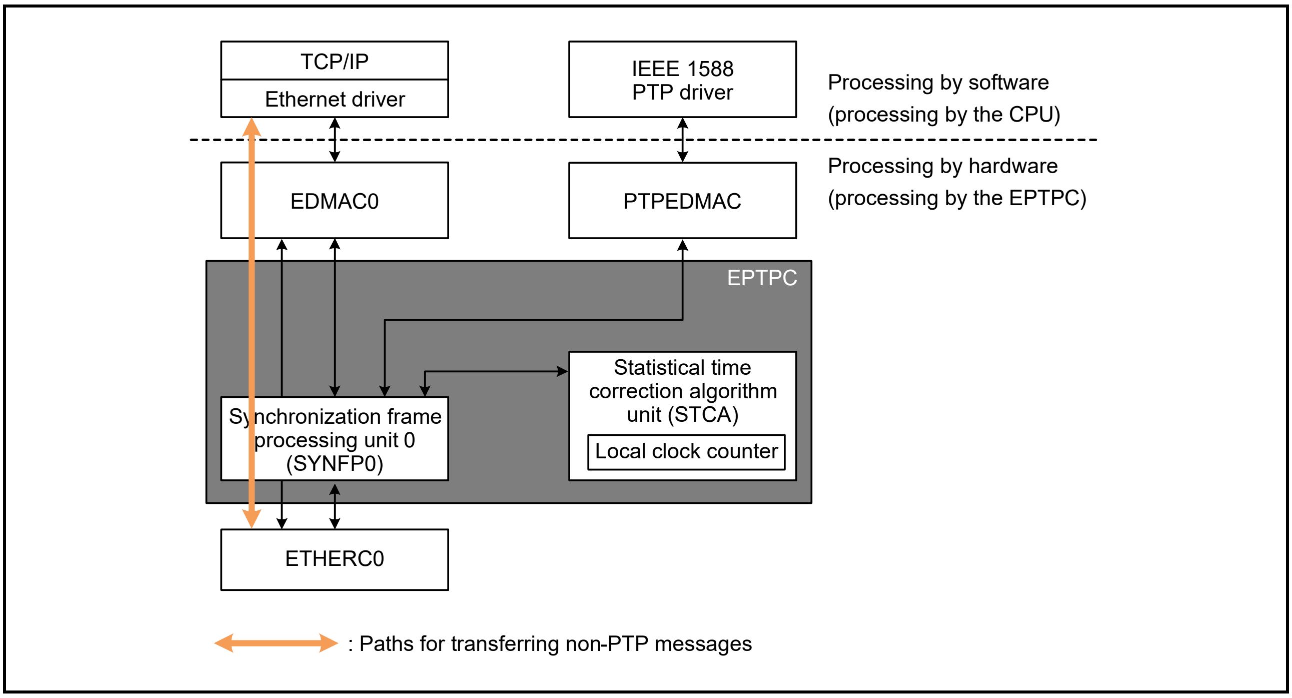


Figure 30.4 Paths for the transmission and reception of non-PTP messages

30.3.3 Transmission and Reception of PTP Messages

The EPTPC hardware automatically handles analysis and extraction of fields from received PTP messages, and generation and transmission of PTP messages. However, the software must still handle the transmission of certain PTP messages. Table 30.10 shows the specifications for control over the transmission and reception of the different PTP message types.

Table 30.10 Control over the transmission and reception of PTP messages

		OC (Ordinary Clock)	
Message type	**Message**	**Master**	**Slave**
Event	Sync	Generation (automatic)	Reception (automatic)
	Delay_Req	Generation (automatic)	Reception (automatic)
	Pdelay_Req	Generation and reception (automatic)	Generation and reception (automatic)
	Pdelay_Resp	Generation and reception (automatic)	Generation and reception (automatic)
General	Announce	Generation (automatic)	Reception (software)
	Follow_Up	—*1	Reception (automatic)
	Delay_Resp	Packet generation	Reception (automatic)
	Pdelay_Resp_Follow_Up	—*1	Reception (automatic)
	Management	Transmission and reception (software)	
	Signaling	Transmission and reception (software)	

Note 1. Control is not required as the clock for this is a one-step clock.

30.3.4 Paths for the Transfer of PTP Messages

Transfer paths for the PTP messages differ based on whether transfer requires processing by software or is automatically processed by hardware.

30.3.4.1 Paths for the transfer of PTP messages requiring processing by software

Figure 30.5 shows the paths for the transfer of PTP messages where transfer requires software processing. The figure shows paths for all message, clock-type, and process combinations for which "(software)" is indicated in Table 30.10.

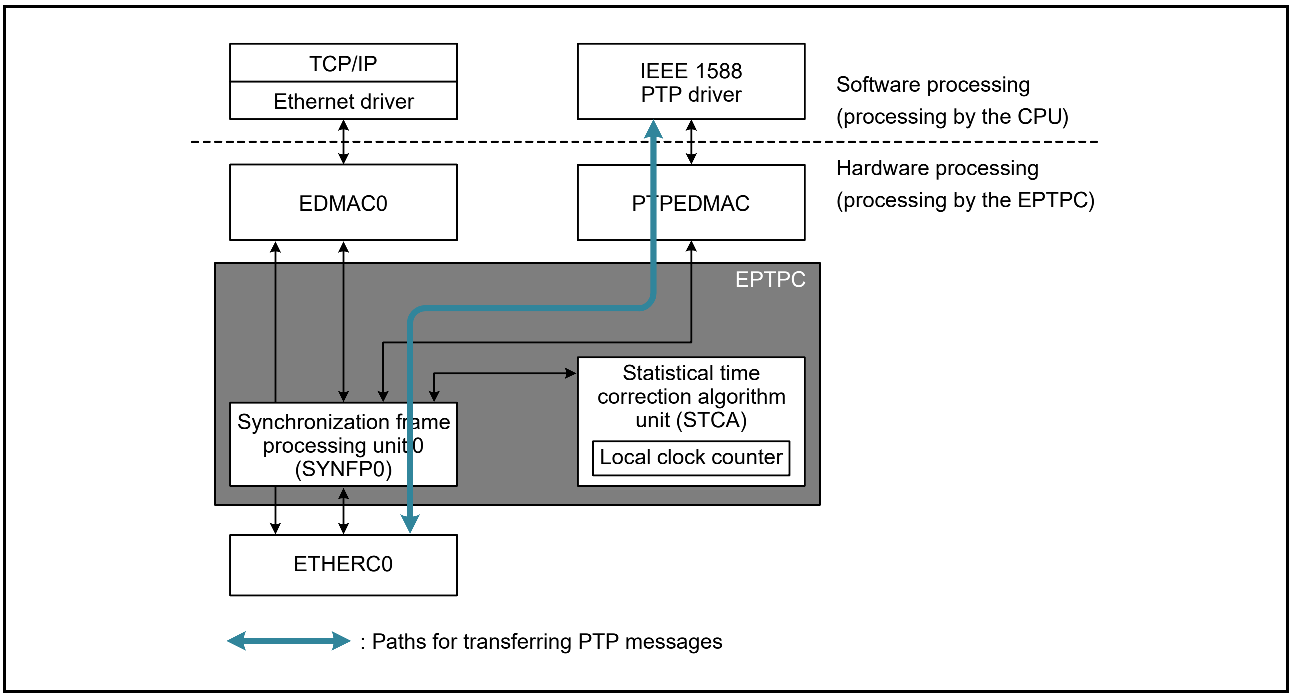


Figure 30.5 Paths for the transfer of PTP messages requiring software processing

30.3.4.2 Paths for the transfer of PTP messages handled automatically by hardware

For PTP messages for which the hardware automatically handles the processing, the SYNFP modules handle transmission and reception.

(1) Generation of and response to PTP messages by hardware

Figure 30.6 shows the transfer paths in the automatic generation of and response to PTP messages by the SYNFP module. The paths in the figure are used for the "Generation (automatic)", "Reception (automatic)", and "Generation and reception (automatic)" operations indicated in Table 30.10.

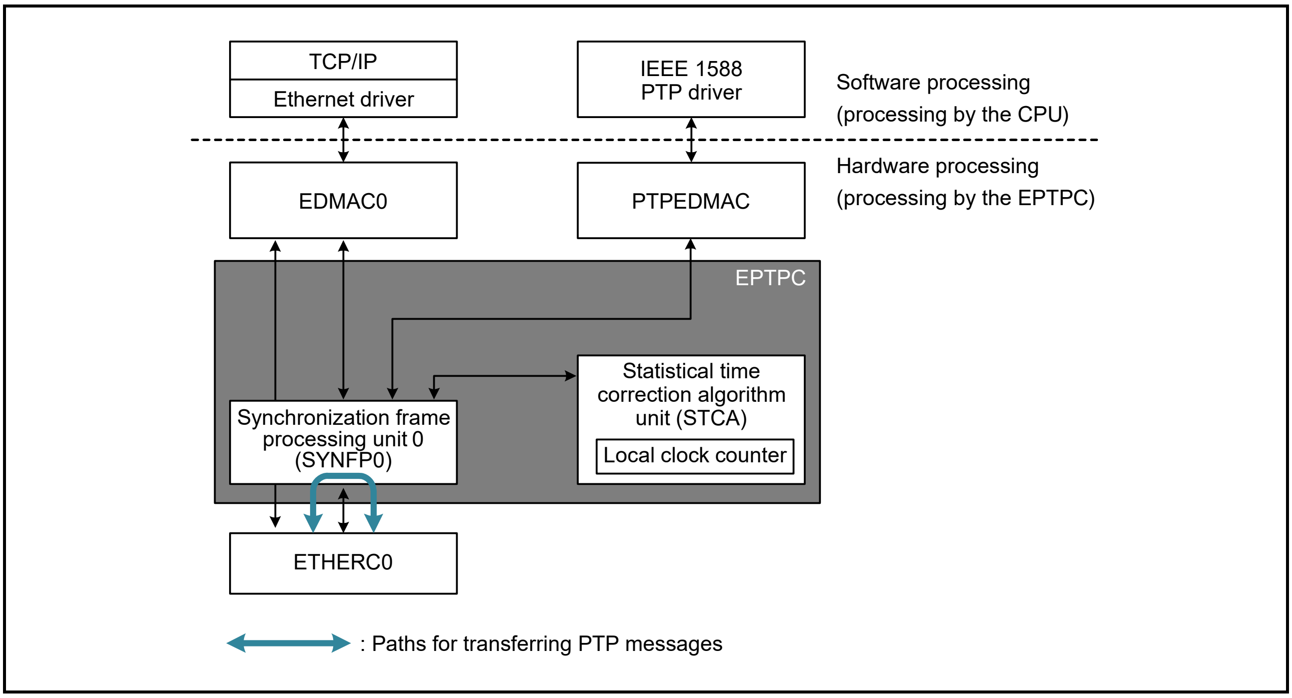


Figure 30.6 Paths for the generation of and response to PTP messages by hardware

30.3.5 Clock Devices

The EPTPC can operate as the clock devices defined in IEEE 1588.

30.3.5.1 End-to-End (E2E)

(1) Master

PTP messages are transmitted and received as described in Table 30.11 in operation as an end-to-end (E2E) master.

Table 30.11 Processing of PTP messages by an E2E master

Message type	Message	The EPTPC...
Event	Sync	Transmits Sync messages at the fixed interval specified in the SYTLIR.SYNC[7:0] bits.
	Delay_Req	When this message is received, transmits a Delay_Resp message in response.
	Pdelay_Req	—
	Pdelay_Resp	—
General	Announce	Transmits Announce messages at the fixed interval specified in the SYTLIR.ANCE[7:0] bits.
	Follow_Up	—
	Delay_Resp	Transmits this as the response to a received Delay_Req messages.
	Pdelay_Resp_Follow_Up	—
	Management	Transmits and receives Management messages by software through the PTPEDMAC.
	Signaling	Transmits and receives Signaling messages by software through the PTPEDMAC.

(2) Slave

PTP messages are transmitted and received as described in Table 30.12 in operation as an E2E slave, and the calculated offsetFromMaster is used to correct the local time information.

Table 30.12 Processing of PTP messages by an E2E slave

Message type	Message	The EPTPC...
Event	Sync	Calculates the offsetFromMaster value when this message is received if twoStepFlag in flagField was FALSE (1-step clock).
	Delay_Req	Transmits Delay_Req messages at random intervals from 0 to the time specified in the SYTLIR.DREQ[7:0] bits × 2.
	Pdelay_Req	—
	Pdelay_Resp	—
General	Announce	Transmits Announce messages by software through the PTPEDMAC.
	Follow_Up	Calculates the offsetFromMaster value when this message is received if twoStepFlag in flagField of the most recently received Sync message was TRUE (two-step clock).
	Delay_Resp	Calculates the meanPathDelay value when this message is received.
	Pdelay_Resp_Follow_Up	—
	Management	Transmits and receives Management messages by software through the PTPEDMAC.
	Signaling	Transmits and receives Signaling messages by software through the PTPEDMAC.

30.3.5.2 Peer-to-Peer (P2P)

(1) Master

PTP messages are transmitted and received as described in Table 30.13 in operation as a Peer-to-Peer (P2P) master.

Table 30.13 Processing of PTP messages by a P2P master

Packet type	Message	The EPTPC...
Event	Sync	Transmits timestamps for transmission at the fixed interval specified in the SYTLIR.SYNC[7:0] bits.
	Delay_Req	—
	Pdelay_Req	• Transmits Pdelay_Req messages at the fixed interval specified in the SYTLIR.DREQ[7:0] bits • Transmits a Pdelay_Resp message in response when this message is received.
	Pdelay_Resp	• Transmits this as the response to a received Pdelay_Req message • Calculates the meanPathDelay value when this message is received if twoStepFlag in flagField was FALSE (one-step clock).
General	Announce	Transmits Announce messages at the fixed interval specified in the SYTLIR.ANCE[7:0] bits.
	Follow_Up	—
	Delay_Resp	—
	Pdelay_Resp_Follow_Up	Calculates the meanPathDelay value when this message is received if twoStepFlag in flagField of the most recently received Pdelay_Resp message was TRUE (two-step clock).
	Management	Transmits Management messages by software through the PTPEDMAC.
	Signaling	Transmits Signaling messages by software through the PTPEDMAC.

(2) Slave

PTP messages are transmitted and received as described in Table 30.14 in operation as a P2P slave, and the calculated offsetFromMaster is used to correct the local time information.

Table 30.14 Processing of PTP messages by a P2P slave

Packet type	Message	The EPTPC...
Event	Sync	Calculates the offsetFromMaster value when this message is received if twoStepFlag in flagField was FALSE (1-step clock).
	Delay_Req	—
	Pdelay_Req	• Transmits Pdelay_Req messages at the fixed interval specified in the SYTLIR.DREQ[7:0] bits • Transmits a Pdelay_Resp message in response when this message is received.
	Pdelay_Resp	• Transmits this as the response to a received Pdelay_Req messages • Calculates the meanPathDelay value when this message is received if twoStepFlag in flagField was FALSE (one-step clock).
General	Announce	Transmits Announce messages by software through the PTPEDMAC.
	Follow_Up	Calculates the offsetFromMaster value when this message is received if twoStepFlag in flagField of the most recently received Sync message was TRUE (two-step clock).
	Delay_Resp	—
	Pdelay_Resp_Follow_Up	Calculates the meanPathDelay value when this message is received if twoStepFlag in flagField of the most recently received Pdelay_Resp message was TRUE (2-step clock).
	Management	Transmits and receives Management messages by software through the PTPEDMAC.
	Signaling	Transmits and receives Signaling messages by software through the PTPEDMAC.

30.3.5.3 Ordinary Clock (OC)

PTP messages are transmitted and received through one Ethernet port in operation as an ordinary clock. An ordinary clock operates as the grand master clock or as a slave clock in the master-slave hierarchy. For operation as an E2E master, E2E slave, P2P master, or P2P slave, see the following sections:

- section 30.3.7, Operation as an E2E Master
- section 30.3.8, Operation as an E2E Slave
- section 30.3.10, Operation as a P2P Master
- section 30.3.11, Operation as a P2P Slave.

30.3.6 EPTPC Initialization

Transmitting and receiving PTP messages requires the settings in the EPTPC registers listed in Table 30.15. Set the registers associated with the Ethernet port used. Also set the registers listed in Table 30.16 if UDP and IPv4 are used for the frame format of the PTP messages.

Table 30.15 Registers requiring settings for EPTPC initialization (1 of 2)

Register name	Settings	Description
STCFR	Example: 0000_0002h	The value of 50 MHz is given as an example. Three other settings are also available.
SYCONFR	Example: 0000_0028h	The setting differs with the type of PTP clock operation.
SYMACRU, SYMACRL	As wanted	—
SYSPVRR	0000_0002h	transportSpecific and version fields
SYDOMR	As wanted	—
SYCIDRU, SYCIDRL	As wanted	—
SYPNUMR	0000_0001h	If the PTP clock operates as an OC, the setting is 0000_0001h.
PPMACRU, PPMACRL	01:1B:19:00:00:00	MAC address for PTP-primary messages
PDMACRU, PDMACRL	01:80:C2:00:00:0E	MAC address for PTP-pdelay messages
DASYMRU, DASYMRL	0000_0000h	—
TSLATR	As wanted	Depends on the link transfer rate and STCA clock frequency

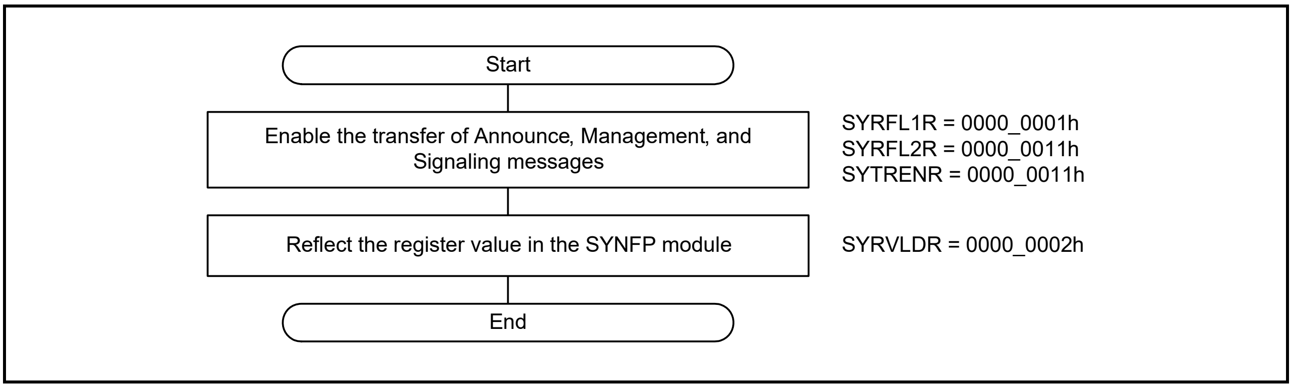
Table 30.15 Registers requiring settings for EPTPC initialization (2 of 2)

Register name	Settings	Description
SYFORMR	As wanted	Four settings are available.
SYLLCCTLR	0000_0003h	LLC-CTL field value for Ethernet frames
PETYPER	0000_88F7h	Ethertype for PTP messages

Table 30.16 Registers requiring additional settings when UDP or IPv4 is used

Register name	Settings	Description
SYIPADDRR	As wanted	Local IP address
PETOSR	As wanted	Set the highest allowable traffic class selector codepoint as the value for the differentiated service (DS) field.
PGTOSR	As wanted	—
PPTTLR	As wanted	TTL field value for PTP-primary messages
PEUDPR	0000_013Fh	UDP port number for event messages
PGUDPR	0000_0140h	UDP port number for general messages
PDIPR	0000_006Bh	IP address for PTP-pdelay messages
PDTTLR	0000_0001h	TTL field value for PTP-pdelay messages

In operation as an OC, set registers as shown in Figure 30.7 to transfer received Announce, Management, and Signaling messages to the PTPEDMAC.



SYRFL1R = 0000_0001h
SYRFL2R = 0000_0011h
SYTRENR = 0000_0011h

SYRVLDR = 0000_0002h

Figure 30.7 Shared settings for PTP devices

30.3.7 Operation as an E2E Master

30.3.7.1 Preparatory setting

Table 30.17 lists the registers for use in operation as an E2E master. When the EPTPC operates as an OC, set the initial value of the time information in advance. See section 30.2.18, Local Clock Counter Initial Value Register (LCIVRU, LCIVRM, LCIVRL) for this value. To reflect the value set in these registers, you must set the SYRVLDR.STUP or ANUP bit to 1.

Table 30.17 Registers used in E2E master operation (1 of 2)

Register name	SYRVLDR register bits used for loading direction	Settings	Description
SYNFR	STUP	0000_0000h	flagField for Sync messages
SYTLIR	STUP ANUP	Example: 0000_0001h	Delay_Resp: 1 s Sync: 1 s Announce: 2 s
ANFR	ANUP	0000_0000h	flagField for Announce messages
GMPR	ANUP	As wanted	—

Table 30.17 Registers used in E2E master operation (2 of 2)

Register name	SYRVLDR register bits used for loading direction	Settings	Description
GMCQR	ANUP	As wanted	—
GMIDRU, GMIDRL	ANUP	As wanted	—
CUOTSR	ANUP	As wanted	timeSource: Internal Oscillator
SRR	ANUP	As wanted	• If the EPTPC operates as a master, set this register to 0000_0000h • If the EPTPC operates as a slave, set this register to the StepsRemoved field value of Announce messages received by the slave plus one
SYRFL1R	STUP	0000_4001h	Enables the processing of Delay_Req messages by the SYNFP module
SYRFL2R	STUP	0000_0011h	Enables the transfer of Signaling and Management messages to the PTPEDMAC
SYTRENR	STUP	0000_0011h	Enables the transmission of Sync and Announce messages

30.3.7.2 Procedure for starting operations

Figure 30.8 shows the procedure for settings to start operation as an E2E master.

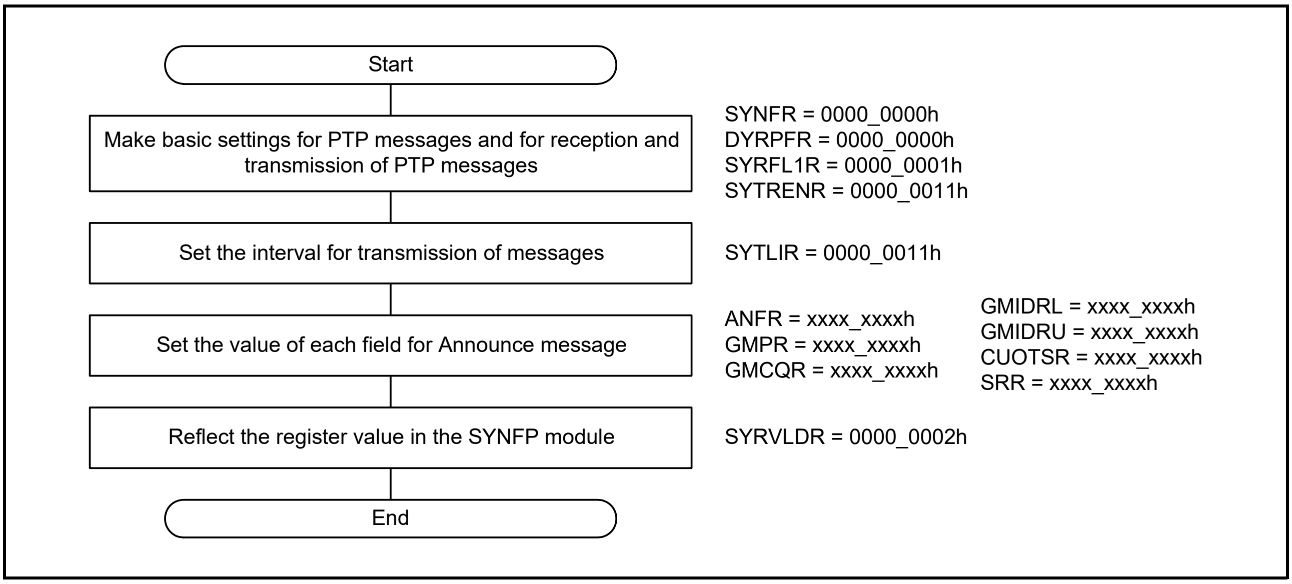


Figure 30.8 Procedure for starting operation as an E2E master

30.3.7.3 Procedure for changing the settings

Increases in the frequency of receiving Delay_Req messages caused by network conditions might lead to an overflow of the FIFO buffer that receives the Delay_Req messages. In such cases, change the value of the logMessageInterval field of Delay_Resp messages so that the slave sending the Delay_Req messages lengthens the interval between the messages. Figure 30.9 shows the procedure for changing the value of the logMessageInterval field.

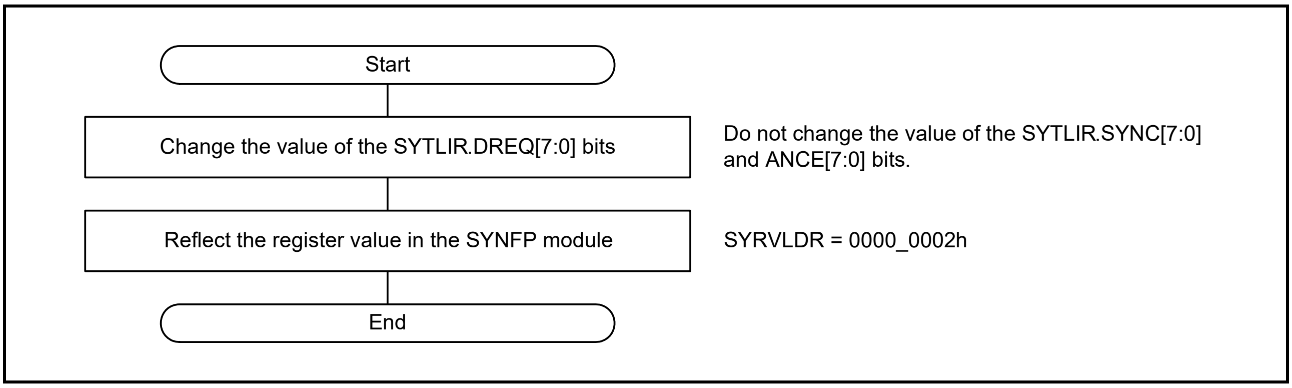


Figure 30.9 **Procedure for changing the value of the logMessageInterval Field for Delay_Resp messages**

30.3.7.4 Procedure for stopping operations

Figure 30.10 shows the procedure for stopping operation as an E2E master. To confirm that the operation is completely stopped, read the SYSR.GENDN and RESDN flags to check that generation of messages and sending of responses are completely stopped.

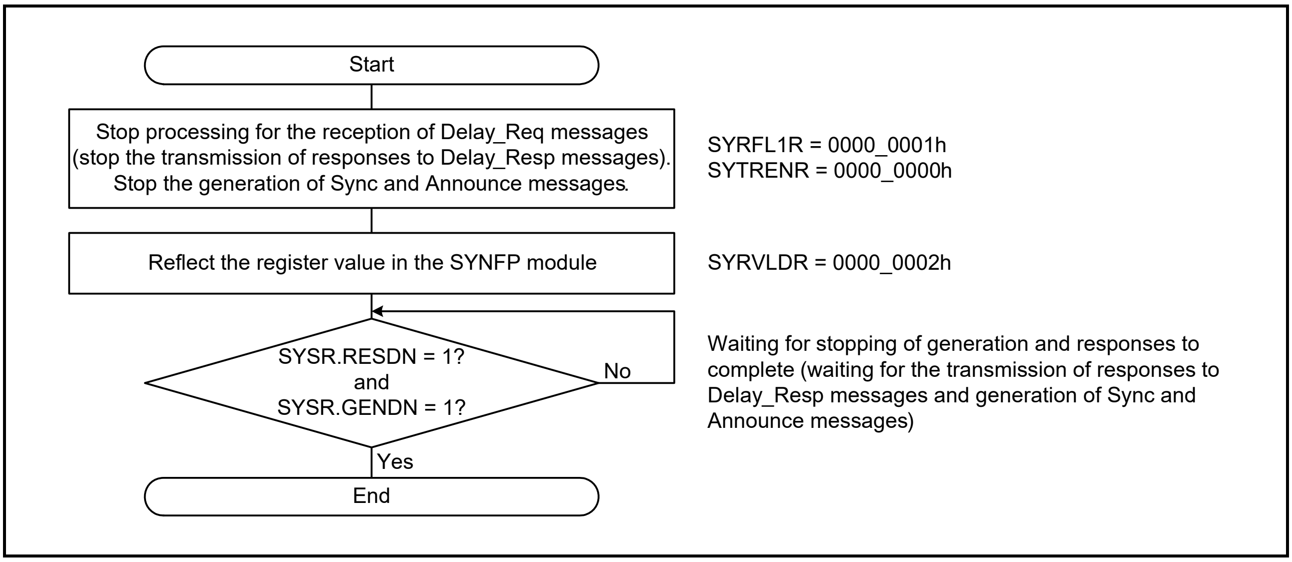


Figure 30.10 **Procedure for stopping operation as an E2E master**

30.3.8 Operation as an E2E Slave

30.3.8.1 Preparatory settings

Table 30.18 lists the registers for use in operation as an E2E slave. To reflect the value set in the register in SYNFP operations, you must set the SYRVLDR.STUP, ANUP, or BMUP bit to 1.

Table 30.18 **Registers used in E2E slave operation (1 of 2)**

Register name	SYRVLDR register bits used for loading direction	Settings	Description
MTCID	BMUP	As wanted	clockIdentity value of the master clock that provides synchronization
MTPID	BMUP	As wanted	portNumber value of the master clock that provides synchronization
SYTLIR	ANUP BMUP	Example: 0000_0000h	Delay_Resp: 1 s*1

Table 30.18 Registers used in E2E slave operation (2 of 2)

Register name	SYRVLDR register bits used for loading direction	Settings	Description
RSTOUTR	STUP	As wanted	—
SYNTOR	—	As wanted	—
SYRFL1R	STUP	0004_0441h	Enables reception of Delay_Resp, Follow_Up, and Sync messages and transfer of Announce messages to the PTPEDMAC
SYRFL2R	STUP	0000_0011h	Enables transfer of Signaling and Management messages to the PTPEDMAC
SYTRENR	STUP	0000_0100h	Enables the generation of Delay_Req messages

Note 1. During the reception of Delay_Resp messages by an E2E slave, the SYTLIR.DREQ[7:0] bits must be adjusted if the value of the SYRLIR.DRESP[7:0] flags is to be altered. The SYTLIR.DREQ[7:0] bits specify a value in the range from -7 to +6. Set the SYTLIR.DREQ[7:0] bits to -7 if the value indicated in the SYRLIR.DRESP[7:0] flags is less than or equal to -8 and to 6 if the value indicated in the SYRLIR.DRESP[7:0] flags is greater than or equal to 7.

30.3.8.2 Procedure for starting operations

Figure 30.11 shows the procedure for settings to start operation as an E2E slave.

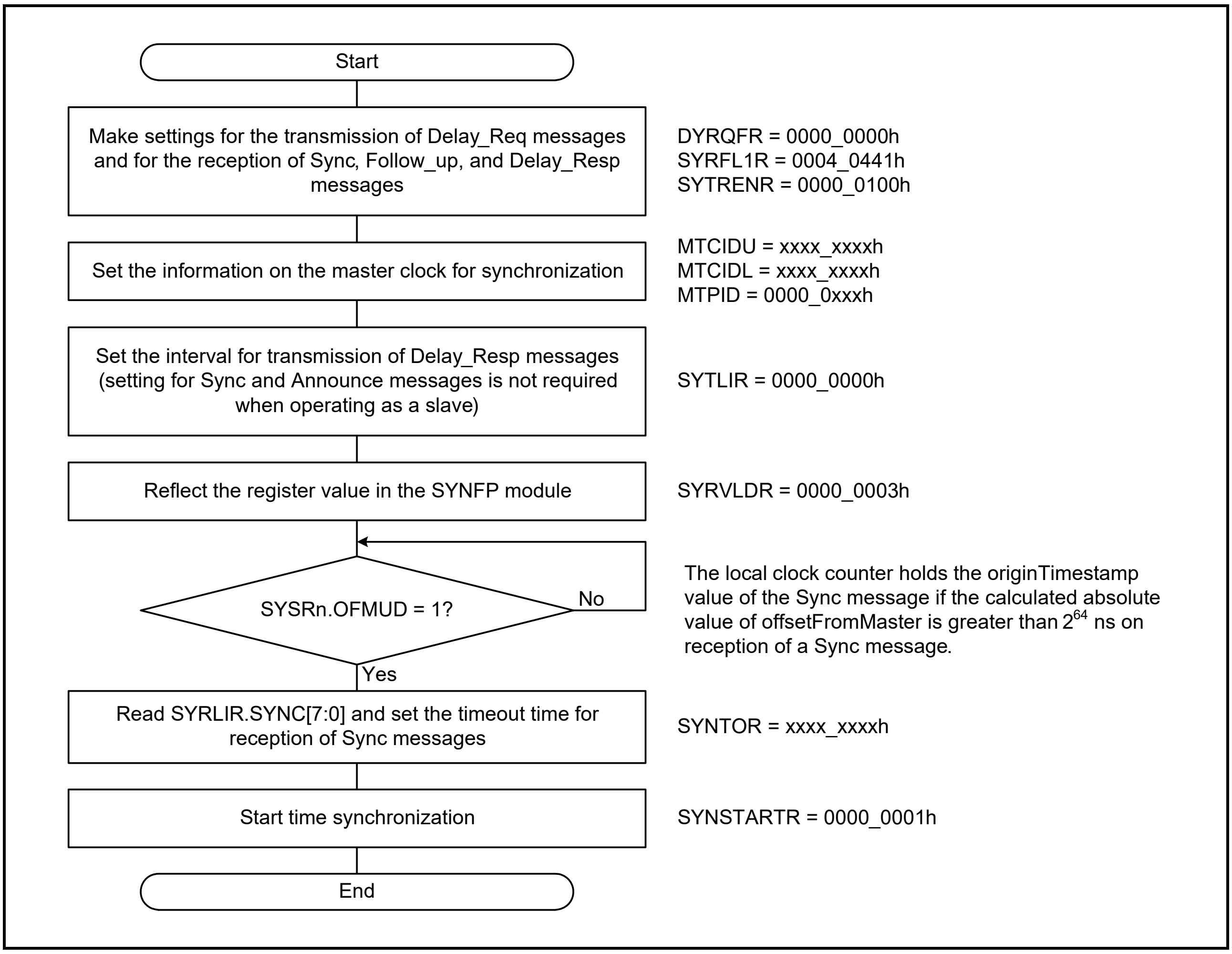


Figure 30.11 Procedure for starting operation as an E2E slave

30.3.8.3 Procedure for changing the settings

IEEE 1588 stipulates that the average interval for the transmission of Delay_Req messages must be adjusted in response to changes in the value of the logMessageInterval field of received Delay_Resp messages. The EPTPC sets the SYSR.INTCHG flag to 1 if the logMessageInterval value of a received message differs from that of the previous message. When this happens, the application must set the SYTLIR.DREQ[7:0] bits to the value in the SYRLIR.DRESP[7:0] bits. The SYTLIR.DREQ[7:0] bits specify a value in the range from -7 to +6. Set the SYTLIR.DREQ[7:0] bits to -7 if the value indicated in the SYRLIR.DRESP[7:0] flags is less than or equal to -8 and to 6 if the value indicated in the SYRLIR.DRESP[7:0] bits is greater than or equal to 7.

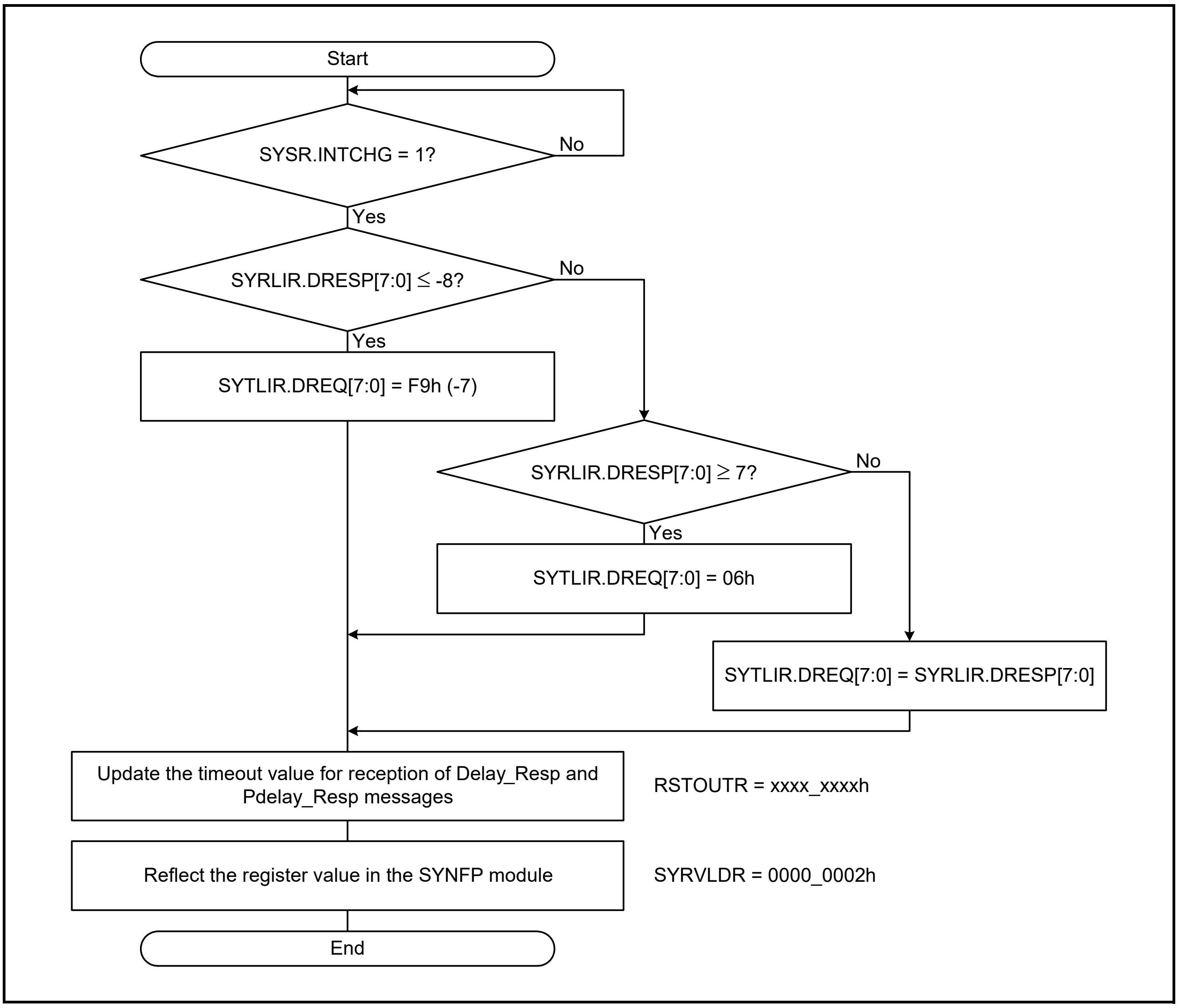


Figure 30.12 Procedure for changing the transmission interval for Delay_Req messages

30.3.8.4 Procedure for stopping operations

Figure 30.13 shows the procedure for stopping operation as an E2E slave. To confirm that operation as an E2E slave is completely stopped, read the SYSR.GENDN flag to check that generation is completely stopped.

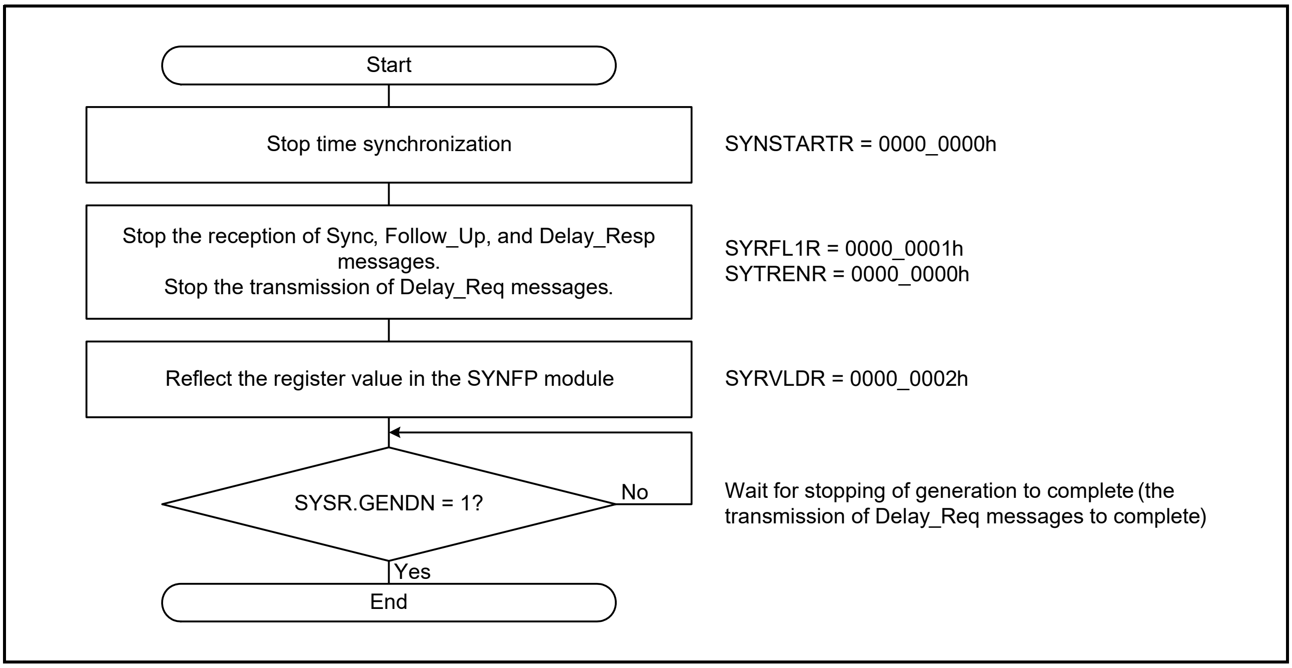


Figure 30.13 Procedure for stopping operation as an E2E slave

30.3.9 P2P Operation (Shared by Master and Slave)

Table 30.19 lists the registers for use in P2P operation. When the EPTPC is to be operated with P2P protocol, the SYNFP module handles the processing of PTP-pdelay messages regardless of whether operation is as a master or slave. The interval for Pdelay_Req transmission and the parameters for monitoring of Pdelay_Resp messages must be set at the same time.

Table 30.19 Registers for use in P2P operation

Register name	SYRVLDR register bits used for loading direction	Settings	Description
MTCID	BMUP	As wanted	clockIdentity value of the synchronized master clock
MTPID	BMUP	As wanted	portNumber value of the synchronized master clock
SYTLIR	ANUP STUP	0000_0000h	Announce: — Sync: — Pdelay_Req: 1 s
RSTOUTR	STUP	As wanted	—
SYRFL1R	STUP	4440_0001h	Enables the reception of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages and the transfer of Announce messages to the PTPEDMAC
SYRFL2R	STUP	0000_0011h	Enables the transfer of Signaling and Management messages to the PTPEDMAC
SYTRENR	STUP	0000_1000h	Enables the generation of Pdelay_Req messages

30.3.9.1 Procedure for starting operations

Figure 30.14 shows the procedure for starting P2P operation (sending and receiving PTP-pdelay messages).

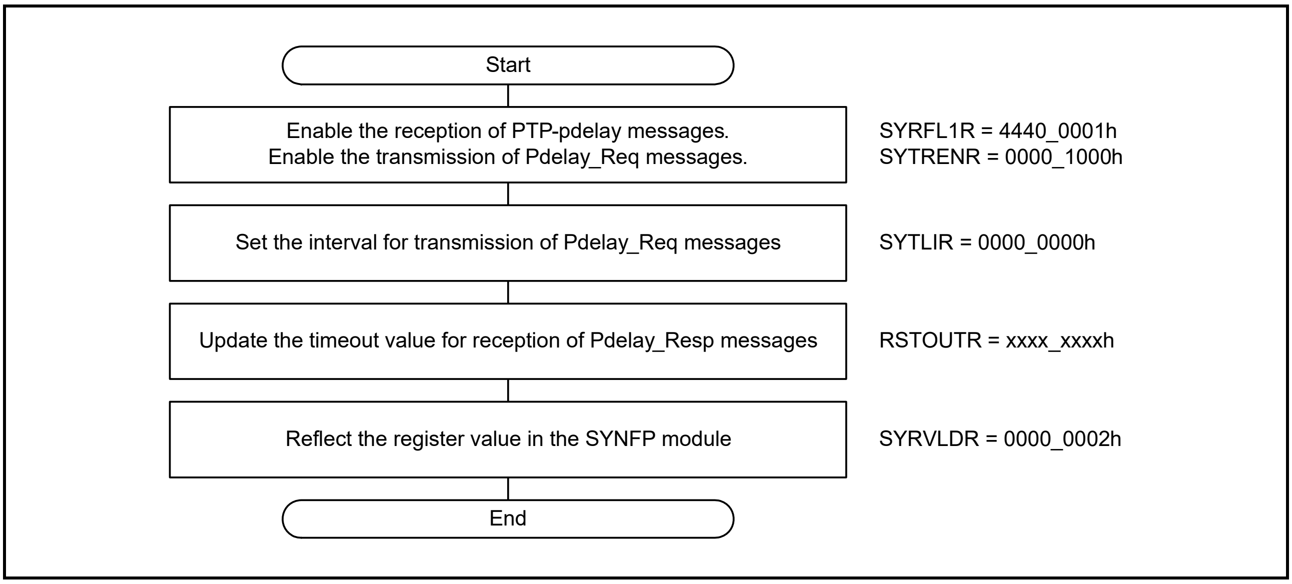


Figure 30.14 Procedure for starting P2P operation

30.3.9.2 Procedure for stopping operations

Figure 30.15 shows the procedure for stopping P2P operation (sending and receiving PTP-pdelay messages).

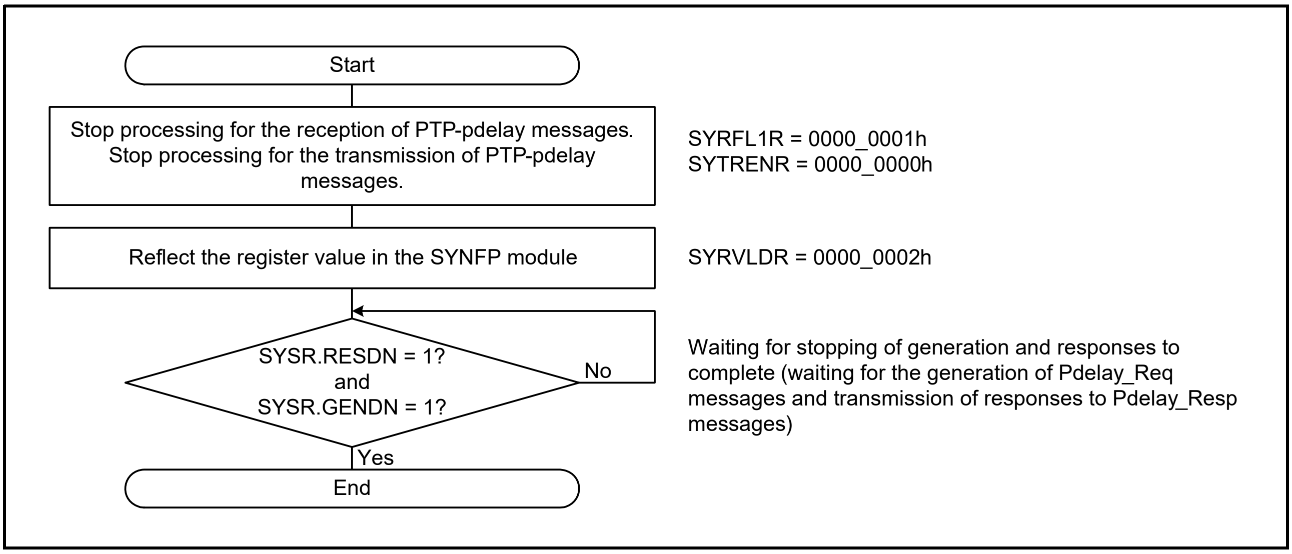


Figure 30.15 Procedure for stopping P2P operation

30.3.10 Operation as a P2P Master

Table 30.20 lists the registers for use in operation as a P2P master. When the EPTPC operates as an OC or BC using both ports as masters, set the initial value of the time information in advance as required. See section 30.2.18, Local Clock Counter Initial Value Register (LCIVRU, LCIVRM, LCIVRL) for this value.

Table 30.20 Registers used in P2P master operation (1 of 2)

Register name	SYRVLDR register bits used for loading direction	Settings	Description
SYCONFR	—	0000_0028h	—
ANFR	ANUP	0000_0000h	flagField for Announce messages
SYNFR	STUP	0000_0000h	flagField for Sync messages

Table 30.20 Registers used in P2P master operation (2 of 2)

Register name	SYRVLDR register bits used for loading direction	Settings	Description
SYTLIR	ANUP STUP	Example: 0000_0001h	Announce: 2 s Sync: 1 s Pdelay_Req: 1 s
GMPR	ANUP	As wanted	Grandmaster Priority1 and Priority2
GMCQR	ANUP	As wanted	Grandmaster Quality
GMIDR	ANUP	As wanted	Grandmaster Identity
CUOTSR	ANUP	As wanted	currentUtcOffset, timeSource
SRR	ANUP	As wanted	StepsRemoved
RSTOUTR	STUP	As wanted	—
SYRFL1R	STUP	4440_0000h	Enables the reception of Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages
SYRFL2R	STUP	0000_0011h	Enables the transfer of Signaling and Management messages to the PTPEDMAC
SYTRENR	STUP	0000_1011h	Enables the transmission of Pdelay_Req, Sync, and Announce messages

30.3.10.1 Procedure for starting operations

When transmission of Sync and Announce messages is started during P2P operation (sending and receiving PTP-pdelay messages), the EPTPC operates as a P2P master. Figure 30.16 shows the procedure for starting operation as a P2P master.

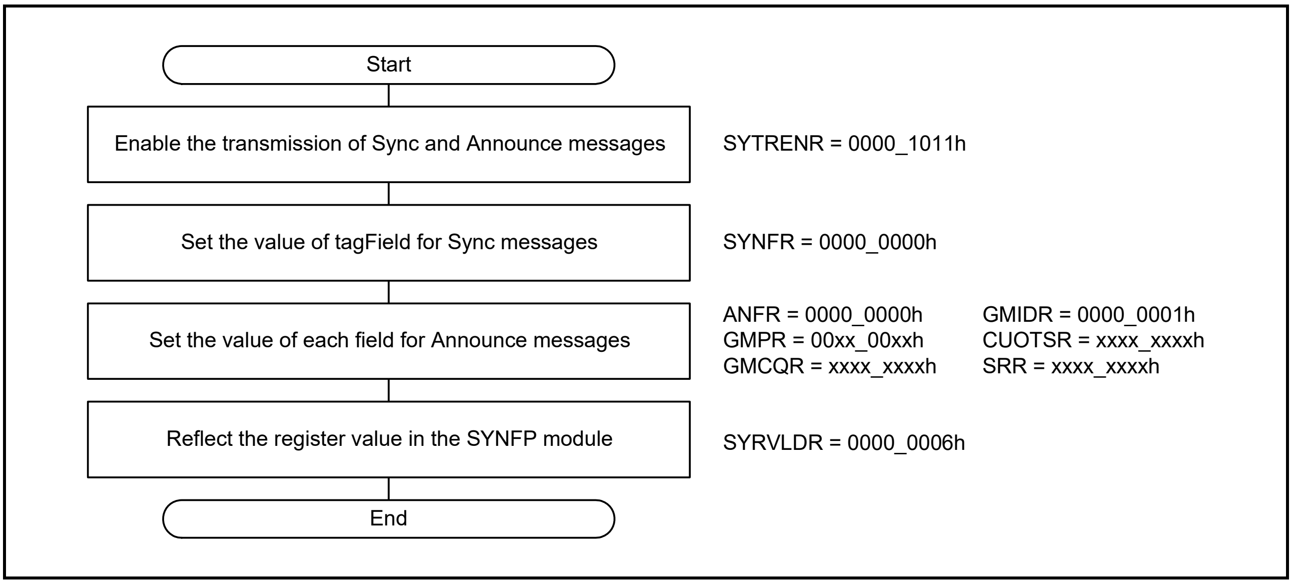


Figure 30.16 Procedure for starting operation as a P2P master

30.3.10.2 Procedure for stopping operations

Figure 30.17 shows the procedure for stopping the transmission of Sync and Announce messages to stop operation as a P2P master.

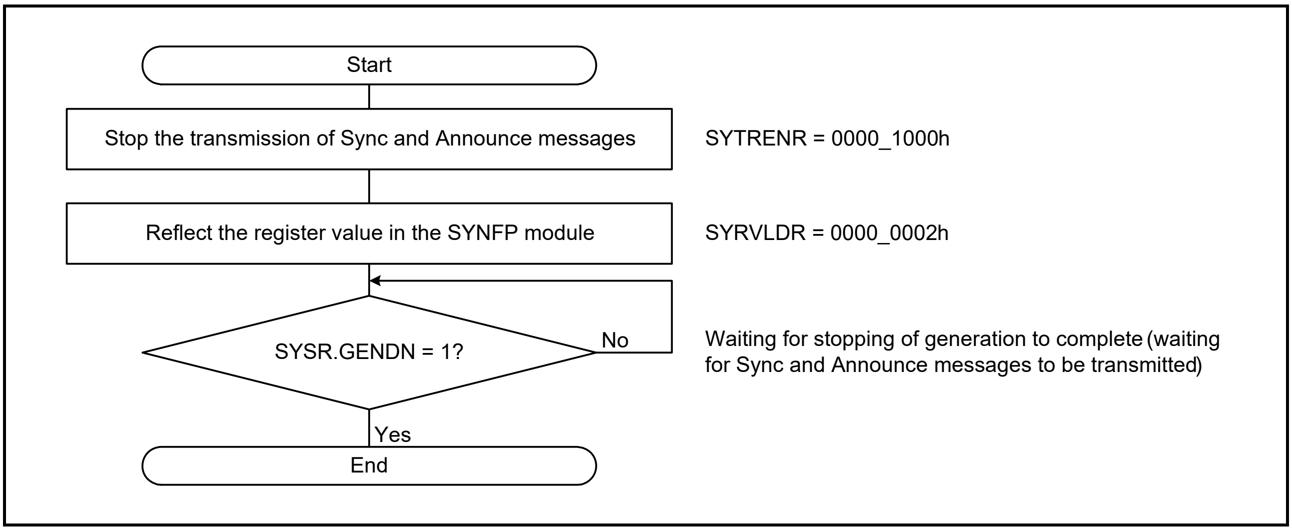


Figure 30.17 Procedure for stopping operation as a P2P master

30.3.11 Operation as a P2P Slave

Table 30.21 lists the registers for use in operation as a P2P slave. Setting a SYNFP module to receive Sync messages and Follow_Up messages during P2P operation results in operation as a P2P slave. Information on the master clock for synchronization must be specified.

Table 30.21 Registers used in P2P slave operation

Register name	SYRVLDR register bits used for loading direction	Settings	Description
MTCID	BMUP	As wanted	clockIdentity value of the synchronized master clock
MTPID	BMUP	As wanted	portNumber value of the synchronized master clock
RSTOUTR	STUP	As wanted	—
SYRFL1R	STUP	4440_0441h	Enables the reception of Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up, Follow_Up, and Sync messages and the transfer of Announce messages to the PTPEDMAC

30.3.11.1 Procedure for starting operations

Figure 30.18 shows the procedure for making the additional settings for shifting to slave operation during P2P operation (sending and receiving PTP-pdelay messages).

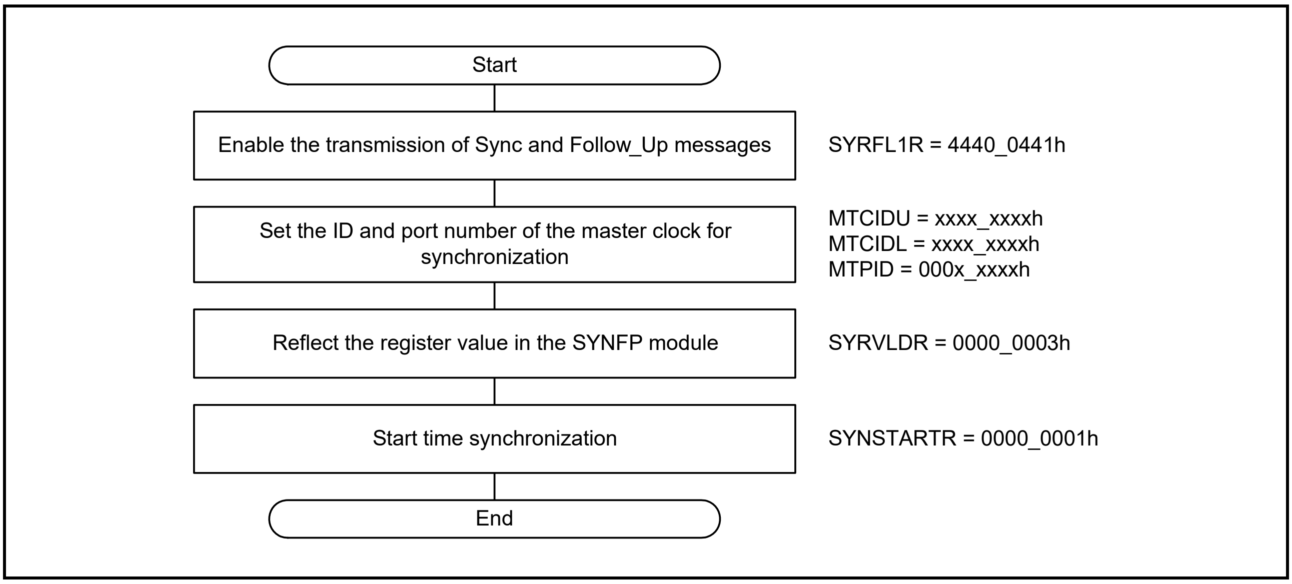


Figure 30.18 Procedure for starting operation as a P2P slave

30.3.11.2 Procedure for stopping operations

Figure 30.19 shows the procedure for stopping the reception of Sync and Follow_Up messages to stop operation as a P2P slave.

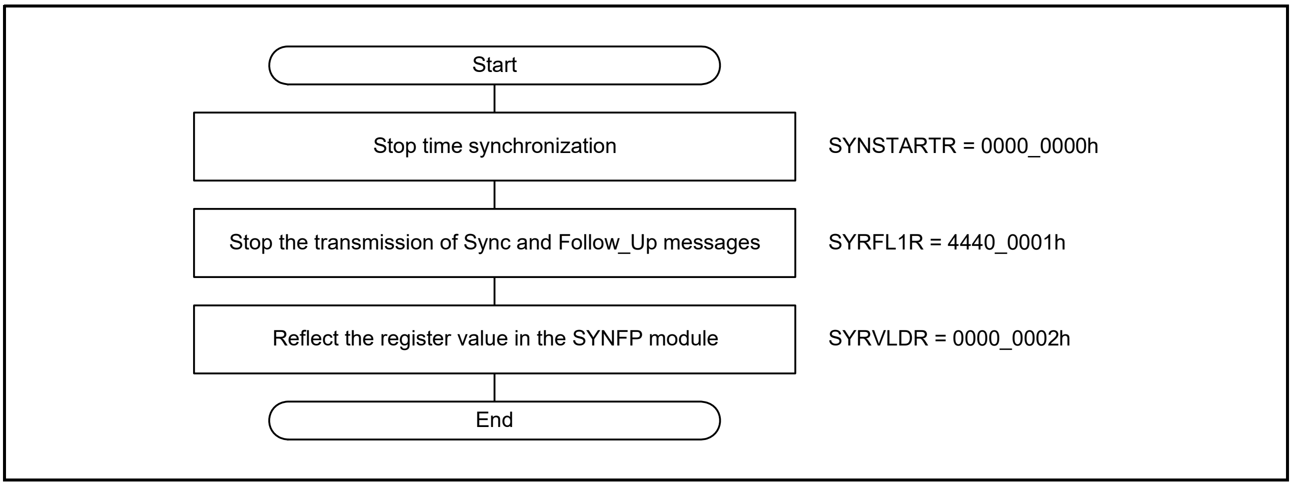


Figure 30.19 Procedure for stopping operation as a P2P slave

30.3.12 Monitoring of Received Messages

30.3.12.1 Reception of announce messages

The EPTPC does not detect timeouts during the reception of Announce messages. To detect timeouts, monitor the reception of Announce messages by software.

30.3.12.2 Reception of sync messages

The STSR.SYNTOUT flag is set to 1 when a timeout occurs during the reception of a Sync message while correcting time synchronization.

The SYSR.OFMUD flag is set to 1 when a Sync message is received, regardless of whether time synchronization is being corrected. Accordingly, the reception of Sync messages is detectable by referencing this flag even when the correction of time synchronization stops because a timeout occurs during the reception of a Sync message.

30.3.12.3 Reception of Delay_Resp and Pdelay_Resp messages

The SYSR.DRPTO flag is set to 1 when a timeout occurs during the reception of a Delay_Resp message after the transmission of a Delay_Req message while operating as an E2E slave, or when a timeout occurs during the reception of a Pdelay_Resp message after the transmission of a Pdelay_Req message while operating as a P2P.

The SYSR.MPDUD flag is set to 1 when a Delay_Resp or Pdelay_Resp message is received, so the reception of these messages is still detectable when a timeout occurs during reception.

30.3.13 Correcting Time Synchronization

A slave detects differences in the clock gradient relative to the master clock. The offsetFromMaster values calculated using the standard IEEE 1588 algorithm are used to calculate the clock gradient, so the result includes elements of network fluctuation that are not frequency differences. The EPTPC has a worst-10 function to eliminate fluctuations caused by network load and other dynamic conditions. With these functions, the time is corrected from the calculated gradient difference values and results of correction are obtained as shown in Figure 30.21.

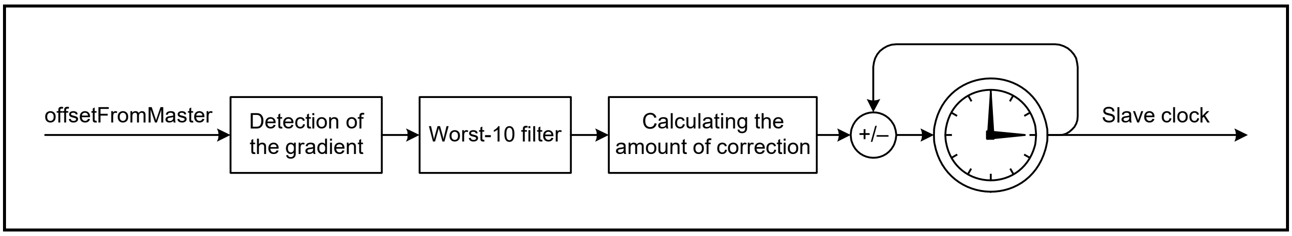


Figure 30.20 Configuration of the time correction circuit

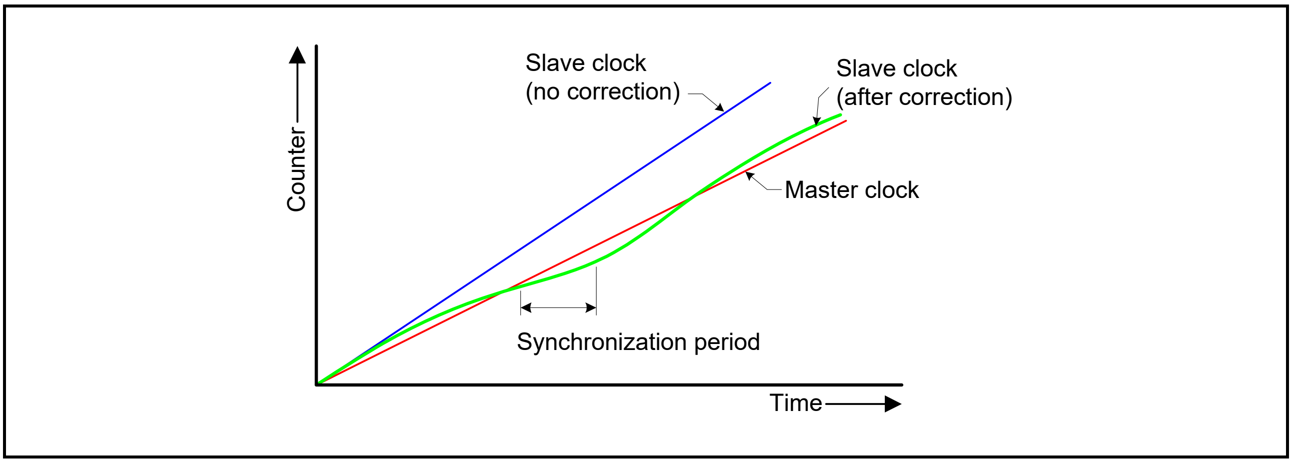


Figure 30.21 Overview of time correction

30.3.13.1 Determining synchronization and loss of synchronization

Loss of synchronization is detected if the absolute value of offsetFromMaster reaches or exceeds the value specified in the SYNTDARU or SYNTDARL register. Synchronization is considered maintained if the absolute value of offsetFromMaster is less than the absolute values of the synchronization detection threshold registers, SYNTDBRU and SYNTDBRL.

The STSR.SYNCOUT flag is set to 1 when synchronization is lost, and the SYNC flags is set to 1 when synchronization is obtained. Hysteresis can be obtained by setting the threshold registers to appropriate different values. In addition, the STMR.DVTH[3:0] and SYTH[3:0] bits can be used to set the consecutive number of times detection must occur for the determination of synchronization and loss of synchronization.

For systems in which control must be aborted if synchronization is lost because of fluctuations in network conditions, set the SYNTDARU and SYNTDARL registers to low values and set the number of times detection is required to trigger a loss of synchronization to one. In systems where these conditions do not apply, set the SYNTDARU and SYNTDARL registers and the number of times detection is required to large values.

Figure 30.22 shows an example of a situation where synchronization is lost and regained. In this example, the number of consecutive times detection is required is three for both synchronization and loss of synchronization.

Note: The setting of the STSR.SYNCOUT flag is 1 when time synchronization starts, even if the condition for determining loss of synchronization is not satisfied at this stage. For this reason, detection of loss of synchronization must be ignored immediately after time synchronization starts.

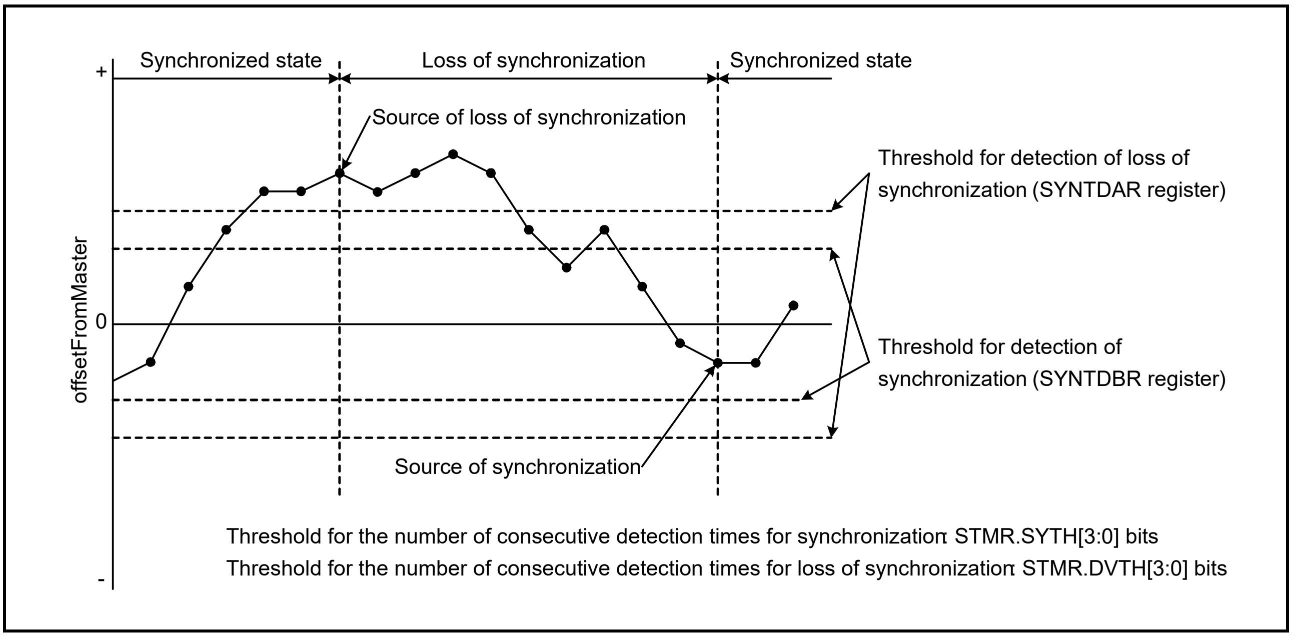


Figure 30.22 **Example of a situation where synchronization is lost and regained, when the number of consecutive detections is set to three in the STMR.DVTH[3:0] and SYTH[3:0] bits**

30.3.13.2 Worst-10 function

The worst-10 function is used to impose limits on exceedingly large and small values among the calculated values for clock gradient differences. These values are collected by observing the transfer over a specified interval, and threshold values to impose limits are extracted from the observed values. Fluctuations in network conditions must be considered in addition to clock errors, and differences in both the positive and negative directions are collected, as shown in Figure 30.23.

The function selects the largest gradient values from the collected values for positive and negative gradient differences, orders them from first to tenth (worst to tenth worst), and uses the tenth worst as a threshold value. Fluctuations in the time kept by a slave clock can be suppressed by continually overwriting the tenth worst value with new values large enough to exceed the threshold. Periodic collections of gradient values can also be made for updating the threshold values during operations or for using the method of setting threshold values from previously measured results.

However, while fluctuations in the time kept by a slave clock can be suppressed by the valid filtering of values for gradient difference (collecting the worst 10 values and using the tenth worst), this slows down the following of time kept by a master clock.

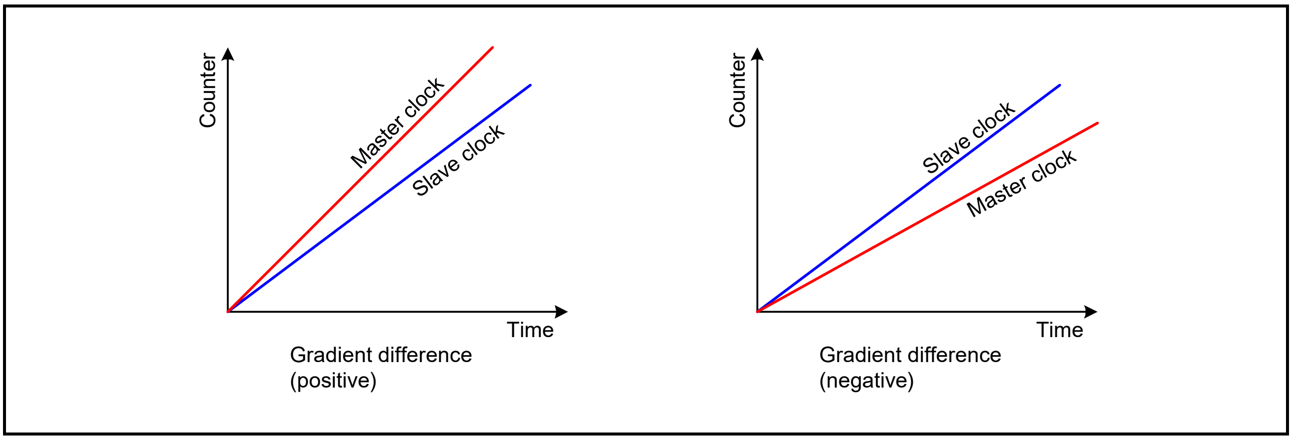


Figure 30.23 Overview of gradient differences

30.3.13.3 Collecting differences in clock gradient and extracting the worst ten values

During slave operation, the EPTPC can calculate the offsetFromMaster values from received messages and calculate gradient differences between the local clock (acting as a slave clock) and master clock from those values. Specifically, the worst ten values are extracted from the sets of collected values for gradient difference. Either automatic filtering by the hardware or software-triggered filtering can be designated for acquisition of the sets of the worst 10 values. Figure 30.24 gives an overview of the collection of gradient difference values.

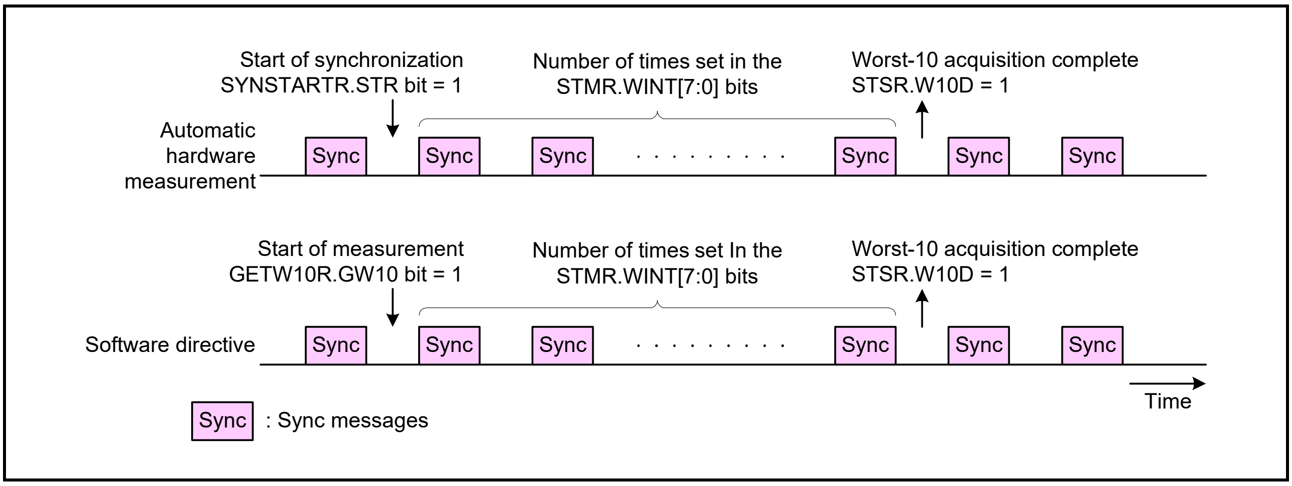


Figure 30.24 Overview of the collection of gradient difference values

(1) Collecting gradient differences and extracting the worst ten values by hardware

The EPTPC automatically collects the gradient difference values by hardware if the STMR.W10S bit is 0.

When the SYNSTARTR.STR bit is set to 1 (starting slave time synchronization), the EPTPC collects gradient difference values for the number of times set in the STMR.WINT[7:0] bits. When the collection of gradient difference values is finished, the tenth largest values on the positive and negative sides are stored as the tenth worst values in the PW10VRU, PW10VRM, and PW10VRL registers and the MW10RU, MW10RM, and MW10RL registers. When acquisition of the worst 10 values completes, the STSR.W10D flag sets to 1. Filtering of gradient difference values by using the stored tenth worst values then proceeds automatically.

If the number of times set in the STMR.WINT[7:0] bits is less than ten, the double of the best of the collected values on the positive side is stored in the PW10VRU, PW10VRM, and PW10VRL registers. The half of the best of the collected values on the negative side is stored in the MW10RU, MW10RM, and MW10RL registers.

(2) Collecting gradient differences and extracting the worst ten values by software

The EPTPC collects gradient difference values by software if the STMR.W10S bit is 1.

When the GETW10R.GW10 bit is set to 1 after time synchronization starts, the EPTPC collects gradient difference values for the number of times set in the STMR.WINT[7:0] bits. When the collection of gradient difference values is finished, the tenth largest values on the positive and negative sides are stored as the tenth worst values in the PW10VRU, PW10VRM, and PW10VRL registers and the MW10RU, MW10RM, and MW10RL registers. When acquisition of the worst 10 values completes, the STSR.W10D flag is set to 1.

Because filtering of gradient difference values proceeds with the values set in the PLIMITRU, PLIMITRM, and PLIMITRL registers as the upper filtering limits and the MLIMITRU, MLIMITRM, and MLIMITRL registers as the lower filtering limit, you must write the values stored in the PW10VRU, PW10VRM, and PW10VRL registers to PLIMITRU, PLIMITRM, and PLIMITRL, and write the values stored in the MW10RU, MW10RM, and MW10RL registers to MLIMITRU, MLIMITRM, and MLIMITRL.

If the number of times set in the STMR.WINT[7:0] bits is less than ten, the double of the best of the collected values on the positive is stored in the PW10VRU, PW10VRM, and PW10VRL registers. The half of the best of the collected values on the negative side is stored in the MW10RU, MW10RM, and MW10RL registers.

The flow in Figure 30.25 shows an example of the procedure for software-triggered acquisition of the worst 10 values.

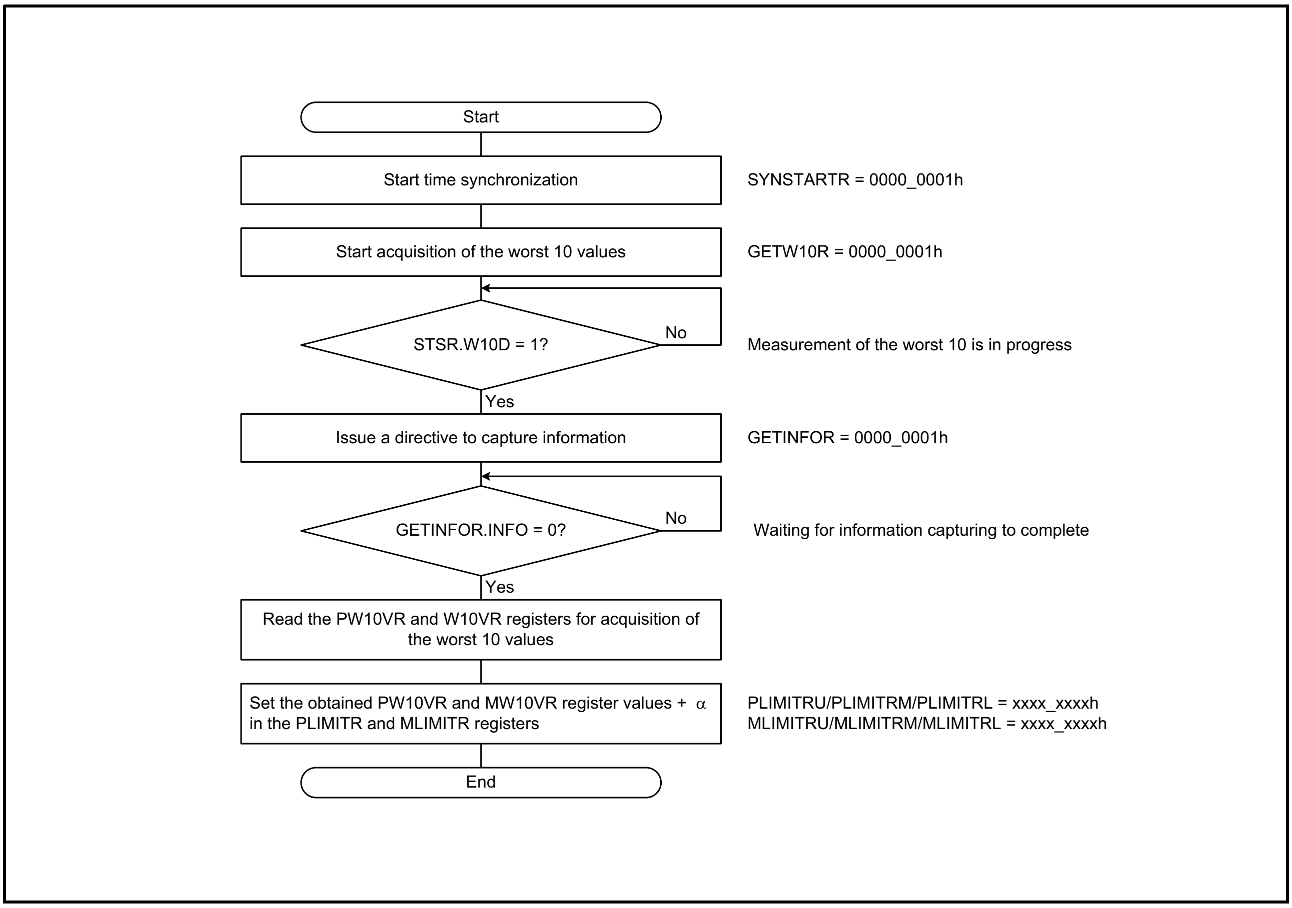


Figure 30.25 Example procedure for software-triggered acquisition of the worst 10 values

30.3.14 Local Clock Counter

The local clock counter retains the synchronized time information. The counter starts counting from 0 after the ETHERC is released from the module-stop state or the EPTPC is released from the software reset state. The local clock counter can then be set to any value. Figure 30.26 shows the procedure for setting the initial value in the local clock counter.

The time information kept by the local clock counter is also readable. Figure 30.27 shows the procedure for reading the time information kept by the local clock counter.

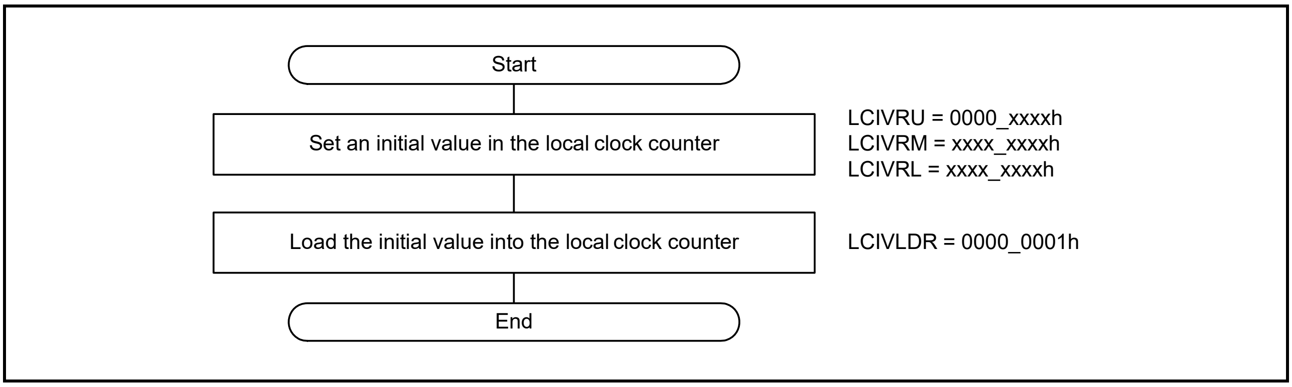


Figure 30.26 Procedure for setting a new initial value in the local clock counter

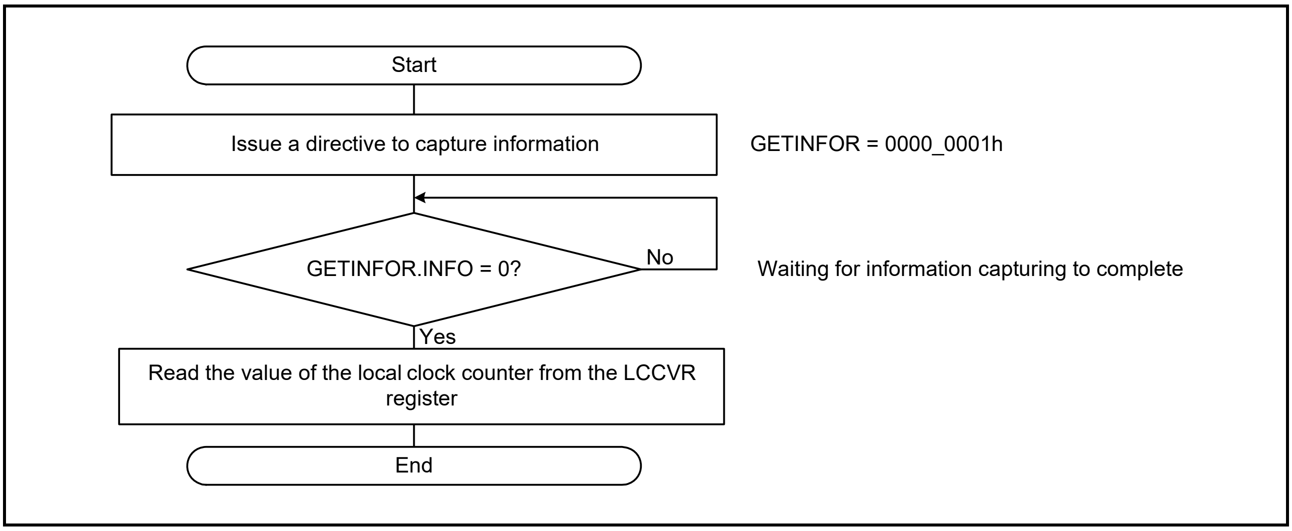


Figure 30.27 Procedure for reading the time kept by the local clock counter

30.3.15 Pulse Output Timer

The STCA module of the EPTPC incorporates six timers (pulse output timers 0 to 5) that operate independently of each other. The pulse output timers produce periodic pulses, and the rising or falling edges of these pulses can be used as interrupt requests or output to the ELC as event signals. The time at which a pulse output timer starts operating (t_{start}), and the period (t_c) and pulse width (t_w) of the output pulses, can be specified.

Figure 30.28 shows the timing of pulse output timer operation, and Table 30.22 lists the constraints on the settings.

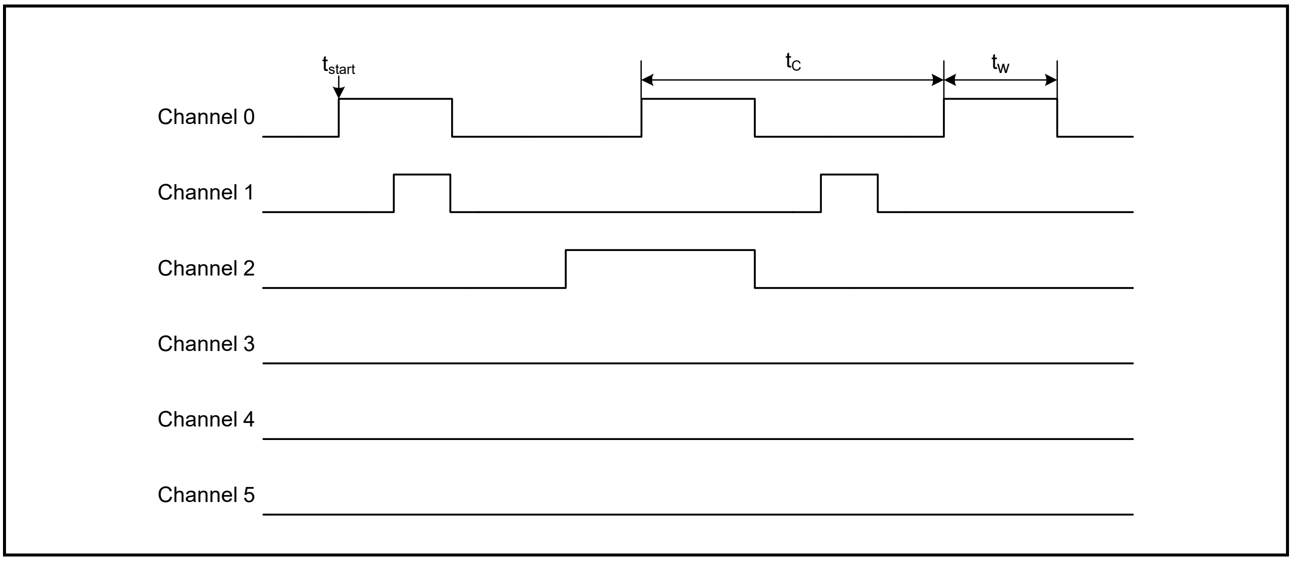


Figure 30.28 Time at which a pulse output timer starts operating

Table 30.22 Constraints on the values that can be specified for a pulse output timer (1 of 2)

Parameter	Constraints
Cycle (t_C)	From 4 cycles of the STCA clock to 1 s
Resolution of the cycle	Set in nanoseconds However, the timing of rising edges is rounded by the period of the system clock (50 ns, 40 ns, 20 ns, or 10 ns).
Pulse width (t_W)	From 2 cycles of the STCA clock to 500 ms

Table 30.22 Constraints on the values that can be specified for a pulse output timer (2 of 2)

Parameter	Constraints
Resolution of the pulse width	Set in nanoseconds However, the timing of falling edges is rounded by the period of the system clock (50 ns, 40 ns, 20 ns, or 10 ns).

30.3.15.1 Procedure for setting a pulse output timer

Figure 30.29 shows the procedure for setting a pulse output timer.

Note: A timer does not produce periodic pulses if the time set in the TMSTTRUm and TMSTTRLm registers (m = 0 to 5) has elapsed. Set the time for a pulse output timer to start at a later time than that when the timer is set.

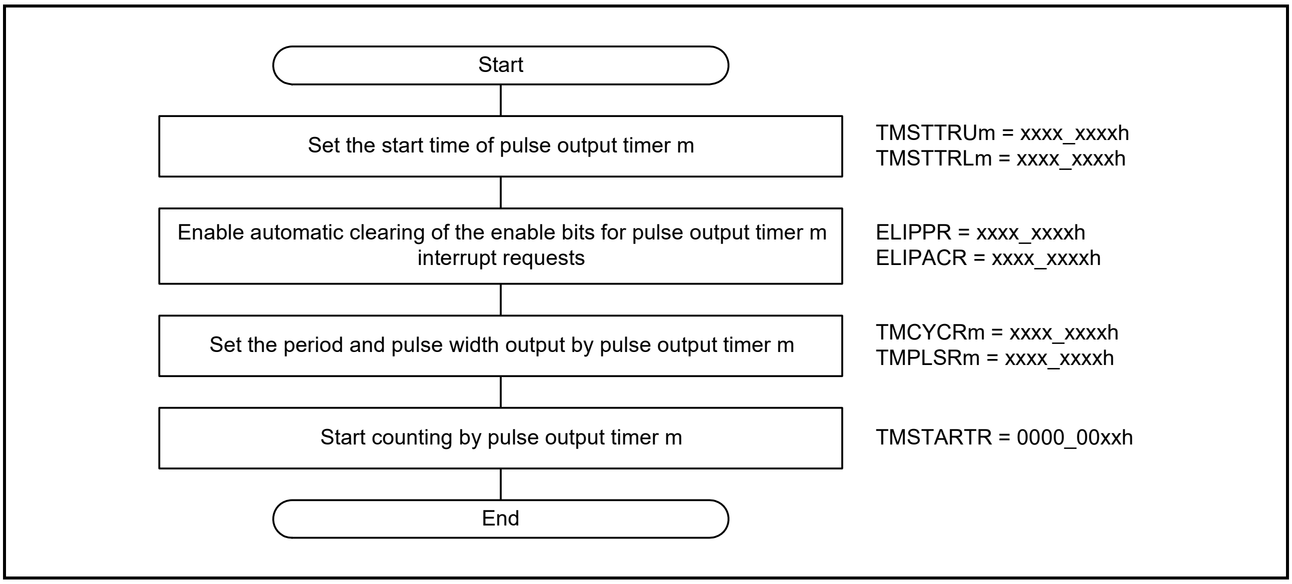


Figure 30.29 Procedure for setting a pulse output timer

30.3.15.2 Output of periodic pulses as interrupt requests or event signals

ETHER_MINT interrupt requests, ETHER_IPLS interrupt requests, or event output signals for the ELC can be generated on detection of rising or falling edges of the periodic pulses from the pulse output timer. The detection edge and the pulse output timer used are configurable, and automatic clearing of enable bits for the ETHER_IPLS interrupt or event output can be set. Make the required settings before setting the TMSTARTR.ENm bit to 1 (starting pulse output timer m).

(1) ETHER_MINT interrupt request

ETHER_MINT interrupt requests can be generated on rising edges of the periodic pulses from the pulse output timers. They cannot be generated on falling edges. Select the pulse output timers for generating these requests in the MITSELR.MINTENm bits. Automatic clearing of the enable bits for ETHER_MINT interrupt requests is not available.

(2) ETHER_IPLS interrupt request

ETHER_IPLS interrupt requests can be generated on either rising or falling edges of the periodic pulses from the pulse output timers. Select the pulse output timers for generating these requests in the IPTSELR.IPTSELm bits. Setting the ELIPACR.PLSP or PLSN bit enables automatic clearing of the enable bits for ETHER_IPLS interrupt requests.

(3) Output of event signals to the ELC

Event signals can be output to the ELC on either rising or falling edges of the periodic pulses from the pulse output timers. Select the pulse output timers for event signal output and the valid edge in the ELIPPR.CYCPm or CYCNm bits. Setting the ELIPACR.CYCPm or CYCNm bits enables automatic clearing of the event output enable bits.

30.3.16 Priority Control in Transmission

30.3.16.1 Arbitration

Contention between multiple requests for the transmission of messages by the SYNFP module are arbitrated in the order of priority shown in Table 30.23.

Table 30.23 Priority for message transmission arbitration

Transmission message	Priority order	Remark
Sync	1 Highest priority	—
Delay_Req, Pdelay_Req	2	There is no device type that simultaneously transmits Delay_Req and Pdelay_Req messages
Delay_Resp, Pdelay_Resp	3	There is no device type that simultaneously transmits Delay_Resp and Pdelay_Resp messages
Announce	4	—
Messages to be transmitted from the PTPEDMAC	5	—
Messages to be transmitted from the EDMAC0	6	—

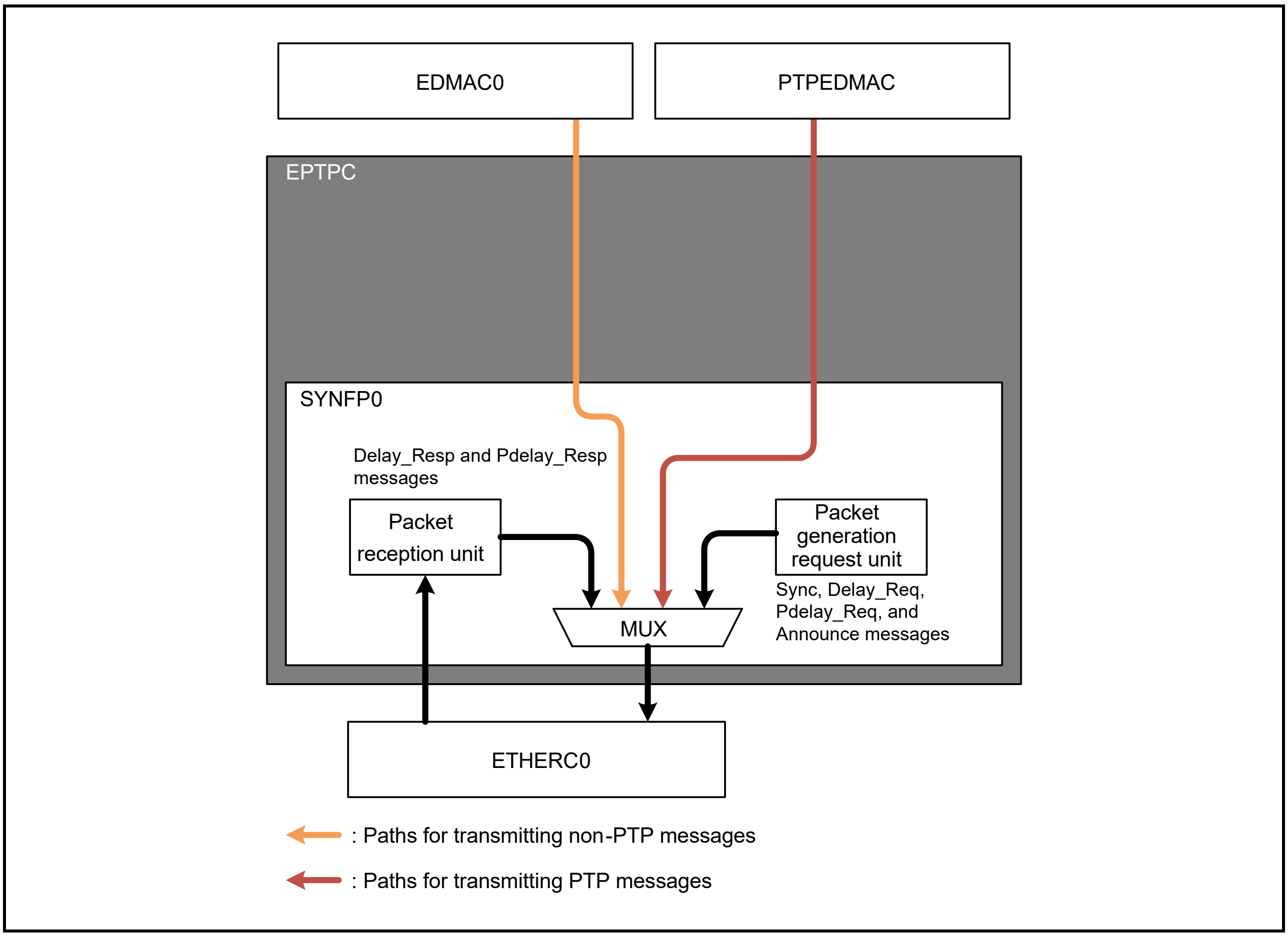


Figure 30.30 Arbitration in message transmission

30.3.16.2 Securing bandwidth for the transmission of sync messages

The EPTPC secures bandwidth for the transmission of Sync messages, and is capable of handling transmission at very precise intervals.

If the transmission of a Sync message at a fixed interval proceeds at the same time that transmission by the PTPEDMAC, because transmission of the Sync message proceeds when the other processing is complete, the interval for transmission is no longer fixed. Securing bandwidth for the transmission of Sync messages limits the transmission of messages from EDMAC0 and the PTPEDMAC, allowing Sync message transmission to be handled without fluctuations. To disable securing of bandwidth for Sync message transmission, set the SYCONFR.SBDIS bit to 1.

Figure 30.31 gives a schematic view of securing bandwidth for the transmission of Sync messages.

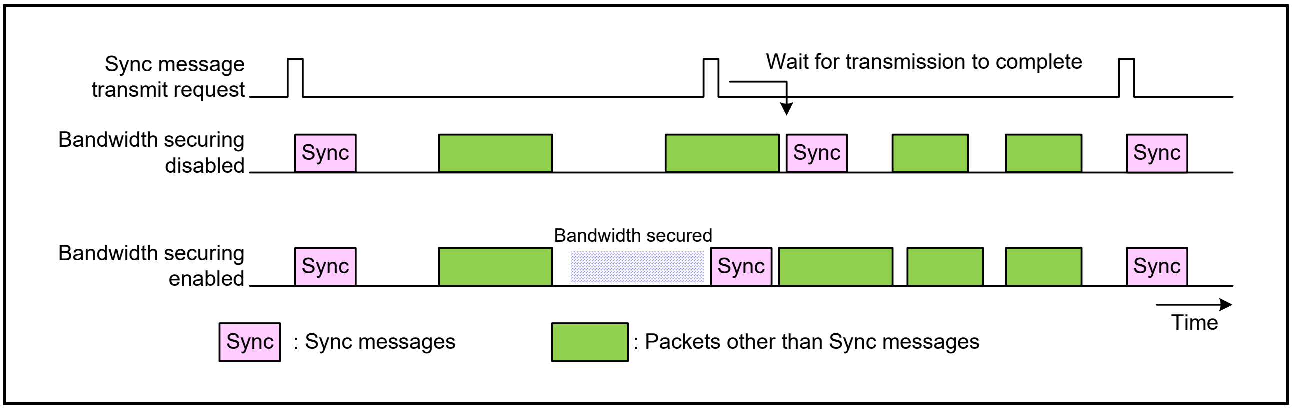


Figure 30.31 Securing of bandwidth for Sync message transmission

30.3.16.3 Securing of transmission interval

In the transmission of messages by the ETHERC, if there is a fixed delay from the time of a request for transmission to the time of transmission on the MII of Ethernet port 0, PTP message timestamps can be used for accurately obtaining the size of the delay during slave operation. However, for continuous transfer where processing of messages to wait for inter-packet gap times is required, delay times might fluctuate.

To enable the ETHERC to secure the reliability of timestamp values, specify an interval for frame transmission in the SYCONFR.TCYC[7:0] bits to control the interval between the completion of transmission and the next request for transmission. This avoids the effects of inter-packet gap times and a fixed delay for transmission.

30.4 Interrupts

The EPTPC provides the ETHER_MINT and ETHER_IPLS interrupt requests. Figure 30.32 shows the relationship between the two interrupt requests. Figure 30.33 shows the details on interrupt requests of the pulse output timer.

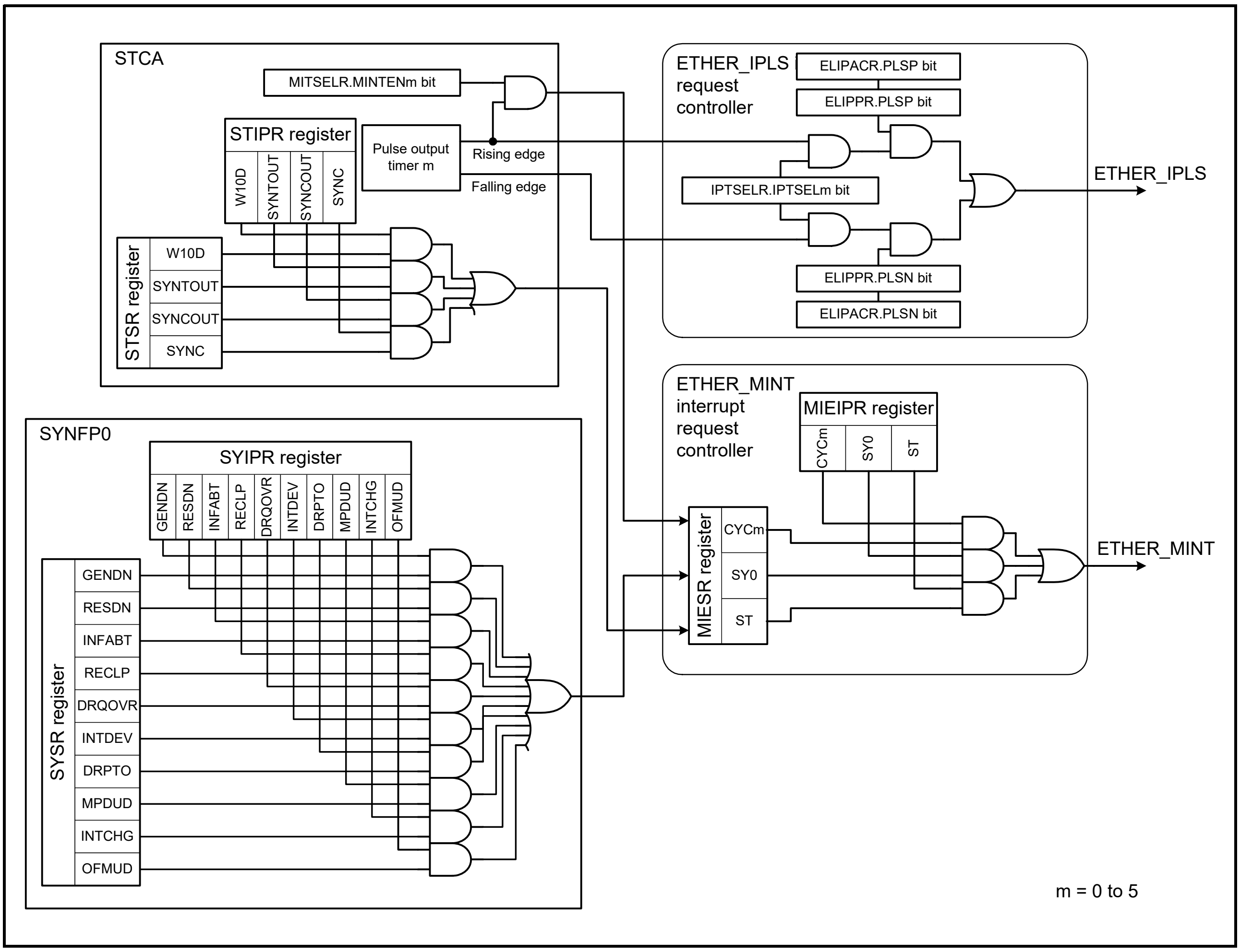


Figure 30.32 ETHER_MINT and ETHER_IPLS interrupt requests

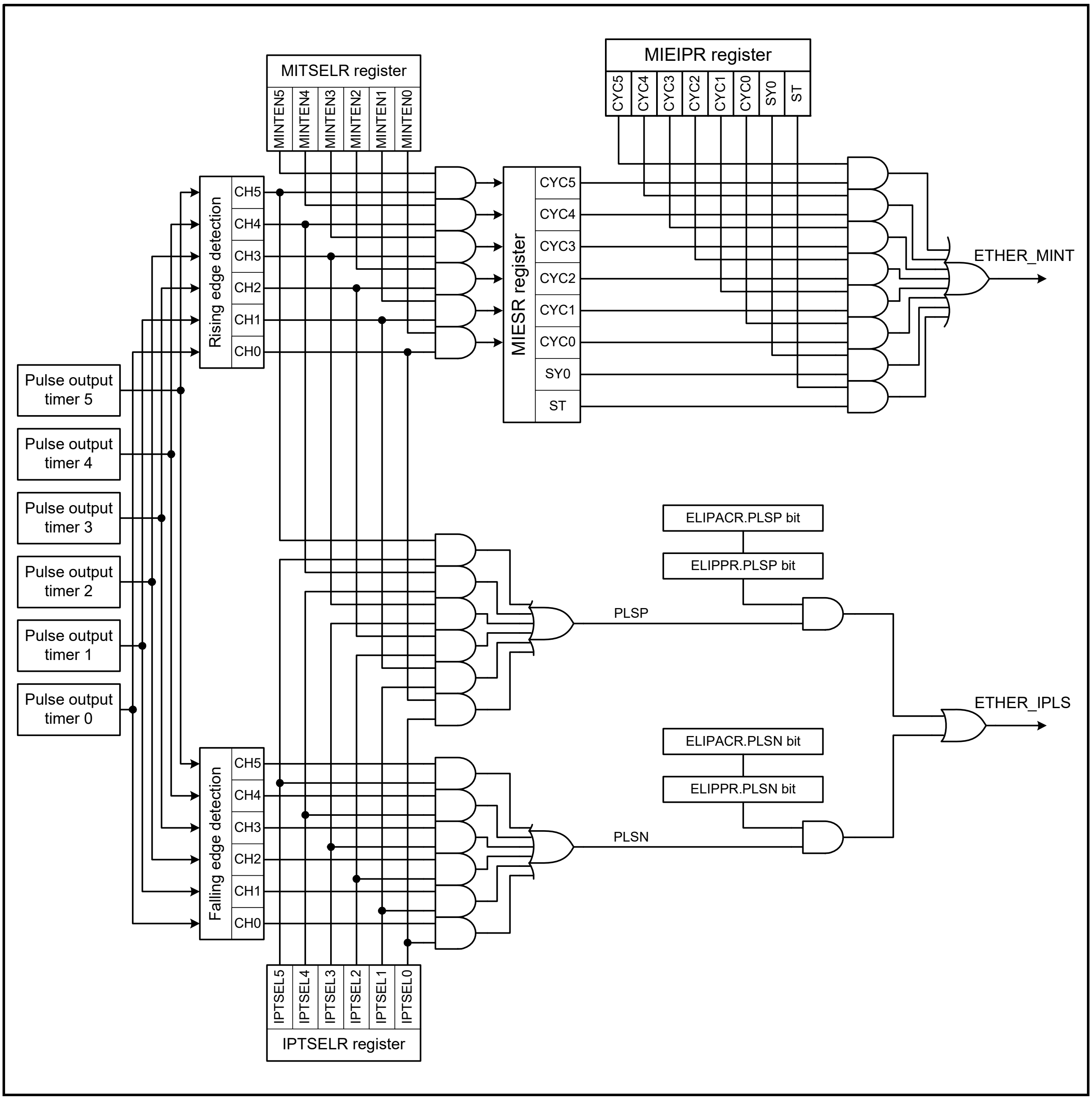


Figure 30.33 Details on interrupt requests of the pulse output timer

30.5 Event Link (Output)

The EPTPC can output an event to the ELC by detecting the rising or falling edge of the pulse from the pulse output timer. Figure 30.34 shows the relationship between the pulse output timer and the ELC.

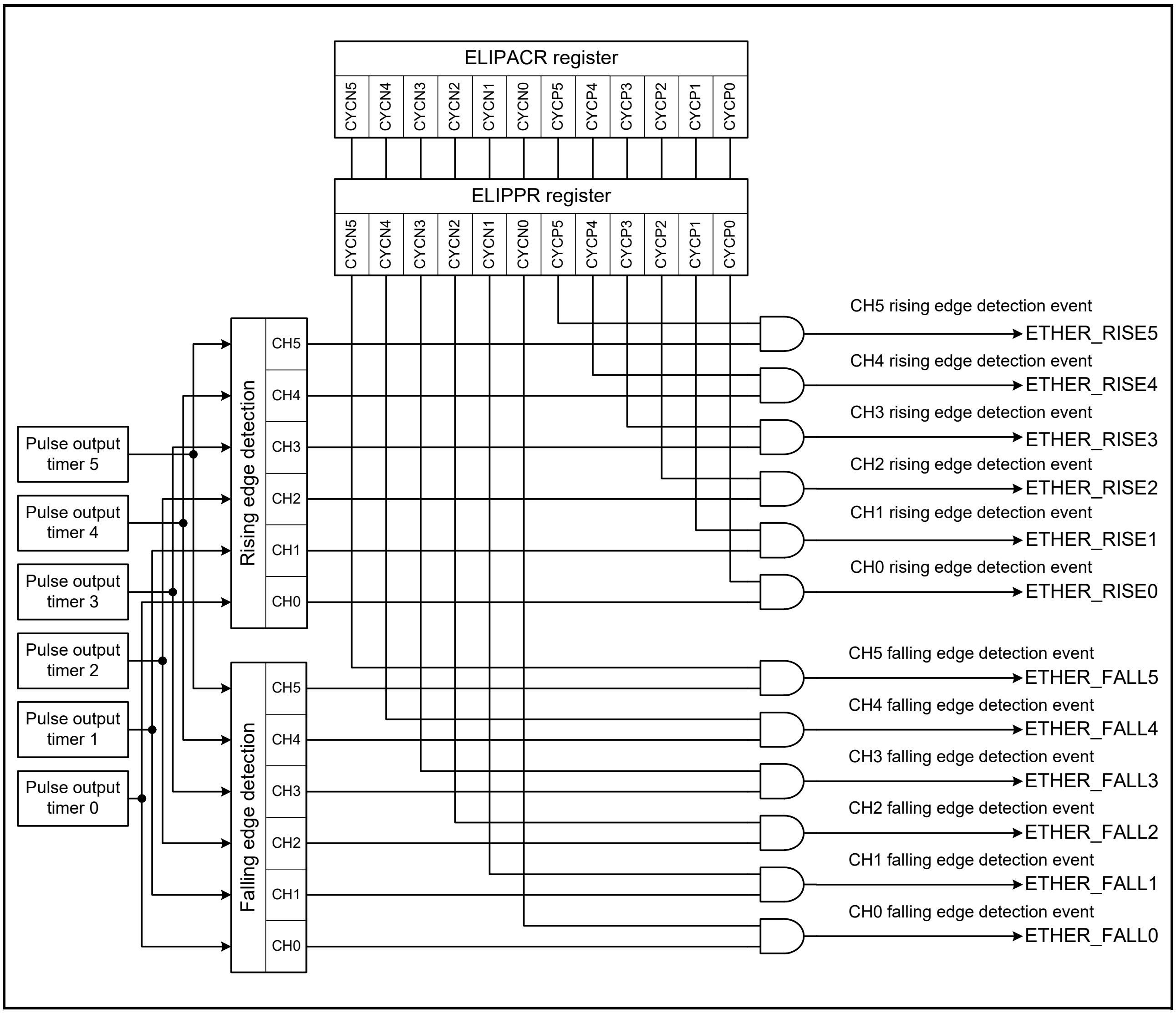


Figure 30.34 Relationship between the pulse output timer and the ELC

30.6 Usage Notes

30.6.1 Constraints on Register Access

When the EPTPC and PTPEDMAC operations are enabled (MSTPCRB.MSTPB13 = 0), some registers in the EPTPC become inaccessible depending on the setting combination of the MSTPCRB.MSTPB15 bit and EPTPC bypass bit (BYPASS.BYPASS0 bit). Table 30.24 to Table 30.25 summarize the constraints on access to the registers.

Table 30.24 Constraints on register access when no channels are bypassed (BYPASS.BYPASS0 = 0)

Constraints on register access				
Ethernet port usage	**Allocation of register addresses for access**			
MSTPB15 setting (EMACC0 and EDMAC0)	**4006 4500h to 4006 45FFh**	**4006 5000h to 4006 503Fh**	**4006 5040h to 4006 53FFh (STCA)**	**4006 5800h to 4006 5BFFh (SYNFP0)**
0	Accessible	Accessible	Accessible	Accessible
1	Accessible	Access prohibited	Access prohibited	Access prohibited

Table 30.25 Constraints on register access when channel 0 is bypassed (BYPASS.BYPASS0 = 1)

Constraints on register access				
Ethernet port usage	**Allocation of register addresses for access**			
MSTPB15 setting (ETHERC0 and EDMAC0)	**4006 4500h to 4006 45FFh**	**4006 5000h to 4006 503Fh**	**4006 5040h to 4006 53FFh (STCA)**	**4006 5800h to 4006 5BFFh (SYNFP0)**
0	Accessible	Access prohibited	Access prohibited	Access prohibited
1	Accessible	Access prohibited	Access prohibited	Access prohibited

Note: Access to an access-prohibited register can lead to a bus timeout error. If a bus timeout error occurs, set the PTRSTR.RESET bit to 1 to reset the EPTPC by software.

30.6.2 Wait Cycles for Register Access

Access to registers in the EPTPC involves the arbitration of different clock signals, specifically the peripheral module clock signal (PCLKA), the STCA clock signal, and the MII clock signals such as TX_CLK. Accordingly, the number of wait cycles for register access differs depending on the combination of the frequency settings for these clock signals.

Table 30.26 gives examples of numbers of wait cycles for different combinations. Add 1 to 2 cycles to these values to obtain the number of access cycles.

Table 30.26 Wait cycles for register access when the STCA clock is 20 MHz

	STCA clock = 20 MHz							
	Peripheral module clock PCLKA = 120 MHz				**Peripheral module clock PCLKA = 20 MHz**			
	MII clock 25 MHz (100 Mbps)		**MII clock 2.5 MHz (10 Mbps)**		**MII clock 25 MHz (100 Mbps)**		**MII clock 2.5 MHz (10 Mbps)**	
Address range	**Read**	**Write**	**Read**	**Write**	**Read**	**Write**	**Read**	**Write**
4006 4500h to 4006 45FFh	2	2	2	2	2	2	2	2
4006 5000h to 4006 503Fh	4	4	4	4	4	4	4	4
4006 5040h to 4006 53FFh (STCA)	7	27 to 41*1	7	27 to 41*1	7	15 to 17*1	7	15 to 17*1
4006 5800h to 4006 5BFFh (SYNFP0)	8	23 to 33*2	8	111 to 209*2	8	15 to 17*2	8	31 to 49*2

Note 1. The number of wait cycles in access to the STCA-related registers (W_{STCA}) can be calculated to the following range from the periods of the peripheral module clock ($t_{c(PCLKA)}$) and STCA clock ($t_{c(STCA)}$).

Minimum value of W_{STCA} = $\mathrm{Int}\,(t_{c(STCA)}/t_{c(PCLKA)}) \times 2 + 15$ ($t_{c(PCLKA)} \leq t_{c(STCA)}$)

= 15 ($t_{c(PCLKA)} > t_{c(STCA)}$)

Maximum value of W_{STCA} = Int ($t_{c(STCA)}/t_{c(PCLKA)}$) × 4 + 17 ($t_{c(PCLKA)} \leq t_{c(STCA)}$)

= 17 ($t_{c(PCLKA)} > t_{c(STCA)}$)

- Int(A) is the calculation of the largest integer not greater than A.
- This calculation assumes that the CPU clock and peripheral module clock have the same periods.

For example, if the frequency of the peripheral module clock is 120 MHz and that of the STCA clock is 1/6 that of the peripheral module clock (= 20 MHz),

Minimum value of W_{STCA} = Int (50 [ns]/8.3 [ns]) × 2 + 15 = 27, and

Maximum value of W_{STCA} = Int (50 [ns]/8.3 [ns]) × 4 + 17 = 41.

If REF50CK0 is used as the STCA clock, the frequency of the STCA clock is 25 MHz.

Note 2. The number of wait cycles in access to the SYNFP-related registers (W_{SYNF}) can be calculated to the following range from the periods of the peripheral module clock ($t_{c(PCLKA)}$) and MII clock ($t_{c(MII)}$).

Minimum value of W_{SYNF} = Int ($t_{c(MII)}/t_{c(PCLKA)}$) × 2 + 15 ($t_{c(PCLKA)} \leq t_{c(MII)}$)

= 15 ($t_{c(PCLKA)} > t_{c(MII)}$)

Maximum value of W_{SYNF} = Int ($t_{c(MII)}/t_{c(PCLKA)}$) × 4 + 17 ($t_{c(PCLKA)} \leq t_{c(MII)}$)

= 17 ($t_{c(PCLKA)} > t_{c(MII)}$)

- Int(A) is the calculation of the largest integer not greater than A.
- This calculation assumes that the CPU clock and peripheral module clock have the same periods.

For example, if the frequency of the peripheral module clock is 120 MHz and the transmission rate is 10 Mbps (so the MII clock is running at 2.5 MHz),

Minimum value of W_{SYNF} = Int (400 [ns]/8.3 [ns]) × 2 + 15 = 111, and

Maximum value of W_{SYNF} = Int (400 [ns]/8.3 [ns]) × 4 + 17 = 209.

31. Ethernet DMA Controller (EDMAC)

31.1 Overview

The MCU provides two channels for the Ethernet DMA Controller (EDMAC), one channel for the Ethernet Controller (ETHERC) and one channel for the Ethernet PTP Controller (EPTPC). EDMAC0 controls data transmission and reception for ETHERC0. The PTPEDMAC controls data transmission and reception for ETHERC0 based on the EPTPC settings.

The EDMAC controls most of the transmit and receive buffer management for communications. This reduces the load on the CPU and allows efficient data transmission and reception. The data transfers are controlled according to the information referred to as descriptors, in memory.

Table 31.1 lists the EDMAC specifications and Figure 31.1 shows the configuration. Figure 31.2 shows the configuration of descriptors and transmit and receive buffers in memory.

Table 31.1 EDMAC specifications

Parameter	Specifications
Data transmission and reception	• Controls data transmission and reception according to descriptors • Supports single buffer frame transmission and reception (1 buffer per frame) and multi-buffer frame transmission and reception (multiple buffers per frame)
Functions	• Minimizes system bus occupancy time using block transfer (32-byte units) • Writes back the transmit or receive frame state to descriptors • Inserts padding in receive data
Module-stop function	Module-stop state can be set to reduce power consumption

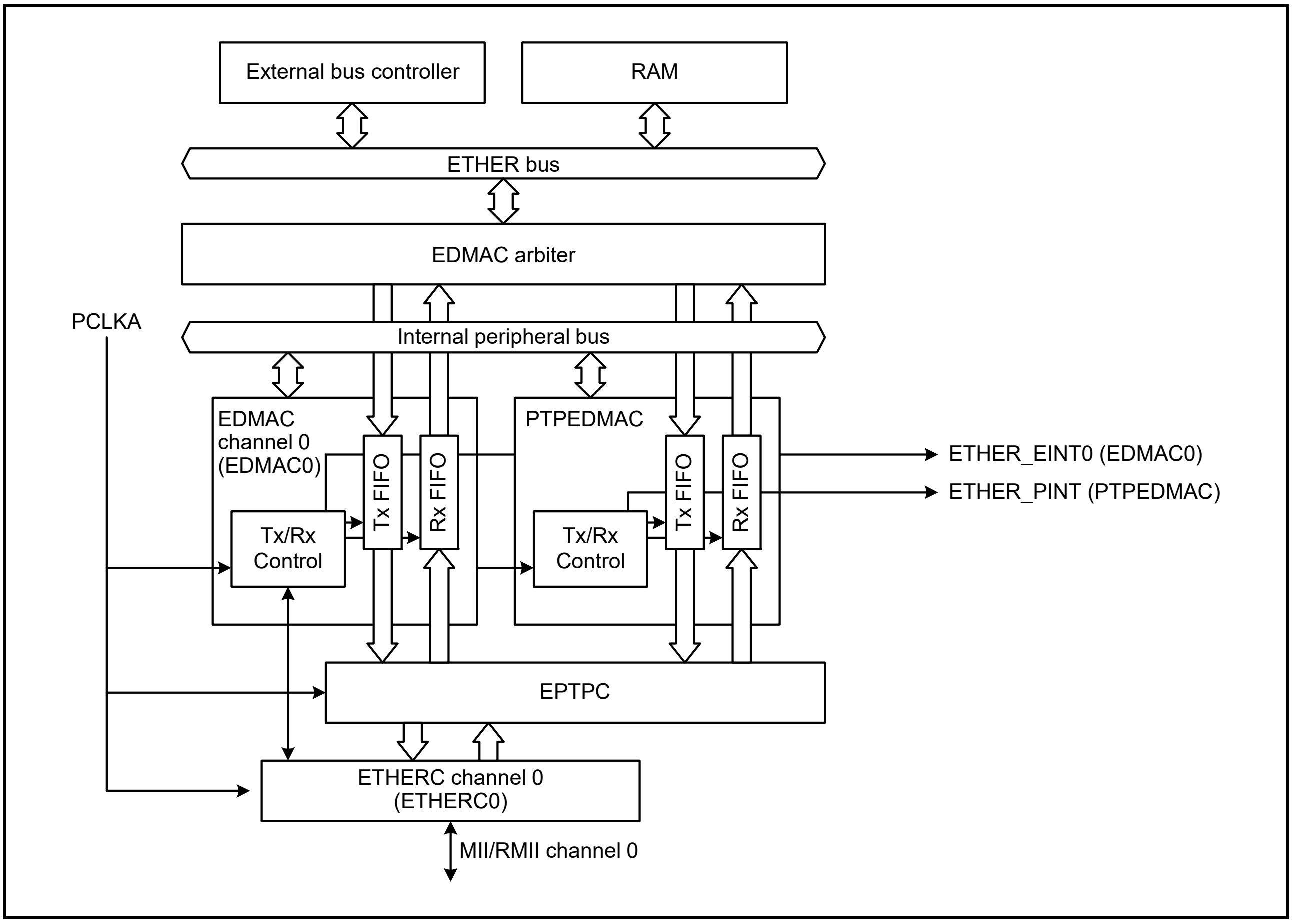


Figure 31.1 EDMAC configuration

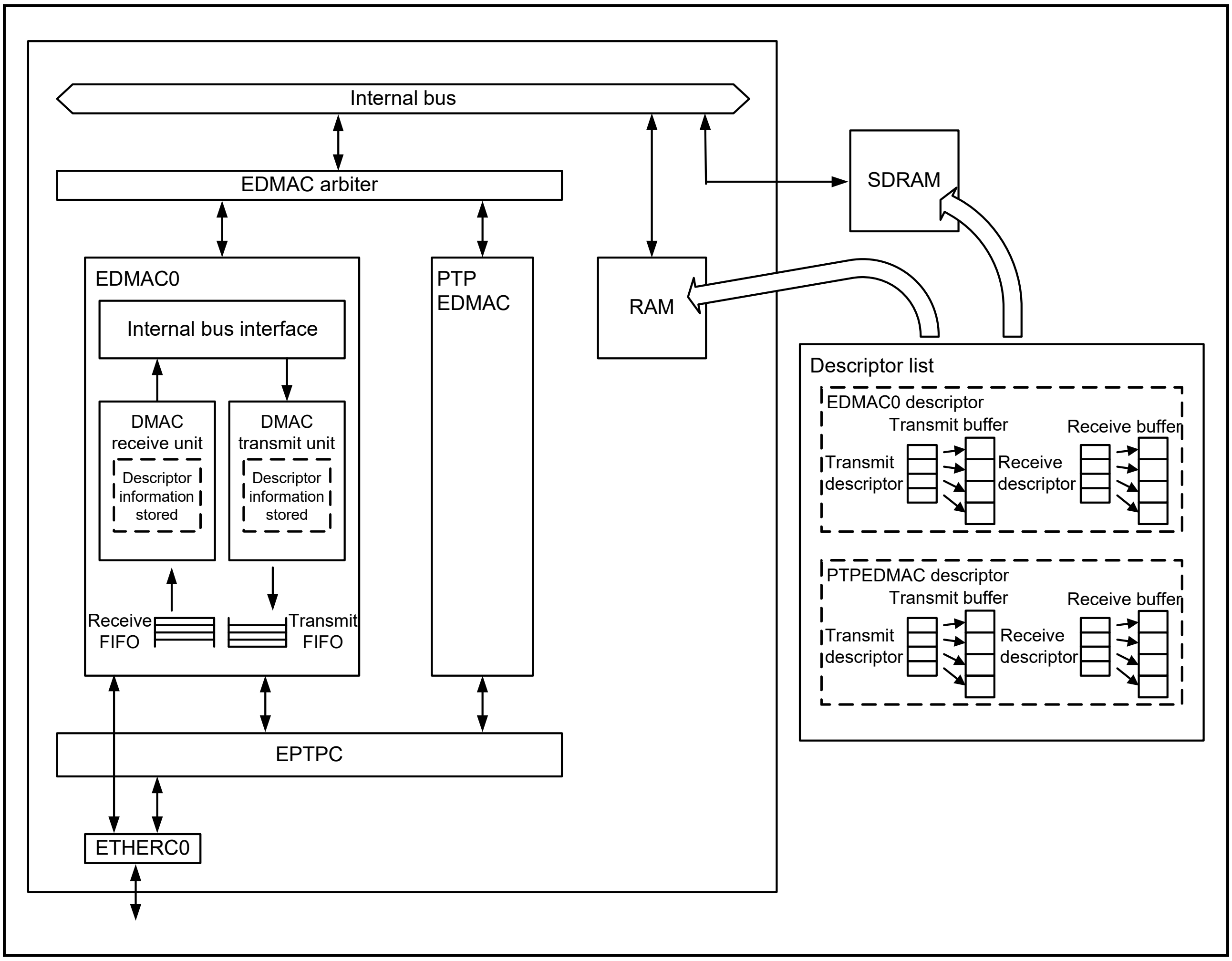


Figure 31.2 Configuration of descriptors and transmit and receive buffers in memory

31.2 Register Descriptions

31.2.1 EDMAC Mode Register (EDMR)

Address(es): EDMAC0.EDMR 4006 4000h, PTPEDMAC.EDMR 4006 4400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	DE	DL[1:0]		—	—	—	SWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SWR	Software Reset	When 1 is written, the associated channels of the EDMAC and ETHERC are reset. Note: The ETHERC is not reset for the PTPEDMAC. The TDLAR, RDLAR, RMFCR, TFUCR, and RFOCR registers are not reset with this bit.The read value is 0.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	DL[1:0]	Transmit/Receive Descriptor Length	b5 b4 0 0: 16 bytes 0 1: 32 bytes 1 0: 64 bytes 1 1: 16 bytes.	R/W
b6	DE	Big Endian Mode/Little Endian Mode*1	0: Big endian mode 1: Little endian mode.	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This setting applies to data for the transmit and receive buffers. It does not apply to transmit and receive descriptors and registers.

The EDMR register controls EDMAC operation. Set the EDMR register during initialization process after a reset. When rewriting this register outside of the initialization process, set the SWR bit to 1 to reset the EDMAC and ETHERC, and then set this register again. If the ETHERC and EDMAC are reset during data transmission or reception, abnormal data might be sent on the line. Do not rewrite this register while the ETHERC transmit or receive function is enabled. It takes 64 cycles of the peripheral module clock (PCLKA) to initialize the ETHERC and EDMAC. Complete the initialization before accessing registers in the ETHERC and EDMAC.

31.2.2 EDMAC Transmit Request Register (EDTRR)

Address(es): EDMAC0.EDTRR 4006 4008h, PTPEDMAC.EDTRR 4006 4408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TR	Transmit Request	When 1 is written, the EDMAC reads the associated descriptor and transmits frames where the TD0.TACT bit is 1. The TR bit clears to 0 after all the valid frames are transmitted. Writing 0 to this bit has no effect.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The EDTRR register controls EDMAC transmission. After the EDMAC transmits one frame, it reads the next descriptor. When the TD0.TACT bit in the descriptor is 1, the EDMAC continues transmission. When the TD0.TACT bit is 0, the EDMAC sets the TR bit to 0 and stops transmission.

31.2.3 EDMAC Receive Request Register (EDRRR)

Address(es): EDMAC0.EDRRR 4006 4010h, PTPEDMAC.EDRRR 4006 4410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RR	Receive Request	0: Disable the receive function*1 1: Read receive descriptor and enable the receive function.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. If the receive function is disabled during frame reception, write-back to the receive descriptor is not performed successfully. Subsequent pointers for reading a receive descriptor become abnormal and the EDMAC cannot operate normally. In this case, to enable the EDMAC receive function again, execute a software reset by setting the EDMR.SWR bit to 1.
To disable the EDMAC receive function without resetting the EDMAC, set the ETHERC0.ECMR.RE bit to 0. After the EDMAC completes reception and write-back to the receive descriptor is confirmed, set the RR bit to 0.

The EDRRR register controls EDMAC reception. When the RR bit sets to 1, the EDMAC reads the receive descriptor. When the RD0.RACT bit is 1, the EDMAC waits for a receive request from the ETHERC. When the EDMAC has received data for the receive buffer size, it reads the next descriptor and waits to receive a frame. If the RD0.RACT bit is 0, the EDMAC sets the RR bit to 0 and stops reception.

31.2.4 Transmit Descriptor List Start Address Register (TDLAR)

Address(es): EDMAC0.TDLAR 4006 4018h, PTPEDMAC.TDLAR 4006 4418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify the start address of the transmit descriptor list. Set the start address according to the descriptor length selected in the EDMR.DL[1:0] bits. • 16-byte boundary: Lower 4 bits = 0000b • 32-byte boundary: Lower 5 bits = 00000b • 64-byte boundary: Lower 6 bits = 000000b.	R/W

The TDLAR register specifies the start address of the transmit descriptor list. Align each descriptor on the associated boundary to the descriptor length selected in the EDMR.DL[1:0] bits. Do not rewrite the TDLAR register during transmission. Rewrite the TDLAR register while the EDTRR.TR bit is 0.

31.2.5 Receive Descriptor List Start Address Register (RDLAR)

Address(es): EDMAC0.RDLAR 4006 4020h, PTPEDMAC.RDLAR 4006 4420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	The start address of the receive descriptor list is set. Set the start address according to the descriptor length selected in the EDMR.DL[1:0] bits. • 16-byte boundary: Lower 4 bits = 0000b • 32-byte boundary: Lower 5 bits = 00000b • 64-byte boundary: Lower 6 bits = 000000b.	R/W

The RDLAR register specifies the start address of the receive descriptor list. Allocate each descriptor on the associated boundary to the descriptor length selected in the EDMR.DL[1:0] bits. Do not rewrite the RDLAR register during reception. Rewrite the RDLAR register while the EDRRR.RR bit is 0.

31.2.6 ETHERC/EDMAC Status Register (EDMAC0.EESR)

Address(es): EDMAC0.EESR 4006 4028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWB	—	—	—	TABT	RABT	RFCOF	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTLF	RTSF	PRE	CERF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CERF	CRC Error Flag	0: CRC error not detected 1: CRC error detected.	R/W
b1	PRE	PHY-LSI Receive Error Flag	0: PHY-LSI receive error not detected 1: PHY-LSI receive error detected.	R/W
b2	RTSF	Frame-Too-Short Error Flag	0: Frame-too-short error not detected 1: Frame-too-short error detected.	R/W
b3	RTLF	Frame-Too-Long Error Flag	0: Frame-too-long error not detected 1: Frame-too-long error detected.	R/W
b4	RRF	Alignment Error Flag	0: Alignment error not detected 1: Alignment error detected.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RMAF	Multicast Address Frame Receive Flag	0: Multicast address frame not received 1: Multicast address frame received.	R/W
b8	TRO	Transmit Retry Over Flag	0: Transmit retry-over condition not detected 1: Transmit retry-over condition detected.	R/W
b9	CD	Late Collision Detect Flag	0: Late collision not detected 1: Late collision detected during frame transmission.	R/W
b10	DLC	Loss of Carrier Detect Flag	0: Loss of carrier not detected 1: Loss of carrier detected during frame transmission.	R/W
b11	CND	Carrier Not Detect Flag	0: Carrier detected when transmission started 1: Carrier not detected during preamble transmission.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	RFOF	Receive FIFO Overflow Flag	0: No overflow occurred 1: Overflow occurred.	R/W
b17	RDE	Receive Descriptor Empty Flag	0: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 1 1: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 0.	R/W
b18	FR	Frame Receive Flag	0: Frame not received 1: Frame received and update of the receive descriptor is complete.	R/W
b19	TFUF	Transmit FIFO Underflow Flag	0: No underflow occurred 1: Underflow occurred.	R/W
b20	TDE	Transmit Descriptor Empty Flag	0: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 1 1: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 0.	R/W
b21	TC	Frame Transfer Complete Flag	0: Transfer not complete or no transfer requested 1: All frames indicated in the transmit descriptor were completely transferred to the transmit FIFO.	R/W

Bit	Symbol	Bit name	Description	R/W
b22	ECI	ETHERC Status Register Source Flag	0: ETHERC status interrupt source not detected 1: ETHERC status interrupt source detected.	R[*1]
b23	ADE	Address Error Flag	0: Invalid memory address not detected (normal operation) 1: Invalid memory address detected.[*2]	R/W
b24	RFCOF	Receive Frame Counter Overflow Flag	0: Receive frame counter did not overflow 1: Receive frame counter overflowed.	R/W
b25	RABT	Receive Abort Detect Flag	0: Frame reception not aborted or no reception requested 1: Frame reception aborted.	R/W
b26	TABT	Transmit Abort Detect Flag	0: Frame transmission not aborted or no transmission requested. 1: Frame transmission aborted.	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	TWB	Write-Back Complete Flag	0: Write-back not complete or no transmission requested 1: Write-back to the transmit descriptor completed.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The ECI flag is read-only. When the source in the ETHERC0.ECSR register is cleared, the ECI flag is also cleared.
Note 2. When an address error is detected, the EDMAC halts the process. To resume operation, set the EDMR.SWR bit to 1 (resetting the EDMAC and ETHERC), and then reconfigure the EDMAC and ETHERC.

The EDMAC0.EESR register indicates the ETHERC and EDMAC communication status. Each flag in the EESR register can be output as an interrupt request signal (ETHER_EINT0) from the EDMAC. Writing 1 clears all of the flags except ECI to 0. Writing 0 does not affect any of the flag values. The interrupt sources are enabled by setting the associated bits in the EDMAC0.EESIPR register.

CERF flag (CRC Error Flag)

The CERF flag sets to 1 when an error is detected while checking the frame check sequence (FCS) field of the receive frame.

PRE flag (PHY-LSI Receive Error Flag)

The PRE flag indicates that the RX_ER signal output from the PHY-LSI is high.

RTSF flag (Frame-Too-Short Error Flag)

The RTSF flag indicates that a received frame is less than 64 bytes.

RTLF flag (Frame-Too-Long Error Flag)

The RTLF flag indicates that a received frame is greater than the upper limit of the receive frame length set in the ETHERC0.RFLR register. The excess data is discarded.

RRF flag (Alignment Error Flag)

The RRF flag indicates that a frame is not an integral number of octets. The last word that is not an integral number of octets is not transferred.

RMAF flag (Multicast Address Frame Receive Flag)

The RMAF flag indicates that a multicast frame was received.

TRO flag (Transmit Retry Over Flag)

The TRO flag indicates that a collision occurred again during the 15th retry of frame transmission.

CD flag (Late Collision Detect Flag)

The CD flag indicates that a late collision was detected during frame transmission.

DLC flag (Loss of Carrier Detect Flag)

The DLC flag indicates that a loss of carrier was detected during frame transmission.

CND flag (Carrier Not Detect Flag)

The CND flag sets to 1 when a carrier is not detected during preamble transmission.

RFOF flag (Receive FIFO Overflow Flag)

The RFOF flag indicates that the receive FIFO overflowed during frame reception.

RDE flag (Receive Descriptor Empty Flag)

The RDE flag indicates that the read receive descriptor is invalid. When this flag sets to 1, set the RD0.RACT bit in the receive descriptor to 1 and set the EDRRR.RR bit to 1 to resume reception.

FR flag (Frame Receive Flag)

The FR flag indicates that a frame was received and the receive descriptor was updated. The FR flag sets to 1 every time a frame is received.

TFUF flag (Transmit FIFO Underflow Flag)

The TFUF flag indicates that no data remains in the transmit FIFO during frame transmission. Incomplete data is sent to the line.

TDE flag (Transmit Descriptor Empty Flag)

The TDE flag indicates that the TD0.TACT bit of the transmit descriptor is 0 while the previous transmit descriptor indicates that the frame is not complete (TD0.TFP[1:0] bits are 10b or 00b) in multi-buffer frame transmission. As a result, an incomplete frame might be sent.

When this flag sets to 1, perform a software reset and then set the EDTRR.TR bit to 1 to resume transmission. Transmission starts from the address stored in the TDLAR register.

TC flag (Frame Transfer Complete Flag)

The TC flag indicates that all the data specified in the transmit descriptor was transmitted from the ETHERC. This flag sets to 1 when one frame was transmitted in single-buffer frame transmission or when the last data of a frame is transmitted in multi-buffer frame transmission and the TD0.TACT bit in the next transmit descriptor is 0. After frame transmission is complete, the EDMAC writes the transfer status back to the descriptor.

ECI flag (ETHERC Status Register Source Flag)

The ECI flag sets to 1 when an interrupt request is generated by the ETHERC.ECSR register.

ADE flag (Address Error Flag)

The ADE flag indicates that the memory address that the EDMAC tried to use for transfer is invalid.

RFCOF flag (Receive Frame Counter Overflow Flag)

The RFCOF flag indicates that the next frame reception started while the number of frames stored in the receive FIFO reached the maximum number of frames (16 frames). The received frame is discarded while the RFCOF flag is 1.

RABT flag (Receive Abort Detect Flag)

The RABT flag indicates that the ETHERC aborted frame reception because of a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or other error.

TABT flag (Transmit Abort Detect Flag)

The TABT flag indicates that the ETHERC aborted frame transmission because of transmit retry over, loss of carrier, no carrier detection, or other error.

TWB flag (Write-Back Complete Flag)

The TWB flag indicates the EDMAC completed writing back to the descriptor after frame transmission. This flag sets to 1 after each frame transmission when the TRIMD.TIM bit is 0. It only sets to 1 when the TRIMD.TIS bit is 1.

31.2.7 PTP/EDMAC Status Register (PTPEDMAC.EESR)

Address(es): PTPEDMAC.EESR 4006 4428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWB	—	—	—	TABT	—	RFCOF	ADE	—	TC	TDE	TFUF	FR	RDE	RFOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MACE	—	—	—	PVER	TYPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	TYPE[3:0]	PTP v2 Message Type Flag	b3 b0 0 0 0 0: Sync 0 0 0 1: Delay_Req 0 0 1 0: Pdelay_Req 0 0 1 1: Pdelay_Resp 1 0 0 0: Follow_Up 1 0 0 1: Delay_Resp 1 0 1 0: Pdelay_Resp_Follow_Up 1 0 1 1: Announce 1 1 0 0: Signaling 1 1 0 1: Management. Other settings are reserved.	R/W
b4	PVER	PTP v2 Packet Flag	0: Current packet is not a PTP v2 packet 1: Current packet is a PTP v2 packet.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MACE	MAC Address Mismatch Flag	0: Source MAC address of transmit frame data matches the set value 1: Source MAC address of transmit frame data does not match the set value.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	RFOF	Receive FIFO Overflow Flag	0: No overflow occurred 1: Overflow occurred.	R/W
b17	RDE	Receive Descriptor Empty Flag	0: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 1 1: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 0.	R/W
b18	FR	Frame Receive Flag	0: Frame not received 1: Frame received and receive descriptor updated.	R/W
b19	TFUF	Transmit FIFO Underflow Flag	0: No underflow occurred 1: Underflow occurred.	R/W
b20	TDE	Transmit Descriptor Empty Flag	0: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 1 1: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 0.	R/W
b21	TC	Frame Transfer Complete Flag	0: Transfer not complete or transfer not requested 1: All frames indicated in the transmit descriptor were completely transferred to the transmit FIFO.	R/W
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	ADE	Address Error Flag	0: Invalid memory address not detected (normal operation) 1: Invalid memory address detected.*1	R/W
b24	RFCOF	Receive Frame Counter Overflow Flag	0: Receive frame counter did not overflow 1: Receive frame counter overflowed.	R/W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b26	TABT	Transmit Abort Detect Flag	0: Frame transmission not aborted or transmission not requested 1: Frame transmission aborted.	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	TWB	Write-Back Complete Flag	0: Write-back not complete or transmission not requested 1: Write-back to the transmit descriptor completed.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. When an address error is detected, the EDMAC halts the process. To resume operation, set the EDMR.SWR bit to 1 (resetting the EDMAC and ETHERC), and then reconfigure the EDMAC and ETHERC.

The PTPEDMAC.EESR register indicates the PTPEDMAC communication status. Each flag in the EESR register can be output as an interrupt request signal (ETHER_PINT) from the PTPEDMAC. Writing 1 clears the flags to 0. Writing 0 does not affect the flag values. All of the interrupt sources, except for the TYPE[3:0] flag, are enabled by setting the associated bits in the PTPEDMAC.EESIPR register.

TYPE[3:0] flags (PTP v2 Message Type Flag)

The TYPE[3:0] flags indicate the type of received PTP message.

PVER flag (PTP v2 Packet Flag)

The PVER flag indicates whether the received packet is a PTP v2 packet.

MACE flag (MAC Address Mismatch Flag)

The MACE flag indicates that the source MAC address is different from the set value.

RFOF flag (Receive FIFO Overflow Flag)

The RFOF flag indicates that the receive FIFO overflowed during frame reception.

RDE flag (Receive Descriptor Empty Flag)

The RDE flag indicates that the read receive descriptor is invalid. When this flag sets to 1, set the RD0.RACT bit in the receive descriptor to 1 and set the EDRRR.RR bit to 1 to resume reception.

FR flag (Frame Receive Flag)

The FR flag indicates that a frame was received and the receive descriptor was updated. The FR flag sets to 1 every time a frame is received.

TFUF flag (Transmit FIFO Underflow Flag)

The TFUF flag indicates that no data remains in the transmit FIFO during frame transmission. Incomplete data is sent to the line.

TDE flag (Transmit Descriptor Empty Flag)

The TDE flag indicates that the TD0.TACT bit of the transmit descriptor is 0 while the previous transmit descriptor indicates that the frame is not complete (TD0.TFP[1:0] bits are 10b or 00b) in multi-buffer frame transmission. As a result, an incomplete frame might be sent.

When this flag sets to 1, perform a software reset and then set the EDTRR.TR bit to 1 to resume transmission. Transmission starts from the address stored in the TDLAR register.

TC flag (Frame Transfer Complete Flag)

The TC flag indicates that all the data specified in the transmit descriptor was transmitted from the ETHERC. This flag sets to 1 when one frame is transmitted in single-buffer frame transmission or when the last data of a frame is transmitted in multi-buffer frame transmission and the TD0.TACT bit in the next transmit descriptor is 0. After frame transmission is complete, the PTPEDMAC writes the transfer status back to the descriptor.

ADE flag (Address Error Flag)

The ADE flag indicates that the memory address that the PTPEDMAC tried to use for transfer is invalid.

RFCOF flag (Receive Frame Counter Overflow Flag)

The RFCOF flag indicates that the next frame reception started while the number of frames stored in the receive FIFO reached the maximum number of frames (16 frames). Received frames are discarded while the RFCOF flag is 1.

TABT flag (Transmit Abort Detect Flag)

The TABT flag indicates that the ETHERC aborted frame transmission because of transmit retry over, loss of carrier, no carrier detection, or other error.

TWB flag (Write-Back Complete Flag)

The TWB flag indicates that the PTPEDMAC completed writing back to the descriptor after frame transmission. This flag sets to 1 after each frame transmission when the TRIMD.TIM bit is 0. It only sets to 1 when the TRIMD.TIS bit is 1.

31.2.8 ETHERC/EDMAC Status Interrupt Enable Register (EDMAC0.EESIPR)

Address(es): EDMAC0.EESIPR 4006 4030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWBIP	—	—	—	TABTIP	RABTIP	RFCOFIP	ADEIP	ECIIP	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	RMAFIP	—	—	RRFIP	RTLFIP	RTSFIP	PREIP	CERFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CERFIP	CRC Error Interrupt Request Enable	0: Disable CRC error interrupt requests 1: Enable CRC error interrupt requests.	R/W
b1	PREIP	PHY-LSI Receive Error Interrupt Request Enable	0: Disable PHY-LSI receive error interrupt requests 1: Enable PHY-LSI receive error interrupt requests.	R/W
b2	RTSFIP	Frame-Too-Short Error Interrupt Request Enable	0: Disable frame-too-short error interrupt requests 1: Enable frame-too-short error interrupt requests.	R/W
b3	RTLFIP	Frame-Too-Long Error Interrupt Request Enable	0: Disable frame-too-long error interrupt requests 1: Enable frame-too-long error interrupt requests.	R/W
b4	RRFIP	Alignment Error Interrupt Request Enable	0: Disable alignment error interrupt requests 1: Enable alignment error interrupt requests.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RMAFIP	Multicast Address Frame Receive Interrupt Request Enable	0: Disable multicast address frame receive interrupt requests 1: Enable multicast address frame receive interrupt requests.	R/W
b8	TROIP	Transmit Retry Over Interrupt Request Enable	0: Disable transmit retry over interrupt requests 1: Enable transmit retry over interrupt requests.	R/W
b9	CDIP	Late Collision Detect Interrupt Request Enable	0: Disable late collision detected interrupt requests 1: Enable late collision detected interrupt requests.	R/W
b10	DLCIP	Loss of Carrier Detect Interrupt Request Enable	0: Disable loss of carrier detected interrupt requests 1: Enable loss of carrier detected interrupt requests.	R/W
b11	CNDIP	Carrier Not Detect Interrupt Request Enable	0: Disable carrier not detected interrupt requests 1: Enable carrier not detected interrupt requests.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	RFOFIP	Receive FIFO Overflow Interrupt Request Enable	0: Disable overflow interrupt requests 1: Enable overflow interrupt requests.	R/W
b17	RDEIP	Receive Descriptor Empty Interrupt Request Enable	0: Disable receive descriptor empty interrupt requests 1: Enable receive descriptor empty interrupt requests.	R/W
b18	FRIP	Frame Receive Interrupt Request Enable	0: Disable frame reception interrupt requests 1: Enable frame reception interrupt requests.	R/W
b19	TFUFIP	Transmit FIFO Underflow Interrupt Request Enable	0: Disable underflow interrupt requests 1: Enable underflow interrupt requests.	R/W
b20	TDEIP	Transmit Descriptor Empty Interrupt Request Enable	0: Disable transmit descriptor empty interrupt requests 1: Enable transmit descriptor empty interrupt requests.	R/W
b21	TCIP	Frame Transfer Complete Interrupt Request Enable	0: Disable frame transmission complete interrupt requests 1: Enable frame transmission complete interrupt requests.	R/W
b22	ECIIP	ETHERC Status Register Source Interrupt Request Enable	0: Disable ETHERC status interrupt requests 1: Enable ETHERC status interrupt requests.	R/W

Bit	Symbol	Bit name	Description	R/W
b23	ADEIP	Address Error Interrupt Request Enable	0: Disable address error interrupt requests 1: Enable address error interrupt requests.	R/W
b24	RFCOFIP	Receive Frame Counter Overflow Interrupt Request Enable	0: Disable receive frame counter overflow interrupt requests 1: Enable receive frame counter overflow interrupt requests.	R/W
b25	RABTIP	Receive Abort Detect Interrupt Request Enable	0: Disable receive abort detected interrupt requests 1: Enable receive abort detected interrupt requests.	R/W
b26	TABTIP	Transmit Abort Detect Interrupt Request Enable	0: Disable transmit abort detected interrupt requests 1: Enable transmit abort detected interrupt requests.	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	TWBIP	Write-Back Complete Interrupt Request Enable	0: Disable write-back complete interrupt requests 1: Enable write-back complete interrupt requests.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The EDMAC0.EESIPR register enables interrupt requests associated with bits in the EDMAC0.EESR register. When a bit in this register is 1, the associated interrupt request is enabled.

31.2.9 PTP/EDMAC Status Interrupt Enable Register (PTPEDMAC.EESIPR)

Address(es): PTPEDMAC.EESIPR 4006 4430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	TWBIP	—	—	—	TABTIP	—	RFCOFIP	ADEIP	—	TCIP	TDEIP	TFUFIP	FRIP	RDEIP	RFOFIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MACEIP	—	—	—	PVERIP	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	PVERIP	PTP v2 Packet Receive Interrupt Request Enable	0: Disable PTP v2 packet receive interrupt requests 1: Enable PTP v2 packet receive interrupt requests.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MACEIP	MAC Address Mismatch Interrupt Request Enable	0: Disable interrupt requests generated when the source MAC address of transmit frame data does not match the set value 1: Enable interrupt requests generated when the source MAC address of transmit frame data does not match the set value.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	RFOFIP	Receive FIFO Overflow Interrupt Request Enable	0: Disable overflow interrupt requests 1: Enable overflow interrupt requests.	R/W
b17	RDEIP	Receive Descriptor Empty Interrupt Request Enable	0: Disable receive descriptor empty interrupt requests 1: Enable receive descriptor empty interrupt requests.	R/W
b18	FRIP	Frame Receive Interrupt Request Enable	0: Disable frame receive interrupt requests 1: Enable frame receive interrupt requests.	R/W
b19	TFUFIP	Transmit FIFO Underflow Interrupt Request Enable	0: Disable underflow interrupt requests 1: Enable underflow interrupt requests.	R/W
b20	TDEIP	Transmit Descriptor Empty Interrupt Request Enable	0: Disable transmit descriptor empty interrupt requests 1: Enable transmit descriptor empty interrupt requests.	R/W
b21	TCIP	Frame Transfer Complete Interrupt Request Enable	0: Disable frame transmission complete interrupt requests 1: Enable frame transmission complete interrupt requests.	R/W
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	ADEIP	Address Error Interrupt Request Enable	0: Disable address error interrupt requests 1: Enable address error interrupt requests.	R/W
b24	RFCOFIP	Receive Frame Counter Overflow Interrupt Request Enable	0: Disable receive frame counter overflow interrupt requests 1: Enable receive frame counter overflow interrupt requests.	R/W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26	TABTIP	Transmit Abort Detect Interrupt Request Enable	0: Disable transmit abort detect interrupt requests 1: Enable transmit abort detect interrupt requests.	R/W
b29 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30	TWBIP	Write-Back Complete Interrupt Request Enable	0: Disable write-back complete interrupt requests 1: Enable write-back complete interrupt requests.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The PTPEDMAC.EESIPR register enables interrupt requests associated with bits in the PTPEDMAC.EESR register. When a bit in this register is 1, the associated interrupt request is enabled.

31.2.10 ETHERC/EDMAC Transmit/Receive Status Copy Enable Register (EDMAC0.TRSCER)

Address(es): EDMAC0.TRSCER 4006 4038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RMAFCE	—	—	RRFCE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	RRFCE	RRF Flag Copy Enable	0: Reflect the EESR.RRF flag status in the RD0.RFE bit of the receive descriptor 1: Do not reflect the EESR.RRF flag status in the RD0.RFE bit of the receive descriptor.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RMAFCE	RMAF Flag Copy Enable	0: Reflect the EESR.RMAF flag status in the RD0.RFE bit of the receive descriptor 1: Do not reflect the EESR.RMAF flag status in the RD0.RFE bit of the receive descriptor.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The EDMAC0.TRSCER register selects whether the receive status indicated in the EDMAC0.EESR.RMAF and RRF flags is reflected in the RFE bit of the receive descriptor as a summary. The bits in this register are associated with bits in the EESR register that have the same number. When the RMAFCE or RRFCE bit is set to 0, the associated receive status is reflected in the RFE bit. When the RMAFCE or RRFCE bit is set to 1, the associated receive status is not reflected.

31.2.11 Missed-Frame Counter Register (RMFCR)

Address(es): EDMAC0.RMFCR 4006 4040h, PTPEDMAC.RMFCR 4006 4440h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MFC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	MFC[15:0]	Missed-Frame Counter	These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMFCR register indicates that the number of frames that could not be stored in the receive FIFO and so were discarded during reception. When the receive FIFO overflows, it stops receiving data, and the rest of frames are discarded. At the same time, the RMFCR register value is incremented. When the RMFCR register value reaches FFFFh, count-up is halted. Writing any value to the RMFCR register clears the counter value to 0.

For frames that are not completely received, after data in the receive FIFO is transferred to the receive buffer, the RACT bit in the receive descriptor 0 (RD0) clears to 0 (descriptor disabled), the RFS9 bit sets to 1 (receive FIFO overflowed), and the EDMAC0.EESR.RFOF or PTPEDMAC.EESR.RFOF flag sets to 1 (overflow detected).

31.2.12 Transmit FIFO Threshold Register (TFTR)

Address(es): EDMAC0.TFTR 4006 4048h, PTPEDMAC.TFTR 4006 4448h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TFT[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	TFT[10:0]	Transmit FIFO Threshold	000h: Store-and-forward mode 001h to 00Ch: Setting prohibited 00Dh to 200h: The threshold is the set value multiplied by 4. Example: 00Dh: 52 bytes 040h: 256 bytes 100h: 1024 bytes 200h: 2048 bytes 201h to 7FFh: Setting prohibited.	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: When starting transmission before one frame data is completely written, take care to prevent an underflow. To prevent a transmit underflow, Renesas recommends using the initial value (store-and-forward mode).

The TFTR register specifies the transmit FIFO threshold at which the first transmission starts. The actual threshold is the set value multiplied by 4.

The ETHERC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes set in this register, when the transmit FIFO is full, or when one frame of data is completely written. Set the TFTR register while the EDTRR.TR bit is 0.

31.2.13 FIFO Depth Register (FDR)

Address(es): EDMAC0.FDR 4006 4050h, PTPEDMAC.FDR 4006 4450h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TFD[4:0]					—	—	—	RFD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	RFD[4:0]	Receive FIFO Depth	b4 b0 01111: 4096 bytes. Other settings are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	TFD[4:0]	Transmit FIFO Depth	b12 b8 00111: 2048 bytes. Other settings are prohibited.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The FDR register specifies the transmit and receive FIFO depths. Set this register to 0000_070Fh before starting transmission and reception.

31.2.14 Receive Method Control Register (RMCR)

Address(es): EDMAC0.RMCR 4006 4058h, PTPEDMAC.RMCR 4006 4458h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RNR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RNR	Receive Request Reset	0: EDRRR.RR bit (receive request bit) is cleared to 0 when one frame is received 1: EDRRR.RR bit (receive request bit) is not cleared to 0 when one frame is received.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMCR register specifies how to control the EDRRR.RR bit when receiving a frame. When the RNR bit is 0, the EDRRR.RR bit clears to 0 when one frame is received, so it must be set to 1 by software to receive the subsequent frame. When the RNR bit is 1, the EDRRR.RR bit does not clear to 0 when one frame is received, and the EDMAC reads the next receive descriptor and continues frame reception. Renesas recommends setting the RNR bit to 1 when receiving data continuously. Set the RMCR register while the EDRRR.RR bit is 0.

31.2.15 Transmit FIFO Underflow Counter (TFUCR)

Address(es): EDMAC0.TFUCR 4006 4064h, PTPEDMAC.TFUCR 4006 4464h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	UNDER[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	UNDER[15:0]	Transmit FIFO Underflow Count	These bits indicate how many times the transmit FIFO underflows. The counter stops when the counter value reaches FFFFh.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TFUCR register indicates how many times the transmit FIFO underflows. Writing any value to the TFUCR register clears the counter value to 0.

31.2.16 Receive FIFO Overflow Counter (RFOCR)

Address(es): EDMAC0.RFOCR 4006 4068h, PTPEDMAC.RFOCR 4006 4468h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVER[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	OVER[15:0]	Receive FIFO Overflow Count	These bits indicate how many times the receive FIFO overflows. The counter stops when the counter value reaches FFFFh.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RFOCR register indicates how many times the receive FIFO overflows. Writing any value to the RFOCR register clears the counter value to 0.

31.2.17 Independent Output Signal Setting Register (IOSR)

Address(es): EDMAC0.IOSR 4006 406Ch, PTPEDMAC.IOSR 4006 446Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ELB	External Loopback Mode	0: Output low on the ET0_EXOUT pin 1: Output high on the ET0_EXOUT pin.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The IOSR register selects the output level of the ETHERC external output pin (ET0_EXOUT) in external loopback mode. The ELB bit value is output on the ET0_EXOUT pin, which can be used to set loopback mode for the PHY-LSI. To use the loopback function of the PHY-LSI through this register, you must connect the PHY-LSI to the ET0_EXOUT pin.

31.2.18 Flow Control Start FIFO Threshold Setting Register (FCFTR)

Address(es): EDMAC0.FCFTR 4006 4070h, PTPEDMAC.FCFTR 4006 4470h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFO[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFDO[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b2 to b0	RFDO[2:0]	Receive FIFO Data PAUSE Output Threshold	b2 b0 0 0 0:When 224 (256 to 32) bytes of data is stored in the receive FIFO 0 0 1:When 480 (512 to 32) bytes of data is stored in the receive FIFO : 1 1 0:When 1760 (1792 to 32) bytes of data is stored in the receive FIFO 1 1 1:When 2016 (2048 to 32) bytes of data is stored in the receive FIFO.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RFFO[2:0]	Receive FIFO Frame PAUSE Output Threshold	b18 b16 0 0 0:When 2 receive frames are stored in the receive FIFO 0 0 1:When 4 receive frames are stored in the receive FIFO 0 1 0:When 6 receive frames are stored in the receive FIFO : 1 1 0:When 14 receive frames are stored in the receive FIFO 1 1 1:When 16 receive frames are stored in the receive FIFO.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The FCFTR register specifies the ETHERC flow control. Set the threshold to automatically transmit a PAUSE frame. The threshold can be set using the data size (RFDO[2:0] bits) and the number of frames (RFFO[2:0] bits) stored in the receive FIFO. Flow control starts when the stored data size or the number of stored frames reaches its threshold.

31.2.19 Receive Data Padding Insert Register (RPADIR)

Address(es): EDMAC0.RPADIR 4006 4078h, PTPEDMAC.RPADIR 4006 4478h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PADS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	PADR[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	PADR[5:0]	Padding Slot	00h: Insert padding at the head of received data 01h: Insert padding between the 1st and 2nd bytes of received data : 3Eh: Insert padding between the 62nd and 63rd bytes of received data 3Fh: Insert padding between the 63rd and 64th bytes of received data.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	PADS[1:0]	Padding Size	b17b16 0 0: Do not insert padding 0 1: Insert 1 byte 1 0: Insert 2 bytes 1 1: Insert 3 bytes.	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RPADIR register specifies insertion of padding for received data. The padding value is 00h. Set the EDMR.SWR bit to 1 to reset before rewriting the PRADIR register.

31.2.20 Transmit Interrupt Setting Register (TRIMD)

Address(es): EDMAC0.TRIMD 4006 407Ch, PTPEDMAC.TRIMD 4006 447Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	TIM	—	—	—	TIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TIS	Transmit Interrupt Enable	0: Disable transmit interrupts 1: Enable transmit Interrupts. Set the EESR.TWB flag to 1 in the mode selected in the TIM bit to report an interrupt.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TIM	Transmit Interrupt Mode	0: Select transmission complete interrupt mode, where an interrupt occurs when a frame is transmitted 1: Select write-back complete interrupt mode, where an interrupt occurs when write-back to the transmit descriptor is complete while the TWBI bit is 1.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TRIMD register specifies the transmit interrupt mode and enables or disables transmit interrupts. When the condition selected in this register is satisfied, the EESR.TWB flag sets to 1, and an interrupt request is output when the EESIPR.TWBIP bit is 1.

31.2.21 Receive Buffer Write Address Register (RBWAR)

Address(es): EDMAC0.RBWAR 4006 40C8h, PTPEDMAC.RBWAR 4006 44C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The RBWAR register indicates the last address that the EDMAC wrote data to when writing to the receive buffer. Check the contents of this register to identify which address in the receive buffer the EDMAC is writing data to. The address that the EDMAC is outputting to the receive buffer might not match the read value of the RBWAR register during data reception. The RBWAR register is read-only. Do not write to this register.

31.2.22 Receive Descriptor Fetch Address Register (RDFAR)

Address(es): EDMAC0.RDFAR 4006 40CCh, PTPEDMAC.RDFAR 4006 44CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The RDFAR register indicates the start address of the last fetched receive descriptor when the EDMAC is fetching descriptor information from the receive descriptor. Check the contents of this register to identify which receive descriptor information the EDMAC is using for active processing. The address of the receive descriptor that the EDMAC is fetching might not match the read value of the RDFAR register during data reception. The RDFAR register is read-only. Do not write to this register.

31.2.23 Transmit Buffer Read Address Register (TBRAR)

Address(es): EDMAC0.TBRAR 4006 40D4h, PTPEDMAC.TBRAR 4006 44D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TBRAR register indicates the last address that the EDMAC read data from when reading data from the transmit buffer. Check the contents of this register to identify which address in the transmit buffer the EDMAC is reading from. The address that the EDMAC is outputting to the transmit buffer might not match the read value of the TBRAR register. The TBRAR register is read-only. Do not write to this register.

31.2.24 Transmit Descriptor Fetch Address Register (TDFAR)

Address(es): EDMAC0.TDFAR 4006 40D8h, PTPEDMAC.TDFAR 4006 44D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The TDFAR register indicates the start address of the last fetched transmit descriptor when the EDMAC is fetching descriptor information from the transmit descriptor. Check the contents of this register to identify which transmit descriptor information the EDMAC is using for active processing. The address of transmit descriptor that the EDMAC fetches might not match the read value of the TDFAR register. The TDFAR is read only. Do not write to this register.

31.3 Operation

The EDMAC transfers data according to the information written in the descriptor. Two descriptors are provided: transmit and receive. A descriptor includes the buffer size, address, and transmit or receive status. The EDMAC transmits or receives data continuously by using sequentially arranged descriptors.

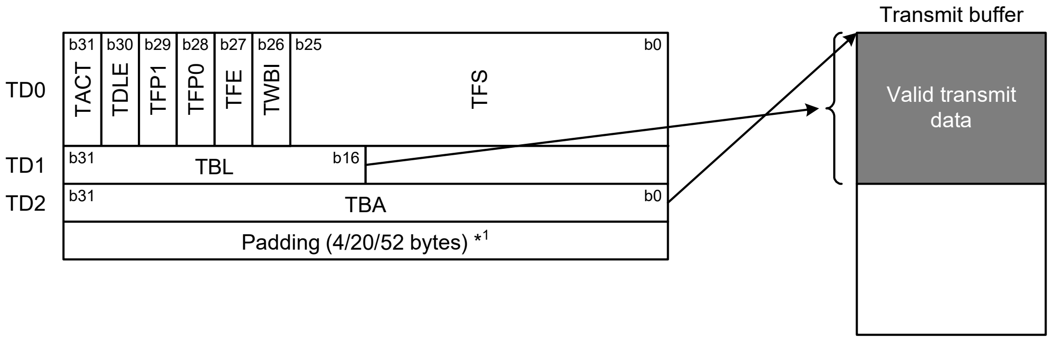
31.3.1 Descriptor Lists and Data Buffers

To transfer data using the EDMAC, create the transmit and receive descriptor lists in memory, set the start address of the transmit descriptor list in the TDLAR register, and set the start address of the receive descriptor list in the RDLAR register. Also, transmit and receive buffers associated with each descriptor are required.

Align the descriptor list on the appropriate address boundary according to the descriptor length set in the EDMR.DL[1:0] bits. The transmit buffer can be aligned on a word boundary, halfword boundary, or byte boundary. When the valid transmit buffer size is 16 bytes or less, align it on a 32-byte boundary. Align the receive buffer on a 32-byte boundary. Set different addresses for the transmit and receive descriptors and buffers for EDMAC0 and the PTPEDMAC.

31.3.1.1 Transmit descriptor

Figure 31.3 shows the relationship between a transmit descriptor and transmit buffer. A transmit descriptor consists of TD0 to TD2. The transmit frame and transmit buffer configuration can be specified as one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the transmit descriptor.



Note 1. The padding size is determined by the descriptor length (16/32/64 bytes).

Figure 31.3 Relationship between transmit descriptor and transmit buffer

(1) Transmit descriptor 0 (TD0)

Bit	Symbol	Bit name	Description	R/W
b25 to b0	TFS	Transmit Frame Status	Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: • For EDMAC0: TFS25 to TFS9: Reserved TFS8: Transmit abort was detected (value is equivalent to the EESR.TABT flag) TFS7 to TFS4: Reserved TFS3: No carrier was detected (value is equivalent to the EESR.CND flag) TFS2: Loss of carrier was detected (value is equivalent to the EESR.DLC flag) TFS1: Late collision during transmission was detected (value is equivalent to the EESR.CD flag) TFS0: Transmit retry over (value is equivalent to the EESR.TRO flag). When a bit sets to 1, it indicates that the associated error occurred during frame transmission. When any of the TFS bits sets to 1, the TFE bit also sets to 1. When any of bits TFS3 to TFS0 sets to 1, TFS8 also sets to 1. • For the PTPEDMAC: TFS25 to TFS9: Reserved TFS8: Transmit abort was detected (value is equivalent to the EESR.TABT flag) TFS7 to TFS1: Reserved TFS0: The transmission source MAC address of the transmit frame data did not match the set value (value is equivalent to the EESR.MACE flag). When a bit sets to 1, it indicates that the associated error occurred during frame transmission. When any of the TFS bits sets to 1, the TFE bit also sets to 1. When TFS0 sets to 1, TFS8 also sets to 1.	R/W
b26	TWBI	Write-Back Complete Interrupt Enable	0: Do not generate interrupt when write-back to this descriptor is complete 1: Generate interrupt when write-back to this descriptor is complete.	R/W
b27	TFE	Transmit Frame Error	0: Frame transmission is successfully complete 1: Error occurred during frame transmission (transmission aborted).	R/W
b29, b28	TFP[1:0]	Transmit Frame Position	b29 b28 0 0: Transmit buffer indicated in this descriptor is the middle of a transmit frame (frame information is incomplete) 0 1: Transmit buffer indicated in this descriptor is the end of a transmit frame (frame information is complete) 1 0: Transmit buffer indicated in this descriptor is the head of a transmit frame (frame information is incomplete) 1 1: Transmit buffer indicated in this descriptor is all of a transmit frame (one buffer per frame).	R/W
b30	TDLE	Transmit Descriptor List End	When this bit is 1, it indicates that this descriptor is the last in the descriptor list.	R/W
b31	TACT	Transmit Descriptor Valid	This bit indicates that this descriptor is valid.	R/W

Note: Bits for write-back are underlined.

TD0 specifies the transmit frame settings and indicates the status after transmission.

TFE bit (Transmit Frame Error)

When the TFE bit is 1, it indicates that any of the TFS bits is 1.

TFP[1:0] bits (Transmit Frame Position)

The TFP[1:0] bits indicate which part of a transmit frame corresponds to the transmit buffer indicated in this descriptor. The TFP[1:0] and TD1.TBL bit settings must be logically consistent in the previous and next descriptors.

TACT bit (Transmit Descriptor Valid)

The TACT bit indicates that this descriptor is valid. The TACT bit is set to 1 by software. This bit clears to 0 when the transmit frame is transferred or when the transmission is aborted.

(2) Transmit descriptor 1 (TD1)

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b31 to b16	TBL	Transmit Buffer Length	Specifies the valid byte length of the associated transmit buffer. Set a value equal to or greater than 1.	R/W

TD1 specifies the valid byte length of the transmit buffer.

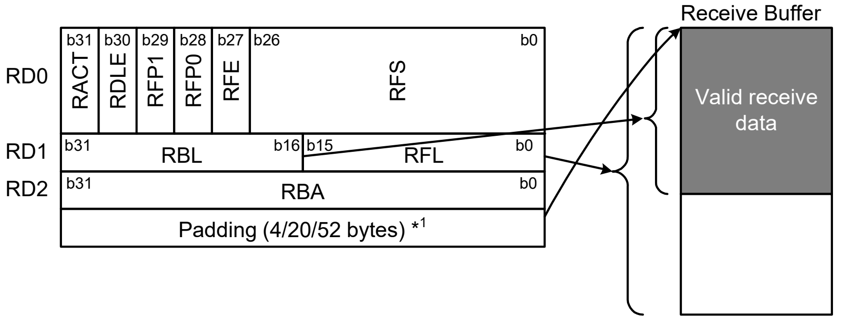
(3) Transmit descriptor 2 (TD2)

Bit	Symbol	Bit name	Description	R/W
b31 to b0	TBA	Transmit Buffer Address	Specifies the start address of the transmit buffer. When the TD1.TBL bit value is 1 to 16 bytes, align it on a 32-byte boundary.	R/W

TD2 specifies the start address of the transmit buffer.

31.3.1.2 Receive descriptor

Figure 31.4 shows the relationship between a receive descriptor and receive buffer. The receive frame and receive buffer configuration can be specified as one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the receive descriptor. If the receive buffer length (RBL) is set to 0, operation indicated in the descriptor is not guaranteed.



Note 1. The padding size is determined by the descriptor length (16/32/64 bytes).

Figure 31.4 Relationship between receive descriptor and receive buffer

(1) Receive descriptor 0 (RD0)

Bit	Symbol	Bit name	Description	R/W
b26 to b0	RFS	Receive Frame Status	Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: • For EDMAC0: RFS26 to RFS10: Reserved RFS9: Receive FIFO overflow (value is equivalent to the EESR.RFOF flag) RFS8: Receive abort was detected (value is equivalent to the EESR.RABT flag) RFS7: Multicast address frame was received (value is equivalent to the EESR.RMAF flag) RFS6 and RFS5: Reserved RFS4: Alignment error was detected (value is equivalent to the EESR.RRF flag) RFS3: Frame-too-long error (value is equivalent to the EESR.RTLF flag) RFS2: Frame-too-short error (value is equivalent to the EESR.RTSF flag) RFS1: PHY-LSI receive error (value is equivalent to the EESR.PRE flag) RFS0: CRC error (value is equivalent to the EESR.CERF flag). When a bit sets to 1, it indicates that the associated error occurred during frame reception. When any of the RFS bits sets to 1, the RFE bit also sets to 1. (Set the TRSCER register to select whether bits RFS7 and RFS4 are reflected in the RFE bit.) When any of bits RFS3 to RFS0 sets to 1, RFS8 also sets to 1. • For the PTPEDMAC: RFS26 to RFS10: Reserved RFS9: Receive FIFO overflow (value is equivalent to the EESR.RFOF flag) RFS8: Reserved RFS4: PTPV2 packet was received (value is equivalent to the EESR.PVER flag) (The PTPEDMAC can only receive PTP packets. If a non-PTP packet is received, the packet is not transferred to the PTPEDAC, and it is discarded.) RFS3 to RFS0: Type of the received PTP message (value is equivalent to the EESR.TYPE[3:0] flags). Each bit indicates the status of the received frame.	R/W
b27	RFE	Receive Frame Error	• For EDMAC0: 0: No error occurred in the received frame 1: Error occurred in the received frame. • For the PTPEDMAC: Reserved.	R/W
b29, b28	RFP[1:0]	Receive Frame Position	b29 b28 0 0: Receive buffer indicated in this descriptor is the middle of a receive frame (frame information is incomplete) 0 1: Receive buffer indicated in this descriptor is the end of a receive frame (frame information is complete) 1 0: Receive buffer indicated in this descriptor is the head of a receive frame (frame information is incomplete) 1 1: Receive buffer indicated in this descriptor is all of a receive frame (one buffer per frame).	R/W
b30	RDLE	Receive Descriptor List End	When this bit is 1, it indicates that this descriptor is the last in the descriptor list.	R/W
b31	RACT	Receive Descriptor Valid	Indicates that this descriptor is valid.	R/W

Note: Bits for write-back are underlined.

RD0 indicates the receive frame status.

RFE bit (Receive Frame Error)

When the RFE bit is 1, it indicates that any of the RFS bits is 1. Set the TRSCER register to select whether the RFS7 and RFS4 bits of EDMAC0 are reflected in the RFE bit.

RFP[1:0] bits (Receive Frame Position)

The RFP[1:0] bits indicate which part of a receive frame corresponds to the receive buffer indicated in this descriptor.

RACT bit (Receive Descriptor Valid)

The RACT bit indicates that this descriptor is valid. The RACT bit is set to 1 by software. This bit clears to 0 when all data is transferred to the receive buffer indicated in RD2 or when the receive buffer becomes full.

(2) Receive descriptor 1 (RD1)

Bit	Symbol	Bit name	Description	R/W
b15 to b0	RFL	Receive Frame Length	Specifies the length (number of bytes) of the receive frame stored in the buffer. This does not include the number of bytes for padding set in the RPADIR register. These bits are written back to the descriptor associated with the end of a frame.	R/W
b31 to b16	RBL	Receive Buffer Length	Specifies the byte length of the associated receive buffer. Set an integral multiple of 32 as the buffer length.	R/W

Note: Bits for write-back are underlined.

RD1 specifies the receive buffer length. When reception is complete, the receive frame length is written back.

(3) Receive descriptor 2 (RD2)

Bit	Symbol	Bit name	Description	R/W
b31 to b0	RBA	Receive Buffer Address	Specifies the start address of the receive buffer. Align the buffer address on a 32-byte boundary.	R/W

RD2 specifies the start address of the receive buffer.

31.3.2 Transmission

When the EDTRR.TR bit is set to 1 while the ETHERC0.ECMR.TE bit is 1, the EDMAC reads the descriptor following the previously used descriptor in the transmit descriptor list (or the descriptor indicated in the TDLAR register after a reset). When the TACT bit is 1 in the transmit descriptor (TD0), the EDMAC sequentially reads transmit data from the start address of the transmit buffer indicated in transmit descriptor 2 (TD2) and transfers it to the ETHERC through the transmit FIFO. The ETHERC creates a transmit frame and starts transmission to the MII or RMII. When all data indicated in the TD1.TBL bit is transferred, write-back is performed based on the TD0.TFP[1:0] bit setting as follows:

- When the TD0.TFP[1:0] bits are 00b or 10b (frame is incomplete), the TD0.TACT bit is written back
- When the TD0.TFP[1:0] bits are 01b or 11b (frame is complete), the TD0.TACT, TD0.TFS, and TD0.TFE bits are written back.

When the TD0.TACT bit in the read descriptor is 1, the EDMAC continues reading descriptors and transmit frames. When the TD0.TACT bit in the read descriptor is 0, the EDMAC sets the EDTRR.TR bit to 0 and stops transmission.

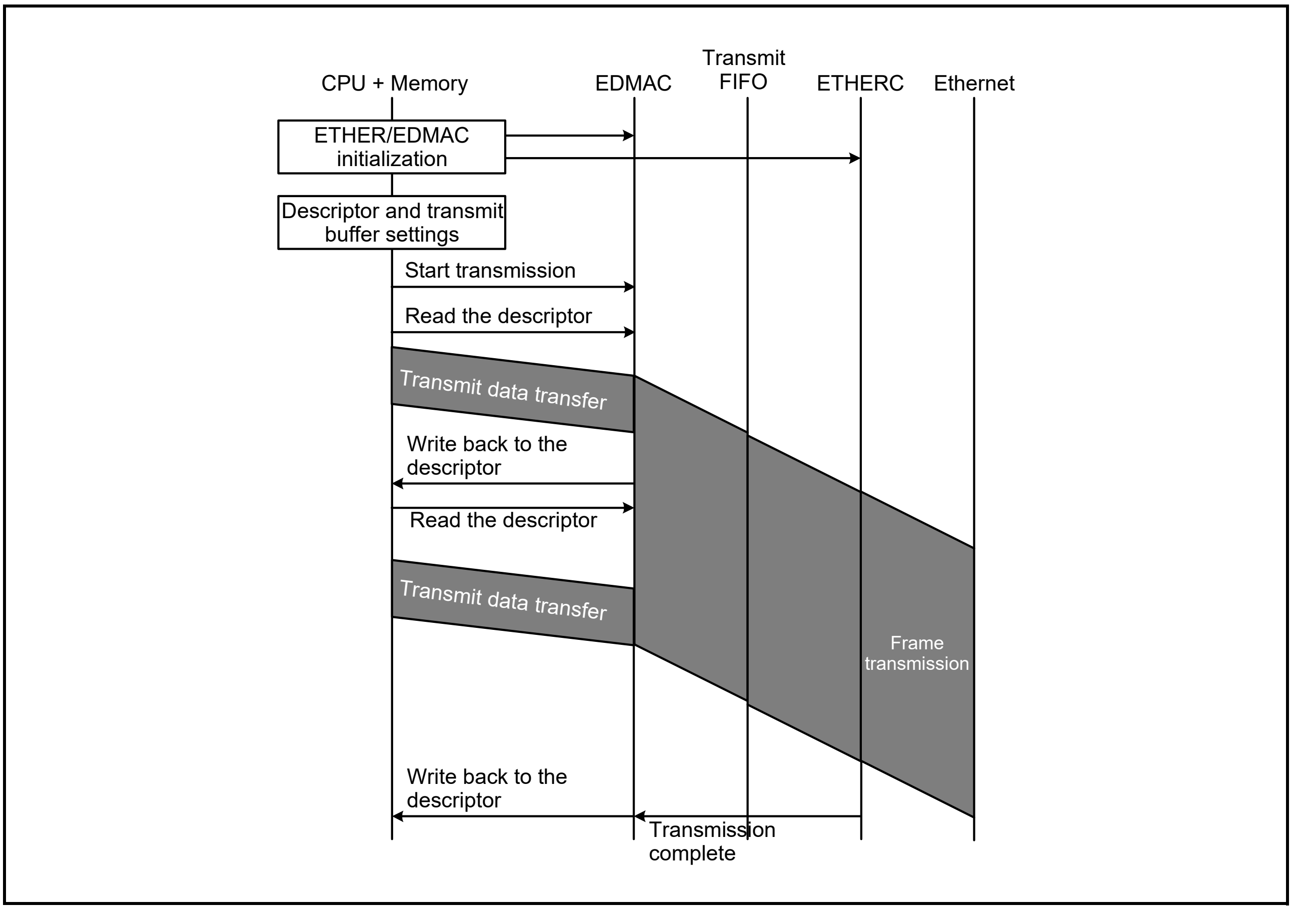


Figure 31.5 Example of transmission flow

31.3.3 Reception

When the EDRRR.RR bit is set to 1 while the ETHERC0.ECMR.RE bit is 1, the EDMAC reads the receive descriptor following the previously used descriptor (or the descriptor indicated in the RDLAR register after a reset) and then waits for reception. When the RD0.RACT bit is 1, if the data stored in the receive FIFO is 32 bytes or more, or if the end byte of the frame is stored in the receive buffer, the EDMAC transfers data from the receive FIFO to the receive buffer indicated in receive descriptor 2 (RD2).

If the data length of the received frame is longer than the buffer length set in the RBL bit in receive descriptor 1 (RD1), the EDMAC writes back 10b or 00b to the RD0.RFP[1:0] bits and 0 to the RD0.RACT bit when the receive buffer becomes full, and then the EDMAC reads the next data. After that, the EDMAC transfers data to another receive buffer.

When the frame reception is complete or when the frame reception is aborted by an error, the EDMAC writes back 11b or 01b to the RD0.RFP[1:0] bits, 0 to the RD0.RACT bit, and the receive frame length to the RD1.RFL bit. When the RMCR.RNR bit is 1, the EDMAC reads the next descriptor and waits for reception. When the RNR bit is 0, the EDMAC sets the EDRRR.RR bit to 0 and stops reception.

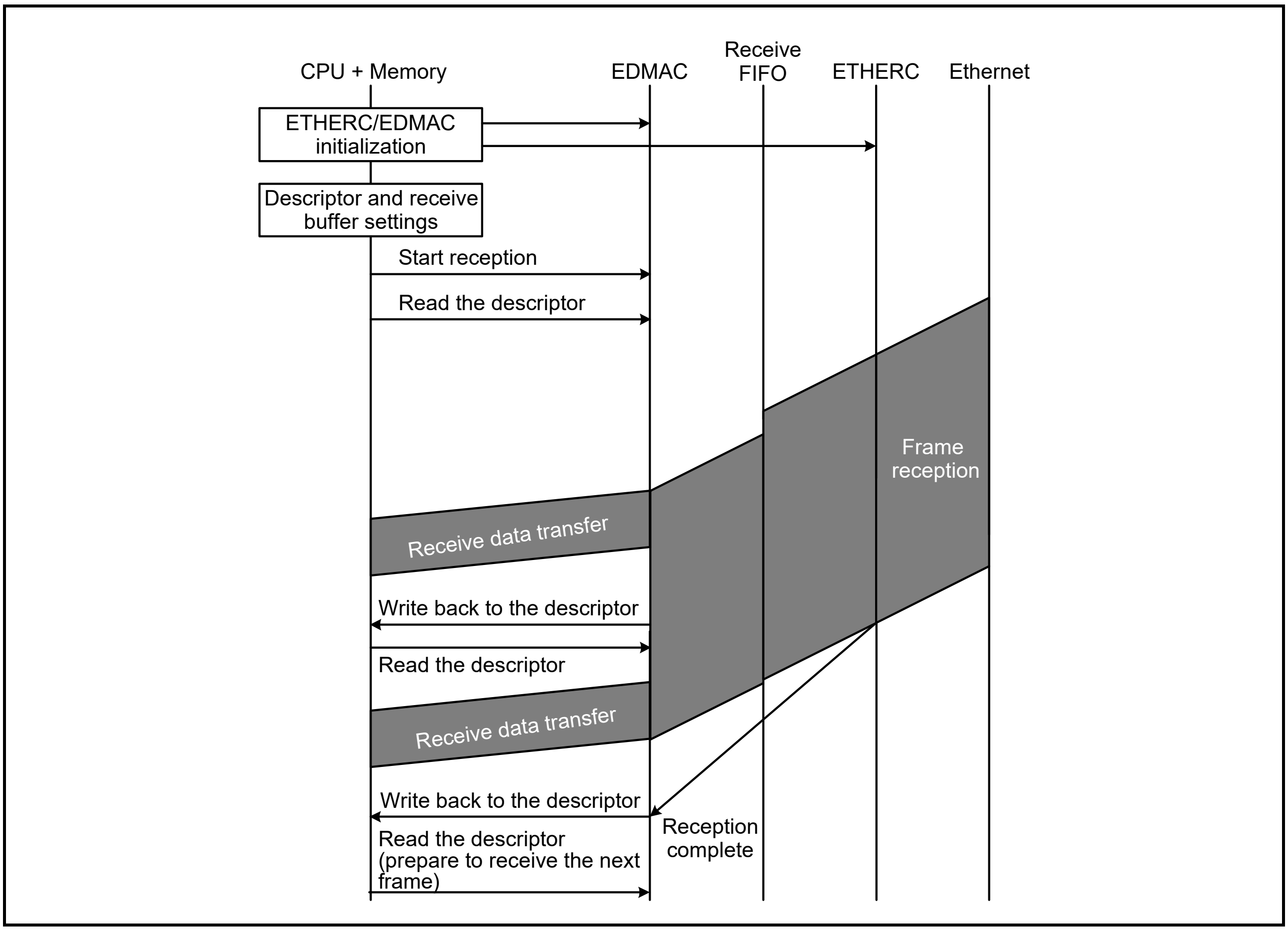


Figure 31.6 Example of reception flow

31.3.4 Multi-Buffer Frame Transmission

31.3.4.1 Error processing while transmitting multi-buffer frame

If an error occurs during multi-buffer frame transmission, the EDMAC performs the processing shown in Figure 31.7. In the figure, when the TACT bit of transmit descriptor 0 (TD0) is 0, the descriptor indicates that all data in the buffer is successfully transmitted. When the TACT bit is 1, the descriptor indicates that data in the buffer is not yet transmitted. If a frame transmit error*[1] occurs in the head or middle of the frame while the TD0.TACT bit is 1, the EDMAC stops data transmission from the transmit FIFO and EDMAC data transfer, and sets the TD0.TACT bit to 0.

After that, the EDMAC reads the next descriptor to see if the descriptor indicates the middle of the frame (TD0.TFP[1:0] bits are 00b) or the end of the frame (TD0.TFP[1:0] bits are 01b). When the descriptor indicates the middle of the frame, the EDMAC sets the TD0.TACT bit to 0 and reads the next descriptor. When the descriptor indicates the end of the frame, in addition to setting the TD0.TACT bit to 0, the EDMAC also writes back to the TD0.TFE and TD0.TFS bits.

After an error occurs, data in the buffer is not transmitted until write-back to the descriptor for the end of the frame. When the associated transmit error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor for the end of the frame.

Note 1. For EDMAC0, a transmit retry-over condition, late collision, or loss of carrier is detected, or a carrier is not detected. For the PTPEDMAC, the MAC address does not match the set value.

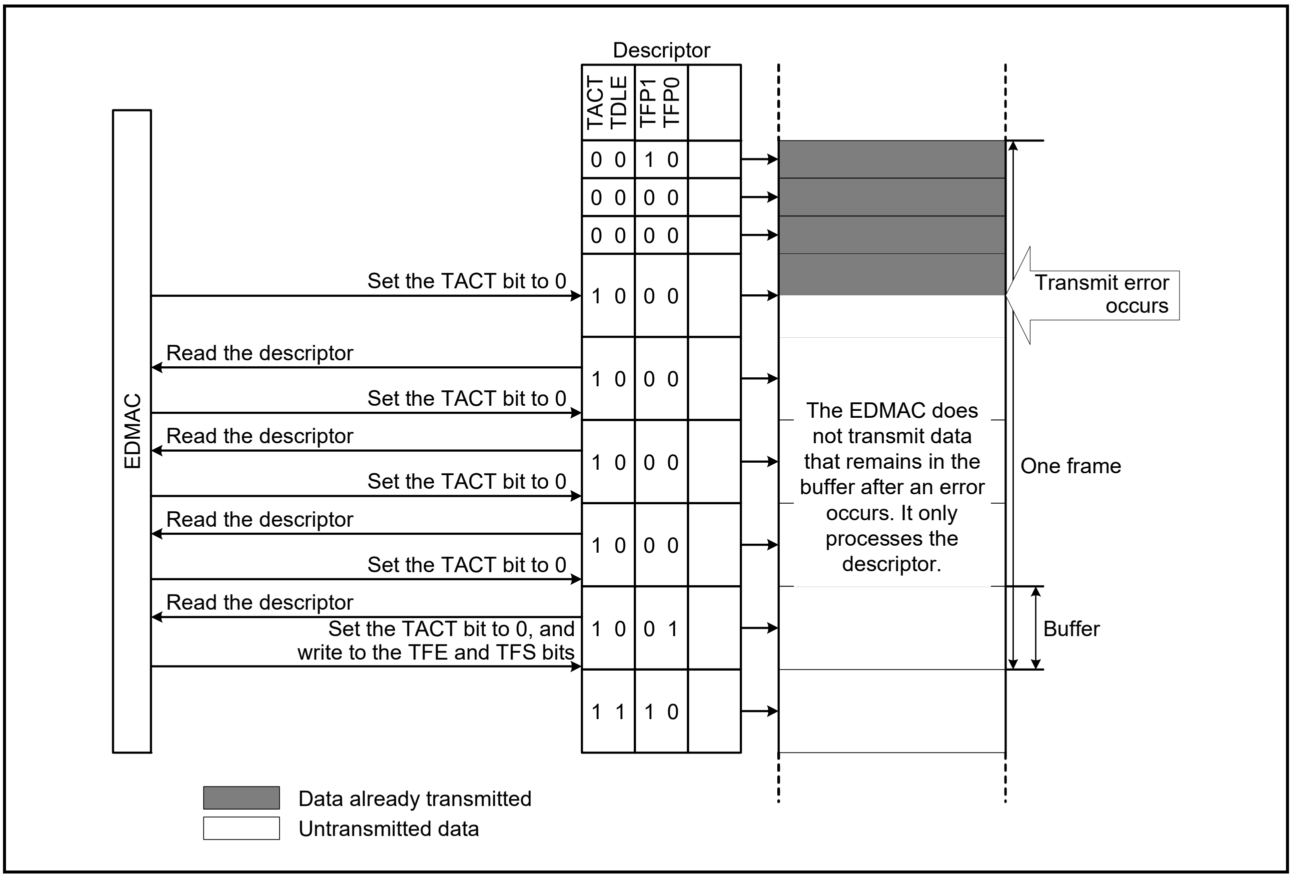


Figure 31.7 EDMAC operation after transmit error occurs

31.3.4.2 Error processing while receiving multi-buffer frame

If an error occurs during multi-buffer frame reception, the EDMAC performs the processing shown in Figure 31.8. In the figure, when the RACT bit of receive descriptor 0 (RD0) is 0, the descriptor indicates that data was successfully received in the buffer. When the RACT bit is 1, the descriptor indicates that data is not yet received in the buffer. If a frame receive error*1 occurs, the EDMAC stops receiving new data, but it transfers data that is already stored in the receive FIFO to the receive buffer.

When the receive buffer becomes full during transfer, the EDMAC sets the RACT bit to 0 and the RFP[1:0] bits to 10b or 00b and reads the next descriptor. After all data in the receive FIFO is transferred, the EDMAC writes back the status to the descriptor.

When the associated receive error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor. When there is a request to receive a new frame, the EDMAC continues reception using the descriptor following the descriptor where the error occurred.

Note 1. For EDMAC0, a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or alignment error is detected. For the PTPEDMAC, a parity error is detected.

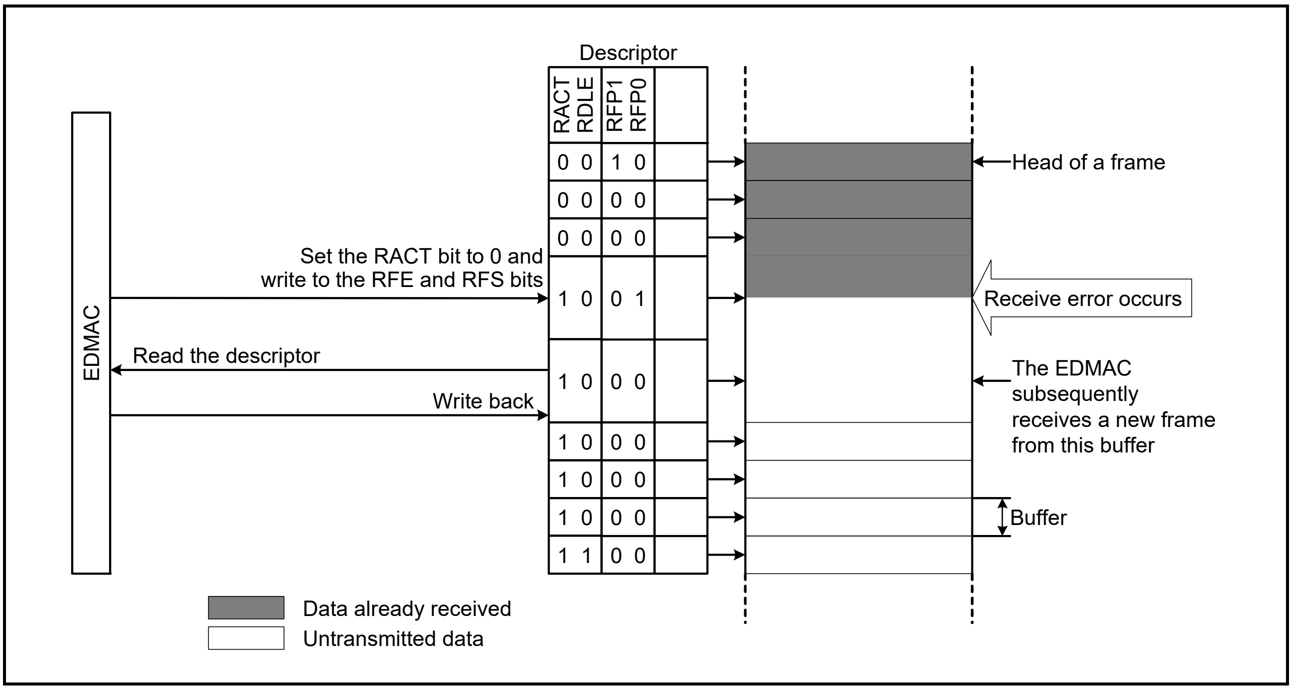


Figure 31.8 EDMAC operation after receive error occurs

31.3.5 EDMAC Channel Priority

This section describes the priority of the two EDMAC channels (EDMAC0, PTPEDMAC). Each time transfer of one channel is complete, that channel takes the lowest priority. This operation is shown in Figure 31.9. After a reset, the priority is EDMAC0 > PTPEDMAC.

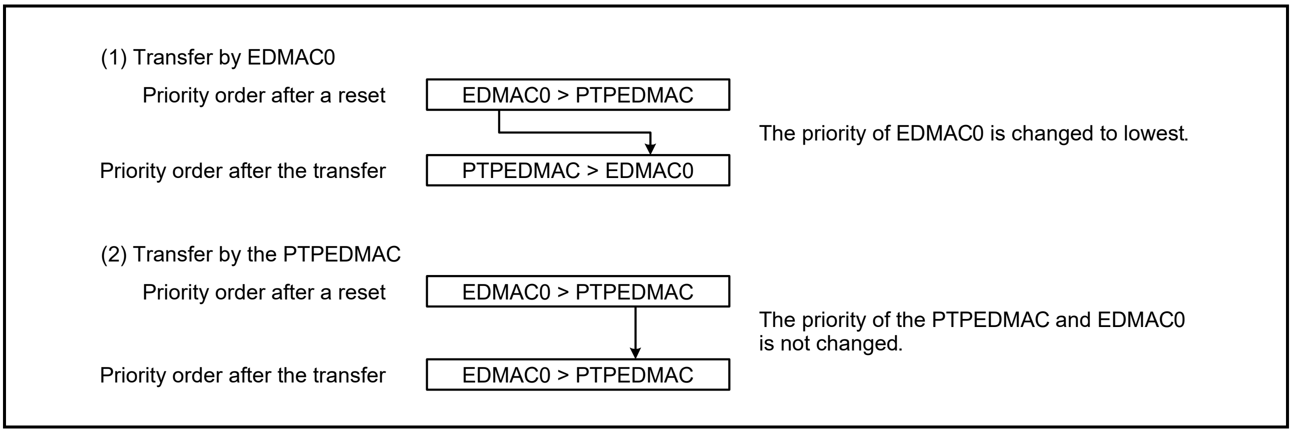


Figure 31.9 Channel priority order

Figure 31.10 shows the change in the channel priority order when transfer requests are concurrently generated to EDMAC0 and the PTPEDMAC.

The operations in the figure are as follows:

1. Transfer requests are concurrently sent to EDMAC0.
2. The EDMAC0 starts a transfer.
3. After EDMAC0 ends the transfer, the priority of EDMAC0 is changed to the lowest.
4. Transfer requests are concurrently sent to EDMAC0 and the PTPEDMAC.
5. Because PTPEDMAC has higher priority than the EDMAC0 at this time, PTPEDMAC starts a transfer and the EDMAC0 waits.
6. After PTPEDMAC ends the transfer, the priority of PTPEDMAC is changed to the lowest.
7. EDMAC0 starts a transfer.
8. After the EDMAC0 ends the transfer, the priority of EDMAC0 is changed to the lowest.

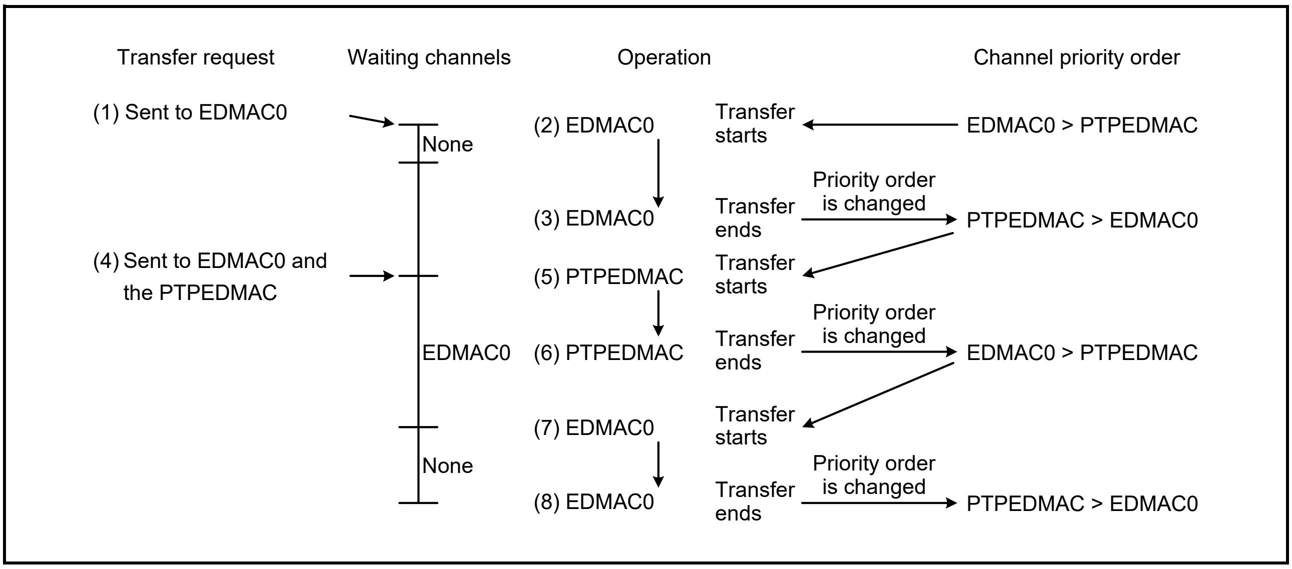


Figure 31.10 Example of channel priority order change

31.4 Interrupts

When any of the status flags in the EESR register sets to 1 while the associated interrupt request enable bit in the EESIPR register is 1, EDMAC0 issues an ETHER_EINT0 interrupt request or the PTPEDMAC issues an ETHER_PINT interrupt request to the CPU.

31.5 Usage Notes

31.5.1 Settings for the Module-Stop Function

The following bits in Module Stop Control Register B (MSTPCRB) enable or disable EDMAC module operation:

- The MSTPB15 bit in enables or disables ETHERC0 and EDMAC0 operation
- The MSTPCRB.MSTPB13 bit enables or disables EPTPC and PTPEDMAC operation.

The modules are initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

Note: When EPTPC and PTPEDMAC operation is enabled (MSTPB13 = 0), some registers in the EPTPC module become inaccessible depending on the combination of the MSTPB15 bit and EPTPC bypass bit (BYPASS.BYPASS0) settings. See section 30.6.1, Constraints on Register Access.

31.5.2 Stopping the EDMAC during Operation

When stopping EDMAC operation by using a Sleep instruction or the module-stop function while the EDMAC is running, confirm that the EDTRR.TR and EDRRR.RR bits are 0. If the EDMAC is stopped while the EDTRR.TR or EDRRR.RR bit is 1, the data for the frame that is being transmitted or received might not be complete, and EDMAC operation after exiting Sleep mode or the module-stop state is not guaranteed.

32. USB 2.0 Full-Speed Module (USBFS)

32.1 Overview

The MCU provides a USB 2.0 Full-Speed module (USBFS) that operates as a host or device controller compliant with the Universal Serial Bus (USB) specification revision 2.0. The host controller supports USB 2.0 full-speed and low-speed transfers, and the device controller supports USB 2.0 full-speed transfers. The USBFS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification.

The USBFS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

Table 32.1 lists the USBFS specifications, Figure 32.1 shows a block diagram, and Table 32.2 lists the I/O pins.

Table 32.1 USBFS specifications

Parameter	Specifications
Features	• USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, device controller, and On-The-Go (OTG) functions (one channel) • Host and device controller can be switched by software • Self-power or bus power mode selectable.
	Host controller features: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Communications with multiple peripheral devices connected through a single hub.
	Device controller features: • Full-speed transfer (12 Mbps)*1 • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation.
Supported transfer types	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer.
Pipe configuration	• FIFO buffer for USB communication • Up to 10 pipes selectable, including the Default Control Pipe (DCP) • Pipes 1 to 9 assignable to any endpoint number.
	Transfer conditions specifiable for each pipe: • Pipe 0: Control transfer with 64-byte single buffer • Pipes 1 and 2: Selectable to bulk transfer with 64-byte double buffer or isochronous transfer with 256-byte double buffer • Pipes 3 to 5: Bulk transfer with 64-byte double buffer • Pipes 6 to 9: Interrupt transfer with 64-byte single buffer.
Other features	• Reception end function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Automatic clearing of the FIFO buffer after the data for the pipe specified in the DnFIFO port (n = 0, 1) is read (DCLRM) • NAK setting function for response PID generated on transfer end (SHTNAK) • On-chip pull-up and pull-down resistors for D+ and D-.
Module-stop function	Module-stop state can be set to reduce power consumption

Note 1. Low-speed transfer (1.5 Mbps) is not supported.

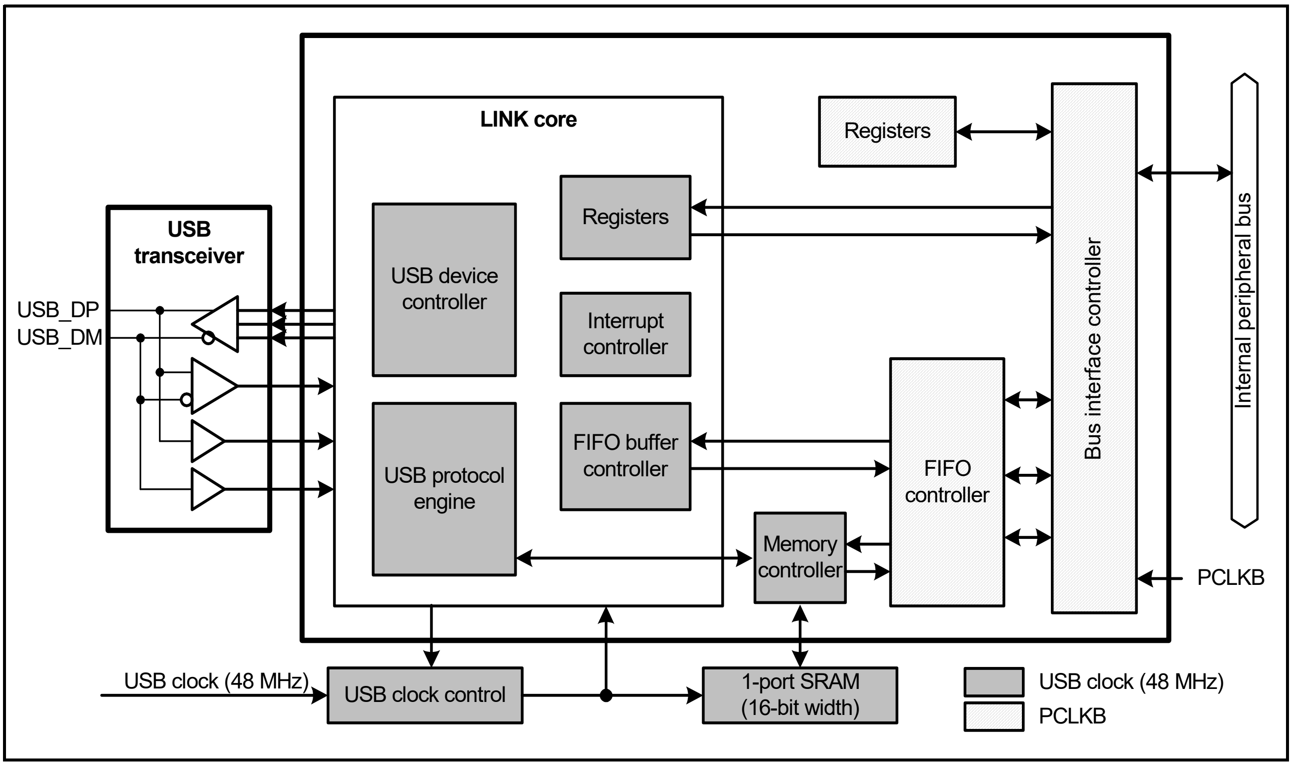


Figure 32.1 USBFS block diagram

Table 32.2 USBFS pin configuration

Port	Pin name	I/O	Function
USBFS	USB_DP	I/O	D+ I/O for the on-chip USB transceiver. Must be connected to the D+ data line of the USB bus.
	USB_DM	I/O	D- I/O pin for the on-chip USB transceiver. Must be connected to the D- data line of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Must be connected to VBUS signal on the USB bus. VBUS pin status (connected or disconnected) can be detected when the USBFS is a device controller.*1
	USB_EXICEN	Output	Low-power control signal for the OTG power supply IC
	USB_VBUSEN	Output	VBUS (5 V) enable signal for the external power supply IC
	USB_OVRCURA USB_OVRCURB	Input	Overcurrent pins for USBFS. Must be connected to external overcurrent detection signals. When the OTG power supply chip is connected, must be connected to the VBUS comparator signals.
	USB_ID	Input	Must be connected to MicroAB connector ID input signal in OTG mode
Shared	VCC_USB	Input	USB transceiver input supply voltage
	VSS_USB	Input	USB ground pin

Note 1. P407 is 5-V tolerant.

32.2 Register Descriptions

32.2.1 System Configuration Control Register (SYSCFG)

Address(es): USBFS.SYSCFG 4009 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	USBE	USBFS Operation Enable	0: Disable 1: Enable.	R/W
b3, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Disable line pull-up 1: Enable line pull-up.	R/W
b5	DRPD	D+/D- Line Resistor Control	0: Disable line pull-down 1: Enable line pull-down.	R/W
b6	DCFM	Controller Function Select	0: Select device controller 1: Select host controller.	R/W
b9 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable	0: Stop clock supply to the USBFS 1: Enable clock supply to the USBFS.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: After writing 1 to the SCKE bit, read it to confirm that it is set to 1.

USBE bit (USBFS Operation Enable)

The USBE bit enables or disables operation of the USBFS.

Changing the USBE bit from 1 to 0 initializes the bits listed in Table 32.3. Only change this bit while the SCKE bit is 1. In host controller mode, this bit must be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flag chattering, and confirming that the USB bus state is stable.

Table 32.3 Registers initialized by writing 0 to the SYSCFG.USBE bit

Selected function	Register	Bit	Remarks
Device controller	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	-
	INTSTS0	DVSQ[2:0]	Value is saved in host controller mode
	USBADDR	USBADDR[6:0]	Value is saved in host controller mode
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller	DVSTCTR0	RHST[2:0]	-
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode

DPRPU bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBFS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 1 in device controller mode and to 0 in host controller mode.

DRPD bit (D+/D- Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D- lines in host controller mode.

Set this bit to 1 in host controller mode and to 0 in device controller mode.

DCFM bit (Controller Function Select)

The DCFM bit selects the host or device function of the USBFS.

Only change this bit when the DPRPU and DRPD bits are both 0.

SCKE bit (USB Clock Enable)

The SCKE bit stops or enables the 48-MHz clock supply to the USBFS.

When this bit is 0, only SYSCFG is permitted to be read from and written to; the other registers related to the USB should not be read from or written to.

32.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USBFS.SYSSTS0 4009 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor	Indicates the status of the USB data lines, see Table 32.4	R
b2	IDMON	External ID0 Input Pin Monitor	0: USB_ID pin is low 1: USB_ID pin is high.	R
b4, b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b5	SOFEA	Active Monitor When the Host Controller Is Selected	0: SOF output stopped 1: SOF output operating.	R
b6	HTACT	USB Host Sequencer Status Monitor	0: Host sequencer completely stopped 1: Host sequencer not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be changed.	R
b15, b14	OVCMON[1:0]	External USB_OVRCURA/ USB_OVRCURB Input Pin Monitor	OVCMON[1] indicates the USB_OVRCURA pin status. OVCMON[0] indicates the USB_OVRCURB pin status.	R

Note 1. Depends on the status of the USB_OVRCURA, USB_OVRCURB, and USB_ID pins.

LNST[1:0] bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines, D+ and D-. For details, see Table 32.4.

In device controller mode, read the LNST[1:0] bits after connection processing (SYSCFG.DPRPU bit = 1). In host controller mode, read them after enabling pull-down of the lines (SYSCFG.DRPD bit = 1).

Table 32.4 Status of the USB data bus lines (D+ and D-) (1 of 2)

LNST[1:0] bits	During full-speed operation	During low-speed operation
00b	SE0	SE0
01b	J-State	K-State
10b	K-State	J-State

Table 32.4 Status of the USB data bus lines (D+ and D-) (2 of 2)

LNST[1:0] bits	During full-speed operation	During low-speed operation
11b	SE1	SE1

SOFEA bit (Active Monitor When the Host Controller Is Selected)

The SOFEA bit is used in host controller mode to check whether the output of the last SOF is complete when the USBFS is suspended because of a 0 setting to the DVSTCTR0.UACT bit.

In host controller mode, check that both the HTACT and SOFEA bits are 0 before setting the SYSCFG.USBE bit to 0 to stop the USBFS or setting the SYSCFG.SCKE bit to 0 to stop the clock signal supply during communication.

HTACT bit (USB Host Sequencer Status Monitor)

The HTACT bit is set to 0 when the host sequencer of the USBFS is completely stopped.

In host controller mode, check that the HTACT bit is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBFS in the suspended state or setting the SCKE bit to 0 to stop the clock signal supply during communication.

OVCMON[1:0] bits (External USB_OVRCURA/ USB_OVRCURB Input Pin Monitor)

The OVCMON[1:0] bits indicate the status of the overcurrent signals from an external power supply IC.

32.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USBFS.DVSTCTR0 4009 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status	• In host controller mode: b2 b0 0 0 0:Communication speed indeterminate (powered state or no connection) 1 x x:USB bus reset in progress 0 0 1:Low-speed connection 0 1 0:Full-speed connection. • In device controller mode b2 b0 0 0 0: Communication speed indeterminate 0 0 1: USB bus reset in progress 0 1 0: USB bus reset in progress or full-speed connection.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Disable downstream port (disable SOF transmission) 1: Enable downstream port (enable SOF transmission).	R/W
b5	RESUME	Resume Output	0: Do not output resume signal 1: Output resume signal.	R/W
b6	USBRST	USB Bus Reset Output	0: Do not output USB bus reset signal 1: Output USB bus reset signal.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Disable downstream port remote wakeup 1: Enable downstream port remote wakeup.	R/W
b8	WKUP	Wakeup Output	0: Do not output remote wakeup signal 1: Output remote wakeup signal.	R/W
b9	VBUSEN	USB_VBUSEN Output Pin Control	0: Output low on external USB_VBUSEN pin 1: Output high on external USB_VBUSEN pin.	R/W
b10	EXICEN	USB_EXICEN Output Pin Control	0: Output low on external USB_EXICEN pin 1: Output high on external USB_EXICEN pin.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	Use this bit when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the Suspend state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

The USBFS controller does not support low-speed connections in device controller mode. When this value is read, abnormal connection processing must be executed in higher level application software.

RHST[2:0] bits (USB Bus Reset Status)

The RHST[2:0] bits indicate the status of the USB bus reset.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] bits to set to 100b. When 0 is written to the USBRST bit and the USBFS ends the SE0 state, the RHST[2:0] bits update to a new value.

In device controller mode, if the USBFS detects a USB bus reset, the RHST[2:0] bits indicate 010b if the DPRPU bit is 1, and a DVST interrupt is generated.

UACT bit (USB Bus Enable)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBFS starts SOF packet output within one frame period after the UACT bit is set to 1. When UACT is set to 0, the USBFS enters the idle state after the SOF packet output.

The USBFS sets the UACT bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (when UACT = 1)
- An EOFERR interrupt is detected during communication (when UACT = 1).

Always write 1 to the UACT bit at the end of the USB bus reset processing (writing 0 to the USBRST bit) or at the end of resume processing from the suspended state (writing 0 to the RESUME bit).

In device controller mode, always set this bit to 0.

RESUME bit (Resume Output)

The RESUME bit controls the resume signal output in host controller mode.

When this bit is set to 1, the USBFS drives the USB port to the K-state and outputs the resume signal. The USBFS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB Suspend state.

The USBFS continues outputting the K-state while the RESUME bit is 1, until the bit is cleared to 0 by software. The RESUME bit must be 1 (resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the Suspend state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).

Always set this bit to 0 in device controller mode.

USBRST bit (USB Bus Reset Output)

The USBRST bit controls the output of the USB bus signal in host controller mode. When this bit set to 1, the USBFS drives the USB port to the SE0 state to reset the USB bus. The USBFS continues outputting SE0 while the USBRST bit is 1, until the bit is cleared to 0 by software. The USBRST bit must be 1 (USB bus reset period) for the time defined in the USB 2.0 specification. Writing 1 to the USBRST bit during communication (UACT bit = 1) or during resume processing (RESUME bit = 1) prevents the USBFS from starting USB bus reset processing until both the UACT and RESUME bits become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).

Always set this bit to 0 in device controller mode.

RWUPE bit (Wakeup Detection Enable)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBFS detects a remote wakeup signal (K-state for 2.5 µs) from a downstream peripheral device, and performs resume processing, driving the K-state. When the RWUPE bit is set to 0, the USBFS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port.

Do not stop the internal clock when the RWUPE bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in device controller mode.

WKUP bit (Wakeup Output)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBFS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBFS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification specifies that the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USBFS writes 1 to the WKUP bit immediately after detecting the Suspend state, the K-state is output after 2 ms.

Only write 1 to the WKUP bit when the device is in the Suspend state (INTSTS0.DVSQ[2:0] bits = 1xxb) and the USB host enables the remote wakeup signal (RWUPE = 1). Do not stop the internal clock while this bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in host controller mode.

HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode.

If the HNPBTOA bit is 1, the internal function control maintains the Suspend state until HNP processing ends, even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. Resume interrupts (RESM) are not generated even if a falling edge of D+ is detected.

The HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is cleared to 0 by software because the HNP processing times out.

32.2.4 CFIFO Port Register (CFIFO/CFIFOL) D0FIFO Port Register (D0FIFO/D0FIFOL) D1FIFO Port Register (D1FIFO/D1FIFOL)

(1) When the MBW bit is 1

Address(es): USBFS.CFIFO 4009 0014h, USBFS.D0FIFO 4009 0018h, USBFS.D1FIFO 4009 001Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FIFOPORT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(2) When the MBW bit is 0

Address(es): USBFS.CFIFOL 4009 0014h, USBFS.D0FIFOL 4009 0018h, USBFS.D1FIFOL 4009 001Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	FIFOPORT[15:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	FIFOPORT[15:0]*1	FIFO Port	Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW, D0FIFOSEL.MBW, and D1FIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND) in the associated port select register. See Table 32.5 and Table 32.6.

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO.

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the port select register
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU.

FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 32.5 and Table 32.6.

Table 32.5 Endian operation in 16-bit access

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 32.6 Endian operation in 8-bit access

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	Access prohibited[*1]	N + 0 data
1	Access prohibited[*1]	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

32.2.5 CFIFO Port Select Register (CFIFOSEL)
D0FIFO Port Select Register (D0FIFOSEL)
D1FIFO Port Select Register (D1FIFOSEL)

CFIFOSEL

Address(es): USBFS.CFIFOSEL 4009 0020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CURPIPE [3:0]	CFIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (Default Control Pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP Is Selected	0: Select reading from the FIFO buffer 1: Select writing to the FIFO buffer.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: Do not rewind buffer pointer 1: Rewind buffer pointer.	W[*1]
b15	RCNT	Read Count Mode	0: The DTLN[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all receive data is read from the CFIFO. In double buffer mode, the DTLN[8:0] value is cleared when all data is read from only a single plane. 1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

ISEL bit (CFIFO Port Access Direction When DCP Is Selected)

After writing a new value to the ISEL bit with the DCP as the selected pipe, read the ISEL bit to check that the written value agrees with the read value before proceeding to the next process. Set the ISEL and CURPIPE[3:0] bits simultaneously.

MBW bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is receiving, set the CURPIPE[3:0] and MBW bits simultaneously. After a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

D0FIFOSEL, D1FIFOSEL

Address(es): USBFS.D0FIFOSEL 4009 0028h, USBFS.D1FIFOSEL 4009 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CURPIPE [3:0]	FIFO Port Access Pipe Specification	b3 b0 0 0 0 0: No pipe specification 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: Disable DMA/DTC transfer request 1: Enable DMA/DTC transfer request.	R/W
b13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read	0: Disable auto buffer clear mode 1: Enable auto buffer clear mode.	R/W
b14	REW	Buffer Pointer Rewind	0: Do not rewind buffer pointer 1: Rewind buffer pointer.	R/W[*1]
b15	RCNT	Read Count Mode	0: Clear DTLN[8:0] bits in (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) when all receive data is read from DnFIFO (after read of a single plane in double buffer mode) 1: Decrement DTLN[8:0] bits each time receive data is read from DnFIFO. n = 0, 1.	R/W

Note 1. Only 0 can be read.

The same pipe must not be specified in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected. The pipe number must not be changed while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the DnFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

MBW bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the DnFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. Set the CURPIPE[3:0] and MBW bits simultaneously. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMA or DTC transfer requests. To enable DMA or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits. To change the CURPIPE[3:0] setting, first set this bit to 0.

DCLRM bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBFS sets the BCLR bit in the FIFO port control register to 1.

When using the USBFS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the CFIFOCTR.DTLN bit. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

32.2.6 CFIFO Port Control Register (CFIFOCTR)
D0FIFO Port Control Register (D0FIFOCTR)
D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USBFS.CFIFOCTR 4009 0022h, USBFS.D0FIFOCTR 4009 002Ah, USBFS.D1FIFOCTR 4009 002Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length	Indicates the receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready	0: FIFO port access disabled 1: FIFO port access enabled.	R
b14	BCLR	CPU Buffer Clear	0: No operation 1: Clear FIFO buffer on the CPU side.	R/W[1]
b15	BVAL	Buffer Memory Valid Flag	0: Invalid (writing 0 has no effect) 1: Writing ended.	R/W

Note 1. Only 0 can be read.

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[8:0] bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0
  The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.
  While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.

- RCNT = 1
  The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when MBW = 0, and by 2 when MBW = 1.
  The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY bit in the FIFO port control register is 1 (set by the USBFS).

BVAL flag (Buffer Memory Valid Flag)

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU side for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer.

The USBFS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

32.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USBFS.INTENB0 4009 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W

Bit	Symbol	Bit name	Description	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b14	RSME	Resume Interrupt Enable*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15	VBSE	VBUS Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W

Note 1. The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register sets to 1 and the associated interrupt request enable bit setting in the INTENB0 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBFS interrupt is requested.

32.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USBFS.INTENB1 4009 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W

Note: The bits in INTENB1 can only be set to 1 in host controller mode. Do not set these bits to 1 in device controller mode.

INTENB1 specifies the interrupt masks in host controller mode and for the setup transaction.

When a status flag in the INTSTS1 register sets to 1 and the associated interrupt request enable bit setting in the INTENB1 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBFS interrupt is requested.

Do not enable interrupts in device controller mode.

32.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USBFS.BRDYENB 4009 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B RDYE	PIPE8B RDYE	PIPE7B RDYE	PIPE6B RDYE	PIPE5B RDYE	PIPE4B RDYE	PIPE3B RDYE	PIPE2B RDYE	PIPE1B RDYE	PIPE0B RDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0	0: Disable interrupt request 1: Enable interrupt request.	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2	0: Disable interrupt request 1: Enable interrupt request.	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3	0: Disable interrupt request 1: Enable interrupt request.	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Disable interrupt request 1: Enable interrupt request.	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Disable interrupt request 1: Enable interrupt request.	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Disable interrupt request 1: Enable interrupt request.	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Disable interrupt request 1: Enable interrupt request.	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8	0: Disable interrupt request 1: Enable interrupt request.	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when a BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register sets to 1 and the associated PIPEnBRDYE bit (n = 0 to 9) setting in the BRDYENB register is 1, the INTSTS0.BRDY flag sets to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBFS generates a BRDY interrupt request. While at least one PIPEnBRDY bit indicates 1, the USB generates the BRDY interrupt request when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by software.

32.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USBFS.NRDYENB 4009 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0	0: Disable interrupt request 1: Enable interrupt request.	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2	0: Disable interrupt request 1: Enable interrupt request.	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3	0: Disable interrupt request 1: Enable interrupt request.	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Disable interrupt request 1: Enable interrupt request.	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Disable interrupt request 1: Enable interrupt request.	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Disable interrupt request 1: Enable interrupt request.	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Disable interrupt request 1: Enable interrupt request.	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8	0: Disable interrupt request 1: Enable interrupt request.	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NRDYENB register enables or disables the INTSTS0.NRDY bit to be set to 1 when a NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register sets to 1 and the associated PIPEnNRDYE (n = 0 to 9) bit setting in the NRDYENB register is 1, the INTSTS0.NRDY flag sets to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBFS generates a NRDY interrupt request. While at least one PIPEnNRDY bit indicates 1, the USBFS generates the NRDY interrupt request when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by software.

32.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USBFS.BEMPENB 4009 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0	0: Disable interrupt request 1: Enable interrupt request.	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1	0: Disable interrupt request 1: Enable interrupt request.	R/W

Bit	Symbol	Bit name	Description	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2	0: Disable interrupt request 1: Enable interrupt request.	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3	0: Disable interrupt request 1: Enable interrupt request.	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Disable interrupt request 1: Enable interrupt request.	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Disable interrupt request 1: Enable interrupt request.	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Disable interrupt request 1: Enable interrupt request.	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Disable interrupt request 1: Enable interrupt request.	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8	0: Disable interrupt request 1: Enable interrupt request.	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when a BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register sets to 1 and the associated PIPEnBEMPE (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag sets to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBFS generates a BEMP interrupt request. While at least one PIPEnBEMP bit indicates 1, the USBFS generates the BEMP interrupt request when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by software.

32.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): USBFS.SOFCFG 4009 003Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor*1	Indicates 1 during the edge processing of an edge interrupt output signal.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Clear BRDY flag by software 1: Clear BRDY flag by the USBFS through a data read from the FIFO buffer or data write to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: Not low-speed communication 1: Low-speed communication.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that these bits are 0 before stopping the clock supply to the USBFS.

EDGESTS bit (Edge Interrupt Output Status Monitor)

The EDGESTS bit indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this bit is 0 before stopping the clock supply to the USBFS.

BRDYM bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

TRNENSEL bit (Transaction-Enabled Time Select)

When the USB port is in use for full- or low-speed communications, the TRNENSEL bit specifies the timing with which the USBFS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected. The bit is only valid in host controller mode. Set this bit to 0 in device controller mode.

32.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): USBFS.INTSTS0 4009 0040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0	0*2	0*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error.	R
b3	VALID	USB Request Reception	0: Setup packet not received 1: Setup packet received.	R/W*4
b6 to b4	DVSQ[2:0]	Device State	Indicates the device state. b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspend state.	R
b7	VBSTS	VBUS Input Status	0: USB_VBUS pin is low 1: USB_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R
b9	NRDY	Buffer Not Ready Interrupt Status	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R
b10	BEMP	Buffer Empty Interrupt Status	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status*5	0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred.	R/W*4
b12	DVST	Device State Transition Interrupt Status*5	0: No device state transition interrupt occurred 1: Device state transition interrupt occurred.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status	0: No SOF interrupt occurred 1: SOF interrupt occurred.	R/W*4
b14	RESM	Resume Interrupt Status*5,*6	0: No resume interrupt occurred 1: Resume interrupt occurred.	R/W*4
b15	VBINT	VBUS Interrupt Status*6	0: No VBUS interrupt occurred 1: VBUS interrupt occurred.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.
Note 2. The value is 1 when the USB_VBUS pin is high and 0 when the USB_VBUS pin is low.
Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.
Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bits, write 0 only to the bits to be cleared. Write 1 to the other bits. Do not write 0 to the status bits indicating 0.
Note 5. The status of the RESM, DVST, and CTRT bits are changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.
Note 6. The USBFS detects a change in the status indicated in the VBINT and RESM bits even while the clock supply is stopped (SCKE bit = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software.

CTSQ[2:0] bits (Control Transfer Stage)

In host controller mode, the read value of the CTSQ[2:0] bits is invalid.

VALID bit (USB Request Reception)

In host controller mode, the read value of the VALID bit is invalid.

DVSQ[2:0] bits (Device State)

The DVSQ[2:0] bits are initialized by a USB bus reset. In host controller mode, the read value is invalid.

BRDY flag (Buffer Ready Interrupt Status)

The BRDY flag indicates the BRDY interrupt status.

The USBFS sets the BRDY bit to 1 when it detects a BRDY interrupt status (PIPEnBRDY = 1, n = 0 to 9) on at least one pipe for which BRDY interrupts are enabled (BRDYENB.PIPEnBRDYE = 1).

For the conditions that cause the PIPEnBRDY status to be asserted, see section 32.3.3.1, BRDY interrupt.

The USBFS sets the BRDY bit to 0 when the software writes 0 to all of the PIPEnBRDY bits associated with the PIPEnBRDYE bits that are set to 1. Writing 0 to the BRDY flag in the software does not clear the flag.

NRDY flag (Buffer Not Ready Interrupt Status)

The NRDY flag indicates the NRDY interrupt status.

The USBFS sets the NRDY bit to 1 when it detects a NRDY interrupt status (PIPEnNRDY = 1, n = 0 to 9) on at least one pipe for which NRDY interrupts are enabled (NRDYENB.PIPEnNRDYE = 1).

For the conditions that cause the PIPEnNRDY status to be asserted, see section 32.3.3.2, NRDY interrupt.

The USBFS sets the NRDY bit to 0 when the software writes 0 to all of the PIPEnNRDY bits associated with the PIPEnNRDYE bits that are set to 1. Writing 0 to the NRDY flag in the software does not clear the flag.

BEMP flag (Buffer Empty Interrupt Status)

The BEMP flag indicates the BEMP interrupt status.

The USBFS sets the BEMP bit to 1 when it detects a BEMP interrupt status (PIPEnBEMP = 1, n = 0 to 9) on at least one pipe for which BEMP interrupts are enabled (BEMPENB.PIPEnBEMPE = 1).

For the conditions that cause the PIPEnBEMP status to be asserted, see section 32.3.3.3, BEMP interrupt.

The USBFS sets the BEMP bit to 0 when the software writes 0 to all of the PIPEnBEMP bits associated with the PIPEnBEMPE bits that are set to 1. Writing 0 to the BEMP flag in the software does not clear the flag.

CTRT flag (Control Transfer Stage Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT flag before the USBFS detects the next control transfer stage transition.

Values read from the CTRT flag in host controller mode are invalid.

DVST flag (Device State Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the DVSQ[2:0] bits and sets the DVST flag to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear the DVST flag before the USBFS detects the next device state transition.

Values read from the DVST flag in host controller mode are invalid.

SOFR flag (Frame Number Refresh Interrupt Status)

In host controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit is set to 1 by software. A SOFR interrupt is detected every 1 ms.

In device controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number. A frame number refresh interrupt is detected every 1 ms.

The USBFS can detect an SOFR interrupt through the internal interpolation function even when a corrupted SOF packet is received from the USB host.

RESM flag (Resume Interrupt Status)

In device controller mode, the USBFS sets the RESM flag to 1 on detecting the falling edge of the signal on the USB_DP pin in the Suspend state (DVSQ[2:0] = 1xxb). Values read from the RESM flag in host controller mode are invalid.

VBINT flag (VBUS Interrupt Status)

The USBFS sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB_VBUS pin input value. The USBFS sets the VBSTS flag to indicate the USB_VBUS pin input value. When a VBUS interrupt occurs, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

32.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USBFS.INTSTS1 4009 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status	0: No SACK interrupt occurred 1: SACK interrupt occurred.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status	0: No SIGN interrupt occurred 1: SIGN interrupt occurred.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status	0: No EOFERR interrupt occurred 1: EOFERR interrupt occurred.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status	0: No ATTCH interrupt occurred 1: ATTCH interrupt occurred.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status	0: No DTCH interrupt occurred 1: DTCH interrupt occurred.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status*2	0: No BCHG interrupt occurred 1: BCHG interrupt occurred.	R/W *1
b15	OVRCR	Overcurrent Input Change Interrupt Status*2	0: No OVRCR interrupt occurred 1: OVRCR interrupt occurred.	R/W *1

Note 1. To clear the bits in INTSTS1, write 0 only to the bits to be cleared. Write 1 to the other bits.

Note 2. The USBFS detects a change in the status in the OVRCR or BCHG bit even when the clock supply is stopped (SYSCFG.SCKE = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply (SYSCFG.SCKE = 1) before clearing the status through the software. No other interrupts can be detected while the clock supply is stopped (SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt in host controller mode. Only enable the status change interrupts indicated in the bits in INTSTS1 in host controller mode.

SACK flag (Setup Transaction Normal Response Interrupt Status)

The SACK flag indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBFS detects the SACK interrupt and sets this flag to 1 when an ACK response is returned from a peripheral device during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

Values read from the SACK flag in device controller mode are invalid.

SIGN flag (Setup Transaction Error Interrupt Status)

The SIGN flag indicates the status of setup transaction error interrupts in host controller mode.

The USBFS detects the SIGN interrupt and sets this flag to 1 when an ACK response is not returned from a peripheral device three consecutive times during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBFS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received.

Values read from the SIGN flag in device controller mode are invalid.

EOFERR flag (EOF Error Detection Interrupt Status)

The EOFERR flag indicates the status of EOF error detection interrupts in host controller mode.

The USBFS detects the EOFERR interrupt and sets this flag to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

After detecting the EOFERR interrupt, the USBFS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state.

The software must terminate all pipes in which communications are being carried out and re-enumerate the USB port.

Values read from the EOFERR flag in device controller mode are invalid.

ATTCH flag (ATTCH Interrupt Status)

The ATTCH flag indicates the status of USB attach detection interrupts in host controller mode.

The USBFS detects the ATTCH interrupt and sets this flag to 1 on detecting a J- or K-state on the full- or low-speed signal level for 2.5 µs. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

Values read from the ATTCH flag in device controller mode are invalid.

DTCH flag (USB Disconnection Detection Interrupt Status)

The DTCH flag indicates the status of USB disconnection detection interrupts in host controller mode.

The USBFS detects the DTCH interrupt and sets this flag to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBFS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state.

The software must terminate all pipes in which communications are being carried out and invoke the wait state for attaching to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

BCHG flag (USB Bus Change Interrupt Status)

The BCHG flag indicates the status of USB bus change interrupts in host controller mode.

The USBFS detects the BCHG interrupt and sets this flag to 1 when a change in the full- or low-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS sets the LNST[1:0] flags to indicate the input state of the USB port. When a BCHG interrupt occurs, eliminate transient elements by repeat reading the LNST[1:0] flags by software until the same value is read at least three times.

Change in the USB bus state can be detected while the internal clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

OVRCR flag (Overcurrent Input Change Interrupt Status)

The OVRCR flag indicates the status of USB_OVRCURA and USB_OVRCURB input pin change interrupts.

The USBFS detects the OVRCR interrupt and sets this flag to 1 when a change (high to low or low to high) occurs in at least one of the input values to the USB_OVRCURA and USB_OVRCURB pins. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

32.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USBFS.BRDYSTS 4009 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B RDY	PIPE8B RDY	PIPE7B RDY	PIPE6B RDY	PIPE5B RDY	PIPE4B RDY	PIPE3B RDY	PIPE2B RDY	PIPE1B RDY	PIPE0B RDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BRDY	BRDY Interrupt Status for Pipe 0*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status for Pipe 1*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status for Pipe 2*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status for Pipe 3*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status for Pipe 4*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status for Pipe 5*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1

Bit	Symbol	Bit name	Description	R/W
b6	PIPE6BRDY	BRDY Interrupt Status for Pipe 6*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status for Pipe 7*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status for Pipe 8*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status for Pipe 9*2	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated in the bits in BRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.
Note 2. When the SOFCFG.BRDYM bit is set to 0, clear BRDY interrupts before accessing the FIFO.

32.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USBFS.NRDYSTS 4009 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status for Pipe 0	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status for Pipe 1	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status for Pipe 2	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status for Pipe 3	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status for Pipe 4	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status for Pipe 5	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status for Pipe 6	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status for Pipe 7	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status for Pipe 8	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status for Pipe 9	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated in the bits in NRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

32.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USBFS.BEMPSTS 4009 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP	PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status for Pipe 0	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b1	PIPE1BEMP	BEMP Interrupt Status for Pipe 1	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b2	PIPE2BEMP	BEMP Interrupt Status for Pipe 2	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b3	PIPE3BEMP	BEMP Interrupt Status for Pipe 3	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b4	PIPE4BEMP	BEMP Interrupt Status for Pipe 4	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b5	PIPE5BEMP	BEMP Interrupt Status for Pipe 5	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b6	PIPE6BEMP	BEMP Interrupt Status for Pipe 6	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b7	PIPE7BEMP	BEMP Interrupt Status for Pipe 7	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b8	PIPE8BEMP	BEMP Interrupt Status for Pipe 8	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b9	PIPE9BEMP	BEMP Interrupt Status for Pipe 9	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/W[*1]
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated in the bits in BEMPSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

32.2.18 Frame Number Register (FRMNUM)

Address(es): USBFS.FRMNUM 4009 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number	Latest frame number.	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error	0: No error occurred 1: Error occurred.	R/W[*1]
b15	OVRN	Overrun/Underrun Detection Status	0: No error occurred 1: Error occurred.	R/W[*1]

Note 1. To clear the status, write 0 only to the bits to be cleared. Write 1 to the other bits.

FRNM[10:0] flags (Frame Number)

The USBFS sets the FRNM[10:0] flags to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

CRCE flag (Receive Data Error)

The CRCE flag sets to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error in host controller mode, the USBFS generates an internal NRDY interrupt.

To clear the CRCE flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

OVRN flag (Overrun/Underrun Detection Status)

The OVRN flag sets to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty.

In device controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty.

32.2.19 Device State Change Register (DVCHGR)

Address(es): USBFS.DVCHGR 4009 004Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DVCHG	Device State Change	0: Disable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits 1: Enable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits.	R/W

For details, see section 32.3.1.5, Release from deep software standby mode because of USB suspend/resume interrupts.

32.2.20 USB Address Register (USBADDR)

Address(es): USBFS.USBADDR 4009 0050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	STSRECOV[3:0]				—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address	In device controller mode, these bits indicate the USB address assigned by the host when the USBFS processed the SET_ADDRESS request successfully.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11 to b8	STSRECOV[3:0]	Status Recovery	• Recovery in device controller mode b11 b8 1 0 0 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 001b (default state) 1 0 1 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 010b (address state) 1 0 1 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 011b (configured state). Other settings are prohibited. • Recovery in host controller mode b11 b8 0 1 0 0: Return to the low-speed state (bits DVSTCTR0.RHST[2:0] = 001b) 1 0 0 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b). Other settings are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

USBADDR[6:0] bits (USB Address)

In device controller mode, the USBADDR[6:0] flags indicate the USB address received when the USBFS processed a SetAddress request successfully. The USBFS sets the USBADDR[6:0] bits to 00h on detecting a USB bus reset.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1. On recovering from a USB power shut-off, the operation can resume from the USB address set before the software shut-off.

In host controller mode, the USBADDR[6:0] bits are invalid.

STSRECOV[3:0] bits (Status Recovery)

Use the STSRECOV[3:0] bits to resume the state of the internal sequencer on recovering from USB power shut-off. For details, see section 32.3.1.5, Release from deep software standby mode because of USB suspend/resume interrupts.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1.

32.2.21 USB Request Type Register (USBREQ)

Address(es): USBFS.USBREQ 4009 0054h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	USB request bmRequestType value	R/W *1
b15 to b8	BREQUEST[7:0]	Request	USB request bRequest value	R/W *1

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBREQ stores setup requests for control transfers.

In device controller mode, the USBREQ stores the received bRequest and bmRequestType values. In host controller mode, it sets to the bRequest and bmRequestType values to be transmitted.

USBREQ is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] bits (Request Type)

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

BREQUEST[7:0] bits (Request)

The BREQUEST[7:0] bits store bRequest value of the USB request.

- In host controller mode:
  Set these bits to the value of the USB request data in setup transmission transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

32.2.22 USB Request Value Register (USBVAL)

Address(es): USBFS.USBVAL 4009 0056h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	USB request wValue value	R/W *1

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

In device controller mode, USBVAL stores the received wValue value. In host controller mode, it sets to the wValue value to be transmitted is set.

USBVAL is initialized by a USB bus reset.

WVALUE[15:0] bits (Value)

The WVALUE[15:0] bits store wValue value of the USB request.

- In host controller mode:
  Set these bits to the value of the wValue field in USB requests of transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

32.2.23 USB Request Index Register (USBINDX)

Address(es): USBFS.USBINDX 4009 0058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	USB request wIndex value	R/W *1

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBINDX stores setup requests for control transfers.

In device controller mode, it stores the received wIndex value. In host controller mode, it sets to the wIndex value to be transmitted.

USBINDX is initialized by a USB bus reset.

WINDEX[15:0] bits (Index)

The WINDEX[15:0] bits hold the wIndex value of a USB request.

- In host controller mode:
  Set these bits to the wIndex value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wIndex value in USB requests received in reception setup transactions. Writing to the bits has no effect.

32.2.24 USB Request Length Register (USBLENG)

Address(es): USBFS.USBLENG 4009 005Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WLENTUH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WLENTUH[15:0]	Length	USB request wLength value	R/W*[1]

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBLENG stores setup requests for control transfers.

In device controller mode, the value of wLength that is received is stored. In host controller mode, the value of wLength to be transmitted is set.

USBLENG is initialized by a USB bus reset.

WLENTUH[15:0] bits (Length)

The WLENTUH[15:0] bits hold the wLength value of a USB request.

- In host controller mode:
  Set these bits to the wLength value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wLength value in USB requests received in reception setup transactions. Writing to the bits has no effect.

32.2.25 DCP Configuration Register (DCPCFG)

Address(es): USBFS.DCPCFG 4009 005Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*[1]	0: Data receiving direction 1: Data transmitting direction.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*[1]	0: Keep pipe open after transfer ends 1: Disable pipe after transfer ends.	R/W

Bit	Symbol	Bit name	Description	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only set this bit while the PID is NAK. Before setting this bit, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

DIR bit (Transfer Direction)

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBFS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that a transfer has ended. The USBFS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received.

32.2.26 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USBFS.DCPMAXP 4009 005Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	Maximum data payload specification (maximum packet size) for the DCP	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b 0 0 1 0: Address 0010b 0 0 1 1: Address 0011b 0 1 0 0: Address 0100b 0 1 0 1: Address 0101b. Other settings are prohibited.	R/W

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit in the port control register to 1.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 40h (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF while MXPS[6:0] is set to 0.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target peripheral device for a control transfer. Set up the device address in the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

32.2.27 DCP Control Register (DCPCTR)

Address(es): USBFS.DCPCTR 4009 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b2	CCPL	Control Transfer End Enable	0: Disable control transfer completion 1: Enable control transfer completion.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: DCP not used for the USB bus 1: DCP in use for the USB bus.	R
b6	SQMON	Sequence Toggle Bit Monitor	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*2	Sets the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	Clears the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	Clears the SUREQ bit in host controller mode. 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0. This bit is read as 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	Sets up token transmission in host controller mode. 0: Invalid (writing 0 has no effect) 1: Transmit setup packet.	R/W
b15	BSTS	Buffer Status	0: Buffer access disabled 1: Buffer access enabled.	R

Note 1. This bit is read as 0.

Note 2. Only set the SQSET and SQCLR bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:

a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.

b. Set PID[1:0] bits to 01b (BUF).
The USBFS then executes the OUT transaction.

- When the receiving direction is set:

a. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.

b. Set PID[1:0] bits to 01b (BUF).
The USBFS then executes the IN transaction.

The USBFS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets the PID[1:0] to STALL (11b)
- When a reception error, such as a CRC error, is detected three times consecutively, the USBFS sets the PID[1:0] bits to NAK (00b)
- On receiving the STALL handshake, the USBFS sets PID[1:0] to STALL (11b).

In device controller mode, the USBFS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBFS sets PID[1:0] to NAK (00b). The USBFS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets PID[1:0] to STALL (11b)
- On detecting a control transfer sequence error, the USBFS sets PID[1:0] to STALL (1xb)
- On detecting a USB bus reset, the USBFS sets PID[1:0] to NAK.

The USBFS does not check the PID[1:0] setting while processing a SET_ADDRESS request.

The PID[1:0] bits are initialized by a USB bus reset.

CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by software while the associated PID[1:0] bits are set to BUF, the USBFS completes the control transfer status stage.

During control read transfers, the USBFS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET_ADDRESS request, the USBFS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBFS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used for the transaction when USBFS changes the PID[1:0] bits from BUF to NAK. The USBFS changes the PBUSY bit from 0 to 1 on start of a USB transaction for the selected pipe. It changes the PBUSY bit from 1 to 0 on completion of one transaction.

After PID is set to NAK by software, the value in the PBUSY bit indicates whether changes to pipe settings can proceed.

For details, see section 32.3.4.1, Pipe control register switching procedures.

SQMON bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBFS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATA-PID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBFS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBFS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR bit (SUREQ Bit Clear)

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 by software. This is not necessary at the end of a normal setup transaction, because the USBFS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to this bit.

SUREQ bit (Setup Token Transmission)

In host controller mode, setting the SUREQ bit to 1 triggers the USBFS to transmit the setup packet. After completing the setup transaction process, the USBFS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBFS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the target USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

BSTS flag (Buffer Status)

The BSTS flag indicates the status of access to the DCP FIFO buffer. The meaning of this flag varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer.

32.2.28 Pipe Window Select Register (PIPESEL)

Address(es): USBFS.PIPESEL 4009 0064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set pipes 1 to 9 using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN registers (n = 0 to 9).

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI registers. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits has no effect.

32.2.29 Pipe Configuration Register (PIPECFG)

Address(es): USBFS.PIPECFG 4009 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	Specifies the endpoint number for the selected pipe. Setting 0000b indicates that the pipe is not used.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Continue pipe operation after transfer ends 1: Disable pipe after transfer ends.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer.	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: Generate BRDY interrupt on transmitting or receiving data 1: Generate BRDY interrupt on completion of reading data.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer. • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited. • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited.	R/W

Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. To change the BFRE, DBLB, or DIR bits after completing USB communication on the selected pipe, in addition to the constraints described in Note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously through the software and clear the FIFO buffer assigned to the pipe.

PIPECFG specifies the transfer type, FIFO buffer access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. The EPNUM[3:0] bits can be set to 0000b for all pipes.

DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBFS uses the selected pipe for receiving. When the software sets this bit to 1, the USBFS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBFS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBFS determines that the transfer has ended on the following conditions:

- A short packet data (including a zero-length packet) was successfully received

- The transaction counter is used and the number of packets specified for the transaction counter are successfully received.

DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBFS to the CPU for the selected pipe.

When the software sets the BFRE bit to 1 and the selected pipe is in the receiving direction, the USBFS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by software and the selected pipe is in the transmitting direction, the USBFS does not generate the BRDY interrupt. For details, see section 32.3.3.1, BRDY interrupt.

TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

32.2.30 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USBFS.PIPEMAXP 4009 006Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	• Pipes 1 and 2 1 byte (001h) to 256 bytes (100h) • Pipes 3 to 5 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) (Bits [8:7] and [2:0] not supported.) • Pipes 6 to 9 1 byte (001h) to 64 bytes (040h) (Bits [8:7] not supported.)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b 0 0 1 0: Address 0010b 0 0 1 1: Address 0011b 0 1 0 0: Address 0100b 0 1 0 1: Address 0101b. Other settings are prohibited.	R/W

Note 1. The value of the MXPS[8:0] bits is 000h when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 040h when a pipe is selected.

Note 2. Only set the MXPS[8:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. Only set the DEVSEL[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEMAXP specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the device address in the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

32.2.31 Pipe Cycle Control Register (PIPEPERI)

Address(es): USBFS.PIPEPERI 4009 006Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	IITV[2:0] *1	Interval Error Detection Interval	Specifies the interval error detection timing for the selected pipe as the n-th power of 2 of the frame timing	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: Do not flush buffer 1: Flush buffer.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe selected in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBFS automatically clears the FIFO buffer if the USBFS fails to receive the IN token from the USB host within the interval set in the IITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBFS only clears the data in the previously used plane.

The USBFS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBFS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal interpolation function.

When the host controller function is selected, set this bit to 0. When the selected pipe is not for isochronous transfer, set this bit to 0.

32.2.32 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

PIPEnCTR (n = 1 to 5)

Address(es): USBFS.PIPE1CTR 4009 0070h, USBFS.PIPE2CTR 4009 0072h, USBFS.PIPE3CTR 4009 0074h, USBFS.PIPE4CTR 4009 0076h, USBFS.PIPE5CTR 4009 0078h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*2	Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disable 1: Enable (initialize all buffers).	R/W
b10	ATREPM	Auto Response Mode*2	0: Disable auto response mode 1: Enable auto response mode.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor	0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer.	R
b15	BSTS	Buffer Status	0: Buffer access by the CPU disabled 1: Buffer access by the CPU enabled.	R

Note 1. Only 0 can be read.

Note 2. Only set the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEnCTR can be set for any pipe selection in the PIPESEL register.

PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction on the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 32.7 and Table 32.8 show the basic operations of the USBFS (when there are no errors in the communication packets) based on the PID[1:0] bit setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by software
- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode.

To specify the response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, transition to NAK and then BUF.

Table 32.7 Operation of the USBFS based on the PID[1:0] setting in host controller mode

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens
01b (BUF)	Bulk or interrupt	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting	Issues tokens regardless of the status of the FIFO buffer associated with the selected pipe.
10b (STALL) or 11b (STALL)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens.

Table 32.8 Operation of the USBFS based on the PID[1:0] setting in device controller mode (1 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Bulk or interrupt	Does not depend on the setting	Returns NAK in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

Table 32.8 Operation of the USBFS based on the PID[1:0] setting in device controller mode (2 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.
	Isochronous	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Does not depend on the setting	Returns STALL in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see section 32.3.4.1, Pipe control register switching procedures.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBFS toggles the SQMON flag on successful completion of the transaction. However, the USBFS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS clears the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS clears the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 32.9 shows the data cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which this processing is required.

Table 32.9 Data cleared by the USBFS when ACLRM = 1 (1 of 2)

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When initializing the selected pipe

Table 32.9 Data cleared by the USBFS when ACLRM = 1 (2 of 2)

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	FIFO buffer toggle control	When changing the PIPECFG.DBLB setting
5	Internal flags related to the transaction count	When forcing the transaction count function to terminate

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBFS responds to the token from the USB host as follows:

- When the selected pipe is set for bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
    a. When the ATREPM bit = 1 and PID = BUF, the USBFS transmits a zero-length packet in response to the IN token.
    b. The USBFS updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USBFS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBFS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is set for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):

    When the ATREPM bit = 1 and PID = BUF, the USBFS returns NAK in response to the OUT token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

INBUFM bit (Transmit Buffer Monitor)

The INBUMFM bit indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBFS sets this bit to 1 when the CPU or DMA/DTC completes writing data to at least one FIFO buffer plane.

The USBFS sets this bit to 0 when the USBFS completes transmission of the data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBFS sets the INBUFM bit to 0 when the USBFS completes transmission of the data from the two FIFO buffer planes before the CPU or DMA/DTC completes writing data to one FIFO buffer plane.

The INBUFM bit indicates the same value as the BSTS bit when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 32.10.

Table 32.10 BSTS bit operation

DIR value	BFRE value	DCLRM value	BSTS bit function
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete
		1	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and clears to 0 on completion of data write
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

PIPEnCTR (n = 6 to 9)

Address(es): USBFS.PIPE6CTR 4009 007Ah, USBFS.PIPE7CTR 4009 007Ch, USBFS.PIPE8CTR 4009 007Eh, USBFS.PIPE9CTR 4009 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*2	Sets the sequence toggle bit for pipe n: 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear*2	Clears the sequence toggle bit for pipe n: 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disable 1: Enable (all buffers initialized).	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status	0: Buffer access disabled 1: Buffer access enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.

Note 2. Only write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 32.7 and Table 32.8 show the basic operation (when there are no errors in the transmitted and received packets) of the USBFS depending on the PID[1:0] setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode.

To specify each response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, transition to NAK and then BUF.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

The USBFS toggles the SQMON bit on successful completion of the transaction. However, the USBFS does not toggle the SQMON bit when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS sets the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS sets the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 32.11 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 32.11 Data cleared by the USBFS when ACLRM = 1

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe	When initializing the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	Internal flags related to the transaction count	When forcing the transaction count function to terminate

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 32.10.

32.2.33 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USBFS.PIPE1TRE 4009 0090h, USBFS.PIPE2TRE 4009 0094h, USBFS.PIPE3TRE 4009 0098h, USBFS.PIPE4TRE 4009 009Ch, USBFS.PIPE5TRE 4009 00A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid (writing 0 has no effect) 1: Clear counter value.	R/W
b9	TRENB	Transaction Counter Enable	0: Disable transaction counter 1: Enable transaction counter.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set each bit in PIPEnTRE while PID is NAK. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

TRCLR bit (Transaction Counter Clear)

When the TRCLR bit sets to 1, the USBFS clears the value of the transaction counter associated with the selected pipe and then sets the TRCLR bit to 0.

TRENB bit (Transaction Counter Enable)

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBFS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting, as follows:

- When the PIPECFG.SHTNAK bit is 1, the USBFS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting
- When the PIPECFG.BFRE bit is 1, the USBFS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data.

For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

32.2.34 PIPEn Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USBFS.PIPE1TRN 4009 0092h, USBFS.PIPE2TRN 4009 0096h, USBFS.PIPE3TRN 4009 009Ah,
USBFS.PIPE4TRN 4009 009Eh, USBFS.PIPE5TRN 4009 00A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter	When written to, this bit specifies the total packets (number of transactions) to be received by the selected pipe. When read from, when PIPEnTRE.TRENB is 0, this bit indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, this bit indicates the current transaction count.	R/W

The PIPEnTRN registers retain their settings during a USB bus reset.

TRNCNT[15:0] bits (Transaction Counter)

The USBFS increments the value of the TRNCNT[15:0] bits by 1 when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting.

The USBFS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied:

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting.

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBFS received a short packet.

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit was set to 1 by software.

For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

32.2.35 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): USBFS.DEVADD0 4009 00D0h, USBFS.DEVADD1 4009 00D2h, USBFS.DEVADD2 4009 00D4h, USBFS.DEVADD3 4009 00D6h, USBFS.DEVADD4 4009 00D8h, USBFS.DEVADD5 4009 00DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: Do not use DEVADDn 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9.

In host controller mode, set all DEVADDn bits before starting communication to any pipes. Only change the bits in DEVADDn when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- DEVADDn is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

In device controller mode, set all bits in this register to 0.

USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. Set these bits to 10b when a full-speed device is connected through the hub. In host controller mode, the USBFS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

32.2.36 PHY Cross Point Adjustment Register (PHYSLEW)

Address(es): USBFS.PHYSLEW 4009 00F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SLEWF01	SLEWF00	SLEWR01	SLEWR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	SLEWR00	Driver Cross Point Adjustment 00	0: Reserved 1: Host or device controller mode.	R/W
b1	SLEWR01	Driver Cross Point Adjustment 01	0: Host or device controller mode 1: Reserved.	R/W

Bit	Symbol	Bit name	Description	R/W
b2	SLEWF00	Driver Cross Point Adjustment 00	0: Reserved 1: Host or device controller mode.	R/W
b3	SLEWF01	Driver Cross Point Adjustment 01	0: Host or device controller mode 1: Reserved.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b21, b20	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PHYSLEW register adjusts the cross point of the driver. In both host and device controller modes, set this register before operating the controller.

32.2.37 Deep Software Standby USB Transceiver Control/Pin Monitor Register (DPUSR0R)

Address(es): USBFS.DPUSR0R 4009 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIXPHY0	DRPD0	—	RPUE0	SRPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	SRPC0	USB Single-ended Receiver Control	0: Disable input through DP and DM inputs 1: Enable input through DP and DM inputs.	R/W
b1	RPUE0*1	DP Pull-Up Resistor Control	0: Disable DP pull-up resistor 1: Enable DP pull-up resistor.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DRPD0*1	D+/D- Pull-Down Resistor Control	0: Disable DP/DM pull-down resistor 1: Enable DP/DM pull-down resistor.	R/W
b4	FIXPHY0	USB Transceiver Output Fix	0: Fix outputs in Normal mode and on return from Deep Software Standby mode 1: Fix outputs on transition to Deep Software Standby mode.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DP0	USB D+ Input	Indicates D+ input signal on the USBFS side	R
b17	DM0	USB D- Input	Indicates D- input signal on the USBFS side	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVCA0	USB OVRCURA Input	Indicates OVRCURA input signal on the USBFS side	R
b21	DOVCB0	USB OVRCURB Input	Indicates OVRCURB input signal on the USBFS side	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBSTS0	USB VBUS Input	Indicates VBUS input signal on the USBFS side	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use this bit during operation in Deep Software Standby mode. For details, see section 32.3.1.5, Release from deep software

standby mode because of USB suspend/resume interrupts.

SRPC0 bit (USB Single-ended Receiver Control)

The SRPC0 bit controls the D+ and D- inputs of the USB transceiver. This bit is only valid when the FIXPHY0 bit is 1.

FIXPHY0 bit (USB Transceiver Output Fix)

The FIXPHY0 bit keeps the outputs of the USB transceiver disabled.

32.2.38 Deep Software Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address(es): USBFS.DPUSR1R 4009 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DPINTE0	USB DP Interrupt Enable/Clear	0: Disable recovery from Deep Software Standby mode by DP input 1: Enable recovery from Deep Software Standby mode by DP input.	R/W
b1	DMINTE0	USB DM Interrupt Enable/Clear	0: Disable recovery from Deep Software Standby mode by DM input 1: Enable recovery from Deep Software Standby mode by DM input.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DOVRCRAE0	USB OVRCURA Interrupt Enable/Clear	0: Disable recovery from Deep Software Standby mode by OVRCURA input 1: Enable recovery from Deep Software Standby mode by OVRCURA input.	R/W
b5	DOVRCRBE0	USB OVRCURB Interrupt Enable/Clear	0: Disable recovery from Deep Software Standby mode by OVRCURB input 1: Enable recovery from Deep Software Standby mode by OVRCURB input.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	DVBSE0	USB VBUS Interrupt Enable/Clear	0: Disable recovery from Deep Software Standby mode by VBUS input 1: Enable recovery from Deep Software Standby mode by VBUS input.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DPINT0	USB DP Interrupt Source Recovery	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode because of DP.	R
b17	DMINT0	USB DM Interrupt Source Recovery	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode because of DM input.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVRCRA0	USB OVRCURA Interrupt Source Recovery	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode because of OVRCURA input.	R

Bit	Symbol	Bit name	Description	R/W
b21	DOVRCRB0	USB OVRCURB Interrupt Source Recovery	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode because of OVRCURB input.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBINT0	USB VBUS Interrupt Source Recovery	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode because of VBUS input.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DPINTE0 bit (USB DP Interrupt Enable/Clear)

The DPINET0 bit enables or disables triggering of recovery from Deep Software Standby mode by the DP input of the USBFS. Writing 0 to this bit while the DPINT0 bit is 1 sets the DPINT0 bit to 0.

DMINTE0 bit (USB DM Interrupt Enable/Clear)

The DMINTE0 bit enables or disables triggering of recovery from Deep Software Standby mode by the DM input of the USBFS. Writing 0 to this bit while the DMINT0 bit is 1 clears the DMINTE0 bit to 0.

DOVRCRAE0 bit (USB OVRCURA Interrupt Enable/Clear)

The DOVRCRAE0 bit enables or disables triggering of recovery from Deep Software Standby mode by the OVRCURA input of the USBFS. Writing 0 to this bit while the DOVRCRA0 bit is 1 clears the DOVRCRAE0 bit to 0.

DOVRCRBE0 bit (USB OVRCURB Interrupt Enable/Clear)

The DOVRCRBE0 bit enables or disables triggering of recovery from Deep Software Standby mode by the OVRCURB input of the USBFS. Writing 0 to this bit while the DOVRCRB0 bit is 1 clears the DOVRCRB0 bit to 0.

DVBSE0 bit (USB VBUS Interrupt Enable/Clear)

The DVBSE0 bit enables or disables triggering of recovery from Deep Software Standby mode by the VBUS input of the USBFS. Writing 0 to this bit while the DVBINT0 bit is 1 clears the DVBINT0 bit to 0.

DPINT0 bit (USB DP Interrupt Source Recovery)

The DPINT0 bit indicates that the system has returned from Deep Software Standby mode because of the DP input of the USBFS. This recovery is only enabled when the DPINTE0 bit is 1. Writing 0 to the DPINTE0 bit while this bit is 1 clears this bit to 0.

DMINT0 bit (USB DM Interrupt Source Recovery)

The DMINT0 bit indicates that the system has returned from Deep Software Standby mode because of the DM input of the USBFS. This recovery is only enabled when the DMINTE0 bit is 1. Writing 0 to the DPINTE0 bit while this bit is 1 clears this bit to 0.

DOVRCRA0 bit (USB OVRCURA Interrupt Source Recovery)

The DOVRCRA0 bit indicates that the system has returned from Deep Software Standby mode because of the OVRCURA input of the USBFS. This recovery is only enabled when the DOVRCRAE0 bit is 1. Writing 0 to the DOVRCRAE0 bit while this bit is 1 clears this bit to 0.

DOVRCRB0 bit (USB OVRCURB Interrupt Source Recovery)

The DOVRCRB0 bit indicates that the system has returned from Deep Software Standby mode because of the OVRCURB input of the USBFS. This recovery is only enabled when the DOVRCRBE0 bit is 1. Writing 0 to the DOVRCRBE0 bit while this bit is 1 clears this bit to 0.

DVBINT0 bit (USB VBUS Interrupt Source Recovery)

The DVBINT0 bit indicates that the system has returned from Deep Software Standby mode because of the VBUS input of the USBFS. This recovery is only enabled when the DVBSE0 bit is 1. Writing 0 to the DVBSE0 bit while this bit is 1 clears this bit to 0.

32.3 Operation

32.3.1 System Control

This section describes register settings required for initializing the USBFS and controlling power consumption.

32.3.1.1 Setting data to the USBFS registers

Setting the SYSCFG.USBE bit to 1 after starting the clock supply (SYSCFG.SCKE bit = 1) enables and starts USBFS operation.

32.3.1.2 Selecting the controller function

The USBFS can operate as either a host or device controller.

Use the SYSCFG.DCFM bit to select one of these USBFS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D- pull-down-disabled state (SYSCFG.DRPD bit = 0).

32.3.1.3 Controlling the USB data bus using resistors

The USBFS provides pull-up and pull-down resistors for the D+ and D- lines. Pull these lines up or down by setting the SYSCFG.DPRPU and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication).

When the SYSCFG.DPRPU bit is set to 0 during communication with a PC, the USBFS disables the pull-up resistor of the USB data line, thereby notifying the USB host of disconnection.

In host controller mode, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D- lines.

Table 32.12 USB data bus resistor control

SYSCFG register settings		USB data bus control		
DRPD bit	**DPRPU bit**	**D-**	**D+**	**Function**
0	0	Open	Open	When resistors not used
0	1	Open	Pull-up	When operating as a device controller at full-speed
1	0	Pull-down	Pull-down	When operating as a host controller
1	1	—	—	Setting prohibited

32.3.1.4 Example external connection circuits

Figure 32.2 shows an example OTG connection in the self-powered system. The USBFS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBFS can use this to notify the USB host of a device disconnect.

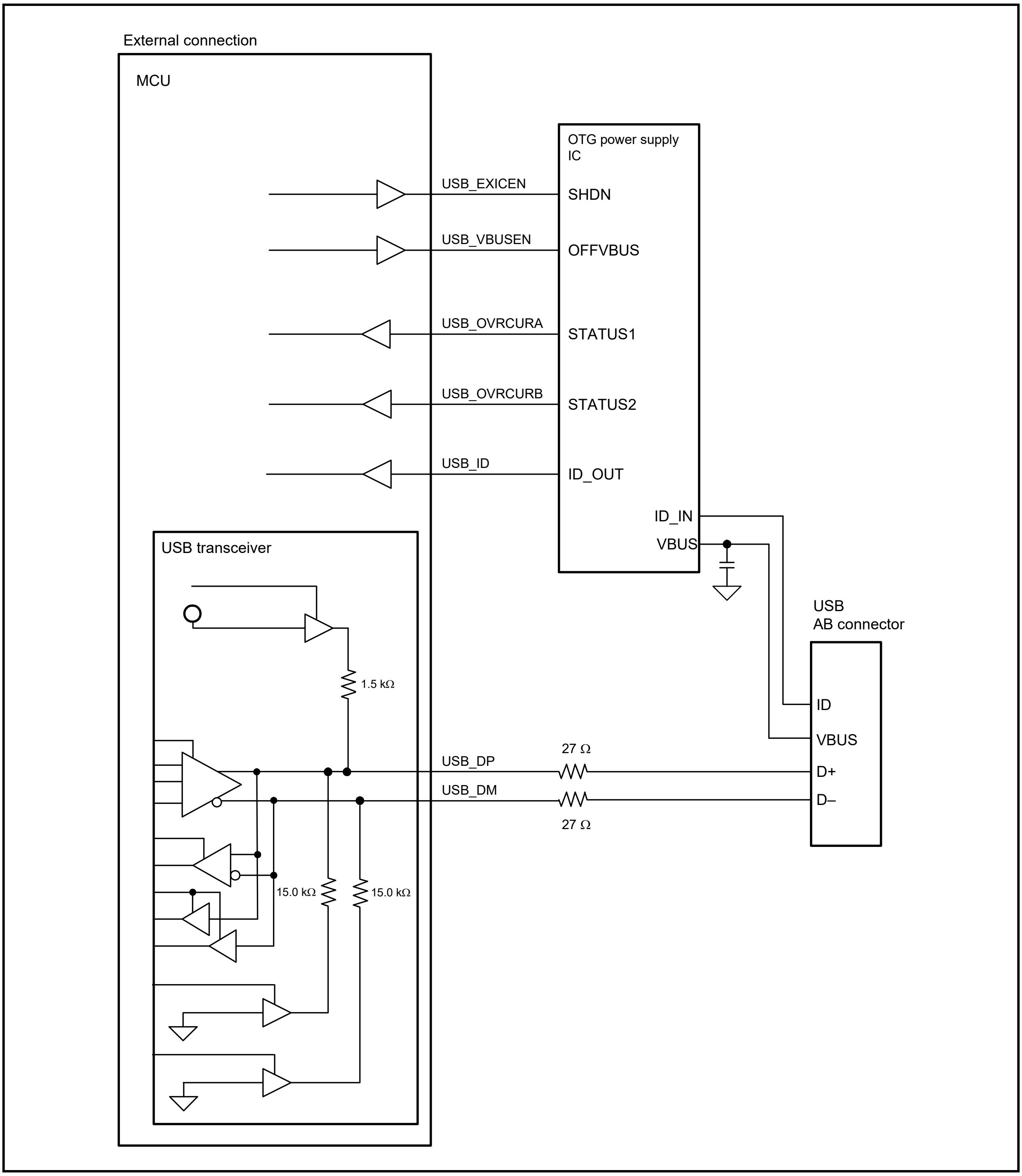
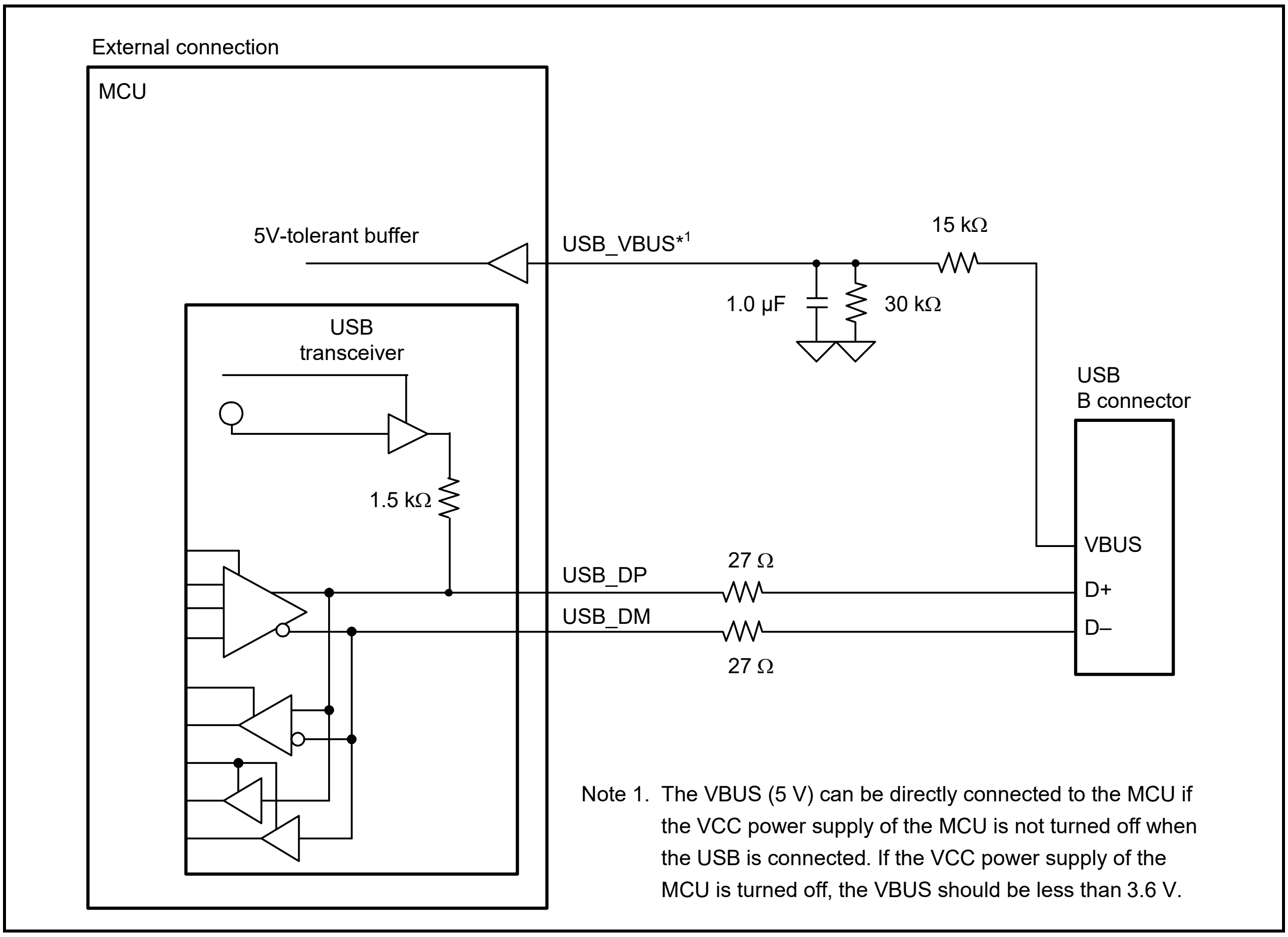


Figure 32.2 Example OTG connection in a self-powered system

Figure 32.3 shows an example device connection in a self-powered system.



Note 1. The VBUS (5 V) can be directly connected to the MCU if the VCC power supply of the MCU is not turned off when the USB is connected. If the VCC power supply of the MCU is turned off, the VBUS should be less than 3.6 V.

Figure 32.3 Example device connection in a self-powered system

Figure 32.4 shows an example host connection.

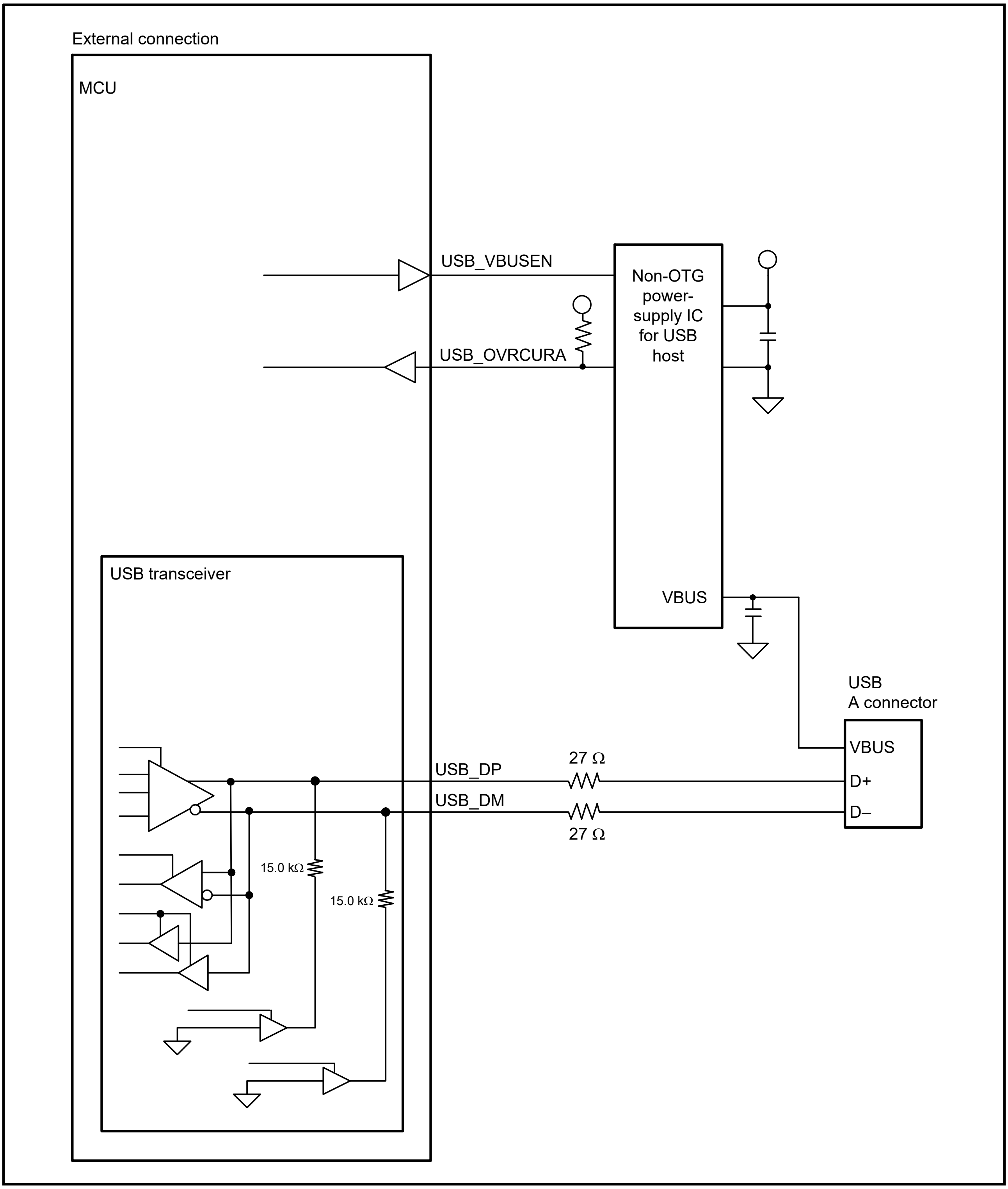


Figure 32.4 Example host connection

Figure 32.5 shows an example device connection in a bus-powered system.

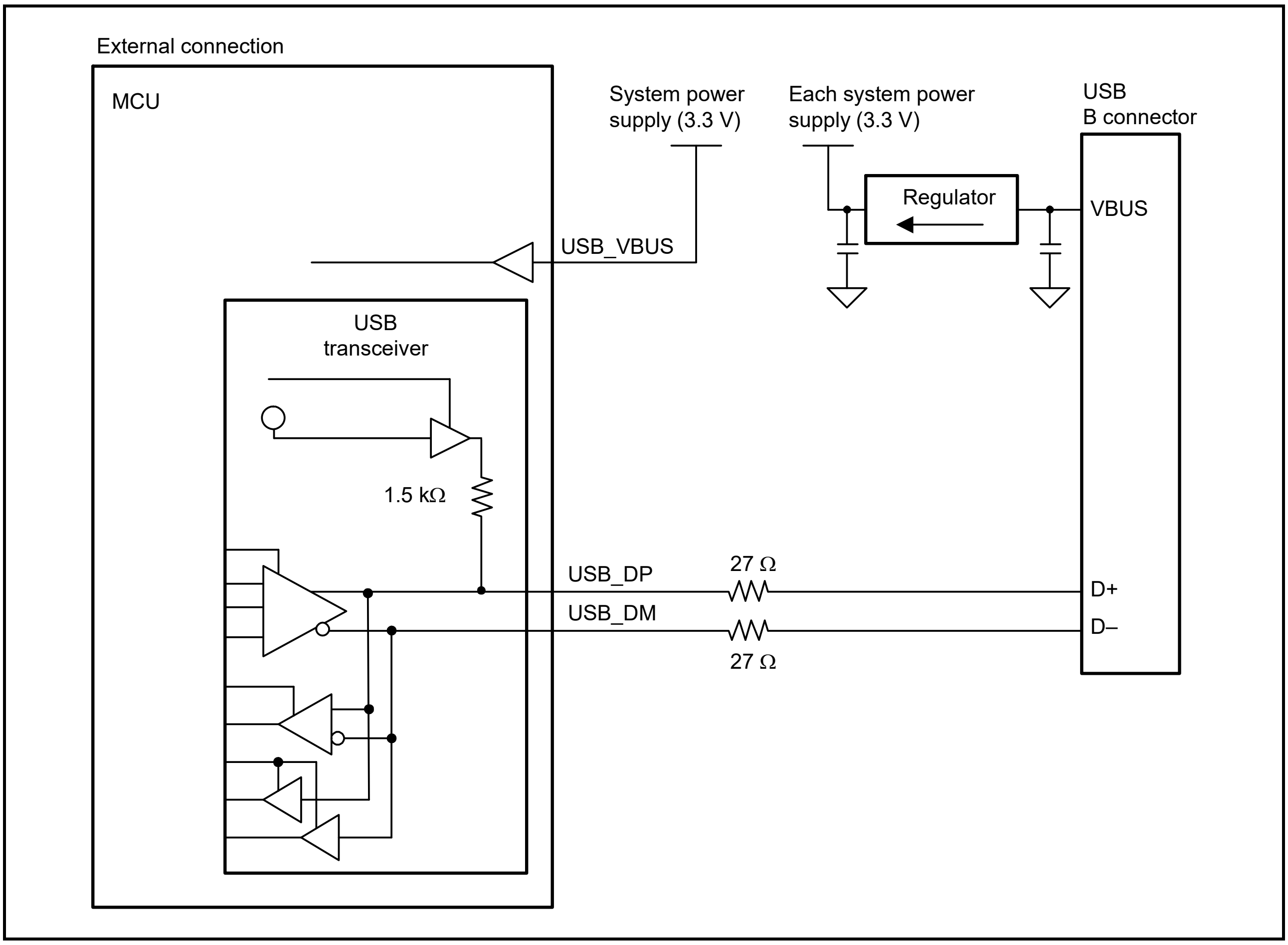


Figure 32.5 Example device connection in a bus-powered state

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

32.3.1.5 Release from deep software standby mode because of USB suspend/resume interrupts

Deep Software Standby mode can be canceled by a USB suspend/resume interrupt. USB suspend/resume interrupts are detected by the USB resume detecting unit, which controls and monitors the USB I/O pins to detect the interrupts.

Figure 32.6 shows a schematic diagram of the connection between the USB resume detecting unit and the USB I/O pins.

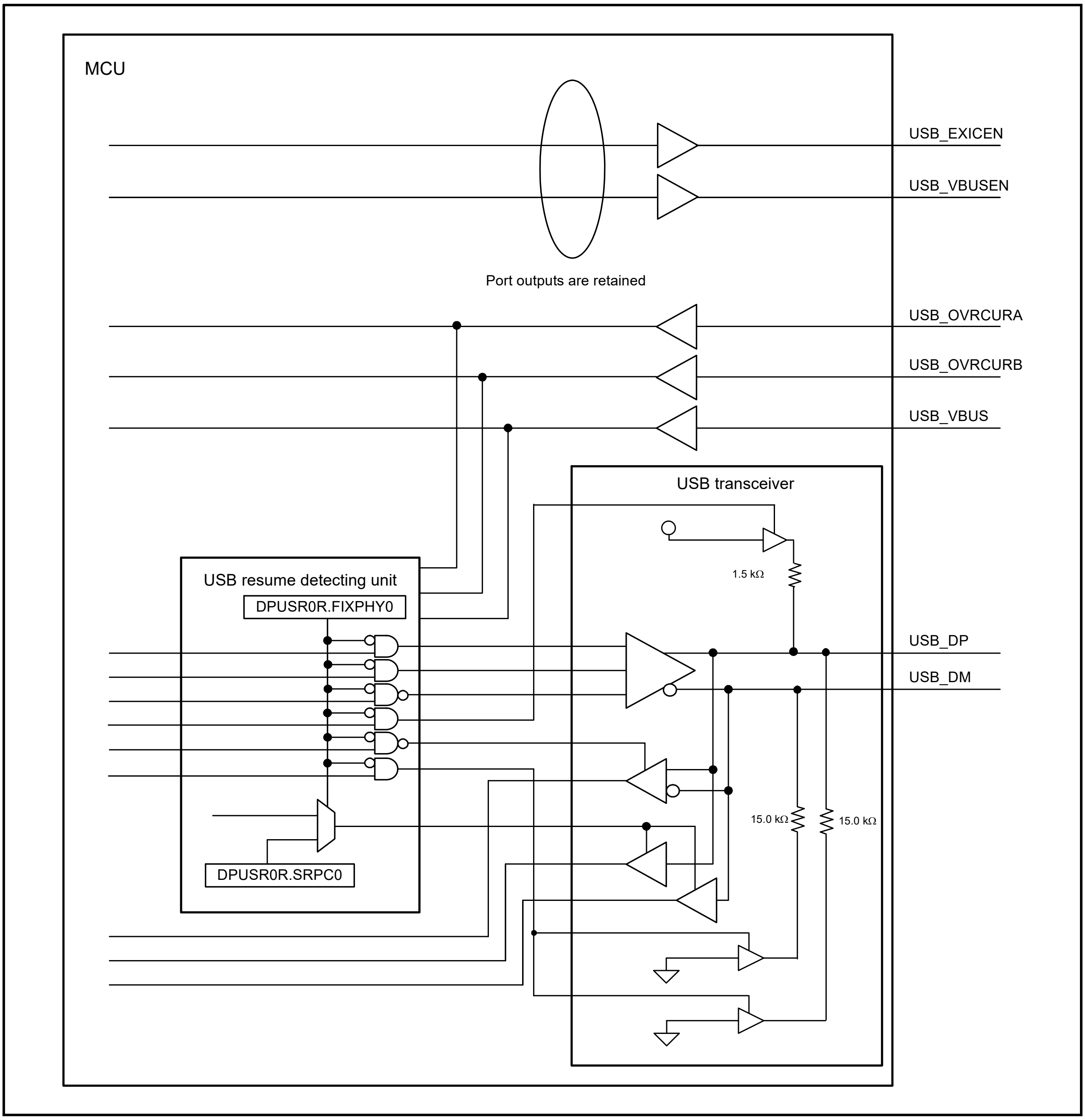


Figure 32.6 Connection between the USB resume detecting unit and the USB I/O pins

Table 32.13 shows the USB suspend and resume interrupt sources and their associated I/O pins.

Table 32.13 USB suspend and resume interrupt sources and their associated I/O pins

USB operating mode	Source	Pin name
Device, OTG	Resume	USB_DP
Host, OTG	Attach or detach	USB_DP, USB_DM
Device	Attach or detach	USB_VBUS
Host	Overcurrent detection	USB_OVRCURA
OTG	Overcurrent detection	USB_OVRCURA, USB_OVRCURB

Figure 32.7 shows the flow for setting the USBFS when entering Deep Software Standby mode from either host or device controller mode. Figure 32.8 shows the flow for setting the USBFS when canceling Deep Software Standby mode

from host controller mode. Figure 32.9 shows the flow for setting the USBFS when canceling Deep Software Standby mode from device controller mode.

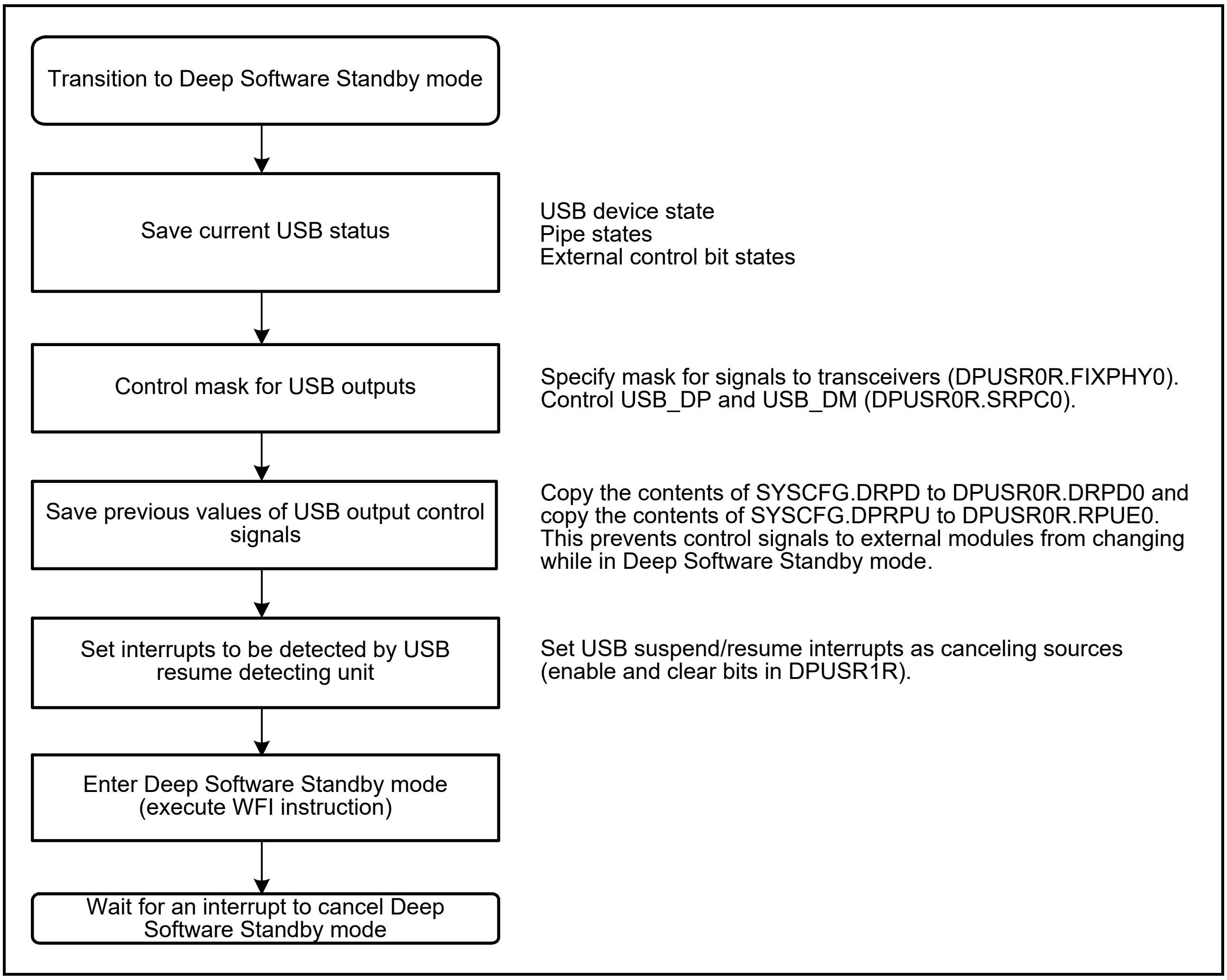


Figure 32.7 USBFS setup flow for transition to Deep Software Standby mode as host or device controller

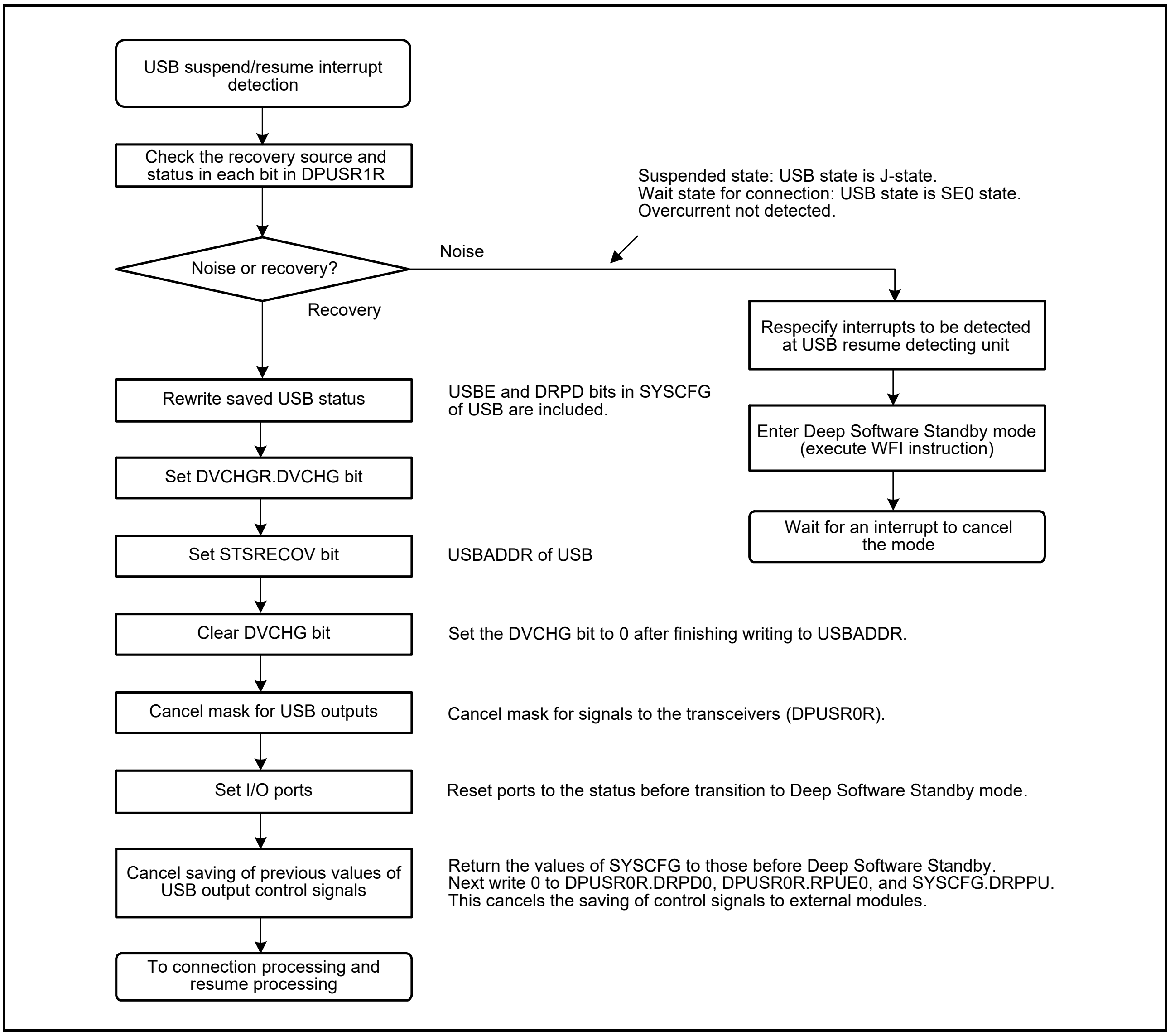


Figure 32.8 USBFS setup flow for canceling Deep Software Standby mode as host controller

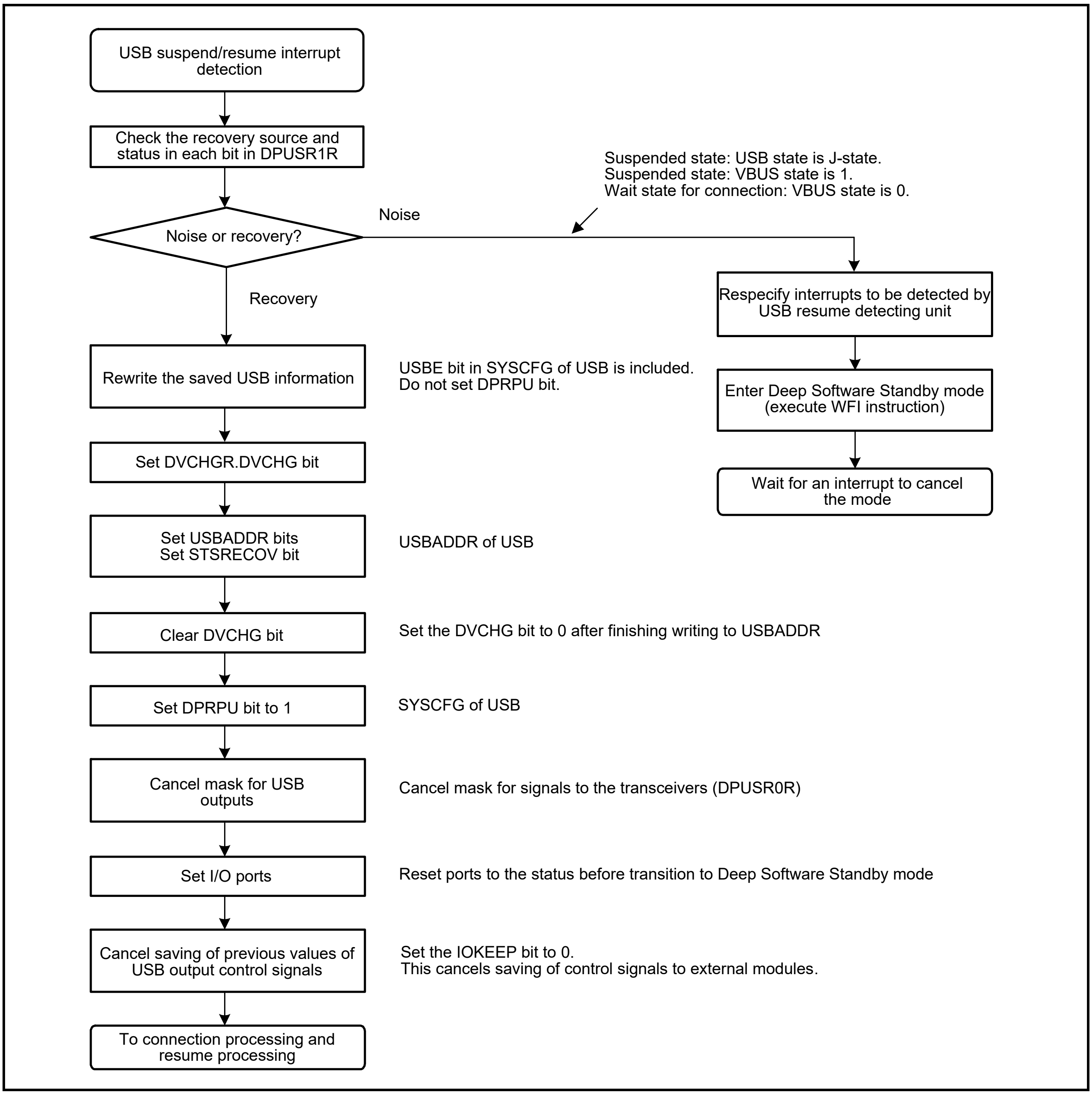


Figure 32.9 USBFS setup flow for canceling Deep Software Standby mode as device controller

32.3.2 Interrupts

Table 32.14 lists the interrupt sources in the USBFS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, a USBFS interrupt request is issued to the Interrupt Controller Unit (ICU) and an USBFS interrupt is generated.

Table 32.14 Interrupt sources (1 of 3)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	• A change in the state of the USB_VBUS input pin was detected (low to high or high to low)	Host or device*1	INTSTS0.VBSTS

Table 32.14 Interrupt sources (2 of 3)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
RESM	Resume interrupt	• A change in the state of the USB bus was detected in the Suspend state (J-state to K-state or J-state to SE0)	Device	—
SOFR	Frame number update interrupt	In host controller mode: • An SOF packet with a different frame number was transmitted In device controller mode: • An SOF packet with a different frame number was received	Host or device	—
DVST	Device state transition interrupt	• One of the following device state transitions was detected: - USB bus reset was detected - Suspend state was detected - SET_ADDRESS request was received - SET_CONFIGURATION request was received	Device	INTSTS0.DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• A control transfer stage transition was detected because of one of the following: - Setup stage completed - Control write transfer status stage transition occurred - Control read transfer status stage transition occurred - Control transfer completed - Control transfer sequence error occurred.	Device	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• The buffer is empty after all FIFO buffer data was transmitted • A packet larger than the maximum packet size was received	Host or device	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	In host controller mode: • A STALL response was received from the peripheral device in response to the issued token • The response from the peripheral device in response to the issued token was not received successfully (no response three times consecutively or packet reception error three times consecutively) • An overrun or underrun error occurred during isochronous transfer In device controller mode: • NAK was returned for an IN or OUT token while the PID[1:0] bits were set to 01b (BUF) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer • An overrun or underrun occurred during data reception in isochronous transfer	Host or device	NRDYSTS.PIPEnNRDY
BRDY	Buffer ready interrupt	• The buffer is ready (readable or writable state)	Host or device	BRDYSTS.PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• USB_OVRCURA or USB_OVRCURB input pin state change was detected (low to high or high to low)	Host	INTSTS1.OVRCR
BCHG	Bus change interrupt	• USB bus state change was detected	Host or device	SYSSTS0.LNST[1:0]
DTCH	Disconnect detection during full-speed operation	• Peripheral device disconnect was detected in full-speed operation	Host	DVSTCTR0.RHST[2:0]
ATTCH	Device connect detection interrupt	• J-state or K-state was detected on the USB bus for 2.5 μs continuously This interrupt can be used to check whether peripheral devices are connected.	Host	—
EOFERR	EOF error detection interrupt	• An EOF error was detected for a peripheral device	Host	—

Table 32.14 Interrupt sources (3 of 3)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
SACK	Setup normal interrupt	• A setup transaction normal response (ACK) was received	Host	—
SIGN	Setup error interrupt	• A setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

Figure 32.10 shows the circuits related to the USBFS interrupts.

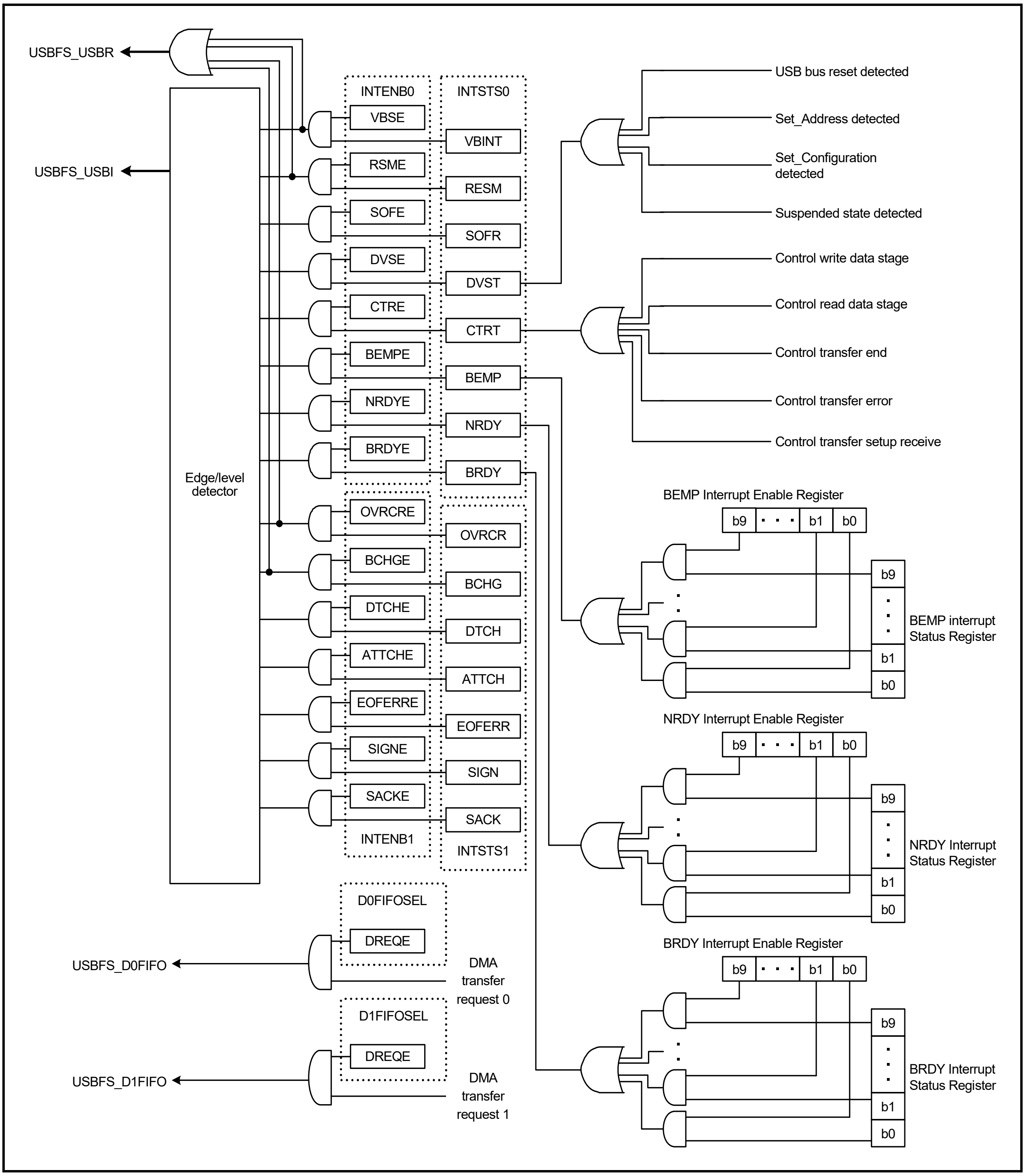


Figure 32.10 USBFS interrupt-related circuits

Table 32.15 shows the interrupts generated by the USBFS.

Table 32.15 USBFS interrupts

Interrupt name	Interrupt status flag	DTC activation	DMAC activation	Priority
USBFS_D0FIFO	DMA transfer request 0	Possible	Possible	High
USBFS_D1FIFO	DMA transfer request 1	Possible	Possible	↑
USBFS_USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnect detection interrupt during full-speed operation, device connect detection interrupt, EOF error detection interrupt, normal setup operation interrupt, and setup error interrupt	Not possible	Not possible	Low
USBFS_USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, and bus change interrupt	Not possible	Not possible	—

32.3.3 Interrupt Descriptions

32.3.3.1 BRDY interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBFS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBFS generates a BRDY interrupt if the software has set the bit in BRDYENB associated with the given pipe to 1 and the INTENB0.BRDYE bit to 1.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

(1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBFS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEnBRDY bit associated with the selected pipe to 1.

(a) For transmitting pipes

- When the DIR bit is changed from 0 to 1 by software
- When packet transmission is complete for a pipe while write-access from the CPU to the FIFO buffer for the pipe is disabled (when the BSTS bit is read as 0)
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete
- When the hardware flushes the buffer of the pipe for isochronous transfers
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

(b) For receiving pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS bit is read as 0). No request trigger is generated for transactions in which a DATA-PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.

In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEnBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEnBRDY bit through software. In this case, the other PIPEnBRDY bit should be set to 1.

Clear the BRDY status before accessing the FIFO buffer.

(2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBFS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets the bit in BRDYSTS associated with the pipe to 1.

On any of the following conditions, the USBFS determines that the last data for a single transfer was received.

- When a short packet including a zero-length packet is received
- When the PIPEn transaction counter register (PIPEnTRN) is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the data is completely read after any of these conditions is satisfied, the USBFS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBFS determines that all data for a single transfer is completely read when the FRDY bit in the FIFO port control register is 1 and the DTLN[8:0] bits are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through the software. With these settings, the USBFS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEnBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEnBRDY bit through the software. In this case, 1s must be written to the PIPEnBRDY bits for the other pipes.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

(3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEnBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For transmitting pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

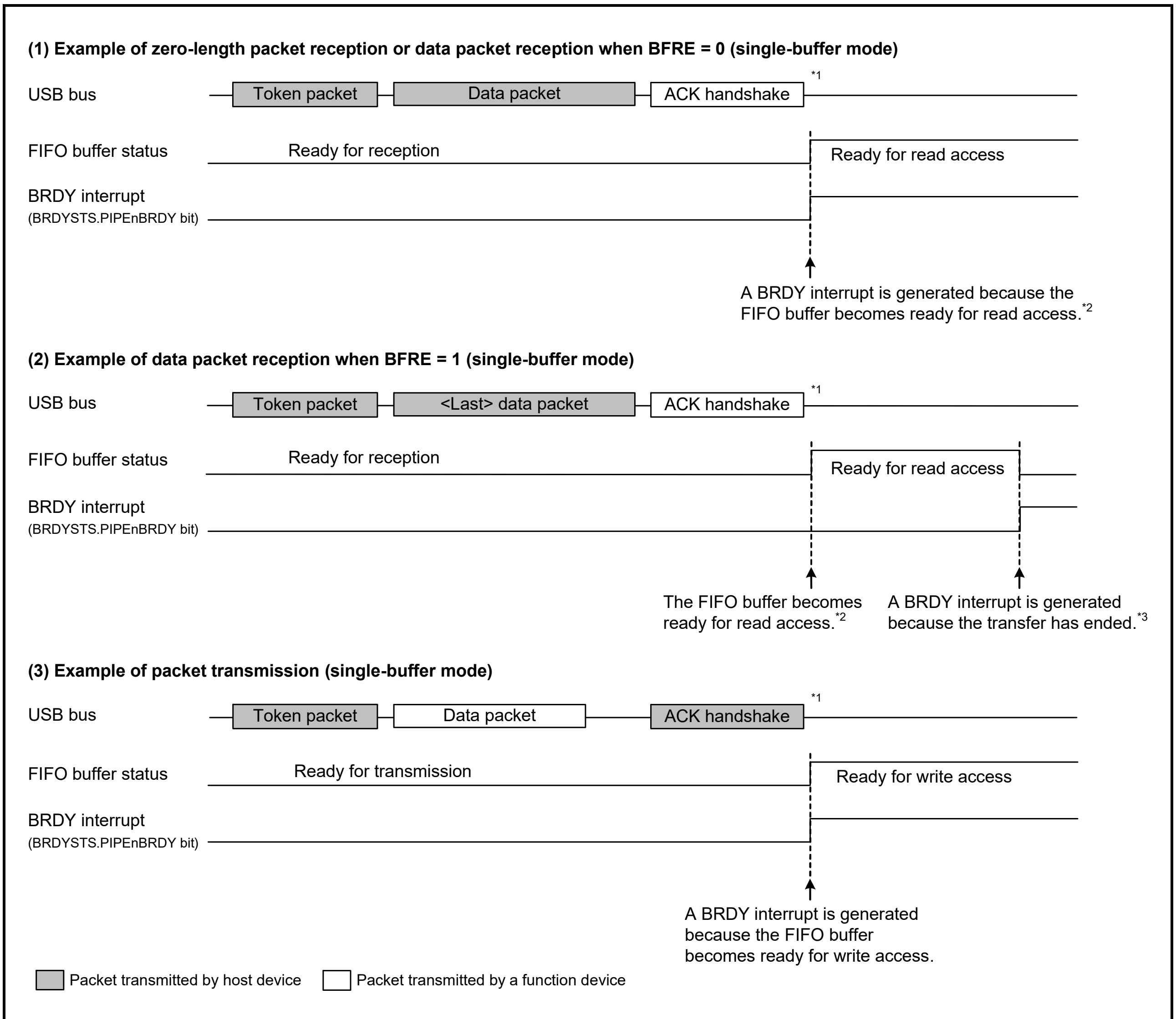
(b) For receiving pipes

The BRDY interrupt status bits set to 1 when the FIFO buffer is ready for read access, and set to 0 when all data is read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEnBRDY bit cannot be set to 0 by software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0.

Figure 32.11 shows the timing of BRDY interrupt generation.



Note 1. The ACK handshake is not used in isochronous transfers.
Note 2. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no data remains unread in the FIFO buffer in the CPU.
Note 3. A transfer ends under either of the following conditions:
(1) When a short packet including a zero-length packet is received
(2) When the number of packets specified in the transaction counter are received

Figure 32.11 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting, as shown in Table 32.16.

Table 32.16 Conditions for clearing the BRDY bit

BRDYM bit	Condition for clearing BRDY bit
0	When all bits in BRDYSTS are set to 0 by software.
1	PIPEnBRDY when the BSTS bits for all pipes have cleared to 0.

32.3.3.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated PIPEnNRDY bit in NRDYSTS to 1. If the associated bit in NRDYENB is set to 1 by software, the USBFS sets the INTSTS0.NRDY bit to 1 and generates a USBFS interrupt.

This section describes the conditions in which the USBFS generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode.

(1) In host controller mode

(a) For transmitting pipes

On any of the following conditions, the USBFS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPEnNRDY bit and the FRMNUM.OVRN bit to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK.
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the PID[1:0] setting for the associated pipe to STALL (11b).

(b) For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBFS discards the received data for the IN token and sets the PIPEnNRDY bit associated with the pipe and the OVRN bit to 1. When a packet error is detected in the received data for the IN token, the USBFS also sets the FRMNUM.CRCE bit to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBFS (when timeout is detected before detection of the DATA packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK.
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe and the CRCE bit to 1.
- When the STALL handshake is received. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

(2) In device controller mode

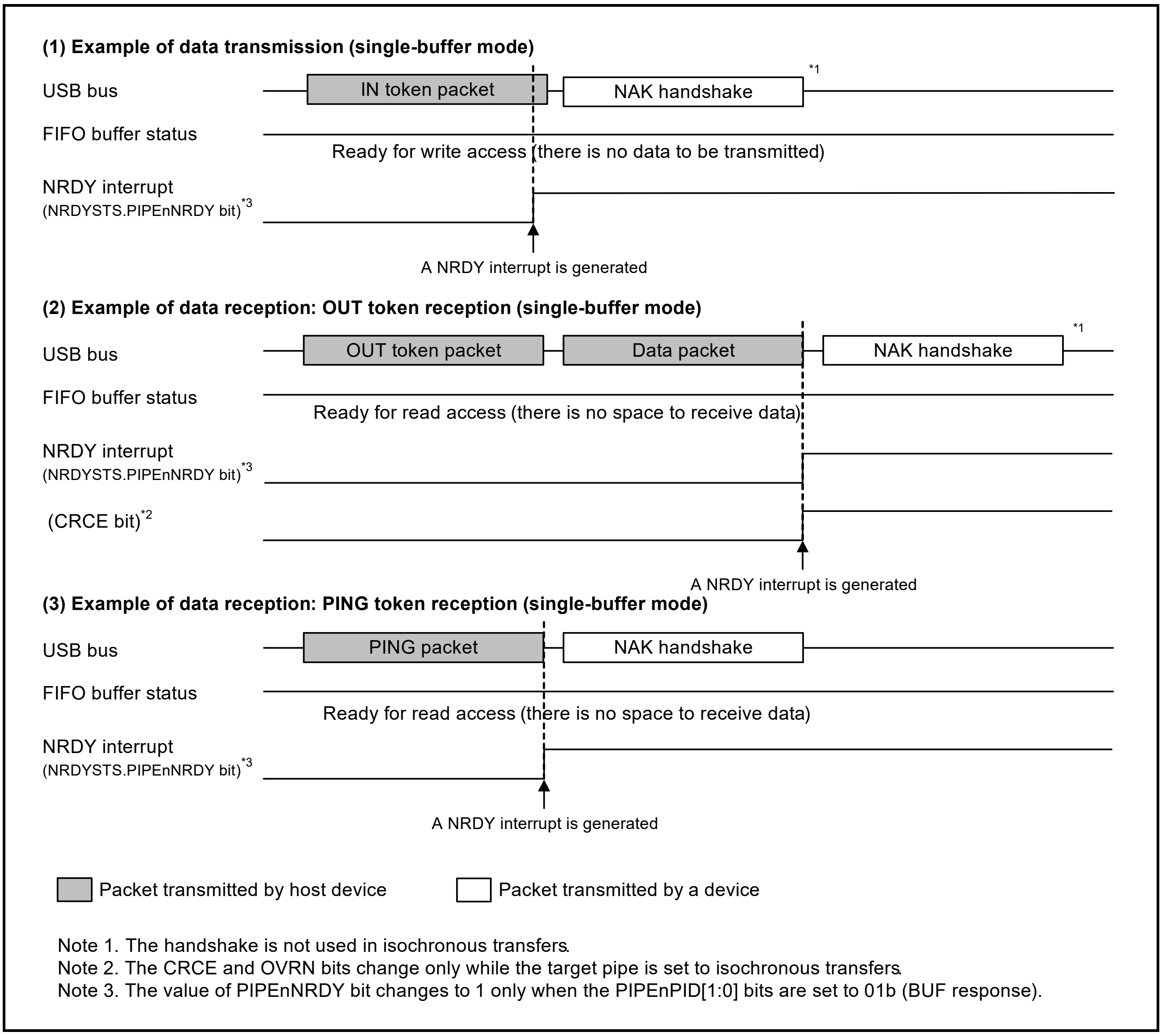
(a) For transmitting pipes

- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS generates a NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPEnNRDY bit to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBFS transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

(b) For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request on reception of the OUT token and sets the PIPEnNRDY bit to 1 and OVRN bit to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1. The NRDY interrupt request is not generated during retransmission because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.
- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBFS generates an NRDY interrupt request when the SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 32.12 shows the timing of NRDY interrupt generation in device controller mode.



Note 1. The handshake is not used in isochronous transfers.
Note 2. The CRCE and OVRN bits change only while the target pipe is set to isochronous transfers.
Note 3. The value of PIPEnNRDY bit changes to 1 only when the PIPEnPID[1:0] bits are set to 01b (BUF response).

Figure 32.12 Timing of NRDY interrupt generation in device controller mode

32.3.3.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated BEMPSTS.PIPEnBEMP bit to 1. If the associated bit in BEMPENB is set to 1 by software, the USBFS sets the INTSTS0.BEMP bit to 1 and generates a USBFS interrupt. This section describes the conditions in which the USBFS generates an internal BEMP interrupt request.

(1) For transmitting pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated in any of the following conditions:

- When the CPU or DMA/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit to 1 in the port control register
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode.

(2) For receiving pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBFS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBFS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated in any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is being performed:
  - Writing 0 to the BEMPSTS.PIPEnBEMP bit clears the status
  - Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect.

Figure 32.13 shows the timing of BEMP interrupt generation in device controller mode.

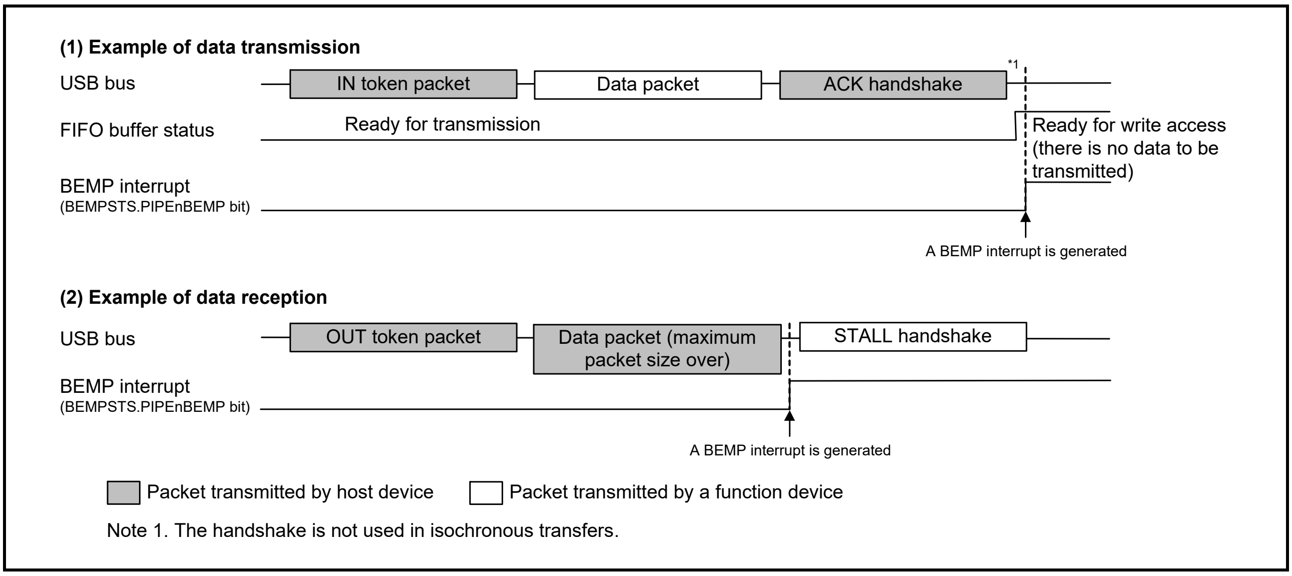


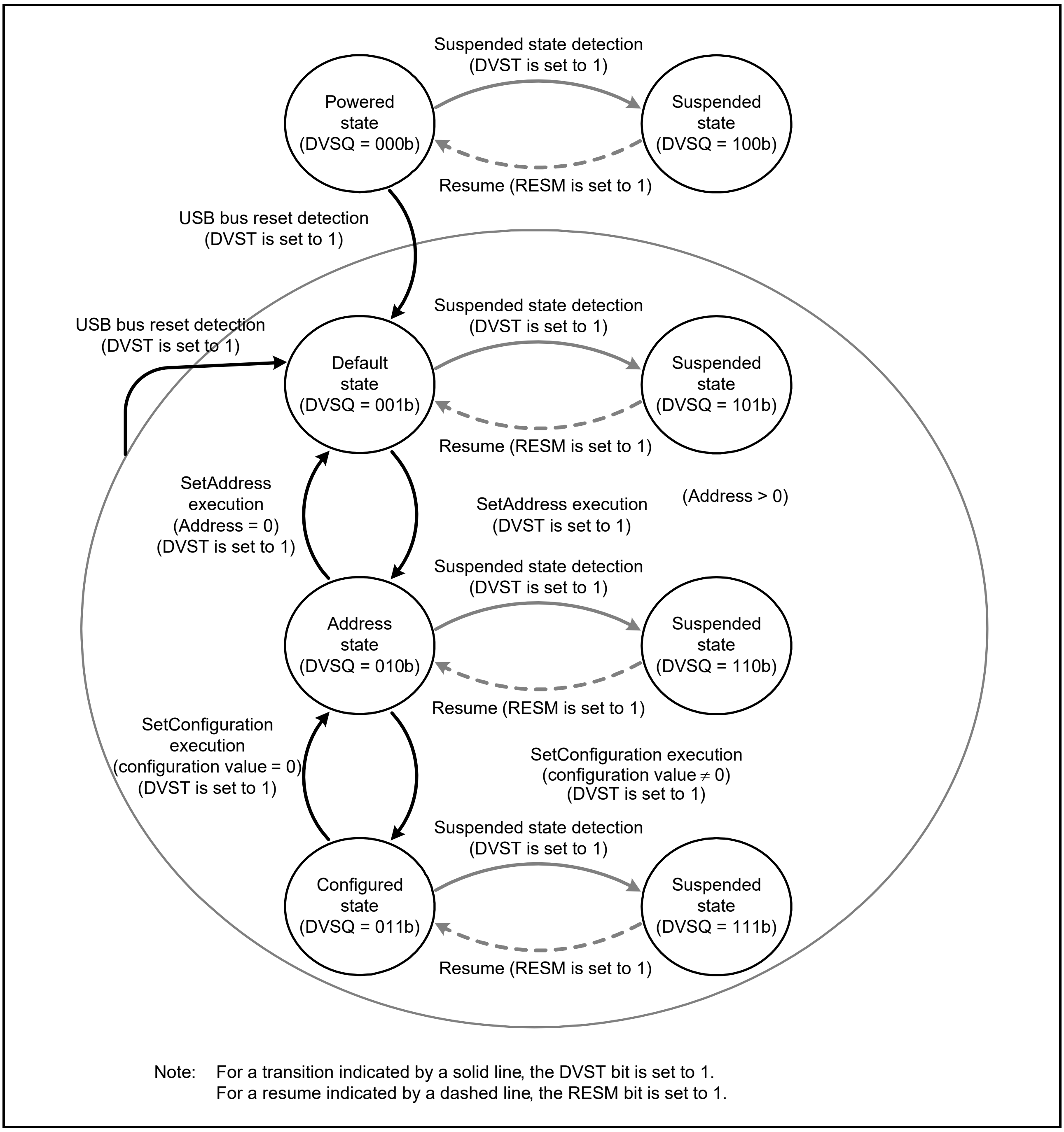
Figure 32.13 Timing of BEMP interrupt generation in device controller mode

32.3.3.4 Device state transition interrupt (device controller mode)

Figure 32.14 shows a diagram of the USBFS device state transitions. The USBFS controls device states and generates device state transition interrupts. However, recovery from the Suspend state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the INTSTS0.DVSQ[2:0] bits.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBFS controls device states, and device state transition interrupts can be generated, only in device controller mode.



Note: For a transition indicated by a solid line, the DVST bit is set to 1.
For a resume indicated by a dashed line, the RESM bit is set to 1.

Figure 32.14 Device state transitions

32.3.3.5 Control transfer stage transition interrupt (device controller mode)

Figure 32.15 shows a diagram of the control transfer stage transitions of the USBFS. The USBFS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

(1) Control read transfer errors

- An OUT token is received but no data is transferred in response to the IN token at the data stage
- An IN token is received at the status stage
- A data packet with DATAPID = DATA0 is received at the status stage.

(2) Control write transfer errors

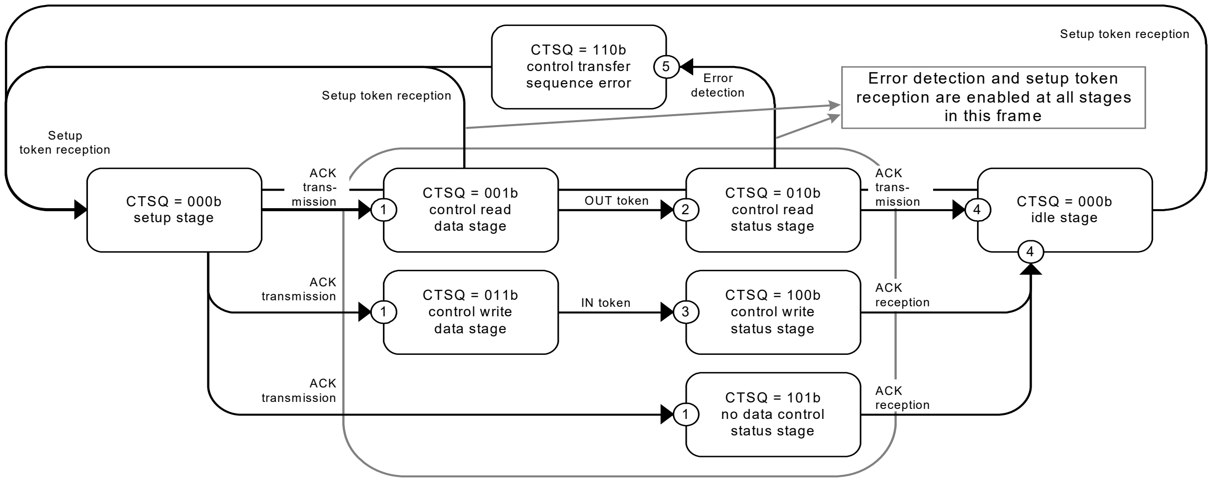
- An IN token is received but no ACK is returned in response to the OUT token at the data stage
- A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
- An OUT token is received at the status stage.

(3) Control write no data transfer errors

- An OUT token is received at the status stage.

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), the CTSQ[2:0] = 110b value is saved until the CTRT bit is set to 0, clearing the interrupt status. While CTSQ[2:0] = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBFS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by software.



Note: CTRT interrupts

① Setup stage completed

② Control read transfer status stage transition

③ Control write transfer status stage transition

④ Control transfer completed

⑤ Control transfer sequence error

Figure 32.15 Control transfer stage transitions

32.3.3.6 Frame update interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBFS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

32.3.3.7 VBUS interrupt

When the USB_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB_VBUS pin can be checked with the INTSTS0.VBSTS bit. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USB_VBUS pin level.

32.3.3.8 Resume interrupt

In device controller mode, a resume interrupt is generated when the device state is the Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

32.3.3.9 OVRCR interrupt

An OVRCR interrupt is generated when the USB_OVRCURA or USB_OVRCURB pin level has changed. The levels of the USB_OVRCURA and USB_OVRCURB pins can be checked in the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent is detected using the OVRCR interrupt.

For OTG connections, the OVRCR interrupt allows you to check whether a change is detected in the VBUS comparator.

32.3.3.10 BCHG interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected and can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated in both host and device controller modes.

32.3.3.11 DTCH interrupt

A DTCH interrupt occurs when a USB bus disconnect is detected in host controller mode. The USBFS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state.

32.3.3.12 SACK interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

32.3.3.13 SIGN interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet is not correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

32.3.3.14 ATTCH interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller mode. To be more specific, an ATTCH interrupt is detected on any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

32.3.3.15 EOFERR interrupt

An EOFERR interrupt occurs when the USBFS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state.

32.3.4 Pipe Control

Table 32.17 lists the pipe settings for the USBFS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBFS provides 10 pipes that are used for data transfer. Set up the pipes based on your system specifications.

Table 32.17 Pipe settings

Register name	Bit name	Setting	Notes
DCPCFG PIPECFG	TYPE	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM	Endpoint number	Pipes 1 to 9: Settable A value other than 0000b must be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Settable only for bulk transfers Pipes 3 to 5: Settable
DCPMAXP PIPEMAXP	DEVSEL	Device select	Referenced only in host controller mode.
	MXPS	Maximum packet size	Compliant with the USB 2.0 specification.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 9: Setting disabled
	IITV	Interval counter	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Monitors the data toggle bit
	PBUSY	Pipe busy status	-
	PID	Response PID	See section 32.3.4.6, Response PID.
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT	Transaction counter	Pipes 1 to 5: Settable

32.3.4.1 Pipe control register switching procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID = NAK).

Do not change the following registers and bits when USB communication is enabled (PID = BUF):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN.

To set these bits when USB communication is enabled (PID = BUF):

1. A request to change the bits in the pipe control register occurs.
2. Set the PID[1:0] bits associated with the pipe to NAK.
3. Wait until the associated PBUSY bit clears to 0.
4. Set the bits in the pipe control register.

The following bits in the pipe control registers can be changed only when the selected pipe information is not set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers when the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI.

To change pipe information, you must set the CURPIPE[3:0] bits in the port select registers to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit in the Port Control Register after the pipe information is changed.

32.3.4.2 Transfer types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting is necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer.

32.3.4.3 Endpoint number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique.

32.3.4.4 Maximum packet size setting

Specify the maximum packet size for each pipe in the DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[8:0] bits. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID = BUF):

- DCP: Set to 8, 16, 32, or 64
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for bulk transfers
- Pipes 1 and 2: Set between 1 and 256 for isochronous transfers
- Pipes 6 to 9: Set between 1 and 64.

32.3.4.5 Transaction counter for pipes 1 to 5 in the receiving direction

When the specified number of transactions is complete in the data packet receiving direction, the USBFS recognizes that the transfer ended. Two transaction counters are provided: one is the PIPEnTRN register, which specifies the number of transactions to be executed, and the other is the current counter, which internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read.

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared.

32.3.4.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBFS operation with different response PID settings.

(1) Software response PID settings in host controller mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:
    - OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
    - IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.
- STALL setting: Using pipes is disabled and no transactions are executed.

Note: Use the DCPCTR.SUREQ bit to execute setup transactions for the DCP.

(2) Software response PID settings in device controller mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions
- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions.

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

Sections (3) and (4) describe situations in which the USBFS writes to the PID[1:0] bits because of specific transaction results.

(3) Hardware response PID settings in host controller mode

- NAK setting: PID = NAK is set in the following cases, and issuing of tokens is automatically stopped:
    - When a non-isochronous transfer is performed and an NRDY interrupt is generated (For details, see section 32.3.3.2, NRDY interrupt.)
    - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
    - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers.
- BUF setting: The USBFS does not write this setting.

- STALL setting: PID = STALL is set in the following cases, and issuing of tokens is automatically stopped:
  - When STALL is received in response to a transmitted token
  - When a received data packet exceeds the maximum packet size.

(4) Hardware response PID settings in device controller mode

- NAK setting: PID = NAK is set in the following cases, and a NAK response is returned to transactions:
  - When the setup token is received normally (DCP only)
  - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers.
- BUF setting: There is no BUF writing by the USBFS.
- STALL setting: PID = STALL is set in the following cases, and a STALL response is returned to transactions:
  - When a received data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected (DCP only).

32.3.4.7 Data PID sequence bit

The USBFS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR and SQSET bits in DCPCTR and PIPEnCTR registers can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBFS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by software in both host and device controller modes.

32.3.4.8 Response PID = NAK function

The USBFS provides a function for disabling pipe operation (PID response = NAK) when the final data packet of a transaction is received. The USBFS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, the software must enable the pipe again (PID response = BUF).

The response PID = NAK function can be used only for bulk transfers.

32.3.4.9 Auto response mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

32.3.4.10 OUT-NAK mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK). Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

32.3.4.11 Null auto response mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode while pipe operation is disabled (response PID = NAK). Next enable pipe operation (response PID = BUF) on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (response PID = NAK) for the period of the zero-length packet transmission (about 10 µs) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

32.3.5 FIFO Buffer

The USBFS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBFS (SIE side).

(1) Buffer status

Table 32.18 and Table 32.19 show the buffer status in the USBFS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in either the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 0 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMA/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

Table 32.18 Buffer status indicated in the BSTS bit

ISEL or DIR	BSTS	FIFO buffer status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet is received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	Transmission has not completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	Transmission is complete. CPU write is allowed.

Table 32.19 Buffer status indicated in the INBUFM bit

DIR	INBUFM	FIFO buffer status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	Transmission is complete. There is no data waiting to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

32.3.6 FIFO Buffer Clearing

Table 32.20 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using BCLR bit in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

Table 32.20 Buffer clearing methods

FIFO buffer clearing mode	Clearing FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto buffer clear mode function

The USBFS discards all received data packets if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

32.3.7 FIFO Port Functions

Table 32.21 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zero-length packet is received (DTLN[8:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[8:0] bits in the port control register.

Table 32.21 FIFO port function settings

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[8:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMA/DTC transfers (only for DnFIFO)
	MBW	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN	Checks the length of receive data

(1) FIFO port selection

Table 32.22 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port select register. After the pipe is selected, the software must check whether the written value can be read correctly from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBFS is modifying the pipe.) Next, the software checks that the FRDY bit in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW bit in the port select register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port select register determines the direction.

Table 32.22 FIFO port access by pipe

Pipe	Access method	Ports that can be used
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	• CFIFO port register • D0FIFO/D1FIFO port register
	DMA/DTC access	D0FIFO/D1FIFO port register

(2) REW bit

It is possible to temporarily stop access to a pipe currently being accessed, access a different pipe, and then continue processing for the first pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

32.3.8 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When buffer access for a pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW bit, and select the pipe targeted for the DMA transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMA transfer.

(2) DnFIFO auto clear mode (D0FIFO and D1FIFO port reading direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBFS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 32.23 shows the packet reception and FIFO buffer clearing processing by software for each of the settings. As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

Table 32.23 Packet reception and FIFO buffer clearing processing by software

Buffer status when packet is received	Register setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing required	No clearing required	No clearing required

32.3.9 Control Transfers Using the DCP

The DCP is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

32.3.9.1 Control transfers in host controller mode

(1) Setup stage

The USQREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the registers and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit clears to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits cleared to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issue setup transactions using this sequence with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

A DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON bit.

(2) Data stage

The data stage is used to transfer data using the DCP FIFO buffer.

Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and set the PID bits = BUF. Completion of data transfer is detected using the BRDY or BEMP interrupt.

For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, the software must send a zero-length packet at the end.

(3) Status stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transfered using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[8:0] bits after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

32.3.9.2 Control transfers in device controller mode

(1) Setup stage

The USBFS sends an ACK response to a normal setup packet for the USBFS. The USBFS operates in the setup stage as follows:

On receiving a new setup packet, the USBFS sets the following bits:

- Sets the INTSTS0.VALID bit to 1
- Sets the DCPCTR.PID[1:0] bits to NAK
- Sets the DCPCTR.CCPL bit to 0.

When the USBFS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID bit = 1, PID

= BUF cannot be set, and the data stage cannot be terminated.

Using the VALID bit function, the USBFS can suspend a request being processed when it receives a new USB request during a control transfer and return a response to the latest request.

In addition, the USBFS automatically detects the direction bit (bmRequestType bit [8]) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and no-data control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBFS, see Figure 32.15.

(2) Data stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF.

After this setting is made, the USBFS automatically executes the status stage based on the data transfer direction determined at the setup stage. The procedure is as follows:

- For control read transfers
  The USBFS receives a zero-length packet from the USB host and transmits an ACK response.
- For control write transfers and no-data control transfers
  The USBFS transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control transfer auto response function

The USBFS automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wLength is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error.

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

32.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (single/double buffer setting) is configurable for bulk transfers. The USBFS provides the following functions for bulk transfers:

- BRDY interrupt function (PIPECFG.BFRE bit), see section 32.3.3.1, (2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1
- Transaction count function (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits), see section 32.3.4.5, Transaction counter for pipes 1 to 5 in the receiving direction
- Response PID = NAK function (PIPECFG.SHTNAK bit), see section 32.3.4.8, Response PID = NAK function
- Auto response mode (PIPEnCTR.ATREPM bit), see section 32.3.4.9, Auto response mode.

32.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBFS performs interrupt transfers based on the timing dictated by the host controller.

In host controller mode, the software can set the timing for issuing tokens using the interval counter.

32.3.11.1 Interval counter for interrupt transfers in host controller mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBFS issues interrupt transfer tokens based on this interval.

(1) Counter initialization

The USBFS initializes the interval counter under the following conditions:

- Power-on reset
  This initializes the IITV[2:0] bits.
- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

The interval counter is not initialized in the following case:

- USB bus reset or USB suspended
  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value saved before entering the USB bus reset state or USB suspend state.

(2) Operation when tokens cannot be transmitted or received even on token generation

No token is generated in the following cases even at token generation time. In these cases, the USBFS tries to execute the transaction in the next interval.

- When the PID is set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT) direction.

32.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBFS provides the following functions for isochronous transfers:

- Notification of isochronous transfer error
- Interval counter specified in the PIPEPERI.IITV[2:0] bits
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function specified in the PIPEPERI.IFIS bit.

32.3.12.1 Error detection in isochronous transfers

The USBFS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by software. Table 32.24 and Table 32.25 show the priority order for errors detected by the USBFS and the associated interrupts.

(a) PID errors

- The PID value of the received packet is invalid.

(b) CRC errors and bit stuffing errors

- A CRC error is found in a received packet or the bit stuffing is illegal.

(c) Maximum packet size exceeded

- The data size of the received packet exceeds the specified maximum packet size.

(d) Overrun and underrun errors

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction.

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction.

(e) Interval errors

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer.

Table 32.24 Error detection for token transmission and reception

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
3	Overrun or underrun error	An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to OUT token.
4	Interval error	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

Table 32.25 Error detection for data packet reception

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated (ignored as a corrupted packet)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit sets to 1 in both host and device controller modes
3	Maximum packet size exceeded error	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes

32.3.12.2 DATA-PID

In device controller mode, the USBFS responds to a received PID as follows:

(1) IN direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mDATA: Not transmitted.

(2) OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mDATA: Packets ignored.

32.3.12.3 Interval counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables the functions as shown in Table 32.26. In host controller mode, the USBFS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.

Table 32.26 Interval counter functions in device controller mode

Transfer direction	Function	Conditions for detection
IN	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer.
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer.

The interval count is performed when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to 2^{IITV} frames.

(1) Counter initialization in device controller mode

The USBFS initializes the interval counter under the following conditions:

- Power-on reset
  This initializes the PIPEPERI.IITV[2:0] bits.
- FIFO buffer initialization using the ACLRM bit
  This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under one of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token when PID = BUF
- An SOF is received after data is received in response to an OUT token when PID = BUF.

The interval counter is not initialized in the following conditions:

- When the PID[1:0] bits are set to NAK or STALL
  This does not stop the interval timer. The USBFS attempts the transaction in the next interval.
- When the USB bus is reset or USBFS is suspended
  This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

(2) Interval counting and transfer control in host controller mode

The USBFS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USBFS issues a token for a selected pipe once every 2^{IITV} frames.

The USBFS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to BUF by software.

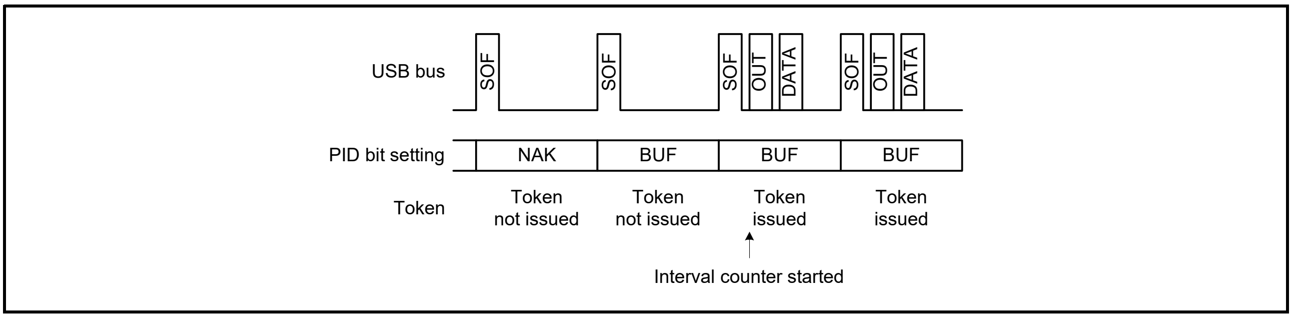


Figure 32.16 Token issuance when IITV = 0

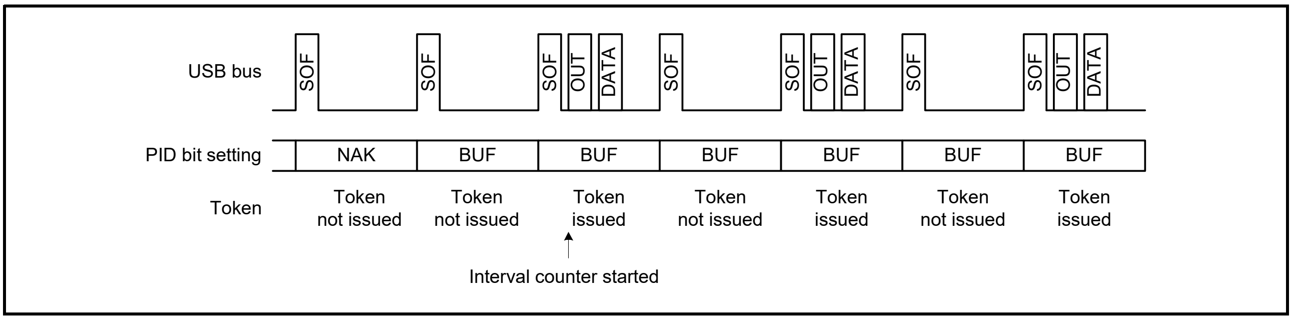


Figure 32.17 Token issuance when IITV = 1

When the selected pipe is set for isochronous transfers, the USBFS performs the following operation in addition to controlling the token issuance interval. The USBFS issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USBFS generates an NRDY interrupt when the USBFS issues an IN token but does not successfully receive a packet from a peripheral device (no response or packet error).

The USBFS sets the FRMNUM.OVRN bit to 1, generating an NRDY interrupt, when the time to issue an IN token occurs while the USBFS cannot receive data because the FIFO buffer is full, because the CPU or DMAC/DTC is too slow in reading data from the FIFO buffer.

(b) When the selected pipe is for isochronous OUT transfers

The USBFS sets the OVRN bit to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, or because the CPU or DMAC/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the USBFS is reset through a reset pin
  This initializes the IITV[2:0] bits.
- When the PIPEnCTR.ACLRM bit is set to 1 by software.

(3) Interval counting and transfer control in device controller mode

(a) When the selected pipe is for isochronous OUT transfers

The USBFS generates an NRDY interrupt when it fails to receive a data packet within the interval set in the PIPEPERI.IITV[2:0] bits.

The USBFS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because the FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal interpolation allows the interrupt to be generated when the SOF packet is received. However, when the IITV bits are set to a value other than 0, the USBFS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to NAK by software after starting the interval timer, the USBFS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:
  Interval counting starts at the next frame after the software changes the PID[1:0] bits of the selected pipe to BUF.
- When the IITV[2:0] bits ≠ 0:
  Interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are changed to BUF.

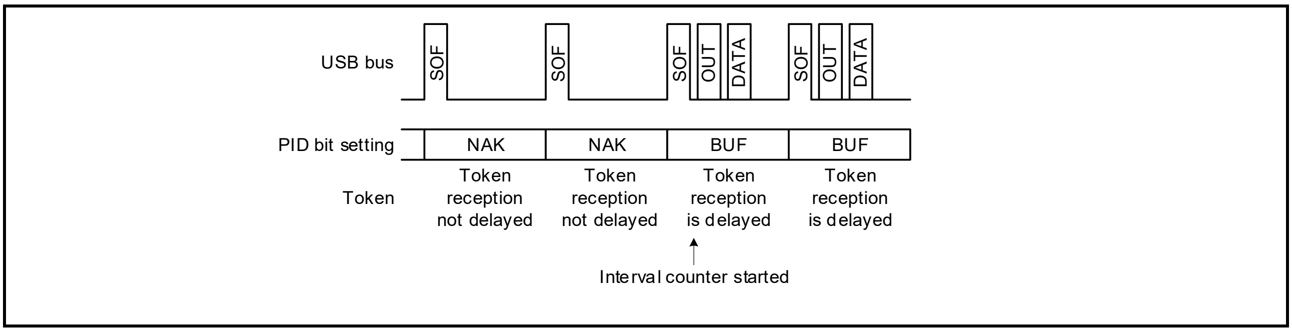


Figure 32.18 Relationship between frames and expected token reception when IITV[2:0] = 0

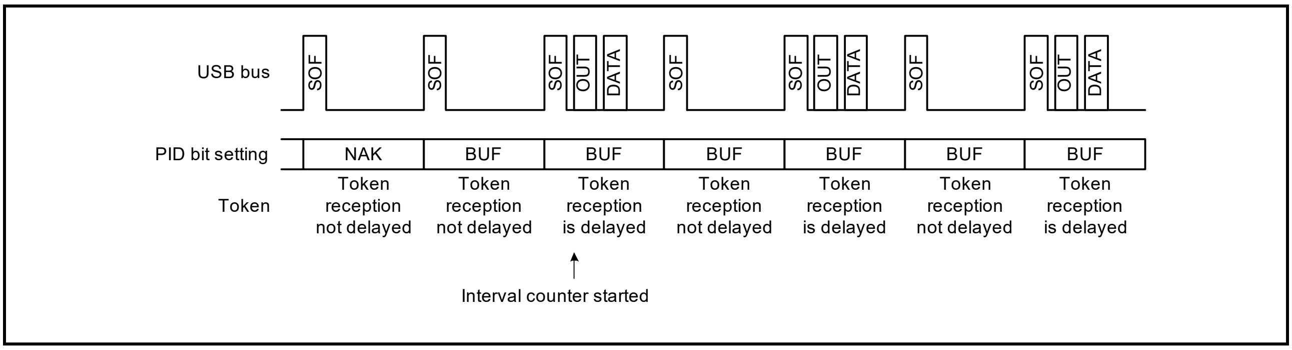


Figure 32.19 Relationship between frames and expected token reception when IITV[2:0] ≠ 0

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit must be 1 for this use case. When IFIS = 0, the USBFS transmits a data packet in response to a received IN token regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBFS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBFS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBFS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLRM bit is set to 1 by software
- When the USBFS detects a USB bus reset.

(4) Transmit data setup for isochronous transfers in device controller mode

With isochronous data transmission using the USBFS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer where data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 32.20 shows an example transmission using the isochronous transfer transmission data setup function when IITV = 0 (every frame) is set.

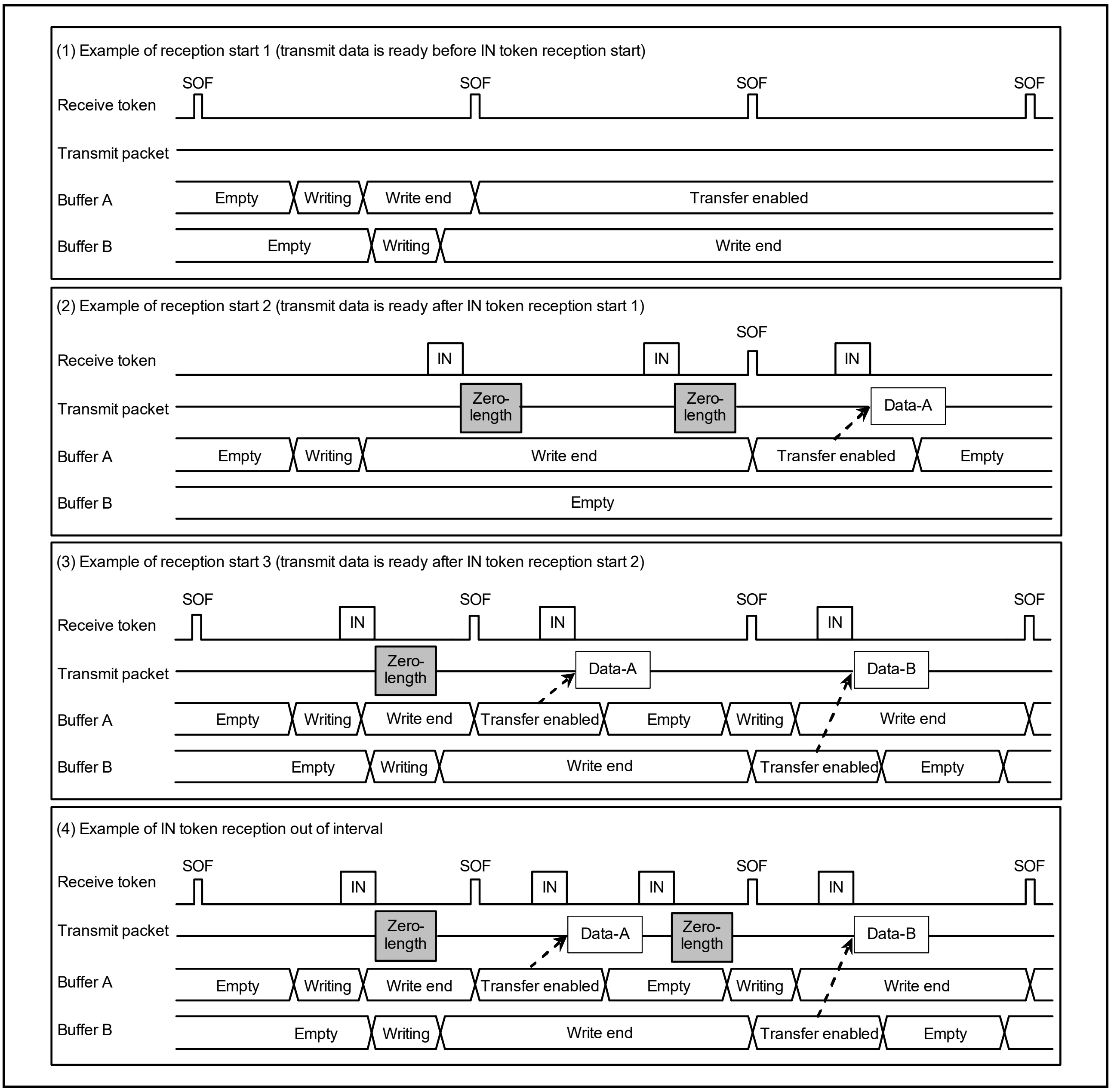


Figure 32.20 Example data setup operation

(5) Transmit buffer flush for isochronous transfers in device controller mode

In device controller mode during isochronous data transmission, if the USBFS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV = 0:
  The buffer flush operation starts from the first frame after the pipe is enabled.

- When IITV ≠ 0:
  The buffer flush operation starts after the first normal transaction.

Figure 32.21 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBFS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.

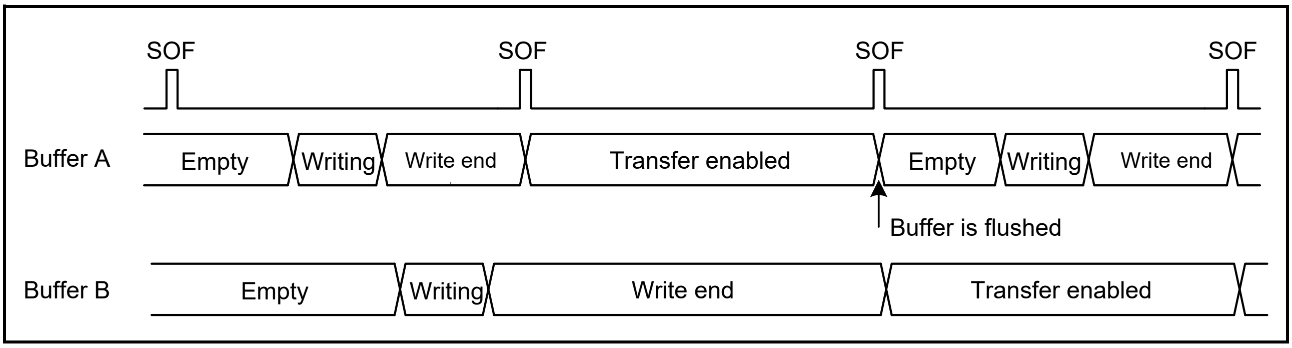


Figure 32.21 Example buffer flush operation

Figure 32.22 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing ①, and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs.
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs.

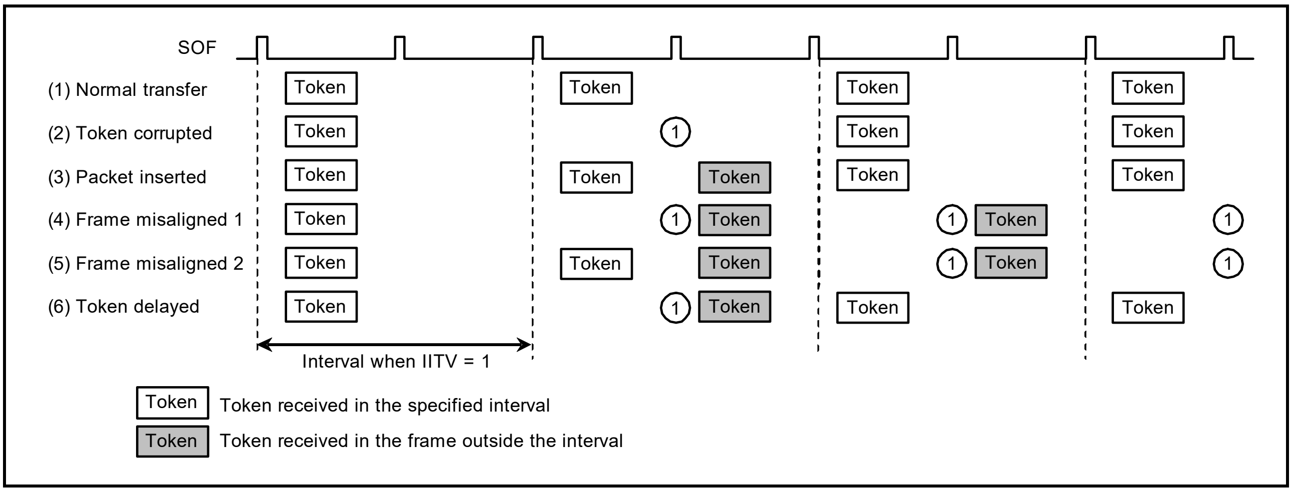


Figure 32.22 Example interval error occurrence when IITV = 1

32.3.13 SOF Interpolation Function

In device controller mode, if packet reception is disabled at intervals of 1 ms because the SOF packet is corrupted or missing, the USBFS interpolates the SOF. SOF interpolation begins when the USBE and SCKE bits in SYSCFG are set to 1 and an SOF packet is received.

The interpolation function is initialized under the following conditions:

- MCU reset
- USB bus reset
- Suspend state detection.

The SOF interpolation operates as follows:

- The interpolation function is not activated until an SOF packet is received.
- When the first SOF packet is received, interpolation is performed by counting 1 ms on the 48-MHz internal clock
- When the second and subsequent SOF packets are received, interpolation is performed at the previous reception interval
- Interpolation is not performed in the Suspend state or on reception of a USB bus reset.

The USBFS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF interpolation if the SOF packet is missing:

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count.

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] bits are not updated.

32.3.14 Pipe Schedule

32.3.14.1 Conditions for generating transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBFS generates transactions under the conditions shown in Table 32.27.

Table 32.27 Conditions for generating transactions

	Conditions for generation				
Transaction	**DIR**	**PID**	**IITV0**	**Buffer state**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.

Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.

Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

32.3.14.2 Transfer schedule

This section describes the transfer scheduling within a frame of the USBFS. After the USBFS sends an SOF, the transfer is carried out in the following sequence:

1. Execution of periodic transfers:
   A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers:
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:
   A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

32.3.14.3 Enabling USB communication

Setting the DVSTCTR0.UACT bit to 1 initiates an SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission and the Suspend state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

32.4 Usage Notes

32.4.1 Settings for the Module-Stop State

USBFS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBFS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

32.4.2 Clearing the Interrupt Status Register on Exiting Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode
- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels software standby is changed in Software Standby mode.

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU exits the Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

32.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR port is set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

33. USB 2.0 High-Speed Module (USBHS)

33.1 Overview

The MCU provides a USB 2.0 High-Speed Module (USBHS) that operates as a host or a device controller compliant with the Universal Serial Bus (USB) Specification revision 2.0. The host controller supports USB 2.0 high-speed, full-speed, and low-speed transfers, and the device controller supports USB 2.0 high-speed and full-speed transfers. The USBHS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification.

The USBHS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

Table 33.1 lists the USBHS specifications, Figure 33.1 shows a block diagram, and Table 33.2 lists the I/O pins.

Table 33.1 USBHS specifications

Parameter	Specifications
Features	• USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, device controller, and On-The-Go (OTG) functions • Software can switch between host and device controller modes.
	Host controller features: • High-speed transfer (480 Mbps), full-speed transfer (12 Mbps), and low-speed transfer (1.5 Mbps) • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Communications with multiple peripheral devices connected through a single hub.
	Device controller features: • High-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF complementation.
Supported transfer types	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer.
Pipe configuration	• FIFO buffer of up to 8.5 KB for USB communications • Up to 10 pipes selectable, including the default control pipe • Programmable pipe configurations • Pipes 1 to 9 assignable to any endpoint number.
	Transfer conditions specifiable for each pipe: • Pipe 0: Control transfer with 64-byte single buffer • Pipes 1 and 2: Bulk isochronous transfer continuous transfer mode with programmable buffer size up to 2 KB and optional double buffer • Pipes 3 to 5: Bulk transfer continuous transfer mode with programmable buffer size up to 2 KB and optional double buffer • Pipes 6 to 9: Interrupt transfer with 64-byte single buffer.
Other features	• Force-end transfer function using transaction count • Function that changes the BRDY interrupt event notification timing • Automatic clearing of the FIFO buffer after data for the pipe specified in the DnFIFO port (n = 0, 1) is read • NAK setting function for response PID generated on transfer end • On-chip pull-up and pull-down resistors for D+ and D- • Support for Link Power Management (LPM) ECN, including a new Sleep state (the L1 state) • Compliance with Battery Charging Class Specification Revision 1.2 • For power reduction, selectable classic-only mode (CL-only mode) in which operation is only USB 1.1-compliant

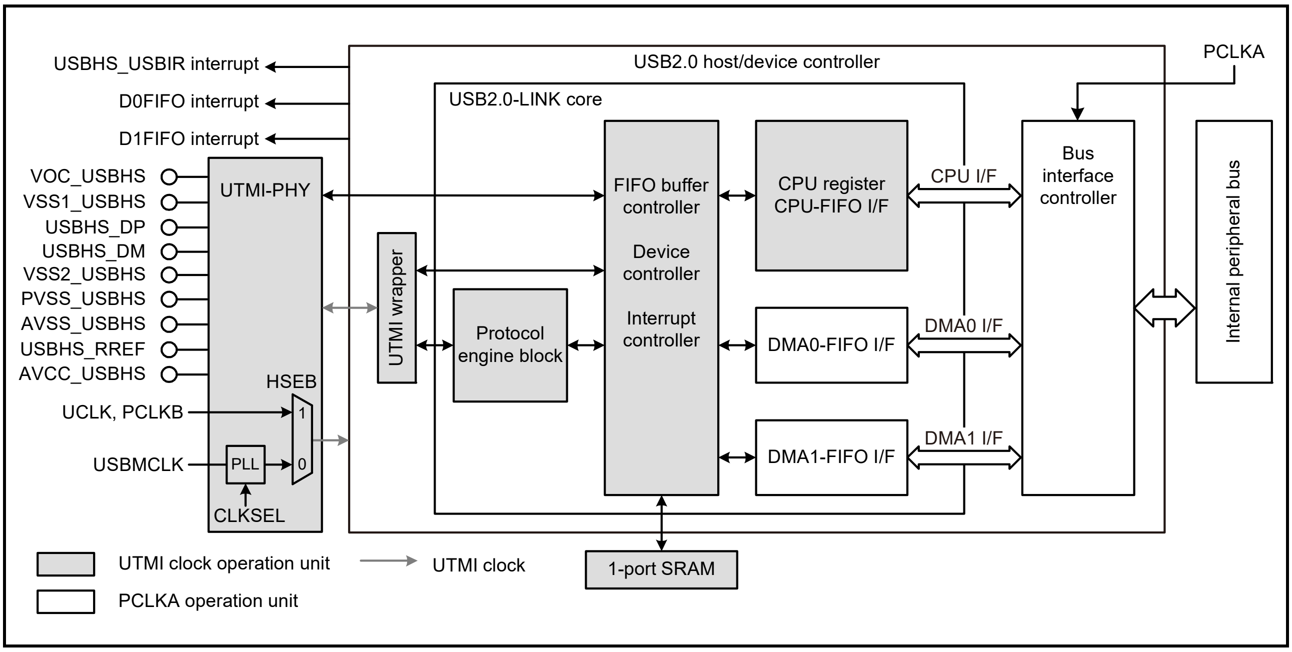


Figure 33.1 USBHS block diagram

Table 33.2 USBHS I/O pins

Pin name	I/O	Function
VCC_USBHS	Input	Power supply pin for the USBHS
VSS1_USBHS VSS2_USBHS	Input	Ground pin for the USBHS
AVCC_USBHS	Input	Analog power supply pin for the USBHS
AVSS_USBHS	Input	Analog ground pin for the USBHS Must be shorted to the PVSS_USBHS pin.
PVSS_USBHS	Input	PLL circuit ground pin for the USBHS Must be shorted to the AVSS_USBHS pin.
USBHS_RREF	I/O	Reference current source pin for the USBHS Must be connected to the AVSS_USBHS pin through a 2.2-kΩ (±1%) resistor.
USBHS_DP	I/O	Input/output pin for the D+ data line of the USB bus
USBHS_DM	I/O	Input/output pin for the D- data line of the USB bus
USBHS_EXICEN	Output	Must be connected to the OTG power supply IC
USBHS_ID	Input	Must be connected to the OTG power supply IC
USBHS_VBUSEN	Output	VBUS power supply enable pin for the USBHS
USBHS_OVRCURA/ USBHS_OVRCURB	Input	Overcurrent pin for the USBHS
USBHS_VBUS	Input	USB cable connection monitor input pin

33.2 Register Descriptions

33.2.1 System Configuration Control Register (SYSCFG)

Address(es): USBHS.SYSCFG 4006 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CNEN	HSE	DCFM	DRPD	DPRPU	—	—	—	USBE
Value after reset:	x	x	x	x	x	x	x	0	0	0	1	0	x	x	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	USBE	USBHS Operation Enable	0: Disable 1: Enable.	R/W
b3 to b1	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Disable line pull-up 1: Enable line pull-up.	R/W
b5	DRPD	D+/D- Line Resistor Control	0: Disable line pull-down 1: Enable line pull-down.	R/W
b6	DCFM	Controller Operation Select	0: Select device controller mode 1: Select host controller mode.	R/W
b7	HSE	High-Speed Operation Enable	0: Disable Device controller mode: full-speed Host controller mode: full- or low-speed. 1: Enable. The controller detects the communication speed.	R/W
b8	CNEN	Single-ended Receiver Enable	0: Disable 1: Enable.	R/W
b15 to b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Writing to the SYSCFG register can proceed while the PHY clock is stopped. However, written values are only reflected in the SYSCFG register after the PHY clock is oscillating again.

USBE bit (USBHS Operation Enable)

The USBE bit enables or disables operation of USBHS.

Changing the USBE bit from 1 to 0 initializes the bits listed in Table 33.3. Only change this bit after specifying the input clock in the PHYSET.CLKSEL[1:0] bits and confirming that the PLLSTA.PLLLOCK flag is 1. In CL-only mode, change the USBE bit after setting the PHYSET.HSEB bit to 1. At that time, the UCLK must be set to 48 MHz and PCLKB must be set to 60 MHz. For the clock settings, see section 33.3.3, Supplying the Clock.

In host controller mode, always set this bit to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] bit chattering, and confirming that the USB bus state is stable.

Table 33.3 Bits initialized by writing SYSCFG.USBE = 0

Selected function	Register	Bit	Remarks
Device controller (DCFM = 0)	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	-
	PL1CTRL1	DVSQ[3:0]	Value is saved in host controller mode
	USBADDR	USBADDR[6:0]	Value is saved in host controller mode
	USBREQ	• BREQUEST[7:0] • BMREQUESTTYPE[7:0].	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller (DCFM = 1)	DVSTCTR0	RHST[2:0]	-
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode
	UFRMNUM	UFRNM[2:0]	Value is saved in device controller mode

DPRPU bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBHS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 1 in device controller mode and to 0 in host controller mode.

DRPD bit (D+/D- Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D- lines in host controller mode.

Set this bit to 1 in host controller mode. Set it to 0 when OTG is not used in device controller mode.

DCFM bit (Controller Operation Select)

The DCFM bit selects the host or device function of the USBHS.

Only change this bit when the DPRPU and DRPD bits are both 0.

HSE bit (High-Speed Operation Enable)

The HSE bit enables or disables high-speed operation.

When this bit is 1, the USBHS operates in high- or full-speed based on the results of the reset handshake.

In host controller mode, setting this bit to 0 allows the USBHS to operate in low- or full-speed. If the DVSTCTR0.RHST[2:0] flags indicate that a low-speed device has attached, set the HSE bit to 0.

In host controller mode, setting this bit to 1 allows the USBHS to operate in high- or full-speed based on the results of the reset handshake. Change the HSE bit after detection of an attach event (ATTCH interrupt) and before the USB bus reset (when DVSTCTR0.USBRST = 1), or after detection of a detach event.

In device controller mode, setting this bit to 0 allows the USBHS to operate in full-speed. Setting the bit to 1 allows the USBHS to perform the reset handshake and then operate in high-speed or full-speed, based on the results.

In device controller mode, only change this bit when the DPRPU bit is 0.

CNEN bit (Single-ended Receiver Enable)

Setting the CNEN bit to 1 enables single-ended receiver operation and selects monitoring of the D+ and D- line states in the SYSSTS0.LNST[1:0] flags. Use this bit to prevent through-current damage that might otherwise be caused during single-ended receiver operation, where the terminals are floating while the USBHS is detached.

In host controller mode, set this bit to 1 after confirming that the PHY clock is being supplied. In device controller mode, set this bit to 1 when the VBUS is detected because of a VBUS interrupt, and set it to 0 when the VBUS line is removed.

33.2.2 CPU Bus Wait Register (BUSWAIT)

Address(es): USBHS.BUSWAIT 4006 0002h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	BWAIT[3:0]			
Value after reset:	x	x	0	0	0	0	0	0	x	x	0	0	1	1	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	BWAIT[3:0]	CPU Bus Access Wait Specification	b3 b0 0 0 0 0: 0 waits (2 access cycles) : : 0 0 1 0: 2 waits (4 access cycles) : : 0 1 0 0: 4 waits (6 access cycles) : : 1 1 1 1: 15 waits (17 access cycles) (initial value).	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	—	Reserved	The read value is undefined. The write value should be 0.	R/W

BWAIT[3:0] bits (CPU Bus Access Wait Specification)

The BWAIT[3:0] bits specify the wait time for access to the USBHS registers.

When accessing the registers at addresses in the range beginning at 4006 0004h, set the cycle time for consecutive access to at least 40.8 ns. The initial value is 1111b (17 cycles), but Renesas recommends that you satisfy this condition by setting the best wait time for the frequency of the CPU clock in your application.

This setting is the same as the wait time for accesses to the FIFO port register. The maximum speed of access to the FIFO port is as follows:

- MBW[1:0] = 10b (32-bit width): Maximum 60 MB/s
- MBW[1:0] = 01b (16-bit width): Maximum 30 MB/s
- MBW[1:0] = 00b (8-bit width): Maximum 15 MB/s

33.2.3 System Configuration Status Register (SYSSTS0)

Address(es): USBHS.SYSSTS0 4006 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flag	Indicates the status of the USB data lines. See Table 33.4.	R
b2	IDMON	USBHS_ID Pin Monitor Flag	0: USBHS_ID pin is low 1: USBHS_ID pin is high.	R
b4, b3	—	Reserved	The read value is undefined.	R

Bit	Symbol	Bit name	Description	R/W
b5	SOFEA	SOF Active Monitor Flag While Host Controller Operation Is Selected	0: SOF output stopped 1: SOF output operating.	R
b6	HTACT	Host Sequencer Status Monitor Flag	0: Host sequencer stopped 1: Host sequencer operating.	R
b13 to b7	—	Reserved	The read value is undefined.	R
b15, b14	OVCMON[1:0]	External USBHS_OVRCURA/USBHS_OVRCURB Input Pin Monitor Flag	OVCMON[1] indicates the USBHS_OVRCURA pin status. OVCMON[0] indicates the USBHS_OVRCURB pin status.	R

LNST[1:0] flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines, D+ and D-. For details, see Table 33.4.

In device controller mode, read the LNST[1:0] flags after setting the SYSCFG.CNEN and SYSCFG.USBE bits to 1. In host controller mode, read them after setting the SYSCFG.DRPD bit to 1.

When you are checking hardware contacts for the battery charging function in device controller mode, read the LNST[1:0] flags after setting the SYSCFG.DRPD, SYSCFG.CNEN, and BCCTRL.IDPSRCE bits to 1. For details, see section 33.3.15, Battery charging detection processing.

Table 33.4 Status of USB data bus lines (D+ and D-)

LNST[1]	LNST[0]	Low-speed operation (host controller mode only)	Full-speed operation	High-speed operation	Chirp operation
0	0	SE0	SE0	Squelch	Squelch
0	1	K-State	J-State	Unsquelch	Chirp J
1	0	J-State	K-State	Invalid	Chirp K
1	1	SE1	SE1	Invalid	Invalid

Chirp:	The reset handshake protocol is being executed when high-speed operation is enabled (HSE bit is 1).
Squelch:	SE0 or idle state
Unsquelch:	High-speed J-state or high-speed K-state
Chirp J:	Chirp J-State
Chirp K:	Chirp K-State

SOFEA flag (SOF Active Monitor Flag While Host Controller Operation Is Selected)

The SOFEA flag is used in host controller mode to check whether the output of the last SOF is complete when the USBHS is suspended because of a 0 setting to the DVSTCTR0.UACT bit.

In host controller mode, check that both the HTACT and SOFEA flags are 0 before setting the SYSCFG.USBE bit to 0 to stop the USBHS or setting the LPSTS.SUSPENDM bit to 0 to stop the clock signal supply during communication.

HTACT flag (Host Sequencer Status Monitor Flag)

The HTACT flag clears to 0 when the host sequencer of the USBHS is completely stopped.

In host controller mode, check that the HTACT flag is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBHS in the Suspend state or setting the LPSTS.SUSPENDM bit to 0 to stop the clock signal supply during communication.

OVCMON[1:0] flags (External USBHS_OVRCURA/USBHS_OVRCURB Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of the overcurrent signals from an external power supply IC.

33.2.4 PLL Status Register (PLLSTA)

Address(es): USBHS.PLLSTA 4006 0006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLLOCK
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	PLLLOCK	PLL Lock Flag	0: PLL not locked 1: PLL locked.	R
b15 to b1	—	Reserved	The read value is undefined.	R

PLLLOCK flag (PLL Lock Flag)

The PLLLOCK flag indicates whether the USB-PHY internal PLL is locked. When not using CL-only mode, make sure that the PLL is locked before starting USB communication.

33.2.5 Device State Control Register 0 (DVSTCTR0)

Address(es): USBHS.DVSTCTR0 4006 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	x	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status Flag	• Host controller mode b2 b0 0 0 0: Communication speed indeterminate (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection 0 1 1: High-speed connection. x: Don't care • Device controller mode b2 b0 0 0 0: Communication speed indeterminate (powered state or no connection) 0 0 1: USB bus reset in progress or low-speed connection 0 1 0: USB bus reset in progress or full-speed connection 0 1 1: USB bus reset in progress or high-speed connection.	R
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b4	UACT	USB Bus Operation Enable for the Host Controller Operation	0: Disable downstream port (disable SOF or micro-SOF transmission) 1: Enable downstream port (enable SOF or micro-SOF transmission).	R/W
b5	RESUME	Resume Signal Output for the Host Controller Operation	0: Do not output resume signal 1: Output resume signal.	R/W
b6	USBRST	USB Bus Reset Output for the Host Controller Operation	0: Do not output USB bus reset signal 1: Output USB bus reset signal.	R/W

Bit	Symbol	Bit name	Description	R/W
b7	RWUPE	Remote Wakeup Detection Enable for the Host Controller Operation	0: Disable downstream port remote wakeup 1: Enable downstream port remote wakeup.	R/W
b8	WKUP	Remote Wakeup Output for the Device Controller Operation	0: Do not output remote wakeup signal 1: Output remote wakeup signal.	R/W
b9	VBUSEN	USBHS_VBUSEN Output Pin Control	0: Output low on external USBHS_VBUSEN pin 1: Output high on external USBHS_VBUSEN pin.	R/W
b10	EXICEN	USBHS_EXICEN Output Pin Control	0: Output low on external USBHS_EXICEN pin 1: Output high on external USBHS_EXICEN pin.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	Use this bit when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the Suspend state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1 is set.	R/W
b15 to b12	—	Reserved	The read value is undefined. The write value should be 0.	R/W

RHST[2:0] flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the USB bus reset status.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] flags to set to 100b. When 0 is written to the USBRST bit and the USBHS ends the SE0 state, the RHST[2:0] flags update to a new value.

In device controller mode, if the USBHS detects a USB bus reset, the RHST[2:0] flags set to 010b if an attach event occurs while the DPRPU bit is 1, and a DVST interrupt is generated.

UACT bit (USB Bus Operation Enable for the Host Controller Operation)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBHS starts SOF packet output within one frame period after the this bit is set to 1. If UACT is set to 0, the USBHS enters the idle state after the SOF packet output.

The USBHS sets the bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (while UACT = 1)
- An EOFERR interrupt is detected during communication (while UACT = 1).

Always write 1 to the UACT bit at the end of the USB bus reset processing (on a 0 write to the USBRST bit) or at the end of resume processing from the Suspend state (on a 0 write to the RESUME bit).

The USBHS clears the UACT bit to 0 if it receives an ACK response to an LPM token while the HL1CTRL1.L1REQ bit is set to 1. The USBHS sets the UACT bit to 1 when it finishes resume processing from the L1 state.

In device controller mode, always set this bit to 0.

RESUME bit (Resume Signal Output for the Host Controller Operation)

The RESUME bit controls the resume signal output in host controller mode.When this bit is set to 1, the USBHS drives the USB port to the K-state and outputs the resume signal. The USBHS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB suspend state. The USBHS continues outputting the K-state while the RESUME bit is 1, until the bit is cleared to 0 by software. The RESUME bit must be 1 (= resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the Suspend state. Write 1 to the UACT bit simultaneously with the end of the resume processing (0 write to the RESUME bit).

Setting the RESUME bit to 1 during transition to the L1 state allows the USBHS to drive the USB port to the K-state and output the resume signal. The USBHS clears the RESUME bit to 0 at the end of the resume period, the value set in the HL1CTRL2.HIRD[3:0] bits.

Always set this bit to 0 in device controller mode.

USBRST bit (USB Bus Reset Output for the Host Controller Operation)

The USBRST bit controls the output of the USB bus signal in host controller mode. When this bit set to 1, the USBHS drives the USB port to the SE0 state to reset the USB bus. The USBHS continues outputting SE0 while the USBRST bit is 1, until the bit is cleared to 0 by software. The USBRST bit must be 1 (= USB bus reset period) for the time defined in the USB 2.0 specification. Writing 1 to the USBRST bit during communication (UACT bit = 1) or during resume processing (RESUME bit = 1) prevents the USBHS from starting USB bus reset processing until both the UACT and RESUME bits clear to 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (0 write to the USBRST bit).

Always set this bit to 0 in device controller mode.

RWUPE bit (Remote Wakeup Detection Enable for the Host Controller Operation)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBHS detects a remote wakeup signal (K-state for 2.5 µs) from a downstream peripheral device, and it performs resume processing, driving the K-state. When the RWUPE bit is set to 0, the USBHS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port.

Do not stop the PHY clock while the RWUPE bit is 1, even in the Suspend state (the LPSTS.SUSPENDM bit must be set to 1). Also, do not reset the USB bus (setting USBRST to 1) from the Suspend state. This is prohibited in the USB 2.0 specification.

The RWUPE bit is also used to enable or disable detection of a remote wakeup signal during transition to the L1 state.

Always set this bit to 0 in device controller mode.

WKUP bit (Remote Wakeup Output for the Device Controller Operation)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBHS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBHS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification dictates that the USB bus idle state must be maintained for 5 ms or longer before a remote wakeup signal is sent. If the USBHS writes 1 to the WKUP bit immediately after detecting the Suspend state, the K-state is output after 2 ms.

Only write 1 to the WKUP bit when the device is in the Suspend state (the PL1CTRL1.DVSQ[3:0] flags are 01xxb) and the USB host enables the remote wakeup signal (RWUPE = 1). Do not stop the PHY clock while this bit is 1, even in the Suspend state (the LPSTS.SUSPENDM bit must be set to 1).

If the WKUP bit is set to 1 during transition to the L1 state, the USBHS outputs the K-state for 50 µs and then clears the bit to 0. Before writing 1 to the bit during the L1 state, check that the PL1CTRL1.DVSQ[3:0] flags are 10xxb.

Always set this bit to 0 in host controller mode.

HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode.

If the HNPBTOA bit is 1, the internal function control maintains the Suspend state until HNP processing ends, even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. Resume interrupts (RESM) are not generated even if a falling edge of D+ is detected.

The HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is cleared to 0 by software because the HNP processing times out.

33.2.6 USB Test Mode Register (TESTMODE)

Address(es): USBHS.TESTMODE 4006 000Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	UTST[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	UTST[3:0]	Test Mode	These bits output the USB test signals. See Table 33.5.	R/W

Bit	Symbol	Bit name	Description	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

UTST[3:0] bits (Test Mode)

Writing values to the UTST[3:0] bits allows the USBHS to output USB test signals in high-speed operation mode. Table 33.5 shows the test mode operation settings.

Table 33.5 Test mode operation settings

	UTST[3:0] bit setting	
Test mode	In device controller mode	In host controller mode
Normal operation	0000b	0000b
Test_J	0001b	1001b
Test_K	0010b	1010b
Test_SE0_NAK	0011b	1011b
Test_Packet	0100b	1100b
Test_Force_Enable	—	1101b
Reserved	0101b to 0111b	1110b to 1111b

Host controller mode

In host controller mode, these bits can be set after setting the SYSCFG.DRPD bit to 1. After the UTST[3:0] bits are set, the USBHS outputs waveforms to the USB port by setting the DVSTCTR0.UACT bit to 1. The USBHS also performs high-speed termination for the USB port by setting these bits in host controller mode.

To set the UTST[3:0] bits in host controller mode:

1. Reset the hardware.
2. Start supplying the PHY clock, and then set the LPSTS.SUSPENDM bit to 1.
3. Set the SYSCFG.DCFM and SYSCFG.DRPD bits to 1. (Setting the SYSCFG.HSE bit to 1 is not required.)
4. Set the SYSCFG.USBE bit to 1.
5. Set the UTST[3:0] bits based on the test requirements.
6. Set the DVSTCTR0.UACT bit to 1.

Assuming the initial steps (1) to (6) are already complete, to change the UTST[3:0] bits in host controller mode:

1. Set the DVSTCTR0.UACT and SYSCFG.USBE bits to 0.
2. Set the SYSCFG.USBE bit to 1.
3. Set the UTST[3:0] bits based on the test requirements.
4. Set the DVSTCTR0.UACT bit to 1.

When the UTST[3:0] bits are set to 1011b (Test_SE0_NAK), the USBHS does not output SOF packets to ports for which the DVSTCTR0.UACT bit is set to 1.

When the UTST[3:0] bits are set to 1101b (Test_Force_Enable), the USBHS outputs SOF packets to ports for which the DVSTCTR0.UACT bit is set to 1. In this test mode, the USBHS does not control the hardware related to attach detection, even if it detects a high-speed detach event (DTCH interrupt).

Before setting the UTST[3:0] bits, set the PID[1:0] bits of all pipe control registers to 00b (NAK response). To return to normal USB communication after setting a test mode, issue a hardware reset.

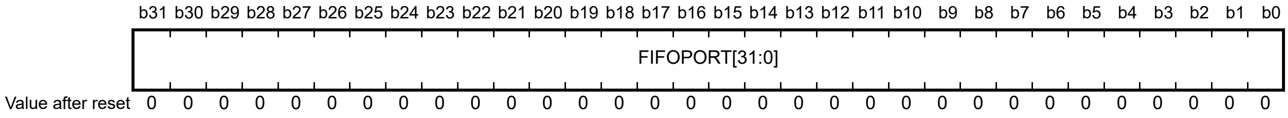
Device controller mode

In device controller mode, set these bits using a SetFeature request from the USB host during high-speed communication. The USBHS does not enter the Suspend state while these bits are 0001b to 0100b. To return to normal USB communication after setting a test mode, issue a hardware reset.

33.2.7 CFIFO Port Register (CFIFO)
D0FIFO Port Register (D0FIFO)
D1FIFO Port Register (D1FIFO)

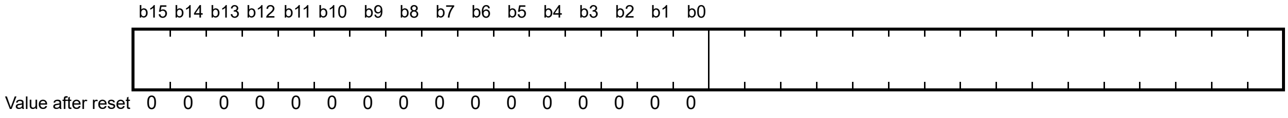
- Access in words

Address(es): USBHS.CFIFO 4006 0014h, USBHS.D0FIFO 4006 0018h, USBHS.D1FIFO 4006 001Ch



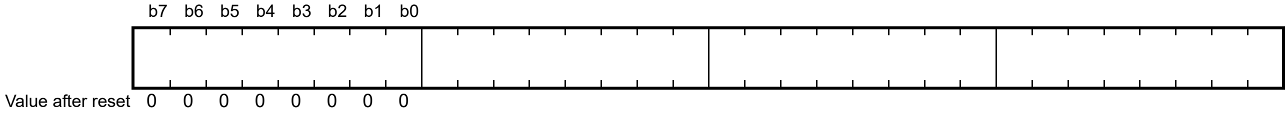
- Access in halfwords

Address(es): USBHS.CFIFOL 4006 0014h, USBHS.CFIFOH 4006 0016h,
USBHS.D0FIFOL 4006 0018h, USBHS.D0FIFOH 4006 001Ah,
USBHS.D1FIFOL 4006 001Ch, USBHS.D1FIFOH 4006 001Eh



- Access in bytes

Address(es): USBHS.CFIFOLL 4006 0014h, USBHS.CFIFOHH 4006 0017h,
USBHS.D0FIFOLL 4006 0018h, USBHS.D0FIFOHH 4006 001Bh,
USBHS.D1FIFOLL 4006 001Ch, USBHS.D1FIFOHH 4006 001Fh



Bit	Symbol	Bit name	Description	R/W
b31 to b0	FIFOPORT*1	FIFO Port	Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits.	R/W

Note 1. The valid bits depend on the MBW[1:0] and BIGEND settings in the associated port selection register.

Three FIFO ports are provided:

- CFIFO
- D0FIFO
- D1FIFO.

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIFO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the Port Selection Register
- Registers configuring one FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports

- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU.

FIFOPORT bit (FIFO Port)

When the FIFOPORT bit is accessed, the USBHS reads the received data from the FIFO buffer or writes the transmission data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY flag in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW[1:0] and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 33.6 to Table 33.8.

Table 33.6 Endian operation in 32-bit access (MBW[1:0] = 10b)

BIGEND	CFIFO, D0FIFO, D1FIFO b31 to b24	CFIFO, D0FIFO, D1FIFO b23 to b16	CFIFO, D0FIFO, D1FIFO b15 to b8	CFIFO, D0FIFO, D1FIFO b7 to b0	Remarks
0	Located at N+3	Located at N+2	Located at N+1	Located at N+0	Transmit data is sent from the address N+0. Receive data is stored from the address N+0.
1	Located at N+0	Located at N+1	Located at N+2	Located at N+3	Transmission data is sent from the address N+3. Receive data is stored from the address N+3.

Table 33.7 Endian operation in 16-bit access (MBW[1:0] = 01b)

BIGEND	CFIFOL, D0FIFOL, D1FIFOL b15 to b8	CFIFOL, D0FIFOL, D1FIFOL b7 to b0	CFIFOH, D0FIFOH, D1FIFOH b15 to b8	CFIFOH, D0FIFOH, D1FIFOH b7 to b0	Remarks
0	Access prohibited*1		Located at N+1	Located at N+0	Transmit data is sent from the address N+0. Receive data is stored from the address N+0.
1	Located at N+0	Located at N+1	Access prohibited*1		Transmit data is sent from the address N+1. Receive data is stored from the address N+1.

Note 1. Writing to or reading from these areas is prohibited.

Table 33.8 Endian operation in 8-bit access (MBW[1:0] = 00b)

BIGEND	CFIFOLL, D1FIFOLL, D0FIFOLL	CFIFOHH, D1FIFOHH, D0FIFOHH
0	Access prohibited*1	Located at N+0
1	Located at N+0	Access prohibited*1

Note 1. Writing to or reading from these locations is prohibited.

33.2.8 CFIFO Port Selection Register (CFIFOSEL)

Address(es): USBHS.CFIFOSEL 4006 0020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	x	x	0	0	x	0	x	x	0	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5	ISEL	FIFO Port Access Direction when DCP Is Selected	0: Select reading from the FIFO buffer 1: Select writing to the FIFO buffer.	R/W
b7, b6	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian.	R/W
b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b11, b10	MBW[1:0]	CFIFO Port Access Bit Width	b11 b10 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Setting prohibited.	R/W
b13, b12	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: Do not rewind buffer pointer (Writing 0 has no effect.) 1: Rewind buffer pointer.	W
b15	RCNT	Read Count Mode	0: Clear DTLN[11:0] flags in the FIFO port control register to 000h when all receive data is read from CFIFO 1: Decrement DTLN[11:0] flags each time receive data is read from CFIFO.	R/W

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number used to read or write data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, the pipe specification is maintained until the access is complete, even if the software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

ISEL bit (FIFO Port Access Direction when DCP Is Selected)

After writing a new value to the ISEL bit while the DCP is the selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set the ISEL and CURPIPE[3:0] bits simultaneously.

BIGEND bit (FIFO Port Endian Control)

Use the BIGEND bit to set the byte endian order of the CFIFO port to be the same as that selected in the endian selection register (MDE).

MBW[1:0] bits (CFIFO Port Access Bit Width)

The MBW[1:0] bits specify the bit width for accessing the CFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, set the CURPIPE[3:0] and MBW[1:0] bits simultaneously. The bit width cannot be changed from 8-bit to 16- or 32-bit, or from 16-bit to 32-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16- or 32-bit width is selected.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double-buffering when reading is already in progress, this setting enables reading either FIFO buffer from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, always check that the FRDY flag is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

When the RCNT bit set to 0, the USBHS clears the CFIFOCTR.DTLN[11:0] flags to 0 on finishing reading all of the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits, or after reading a single plane in double buffer mode.

With this bit set to 1, the USBHS decrements the value in the CFIFOCTR.DTLN[11:0] flags each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits.

33.2.9 D0FIFO Port Selection Register (D0FIFOSEL) D1FIFO Port Selection Register (D1FIFOSEL)

Address(es): USBHS.D0FIFOSEL 4006 0028h, USBHS.D1FIFOSEL 4006 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	x	0	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Specification	b3 b0 0 0 0 0: No pipe specification 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b7 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian.	R/W
b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b11, b10	MBW[1:0]	FIFO Port Access Bit Width	b11 b10 0 0: 8-bit width 0 1: 16-bit width 1 0: 32-bit width 1 1: Setting prohibited.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: Disable DMA/DTC transfer request. 1: Enable DMA/DTC transfer request.	R/W
b13	DCLRM	Auto FIFO Buffer Clear Mode after Specified Pipe is Read	0: Disable auto buffer clear mode 1: Enable auto buffer clear mode.	R/W
b14	REW	Buffer Pointer Rewind	0: Do not rewind buffer pointer (writing 0 has no effect) 1: Rewind buffer pointer.	W
b15	RCNT	Read Count Mode	0: Clear DTLN[11:0] flags in the FIFO port control register to 000h when all receive data is read from DnFIFO (after read of a single plane in double buffer mode) 1: Decrement DTLN[11:0] flags each time receive data is read from DnFIFO. n = 0, 1.	R/W

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number used to read or write data through the DnFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, the pipe specification is maintained until the access is complete, even if the software attempts to change the CURPIPE[3:0] setting. Access continues after the current value is written back to the CURPIPE[3:0] bits.

BIGEND bit (FIFO Port Endian Control)

Use the BIGEND bit to set the byte endian order of the D0FIFO or D1FIFO port to be the same as that selected in the endian selection register (MDE).

MBW[1:0] bits (FIFO Port Access Bit Width)

The MBW[1:0] bits specify the bit width for accessing the DnFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. When reading the FIFO buffer, read with the access size set in MBW.

When the selected pipe is transmitting, set the CURPIPE[3:0] and MBW[1:0] bits simultaneously. The bit width cannot be changed from 8-bit to 16- or 32-bit, or from 16-bit to 32-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16- or 32-bit width is selected.

DREQE bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMA or DTC transfer requests. Only change the settings of DREQE bit when the CURPIPE[3:0] bits are 0000b.

To enable DMA or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits to 0000b, and then set the CURPIPE[3:0] bits to the PIPE number for the transfer.

DCLRM bit (Auto FIFO Buffer Clear Mode after Specified Pipe is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBHS sets the BCLR bit in the FIFO port control register to 1.

When using the USBHS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double-buffering when reading is already in progress, this setting enables reading either FIFO buffer from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, always check that the FRDY flag is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

When the RCNT bit set to 0, the USBHS clears the DnFIFOCTR.DTLN[11:0] flags (n = 0, 1) to 0 on finishing reading all of the received data in the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits, or after reading a single plane in double buffer mode.

With this bit set to 1, the USBHS decrements the value in the CFIFOCTR.DTLN[11:0] flags each time it reads data received from the FIFO buffer assigned to the pipe specified in the CURPIPE[3:0] bits. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

33.2.10 CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USBHS.CFIFOCTR 4006 0022h, USBHS.D0FIFOCTR 4006 002Ah, USBHS.D1FIFOCTR 4006 002Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Value after reset:	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b11 to b0	DTLN[11:0]	Receive Data Length Flag	Receive data length The meaning of the values differs depending on the RCNT bit setting in the port selection register. For details, see the description of the DTLN[11:0] bits.	R
b12	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access disabled 1: FIFO port access enabled.	R
b14	BCLR	CPU Buffer Clear	0: No operation 1: Clear FIFO buffer on the CPU side. Writing 0 to this bit has no effect. This bit is read as 0.	R/W
b15	BVAL	FIFO Buffer Valid Flag	0: Invalid (writing 0 has no effect) 1: Writing ended. Set this bit to 1 when data is completely written to the FIFO buffer on the CPU side for the selected pipe (CURPIPE[3:0] setting).	R/W

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[11:0] flags (Receive Data Length Flag)

The DTLN[11:0] flags indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[11:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0:
  The USBHS sets the DTLN[11:0] flags to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data in the FIFO buffer (or until it has read a single plane in double buffer mode).
  While the PIPECFG.BFRE bit is 1, the USBHS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.
- RCNT = 1:
  The USBHS decrements the value indicated in the DTLN[11:0] flags each time the CPU or DMA/DTC reads the receive data from the FIFO buffer. (The value is decremented by 1 when MBW[1:0] = 00b, by 2 when MBW[1:0] = 01b, and by 4 when MBW[1:0] = 10b.)
  The USBHS sets these flags to 0 when all the data is read from the FIFO buffer. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBHS sets these bits to indicate the length of the receive data in the latter plane when all of the data is read from the former plane.
  When the RCNT bit is 1, reading the DTLN[11:0] flags while the FIFO buffer is being read returns the latest value within 150 ns after the FIFO port read cycle.

FRDY flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBHS sets the FRDY flag to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit is 1.

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBHS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBHS to clear both sets of FIFO buffers regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY flag in the FIFO port control register is 1 (set by the USBHS).

BVAL bit (FIFO Buffer Valid Flag)

Set the BVAL bit to 1 when data is completely written to the FIFO buffer on the CPU for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this bit to 1 in the following cases:

- To transmit a short packet, set this bit to 1 after data is written
- To transmit a zero-length packet, set this bit to 1 before data is written to the FIFO buffer
- Set this bit to 1 after the specified number of data bytes is written for the pipe in continuous transfer mode, where the number is a natural integer multiple of the maximum packet size and less than the buffer size.

The USBHS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When the selected pipe is in use for transmission, simultaneously setting the BVAL flag and the BCLR bit to 1 causes the USBHS to clear the data that is already written and enables transmission of a zero-length packet. When data of the maximum packet size is written for the pipe in non-continuous transfer mode, the USBHS sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBHS). When the selected pipe is receiving, do not set the BVAL flag to 1.

33.2.11 Interrupt Enable Register 0 (INTENB0)

Address(es): USBHS.INTENB0 4006 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b10	BEMPE	Buffer Empty Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b12	DVSE	Device State Transition Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b13	SOFE	Frame Number Update Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b14	RSME	Resume Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15	VBSE	VBUS Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W

Note: The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register sets to 1 and the associated interrupt request enable bit setting in the INTENB0 register is 1, the USBHS issues a USBHS interrupt request.

Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBHS interrupt is requested.

33.2.12 Interrupt Enable Register 1 (INTENB1)

Address(es): USBHS.INTENB1 4006 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	L1RSMENDE	LPMENDE	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	PDDETINTE	PDDETINT Detection Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b3 to b1	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b5	SIGNE	Setup Transaction Error Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b6	EOFERRE	EOF Error Detection Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b8	LPMENDE	LPM Transaction End Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b9	L1RSMENDE	L1 Resume End Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b10	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b12	DTCHE	Disconnection Detection Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b13	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15	OVRCRE	OVRCRE Interrupt Request Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W

When a status flag in the INTSTS1 register sets to 1 and the associated interrupt request enable bit setting in the INTENB1 register is 1, the USBHS issues a USBHS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBHS interrupt is requested.

33.2.13 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USBHS.BRDYENB 4006 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	PIPEBRDYE [9:0]	BRDY Interrupt Request Enable for Pipes [9:0]*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Each bit number corresponds to the same pipe number.

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when a BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register sets to 1 and the associated PIPEBRDYEn (n = 9 to 0) bit setting in the BRDYENB register is 1, the INTSTS0.BRDY flag sets to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBHS generates a BRDY interrupt request. While at least one PIPEBRDYEn flag indicates 1, the INTSTS0.BRDY flag sets to 1 when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by software.

33.2.14 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USBHS.NRDYENB 4006 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPENRDYE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	PIPENRDYE [9:0]	NRDY Interrupt Enable for Pipes [9:0]*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Each bit number corresponds to the same pipe number.

The NRDYENB register enables or disables the INTSTS0.NRDY bit to be set to 1 when a NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register sets to 1 and the associated PIPENRDYEn (n = 0 to 9) bit setting in the NRDYENB register is 1, the INTSTS0.NRDY flag sets to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBHS generates a NRDY interrupt request. While at least one PIPEBRDYEn flag indicates 1, the INTSTS0.NRDY flag sets to 1 when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by software.

33.2.15 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USBHS.BEMPENB 4006 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBEMPE[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	PIPEBEMPE [9:0]	BEMP Interrupt Enable for Pipes [9:0]*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Each bit number corresponds to the same pipe number.

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when a BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register sets to 1 and the associated PIPEBEMPEn (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag sets to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBHS generates a BEMP interrupt request. While at least one PIPEBEMPEn flag indicates 1, the INTSTS0.BEMP flag sets to 1 when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by software.

33.2.16 SOF Output Configuration Register (SOFCFG)

Address(es): USBHS.SOFCFG 4006 003Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	INTL	EDGESTS	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	x	0	0	0	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b4	EDGESTS	Interrupt Edge Processing Status Flag*1	Indicates 1 during the edge processing of an edge interrupt output signal.	R
b5	INTL	Interrupt Output Sense Select*2	0: Edge detection 1: Level detection.	R/W
b6	BRDYM	PIPEBRDY Interrupt Status Clear Timing*3	0: Clear BRDY flag through software 1: Clear BRDY flag by the USBHS through a data read from the FIFO buffer or data write to the FIFO buffer.	R/W
b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*4	0: Not low-speed communication 1: Low-speed communication.	R/W
b15 to b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supply to the USBHS.
Note 2. When the INTL bit is set to 0, to stop the PHY clock (LPSTS.SUSPENDM = 0) after clearing the interrupt status, write 0 to the LPSTS.SUSPENDM bit after confirming that the EDGESTS flag is cleared to 0.
Note 3. When setting the BRDYM bit to 1, set the INTL bit to 1.
Note 4. The setting in the TRNENSEL bit is only valid in host controller mode. Even in host controller mode, the setting of this bit has no effect on the transaction-enabled time during high-speed communication.

EDGESTS flag (Interrupt Edge Processing Status Flag)

The EDGESTS flag indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this flag is 0 before stopping the PHY clock.

BRDYM bit (PIPEBRDY Interrupt Status Clear Timing)

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

TRNENSEL bit (Transaction-Enabled Time Select)

When the USB port is in use for full- or low-speed communications, the TRNENSEL bit specifies the timing with which the USBHS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected through a hub. The bit is only valid in host controller mode. Set this bit to 0 when the interface is in use as a device controller.

33.2.17 PHY Setting Register (PHYSET)

Address(es): USBHS.PHYSET 4006 003Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HSEB	—	—	—	REPSTART	—	REPSEL[1:0]		—	—	CLKSEL[1:0]		CDPEN	—	PLLRESET	DIRPD
Value after reset:	x	x	x	x	0	x	0	0	x	x	1	1	0	x	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	DIRPD	Power-Down Control	0: Do not enter low power mode 1: Enter low power mode.	R/W
b1	PLLRESET	PLL Reset Control*1	0: Disable PLL reset control for UTMI_PHY 1: Enable PLL reset control for UTMI_PHY.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CDPEN	Charging Downstream Port Enable	0: Disable downstream port charging 1: Enable downstream port charging.	R/W
b5, b4	CLKSEL[1:0]	Input System Clock Frequency	b5 b4 0 0: 12 MHz 0 1: Setting prohibited 1 0: 20 MHz 1 1: 24 MHz.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	REPSEL[1:0]	Terminating Resistance Adjustment Cycle	b9 b8 0 0: No cycle is set 0 1: Adjust terminating resistance at 16-second intervals 1 0: Adjust terminating resistance at 64-second intervals 1 1: Adjust terminating resistance at 128-second intervals.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	REPSTART	Forcibly Start Terminating Resistance Adjustment	0: Force terminating resistance adjustment to start 1: Do not force terminating resistance adjustment to start.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	HSEB	CL-only mode	0: Disable CL-only mode 1: Enable CL-only mode.	R/W

Note 1. Because the value of the PLLRESET bit is 1 after a reset, changing the setting after release from the reset state is not required. Do not set the PLLRESET bit to 1 after setting the PLLRESET bit to 0. Operation is not guaranteed.

CLKSEL[1:0] bits (Input System Clock Frequency)

The CLKSEL[1:0] bits select the transfer clock source for the USBHS.

For the transfer clock generated in the USB-PHY internal PLL, these bits set the input clock frequency. To input the clock source from the EXTAL pin, the USB 2.0 clock specification must be strictly followed.

Writing to the CKSEL[1:0] bits is invalid in CL-only mode because the internal PLL is stopped (see the description for HSEB bit (CL-only mode)). For the clock settings, see section 33.3.3, Supplying the Clock.

HSEB bit (CL-only mode)

The HSEB bit selects whether the USBHS operates in CL-only mode. High-speed transfer by the USBHS requires the use of internal high-speed analog circuits including the PLL, clock, and data recovery (CDR) circuit in the USB-PHY block.

CL-only mode limits the transfer to the USB 1.1 specification (full- and low-speed transfer only). Power consumption can be reduced by stopping the internal PLL of the PHY module and other high-speed analog circuits.

In CL-only mode, the USBHS requires supply clocks of 48 MHz and 60 MHz, generated in the Clock Generation Circuit. For the clock supply method, see section 9, Clock Generation Circuit.

33.2.18 Interrupt Status Register 0 (INTSTS0)

Address(es): USBHS.INTSTS0 4006 0040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag*1	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error.	R
b3	VALID	USB Request Reception Flag*1	0: Setup packet not received 1: Setup packet received.	R/(W) *3
b6 to b4	DVSQ[2:0]	Device State*1	Indicates the device state. b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspend state.	R
b7	VBSTS	VBUS Input Status Flag	0: USBHS_VBUS pin is low 1: USBHS_VBUS pin is high.	R
b8	BRDY	BRDY Interrupt Status Flag	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R
b9	NRDY	NRDY Interrupt Status Flag	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R
b10	BEMP	BEMP Interrupt Status Flag	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*2	0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred.	R/(W) *3
b12	DVST	Device State Transition Interrupt Status Flag*2	0: No device state transition interrupt occurred 1: Device state transition interrupt occurred.	R/(W) *3
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: No SOF interrupt occurred 1: SOF interrupt occurred.	R/(W) *3

Bit	Symbol	Bit name	Description	R/W
b14	RESM	Resume Interrupt Status Flag*2, *4	0: No resume interrupt occurred 1: Resume interrupt occurred.	R/(W) *3
b15	VBINT	VBUS Interrupt Status Flag*4	0: No VBUS interrupt occurred on detecting a change in the USBHS_VBUS pin 1: VBUS interrupt occurred on detecting a change in the USBHS_VBUS pin.	R/(W) *3

x: Don't care

Note 1. The CTSQ[2:0], VALID, and DVSQ[2:0] flags are only valid in device controller mode.

Note 2. The status of the CTRT, DVST, and RESM flags are changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.

Note 3. To clear the CTRT, DVST, SOFR, RESM, or VBINT flags, write 0 only to the flags to be cleared. Write 1 to the other flags. Do not write 0 to the status flags indicating 0.

Note 4. The USBHS detects a change in the status in the RESM or VBINT flag even while the clock supply is stopped (LPSTS.SUSPENDM = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software.

BRDY flag (BRDY Interrupt Status Flag)

The BRDY flag indicates the BRDY interrupt state. For the conditions that cause the flag to be set, see section 33.2.13, BRDY Interrupt Enable Register (BRDYENB).

The USBHS clears the BRDY flag to 0 when 0 is written to the BRDYSTS.PIPEBRDYn (n = 0 to 9) flags for all pipes for which the BRDY interrupt is enabled (BRDYENB.PIPEBRDYEn bits). Writing 0 to the BRDY flag in the software does not clear the flag.

NRDY flag (NRDY Interrupt Status Flag)

The NRDY flag indicates the NRDY interrupt state. For the conditions that cause the flag to be set, see section 33.2.14, NRDY Interrupt Enable Register (NRDYENB).

The USBHS clears the NRDY flag to 0 when 0 is written to the NRDYSTS.PIPENRDYn (n = 0 to 9) flags for all pipes for which the NRDY interrupt is enabled (NRDYENB.PIPENRDYEn bits). Writing 0 to the NRDY flag in the software does not clear the flag.

BEMP flag (BEMP Interrupt Status Flag)

The BEMP indicates the BEMP interrupt state. For the conditions that cause the flag to be set, see section 33.2.15, BEMP Interrupt Enable Register (BEMPENB).

The USBHS clears the BEMP flag to 0 when 0 is written to the BEMPSTS.PIPEBEMPn (n = 0 to 9) flags for all pipes for which the BEMP interrupt is enabled (BEMPENB.PIPEBEMPEn bits). Writing 0 to the BEMP flag in the software does not clear the flag.

CTRT flag (Control Transfer Stage Transition Interrupt Status Flag)

In device controller mode, the USBHS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT flag before the USBHS detects the next control transfer stage transition.

Values read from the CTRT flag in host controller mode are invalid.

DVST flag (Device State Transition Interrupt Status Flag)

In device controller mode, the USBHS updates the value of the PL1CTRL1.DVSQ[3:0] bits and sets the DVST flag to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear the DVST flag before the USBHS detects the next device state transition.

Values read from the DVST flag in host controller mode are invalid.

SOFR flag (Frame Number Refresh Interrupt Status Flag)

In host controller mode, the USBHS sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit is set to 1 by software. An SOFR interrupt is detected every 1 ms.

In device controller mode, the USBHS sets the SOFR flag to 1 on updating the frame number. An SOFR interrupt is detected every 1 ms. The USBHS can detect an SOFR interrupt through the SOF complementation function even when a corrupted SOF packet is received from the USB host. See section 33.3.13, SOF Complementation Function.

RESM flag (Resume Interrupt Status Flag)

In device controller mode, the USBHS sets the RESM flag to 1 on detecting the falling edge of the signal on the USBHS_DP pin in the Suspend state (PL1CTRL1.DVSQ[3:0] = 01xxb).

Values read from the RESM flag in host controller mode are invalid.

VBINT flag (VBUS Interrupt Status Flag)

The USBHS sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USBHS_VBUS pin input value. The USBHS sets the VBSTS flag to indicate the USBHS_VBUS pin input value. When a VBINT interrupt occurs, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

33.2.19 Interrupt Status Register 1 (INTSTS1)

Address(es): USBHS.INTSTS1 4006 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	L1RSMEND	LPMEND	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT
Value after reset:	0	0	x	0	0	x	0	0	x	0	0	0	x	x	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	PDDETINT	PDDET Detection Interrupt Status Flag*1	0: No PDDET interrupt occurred 1: PDDET interrupt occurred.	R/(W) *2
b3 to b1	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag	0: No SACK interrupt occurred 1: SACK interrupt occurred.	R/(W) *2
b5	SIGN	Setup Transaction Error Interrupt Status Flag	0: No SIGN interrupt occurred 1: SIGN interrupt occurred.	R/(W) *2
b6	EOFERR	EOF Error Detection Interrupt Status Flag	0: No EOFERR interrupt occurred 1: EOFERR interrupt occurred.	R/(W) *2
b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b8	LPMEND	LPM Transaction End Interrupt Status Flag	0: No LPMEND interrupt occurred 1: LPMEND interrupt occurred.	R/(W) *2
b9	L1RSMEND	L1 Resume End Interrupt Status Flag	0: No L1RSMEND interrupt occurred 1: L1RSMEND interrupt occurred.	R/(W) *2
b10	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b11	ATTCH	USB Connection Detection Interrupt Status Flag	0: No ATTCH interrupt occurred 1: ATTCH interrupt occurred.	R/(W) *2
b12	DTCH	USB Disconnection Detection Interrupt Status Flag	0: No DTCH interrupt occurred. 1: DTCH interrupt occurred	R/(W) *2
b13	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status Flag*1	0: No BCHG interrupt occurred 1: BCHG interrupt occurred.	R/(W) *2
b15	OVRCR	OVRCR Interrupt Status Flag*1	0: No OVRCR interrupt occurred 1: OVRCR interrupt occurred.	R/(W) *2

Note: Only enable the status change interrupts indicated in the flags in INTSTS1 in host controller mode, except for the PDDET detection interrupt.

Note 1. The USBHS detects a change in the status in the PDDETINT, BCHG, or OVRCR flag even while the clock supply is stopped (LPSTS.SUSPENDM = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software. No other interrupts can be detected while the clock supply is stopped (LPSTS.SUSPENDM = 0).

Note 2. To clear the flags in INTSTS1, write 0 only to the flags to be cleared. Write 1 to the other bits.

PDDETINT flag (PDDET Detection Interrupt Status Flag*[1])

The USBHS sets the PDDETINT flag to 1 on detecting a level change (high to low or low to high) in the PDDET pin input value. When the PDDETINT interrupt is generated, perform debouncing by reading the PDDETSTS flag at least three times through software processing and checking that the values read are the same.

SACK flag (Setup Transaction Normal Response Interrupt Status Flag)

The SACK flag indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBHS detects the SACK interrupt and sets this bit to 1 when an ACK response is returned from a peripheral device during the setup transactions issued by the USBHS. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

Values read from the SACK flag in device controller mode are invalid.

SIGN flag (Setup Transaction Error Interrupt Status Flag)

The SIGN flag indicates the status of setup transaction error interrupts in host controller mode.

The USBHS detects the SIGN interrupt and sets this bit to 1 when an ACK response is not returned from a peripheral device three consecutive times during the setup transactions issued by the USBHS. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBHS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received.

Values read from the SIGN flag in device controller mode are invalid.

EOFERR flag (EOF Error Detection Interrupt Status Flag)

The EOFERR flag indicates the status of EOF error detection interrupts in host controller mode.

The USBHS detects the EOFERR interrupt and sets this bit to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

After detecting the EOFERR interrupt, the USBHS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state.

The software must terminate all pipes in which communications are being carried out and re-enumerate the USB port.

Values read from the EOFERR flag in device controller mode are invalid.

LPMEND flag (LPM Transaction End Interrupt Status Flag)

The PLPMEND flag indicates the status of LPM transaction end interrupts in host controller mode.

When the HL1CTRL1.L1REQ bit sets to 1, the USBHS sends an LPM token. When the LPM transaction is ended because a response from the function device or a timeout is detected, the USBHS sets this flag to 1.

Values read from the LPMEND flag in device controller mode are invalid.

L1RSMEND flag (L1 Resume End Interrupt Status Flag)

The L1RSMEND flag indicates the status of L1 resume end interrupts in host controller mode.

When performing resume processing after transitioning to the L1 state because an ACK was received in response to an LPM token, the USBHS sets this flag to 1.

Values read from the L1RSMEND flag in device controller mode are invalid.

ATTCH flag (USB Connection Detection Interrupt Status Flag)

The ATTCH flag indicates the status of USB attach detection interrupts in host controller mode.

The USBHS detects the ATTCH interrupt and sets this bit to 1 on detecting a J- or K-state on the full- or low-speed signal level for 2.5 µs. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects the ATTCH interrupt on any of the following conditions:

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

Values read from the ATTCH flag in device controller mode are invalid.

DTCH flag (USB Disconnection Detection Interrupt Status Flag)

The DTCH flag indicates the status of USB detach detection interrupts in host controller mode.

The USBHS detects the DTCH interrupt and sets this bit to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBHS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state.

The software must terminate all pipes in which communications are being carried out and invoke the wait state for attaching to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

BCHG flag (USB Bus Change Interrupt Status Flag*1)

The BCHG flag indicates the status of USB bus change interrupts in host controller mode.

The USBHS detects the BCHG interrupt and sets this bit to 1 when a change in the full-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by software, the USBHS generates the interrupt.

The USBHS sets the SYSSTS0.LNST[1:0] flags to indicate the input state of the USB port. When a BCHG interrupt occurs, eliminate transient elements by repeat reading the LNST[1:0] bits by software until the same value is read at least three times.

Changes in the USB bus state can be detected while the PHY clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

OVRCR flag (OVRCR Interrupt Status Flag*1)

The OVRCR flag indicates the input status on the USBHS_OVCUR0A pin or changes on the USBHS_OVCUR0B pin. If the INTENB1.OVRCRE bit sets to 1, the USBHS requests the interrupt.

The USBHS sets the SYSSTS0.OVCMON[1:0] flags to indicate the input state of the USBHS_OVCUR0A and USBHS_OVCUR0B pins.

These pins allow overcurrent detection by software in host controller mode. To implement this function, connect the overcurrent signal from the external power supply IC that supplies VBUS to connected USB devices to the OVCUR0A or OVCUR0B pin. On detection of an OVRCR interrupt, eliminate transients by repeatedly reading the OVCMON[1:0] flags through the software until the same value is read at least three times.

33.2.20 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USBHS.BRDYSTS 4006 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	PIPEBRDY[9:0]	BRDY Interrupt Status Flag for Pipe[9:0]*1	0: No BRDY interrupt occurred 1: BRDY interrupt occurred.	R/(W)*2
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Each bit number corresponds to the same pipe number.

Note 2. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated in the PIPEBRDY[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.
When the SOFCFG.BRDYM bit is set to 0, clear BRDY interrupts before accessing the FIFO.

PIPEBRDY[9:0] flags (BRDY Interrupt Status Flag for Pipe[9:0])

When the BRDY interrupt is detected, the USBHS sets the associated bit in the PIPEBRDY[9:0] flags to 1. For details on BRDY interrupts, see section 33.3.6.1, BRDY interrupt.

33.2.21 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USBHS.NRDYSTS 4006 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPENRDY[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	PIPENRDY[9:0]	NRDY Interrupt Status Flag for Pipe[9:0]*1	0: No NRDY interrupt occurred 1: NRDY interrupt occurred.	R/(W)*2
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Each bit number corresponds to the same pipe number.

Note 2. To clear the status indicated in the PIPENRDY[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.

PIPENRDY[9:0] flags (NRDY Interrupt Status Flag for Pipe[9:0])

If an internal NRDY interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS sets the associated bit in the PIPENRDY[9:0] flags to 1. For details on NRDY interrupts, see section 33.3.6.2, NRDY interrupt.

33.2.22 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USBHS.BEMPSTS 4006 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPEBEMP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	PIPEBEMP[9:0]	BEMP Interrupt Status Flag for Pipe[9:0]*1	0: No BEMP interrupt occurred 1: BEMP interrupt occurred.	R/(W) *2
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Each bit number corresponds to the same pipe number.
Note 2. To clear the status indicated in the PIPEBEMP[9:0] flags, write 0 only to the bits to be cleared. Write 1 to the other bits.

PIPEBEMP[9:0] flags (BEMP Interrupt Status Flag for Pipe[9:0])

If an BEMP interrupt is detected while the PID[1:0] bits in a pipe control register are 01b (BUF response), the USBHS sets the associated bit in the PIPEBEMP[9:0] flags to 1. For details on BEMP interrupts, see section 33.3.6.3, BEMP interrupt.

33.2.23 Frame Number Register (FRMNUM)

Address(es): USBHS.FRMNUM 4006 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	x	x	x	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number Flag	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	CRC Error Detection Status Flag	0: No error occurred 1: Error occurred.	R/(W)
b15	OVRN	Overrun/Underrun Detection Status Flag	0: No error occurred 1: Error occurred.	R/(W)

Note: The OVRN flag is for debugging. Design the timing so that no overrun or underrun occurs in the system.

FRNM[10:0] flags (Frame Number Flag)

The USBHS sets the FRNM[10:0] flags to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

CRCE flag (CRC Error Detection Status Flag)

The CRCE flag sets to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error, the USBHS generates an internal NRDY interrupt.

To clear the CRCE flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

OVRN flag (Overrun/Underrun Detection Status Flag)

The OVRN flag sets to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty.

In device controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty.

33.2.24 µFrame Number Register (UFRMNUM)

Address(es): USBHS.UFRMNUM 4006 004Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	UFRNM[2:0]	Microframe Number	Microframe number	R
b14 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DVCHG	Device State Change	0: Disable writes to the USBADDR.STSRECOV0[2:0] and USBADDR.USBADDR[6:0] bits 1: Enable writes to the USBADDR.STSRECOV0[2:0] and USBADDR.USBADDR[6:0] bits.	R/W

UFRNM[2:0] flags (Microframe Number)

The USBHS sets the UFRNM[2:0] flags to indicate the microframe number during high-speed operation. When not in high-speed operation, the USBHS sets these bits to 00h.

Read these bits repeatedly until the same value is read twice.

33.2.25 USB Address Register (USBADDR)

Address(es): USBHS.USBADDR 4006 0050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	STSRECOV0[2:0]			—	USBADDR[6:0]						
Value after reset:	x	x	x	x	x	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address Flag	In device controller mode, these flags indicate the USB address assigned by the host when the USBHS processed the SET_ADDRESS request successfully.	R
b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b10 to b8	STSRECOV0[2:0]	Status Recovery	• Recovery in device controller mode b10 b8 0 0 1:Return to the full-speed connection and Default state 0 1 0:Return to the full-speed connection and Address state 0 1 1:Return to the full-speed connection and Configured state 1 0 0:Return to the suspend connection and Suspend state 1 0 1:Return to the high-speed connection and Default state 1 1 0:Return to the high-speed connection and Address state 1 1 1:Return to the high-speed connection and Configured state. Other settings are prohibited. • Recovery in host controller mode b10 b8 0 1 0:Return to the low-speed state (bits DVSTCTR0.RHST[2:0] = 001b) 1 0 0:Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) 1 1 0:Return to the high-speed state (bits DVSTCTR0.RHST[2:0] = 011b). Other settings are prohibited.	R/W
b15 to b11	—	Reserved	The read value is undefined. The write value should be 0.	R/W

USBADDR[6:0] flags (USB Address Flag)

In device controller mode, the USBADDR[6:0] flags indicate the USB address received when the USBHS processed a SetAddress request successfully. The USBHS sets the USBADDR[6:0] bits to 00h on detecting a USB bus reset.

In host controller mode, the USBADDR[6:0] bits are invalid.

STSRECOV0[2:0] bits (Status Recovery)

Use the STSRECOV[3:0] bits to resume the state of the internal sequencer on recovering from USB power shut-off. For details, see section 33.3.17, Release from Deep Software Standby Mode Because of USB Suspend/Resume Interrupts.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1.

33.2.26 USB Request Type Register (USBREQ)

Address(es): USBHS.USBREQ 4006 0054h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	USB request bmRequestType value	R/W*1
b15 to b8	BREQUEST[7:0]	Request	USB request bRequest value	R/W*1

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

BMREQUESTTYPE[7:0] bits (Request Type)

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

BREQUEST[7:0] bits (Request)

The BREQUEST[7:0] bits hold the bRequest value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

33.2.27 USB Request Value Register (USBVAL)

Address(es): USBHS.USBVAL 4006 0056h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	USB request wValue value	R/W*1

Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WVALUE[15:0] bits (Value)

The WVALUE[15:0] bits hold the wValue value of USB requests.

- In host controller mode:
  Set these bits to the wValue value for USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

33.2.28 USB Request Index Register (USBINDX)

Address(es): USBHS.USBINDX 4006 0058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	USB request wIndex value	R/W*1

Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WINDEX[15:0] bits (Index)

The WINDEX[15:0] bits hold the wIndex value of USB requests.

- In host controller mode:
  Set these bits to the wIndex value of USB requests in transmission setup transactions. Do not change the value of

the bits while the DCPCTR.SUREQ bit is 1.

- In device controller mode:
  These bits indicate the wIndex value of USB requests received in reception setup transactions. Writing to the bits has no effect.

33.2.29 USB Request Length Register (USBLENG)

Address(es): USBHS.USBLENG 4006 005Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WLENTUH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WLENTUH[15:0]	Length	USB request wLength value	R/W*1

Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

WLENTUH[15:0] bits (Length)

The WLENTUH[15:0] bits hold the wLength value of USB requests.

- In host controller mode:
  Set the wLength value of USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wLength value of USB requests in reception setup transactions. Writing to the bits has no effect.

33.2.30 DCP Configuration Register (DCPCFG)

Address(es): USBHS.DCPCFG 4006 005Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	DIR	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	0	x	x	0	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction	0: Data receiving direction 1: Data transmitting direction.	R/W
b6, b5	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b7	SHTNAK	Pipe Blocking on End of Transfer	0: Keep pipe open after transfer ends 1: Disable pipe after transfer ends.	R/W
b8	CNTMD	Continuous Transfer Mode	0: Non-continuous transfer mode 1: Continuous transfer mode.	R/W
b15 to b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. Only set the bits in this register while the PID is NAK. Before setting the bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBHS, checking the PBUSY bit through software is not necessary.

DIR bit (Transfer Direction)

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

SHTNAK bit (Pipe Blocking on End of Transfer)

The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBHS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that a transfer has ended. The USBHS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received.

CNTMD bit (Continuous Transfer Mode)

The CNTMD bit indicates whether transfer through the default control pipe is in continuous transfer mode.

33.2.31 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USBHS.DCPMAXP 4006 005Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	x	x	x	x	x	1	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	Maximum data payload specification (maximum packet size) for the DCP	R/W
b11 to b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b 0 0 1 0: Address 0010b 0 0 1 1: Address 0011b 0 1 0 0: Address 0100b 0 1 0 1: Address 0101b 0 1 1 0: Address 0110b 0 1 1 1: Address 0111b 1 0 0 0: Address 1000b 1 0 0 1: Address 1001b 1 0 1 0: Address 1010b Other settings are prohibited.	R/W

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting this bit, check that the CSSTS and PBUSY bits are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK), and the CFIFOSEL.CURPIPE[3:0] bits to 0000b. If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bits are 0. Before setting these bits, check that the CSSTS and PBUSY flags are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK), and the DCPCTR.SUREQ[3:0] bits to 0. If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through software is not necessary.

MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 40h (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF while MXPS[6:0] is set to 0.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target peripheral device for a control transfer. Set up the device address in the associated DEVADDm (m = 0 to A) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

33.2.32 DCP Control Register (DCPCTR)

Address(es): USBHS.DCPCTR 4006 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	CSCLR	CSSTS	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	PINGE	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	x	x	x	0	0	1	0	0	x	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends on buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b2	CCPL	Control Transfer End Enable	0: Disable control transfer completion 1: Enable control transfer completion.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b4	PINGE	PING Token Issue Enable*1	0: Disable PING token 1: Enable normal PING operation.	R/W
b5	PBUSY	Pipe Busy Flag	0: DCP not used for the USB bus 1: DCP in use for the USB bus.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*1	Sets the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W
b8	SQCLR	Sequence Toggle Bit Clear*1	Clears the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W
b10, b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	Clears the SUREQ bit in host controller mode. 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0. This bit is read as 0.	R/W
b12	CSSTS	CSSTS Status Flag	0: Start-split (SSPLIT) transaction, or processing for devices that are not using split transactions, in progress. 1: Complete-split (CSPLIT) transaction in progress.	R
b13	CSCLR	CSSTS Status Flag Clear	Clears the CSSTS flag in host controller mode for split transactions, resuming the next DCP transfer from SSPLIT. 0: Invalid (writing 0 has no effect) 1: Clear CSSTS to 0. This bit is read as 0.	R/W
b14	SUREQ	SETUP Token Transmission	Sets up token transmission in host controller mode. 0: Invalid (writing 0 has no effect) 1: Transmit setup packet.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access disabled 1: Buffer access enabled.	R

Note 1. Only set the SQSET, SQCLR, and PINGE bits while PID is NAK. Before setting these bits, check that the CSSTS and PBUSY bits are 0, and then change the DCPCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the DCPCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY bits through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:
  a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
  The USBHS then executes the OUT transaction (or PING transaction).
- When the receiving direction is set:
  c. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  d. Set PID[1:0] bits to 01b (BUF).
  The USBHS then executes the IN transaction.

The USBHS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by software and the USBHS has received data exceeding MaxPacketSize, the USBHS sets PID[1:0] to STALL (11b)
- When a reception error, such as a CRC error, is detected three times consecutively, the USBHS sets PID[1:0] to NAK (00b)
- On receiving the STALL handshake, the USBHS sets PID[1:0] to STALL (11b).

In device controller mode, the USBHS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBHS sets PID[1:0] to NAK (00b). The USBHS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by software and the USBHS has received data exceeding MaxPacketSize, the USBHS sets PID[1:0] to STALL (11b)
- On detecting a control transfer sequence error, the USBHS sets PID[1:0] to STALL (1xb)
- On detecting a USB bus reset, the USBHS sets PID[1:0] to NAK.

The USBHS does not check the PID[1:0] setting while processing a SET_ADDRESS request.

CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by software while the associated PID[1:0] bits are set to BUF, the USBHS completes the control transfer status stage.

During control read transfers, the USBHS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET_ADDRESS request, the USBHS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBHS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

PINGE bit (PING Token Issue Enable)

In host controller mode, when the software sets the PINGE bit to 1, the USBHS issues a PING token for transfer in the transmitting direction, which triggers the transfer to start. If an ACK handshake is detected in the PING transaction, the OUT transaction is executed in the next transaction. If a NAK or NYET handshake is detected in the OUT transaction, the PING transaction is executed in the next transaction.

If the software sets this bit to 0, the USBHS issues no PING token for transfer in the transmitting direction. All transfers in the transmitting direction are executed in the OUT transaction.

PBUSY flag (Pipe Busy Flag)

The PBUSY bit indicates whether DCP is used for the transaction when USBHS changes the PID[1:0] bits from BUF to NAK. The USBHS changes the PBUSY flag from 0 to 1 on start of a USB transaction for the selected pipe. It changes the PBUSY flag from 1 to 0 on completion of one transaction.

After PID is set to NAK by software, the value in the PBUSY flag indicates whether changes to pipe settings can proceed.

For details, see section 33.3.7.1, Pipe control register switching procedures.

SQMON flag (Sequence Toggle Bit Monitor Flag)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBHS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATA-PID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBHS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBHS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR bit (SUREQ Bit Clear)

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 through software. This is not necessary at the end of a normal setup transaction, because the USBHS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to the SUREQCLR bit.

CSSTS flag (CSSTS Status Flag)

In host controller mode, the CSSTS flag indicates the complete-split state in split transactions for pipes that are not isochronous. The USBHS sets the CSSTS flag to 1 at the beginning of a complete-split transaction and sets the flag back to 0 when it detects transaction completion.

Values read from the CSSTS flag in device controller mode are invalid.

CSCLR bit (CSSTS Status Flag Clear)

In host controller mode, setting the CSCLR bit to 1 clears the CSSTS bit to 0.

Set this bit to 1 through software when forcing the next transfer to restart from start-split in transfers using split transactions. This is not necessary at the end of a successful complete-split transaction in a normal split transaction, because the USBHS automatically clears the CSSTS flag to 0.

Only control the CSSTS flag through the CSCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a port disconnection was detected. Writing 1 to this bit while the CSSTS flag is 0 has no effect; the flag remains 0.

In device controller mode, always write 0 to this bit.

SUREQ bit (SETUP Token Transmission)

In host controller mode, setting the SUREQ bit to 1 triggers the USBHS to transmit the setup packet. After completing the setup transaction process, the USBHS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBHS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the wanted USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

BSTS flag (Buffer Status Flag)

The BSTS flag indicates the status of access to the DCP FIFO buffer. The meaning of this flag varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer.

33.2.33 Pipe Window Select Register (PIPESEL)

Address(es): USBHS.PIPESEL 4006 0064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Set pipes 1 to 9 using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN registers (n = 0 to 9).

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI registers. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits has no effect.

33.2.34 Pipe Configuration Register (PIPECFG)

Address(es): USBHS.PIPECFG 4006 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	x	x	x	0	0	0	0	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	Specifies the endpoint number for the selected pipe. Setting 0000b indicates the pipe is not used.	R/W
b4	DIR	Transfer Direction*2, *3	0: Receiving direction 1: Transmitting direction.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Continue pipe operation after transfer ends 1: Disable pipe after transfer ends.	R/W
b8	CNTMD	Continuous Transfer Mode*2, *3	0: Discontinuous transfer mode 1: Continuous transfer mode.	R/W
b9	DBLB	Double Buffer Mode*2, *3	0: Single buffer 1: Double buffer.	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2, *3	0: Generate BRDY interrupt on transmitting or receiving data 1: Generate BRDY interrupt on completion of reading data.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer. • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited. • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited.	R/W

Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

Note 2. Only set the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 3. To change the BFRE, DBLB, or DIR bit after completing USB communication on the selected pipe, in addition to the constraints described in note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously through software and clear the FIFO buffer assigned to the pipe.

EPNUM[3:0] bit (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. (The EPNUM[3:0] bits can be set to 0000b for all pipes.)

DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBHS uses the selected pipe for receiving. When the software sets this bit to 1, the USBHS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBHS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBHS determines that the transfer has ended on the following conditions:

- Short packet data (including a zero-length packet) was successfully received
- The transaction counter is used and the number of packets specified for the transaction counter were successfully received.

CNTMD bit (Continuous Transfer Mode)

The CNTMD bit specifies whether to operate the selected pipe in continuous transfer mode. The bit is valid for pipes 1 to 5 of the bulk transfer type.

Based on this bit setting, the USBHS determines the completion of transmission or reception for the FIFO buffer allocated to the selected pipe as shown in Table 33.9.

Table 33.9 Relationship between the CNTMD setting and methods for determining completion of FIFO buffer transmission or reception

CNTMD bit setting	Methods for determining readable state and transmittable state
0	Condition for FIFO buffer readable state in receiving direction (DIR = 0): • The USBHS received one packet.
	Conditions for FIFO buffer transmittable state in transmitting direction (DIR = 1): When one of (1) or (2) of the following is satisfied: (1) Software (or DMAC/DTC) wrote data of the maximum packet size to the FIFO buffer. (2) Software (or DMAC/DTC) wrote data of the short packet size (including 0 bytes) to the FIFO buffer and set the BVAL flag in the port control register to 1.
1	Condition for FIFO buffer readable state in receiving direction (DIR = 0): (1) The byte count of data received in the FIFO buffer allocated to the selected pipe is equal to the allocated byte count ((BUFSIZE + 1) × 64). (2) The USBHS received a short packet, other than a zero-length packet. (3) The USBHS received a zero-length packet when data was already contained in the FIFO buffer allocated to the selected pipe. (4) Software received the number of packets specified for the transaction counter set for the selected pipe.
	Conditions for FIFO buffer transmittable state in transmitting direction (DIR = 1): When one of (1) to (3) of the following is satisfied. (1) The amount of data written by software (or DMAC/DTC) is equal to the size of the FIFO buffer allocated to the selected pipe. (2) The software (or DMAC/DTC) wrote data of smaller size than that of the FIFO buffer allocated to the selected pipe (including 0 bytes) and set the BVAL flag in the port control register to 1. (3) The software (or DMAC/DTC) wrote data of smaller size than that of one FIFO buffer allocated to the selected pipe (including 0 bytes) and asserted the DENDx_N signal on the last write.

DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

When the software sets this bit to 1, the USBHS allocates twice the FIFO buffer size specified in the PIPEBUF.BUFSIZE[5:0] bits for the selected pipe. The FIFO buffer size that the USBHS allocates to the selected pipe is as follows:

$$(\text{BUFSIZE} + 1) \times 64 \times (\text{DBLB} + 1)\ [\text{bytes}]$$

BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBHS to the CPU for the selected pipe.

When the software sets the BFRE bit to 1 and the selected pipe is in the receiving direction, the USBHS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by software and the selected pipe is in the transmitting direction, the USBHS does not generate the BRDY interrupt. For details, see section 33.3.6.1, BRDY interrupt.

TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

33.2.35 Pipe Buffer Register (PIPEBUF)

Address(es): USBHS.PIPEBUF 4006 006Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	BUFSIZE[4:0]					—	—	BUFNMB[7:0]							
Value after reset:	x	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	BUFNMB[7:0]	Buffer Number	Specifies the FIFO buffer number of the selected pipe (04h to 87h).	R/W
b9, b8	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b14 to b10	BUFSIZE[4:0]	Buffer Size	00h: 64 bytes 01h: 128 bytes ... 1Fh: 2 KB.	R/W
b15	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. Only set the bits in the PIPEBUF register while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

BUFNMB[7:0] bits (Buffer Number)

The BUFNMB[7:0] bits specify the first block number of the FIFO buffer to be allocated to the selected pipe.

The USBHS allocates the FIFO buffer blocks to the selected pipe as follows:

$$\text{Block number: BUFNMB to block number: BUFNMB} + (\text{BUFSIZE} + 1) \times (\text{DBLB} + 1) - 1$$

Set a value within the memory size range for these bits (0 [00h] to 8640 [87h] for 8.5 KB), while observing the following conditions:

- 00h is for DCP only
- 04h is for pipe 6 only, but is available for other pipes when pipe 6 is not used. When pipe 6 is selected, writes to these bits are disabled. The USBHS automatically allocates 04h to the BUFNMB bits for pipe 6.
- 05h is for pipe 7 only, but is available for other pipes when pipe 7 is not used. When pipe 7 is selected, writes to these bits are disabled. The USBHS automatically allocates 05h to the BUFNMB bits for pipe 7.
- 06h is for pipe 8 only, but is available for other pipes when pipe 8 is not used. When pipe 8 is selected, writes to these bits are disabled. The USBHS automatically allocates 06h to the BUFNMB bits for pipe 8.
- 07h is for pipe 9 only, but is available for other pipes when pipe 9 is not used. When pipe 9 is selected, writes to these bits are disabled. The USBHS automatically allocates 07h to the BUFNMB bits for pipe 9.

BUFSIZE[4:0] bits (Buffer Size)

The BUFSIZE[4:0] bits specify the FIFO buffer size (number of blocks) to be allocated to the selected pipe. One block is 64 bytes.

When the software sets the DBLB bit to 1, the USBHS allocates twice the FIFO buffer size specified in these bits to the selected pipe. The DBLB = 1 setting is valid for pipes 1 to 5.

The USBHS allocates the FIFO buffer blocks to the selected pipe as follows:

$$(\text{BUFSIZE} + 1) \times 64 \times (\text{DBLB} + 1)\ [\text{bytes}]$$

Set the value within the following range:

- For pipes 1 to 5, set a value from 00h to 1Fh (up to 2 KB)
- For pipes 6 to 9, only set a value of 00h (64 bytes).

33.2.36 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USBHS.PIPEMAXP 4006 006Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	MXPS[10:0]										
Value after reset:	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b10 to b0	MXPS[10:0] *1, *2	Maximum Packet Size	• Pipes 1 and 2 1 byte (001h) to 1024 bytes (400h) • Pipes 3 to 5 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h), 512 bytes (200h) (Bits 2 to 0 not supported.) • Pipes 6 to 9 1 byte (001h) to 64 bytes (040h) (Bits 10 to 7 not supported.)	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0] *3	Device Select	b15 b12 0 0 0 0: Address 0000b 0 0 0 1: Address 0001b ... 1 0 0 1: Address 1001b 1 0 1 0: Address 1010b 1011 to 1111: Reserved.	R/W

Note 1. The initial value of the MXPS[10:0] bits is 00h when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 40h when a pipe is selected.

Note 2. Only set the MXPS[10:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the

PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the CSSTS and PBUSY flags through the software is not necessary.

Note 3. Only set the DEVSEL[3:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

MXPS[10:0] bits (Maximum Packet Size)

The MXPS[10:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[10:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

To communicate on an isochronous pipe using a split transaction, set the value in the MXPS[10:0] bits to 188 bytes or less.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the device address in the associated DEVADDm (m = 0 to A) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

33.2.37 Pipe Cycle Control Register (PIPEPERI)

Address(es): USBHS.PIPEPERI 4006 006Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	x	x	x	0	x	x	x	x	x	x	x	x	x	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval	Specifies the interval error detection timing for the selected pipe as the n-th power of 2 of the frame timing.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: Do not flush buffer 1: Flush buffer.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe specified in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBHS automatically clears the FIFO buffer if the USBHS fails to receive the IN token from the USB host within the interval set in the IITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBHS only clears the data in the previously used plane.

The USBHS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBHS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal complementation function.

In host controller mode, set the IITV[2:0] bits to 000b.

Set the IITV[2:0] bits to 000b when the selected pipe is not used for isochronous transfers.

33.2.38 Pipe n Control Register (PIPEnCTR) (n = 1 to 9)

Address(es): USBHS.PIPE1CTR 4006 0070h, USBHS.PIPE2CTR 4006 0072h, USBHS.PIPE3CTR 4006 0074h, USBHS.PIPE4CTR 4006 0076h, USBHS.PIPE5CTR 4006 0078h, USBHS.PIPE6CTR 4006 007Ah, USBHS.PIPE7CTR 4006 007Ch, USBHS.PIPE8CTR 4006 007Eh, USBHS.PIPE9CTR 4006 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	CSCLR	CSSTS	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	x	0	0	0	0	0	0	x	x	x	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends on buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b4 to b2	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*1	Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W
b8	SQCLR	Sequence Toggle Bit Clear*1	Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W
b9	ACLRM	Auto Buffer Clear Mode*2	0: Disable 1: Enable (initialize all buffers).	R/W
b10	ATREPM	Auto Response Mode*1, *3	0: Disable auto response mode 1: Enable auto response mode.	R/W
b11	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b12	CSSTS	CSSTS Status Flag	0: Start-split (SSPLIT) transaction, or processing for devices that are not using split transactions, in progress. 1: Complete-split (CSPLIT) transaction in progress.	R
b13	CSCLR	CSPLIT Status Clear	Clears the CSSTS flag for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear CSSTS to 0.	W
b14	INBUFM	Transmit Buffer Monitor Flag*3	0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer.	R
b15	BSTS	Buffer Status Flag	0: Buffer access disabled 1: Buffer access enabled.	R

Note 1. Only set the ATREPM bit while PID is NAK. Before setting this bit, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 2. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

Note 3. The ATREPM bit and the INBUFM flag in the PIPE6CTR to PIPE9CTR registers are reserved. The read value is undefined. The write value must be 0.

PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction on the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 33.10 and Table 33.11 show the basic operations of the USBHS (when there are no errors in the communication packets) based on the PID[1:0] bit setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBHS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBHS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBHS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by software
- The USBHS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBHS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBHS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBHS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode.

To specify the response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF).

Table 33.10 Operation of the USBHS based on the PIPEnCTR.PID[1:0] setting in host controller mode

PID[1:0] value	Transfer type (TYPE[1:0] value)	Transfer direction (DIR value)	USBHS operation
00b (NAK)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens
01b (BUF)	Bulk or Interrupt	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting.	Issues tokens when the DVSTCTR0.UACT bit is 1, regardless of the state of the FIFO buffer associated with the selected pipe. Does not issue tokens when UACT = 0.
10b (STALL) or 11b (STALL)	Does not depend on the setting.	Does not depend on the setting.	Does not issue tokens.

Table 33.11 Operation of the USBHS based on the PIPEnCTR.PID[1:0] setting in device controller mode

PID[1:0] value	Transfer type (TYPE[1:0] value)	Transfer direction (DIR value)	USBHS operation
00b (NAK)	Bulk or Interrupt	Does not depend on the setting	Returns NAK in response to the token from the USB host
	Isochronous	Receiving direction (DIR = 0)	Returns nothing in response to the token from the USB host
		Transmitting direction (DIR = 1)	Transmits a zero-length packet in response to the token from the USB host
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK or NYET in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK. Returns ACK in response to the PING token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK response in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, returns NAK.
	Bulk or Interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.
		Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or Interrupt	Does not depend on the setting.	Returns STALL in response to the token from the USB host
	Isochronous	Does not depend on the setting.	Returns nothing in response to the token from the USB host

PBUSY flag (Pipe Busy Flag)

The PBUSY flag indicates whether the selected pipe is being used for the current transaction.

The USBHS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see section 33.3.7.1, Pipe control register switching procedures.

SQMON flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBHS toggles the SQMON flag on successful completion of the transaction. However, the USBHS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBHS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBHS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0.

In host controller mode, when this bit is set to 1 for a bulk OUT transfer pipe, the USBHS starts the next transfer for the selected pipe from a PING token.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 33.12 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 33.12 Data cleared by the USBHS when ACLRM = 1

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When clearing all data in the FIFO buffer allocated to the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBHS responds to the token from the USB host as follows:

- When the selected pipe is set for bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
  a. When the ATREPM bit = 1 and PID = BUF, the USBHS transmits a zero-length packet in response to the IN token.
  b. The USBHS updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USBHS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBHS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is set for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):

  When the ATREPM bit = 1 and PID = BUF, the USBHS returns NAK in response to the OUT token or PING token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

CSSTS flag (CSSTS Status Flag)

In host controller mode, the CSSTS flag indicates the complete-split status of a split transaction. It is valid for pipes that are not the isochronous transfer type.

The USBHS sets the CSSTS flag to 1 at the beginning of the complete-split transaction, and sets the CSSTS flag to 0 on detecting completion of the complete-split transaction. If a detach event is detected during the transaction, the CSSTS flag might stay set to 1. In this case, clear the CSSTS flag by setting the CSCLR bit to 1.

Values read from the CSSTS flag in device controller mode are invalid.

CSCLR bit (CSPLIT Status Clear)

In host controller mode, if the software sets the CSCLR bit to 1, the USBHS clears the CSSTS flag to 0. In split transactions, set the CSCLR bit to 1 by software to force the next transfer to restart from start-split. Because the USBHS automatically clears the CSSTS flag to 0 at the end of a successful complete-split transaction in a normal split transaction, clearing the flag through software is not required. Only clear the CSSTS flag using the CSCLR bit when the DVSTCTR0.UACT bit is set to 0 or when no transfer was made after a detach detect. If the CSCLR bit is set to 1 while the CSSTS flag is 0, the CSSTS flag remains 0.

In device controller mode, always write 0 to the CSCLR bit.

INBUFM flag (Transmit Buffer Monitor Flag)

The INBUMFM flag indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBHS sets this bit to 1 when the CPU or DMA/DTC completes writing data to at least one FIFO buffer plane.

The USBHS sets this bit to 0 when the USBHS completes transmission of data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBHS sets the INBUFM flag to 0 when the USBHS completes transmission of data from the two FIFO buffer planes before the CPU or DMA/DTC completes writing data to one FIFO buffer plane.

The INBUFM flag indicates the same value as the BSTS flag when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

BSTS flag (Buffer Status Flag)

The BSTS flag indicates the FIFO buffer status for the selected pipe. The meaning of the BSTS flag depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 33.13.

Table 33.13 BSTS flag operation

DIR value	BFRE value	DCLRM value	Meaning of BSTS flag
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete
		1	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and clears to 0 on completion of data write
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

33.2.39 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USBHS.PIPE1TRE 4006 0090h, USBHS.PIPE2TRE 4006 0094h, USBHS.PIPE3TRE 4006 0098h, USBHS.PIPE4TRE 4006 009Ch, USBHS.PIPE5TRE 4006 00A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid (writing 0 has no effect) 1: Clear current counter value.	R/W
b9	TRENB	Transaction Counter Enable	0: Disable transaction counter 1: Enable transaction counter.	R/W
b15 to b10	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note: Only change the PIPEnTRE settings while the PIPEnCTR.CSSTS flag is 0 and the PIPEnCTR.PID[1:0] bits are 00b (NAK response). Only change the PIPEnCTR.PID[1:0] bits of the selected pipe from 01b (BUF response) to 00b (NAK response) after confirming that the value of the PIPEnCTR.PBUSY and PIPEnCTR.CSSTS flags is 0. However, software processing to check the PIPEnCTR.PBUSY flag is not required if the USBHS has changed the PID[1:0] bits to 00b (NAK response).

TRCLR bit (Transaction Counter Clear)

When the TRCLR bit sets to 1, the USBHS clears the count value of the transaction counter associated with the selected pipe and then clears the TRCLR bit to 0.

TRENB bit (Transaction Counter Enable)

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBHS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting as follows:

- When the PIPECFG.SHTNAK bit is 1, the USBHS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting
- When the PIPECFG.BFRE bit is 1, the USBHS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data.

For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

33.2.40 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USBHS.PIPE1TRN 4006 0092h, USBHS.PIPE2TRN 4006 0096h, USBHS.PIPE3TRN 4006 009Ah, USBHS.PIPE4TRN 4006 009Eh, USBHS.PIPE5TRN 4006 00A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter*1	• When written to: Specifies the total packets (number of transactions) to be received by pipe n. • When read from: When PIPEnTRE.TRENB is 0, indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, indicates the current transaction count.	R/W

Note 1. Only set the TRNCNT[15:0] bits while PID is NAK and PIPEnTRE.TRENB is 0. Before setting these bits, check that the PIPEnCTR.CSSTS and PIPEnCTR.PBUSY flags are 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PIPEnCTR.PID[1:0] bits are changed to 00b (NAK) by the USBHS, checking the PBUSY flag through the software is not necessary.

The PIPEnTRN registers retain their settings during a USB bus reset.

TRNCNT[15:0] bits (Transaction Counter)

The USBHS increments the value of the TRNCNT[15:0] bits by one when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit is 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting.

The USBHS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions is satisfied.

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting.

Both the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBHS received a short packet.

Both the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit was set to 1 by software.

For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is enabled only when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

33.2.41 Device Address m Configuration Register (DEVADDm) (m = 0 to A)

Address(es): USBHS.DEVADD0 4006 00D0h, USBHS.DEVADD1 4006 00D2h, USBHS.DEVADD2 4006 00D4h, USBHS.DEVADD3 4006 00D6h, USBHS.DEVADD4 4006 00D8h, USBHS.DEVADD5 4006 00DAh, USBHS.DEVADD6 4006 00DCh, USBHS.DEVADD7 4006 00DEh, USBHS.DEVADD8 4006 00E0h, USBHS.DEVADD9 4006 00E2h, USBHS.DEVADDA 4006 00E4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	UPPHUB[3:0]				HUBPORT[2:0]			USBSPD[1:0]		—	—	—	—	—	—
Value after reset:	x	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: Do not use DEVADDm 0 1: Low speed 1 0: Full speed 1 1: High speed.	R/W
b10 to b8	HUBPORT[2:0]	Communication Target Connecting Hub Port	b10 b8 0 0 0: Connect directly to the USBHS port 001 to 111: Port number of the hub.	R/W
b14 to b11	UPPHUB[3:0]	Communication Target Connecting Hub Register	b14 b11 0 0 0 0: Connect directly to the USBHS port 0001 to 1010: USB address of the hub 1011 to 1111: Reserved.	R/W
b15	—	Reserved	The read value is undefined. The write value should be 0.	R/W

The DEVADDm register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9.

In host controller mode, set all DEVADDm bits before starting communication to any pipes. Only change the bits in DEVADDm when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- DEVADDm is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

In device controller mode, set all bits in this register to 0.

USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. In host controller mode, the USBHS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

HUBPORT[2:0] bits (Communication Target Connecting Hub Port)

In host controller mode, the USBHS generates packets based on the HUBPORT[2:0] setting when performing a split transaction.

UPPHUB[3:0] bits (Communication Target Connecting Hub Register)

In host controller mode, the USBHS generates packets based on the UPPHUB[3:0] setting when performing a split transaction.

33.2.42 Low Power Control Register (LPCTRL)

Address(es): USBHS.LPCTRL 4006 0100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	HWUPM	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	0	0	x	x	x	x	x	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b7	HWUPM	Resume Return Mode Setting	0: Hardware does not recover while CPU clock inactive 1: Hardware recovers while CPU clock inactive.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15 to b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W

HWUPM bit (Resume Return Mode Setting)

The HWUPM bit specifies whether to enable hardware processing for return from low power mode even while the CPU clock is inactive.

In device controller mode, processing for return from low power mode on detecting Resume is enabled even while the CPU clock is inactive.

This bit specifies whether to detect Resume while the CPU clock is inactive. The PL1CTRL1.L1EXTMD bit controls whether to make a hardware return. To make a hardware return from the LPM L1 low power state while the CPU clock is inactive, set this bit and the PL1CTRL1.L1EXTMD bit to 1.

33.2.43 Low Power Status Register (LPSTS)

Address(es): USBHS.LPSTS 4006 0102h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	SUSPENDM	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	0	x	0	x	x	x	0	x	x	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b2	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11 to b9	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b14	SUSPENDM	UTMI SuspendM Control	0: UTMI suspension mode 1: UTMI normal mode.	R/W
b15	—	Reserved	The read value is undefined. The write value should be 0.	R/W

SUSPENDM bit (UTMI SuspendM Control)

The SUSPENDM bit controls the SuspendM signal to be sent to the PHY designed under the UTMI specification. The initial value is 0 with the UTMI is in suspension mode.

Set this bit to 1 to supply the PHY clock to operate the USB2.0 host or device controller.

In compliance with the UTMI specification, clock output is normally controlled by the SuspendM signal. When the SUSPENDM bit is 0, the clock to LINK is stopped. Because the PHY in this MCU follows the UTMI specification, setting the SUSPENDM bit to 1 is required to supply the PHY clock. For the clock settings, see section 33.3.3, Supplying the Clock.

When the SUSPENDM bit is 0, the USBHS cannot be written to but can be read from. The registers listed in Table 33.14 are writable even when the SUSPENDM bit is 0.

Table 33.14 Registers that can be written to by software when SUSPENDM = 0

Address	Register or bit name
4006 0000h	SYSCFG register
4006 0002h	BUSWAIT register
4006 0032h	INTENB1.PDDETINTE bit
4006 0100h	LPCTRL register
4006 0102h	LPSTS register
4006 0140h	BCCTRL register

The value written to the SYSCFG register while the PHY clock is inactive is updated only after the PHY clock begins oscillating. The PHY clock oscillates in the following cases described in this section.

When SUSPENDM bit is set to 1, the PLLSTA.PLLLOCK flag is set to 1 after the predetermined time has passed. The USB-PHY internal PLL is stopped when the SUSPENDM bit is set to 0.

For details on CL-only mode, see section 33.2.17, PHY Setting Register (PHYSET).

If the PL1CTRL1.L1EXTMD bit is 0, setting or clearing of this bit is controlled by software. If the PL1CTRL1.L1EXTMD bit is 1, transitions to the L1 or L2 state of this bit are controlled by software and recovery from the L1 or L2 state is controlled by hardware.

33.2.44 Battery Charging Control Register (BCCTRL)

Address(es): USBHS.BCCTRL 4006 0140h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PDDETSTS	CHGDETSTS	—	—	DCPMODE	VDMSRCE	IDPSINKE	VDPSRCE	IDMSINKE	IDPSRCE
Value after reset:	x	x	x	x	x	x	0	0	x	x	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IDPSRCE	IDPSRC Control*2	0: Disable IDP_SRC circuit 1: Enable IDP_SRC circuit.	R/W
b1	IDMSINKE	IDMSINK Control*2	0: Disable IDM_SINK circuit 1: Enable IDM_SINK circuit.	R/W
b2	VDPSRCE	VDPSRC Control*2	0: Disable VDP_SRC circuit 1: Enable VDP_SRC circuit.	R/W
b3	IDPSINKE	IDPSINK Control*2	0: Disable IDP_SINK circuit 1: Enable IDP_SINK circuit.	R/W
b4	VDMSRCE	VDMSRC Control*2	0: Disable VDM_SRC circuit 1: Enable VDM_SRC circuit.	R/W
b5	DCPMODE	DCP Mode Control	0: Disable RDCP_DAT resistor 1: Enable RDCP_DAT resistor.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CHGDETSTS	CHGDET Status Flag	0: The CHGDET pin is at low level. 1: The CHGDET pin is at high level.	R
b9	PDDETSTS	PDDET Status Flag	0: The PDDET pin is at low level. 1: The PDDET pin is at high level.	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. All bits in the BCCTRL register can be changed while the UTMI clock is inactive.

Note 2. In device controller mode, set the IDPSRCE, IDMSINKE, VDPSRCE, IDPSINKE, and VDMSRCE bits to 1 after setting the SYSCFG.DRPD bit to 0.

IDPSRCE bit (IDPSRC Control)

In device controller mode, set the IDPSRCE bit to 1 to perform data contact detection.

The Battery Charging Standard provides two ways to handle data contact detection, one through the software and one using hardware to contact the data line. The IDPSRE bit uses the hardware method.

When the IDPSRE bit is set to 1, the USBHS enables the IDP_SRC circuit and, at the same time, controls D- pull-down. (D- pull-down is controlled with the VUH_DMPULLDOWN signal.)

IDMSINKE bit (IDMSINK Control)

In device controller mode, set the IDMSINKE bit to 1 to perform primary detection.

VDPSRCE bit (VDPSRC Control)

In device controller mode, set the VDPSRCE bit to 1 to perform primary detection.

IDPSINKE bit (IDPSINK Control)

In device controller mode, set the IDPSINKE bit to 1 to perform secondary detection. In host controller mode, set this bit to 1 to enable the portable device detection circuit.

VDMSRCE bit (VDMSRC Control)

In device controller mode, set the VDMSRCE bit to 1 to perform secondary detection. Setting this bit to 1 enables the DCP detection circuit. In host controller mode, set this bit to 1 when a portable device is detected. Setting this bit to 1 allows the device that is performing primary detection to determine the charger detection method.

DCPMODE bit (DCP Mode Control)

Set the DCPMODE bit to 1 to operate as a dedicated charging port (DCP). Setting this bit to 1 disables USB communication.

CHGDETSTS flag (CHGDET Status Flag)

The CHGDETSTS flag indicates the charger port detection state.

PDDETSTS flag (PDDET Status Flag)

The PDDETSTS flag indicates the following states based on the controller mode:

- In host controller mode: PD detection state
- In device controller mode: DCP detection state.

33.2.45 Function L1 Control Register 1 (PL1CTRL1)

Address(es): USBHS.PL1CTRL1 4006 0144h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	L1EXTMD	—	—	HIRDTHR[3:0]				DVSQ[3:0]				L1NEGOMD	L1RESPMD[1:0]		L1RESPEN
Value after reset:	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	L1RESPEN	L1 Response Enable	0: Do not support LPM 1: Support LPM.	R/W
b2, b1	L1RESPMD[1:0]	L1 Response Mode	b2 b1 0 0: NYET response 0 1: ACK response 1 0: STALL response 1 1: Response based on L1NEGOMD setting.	R/W
b3	L1NEGOMD	L1 Response Negotiation Control	0: Return ACK when received HIRD is larger than HIRDTHR[3:0]. Otherwise (including when HIRD = HIRDTHR[3:0]), return NYET. 1: Return ACK when received HIRD is smaller than HIRDTHR[3:0]. Otherwise (including when HIRD = HIRDTHR[3:0]), return NYET. This bit is only valid when the L1RESPMD[1:0] value is 11b.	R/W
b7 to b4	DVSQ[3:0]	DVSQ Extension Flag	b7 b4 0 0 0 0: Powered state 0 0 0 1: Default state 0 0 1 0: Address state 0 0 1 1: Configured state 0 1 x x: Suspend state 1 0 x x: L1 state.	R
b11 to b8	HIRDTHR[3:0]	L1 Response Negotiation Threshold Value	HIRD threshold value used when the L1RESPMD[1:0] bits are 11b. The format is the same as the HIRD field in HL1CTRL.	R/W
b13, b12	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b14	L1EXTMD	PHY Control Mode at L1 Return	0: Do not set LPSTS.SUSPENDM bit through hardware when Host K is received 1: Set LPSTS.SUSPENDM bit through hardware when Host K is received.	R/W
b15	—	Reserved	The read value is undefined. The write value should be 0.	R/W

L1RESPEN bit (L1 Response Enable)

If the USBHS receives an LPM token while the L1RESPEN bit is 0, it returns no response. If the USBHS receives an

LPM token while this bit is 1, it returns a response based on the L1RESPMD[1:0] setting.

L1RESPMD[1:0] bits (L1 Response Mode)

When the L1RESPEN bit is set to 1, the USBHS returns a response to the LPM token based on the setting in the L1RESPMD[1:0] bits.

L1NEGOMD bit (L1 Response Negotiation Control)

The L1NEGOMD bit specifies the negotiation function for the HIRD value.

HIRDTHR[3:0] bits (L1 Response Negotiation Threshold Value)

The HIRDTHR[3:0] bits specify the HIRD threshold value used for L1NEGOMD. The format of the set value is the same as the HIRD field in HL1CTRL.

L1EXTMD bit (PHY Control Mode at L1 Return)

The L1EXTMD bit specifies the LPSTS.SUSPENDM bit control method when a host K signal is received in the L1 state while the LPSTS.SUSPENDM bit is 0 and the PHY is inactive.

Similar to the Suspend constraints, because the minimum host K period is 50 µs, the PHY might not recover within the host K period specified for software settings on return. The initial value is within software control, so set this bit to 1 during the initialization process when the L1 state is supported.

The LPSTS.SUSPENDM bit is controlled by software on transition to the L1 state regardless of the setting in this bit. It is not cleared by hardware.

When this bit is set to 1, the LPSTS.SUSPENDM bit is also set to 1 on return from L2.

33.2.46 Function L1 Control Register 2 (PL1CTRL2)

Address(es): USBHS.PL1CTRL2 4006 0146h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RWEMON	HIRDMON[3:0]				—	—	—	—	—	—	—	—
Value after reset:	x	x	x	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b11 to b8	HIRDMON[3:0]	HIRD Value Monitor	When set, indicates that the HIRD field value reflects the last-received LPM token.	R
b12	RWEMON	RWE Value Monitor	When set, indicates that the RWE bit value reflects the last-received LPM token.	R
b15 to b13	—	Reserved	The read value is undefined. The write value should be 0.	R/W

HIRDMON[3:0] bits (HIRD Value Monitor)

Access the HIRDMON[3:0] bits when monitoring the HIRD field value of the received LPM token. The bits reflect the HIRD field value of the last received LPM token.

RWEMON bit (RWE Value Monitor)

Access the RWEMON bit when monitoring the RWE field value of the received LPM token. The bits reflect the RWE field value of the last received LPM token.

33.2.47 Host L1 Control Register 1 (HL1CTRL1)

Address(es): USBHS.HL1CTRL1 4006 0148h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	L1STATUS[1:0]		L1REQ
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	L1REQ	L1 Transition Request	Set this bit to 1 when requesting a transition to the L1 state. This bit is cleared to 0 by the hardware when the LPM transaction is complete.	R/W
b2, b1	L1STATUS [1:0]	L1 Request Completion Status	Indicates the result of the LPM transaction made by the L1REQ bit: b2 b1 0 0: ACK received 0 1: NYET received 1 0: STALL received 1 1: Transaction error.	R
b15 to b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W

L1REQ bit (L1 Transition Request)

Set the L1REQ bit to 1 to transition to the L1 state. When the USBHS detects that this bit is 1, it starts the LPM transaction. The USBHS clears this bit to 0 through hardware on completion of the transaction.

L1STATUS[1:0] bits (L1 Request Completion Status)

The L1STATUS[1:0] bits indicate the result of the LPM transaction initiated by the L1REQ bit.

33.2.48 Host L1 Control Register 2 (HL1CTRL2)

Address(es): USBHS.HL1CTRL2 4006 014Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BESL	—	—	L1RWE	HIRD[3:0]				—	—	—	—	L1ADDR[3:0]			
Value after reset:	0	x	x	0	0	0	0	0	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	L1ADDR[3:0]	LPM Token DeviceAddress	Specify the value to be set in the ADDR field of the LPM token	R/W
b7 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b11 to b8	HIRD[3:0]	LPM Token HIRD	Specify the value to be set in the HIRD field of the LPM token	R/W
b12	L1RWE	LPM Token L1 RemoteWake Enable	Specify the value to be set in the RWE field of the LPM token	R/W
b14, b13	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b15	BESL	BESL & Alternate HIRD	Selects the K-State drive period on L1 Resume	R/W

L1ADDR[3:0] bits (LPM Token DeviceAddress)

The L1ADDR[3:0] bits specify the value to be set in the ADDR field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1.

HIRD[3:0] bits (LPM Token HIRD)

The HIRD[3:0] bits specify the value to be set in the HIRD field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1. Table 33.15 shows the relationship between the HIRD settings and the HIRD field values.

Table 33.15 Relationship between the HIRD bit settings and the HIRD field values

HIRD[3:0] setting	When BESL = 0	When BESL = 1
0000b	50 µs (setting prohibited)	75 µs
0001b	125 µs	100 µs
0010b	200 µs	150 µs
0011b	275 µs	250 µs
0100b	350 µs	350 µs
0101b	425 µs	450 µs
0110b	500 µs	950 µs
0111b	575 µs	1950 µs
1000b	650 µs	2950 µs
1001b	725 µs	3950 µs
1010b	800 µs	4950 µs
1011b	875 µs	5950 µs
1100b	950 µs	6950 µs
1101b	1025 µs (setting prohibited)	7950 µs
1110b	1100 µs (setting prohibited)	8950 µs
1111b	1175 µs (setting prohibited)	9950 µs

Note: The set value of the HIRD bit is used for the host K drive period on host resume and for the host K period on remote wakeup.

L1RWE bit (LPM Token L1 RemoteWake Enable)

The L1RWE bit specifies the value to be set in the RWE field of the LPM token that the USBHS transmits when the HL1CTRL1.L1REQ bit is set to 1.

The USBHS does not control detection of the remote wakeup signal in the L1 state with this bit. The remote wakeup signal is controlled by the DVSTCTR0.RWUPE bit, as with Suspend.

BESL bit (BESL & Alternate HIRD)

The BESL bit selects the K-state drive period on L1 Resume. For details, see the description of the HIRD bits.

33.2.49 Deep Software Standby USB Transceiver Control/Pin Monitor Register (DPUSR0R)

Address(es): USBHS.DPUSR0R 4006 0160h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTSHM	—	DOVCBHM	DOVCAHM	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b19 to b0	—	Reserved	These bits are read as 0	R
b20	DOVCAHM	OVRCURA Input Flag	Indicates OVRCURA input signal on the USBHS side	R
b21	DOVCBHM	OVRCURB Input Flag	Indicates OVRCURB input signal on the USBHS side	R
b22	—	Reserved	This bit is read as 0	R
b23	DVBSTSHM	VBUS Input Flag	Indicates VBUS input signal on the USBHS side	R
b31 to b24	—	Reserved	These bits are read as 0	R

33.2.50 Deep Software Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address(es): USBHS.DPUSR1R 4006 0164h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTSH	—	DOVCBH	DOVCAH	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DVBSTSHE	—	DOVCBHE	DOVCAHE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DOVCAHE	OVRCURA Interrupt Enable Clear	0: Disable recovery from Deep Software Standby mode 1: Enable recovery from Deep Software Standby mode.	R/W
b5	DOVCBHE	OVRCURB Interrupt Enable Clear	0: Disable recovery from Deep Software Standby mode 1: Enable recovery from Deep Software Standby mode.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	DVBSTSHE	VBUS Interrupt Enable/Clear	0: Disable recovery from Deep Software Standby mode 1: Enable recovery from Deep Software Standby mode.	R/W
b19 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVCAH	OVRCURA Interrupt Source Return Status Flag	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode.	R
b21	DOVCBH	OVRCURB Interrupt Source Return Status Flag	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode.	R

Bit	Symbol	Bit name	Description	R/W
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBSTSH	VBUS Interrupt Source Return Status Flag	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.51 Deep Software Standby USB Suspend/Resume Interrupt Register (DPUSR2R)

Address(es): USBHS.DPUSR2R 4006 0168h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DMINTE	DPINTE	—	—	DMVAL	DPVAL	—	—	DMINT	DPINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DPINT	Indication of Return from DP Interrupt Source	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode.	R
b1	DMINT	Indication of Return from DM Interrupt Source	0: System has not recovered from Deep Software Standby mode 1: System recovered from Deep Software Standby mode.	R
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DPVAL	DP Input	Indicates DP input signal on the USBHS side	R
b5	DMVAL	DM Input	Indicates DM input signal on the USBHS side	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DPINTE	DP Interrupt Enable Clear	0: Disable recovery from Deep Software Standby mode 1: Enable recovery from Deep Software Standby mode.	R/W
b9	DMINTE	DM Interrupt Enable Clear	0: Disable recovery from Deep Software Standby mode 1: Enable recovery from Deep Software Standby mode.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.2.52 Deep Software Standby USB Suspend/Resume Command Register (DPUSRCR)

Address(es): USBHS.DPUSRCR 4006 016Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIXPHYPD	FIXPHY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FIXPHY	USB Transceiver Control Fix	0: Normal mode 1: Invoke/recover from Deep Software Standby mode.	R/W
b1	FIXPHYPD	USB Transceiver Control Fix for PLL	0: Normal mode 1: Invoke/recover from Deep Software Standby mode.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

33.3 Operation

33.3.1 System Control

This section describes register settings required for initializing the USBHS and controlling power consumption.

33.3.1.1 Setting data to the USBHS registers

Setting the SYSCFG.USBE bit to 1 after starting the PHY clock supply enables and starts USBHS operation. For information on how to supply the PHY clock, see section 33.3.3, Supplying the Clock.

33.3.1.2 Selecting the controller function

The USBHS can operate as a host or device controller.

Use the SYSCFG.DCFM bit to select one of these USBHS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D- pull-down-disabled state (SYSCFG.DRPD bit = 0).

33.3.2 Controlling the USB data bus using resistors

The USBHS provides pull-up and pull-down resistors for the D+ and D- lines. Pull these lines up or down by setting the SYSCFG.DPRPU and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication).

When the SYSCFG.DPRPU bit is set to 0 during communication with a PC, the USBHS disables the pull-up resistor for the USB data line, thereby notifying the USB host of disconnection.

In host controller mode, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D- lines.

Table 33.16 shows the settings for controlling the resistors for the USB data bus. Control the USB data bus appropriately for your system using the DRPD and DPRPU bit settings.

Table 33.16 Control settings for the USB data bus resistors (excluding OTG operation)

SYSCFG register settings		USB data bus control		
DRPD bit	**DPRPU bit**	**D-Line**	**D+Line**	**Function**
0	0	Open	Open	When resistors not used
0	1	Open	Pull-Up	When operating as a device controller at full-speed
1	0	Pull-Down	Pull-Down	When operating as a host controller
1	1	—	—	Setting prohibited except during OTG operation

33.3.3 Supplying the Clock

Table 33.17 shows the two input clocks required for the USBHS.

Table 33.17 Input clocks

Input clock name	Description
PCLKA	Peripheral module clock A input. There is no constraint on the frequency of the PCLKA input.
PHY clock	PHY clock generated from external input or internal supply • External input: The clock is generated by the USB-PHY internal PLL based on a 12-MHz, 20-MHz, or 24-MHz clock supplied to the EXTAL pin from outside the MCU. For the external clock specifications, especially the jitter characteristics, strictly follow the specifications of ±50 ppm. • Internal supply: The clock is generated by supplying 48 MHz and 60 MHz to the USB-PHY module and selecting CL-only mode (PHYSET.HSEB). High-speed operation is not supported with this mode.

Figure 33.2 illustrates the PHY clock settings.

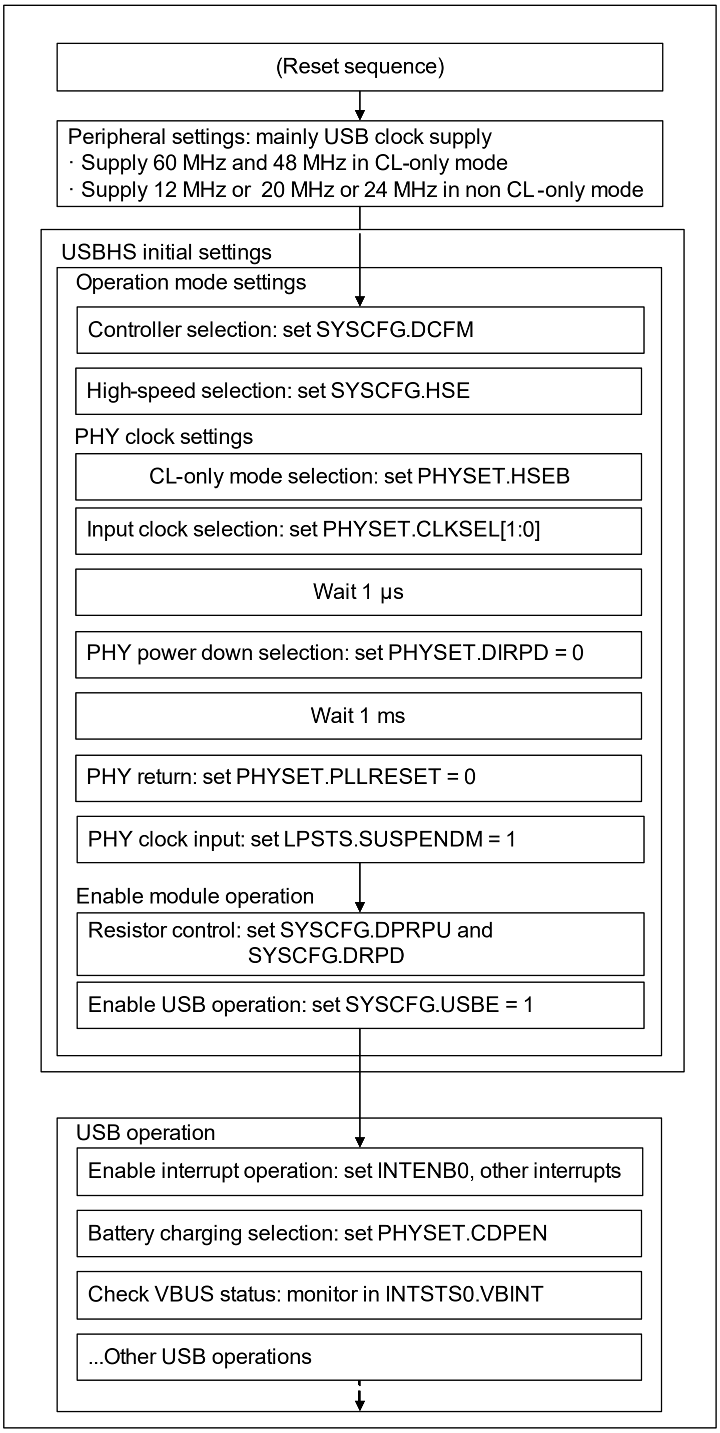


Figure 33.2 PHY clock settings

33.3.4 Constraints on Stopping the Clock

PCLKA and PHY clock can be stopped during disconnection or suspension. However, to stop any of these clocks while the USB is suspended in device controller mode, the stopped clock must be resupplied using the resume interrupt. The PHY clock must be resupplied within 5.5 ms after the resume interrupt is generated.

33.3.5 Interrupts

Table 33.18 lists the interrupt sources in the USBHS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, the USBHS issues a USBHS interrupt request to the Interrupt Controller Unit (ICU) and a USBHS interrupt is generated.

Table 33.18 Interrupt sources (1 of 2)

Flag to be set to 1	Interrupt name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	• A change in the state of the USB_VBUS input pin is detected (low to high or high to low)	Host or function*1	INTSTS0.VBSTS
RESM	Resume interrupt	• A change in the state of the USB bus is detected in the Suspend state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	In host controller mode: • An SOF packet with a different frame number is transmitted In device controller mode: • When SOFRM is 0: An SOF packet with a different frame number is received • When SOFRM is 1: Failed to receive an SOF packet with the μ frame number 0 because the packet is corrupted.	Host or function	—
DVST	Device state transition interrupt	• A device state transition is detected because of one of the following: - USB bus reset is detected - Suspend state is detected - SET_ADDRESS request is received - SET_CONFIGURATION request is received.	Function	PL1CTRL.DVSQ[3:0]
CTRT	Control transfer stage transition interrupt	• A control transfer stage transition is detected because of one of the following: - Setup stage completed - Control write transfer status stage transition occurred - Control read transfer status stage transition occurred - Control transfer completed - Control transfer sequence error occurred	Function	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• The buffer is empty after all FIFO buffer data is transmitted • A packet larger than the maximum packet size is received	Host or function	BEMPSTS.PIPEBEMP
NRDY	Buffer not ready interrupt	In host controller mode: • A STALL response is received from the peripheral device in response to the issued token • The response from the peripheral device in response to the issued token is not received successfully (no response three times consecutively or packet reception error three times consecutively) • An overrun or underrun error occurred during isochronous transfer In device controller mode: • NAK is returned for an IN or OUT token while the PID[1:0] bits were set to 01b (BUF) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer • An interval error occurred during data reception in isochronous transfer	Host or function	NRDYSTS.PIPENRDY
BRDY	Buffer ready interrupt	• The buffer is ready (read or write state)	Host or function	BRDYSTS.PIPEBRDY

Table 33.18 Interrupt sources (2 of 2)

Flag to be set to 1	Interrupt name	Interrupt source	Applicable controller function	Status flag
OVRCR	Overcurrent input change interrupt	• USBHS_OVRCR0A pin or USBHS_OVRCR0B pin state change is detected (low to high or high to low)	Host or function	SYSSTS0.OVCMON[1:0]
BCHG	Bus change interrupt	• USB bus state change is detected	Host	—
DTCH	Device disconnect detection interrupt	• Peripheral device disconnect is detected	Host	—
ATTCH	Device connect detection interrupt	• J-state or K-state is detected on the USB bus for 2.5 µs continuously This interrupt can be used to check whether peripheral devices are connected.	Host	—
EOFERR	EOF error detection interrupt	• An EOF error is detected for a peripheral device	Host	—
SACK	Setup normal interrupt	• A setup transaction normal response (ACK) is received	Host	—
SIGN	Setup error interrupt	• A setup transaction error (no response or ACK packet corruption) is detected three consecutive times	Host	—
PDDETINT	PDDETSTS change detect interrupt	• PDDET pin change is detected	Host or function	BCCTRL.PDDETSTS
LPMEND	LPM transaction end interrupt	• LPM transaction is complete	Host	PL1CTRL.DVSQ[3:0]
L1RSMEND	L1 resume end interrupt	• Resume (from L1 state) processing is complete	Host	PL1CTRL.DVSQ[3:0]

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

33.3.5.1 Selecting the USBHS interrupt detection method

Table 33.19 shows operations for an USBHS interrupt output from the USBHS. In case two or more interrupt sources are generated, the USBHS interrupt output method can be set in the SOFCFG.INTL bit. Set the USBHS interrupt output operation based on your system.

Table 33.19 USBHS interrupt operation

USBHS interrupt output (INTL setting)	When one interrupt source is generated	When two or more interrupt sources are generated
Edge detection (SOFCFG.INTL bit = 0)	Low level output until the source is cleared	When one source is cleared, the USBHS interrupt is negated for 32 clocks at 48 MHz (high pulse output)
Level detection (SOFCFG.INTL bit = 1)	Low level output until the source is cleared	Low level output until all sources are cleared

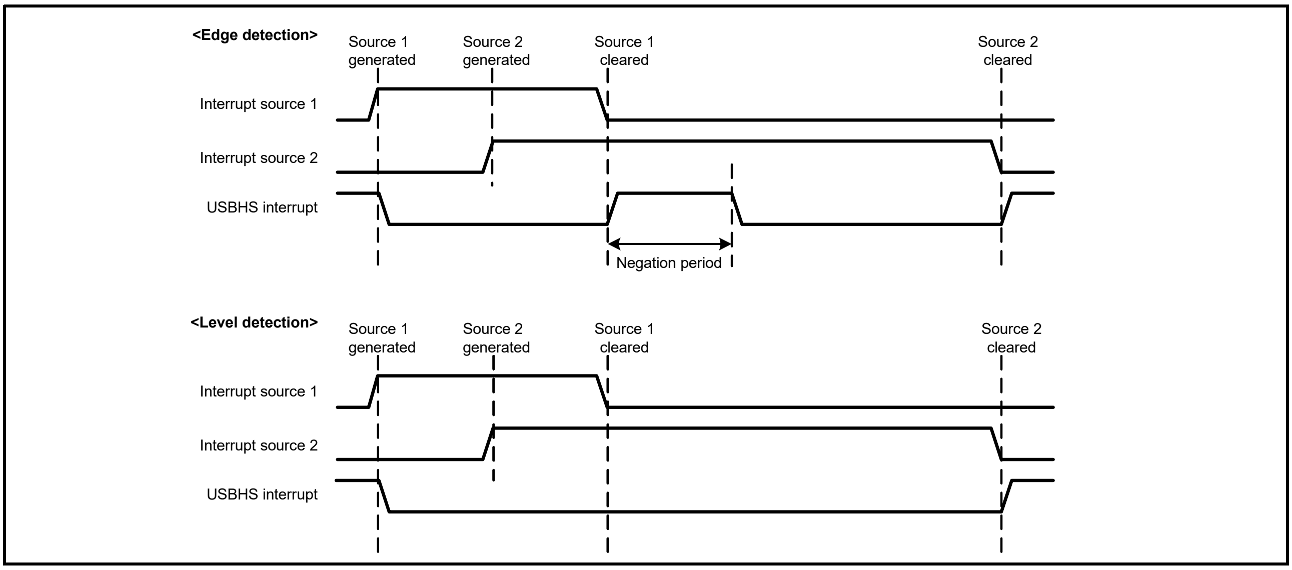


Figure 33.3 USBHS interrupt operation

Figure 33.4 shows an interrupt association chart of the USBHS.

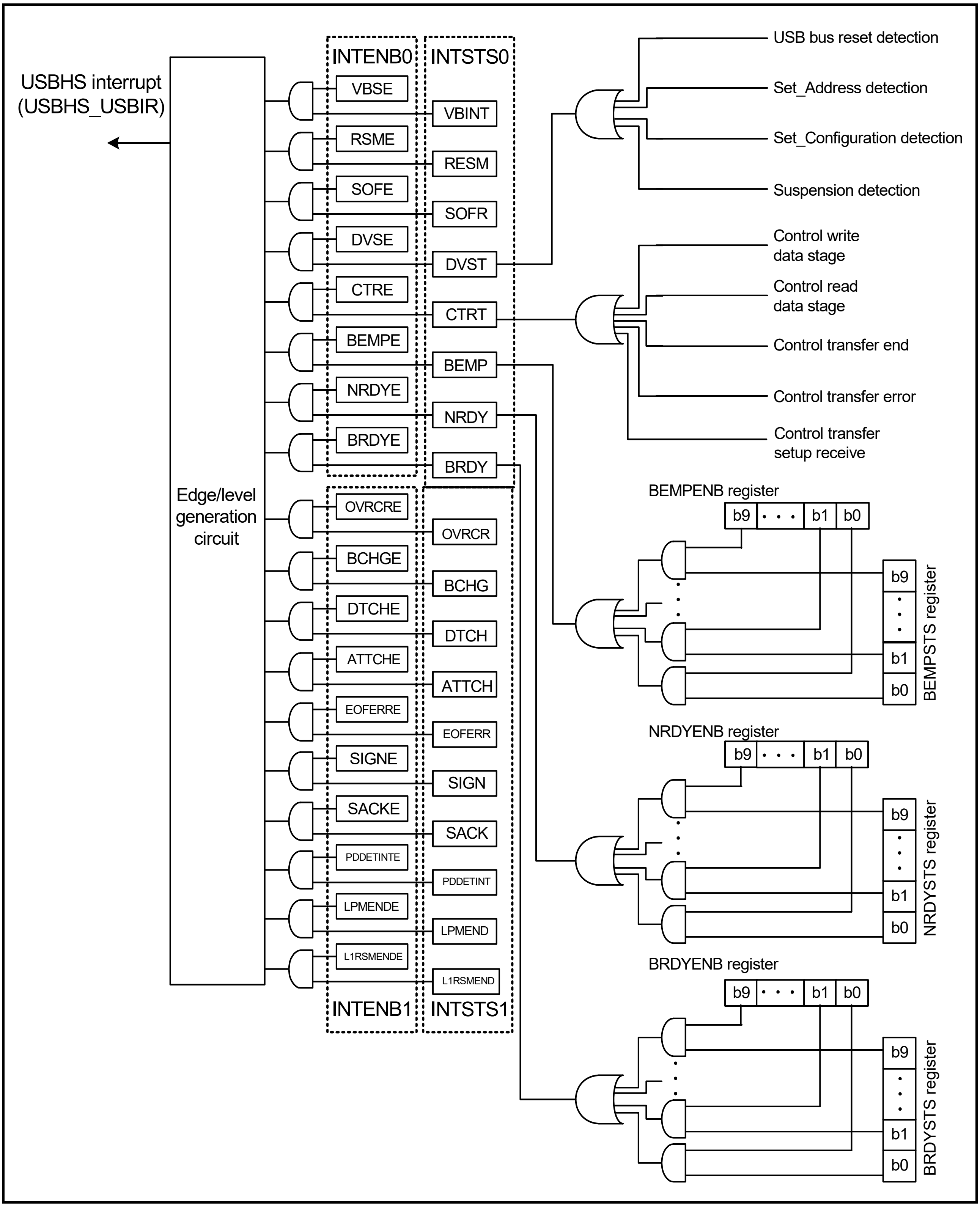


Figure 33.4 USBHS interrupt-related circuits

Table 33.20 shows the interrupts generated by the USBHS.

Table 33.20 USBHS interrupts

Interrupt name	Interrupt status flag	DTC activation	DMAC activation
USBHS_D0FIFO	DMA transfer request 0	Possible	Possible
USBHS_D1FIFO	DMA transfer request 1	Possible	Possible
USBHS_USBIR	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent interrupt, bus change interrupt, device disconnect detection interrupt, device connect detection interrupt, EOF error detection interrupt, normal setup operation interrupt, setup error interrupt, PDDETSTS change detection interrupt, LPM transaction end interrupt, and L1 resume end interrupt	Not possible	Not possible

33.3.6 Interrupt Descriptions

33.3.6.1 BRDY interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBHS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBHS generates a BRDY interrupt if the software sets the bit in BRDYENB associated with the given pipe to 1 and INTENB0.BRDYE bit to 1.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

(1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBHS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEBRDY flag associated with the pipe to 1.

(a) For transmitting pipes

- When the DIR bit is changed from 0 to 1 by software
- When writing by the CPU to the FIFO buffer is disabled for a pipe (when the BSTS flag is read as 0) and the USBHS has completed packet transmission. In continuous transfer, a BRDY interrupt is generated on completion of the transmission of data from one FIFO buffer.
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete
- When the hardware flushes the buffer of the pipe for isochronous transfers
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

(b) For receiving pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS flag is read as 0). No request trigger is generated for transactions in which DATA-PID mismatch has occurred. In continuous transmission or reception mode, the request trigger is not generated when the data is of the specified maximum packet size and the buffer has available space. When a short packet is received, the request trigger is generated even if the FIFO buffer has available space. When the transaction counter is used, the request trigger is generated on receiving the specified number of packets. In this case, the request trigger is generated even if the FIFO buffer has available space.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.

In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEBRDY flag through the software. In this case, 1s must be written to the associated bits for the other pipes. Clear the BRDY status before accessing the FIFO buffer.

(2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBHS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets the bit in BRDYSTS associated with the pipe to 1.

On any of the following conditions, the USBHS determines that the last data for a single transfer was received:

- When a short packet including a zero-length packet is received
- When the PIPEnTRN register is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the data is completely read after any of these conditions is satisfied, the USBHS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBHS determines that all data for a single transfer is completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[11:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through software. With these settings, the USBHS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEBRDY flag through the software. In this case, 1s must be written to the PIPEBRDY bits for the other pipes.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

(3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEBRDY flag values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status bits are set to 1 or 0 by the USBHS depending on the FIFO buffer status.

(a) For transmitting pipes

The BRDY interrupt status bits are set to 1 when the FIFO port is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

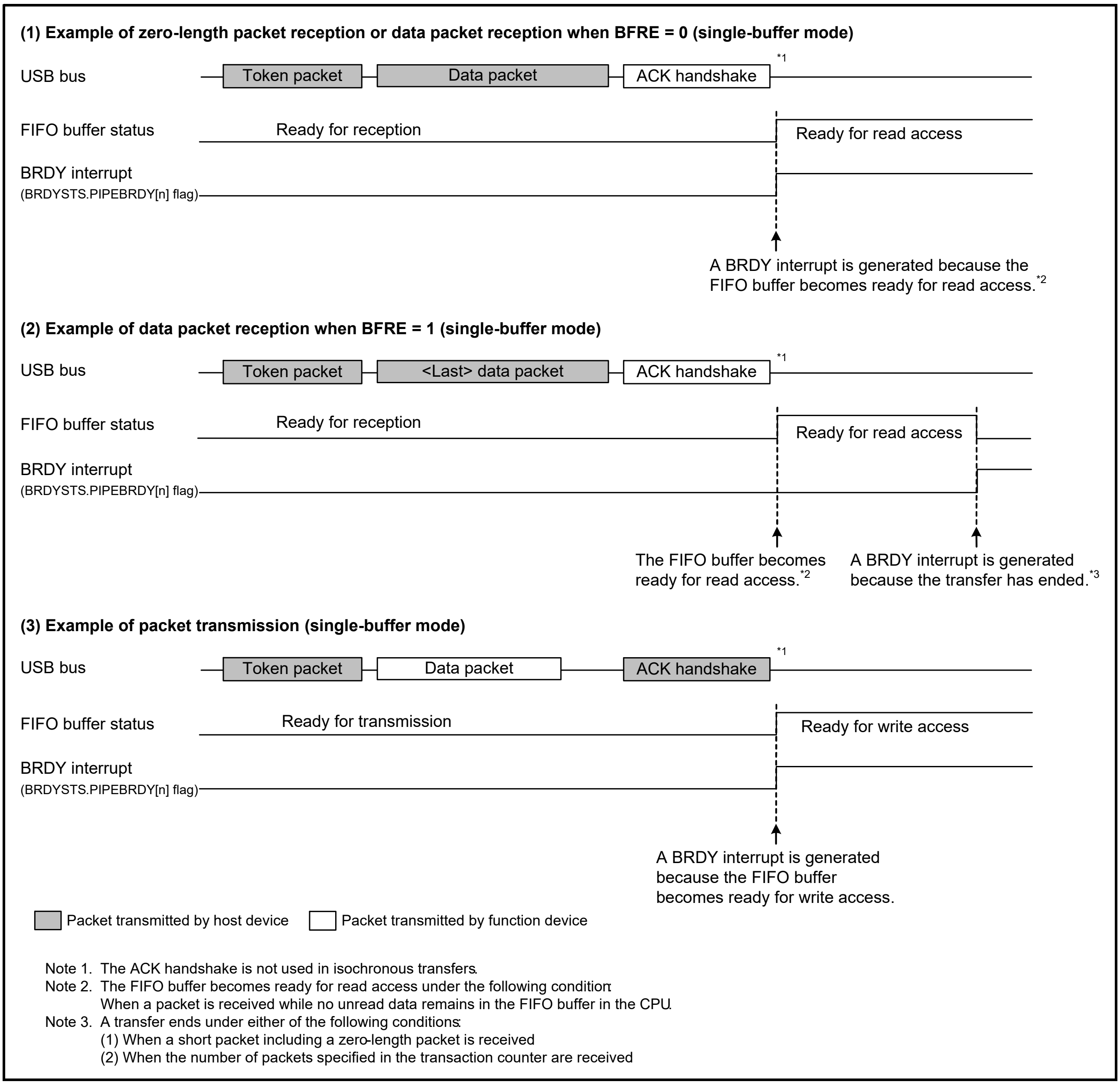
(b) For receiving pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data is read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEBRDY flag cannot be set to 0 by software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0, and the SOFCFG.INTL bit to 1 for level detection.

Figure 33.5 shows the timing of BRDY interrupt generation.



Note 1. The ACK handshake is not used in isochronous transfers.
Note 2. The FIFO buffer becomes ready for read access under the following condition:
When a packet is received while no unread data remains in the FIFO buffer in the CPU.
Note 3. A transfer ends under either of the following conditions:
(1) When a short packet including a zero-length packet is received
(2) When the number of packets specified in the transaction counter are received

Figure 33.5 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting value, as shown in Table 33.21.

Table 33.21 Conditions for clearing the BRDY flag

BRDYM bit	Condition for clearing BRDY flag
0	The USBHS clears the BRDY flag to 0 when all bits in BRDYSTS are set to 0 by software
1	The USBHS clears the BRDY flag to 0 when the BSTS flags for all pipes have cleared to 0

33.3.6.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF response) by software, the USBHS sets the associated NRDYSTS.PIPENRDY flag to 1. If the associated bit in NRDYENB is set to 1 by software, the USBHS sets the INTSTS0.NRDY flag to 1 and generates a USBHS interrupt.

This section describes the conditions in which the USBHS generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode.

(1) In host controller mode when no split transactions occur in the connection

(a) For transmitting pipes

On any of the following conditions, the USBHS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPENRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two conditions occurs three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device
  - An error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the PID[1:0] setting for the associated pipe to 00b (NAK response)
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device (includes STALL for both OUT and PING). In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the PID[1:0] setting for the associated pipe to 11b (STALL response).

(b) For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBHS discards the received data for the IN token and sets the associated PIPENRDY flag and the OVRN flag to 1. When a packet error is detected in the received data for the IN token, the USBHS also sets the FRMNUM.CRCE flag to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBHS (when timeout is detected before detection of the DATA packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated PIPENRDY flag to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response).
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe and the CRCE flag to 1.
- When the STALL handshake is received. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

(2) In host controller mode when split transactions occur in the connection

(a) For transmitting pipes

On any of the following conditions, the USBHS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS sets the associated RDYSTS.PIPENRDY flag for the given pipes to 1 on issuing a start-split transaction and sets the FRMNUM.OVRN flag to 1. The USBHS also transmits a zero-length packet following the OUT token.

- For non-isochronous transfer pipes, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for start-split and complete-split transactions (when timeout is detected before detection of the handshake packet from the hub)
  - An error is detected in the packet from the hub. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response).When an NRDY interrupt is detected on complete-split issuance, the USBHS clears the CSSTS flag to 0.
- When a STALL handshake is received for the complete-split transaction. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, changes the associated PID[1:0] setting for the pipe to 11b (STALL response), and clears the CSSTS flag to 0. An interrupt is not detected during setup transaction.

(b) For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the given pipe and the FRMNUM.OVRN flag to 1 on start-split issuance. The USBHS discards the received data for the IN token.
- During bulk-pipe transfers or transfers other than setup transactions with the DCP, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for the IN token the USBHS issued on issuance of the start-split or complete-split transactions (when timeout is detected before detection of the data packet from the hub)
  - An error is detected in the packet from the hub. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1 and changes the associated PID[1:0] setting for the pipe to 00b (NAK response). When this condition occurs during complete-split, the USBHS clears the CSSTS flag to 0.
- During a complete-split transaction for isochronous transfer or interrupt transfer pipes, when any combination of the following two cases occurs three consecutive times:
  - No response is returned from the hub for the IN token issued by the USBHS (when a timeout is detected before detection of the DATA packet from the hub)
  - An error is detected in the packet from the hub. On generating this condition for an interrupt transfer pipe, the USBHS sets the associated NRDYSTS.PIPENRDY flag to 1, changes the associated PID[1:0] setting for the pipe to 00b (NAK response), and clears the CSSTS flag to 0. On generating this condition for the pipe for isochronous transfers, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, CRCE flag to 1, and clears the CSSTS bit to 0. It does not change the PID[1:0] setting.
- During a complete-split transaction, when the STALL handshake is received for a non-isochronous transfer pipe. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for the pipe to 1, changes the associated PID[1:0] setting for the pipe to 11b (STALL response), and clears the CSSTS flag to 0.
- During a complete-split transaction, when the NYET handshake is received for an isochronous transfer or interrupt transfer pipe for the microframe number = 4. In this case, the USBHS sets the associated NRDYSTS.PIPENRDY flag for each pipe to 1 and the CRCE flag to 1, and clears the CSSTS flag to 0. It does not change the PID[1:0] setting.

(3) In device controller mode

(a) For transmitting pipes

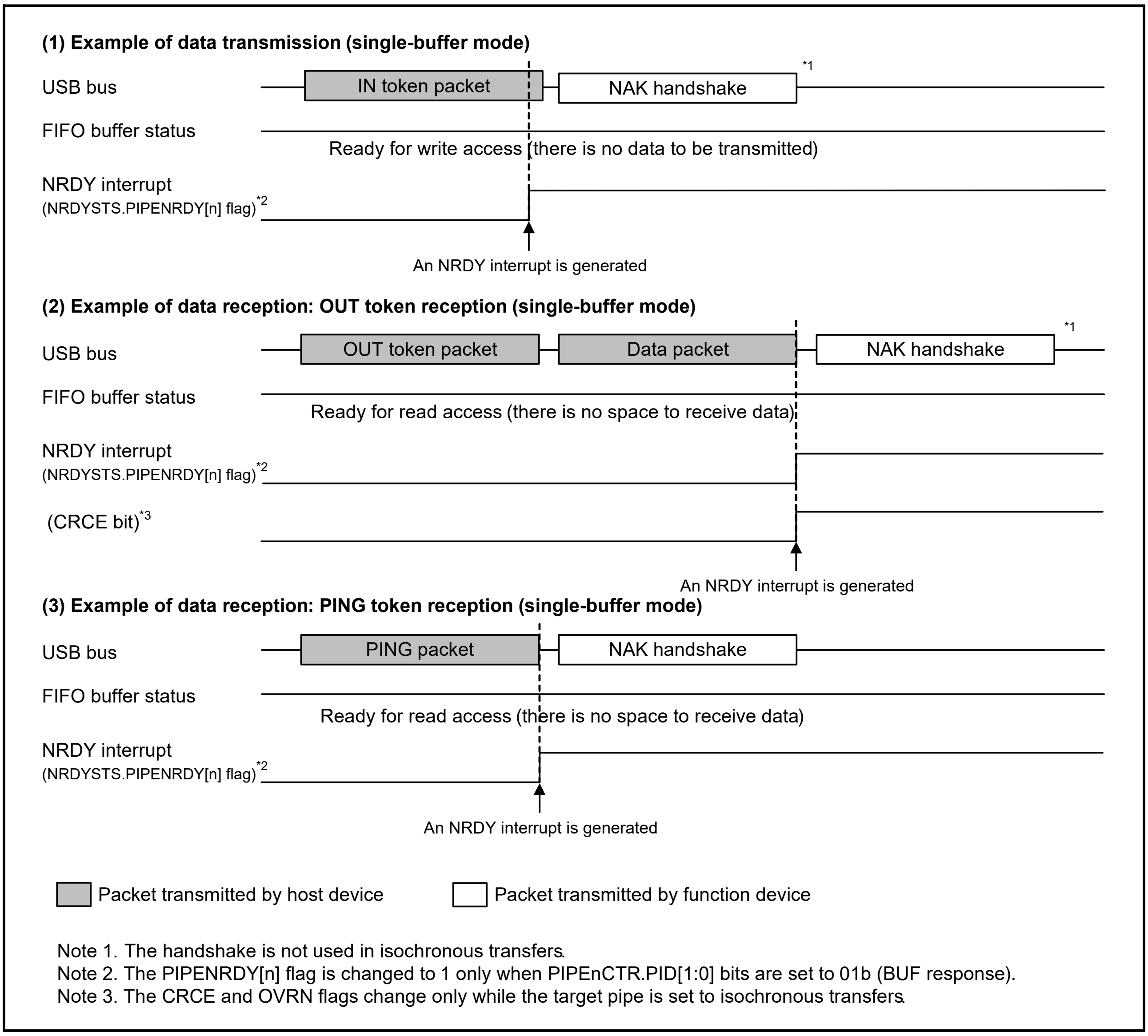
- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBHS generates an NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPENRDY flag to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBHS transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBHS generates an NRDY interrupt request on reception of the OUT token and sets the NRDYSTS.PIPENRDY flag to 1 and the FRMNUM.OVRN flag to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBHS generates an NRDY interrupt request when an NAK handshake is transferred after the data following the OUT token is received, and sets the NRDYSTS.PIPENRDY flag to 1. The NRDY interrupt request is not generated during retransmission because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.

- On receiving a PING token when there is no space available in the FIFO buffer. The USBHS generates an NRDY interrupt request on reception of the PING token, setting the NRDYSTS.PIPENRDY flag to 1.
- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBHS generates an NRDY interrupt request when the SOF is received, and sets the NRDYSTS.PIPENRDY flag to 1.

Figure 33.6 shows the timing of NRDY interrupt generation in device controller mode.



Note 1. The handshake is not used in isochronous transfers.
Note 2. The PIPENRDY[n] flag is changed to 1 only when PIPEnCTR.PID[1:0] bits are set to 01b (BUF response).
Note 3. The CRCE and OVRN flags change only while the target pipe is set to isochronous transfers.

Figure 33.6 Timing of NRDY interrupt generation in device controller mode

33.3.6.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits in the pipe control register are set to 01b (BUF response) by software, the USBHS sets the associated BEMPSTS.PIPEBEMP flag to 1. If the associated BEMPENB bit is set to 1 by software, the USBHS sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt. This section describes the conditions in which the USBHS generates an internal BEMP interrupt request.

(1) For transmitting pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated in any of the following conditions:

- When the CPU or DMA/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode
- When the buffer is cleared (emptied) by setting 1 to the PIPEnCTR.ACLRM or the BCLR bit in the port control register
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode.

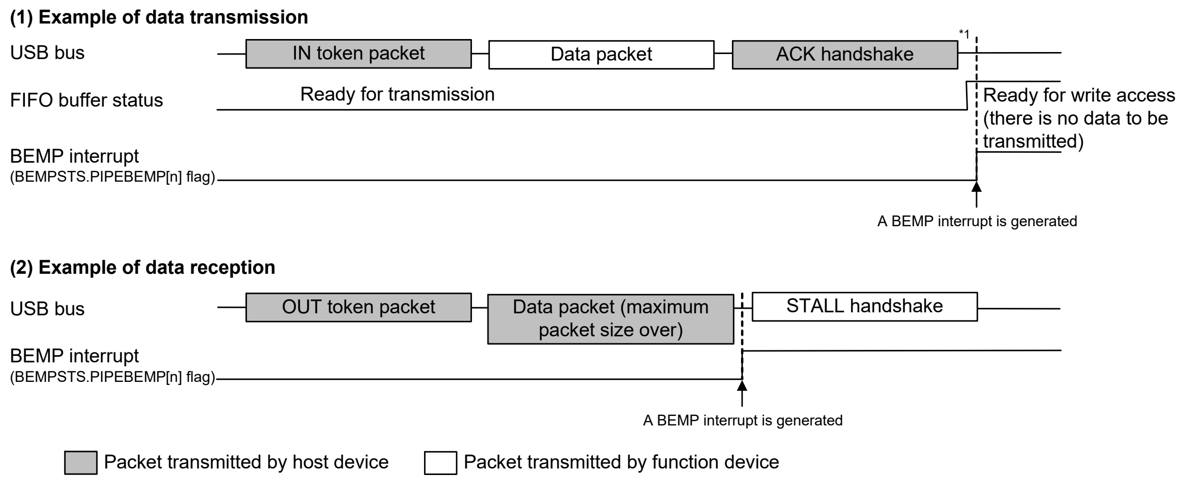
(2) For receiving pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBHS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEBEMP flag to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBHS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated in any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is being performed:
  - Writing 0 to the BEMPSTS.PIPEBEMP flag clears the status
  - Writing 1 to the BEMPSTS.PIPEBEMP flag has no effect.

Figure 33.7 shows the timing of BEMP interrupt generation in device controller mode.



Note 1. The handshake is not used in isochronous transfers.

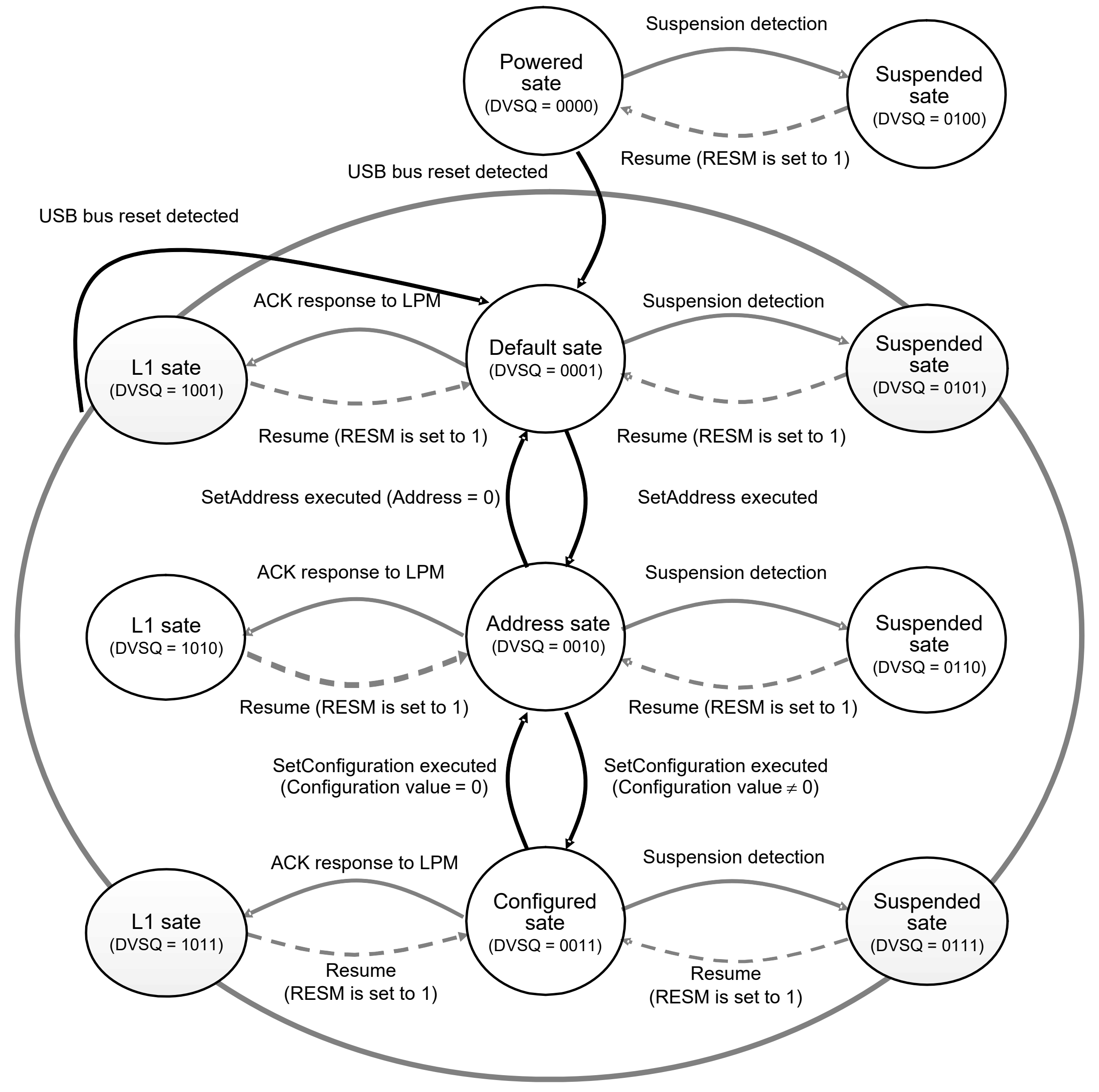
Figure 33.7 Timing of BEMP interrupt generation in device controller mode

33.3.6.4 Device state transition interrupt (device controller mode)

Figure 33.8 shows a diagram of the USBHS device state transitions. The USBHS controls device states and generates device state transition interrupts. However, recovery from the Suspend state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the PL1CTRL.DVSQ[3:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBHS controls device states, and device state transition interrupts can be generated, only in device controller mode.



Note: For a transition, indicated by a solid line, the DVST bit is set to 1. For a resume, indicated by a dashed line, the RESM bit is set to 1.

Figure 33.8 Device state transitions

33.3.6.5 Control transfer stage transition interrupt (device controller mode)

Figure 33.9 shows a diagram of the control transfer stage transitions of the USBHS. The USBHS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. When an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

(1) Control read transfer errors

- An OUT token or PING token is received but no data is transferred in response to the IN token at the data stage
- An IN token is received at the status stage
- A data packet with DATAPID = DATA0 is received at the status stage.

(2) Control write transfer errors

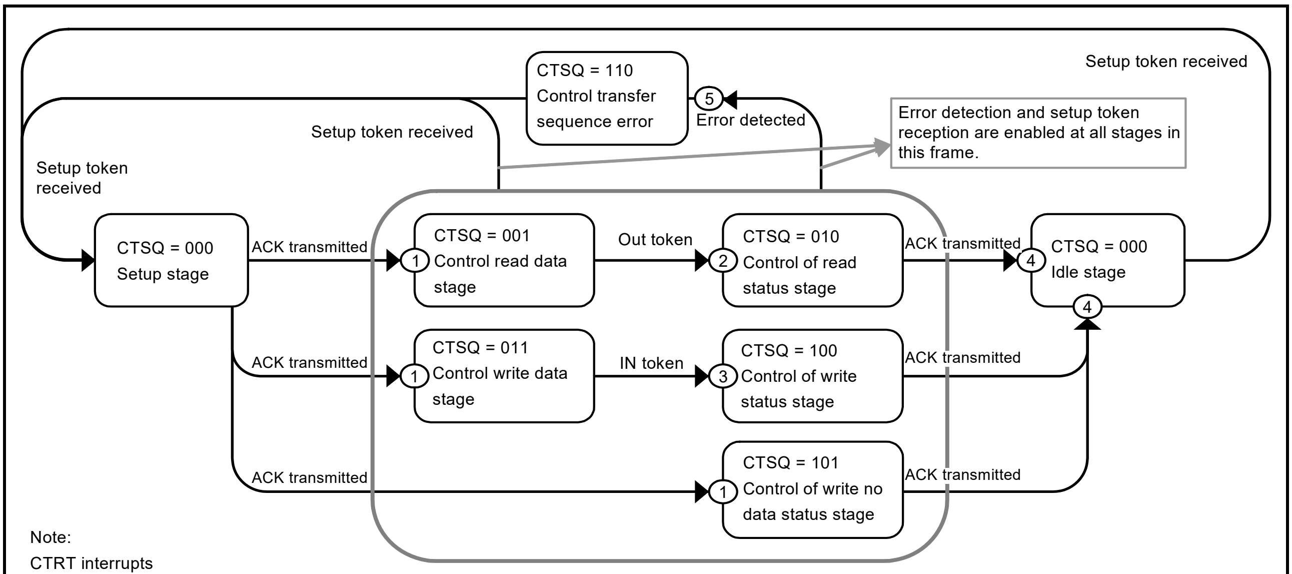
- An IN token is received but no ACK returned in response to the OUT token in the data stage
- A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
- An OUT token or PING token is received in the status stage.

(3) Control write no data transfer errors

- An OUT token or PING token is received at the status stage.

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT flag = 1), the CTSQ[2:0] = 110b value is saved until CTRT flag clears to 0, clearing the interrupt status. While CTSQ[2:0] bits = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBHS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by software.



Note:
CTRT interrupts
(1) Setup stage end
(2) Control read transfer status stage transition
(3) Control write transfer status stage transition
(4) Control transfer end
(5) Control transfer sequence error

Figure 33.9 Control transfer stage transitions

33.3.6.6 Frame update interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBHS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

33.3.6.7 VBUS interrupt

When the USBHS_VBUS pin level changes, a VBUS interrupt is generated. The level of the USBHS_VBUS pin can be checked in the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USBHS_VBUS pin level.

33.3.6.8 Resume interrupt

In device controller mode, a resume interrupt is generated when the device is in the Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

33.3.6.9 OVRCR interrupt

An OVRCR interrupt is generated when the USBHS_OVRCURA or USBHS_OVRCURB pin level has changed. The levels of the USBHS_OVRCURA and USBHS_OVRCURB pins can be checked in the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent is detected using the OVRCR interrupt.

For OTG connections, the OVRCR interrupt allows you to check whether a change is detected in the VBUS comparator.

33.3.6.10 BCHG interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected. It can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated in both host and device controller modes.

33.3.6.11 DTCH interrupt

A DTCH interrupt occurs when a USB bus disconnect is detected in host controller mode. The USBHS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. Regardless of the value set in the associated interrupt enable bit, the USBHS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state.

33.3.6.12 SACK interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

33.3.6.13 SIGN interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet is not correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

33.3.6.14 ATTCH interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs in host controller mode. To be more specific, an ATTCH interrupt is detected in any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

33.3.6.15 EOFERR interrupt

An EOFERR interrupt occurs when the USBHS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBHS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state.

33.3.6.16 PDDETINT interrupt

The USBHS sets the INTSTS1.PDDETINT flag to 1 on detecting a level change (high to low or low to high) in the PDDET pin input value and generates the PDDETINT interrupt. When the PDDETINT interrupt is generated, use software to repeatedly read the BCCTRL.PDDETSTS flag until the same value is read three or more times, and perform debounce processing.

33.3.6.17 LPMEND interrupt

When the LPM transaction ends because a response from the peripheral device or a timeout is detected, the INTSTS1.LPMEND flag sets to 1 and the LPMEND interrupt is generated.

33.3.6.18 L1RSMEND interrupt

When performing resume processing when the USBHS has transitioned to the L1 state because an ACK is received in response to an LPM token, the USBHS sets the INTSTS1.L1RSMEND flag to 1 on completion of the resume processing.

33.3.7 Pipe Control

Table 33.22 lists the pipe settings for the USBHS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBHS provides 10 pipes for data transfer. Set up the pipes based on your system specifications.

Table 33.22 Pipe settings (1 of 2)

Register name	Bit name	Setting	Notes
DCPCFG PIPECFG	TYPE[1:0]	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	CNTMD	Selection of continuous transfer or discontinuous transfer	Pipes 1, 2: Settable only for bulk transfers Pipes 3 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Settable Set this number to a value other than 0000 when one or more pipes are used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1, 2: Settable only for bulk transfers Pipes 3 to 5: Settable
PIPEBUF	BUFSIZE	FIFO buffer size	DCP: Setting disabled (fixed to 256 bytes) Pipes 1 to 5: Settable up to 2 KB Pipes 6 to 9: Setting disabled (fixed to 64 bytes)
	BUFNMB	FIFO buffer number	DCP: Setting disabled (fixed to 0h-3h area) Pipes 1 to 5: Setting disabled (8h-87h area specifiable) Pipes 6 to 9: Setting disabled (fixed to 4h-7h area)
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Device select	Viewable only in host controller mode
	MXPS	Maximum packet size	Setting compliant with USB specification
PIPEPERI	IFIS	Buffer flush	Pipes 1, 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Setting disabled
	IITV[2:0]	Interval counter	Pipes 1, 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode

Table 33.22 Pipe settings (2 of 2)

Register name	Bit name	Setting	Notes
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	CSCLR	CSSTS clear	Controllable only in host controller mode
	CSSTS	Split status check	Viewable only in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence check	Monitors the data toggle bit
	PBUSY	PIPE busy check	-
	PID[1:0]	Response PID	-
PIPEnTRE	TRENB	Transaction count enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT	Transaction counter	Pipes 1 to 5: Settable

33.3.7.1 Pipe control register switching procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID[1:0] bits are 00b (NAK response)). Figure 33.10 shows pipe control register switching procedures when USB communication is enabled (PID[1:0] bits are 00b (BUF response)).

Do not change the following registers and bits when USB communication is enabled (PID[1:0] bits are 01b (BUF response)):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN
- Bits in DEVADDm (m = 0 to A).

To set the CSCLR bits and bits in DEVADDm (m = 0 to A), follow the procedures described in section 33.2, Register Descriptions.

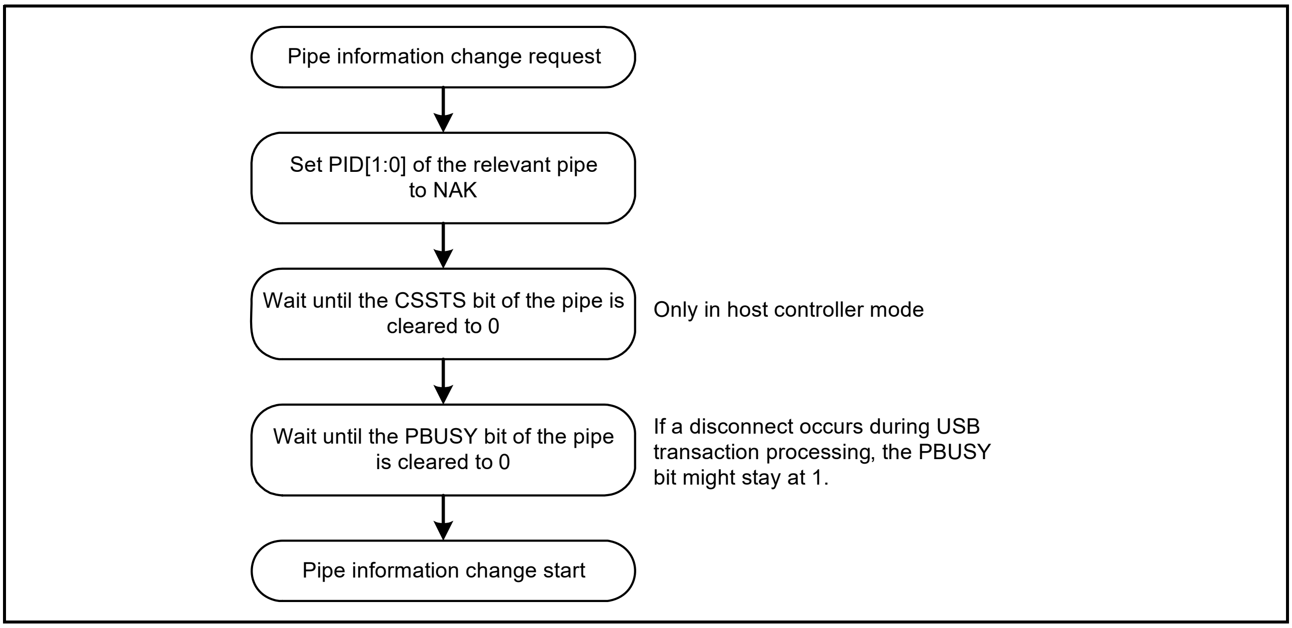


Figure 33.10 Procedure for changing pipe information when USB communication is enabled and PID[1:0] bits are 01b (BUF response)

The following bits in the pipe control registers can be changed only when the selected pipe information is not set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers while the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI
- PIPEnCTR and ACLRM bits.

To change pipe information, you must set the CURPIPE[3:0] bits to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit after the pipe information is changed.

33.3.7.2 Transfer types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk transfer or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer.

33.3.7.3 Endpoint number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique.

33.3.7.4 Maximum packet size setting

Specify the maximum packet size for each pipe in the MXPS bits in DCPMAXP and PIPEMAXP. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID[1:0] bits are set to 01b (BUF response)):

- DCP: Set to 64 for high-speed operation
- DCP: Set to 8, 16, 32, or 64 for full-speed operation
- Pipes 1 to 5: Set to 512 for high-speed bulk transfers
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for full-speed bulk transfers
- Pipes 1, 2: Set between 1 and 1024 for high-speed isochronous transfers
- Pipes 1, 2: Set between 1 and 1023 for full-speed isochronous transfers
- Pipes 6 to 9: Set between 1 and 64.

High-bandwidth interrupt transfers and isochronous transfers are not supported.

33.3.7.5 Transaction counter for pipes 1 to 5 in the receiving direction

When the specified number of transactions is complete in the data packet receiving direction, the USBHS recognizes that the transfer has ended. Two transaction counters are provided. One is the PIPEnTRN register, which specifies the number of transactions to be executed, and the other is the current counter, which internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to 00b (NAK response) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read.

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and the PIPEnCTR.PID[1:0] bits are set to 01b (BUF response), the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared.

33.3.7.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBHS operation with different response PID settings.

(1) Software response PID settings in host controller mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:
  OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
  IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.
- STALL setting: Using pipes is disabled and no transactions are executed.

Note: Use the SUREQ bit to execute setup transactions for the DCP.

(2) Software response PID settings in device controller mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions
- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions.

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

(3) and (4) describe situations in which the USBHS writes to the PID[1:0] bits because of specific transaction results.

(3) Hardware response PID settings in host controller mode

- NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, and issuing of tokens is automatically stopped:
    - When a non-isochronous transfer is performed and an NRDY interrupt is generated. For details, see section 33.3.6.2, NRDY interrupt.
    - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
    - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers.
- BUF setting: The USBHS does not write this setting
- STALL setting: The PID[1:0] bits are set to STALL in the following cases, and issuing of tokens is automatically stopped:
    - When STALL is received in response to a transmitted token
    - When a received data packet exceeds the maximum packet size.

(4) Hardware response PID settings in device controller mode

- NAK setting: The PID[1:0] bits are set to 00b (NAK response) in the following cases, and a NAK response is returned to transactions:
    - When the setup token is received normally (DCP only)
    - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers.
- BUF setting: The USBHS does not write this setting
- STALL setting: The PID[1:0] bits are set to STALL in the following cases, and a STALL response is returned to transactions:
    - When a received data packet exceeds the maximum packet size
    - When a control transfer sequence error is detected.

33.3.7.7 Data PID sequence bit

The USBHS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBHS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by software in both host and device controller modes.

33.3.7.8 Response PID = NAK function

The USBHS provides a function for disabling pipe operation (PID[1:0] bits are set to 00b (NAK response)) when the final data packet of a transaction is received. The USBHS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, software must enable the pipe again (PID[1:0] bits are set to 01b (BUF response)).

The response PID = NAK function can be used only for bulk transfers.

33.3.7.9 Auto response mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

33.3.7.10 OUT-NAK mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next, enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK). Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

33.3.7.11 Null auto response mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode while pipe operation is disabled (PID[1:0] bits are set to 00b (NAK response)). Next, enable pipe operation (PID[1:0] bits are set to 01b (BUF response)), on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (PID[1:0] bits are set to 00b (NAK response)) for the period of the zero-length packet transmission (about 10 µs) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (PID[1:0] bits are set to 01b (BUF response)).

33.3.8 FIFO Buffer

The USBHS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBHS (SIE side).

33.3.8.1 Buffer status

Table 33.23 and Table 33.24 show the buffer status in the USBHS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 1 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMA/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

Table 33.23 Buffer status indicated in the BSTS flag

ISEL or DIR	BSTS	FIFO buffer status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet is received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	Transmission is not complete. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	Transmission is complete. CPU write is allowed.

Table 33.24 Buffer status indicated in the INBUFM bit

DIR	INBUFM	FIFO buffer status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	Transmission is complete. There is no data waiting to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

33.3.8.2 FIFO buffer clearing

Table 33.25 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using the BCLR bit in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

Table 33.25 Buffer clearing methods

FIFO buffer clearing mode	Clearing the FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	• CFIFOCTR • DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid.	1: Mode valid 0: Mode invalid.

(1) Auto buffer clear mode function

The USBHS discards all received data packets if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

33.3.8.3 FIFO port functions

Table 33.26 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zero-length packet is received (DTLN[11:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[11:0] flags in the port control register.

Table 33.26 FIFO port function settings

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[11:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMA/DTC transfers (only for DnFIFO)
	MBW[1:0]	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE[3:0]	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN[11:0]	Checks the length of receive data

33.3.8.4 FIFO port selection

Table 33.27 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port selection register. After a pipe is selected, the software must check whether the written value can be correctly read from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBHS is modifying the pipe.) Next, the software checks that the FRDY flag in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW[1:0] bits in the port selection register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port selection register determines the direction.

Table 33.27 FIFO port access by pipe

Pipe	Access Method	Port that can be used
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMA/DTC access	D0FIFO/D1FIFO port register

(1) REW bit

It is possible to temporarily stop access to a pipe being accessed, access a different pipe, and then continue processing for the first pipe again. The REW bit in the port selection register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port selection register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with the REW bit set to 0, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

33.3.8.5 DMA/DTC transfers (D0FIFO and D1FIFO ports)

For pipes 1 to 9, the FIFO port can be accessed using the DMAC/DTC. When buffer access for the pipe targeted for DMA/DTC transfer is enabled, a DMA/DTC transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW[1:0] bits, and select the pipe targeted for the DMA/DTC transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMA transfer.

(1) DnFIFO auto clear mode (D0FIFO and D1FIFO port reading direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBHS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 33.28 shows the packet reception and FIFO buffer clearing processing by software for each of the different settings. As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA/DTC transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

Table 33.28 Packet reception and FIFO buffer clearing processing by software

Buffer status when packet is received	**Register setting**			
	DCLRM = 0		**DCLRM = 1**	
	BFRE = 0	**BFRE = 1**	**BFRE = 0**	**BFRE = 1**
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing required	No clearing required	No clearing required

33.3.8.6 Allocating the FIFO buffer

Figure 33.11 shows an example of a memory map of the FIFO buffer. The FIFO buffer is an area shared by the USBHS and the control CPU of the application. There are two situations for the FIFO buffer: (1) access rights are given to the application (CPU side), and (2) access rights are given to the USBHS (SIE side).

An independent area is set for the FIFO buffer for each pipe. A memory area is determined by the first block number and the number of blocks (specified in the BUFNMB[7:0] and BUFSIZE[4:0] bits in PIPEBUF), where 64 bytes is regarded as one block. When the continuous transfer mode is selected in the CNTMD bit in PIPECFG, set the BUFSIZE[4:0] bits to an integral multiple of the maximum packet size. When double buffering is selected in the DBLB bit in PIPECFG, twice the memory area specified in the BUFSIZE[4:0] bits in PIPEBUF is allocated to the same pipe.

Three FIFO ports are used to access (read data from and write data to) the FIFO buffer. Specify the number of the pipe to be allocated to the FIFO port in the CURPIPE[3:0] bits in C/DnFIFOSEL.

The FIFO buffer status of each pipe can be checked in the DCPCTR.BSTS, PIPEnCTR, and INBUFM bits. The FIFO port access rights can be checked in the FRDY flag in C/DnFIFOCTR.

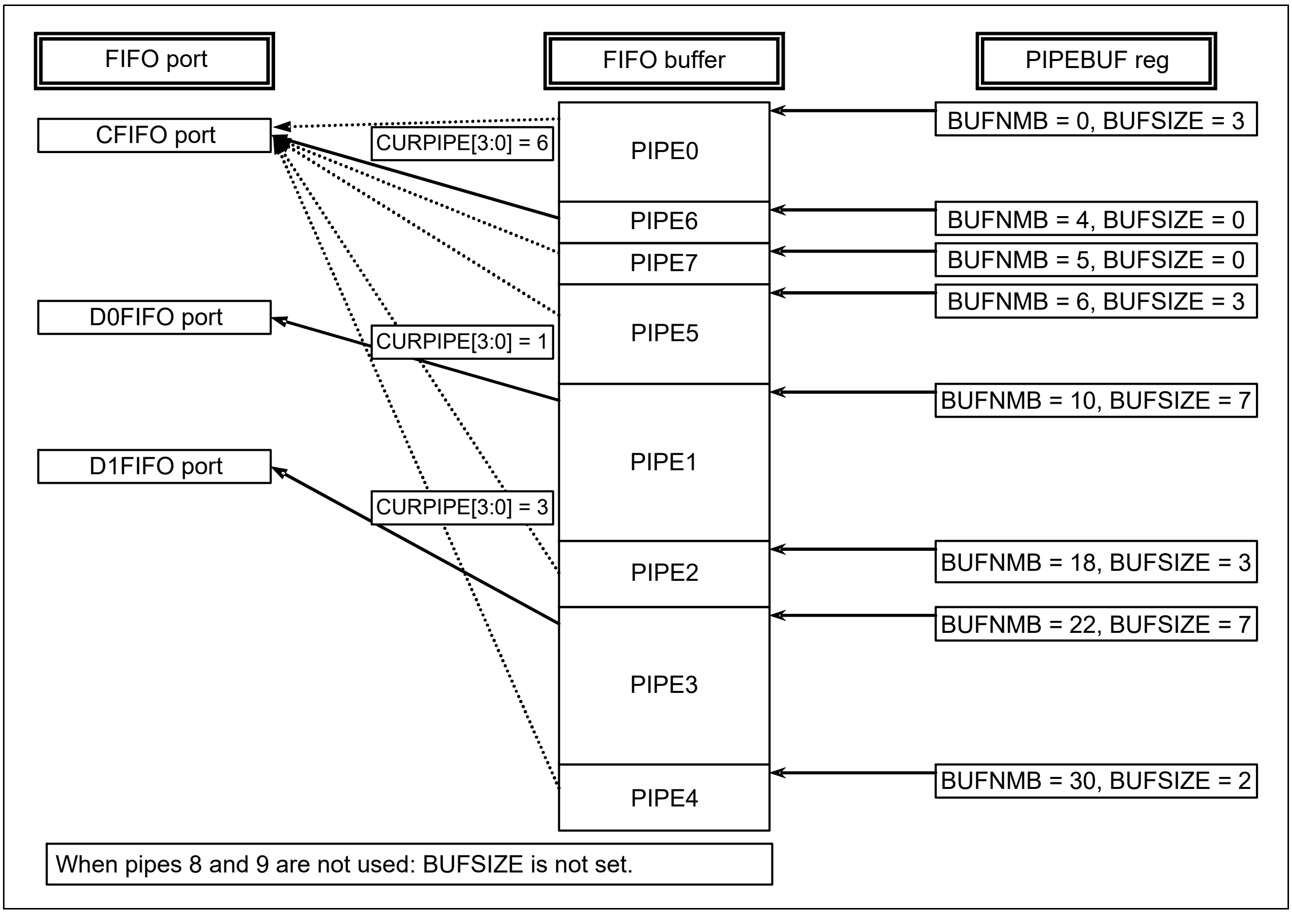


Figure 33.11 Example memory map of the FIFO buffer

33.3.9 Control Transfers Using the DCP

The Default Control Pipe (DCP) is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

33.3.9.1 Control transfers in host controller mode

(1) Setup stage

The USBREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the register and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit clears to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits cleared to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issue setup transactions using this sequence with the assigned USBAddress set in the DEVSEL[3:0] bits and the bits in DEVADDm (m = 0 to A) corresponding to the specified USBAddress set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

The DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON flag.

(2) Data stage

The data stage is used to transfer data using the DCP FIFO buffer.

Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID to DATA1 in the DCPCTR.SQSET bit and set the PID[1:0] bits to 01b (BUF response). Completion of data transfer is detected using the BRDY or BEMP interrupt.

Data transfer of multiple packets is enabled in continuous transfer mode. However, when continuous transfer is specified in the receiving direction, the BRDY interrupt is not generated unless the buffer becomes full or a short packet is received (for 256 bytes or less, which is an integral multiple of the maximum packet size). If the transmit data size is an integral multiple of the maximum packet size, control the control write transfer through the software to transmit a zero-length packet last.

(3) Status stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transfered using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[11:0] flags after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

33.3.9.2 Control transfers in device controller mode

(1) Setup stage

The USBHS returns an ACK response to a normal setup packet for the USBHS. The USBHS operates in the setup stage as follows:

On receiving a new setup packet, the USBHS sets the following bits:

- Sets the INTSTS0.VALID flag to 1
- Sets the DCPCTR.PID[1:0] bits to 00b (NAK response)
- Sets the DCPCTR.CCPL bit to 0.

When the USBHS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID flag = 1, the PID[1:0] bits cannot be set to 01b (BUF response), and the data stage cannot be terminated.

Using the VALID flag function, the USBHS can suspend a request being processed when it receives a new USB request during a control transfer and return a response to the latest request.

In addition, the USBHS automatically detects the direction bit (bmRequestType bit 8) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and no-data control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBHS, see Figure 33.9.

(2) Data stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

In high-speed control write transfers, a NYET handshake response is returned based on the FIFO buffer status.

(3) Status stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF response).

After this setting is made, the USBHS automatically executes the status stage based on the data transfer direction determined at the setup stage. The status stage is executed as follows:

- For control read transfers:
  The USBHS receives a zero-length packet from the USB host and transmits an ACK response
- For control write transfer and no data control transfer:
  The USBHS transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control transfer auto response function

The USBHS automatically responds to a normal SET_ADDRESS request. If the SET_ADDRESS request contains any of the following errors, a response must be returned by software.

- When bmRequestType is not 00h: except control write transfer
- When wIndex is not 00h: request error
- When wLength is not 00h: except no data control transfer
- When wValue is larger than 7Fh: request error
- When PL1CTRL.DVSQ[3:0] flags are 0011b (Configured): control transfer of device state error

A response by the corresponding software is required to all requests other than SET_ADDRESS.

33.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (setting of single buffer/double buffer or continuous/discontinuous transfer mode) is configurable for bulk transfers. The FIFO buffer size can be set up to 2 KB. The USBHS manages the FIFO buffer state and automatically responds to the PING packet and the NYET handshake.

33.3.10.1 PING packet control in host controller mode

In the OUT direction, a PING packet is automatically transmitted by the USBHS. The USBHS starts communication in the transmitting direction beginning with the PING packet. When it receives an ACK handshake in response to the PING packet, the USBHS transmits an OUT packet. The USBHS returns to the PING transmission state on receiving a NAK or NYET response during an OUT transaction. The procedure is as follows:

Starting OUT data transmission
→ (1) Transmit PING packet
(2) Receive NAK handshake
(3) Transmit PING packet
(4) Receive ACK handshake
(5) Transmit OUT data packet
(6) Receive ACK handshake
(7) Transmit OUT data packet
:
(8) Receive NAK/NYET handshake

The USBHS returns to the PING packet transmission state when a hardware reset is issued, the NYET or NAK handshake is received, the sequence toggle bit is cleared (SQCLR), or the buffer clear bit (ACLRM) is set.

33.3.10.2 NYET handshake control in device controller mode

Table 33.29 lists responses to received tokens during bulk and control transfers. The USBHS returns a NYET response when an available area for only one packet is left in the FIFO buffer when the USBHS has received an OUT token during a bulk or control transfer. When the USBHS receives a short packet, however, it returns an ACK response instead of NYET even when this condition occurs.

Table 33.29 Responses to received tokens

PID[1:0] bit setting	FIFO buffer state	Received token	Response	Note
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY	OUT/PING	ACK	When OUT token is received, data packet is received.*1
	RCV-BRDY	OUT	NYET	Data packet is received*2
	RCV-BRDY	OUT (Short)	ACK	Data packet is received*2
	RCV-BRDY	PING	ACK	*2
	RCV-NRDY	OUT/PING	NAK	—
	TRN-BRDY	IN	DATA0/1	Data packet is transmitted
	TRN-NRDY	IN	NAK	—

Note 1. RCV-BRDY: An available area for two packets is left in the FIFO buffer when an OUT token or a PING token is received.
Note 2. RCV-BRDY: An available area for only one packet is left in the FIFO buffer when an OUT token is received.
RCV-NRDY: No available area is left in the FIFO buffer when a PING token is received.
TRN-BRDY: The FIFO buffer contains transmit data when an IN token is received.
TRN-NRDY: The FIFO buffer contains no transmit data when an IN token is received.

33.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBHS performs interrupt transfers based on the timing dictated by the host controller. In the interrupt transfer, the USBHS ignores PING packets (no response) and does not transmit the NYET handshake, but returns an ACK, NAK, or STALL response.

In host controller mode, the software can set the timing for issuing tokens using the interval counter. The USBHS does not issue a PING token but issues an OUT token, including for transfers in the OUT direction.

The USBHS does not support high-bandwidth interrupt transfers.

33.3.11.1 Interval counter for interrupt transfers in host controller mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBHS issues interrupt transfer tokens based on this interval.

(1) Initializing the counter

The USBHS initializes the interval counter under the following conditions:

- Power-on reset:
  This initializes the IITV[2:0] bits
- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

(2) Operation when tokens cannot be transmitted or received even on token generation

No token is generated in the following cases even at token generation time. In these cases, the USBHS tries to execute the transaction in the next interval.

- When the PID[1:0] bits are set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT)

direction.

33.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBHS does not support high-bandwidth isochronous transfers but provides the following functions for isochronous transfers:

- Notification of isochronous transfer error
- Interval counter (specified in the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified in the PIPEPERI.IFIS bit)
- SOF pulse output function.

33.3.12.1 Error detection in isochronous transfers

The USBHS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by software. Table 33.30 and Table 33.31 show the priority order for errors detected by the USBHS and the associated interrupts.

(1) PID errors

- The PID value of the received packet is invalid.

(2) CRC errors and bit stuffing errors

- A CRC error is found in a received packet or the bit stuffing is illegal.

(3) Maximum packet size exceeded

- The data size of the received packet exceeds the specified maximum packet size.

(4) Overrun and underrun errors

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction.

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction.

(5) Interval error

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer.

Table 33.30 Error detection for token transmission and reception (1 of 2)

Detection priority	Error type	Interrupt generated at error detection and status
1	PID error	No interrupts are generated in either host or device controller mode. (Ignored as a corrupted packet.)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode. (Ignored as a corrupted packet.)
3	Overrun or underrun error	An NRDY interrupt is generated to set the OVRN flag to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to the OUT token.

Table 33.30 Error detection for token transmission and reception (2 of 2)

4	Interval error	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

Table 33.31 Error detection for data packet reception

Detection priority	Error type	Interrupt generated at error detection and status
1	PID error	No interrupt is generated. (Ignored as a corrupted packet.)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit sets to 1 in both host and device controller modes.
3	Maximum packet size exceeded error	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes.

33.3.12.2 DATA PID

The USBHS does not support high-bandwidth transfers. In device controller mode, the USBHS responds as follows to a received PID:

(1) IN direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mDATA: Not transmitted.

(2) OUT direction (full-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mDATA: Packets ignored.

(3) OUT direction (high-speed operation)

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Received normally as data packet PID
- mDATA: Received normally as data packet PID.

33.3.12.3 Interval counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables functions as shown in Table 33.32. In host controller mode, the USBHS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.

Table 33.32 Interval counter functions in device controller mode

Transfer direction	Function	Conditions for detection
IN	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer

The interval count is performed when an SOF is received or for complemented SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to 2IITV (μ) frames.

(1) Counter initialization in device controller mode

The USBHS initializes the interval counter under the following conditions:

- Power-on reset:
  This initializes the PIPEPERI.IITV[2:0] bits
- FIFO buffer initialization using the ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under either of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token, with the PID[1:0] bits set to 01b (BUF response)
- An SOF is received after data is received in response to an OUT token, with PID[1:0] bits set to 01b (BUF response).

The interval counter is not initialized under the following conditions:

- When the PID[1:0] bits are set to NAK or STALL
  This does not stop the interval timer. The USBHS attempts the transaction in the next interval.
- USB bus reset and USB suspension
  This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

(2) Interval counting and transfer control in host controller mode

The USBHS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USBHS issues a token for a selected pipe once every 2^{IITV} frames.

The USBHS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to 01b (BUF response) by software.

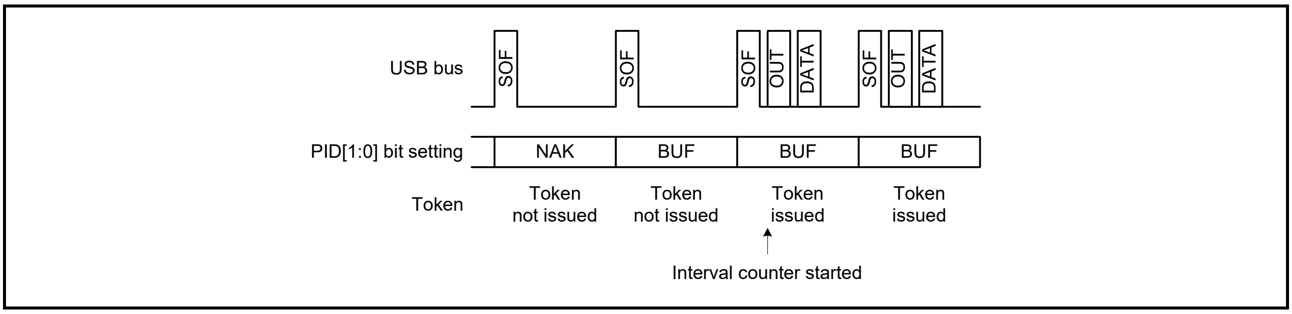


Figure 33.12 Token issuance when IITV[2:0] = 0

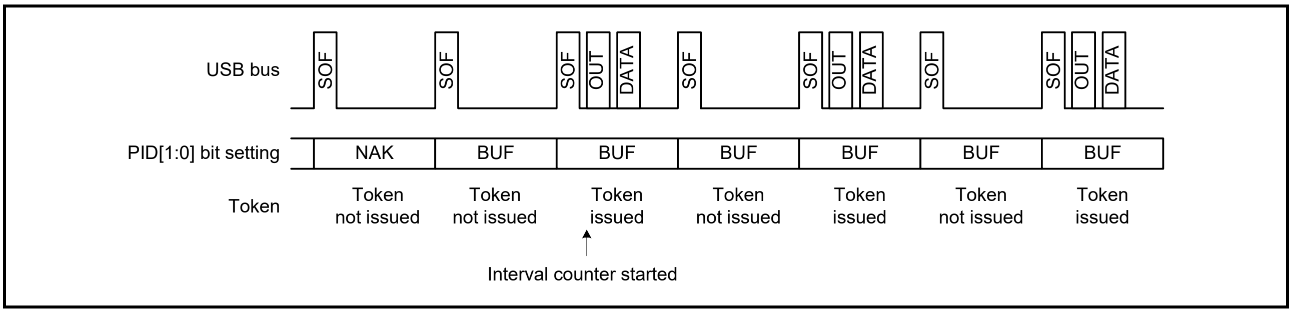


Figure 33.13 Token issuance when IITV[2:0] = 1

When the selected pipe is set for isochronous transfers, the USBHS carries out the following operation in addition to controlling the token issuance interval. The USBHS issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USBHS generates an NRDY interrupt when the USBHS issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

(b) When the selected pipe is for isochronous OUT transfers

The USBHS sets the OVRN flag to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, because the CPU or DMA/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the MCU is reset
  This initializes the IITV[2:0] bits
- When the PIPEnCTR.ACLRM bit is set to 1 by software.

(3) Interval counting and transfer control in device controller mode

(a) When the selected pipe is for isochronous OUT transfers

The USBHS generates an NRDY interrupt when it fails to receive a data packet within the interval set in the PIPEPERI.IITV[2:0] bits.

The USBHS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal complementation allows the interrupt to be generated when the SOF packet is received. However, when the IITV[2:0] bits are set to a value other than 0, the USBHS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to 00b (NAK response) by software after starting the interval timer, the USBHS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:
  The interval counting starts when the PID[1:0] bits of the selected pipe are changed to BUF
- When the IITV[2:0] bits ≠ 0:
  The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are changed to 01b (BUF response).

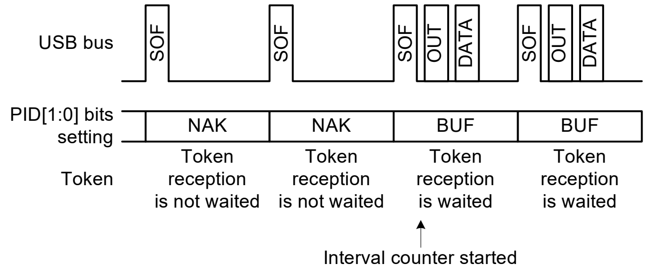


Figure 33.14 Relationship between frames and expected token reception when IITV[2:0] = 0

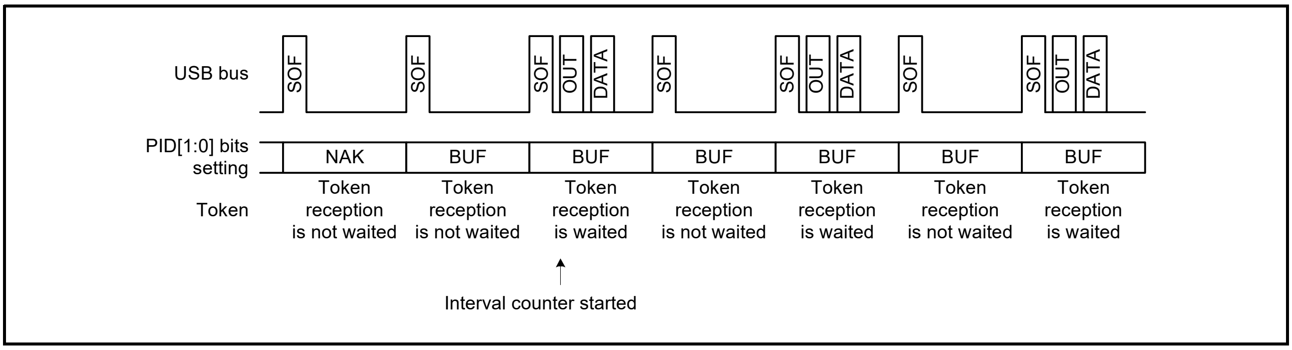


Figure 33.15 Relationship between frames and expected token reception when IITV[2:0] ≠ 0

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit must be 1 for this use case. When the IFIS bit is cleared to 0, the USBHS transmits a data packet in response to a received IN token, regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBHS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBHS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal complementation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBHS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLRM bit is set to 1 by software
- When the USBHS detects a USB bus reset.

(4) Transmit data setup for isochronous transfers in device controller mode

With isochronous data transmission using the USBHS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer in which data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 33.16 shows an example of transmission using the isochronous transfer transmit data setup function when the IITV[2:0] bits are set to 0 (every frame).

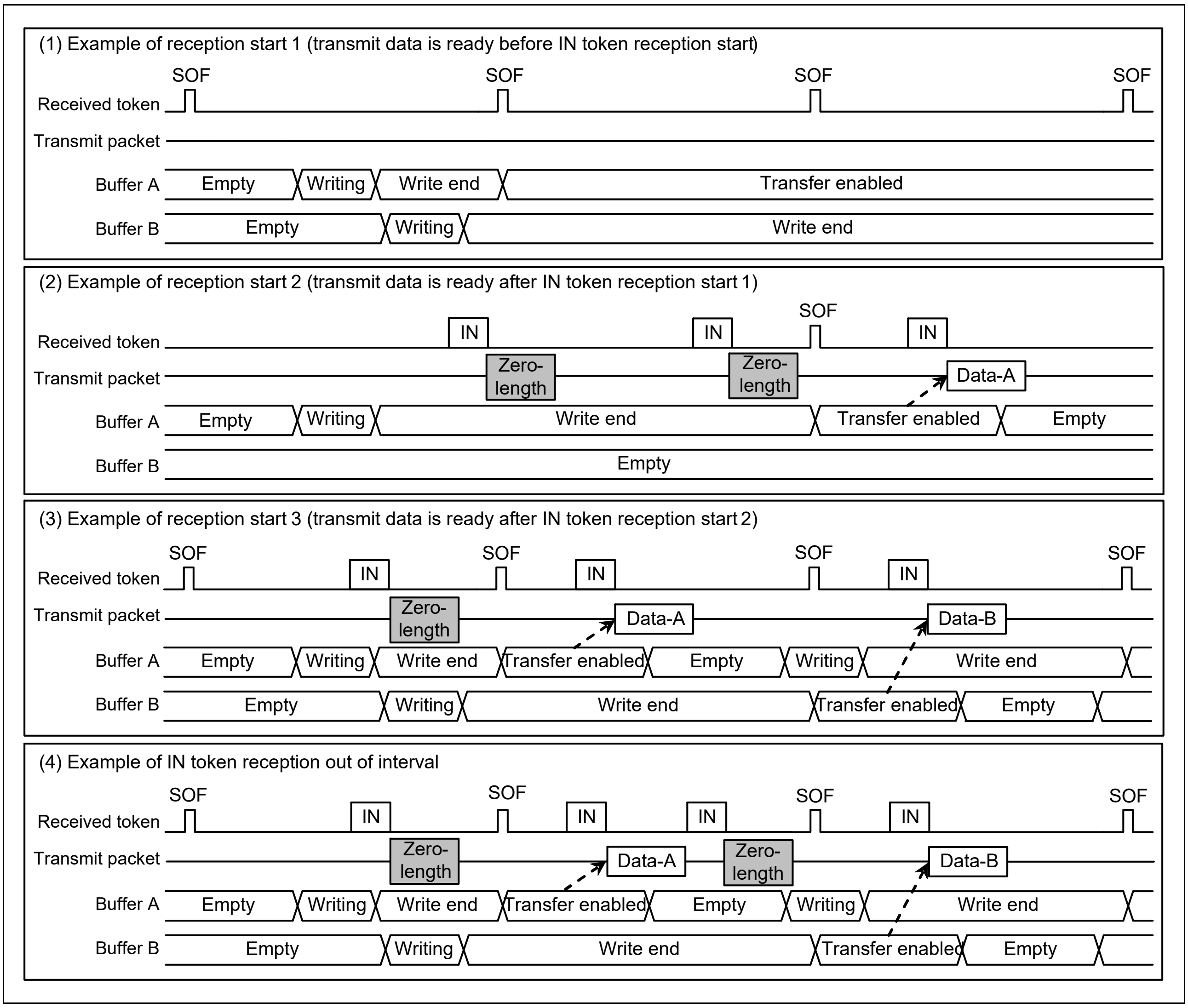


Figure 33.16 Example data setup operation

(5) Isochronous transfer transmit buffer flush in device controller mode

In device controller mode during isochronous data transmission, if the USBHS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV[2:0] = 0:
  The buffer flush operation starts from the first frame after the pipe is enabled
- When IITV[2:0] ≠ 0:
  The buffer flush operation starts after the first normal transaction.

Figure 33.17 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBHS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.

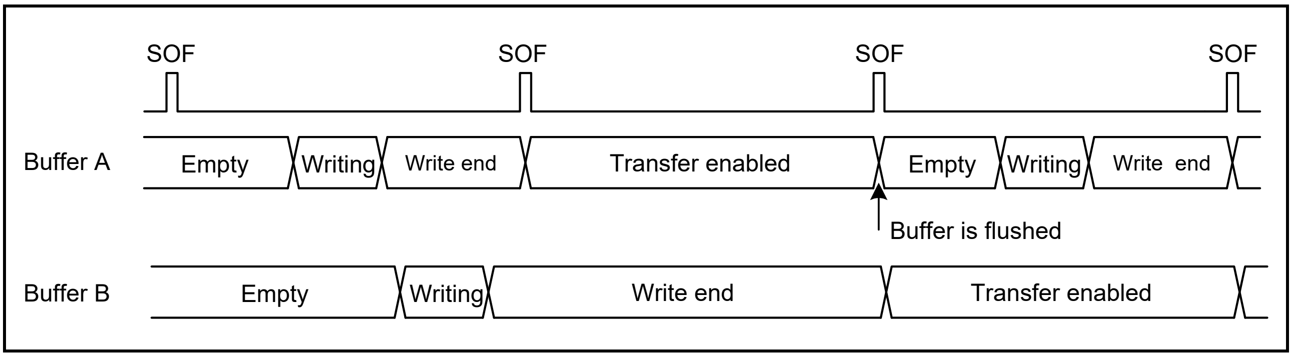


Figure 33.17 Example buffer flush operation

Figure 33.18 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing ①, and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs.
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs.

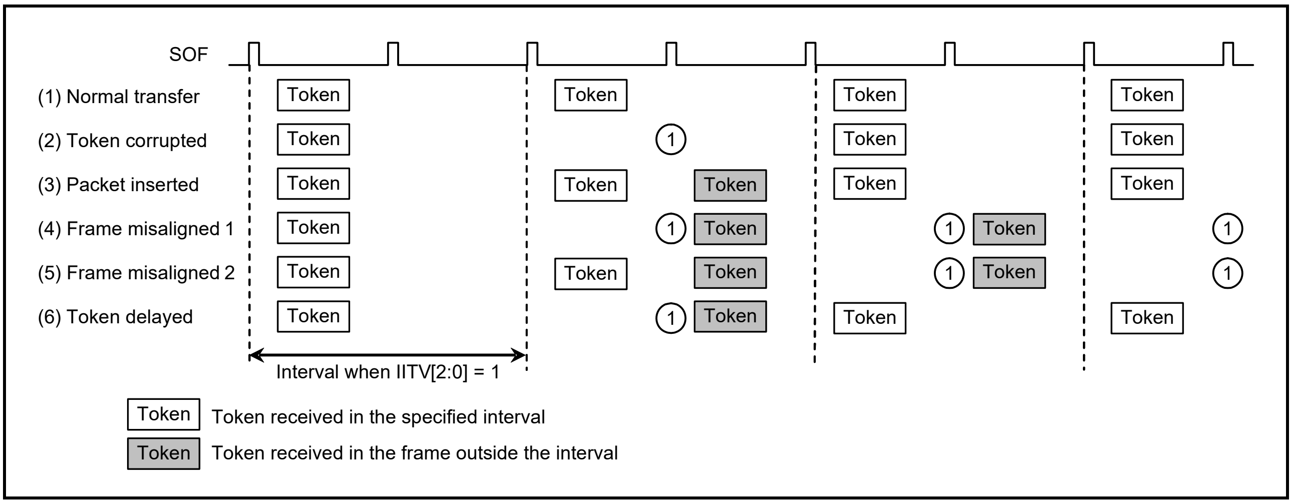


Figure 33.18 Example interval error occurrence when PIPEPERI.IITV[2:0] = 1

33.3.13 SOF Complementation Function

In device controller mode, if packet reception is disabled at intervals of 1 ms in full-speed mode or 125 µs in high-speed mode because the SOF packet is missing or corrupted, the USBHS complements the SOF. SOF complementation begins when the SYSCFG.USBE and LPSTS.SUSPENDM bits are set to 1 and an SOF packet is received. The complementation function is initialized under the following conditions:

- Power-on reset
- USB bus reset

- Suspend state detection.

The SOF complementation function operates as follows:

- The frame interval (125 µs or 1 ms) is determined by the reset handshake protocol result
- The complementation function is not activated until an SOF packet is received
- When the first SOF packet is received, complementation is performed by counting 125 µs or 1 ms on the 48-MHz internal clock
- When the second or subsequent SOF packets are received, complementation is performed at the previous reception interval
- Complementation is not performed in the Suspend state or on reception of a USB bus reset. During high-speed operation, complementation continues for 3 ms from the last packet on transition to the Suspend state.

The USBHS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF complementation if the SOF packet is missing:

- Updating of the frame number and micro frame number
- SOFR interrupt and micro-SOF lock
- SOF pulse output
- Isochronous transfer interval count.

If an SOF packet is missing during full-speed operation, the FFRMNUM.FRNM[10:0] flags are not updated. If a micro-SOF packet is missing during high-speed operation, the URMNUM.UFRNM[2:0] bits are updated.

However, if a micro-SOF packet is missing while the UFRNM[2:0] bits are set to 000b, the FRNM bits are not updated. In this case, even if a subsequent micro-SOF packet with a value other than UFRNM[2:0] bits = 000b is received successfully while UFRNM[2:0] bits are set to the value other than 000b, the FRNM bits are not updated.

33.3.14 Pipe Schedule

33.3.14.1 Conditions for generating transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBHS generates transactions under the conditions as shown in Table 33.33.

Table 33.33 Conditions for generating transactions

	Conditions for generation				
Transaction	**DIR**	**PID[1:0]**	**IITV0**	**Buffer state**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	—*1	Receive area exists	—*1
	OUT	BUF	—*1	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.

Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.

Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

33.3.14.2 Transfer schedule

This section describes the transfer scheduling within a frame of the USBHS. After the USBHS sends an SOF, the transfer is carried out in the following sequence:

1. Execution of periodic transfers:
   A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers:
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:
   A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

33.3.14.3 Enabling USB communication

Setting the DVSTCTR0.UACT bit to 1 initiates an SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission, and the Suspend state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

33.3.15 Battery charging detection processing

The USBHS provides control over the data contact detection processing (D+ line contact checking), primary detection processing (charger detection processing), and secondary detection processing (charger determination processing) as defined in the Battery Charging Specification.

This section describes operations required in device and host controller modes.

33.3.15.1 Processing in device controller mode

To operate a function device as a battery charging portable device:

1. Start primary detection processing after detecting contact with the D+ and D- lines. The Battery Charging Specification describes two processing methods for Data Contact Detection. The USBHS supports both methods as follows:
   - Software processing
     After a VBINT interrupt or polling of the VBSTS flag indicates a change in the state of the USBHS_VBUS input pin, software controls a wait from 300 to 900 ms. The BCCTRL.VDPSRCE and IDMSINKE bits are then both set to 1, enabling the VDP_SRC and IMP_SINK circuits, respectively, to start primary detection processing.
   - Hardware processing
     Apply 7 to 13 µA of current to the D+ line to hold the D+ line at the logical high level. This is done to detect the D+ and D- lines going to the logical low level because of pull-down resistors on the host device side when the D+ and D- lines come in contact with those of the host. Monitor the SYSSTS0.LNST[1:0] flags while the BCCTRL.IDPSRCE bit is set to 1, enabling the IDP_SRC circuit, to see when the level on the D+ line changes from high to low. After detecting a low level on the D+ line, clear the BCCTRL.IDPSRCE bit to 0, disabling the IDP_SRC circuit, and set both the BCCTRL.VDPSRCE and IDMSINKE bits to 1, enabling the VDP_SRC and IDM_SINK circuits, respectively, to start primary detection processing. The VDPSRCE and IDMSINKE bits must be set to 1 simultaneously.
2. After the start of primary detection processing followed by a software-controlled wait of 40 ms, check the BCCTRL.CHGDETSTS flag. A value of 1 indicates detection of a charger, and secondary detection processing starts.*1
3. To start secondary detection processing, clear both the BCCTRL.VDPSRCE and IDMSINKE bits to 0, disabling the VDP_SRC and IDM_SINK circuits, respectively. Next, set both the BCCTRL.VDMSRCE and IDPSINKE bits to 1, enabling the VDM_SRC and IDP_SINK circuits, respectively.

4. After the start of secondary detection processing followed by a software-controlled wait of 40 ms, check the BCCTRL.PDDETSTS flag. A value of 1 indicates that secondary detection processing is complete.

Note 1. In primary detection processing, detection of a voltage above the range from 0.25 to 0.4 V and below the range from 0.8 to 2.0 V on the D-Line indicates that the other device is a host device that supports battery charging (charging downstream port). The BCCTRL.CHGDETSTS flag in the PHY block only indicates whether the voltage on the D- line is higher than the range from 0.25 to 0.4 V, so add processing as required to read the SYSSTS0.LNST[1:0] flags and confirm that the voltage on the D- line is also below the range from 0.8 to 2.0 V.

Figure 33.19 illustrates this processing flow.

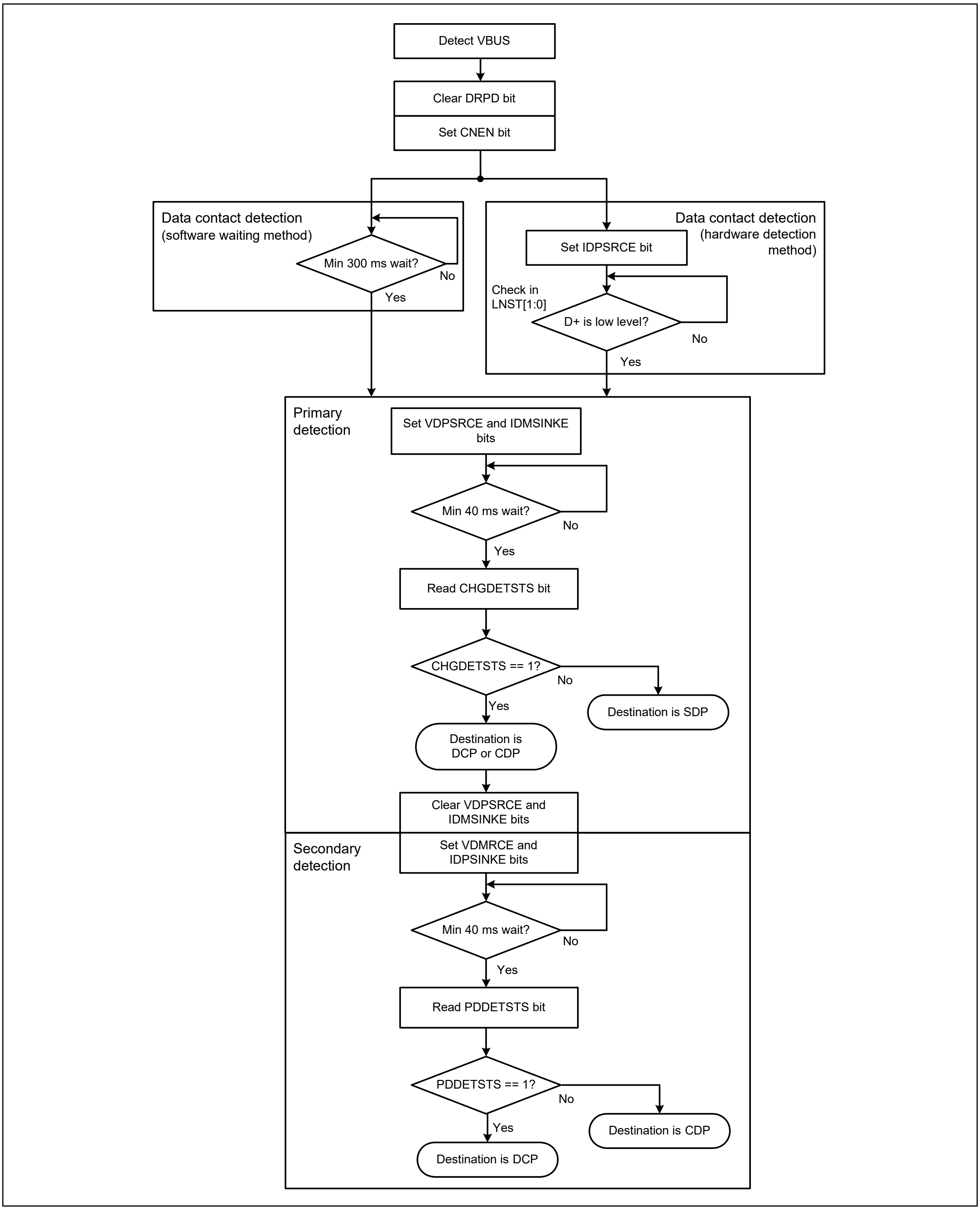


Figure 33.19 Processing flow as portable device

33.3.15.2 Processing in host controller mode

In host controller mode, driving the D- line is required for a portable device to perform primary detection. The USBHS supports the following two primary detection methods:

- When the hardware has a portable device detection function

- When the hardware does not have the function or the function is present but not used.

Figure 33.20 and Figure 33.21 show the processing flows for these methods.

(1) When the hardware has a portable device detection function

a. Start driving the USBHS_VBUS input pin.

b. Set the BCCTRL.IDMSINKE bit to 1 to enable the portable device detection circuit.

c. Monitor the portable device detection signal and start driving the D-line when the level of the portable device detection signal is high*1.

d. Stop driving the D-line when the portable device detection signal is at the low level*1.

Note 1. The PDDETINT interrupt indicates a change in the level of the portable device detection signal (EUH_CPDDET), and the current level can be obtained by reading the PDDETSTS flag.

(2) When the hardware does not have a portable device detection function or the function is not used

Software handles the timing of steps a. and b.

a. After a disconnect is detected, start driving the D-line within 200 ms.

b. After a connect is detected, stop driving the D-line within 10 ms.

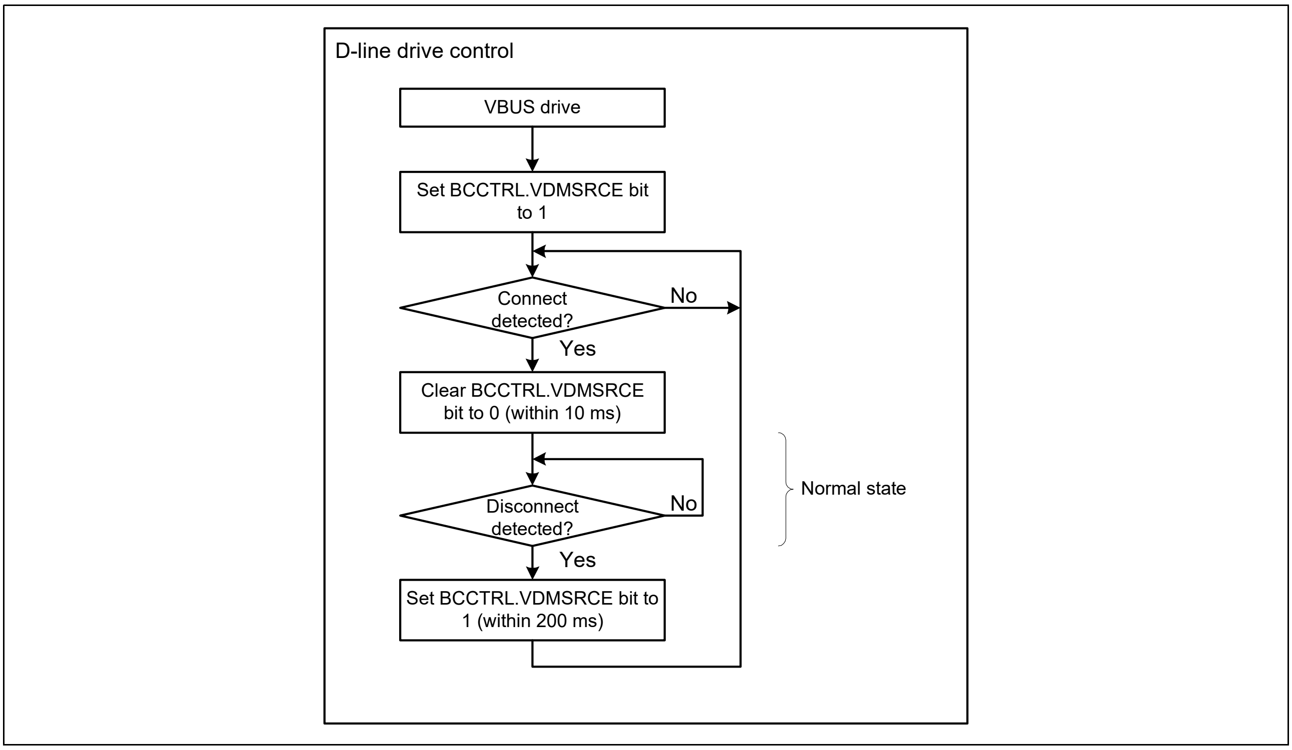


Figure 33.20 Processing flow as charging downstream port without hardware portable device detection function or when function is not used

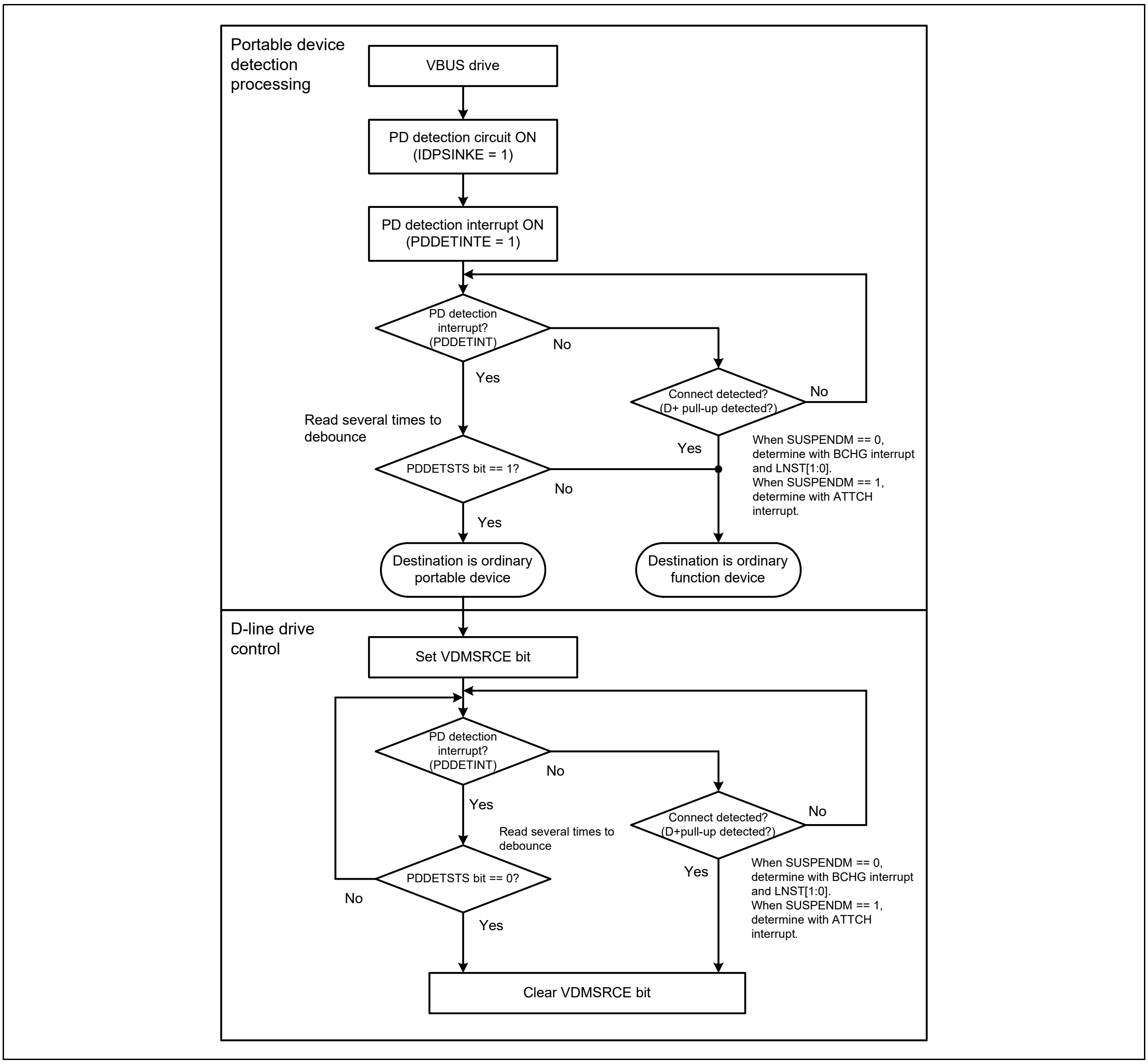


Figure 33.21 Processing flow as charging downstream port with hardware portable device detection function

33.3.16 Link Power Management Processing

The Link Power Management standard defines the existing Suspend state as the L2 state and also defines the L1 state as a state that allows transition and return with lower latency than the L2 state (Suspend). Table 33.34 provides a comparison between the L2 (Suspend) and L1 states.

Table 33.34 Comparison between L2 (Suspend) state and L1 state (1 of 2)

Parameter	L1 state	L2 (Suspend) state
Transition	LPM transaction	Idle for 3 ms
Return caused by host	Host: Minimum drive period (75 µs to 1.175 ms) can be specified by the host. Function: 10-µs K drive	Host: Minimum 20-ms K drive Function: 10-ms K drive

Table 33.34 Comparison between L2 (Suspend) state and L1 state (2 of 2)

Parameter	L1 state	L2 (Suspend) state
Return caused by function	Device: 50-μs K drive Function: 60- to 990-μs K drive Device: 10-μs K drive	Function: 1- to 15-ms K drive Host: Minimum 20-ms K drive Function: 10-ms K drive
Signaling	Low- and full-speed idle	Low- and full-speed idle

33.3.16.1 Processing in device controller mode

(1) Descriptor contents

In device controller mode, the USBHS must return its descriptor on receiving the GetDescriptor command.

Change the content of the descriptor to be returned depending on whether the transition to and return from the L1 state corresponds to the processing for the LPM transaction. The following table shows the relationship between LPM correspondence and the descriptor.

Table 33.35 Relationship between LPM correspondence and descriptor

Correspondence with LPM	bcdUSB field	USB2.0 extension descriptor		Response to received LPM request	Notes
		Provided/ not provided	Value of LPM bit		
Does not correspond	0200h	Not provided	—	No response	Normal operation when the LPM is not supported
	0201h	Provided	0	STALL	Setting for clear non-correspondence to LPM. In this case, a STALL response must be returned.
Corresponds	0201h	Provided	1	ACK or NYET	Normal operation when the LPM is supported

Declare whether to correspond to the transition to and return from L1 in the LPM bit in the USB 2.0 extension descriptor. To provide the USB2.0 extension descriptor, the bcdUSB field of the device descriptor must be set to a value of 0201h or larger.

When the LPM is not supported, the USB2.0 extension descriptor is not provided and the bcdUSB field value must be 0200h. If an LPM token is received in this case, it must be ignored. It is also possible to set the bcdUSB field value to 0201h and the LPM bit in the USB2.0 extension descriptor to 0 (LPM tokens not supported). In this case, the LPM token cannot be ignored and a STALL response must be returned.

When the LPM token is supported, set the bcdUSB field value to 0201h and set the LPM bit in the USB 2.0 extension descriptor to 1 (LPM tokens supported). This allows acknowledgment when returning a NYET or ACK response to the LPM token.

(2) Processing during LPM token reception

Transition to and return from the L1 state in device controller mode is as follows:

a. When the USBHS receives an LPM token from the host, the L1RESPEN, L1RESPMD[1:0], and L1NEGOMD settings in PL1CTRL1 determine whether a response packet is sent or the token is ignored and, if a response is to be sent, whether it is an ACK, NYET, or STALL packet.

b. If an ACK response to the LPM token is sent and the host does not transmit another LPM token in 8 μs, the USBHS enters the L1 state. The USBHS handles detection of the newly transmitted packet and the transition to the L1 state. The DVST interrupt can be used to detect the transition.

c. Two types of processing can return the USBHS from the L1 state:

- When the host drives the D-line in the K-state:
  The function device detects the K-state and starts processing the return from the L1 state in response to an

RESM interrupt request

- When the function device outputs a remote wakeup signal:
  If the software on the function device sets the DVSTCTR0.WKUP bit to 1, it sends a remote wakeup signal to the host.

The software clears the DVSTCTR0.WKUP bit to 0 on returning from the L2 (Suspend) state, and the USBHS clears the DVSTCTR0.WKUP bit to 0 for return from the L1 state.

(3) HIRD field value negotiation function

The HIRD field value included in the LPM token indicates the host K-drive period on return from the L1 state. The HIRD field value can be adjusted according to the requirements of the target system. For example, a small HIRD field value is better for systems focusing on higher transfer efficiency, while a large HIRD field value is better for systems focusing on low power consumption.

Based on the L1NEGOMD and HIRDTHR[3:0] settings in PL1CTRL1, an ACK response is returned when the received HIRD field value is in the expected range, and otherwise a NYET response is returned, requesting the host to change the HIRD field value.

Note: This HIRD field value negotiation function at the host must also support negotiation processing.

33.3.16.2 Processing in host controller mode

(1) Processing during LPM token transmission

Transition to and return from the L1 state in host controller mode is as follows:

a. When the HL1CTRL.L1REQ bit is set to 1, an LPM token is sent to the function device from the host device.

b. If an ACK response is received from the function device, a transition to the L1 state starts within 10 µs and is complete within 50 µs. If a transaction error is detected, another LPM token is transmitted within 8 µs. Retransmission can proceed up to two times. The USBHS handles all of this processing.

c. Two types of processing can return the USBHS from the L1 state:

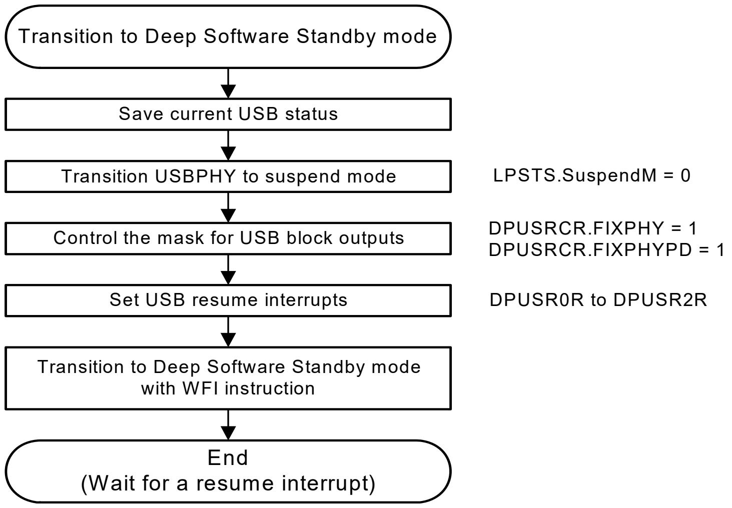
- When the host drives the D-line for the K state:
  When the DVSTCTR0.RESUME bit is set to 1, the host device starts driving the D-line for the K-state and starts processing the return
- When the function device generates a remote wakeup signal:
  When the host device detects a remote wakeup signal from the function device, it sets the DVSTCTR0.RESUME bit to 1 and starts driving the D-line for the K-state.

Unlike when returning from the Suspend (L2) state, the USBHS clears the DVSTCTR0.RESUME bit to 0. After clearing the RESUME bit, it sets the DVSTCTR0.UACT bit to 1 and issues an L1RSMEND interrupt request.

33.3.17 Release from Deep Software Standby Mode Because of USB Suspend/Resume Interrupts

Deep Software Standby mode can be canceled by a USB suspend/resume interrupt. USB suspend/resume interrupts are detected by the USB resume detecting unit, which controls and monitors the USB I/O pins to detect the interrupts.

Figure 33.22 shows the flow for setting the USBHS when entering Deep Software Standby mode from either host or device controller mode. Figure 33.23 and Figure 33.24 show the flows for setting the USBHS when canceling Deep Software Standby mode from host controller mode. Figure 33.25 shows the flow for setting the USBHS when canceling Deep Software Standby mode from device controller mode.



Note: When connected to an external device, transition to Deep Software Standby mode is enabled only on USB Suspend.

Note: Never invoke Deep Software Standby mode in the following cases:
- In an L1 Suspend status with the USB-LPM protocol.
- When remote wakeup is enabled in host controller mode.

Figure 33.22 USBHS setup flow for transition to Deep Software Standby mode as a host or device controller

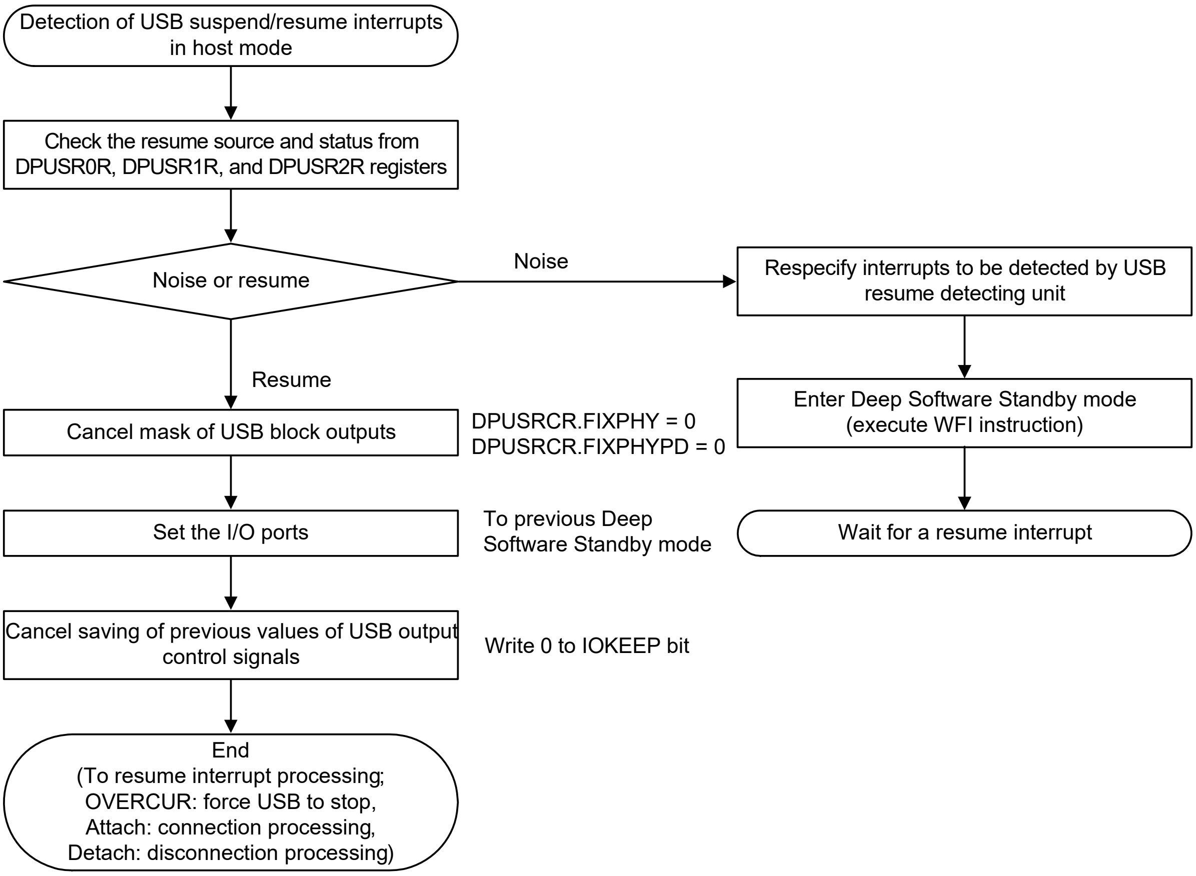


Figure 33.23 USBHS setup flow for canceling Deep Software Standby mode as a host controller (1)

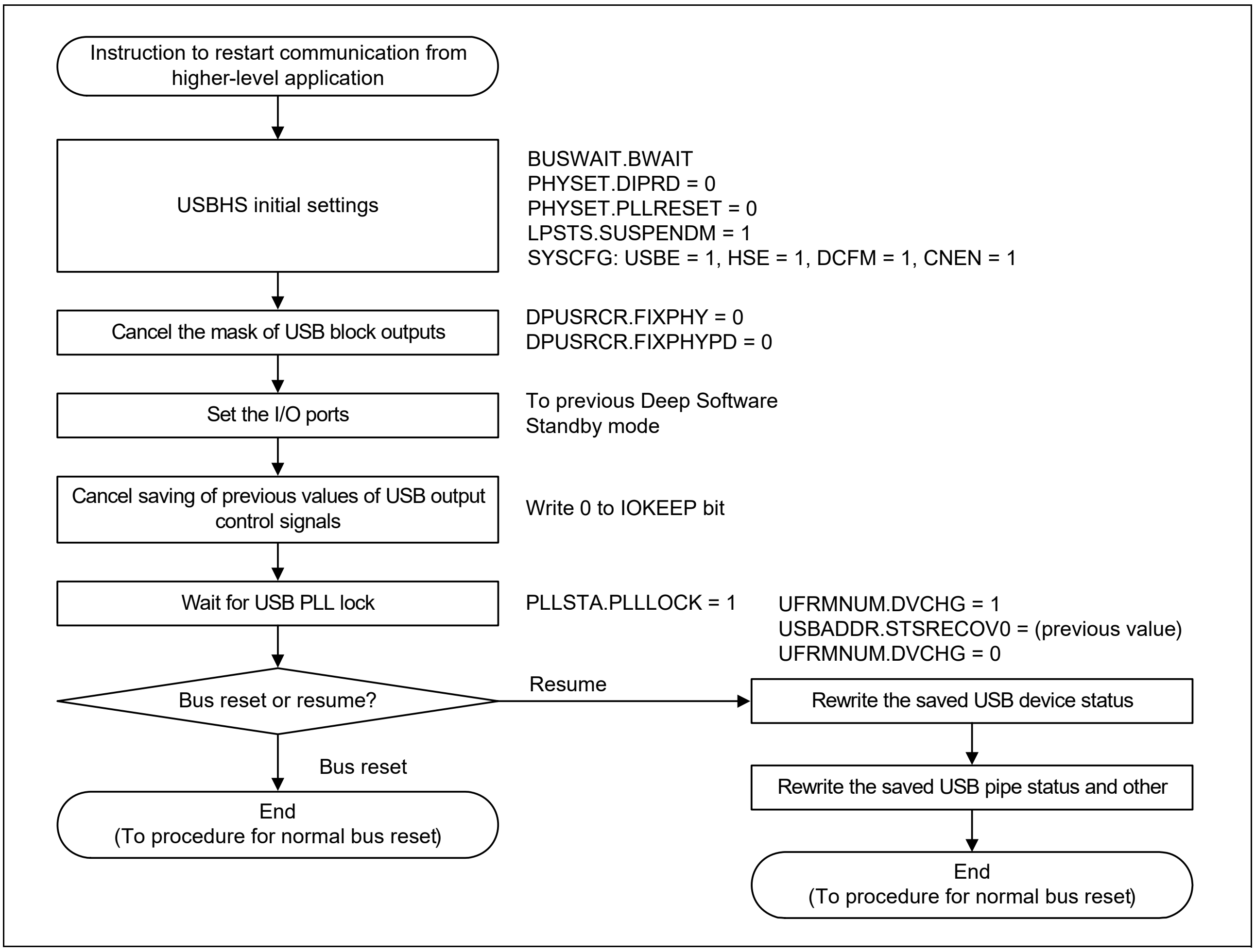
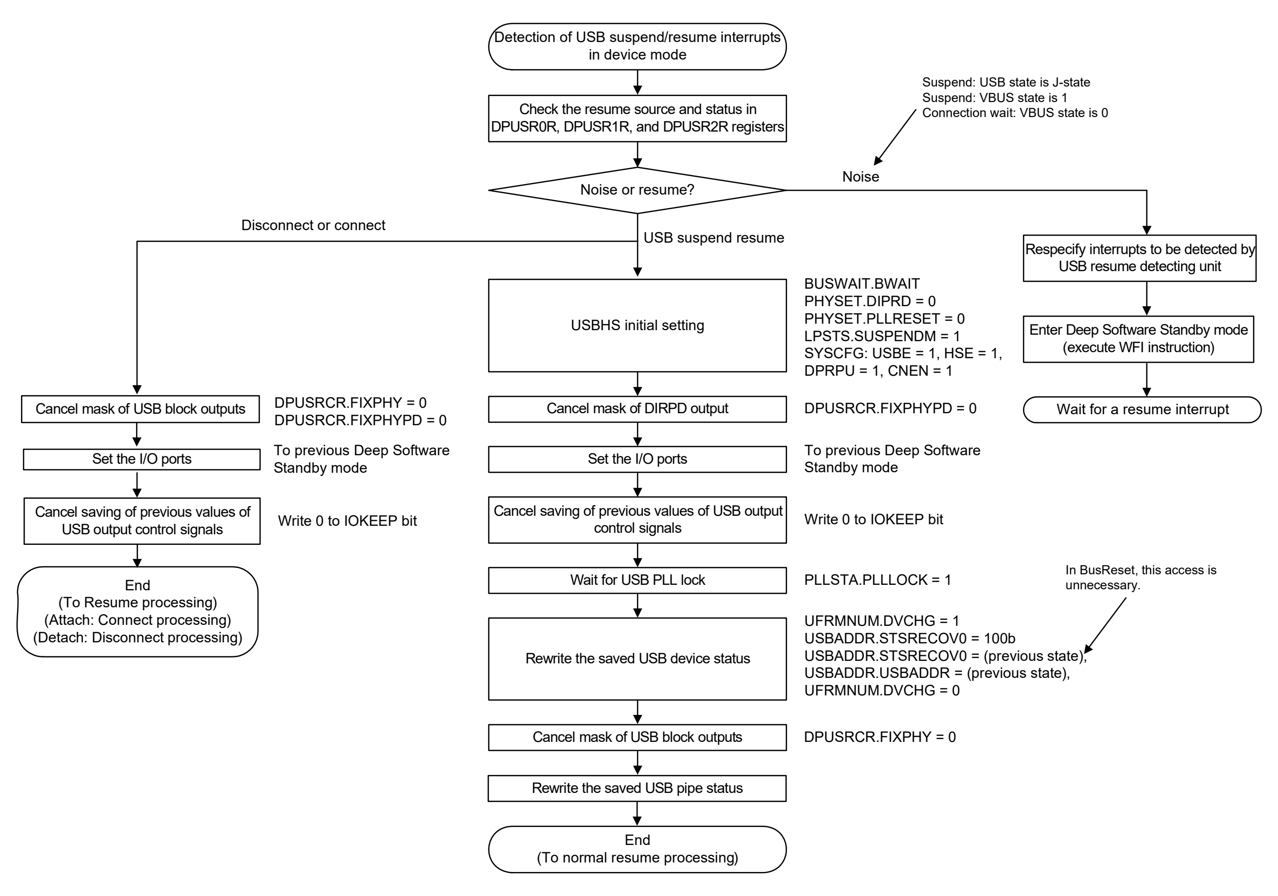


Figure 33.24 USBHS setup flow for canceling Deep Software Standby mode as a host controller (2)



Note: In BusReset return, the write to the STSRECOV0 bit must be within 2.5 µS of USBHS-PHY PLL oscillation start.

Figure 33.25 USBHS setup flow for canceling Deep Software Standby mode as a device controller (1)

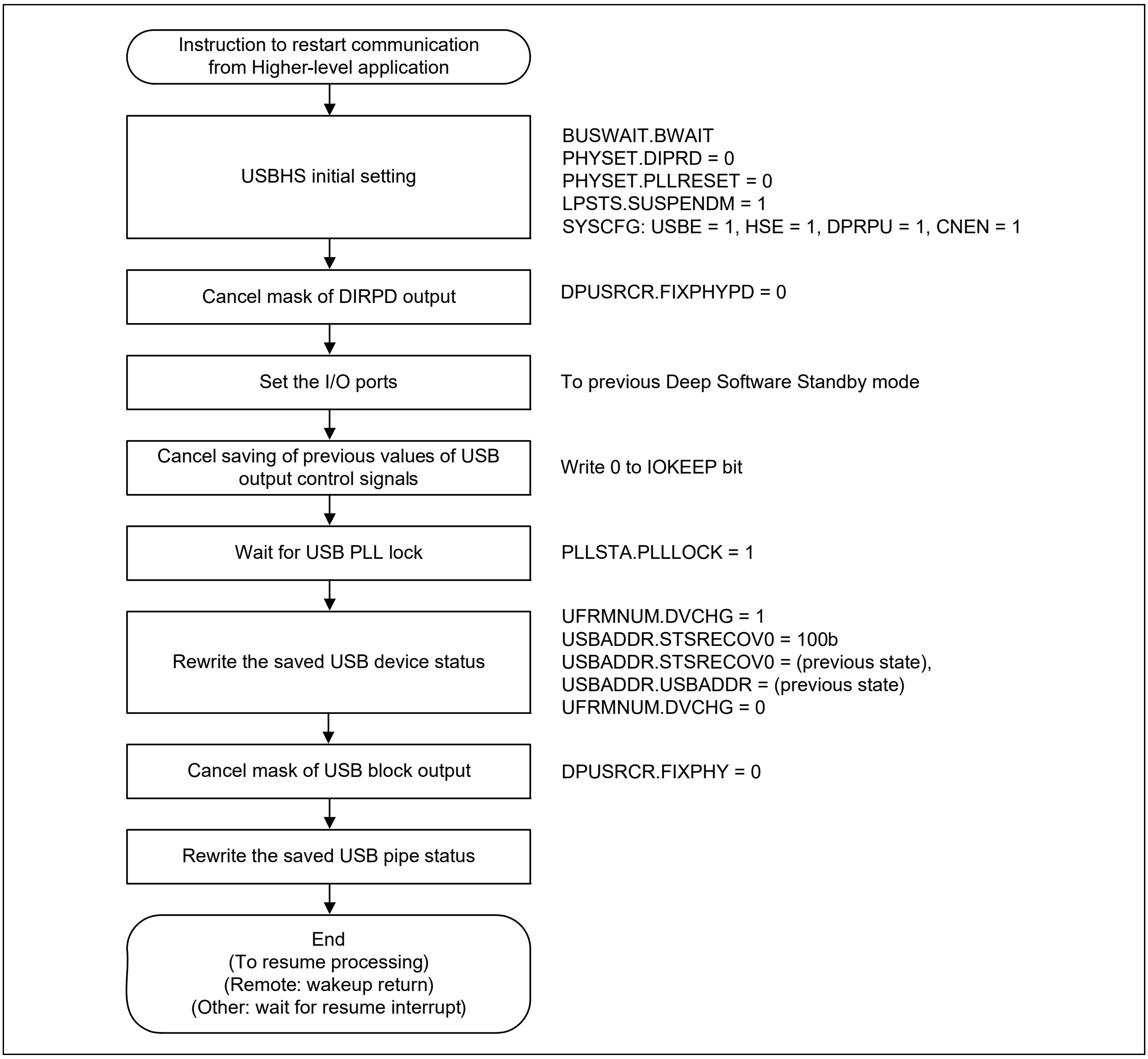


Figure 33.26 USBHS setup flow for canceling Deep Software Standby mode as a device controller (2)

33.3.18 Example External Connection Circuits

Figure 33.27 shows an example OTG connection in a self-powered system. The USBHS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBHS can use this to notify the USB host of a device disconnect.

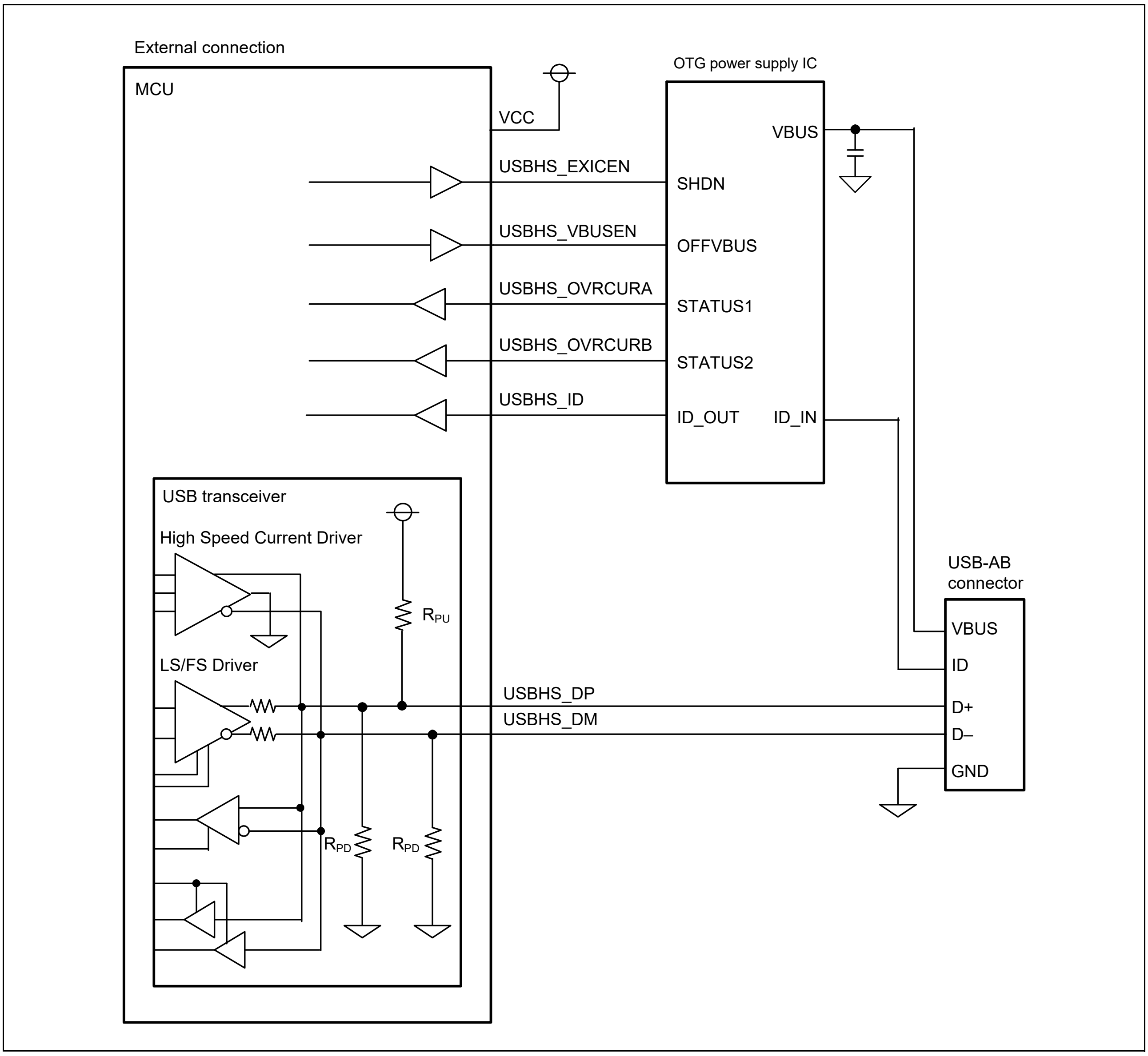
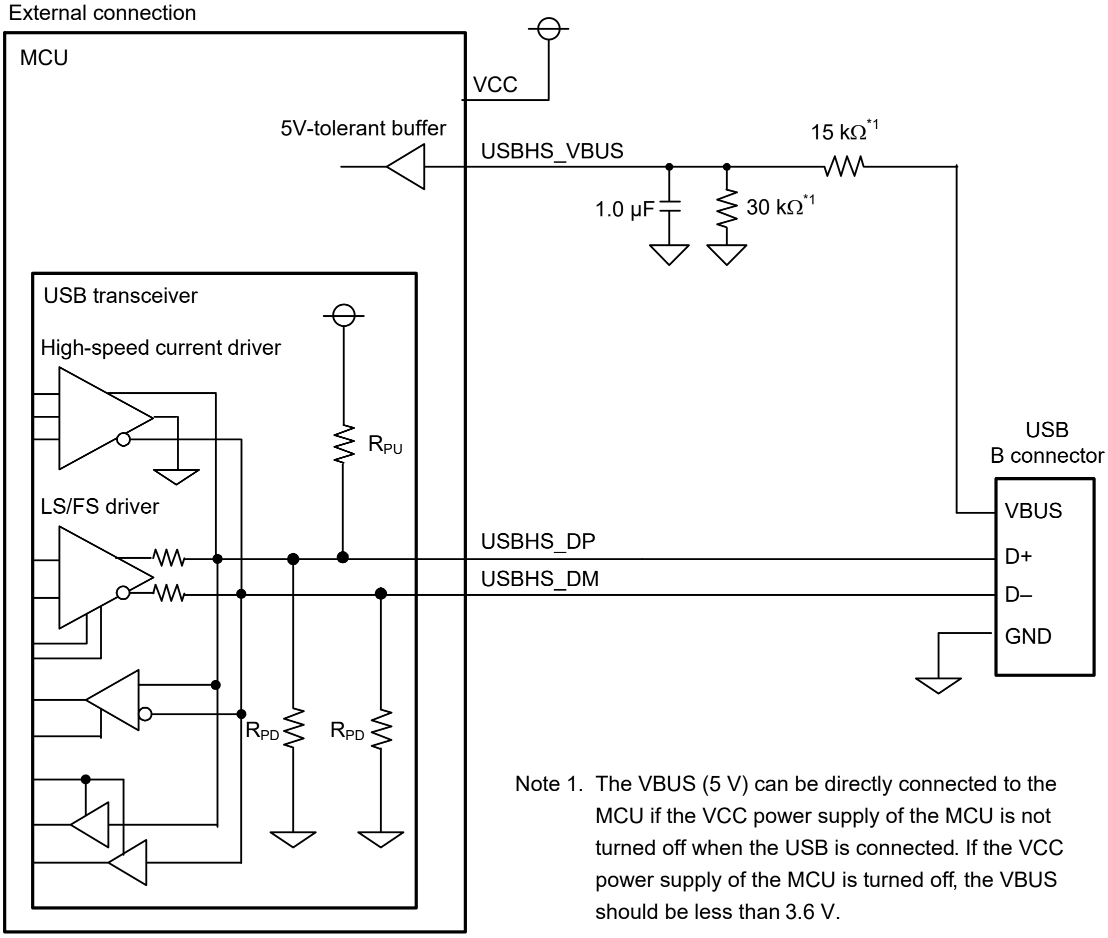


Figure 33.27 Example OTG connection in a self-powered system

Figure 33.28 shows an example USB device connection in a self-powered system.



Note 1. The VBUS (5 V) can be directly connected to the MCU if the VCC power supply of the MCU is not turned off when the USB is connected. If the VCC power supply of the MCU is turned off, the VBUS should be less than 3.6 V.

Figure 33.28 Example device connection in self-powered system

Figure 33.29 shows an example USB device connection in a bus-powered system.

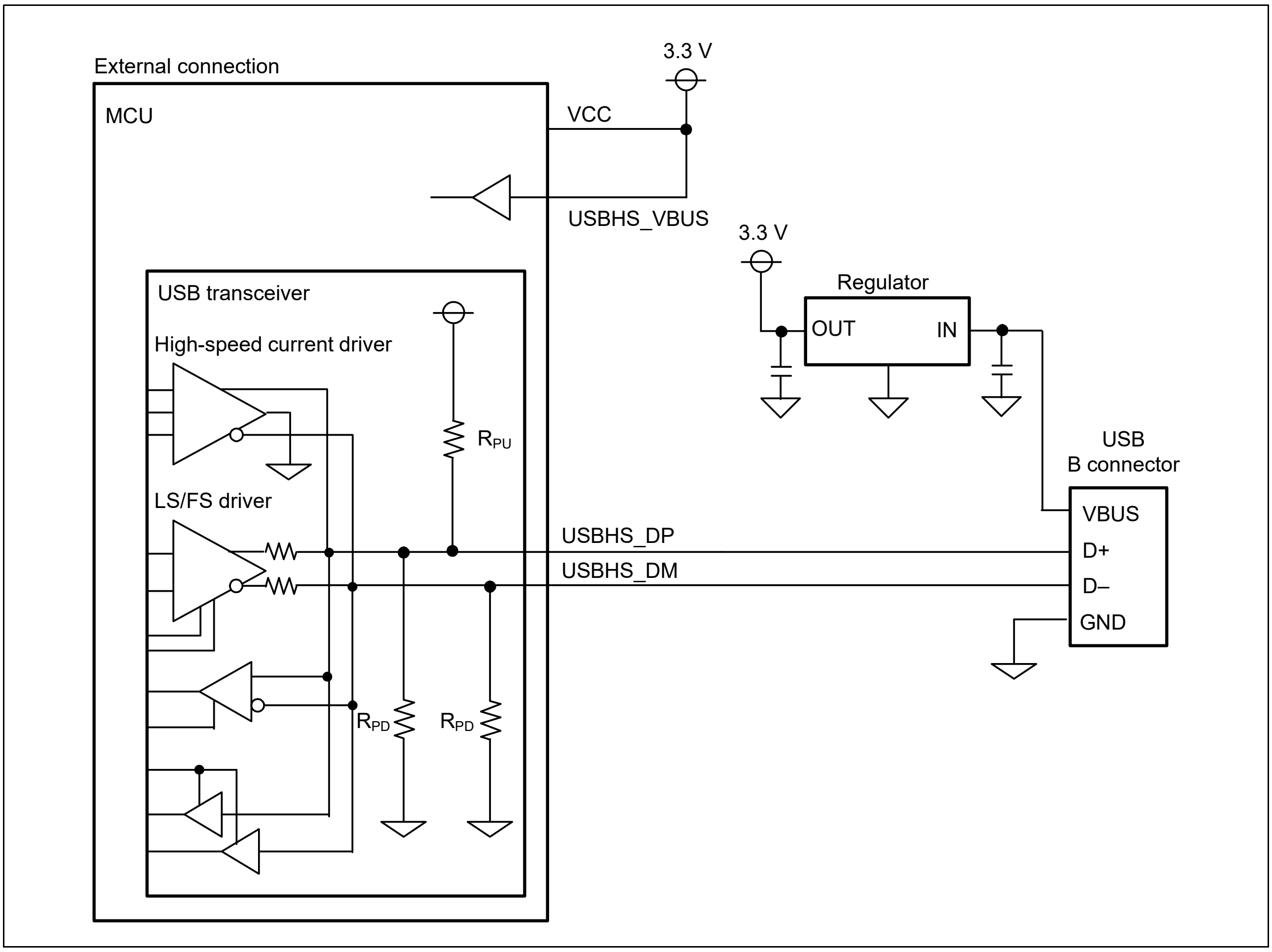


Figure 33.29 Example device connection in a bus-powered system

Figure 33.30 shows an example USB host connection.

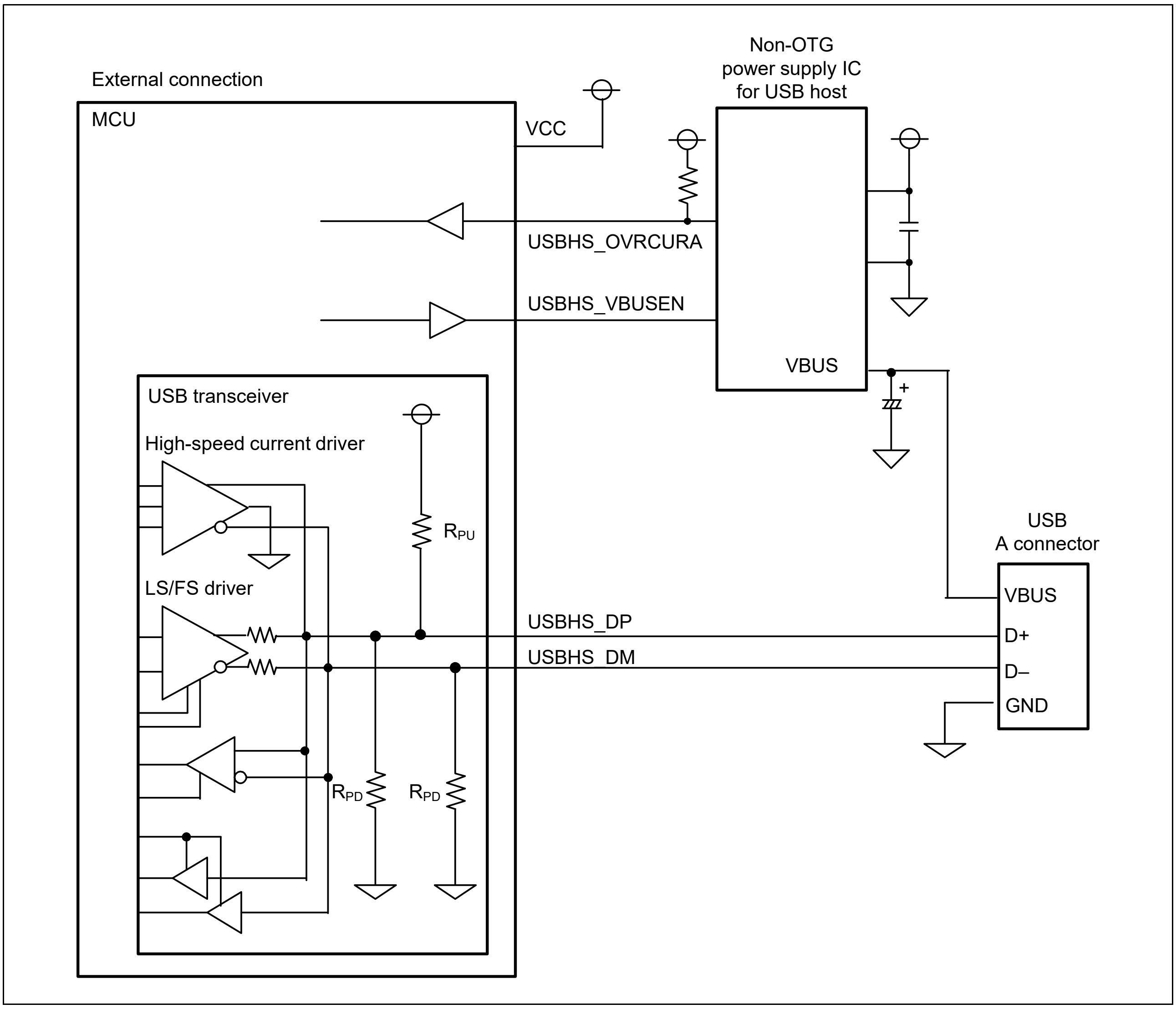


Figure 33.30 Example USB host connection

33.4 Usage Notes

33.4.1 Settings for the Module-Stop Function

USBHS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. After releasing module stop, make settings required to activate the PHY circuit, including the input system clock frequency setting, and then clear the PHYSET.DIRPD bit to 0. For details, see section 11, Low Power Modes.

33.4.2 Setup for Transitioning to Deep Software Standby Mode

Before transitioning to Deep Software Standby mode, clear the DVSTCTR0.VBUSEN bit to 0.

33.4.3 Clearing the Interrupt Status Register on Exiting Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode
- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels Software Standby is changed in Software Standby mode.

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU exits the Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

33.4.4 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR ports are set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

34. Serial Communications Interface (SCI)

34.1 Overview

The Serial Communications Interface (SCI) is configurable to five asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple IIC (master-only)
- Simple SPI
- Smart card interface.

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. Each SCI has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.

Table 34.1 lists the SCI specifications, Figure 34.1 shows a block diagram of SCI channel n, and Table 34.2 lists the I/O pins by mode.

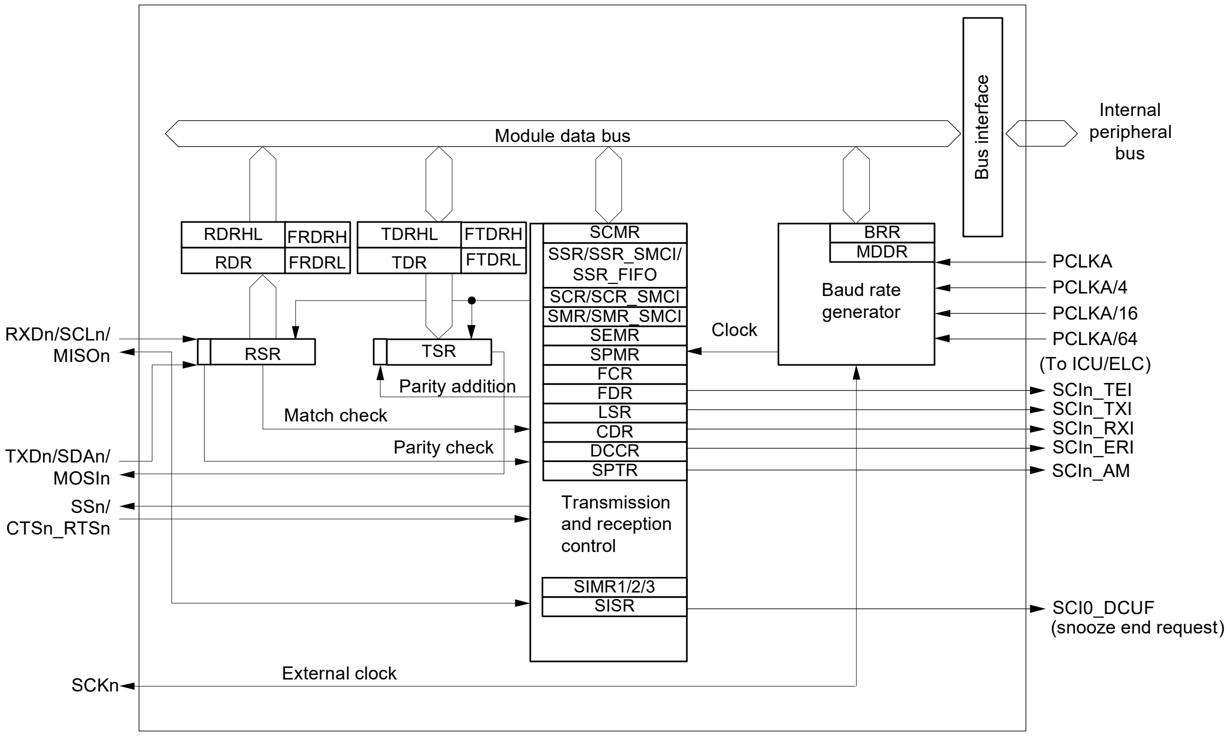
Table 34.1 SCI specifications (1 of 2)

Parameter	Specifications
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple IIC • Simple SPI.
Transfer speed	Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications	• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering
I/O pins	See Table 34.2
Data transfer	Selectable as LSB-first or MSB-first transfer
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, receive data ready, and address match Completion of generation of a start condition, restart condition, or stop condition (for simple IIC mode)
Module-stop function	Module-stop state can be set for each channel
Snooze end request	SCI0 address mismatch (SCI0_DCUF)

Table 34.1 SCI specifications (2 of 2)

Parameter		Specifications
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register
	Address non-match (SCI0 only) receive data	Snooze end request can be issued upon detecting a non-match between the received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by reading from SPTR register
	Clock source	Selectable to internal or external clock
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled among multiple processors
	Noise cancellation	Digital noise filters included on signal paths from the RXDn pin inputs
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
Smart card interface mode	Error processing	Error signal can be automatically transmitted upon detecting a parity error during reception
		Data can be automatically retransmitted upon receiving an error signal during transmission
	Data type	Both direct and inverse convention supported
Simple IIC mode	Transfer format	I^{2}C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters, and provide an adjustable interval for noise cancellation
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	SS input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high
	Clock settings	Configurable among four clock phase and clock polarity settings
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output for receive error or error signal detection (SCIn_ERI, n = 0 to 9)
		Receive data full event output (SCIn_RXI, n = 0 to 9)*1
		Transmit data empty event output (SCIn_TXI, n = 0 to 9)*1
		Transmit end event output (SCIn_TEI, n = 0 to 9)*1
		Address match event output (SCIn_AM, n = 0 to 9)

Note 1. Using this event link function is prohibited when FIFO operation is selected in asynchronous mode.



RSR: Receive Shift Register
RDR: Receive Data Register
TSR: Transmit Shift Register
TDR: Transmit Data Register
SMR/SMR_SMCI: Serial Mode Register
SCR/SCR_SMCI: Serial Control Register
SSR/SSR_SMCI/SSR_FIFO: Serial Status Register
SCMR: Smart Card Mode Register
BRR: Bit Rate Register
MDDR: Modulation Duty Register
SEMR: Serial Extended Mode Register
SPMR: SPI Mode Register
SIMR1/2/3: IIC Mode Register 1/2/3
SISR: IIC Status Register

RDRHL: Receive 9-Bit Data Register
FRDRH/L: Receive FIFO Data Register
TDRHL: Transmit 9-Bit Data Register
FTDRH/L: Transmit FIFO Data Register
FCR: FIFO Control Register
FDR: FIFO Data Count Register
LSR: Line Status Register
CDR: Compare Match Data Register
DCCR: Data Compare Match Control Register
SPTR: Serial Port Register

Figure 34.1 SCI channel n block diagram

Table 34.2 SCI I/O pins (1 of 3)

Channel	Pin name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	RXD0/SCL0/MISO0	I/O	SCI0 receive data input SCI0 IIC clock input/output SCI0 slave transmit data input/output
	TXD0/SDA0/MOSI0	I/O	SCI0 transmit data output SCI0 IIC data input/output SCI0 master transmit data input/output
	SS0/CTS0_RTS0	I/O	SCI0 chip select input, active low SCI0 transfer start control input/output, active low

Table 34.2 SCI I/O pins (2 of 3)

Channel	Pin name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1/SCL1/MISO1	I/O	SCI1 receive data input SCI1 IIC clock input/output SCI1 slave transmit data input/output
	TXD1/SDA1/MOSI1	I/O	SCI1 transmit data output SCI1 IIC data input/output SCI1 master transmit data input/output
	SS1/CTS1_RTS1	I/O	SCI1 chip select input, active low SCI1 transfer start control input/output, active low
SCI2	SCK2	I/O	SCI2 clock input/output
	RXD2/SCL2/MISO2	I/O	SCI2 receive data input SCI2 IIC clock input/output SCI2 slave transmit data input/output
	TXD2/SDA2/MOSI2	I/O	SCI2 transmit data output SCI2 IIC data input/output SCI2 master transmit data input/output
	SS2/CTS2_RTS2	I/O	SCI2 chip select input, active low SCI2 transfer start control input/output, active low
SCI3	SCK3	I/O	SCI3 clock input/output
	RXD3/SCL3/MISO3	I/O	SCI3 receive data input SCI3 IIC clock input/output SCI3 slave transmit data input/output
	TXD3/SDA3/MOSI3	I/O	SCI3 transmit data output SCI3 IIC data input/output SCI3 master transmit data input/output
	SS3/CTS3_RTS3	I/O	SCI3 chip select input, active low SCI3 transfer start control input/output, active low
SCI4	SCK4	I/O	SCI4 clock input/output
	RXD4/SCL4/MISO4	I/O	SCI4 receive data input SCI4 IIC clock input/output SCI4 slave transmit data input/output
	TXD4/SDA4/MOSI4	I/O	SCI4 transmit data output SCI4 IIC data input/output SCI4 master transmit data input/output
	SS4/CTS4_RTS4	I/O	SCI4 chip select input, active low SCI4 transfer start control input/output, active low
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5/SCL5/MISO5	I/O	SCI5 receive data input SCI5 IIC clock input/output SCI5 slave transmit data input/output
	TXD5/SDA5/MOSI5	I/O	SCI5 transmit data output SCI5 IIC data input/output SCI5 master transmit data input/output
	SS5/CTS5_RTS5	I/O	SCI5 chip select input, active low SCI5 transfer start control input/output, active low
SCI6	SCK6	I/O	SCI6 clock input/output
	RXD6/SCL6/MISO6	I/O	SCI6 receive data input SCI6 IIC clock input/output SCI6 slave transmit data input/output
	TXD6/SDA6/MOSI6	I/O	SCI6 transmit data output SCI6 IIC data input/output SCI6 master transmit data input/output
	SS6/CTS6_RTS6	I/O	SCI6 chip select input, active low SCI6 transfer start control input/output, active low

Table 34.2 SCI I/O pins (3 of 3)

Channel	Pin name	I/O	Function
SCI7	SCK7	I/O	SCI7 clock input/output
	RXD7/SCL7/MISO7	I/O	SCI7 receive data input SCI7 IIC clock input/output SCI7 slave transmit data input/output
	TXD7/SDA7/MOSI7	I/O	SCI7 transmit data output SCI7 IIC data input/output SCI7 master transmit data input/output
	SS7/CTS7_RTS7	I/O	SCI7 chip select input, active low SCI7 transfer start control input/output, active low
SCI8	SCK8	I/O	SCI8 clock input/output
	RXD8/SCL8/MISO8	I/O	SCI8 receive data input SCI8 IIC clock input/output SCI8 slave transmit data input/output
	TXD8/SDA8/MOSI8	I/O	SCI8 transmit data output SCI8 IIC data input/output SCI8 master transmit data input/output
	SS8/CTS8_RTS8	I/O	SCI8 chip select input, active low SCI8 transfer start control input/output, active low
SCI9	SCK9	I/O	SCI9 clock input/output
	RXD9/SCL9/MISO9	I/O	SCI9 receive data input SCI9 IIC clock input/output SCI9 slave transmit data input/output
	TXD9/SDA9/MOSI9	I/O	SCI9 transmit data output SCI9 IIC data input/output SCI9 master transmit data input/output
	SS9/CTS9_RTS9	I/O	SCI9 chip select input, active low SCI9 transfer start control input/output, active low

34.2 Register Descriptions

34.2.1 Receive Shift Register (RSR)

RSR is a shift register that receives serial data input from the RXDn pin and converts it into parallel data. When one frame of data is received, it is automatically transferred to the RDR register, RDRHL register, or receive FIFO. The RSR register cannot be directly accessed by the CPU.

34.2.2 Receive Data Register (RDR)

Address(es): SCI0.RDR 4007 0005h, SCI1.RDR 4007 0025h, SCI2.RDR 4007 0045h, SCI3.RDR 4007 0065h, SCI4.RDR 4007 0085h, SCI5.RDR 4007 00A5h, SCI6.RDR 4007 00C5h, SCI7.RDR 4007 00E5h, SCI8.RDR 4007 0105h, SCI9.RDR 4007 0125h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

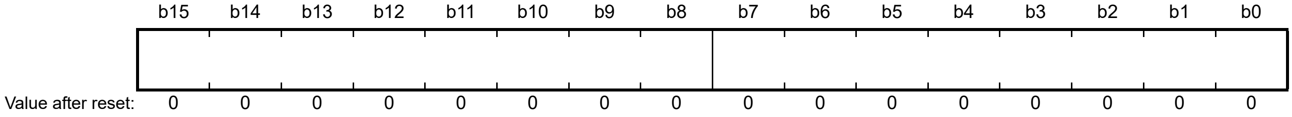
RDR is an 8-bit register that stores receive data. When one frame of serial data is received, it is transferred from the RSR register to the RDR register, and the RSR register can receive more data. Because RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.

Read the RDR register only once after a receive data full interrupt (SCIn_RXI) occurs.

Note: If the next frame of data is received before the receive data is read from the RDR register, an overrun error occurs. The RDR register cannot be written to by the CPU.

34.2.3 Receive 9-Bit Data Register (RDRHL)

Address(es): SCI0.RDRHL 4007 0010h, SCI1.RDRHL 4007 0030h, SCI2.RDRHL 4007 0050h, SCI3.RDRHL 4007 0070h, SCI4.RDRHL 4007 0090h, SCI5.RDRHL 4007 00B0h, SCI6.RDRHL 4007 00D0h, SCI7.RDRHL 4007 00F0h, SCI8.RDRHL 4007 0110h, SCI9.RDRHL 4007 0130h



RDRHL is a 16-bit register that stores receive data. Use the RDRHL register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of RDRHL are a shadow register of RDR, so access to the RDRHL register affects the RDR register. Access to the RDRHL register is prohibited if 7-bit or 8-bit data length is selected.

After one frame of data is received, the received data is transferred from the RSR register to the RDRHL register, allowing the RSR register to receive more data.

The RSR and RDRHL registers have a double-buffered construction to enable continuous reception. The RDRHL register must be read only when a receive data full interrupt (SCIn_RXI) request is issued. An overrun error occurs when the next frame of data is received before the received data is read from the RDRHL register.

The CPU cannot write to the RDRHL register. Bits [15:9] are fixed to 0. These bits are read as 0. The write value should be 0.

34.2.4 Receive FIFO Data Register H, L, HL (FRDRH, FRDRL, FRDRHL)

Receive FIFO Data Register H (FRDRH)

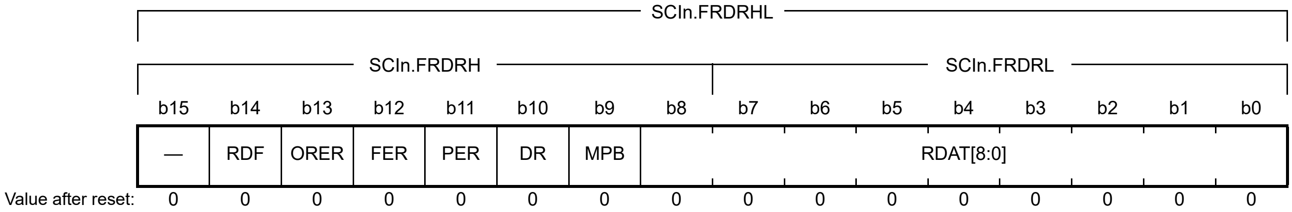
Address(es): SCI0.FRDRH 4007 0010h, SCI1.FRDRH 4007 0030h, SCI2.FRDRH 4007 0050h, SCI3.FRDRH 4007 0070h, SCI4.FRDRH 4007 0090h, SCI5.FRDRH 4007 00B0h, SCI6.FRDRH 4007 00D0h, SCI7.FRDRH 4007 00F0h, SCI8.FRDRH 4007 0110h, SCI9.FRDRH 4007 0130h

Receive FIFO Data Register L (FRDRL)

Address(es): SCI0.FRDRL 4007 0011h, SCI1.FRDRL 4007 0031h, SCI2.FRDRL 4007 0051h, SCI3.FRDRL 4007 0071h, SCI4.FRDRL 4007 0091h, SCI5.FRDRL 4007 00B1h, SCI6.FRDRL 4007 00D1h, SCI7.FRDRL 4007 00F1h, SCI8.FRDRL 4007 0111h, SCI9.FRDRL 4007 0131h

Receive FIFO Data Register HL (FRDRHL)

Address(es): SCI0.FRDRHL 4007 0010h, SCI1.FRDRHL 4007 0030h, SCI2.FRDRHL 4007 0050h, SCI3.FRDRHL 4007 0070h, SCI4.FRDRHL 4007 0090h, SCI5.FRDRHL 4007 00B0h, SCI6.FRDRHL 4007 00D0h, SCI7.FRDRHL 4007 00F0h, SCI8.FRDRHL 4007 0110h, SCI9.FRDRHL 4007 0130h



Bit	Symbol	Bit name	Description	R/W
b8 to b0	RDAT[8:0]	Serial Receive Data	Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected. Stores the serial receive data.	R
b9	MPB	Multi-Processor Bit Flag	Stores the value of the multi-processor bit in the serial receive data (RDAT[8:0]): 0: Data transmission cycle 1: ID transmission cycle. Valid only in asynchronous mode with SMR.MP = 1, and with FIFO selected.	R

Bit	Symbol	Bit name	Description	R/W
b10	DR	Receive Data Ready Flag	This flag is the same as SSR_FIFO.DR: 0: Receiving is in progress, or no received data remains in the FRDRH and FRDRL registers after successfully completed reception 1: Next receive data is not received for a period after successfully completed reception.	R*1
b11	PER	Parity Error Flag	0: No parity error occurred in the first data of FRDRH and FRDRL 1: Parity error occurred in the first data of FRDRH and FRDRL.	R
b12	FER	Framing Error Flag	0: No framing error occurred in the first data of FRDRH and FRDRL 1: Framing error occurred in the first data of FRDRH and FRDRL.	R
b13	ORER	Overrun Error Flag	This flag is the same as SSR_FIFO.ORER: 0: No overrun error occurred 1: Overrun error occurred.	R*1
b14	RDF	Receive FIFO Data Full Flag	This flag is the same as SSR_FIFO.RDF: 0: The amount of receive data written in FRDRH and FRDRL is less than the specified receive triggering number 1: The amount of receive data written in FRDRH and FRDRL is equal to or greater than the specified receive triggering number.	R*1
b15	—	Reserved	This bit is read as 0.	R

Note 1. If this flag is read, it indicates the same value as that read from the SSR_FIFO register. Write 0 to the SSR_FIFO register to clear the flag.

FRDRHL is a 16-bit register that consists of the 8-bit FRDRH and FRDRL registers.

FRDRH and FRDRL constitute a 16-stage FIFO register that stores serial receive data and related status information readable by software. This register is only valid in asynchronous mode, including multi-processor mode, or clock synchronous mode.

The SCI completes reception of one frame of serial data by transferring the received data from the Receive Shift Register (RSR) into FRDRH and FRDRL for storage. Continuous reception is executed until 16 stages are stored. If data is read when there is no received data in FRDRH and FRDRL, the value is undefined. When FRDRH and FRDRL are full, subsequent serial receive data is lost. The CPU can read from the FRDRH and FRDRL registers but cannot write to them.

Reading 1 from the RDF, ORER, or DR flags of the FRDRH register is the same as reading from those bits in the SSR_FIFO register. When writing 0 to clear a flag in the SSR_FIFO register after reading the FRDRH register, write 0 only to the flag that is to be cleared and write 1 to the other flags.

When reading both the FRDRH and FRDRL registers, read in order from FRDRH to FRDRL. The FRDRHL register can be accessed in 16-bit units.

34.2.5 Transmit Data Register (TDR)

Address(es): SCI0.TDR 4007 0003h, SCI1.TDR 4007 0023h, SCI2.TDR 4007 0043h, SCI3.TDR 4007 0063h,
SCI4.TDR 4007 0083h, SCI5.TDR 4007 00A3h, SCI6.TDR 4007 00C3h, SCI7.TDR 4007 00E3h,
SCI8.TDR 4007 0103h, SCI9.TDR 4007 0123h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TDR is an 8-bit register that stores transmit data.

When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.

The double-buffered structure of the TDR and TSR registers enables continuous serial transmission. If the next transmit data is already written to TDR when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.

The CPU can read from or write to TDR at any time. Only write transmit data to TDR once after each instance of the transmit data empty interrupt (SCIn_TXI).

34.2.6 Transmit 9-Bit Data Register (TDRHL)

Address(es): SCI0.TDRHL 4007 000Eh, SCI1.TDRHL 4007 002Eh, SCI2.TDRHL 4007 004Eh, SCI3.TDRHL 4007 006Eh,
SCI4.TDRHL 4007 008Eh, SCI5.TDRHL 4007 00AEh, SCI6.TDRHL 4007 00CEh, SCI7.TDRHL 4007 00EEh,
SCI8.TDRHL 4007 010Eh, SCI9.TDRHL 4007 012Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

TDRHL is a 16-bit register that stores transmit data. Use the TDRHL register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of TDRHL are a shadow register of TDR, so access to TDRHL affects the TDR register. Access to the TDRHL register is prohibited if 7-bit or 8-bit data length is selected.

When empty space is detected in the TSR register, the transmit data stored in the TDRHL register is transferred to the TSR register and transmission is started.

The TSR and TDRHL registers have a double-buffered structure to support continuous transmission. When the next data to be transmitted is stored in TDRHL after one frame of data is transmitted, the transmitting operation is continued by transferring the data from the TDRHL register to the TSR register.

The CPU can read from and write to the TDRHL register. Bits [15:9] in the TDRHL register are fixed to 1. These bits are read as 1. The write value should be 1.

Write transmit data to the TDRHL register only once when a transmit data empty interrupt (SCIn_TXI) request is issued.

34.2.7 Transmit FIFO Data Register H, L, HL (FTDRH, FTDRL, FTDRHL)

Transmit FIFO Data Register H (FTDRH)

Address(es): SCI0.FTDRH 4007 000Eh, SCI1.FTDRH 4007 002Eh, SCI2.FTDRH 4007 004Eh, SCI3.FTDRH 4007 006Eh, SCI4.FTDRH 4007 008Eh, SCI5.FTDRH 4007 00AEh, SCI6.FTDRH 4007 00CEh, SCI7.FTDRH 4007 00EEh, SCI8.FTDRH 4007 010Eh, SCI9.FTDRH 4007 012Eh

Transmit FIFO Data Register L (FTDRL)

Address(es): SCI0.FTDRL 4007 000Fh, SCI1.FTDRL 4007 002Fh, SCI2.FTDRL 4007 004Fh, SCI3.FTDRL 4007 006Fh, SCI4.FTDRL 4007 008Fh, SCI5.FTDRL 4007 00AFh, SCI6.FTDRL 4007 00CFh, SCI7.FTDRL 4007 00EFh, SCI8.FTDRL 4007 010Fh, SCI9.FTDRL 4007 012Fh

Transmit FIFO Data Register HL (FTDRHL)

Address(es): SCI0.FTDRHL 4007 000Eh, SCI1.FTDRHL 4007 002Eh, SCI2.FTDRHL 4007 004Eh, SCI3.FTDRHL 4007 006Eh, SCI4.FTDRHL 4007 008Eh, SCI5.FTDRHL 4007 00AEh, SCI6.FTDRHL 4007 00CEh, SCI7.FTDRHL 4007 00EEh, SCI8.FTDRHL 4007 010Eh, SCI9.FTDRHL 4007 012Eh

	SCIn.FTDRHL															
	SCIn.FTDRH								SCIn.FTDRL							
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b8 to b0	TDAT[8:0]	Serial Transmit Data	Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected. Specifies the serial transmit data.	W
b9	MPBT	Multi-Processor Transfer Bit Flag	Specifies the multi-processor bit in the transmission frame: 0: Data transmission cycle 1: ID transmission cycle. Valid only in asynchronous mode and SMR.MP = 1, and with FIFO selected.	W
b15 to b10	—	Reserved	The write value should be 1.	W

FTDRHL is a 16-bit register that consists of the 8-bit FTDRH and FTDRL registers.

FTDRH and FTDRL constitute a 16-stage FIFO register that stores data for serial transmission and a multi-processor transfer bit. This register is only valid in asynchronous mode, including multi-processor mode, or clock synchronous mode.

When the SCI detects that the Transmit Shift Register (TSR) is empty, it transfers data written in the FTDRH and FTDRL registers to the TSR register and starts serial transmission. Continuous serial transmission is executed until no transmit data is left in FTDRH and FTDRL. When FTDRHL is full of transmit data, no more data can be written. If writing new data is attempted, the data is ignored. The CPU can write to the FTDRH and FTDRL registers but cannot read them.

When writing to both the FTDRH and FTDRL registers, write in order from FTDRH to FTDRL.

MPBT flag (Multi-Processor Transfer Bit Flag)

The MPBT flag specifies the value of the multi-processor bit of the transmit frame. When FCR.FM = 1, SSR.MPBT is invalid.

34.2.8 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first automatically transfers transmit data from TDR, TDRHL, or transmit FIFO to the TSR register, and then sends the data to the TXDn pin. The CPU cannot directly access the TSR register.

34.2.9 Serial Mode Register (SMR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SMR 4007 0000h, SCI1.SMR 4007 0020h, SCI2.SMR 4007 0040h, SCI3.SMR 4007 0060h, SCI4.SMR 4007 0080h, SCI5.SMR 4007 00A0h, SCI6.SMR 4007 00C0h, SCI7.SMR 4007 00E0h, SCI8.SMR 4007 0100h, SCI9.SMR 4007 0120h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLKA clock (n = 0)*1 0 1: PCLKA/4 clock (n = 1)*1 1 0: PCLKA/16 clock (n = 2)*1 1 1: PCLKA/64 clock (n = 3).*1	R/W*4
b2	MP	Multi-Processor Mode	Valid only in asynchronous mode: 0: Disable multi-processor communications function 1: Enable multi-processor communications function.	R/W*4
b3	STOP	Stop Bit Length	Valid only in asynchronous mode: 0: 1 stop bit 1: 2 stop bits.	R/W*4
b4	PM	Parity Mode	Valid only when the PE bit is 1: 0: Even parity 1: Odd parity.	R/W*4
b5	PE	Parity Enable	Valid only in asynchronous mode: • When transmitting: 0: Do not add parity bit 1: Add parity bit. • When receiving: 0: Do not check parity bit 1: Check parity bit.	R/W*4
b6	CHR	Character Length	Selects the transmit/receive character length in combination with the SCMR.CHR1 bit: CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length.*3 Valid only in asynchronous mode.*2	R/W*4
b7	CM	Communication Mode	0: Asynchronous mode or simple IIC mode 1: Clock synchronous mode or simple SPI mode.	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register. See section 34.2.17, Bit Rate Register (BRR).
Note 2. In any mode other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB-first is fixed and the MSB (bit [7]) in the TDR register is not transmitted in transmit mode.
Note 4. Writable only when SCR.TE = 0 and SCR.RE = 0 (both serial transmission and reception are disabled).

The SMR register sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator. For the relationship between the settings of these bits and the baud rate, see section 34.2.17, Bit Rate Register (BRR).

MP bit (Multi-Processor Mode)

The MP bit disables or enables the multi-processor communications function. The PE and PM bit settings are invalid in multi-processor mode.

STOP bit (Stop Bit Length)

The STOP bit selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM bit (Parity Mode)

The PM bit selects the parity mode (even or odd) for transmission and reception. The PM bit setting is invalid in multi-processor mode.

PE bit (Parity Enable)

When the PE bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception. Regardless of the PE bit setting, the parity bit is not added or checked in multi-processor format.

CHR bit (Character Length)

The CHR bit selects the data length for transmission and reception in combination with the SCMR.CHR1 bit. In modes other than asynchronous, a fixed data length of 8 bits is used.

CM bit (Communication Mode)

The CM bit selects the communication mode:

- Asynchronous mode or simple IIC mode
- Clock synchronous mode or simple SPI mode.

34.2.10 Serial Mode Register for Smart Card Interface Mode (SMR_SMCI) (SCMR.SMIF = 1)

Address(es): SCI0.SMR_SMCI 4007 0000h, SCI1.SMR_SMCI 4007 0020h, SCI2.SMR_SMCI 4007 0040h, SCI3.SMR_SMCI 4007 0060h, SCI4.SMR_SMCI 4007 0080h, SCI5.SMR_SMCI 4007 00A0h, SCI6.SMR_SMCI 4007 00C0h, SCI7.SMR_SMCI 4007 00E0h, SCI8.SMR_SMCI 4007 0100h, SCI9.SMR_SMCI 4007 0120h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLKA clock (n = 0)*1 0 1: PCLKA/4 clock (n = 1)*1 1 0: PCLKA/16 clock (n = 2)*1 1 1: PCLKA/64 clock (n = 3).*1	R/W*2
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 34.3 lists the combinations of the SCMR.BCP2 and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	Valid only when the PE bit is 1: 0: Even parity 1: Odd parity.	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: Non-block transfer mode operation 1: Block transfer mode operation.	R/W*2
b7	GM	GSM Mode	0: Non-GSM mode operation 1: GSM mode operation.	R/W*2

Note 1. n is the decimal notation of the value of n in the BRR register. See section 34.2.17, Bit Rate Register (BRR).
Note 2. Writable only when SCR_SMCI.TE = 0 and SCR_SMCI.RE = 0 (both serial transmission and reception are disabled).

The SMR_SMCI register sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bit (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator. For the relationship between the settings of these bits and the baud rate, see section 34.2.17, Bit Rate Register (BRR).

BCP[1:0] bits (Base Clock Pulse)

The BCP[1:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set these bits in combination with the SCMR.BCP2 bit.

For details, see section 34.6.4, Receive Data Sampling Timing and Reception Margin.

Table 34.3 Combinations of SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
0	00	93 clock cycles (S = 93)*1
0	01	128 clock cycles (S = 128)*1
0	10	186 clock cycles (S = 186)*1
0	11	512 clock cycles (S = 512)*1
1	00	32 clock cycles (S = 32)*1 (initial value)
1	01	64 clock cycles (S = 64)*1
1	10	372 clock cycles (S = 372)*1
1	11	256 clock cycles (S = 256)*1

Note 1. See section 34.2.17, Bit Rate Register (BRR).

PM bit (Parity Mode)

The PM bit selects the parity mode for transmission and reception (even or odd). For details on the usage of this bit in smart card interface mode, see section 34.6.2, Data Format (Except in Block Transfer Mode).

PE bit (Parity Enable)

Set the PE bit to 1. The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK bit (Block Transfer Mode)

Setting the BLK bit to 1 enables block transfer mode operation. For details, see section 34.6.3, Block Transfer Mode.

GM bit (GSM Mode)

Setting the GM bit to 1 enables GSM mode operation. In GSM mode, the SSR_SMCI.TEND flag set timing is moved forward to 11.0 ETUs (elementary time unit = 1-bit transfer time) from the start bit, and clock output control is added. For details, see section 34.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 34.6.8, Clock Output Control.

34.2.11 Serial Control Register (SCR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SCR 4007 0002h, SCI1.SCR 4007 0022h, SCI2.SCR 4007 0042h, SCI3.SCR 4007 0062h,
SCI4.SCR 4007 0082h, SCI5.SCR 4007 00A2h, SCI6.SCR 4007 00C2h, SCI7.SCR 4007 00E2h,
SCI8.SCR 4007 0102h, SCI9.SCR 4007 0122h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• Asynchronous mode: b1 b0 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port based on the I/O port settings 0 1: On-chip baud rate generator A clock with the same frequency as the bit rate is output from the SCKn pin 1 x: External clock Input a clock with a frequency 16 times the bit rate from the SCKn pin when the SEMR.ABCS bit is 0. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. • Clock synchronous mode: b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin 1 x: External clock. The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: Disable SCIn_TEI interrupt requests 1: Enable SCIn_TEI interrupt requests.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	Valid in asynchronous mode when SMR.MP = 1: 0: Non-multi processor reception 1: When data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, and FER in SSR to 1 is disabled. When data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi processor reception is resumed.	R/W*3
b4	RE	Receive Enable	0: Disable serial reception 1: Enable serial reception.	R/W*2
b5	TE	Transmit Enable	0: Disable serial transmission 1: Enable serial transmission.	R/W*2
b6	RIE	Receive Interrupt Enable	0: Disable SCIn_RXI and SCIn_ERI interrupt requests 1: Enable SCIn_RXI and SCIn_ERI interrupt requests.	R/W
b7	TIE	Transmit Interrupt Enable	0: Disable SCIn_TXI interrupt requests 1: Enable SCIn_TXI interrupt requests.	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, when the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. When the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

Note 3. When writing a new value to a bit other than the MPIE bit of this register in multi-processor mode (SMR.MP bit = 1), write 0 to the MPIE bit using the store instruction to avoid accidentally setting the MPIE bit to 1 by a read-modify-write operation when using a bit manipulation instruction.

The SCR register controls operation and clock source selection for transmission and reception.

CKE[1:0] bits (Clock Enable)

The CKE[1:0] bits select the clock source and the SCKn pin function.

TEIE bit (Transmit End Interrupt Enable)

The TEIE bit enables or disables SCIn_TEI interrupt requests. Set TEIE to 0 to disable an SCIn_TEI interrupt request.

In simple IIC mode, SCIn_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE bit (Multi-Processor Interrupt Enable)

When the MPIE bit is set to 1 and data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags RDRF, ORER, and FER in SSR/SSR_FIFO to 1 is disabled. When data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi processor reception resumes. For details, see section 34.4, Multi-Processor Communication Function.

When the MPB bit in the SSR register is 0, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, SCIn_RXI and SCIn_ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting of the ORER and FER flags to 1 is enabled.

Set MPIE to 0 if the multi-processor communications function is not used.

RE bit (Receive Enable)

The RE bit enables or disables serial reception. When the RE bit is set to 1, serial reception starts by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Set the reception format in the SMR register before setting the RE bit to 1.

In non-FIFO operation, when reception is halted by setting the RE bit to 0, the RDRF, ORER, FER, and PER flags in the SSR register are not affected, and the previous values are retained.

When FIFO operation is selected and reception is halted by setting the RE bit to 0, the RDF, ORER, FER, PER, and DR flags in SSR_FIFO are not affected and the previous values are retained.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission.

When the TE bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Set the transmission format in the SMR register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt requests are disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR/SSR_FIFO then setting the flag to 0, or by setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

The TIE bit enables or disables SCIn_TXI interrupt requests. SCIn_TXI interrupt requests are disabled by setting the TIE bit to 0. Set the TIE bit to 1 while the TE bit is 1. The SCIn_TXI interrupt occurs after TE and TIE bits are set to 1 simultaneously, before transfer starts.

34.2.12 Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1)

Address(es): SCI0.SCR_SMCI 4007 0002h, SCI1.SCR_SMCI 4007 0022h, SCI2.SCR_SMCI 4007 0042h, SCI3.SCR_SMCI 4007 0062h, SCI4.SCR_SMCI 4007 0082h, SCI5.SCR_SMCI 4007 00A2h, SCI6.SCR_SMCI 4007 00C2h, SCI7.SCR_SMCI 4007 00E2h, SCI8.SCR_SMCI 4007 0102h, SCI9.SCR_SMCI 4007 0122h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When SMR_SMCI.GM = 0: b1 b0 0 0: Disable output The SCKn pin is available for use as an I/O port if set up in the I/O port settings 0 1: Output clock 1 x: Setting prohibited. • When SMR_SMCI.GM = 1: b1 b0 0 0: Fix output low x 1: Output clock 1 0: Fix output high.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	Set this bit to 0 in smart card interface mode	R/W
b3	MPIE	Multi-Processor Interrupt Enable	Set this bit to 0 in smart card interface mode	R/W
b4	RE	Receive Enable	0: Disable serial reception 1: Enable serial reception.	R/W*2
b5	TE	Transmit Enable	0: Disable serial transmission 1: Enable serial transmission.	R/W*2
b6	RIE	Receive Interrupt Enable	0: Disable SCIn_RXI and SCIn_ERI interrupt requests 1: Enable SCIn_RXI and SCIn_ERI interrupt requests.	R/W
b7	TIE	Transmit Interrupt Enable	0: Disable SCIn_TXI interrupt requests 1: Enable SCIn_TXI interrupt requests.	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written to TE and RE.

The SCR_SMCI register sets transmission and reception control, interrupt control, and clock source selection for transmission and reception.

For details on interrupt requests, see section 34.10, Interrupt Sources.

CKE[1:0] bits (Clock Enable)

The CKE[1:0] bits control the clock output from the SCKn pin. In GSM mode, clock output can be dynamically switched. For details, see section 34.6.8, Clock Output Control.

TEIE bit (Transmit End Interrupt Enable)

Set the TEIE bit to 0 in smart card interface mode.

MPIE bit (Multi-Processor Interrupt Enable)

Set the MPIE bit to 0 in smart card interface mode.

RE bit (Receive Enable)

The RE bit enables or disables serial reception. When the RE bit is set to 1, serial reception starts by detecting the start bit. Set the reception format in the SMR_SMCI register before setting the RE bit to 1.

If reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in SSR_SMCI are not affected and the previous values are retained.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission. When the TE bit is set to 1, serial transmission is started by writing transmit data to TDR. Set the transmission format in the SMR_SMCI register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt requests are disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR_SMCI register, and then setting the flag to 0, or by setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

The TIE bit enables or disables SCIn_TXI interrupt requests. SCIn_TXI interrupt requests are disabled by setting the TIE bit to 0. Set the TIE bit to 1 while the TE bit is 1. The SCIn_TXI interrupt occurs after TE and TIE bits are set to 1 simultaneously, before transfer starts.

34.2.13 Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0)

Address(es): SCI0.SSR 4007 0004h, SCI1.SSR 4007 0024h, SCI2.SSR 4007 0044h, SCI3.SSR 4007 0064h, SCI4.SSR 4007 0084h, SCI5.SSR 4007 00A4h, SCI6.SSR 4007 00C4h, SCI7.SSR 4007 00E4h, SCI8.SSR 4007 0104h, SCI9.SSR 4007 0124h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the value of the multi-processor bit in the transmission frame: 0: Data transmission cycle 1: ID transmission cycle.	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame: 0: Data transmission cycle 1: ID transmission cycle.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted 1: Character transfer is complete.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: Parity error occurred.	R/(W)*1
b4	FER	Framing Error Flag	0: No framing error occurred 1: Framing error occurred.	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: Overrun error occurred.	R/(W)*1
b6	RDRF	Receive Data Full Flag	0: No received data in RDR register 1: Received data in RDR register.	R/(W)*1
b7	TDRE	Transmit Data Empty Flag	0: Transmit data in TDR register 1: No transmit data in TDR register.	R/(W)*1

Note 1. Only 0 can be written to clear the flag after reading 1.

The SSR register provides SCI status flags and transmission and reception multi-processor bits.

MPBT bit (Multi-Processor Bit Transfer)

The MPBT bit sets the value of the multi-processor bit in the transmit frame.

MPB bit (Multi-Processor)

The MPB bit holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled) and the FCR.FM bit is set to 0 (non-FIFO selected). When the SCR.TE bit is set to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated on transmission of the tail-end bit of a character being transmitted.

[Clearing conditions]

- When transmit data is written to the TDR register while the SCR.TE bit is 1
- When 0 is written to TDRE after 1 is read while the SCR.TE bit is 1.

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

  Although receive data is transferred to the RDR register when the parity error occurs, no SCIn_RXI interrupt request occurs. When the PER flag is set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER flag (Framing Error Flag)

The FER flag indicates that a framing error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When 0 is sampled as the stop bit during reception in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

  In 2-stop-bit mode, only the first stop bit is checked. The second stop bit is not checked. Although receive data is transferred to the RDR register when the framing error occurs, no SCIn_RXI interrupt request occurs. When the FER flag is to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to FER after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the FER flag is not affected and retains its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error and a framing error is read from the RDR register.

The data received before an overrun error occurred is saved in the RDR register, but data received after the error is lost. When the ORER flag is set to 1, receive data is not forwarded to the RDR register. In clock synchronous mode, serial transmission and reception are stopped.

[Clearing condition]

- When 0 is written to the ORER flag after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to the RDRF flag after 1 is read
- When data is read from the RDR register.

Note: Do not clear RDRF flag by accessing RDRF bit in the SSR register unless communication is aborted.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0
- When data is transmitted from the TDR register to the TSR register.

[Clearing conditions]

- When 0 is written to the TDRE flag after 1 is read
- When the SCR.TE bit is 1 and data is written to the TDR register.

Note: Do not clear TDRE flag by accessing TDRE bit in the SSR register unless communication is aborted.

34.2.14 Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1)

Address(es): SCI0.SSR_FIFO 4007 0004h, SCI1.SSR_FIFO 4007 0024h, SCI2.SSR_FIFO 4007 0044h, SCI3.SSR_FIFO 4007 0064h, SCI4.SSR_FIFO 4007 0084h, SCI5.SSR_FIFO 4007 00A4h, SCI6.SSR_FIFO 4007 00C4h, SCI7.SSR_FIFO 4007 00E4h, SCI8.SSR_FIFO 4007 0104h, SCI9.SSR_FIFO 4007 0124h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDFE	RDF	ORER	FER	PER	TEND	—	DR
Value after reset:	1	0	0	0	0	0	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	DR	Receive Data Ready Flag	0: Receiving is in progress, or no received data remains in FRDRHL after successfully completed reception (receive FIFO empty) 1: Next receive data is not received for a period after normal receiving is complete, when the amount of data stored in the FIFO is equal to or less than the receive triggering number.	R/(W)*1
b1	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Bit	Symbol	Bit name	Description	R/W
b2	TEND	Transmit End Flag	0: A character is being transmitted 1: Character transfer is complete.	R/(W)*1
b3	PER	Parity Error Flag	0: No parity error occurred 1: Parity error occurred.	R/(W)*1
b4	FER	Framing Error Flag	0: No framing error occurred 1: Framing error occurred.	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: Overrun error occurred.	R/(W)*1
b6	RDF	Receive FIFO Data Full Flag	0: The amount of receive data written in FRDRHL is less than the specified receive triggering number 1: The amount of receive data written in FRDRHL is equal to or greater than the specified receive triggering number.	R/(W)*1
b7	TDFE	Transmit FIFO Data Empty Flag	0: The amount of transmit data written in FTDRHL exceeds the specified transmit triggering number 1: The amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number.	R/(W)*1

Note 1. Only 0 can be written, to clear the flag after reading 1.

The SSR_FIFO register provides the SCI with FIFO mode status flags.

DR flag (Receive Data Ready Flag)

The DR flag indicates that the amount of data stored in the Receive FIFO Data Register (FRDRHL) falls below the specified receive triggering number, and that no next data is received after 15 ETUs (elementary time units) from the last stop bit in asynchronous mode. This flag is valid only in asynchronous mode, including multi-processor mode, when FIFO operation is selected.

In clock synchronous mode, the DR flag is not set to 1.

[Setting condition]

- When FRDRHL contains less data than the specified receive triggering number, and no next data is received after 15 ETUs*1 from the last stop bit, and the SSR_FIFO.FER and SSR_FIFO.PER flags are 0.

[Clearing conditions]

- When 1 is read from DR, after all received data is read
- When the FCR.FM bit is changed from 0 to 1.

Note 1. This is equivalent to 1.5 frames in the 8-bit format with one stop bit.
The DR flag is only set to 1 when FIFO is selected in asynchronous mode, including multi-processor mode. It is not set to 1 in other operation modes.

TEND flag (Transmit End Flag)

The TEND flag indicates that FTDRHL does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- When FTDRHL does not contain transmit data when the last bit of a 1-byte serial character is transmitted.

[Clearing conditions]

- When transmit data is written to FTDRHL while the SCR.TE bit is 1
- When 0 is written to the TEND flag after 1 is read while the SCR.TE bit is 1
- When the FCR.FM bit is changed from 0 to 1.

PER flag (Parity Error Flag)

The PER flag indicates whether there is a parity error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When data is received and a parity error is detected, when the address match function is disabled (DCCR.DCME = 0).

[Clearing condition]

- When 0 is written to the PER flag after 1 is read.

The reception operation is continuous, and the receive data is stored in the FRDRHL register, even when a parity error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER flag (Framing Error Flag)

The FER flag indicates whether there is a framing error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When 0 is sampled as the stop bit during reception when the address match function is disabled (DCCR.DCME = 0).

[Clearing condition]

- When 0 is written to the FER flag after 1 is read.

The reception operation is continuous, and the receive data is stored in the FRDRHL register, even when a framing error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the FER flag is not affected and retains its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that the receive operation stopped abnormally because an overrun error occurred.

[Setting condition]

- When the next serial reception completes while the receive FIFO is full with 16-byte receive data.

[Clearing condition]

- When 0 is written to the ORER flag after 1 is read.

When the SCR.RE bit is set to 0 (serial reception is disabled), the ORER flag is not affected and retains its previous value.

RDF flag (Receive FIFO Data Full Flag)

The RDF flag indicates that receive data was transferred to the FRDRHL register, and the amount of data in FRDRHL is equal to or exceeds the specified receive triggering number. When RTRG is set to 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0.

[Setting condition]

- When the amount of receive data equal to or greater than the specified receive triggering number is stored in FRDRHL,*1 and the FIFO is not empty.

[Clearing conditions]

- When 0 is written to the RDF flag after 1 is read
- When FRDRHL is read by the DMAC or DTC, but only when the block transfer is the last transmission
- When the setting and clearing conditions occur at the same time, the RDF flag is set to 0. After that, when the amount of data stored in the FRDRHL register is equal to or greater than the RTRG value, RDF is set to 1 after 1

PCLKA.

Note: Do not clear RDF flags by accessing RDF bit in the SSR register before reading receive data unless communication is aborted.

Note 1. Because FRDRHL is a 16-stage FIFO register, the maximum amount of data that can be read when RDF is 1 is equivalent to the specified receive triggering number. If an attempt is made to read after all the data in FRDRHL is read, the data is undefined.

TDFE flag (Transmit FIFO Data Empty Flag)

The TDFE flag indicates that data is transferred from the FTDRHL register into the TSR register, the amount of data in FTDRHL is below the specified transmit triggering number, and writing of transmit data to FTDRHL is enabled.

[Setting conditions]

- When the TE bit in SCR is 0
- When the amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number.*1

[Clearing conditions]

- When writing to FTDRHL is executed on the last transmission while the DTC or DMAC is activated
- When 0 is written to the TDFE flag after reading 1 is read.
The setting conditions are given priority when TE = 0. When the setting condition and clearing condition occur at the same time, the TDFE flag is set to 0. After that, when the amount of data stored in the FTDRHL register is equal to or less than the TTRG value, TDFE is set to 1 after 1 PCLKA.

Note: Do not clear TDFE flags by accessing TDFE bit in the SSR register before writing transmit data unless communication is aborted.

Note 1. Because the FTDRHL register is a 16-stage FIFO register, when the TDFE flag is 1, the maximum amount of data that can be written to the FTDRHL register is 16 minus FDR.T[4:0] bytes. If more data is written, data is discarded.

34.2.15 Serial Status Register for Smart Card Interface Mode (SSR_SMCI) (SCMR.SMIF = 1)

Address(es): SCI0.SSR_SMCI 4007 0004h, SCI1.SSR_SMCI 4007 0024h, SCI2.SSR_SMCI 4007 0044h, SCI3.SSR_SMCI 4007 0064h, SCI4.SSR_SMCI 4007 0084h, SCI5.SSR_SMCI 4007 00A4h, SCI6.SSR_SMCI 4007 00C4h, SCI7.SSR_SMCI 4007 00E4h, SCI8.SSR_SMCI 4007 0104h, SCI9.SSR_SMCI 4007 0124h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Set this bit to 0 in smart card interface mode	R/W
b1	MPB	Multi-Processor	Set this bit to 0 in smart card interface mode	R
b2	TEND	Transmit End Flag	0: A character is being transmitted 1: Character transfer is complete.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: Parity error occurred.	R/(W)*
b4	ERS	Error Signal Status Flag	0: No low error signal response 1: Low error signal response occurred.	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: Overrun error occurred.	R/(W)*1
b6	RDRF	Receive Data Full Flag	0: No received data in RDR register 1: Received data in RDR register.	R/(W)*1

Bit	Symbol	Bit name	Description	R/W
b7	TDRE	Transmit Data Empty Flag	0: Transmit data in TDR register 1: No transmit data in TDR register.	R/(W)*1

Note 1. Only 0 can be written, to clear the flag after 1 is read.

The SSR_SMCI register provides the SCI with smart card interface mode status flags.

TEND flag (Transmit End Flag)

When there is no error signal from the receiving side, the TEND flag is set to 1 when more data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit = 0 (serial transmission is disabled).
  When the SCR_SMCI.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period elapses after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated.

  The set timing is determined by the following register settings:
  - When SMR_SMCI.GM = 0 and SMR_SMCI.BLK = 0, 12.5 ETUs after the start of transmission
  - When SMR_SMCI.GM = 0 and SMR_SMCI.BLK = 1, 11.5 ETUs after the start of transmission
  - When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 0, 11.0 ETUs after the start of transmission
  - When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 1, 11.0 ETUs after the start of transmission.

[Clearing conditions]

- When transmit data is written to the TDR register while the SCR_SMCI.TE bit is 1
- When 0 is written to the TDRE flag after 1 is read while the SCR_SMCI.TE bit is 1.

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception. Although receive data is transferred to RDR when a parity error occurs, no SCIn_RXI interrupt request occurs. After the PER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to the PER flag after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the RE bit in SCR_SMCI is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled.

[Clearing condition]

- When 0 is written to the ERS flag after 1 is read.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error is read from the RDR register.

The data received before an overrun error occurred is saved in the RDR, but data received after the error is lost. When the ORER flag is set to 1, receive data is not forwarded to the RDR register.

[Clearing condition]

- When 0 is written to the ORER flag after 1 is read. After writing 0 to this flag, read it to verify that its value is 0.

When the RE bit in SCR_SMCI is set to 0, the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to the RDRF flag after 1 is read
- When data is read from the RDR register.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit is 0
- When data is transmitted from the TDR register to the TSR register.

[Clearing conditions]

- When 0 is written to the TDRE flag after 1 is read
- When the SCR_SMCI.TE bit is 1 and data is written to the TDR register.

Note: Do not clear TDRE flags by accessing TDRE bit in the SSR register unless communication is aborted.

34.2.16 Smart Card Mode Register (SCMR)

Address(es): SCI0.SCMR 4007 0006h, SCI1.SCMR 4007 0026h, SCI2.SCMR 4007 0046h, SCI3.SCMR 4007 0066h, SCI4.SCMR 4007 0086h, SCI5.SCMR 4007 00A6h, SCI6.SCMR 4007 00C6h, SCI7.SCMR 4007 00E6h, SCI8.SCMR 4007 0106h, SCI9.SCMR 4007 0126h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (asynchronous mode, clock synchronous mode, simple SPI mode, or simple IIC mode) 1: Smart card interface mode.	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert	0: TDR register contents are transmitted as they are. Receive data is stored as received in the RDR register. 1: TDR register contents are inverted before transmission. Receive data is stored in inverted form in the RDR register. The SINV bit can be used in the following modes: • Smart card interface mode • Asynchronous mode (including multi-processor mode) • Clock synchronous mode • Simple SPI mode. Set the SINV bit to 0 for operation in simple IIC mode.	R/W*1

Bit	Symbol	Bit name	Description	R/W
b3	SDIR	Transmitted/Received Data Transfer Direction	0: Transfer LSB-first 1: Transfer MSB-first. The SDIR bit can be used in the following modes: • Smart card interface mode • Asynchronous mode (including multi-processor mode) • Clock synchronous mode • Simple SPI mode. Set the SDIR bit to 1 for operation in simple IIC mode.	R/W*[1]
b4	CHR1	Character Length 1	Valid only in asynchronous mode.*[2] Selects the transmit/receive character length in combination with the SMR.CHR bit: CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length.*[3]	R/W*[1]
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR_SMCI.BCP[1:0] bits. Table 34.4 lists the combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits.	R/W*[1]

Note 1. Writable only when the TE and RE bits in SCR/SCR_SMCI are 0 (both serial transmission and reception are disabled).
Note 2. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 3. LSB-first must be selected and the value of the MSB (bit [7]) in TDR cannot be transmitted.

The SCMR register selects the smart card interface and communication format.

SMIF bit (Smart Card Interface Mode Select)

Setting the SMIF bit to 1 selects smart card interface mode. Setting it to 0 selects all other modes:

- Asynchronous mode, including multi-processor mode
- Clock synchronous mode
- Simple SPI mode
- Simple IIC mode.

SINV bit (Transmitted/Received Data Invert)

The SINV bit inverts the transmit and receive data logic level. It does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR or SMR_SMCI.

CHR1 bit (Character Length 1)

The CHR1 bit selects the data length of transmit and receive data in combination with the CHR bit in the SMR register. A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 bit (Base Clock Pulse 2)

The BCP2 bit selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR_SMCI.BCP[1:0] bits.

Table 34.4 Combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits (1 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
0	00	93 clock cycles (S = 93)*[1]
0	01	128 clock cycles (S = 128)*[1]
0	10	186 clock cycles (S = 186)*[1]
0	11	512 clock cycles (S = 512)*[1]
1	00	32 clock cycles (S = 32)*[1] (Initial Value)

Table 34.4 Combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits (2 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
1	01	64 clock cycles (S = 64)*1
1	10	372 clock cycles (S = 372)*1
1	11	256 clock cycles (S = 256)*1

Note 1. See section 34.2.17, Bit Rate Register (BRR).

34.2.17 Bit Rate Register (BRR)

Address(es): SCI0.BRR 4007 0001h, SCI1.BRR 4007 0021h, SCI2.BRR 4007 0041h, SCI3.BRR 4007 0061h, SCI4.BRR 4007 0081h, SCI5.BRR 4007 00A1h, SCI6.BRR 4007 00C1h, SCI7.BRR 4007 00E1h, SCI8.BRR 4007 0101h, SCI9.BRR 4007 0121h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

BRR is an 8-bit register that adjusts the bit rate.

As each SCI channel has independent baud rate generator control, different bit rates can be set for each channel. Table 34.5 shows the relationship between the setting (N) in the BRR and the bit rate (B) for asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

The initial value of the BRR register is FFh. The BRR register can be read by the CPU, but it can be written to only when the TE and RE bits in SCR/SCR_SMCI are 0.

Table 34.5 Relationship between N setting in BRR and bit rate B

	SEMR settings				
Mode	BGDM bit	ABCS bit	ABCSE bit	BRR setting	Error
Asynchronous, multi-processor transfer	0	0	0	$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1\} \times 100$
	1	0	0	$N = \frac{PCLKA \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 32 \times 2^{2n-1} \times (N + 1)} - 1\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKA \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 16 \times 2^{2n-1} \times (N + 1)} - 1\} \times 100$
	Don’t care	Don’t care	1	$N = \frac{PCLKA \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 12 \times 2^{2n-1} \times (N + 1)} - 1\} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLKA \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	-
Smart card interface				$N = \frac{PCLKA \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times S \times 2^{2n+1} \times (N + 1)} -1\} \times 100$
Simple IIC*1				$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	-

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator ($0 \leq N \leq 255$)
PCLKA: Operating frequency (MHz)

n and S: Determined by the SMR/SMR_SMCI and SCMR register settings as listed in Table 34.7 and Table 34.8.

Note 1. Adjust the bit rate so that the widths of high and low level of the SCL output in simple IIC mode satisfy the I²C bus standard.

Table 34.6 Calculating widths of SCL high and low levels

Mode	SCL	Formula (result in seconds)
IIC	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLKA \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLKA \times 10^6}$

Table 34.7 Clock source settings

SMR or SMR_SMCI.CKS[1:0] bit setting		
CKS[1:0] bits	Clock source	n
0 0	PCLKA clock	0
0 1	PCLKA/4 clock	1
1 0	PCLKA/16 clock	2
1 1	PCLKA/64 clock	3

Table 34.8 Base clock settings in smart card interface mode

SCMR.BCP2 bit setting	SMR_SMCI.BCP[1:0] bit setting		
BCP2 bit	BCP[1:0] bits	Base clock cycles for 1-bit period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 34.9 and Table 34.10 list examples of BRR (N) settings in asynchronous mode. Table 34.11 lists the maximum bit rate settable for each operating frequency. Table 34.15 lists examples of BRR (N) settings in smart card interface mode.

In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 34.6.4, Receive Data Sampling Timing and Reception Margin. Table 34.12 and Table 34.14 list the maximum bit rates with external clock input.

When either the Asynchronous Mode Base Clock Select bit (ABCS) or the Baud Rate Generator Double-speed Mode Select bit (BGDM) in the Serial Extended Mode Register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice the value listed in Table 34.16. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 34.9 Examples of BRR settings for different bit rates in asynchronous mode (1) (1 of 2)

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08

Table 34.9 Examples of BRR settings for different bit rates in asynchronous mode (1) (2 of 2)

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	-0.17	3	70	0.03	3	75	0.48	3	79	-0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	-0.93	0	51	0.16	0	55	0.00	0	58	-0.69	0	63	0.00
19200	0	22	-0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	-1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	-2.34	0	15	0.00

Note: In this example, SEMR.ABCS = 0, SEMR.ABCSE = 0, and SEMR.BGDM = 0.
When either the ABCS or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM are set to 1, the bit rate increases four times.

Table 34.10 Examples of BRR settings for different bit rates in asynchronous mode (2)

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	-0.25	3	110	-0.02	3	132	0.13	3	145	0.33	3	177	-0.25
150	3	64	0.16	3	80	0.47	3	97	-0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	-0.15	2	194	0.16	2	214	-0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	-0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	-0.15	1	194	0.16	1	214	-0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	-0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	-0.15	0	194	0.16	0	214	-0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	-0.35	0	106	0.39	0	129	0.16
19200	0	32	-1.36	0	40	-0.76	0	48	-0.35	0	53	-0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	-0.54	0	32	-1.36

Bit rate (bps)	Operating frequency PCLKA (MHz)								
	50			60			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	-0.02	—	—	—	—	—	—
150	3	162	-0.15	3	194	0.16	—	—	—
300	3	80	0.47	3	97	-0.35	3	194	0.16
600	2	162	-0.15	3	48	-0.35	3	97	-0.35
1200	2	80	0.47	2	97	-0.35	3	48	-0.35
2400	1	162	-0.15	2	48	-0.35	2	97	-0.35
4800	1	80	0.47	1	97	-0.35	2	48	-0.35
9600	0	162	-0.15	1	48	-0.35	1	97	-0.35
19200	0	80	0.47	0	97	-0.35	1	48	-0.35
31250	0	49	0.00	0	59	0.00	0	119	0
38400	0	40	-0.76	0	48	-0.35	0	97	-0.35

Note: In this example, SEMR.ABCS = 0, SEMR.ABCSE = 0, and SEMR.BGDM = 0.
When either the ABCS or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 34.11 Maximum bit rate for each operating frequency in asynchronous mode (1 of 2)

PCLKA (MHz)	SEMR settings					Maximum bit rate (bps)	PCLKA (MHz)	SEMR settings					Maximum bit rate (bps)
	BGDM bit	ABCS bit	ABCSE bit	n	N			BGDM bit	ABCS bit	ABCSE bit	n	N	
8	0	0	0	0	0	250000	16	0	0	0	0	0	500000
		1	0	0	0	500000			1	0	0	0	1000000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2000000
	Don’t care	Don’t care	1	0	0	1333333		Don’t care	Don’t care	1	0	0	2666666
9.8304	0	0	0	0	0	307200	17.2032	0	0	0	0	0	537600
		1	0	0	0	614400			1	0	0	0	1075200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2150400
	Don’t care	Don’t care	1	0	0	1638400		Don’t care	Don’t care	1	0	0	2867200
10	0	0	0	0	0	312500	18	0	0	0	0	0	562500
		1	0	0	0	625000			1	0	0	0	1125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	2250000
	Don’t care	Don’t care	1	0	0	1666666		Don’t care	Don’t care	1	0	0	3000000
12	0	0	0	0	0	375000	19.6608	0	0	0	0	0	614400
		1	0	0	0	750000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	2457600
	Don’t care	Don’t care	1	0	0	2000000		Don’t care	Don’t care	1	0	0	3276800

Table 34.11 Maximum bit rate for each operating frequency in asynchronous mode (2 of 2)

	SEMR settings					Maximum bit rate (bps)		SEMR settings					Maximum bit rate (bps)
PCLKA (MHz)	BGDM bit	ABCS bit	ABCSE bit	n	N		PCLKA (MHz)	BGDM bit	ABCS bit	ABCSE bit	n	N	
12.288	0	0	0	0	0	384000	20	0	0	0	0	0	625000
		1	0	0	0	768000			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	2500000
	Don’t care	Don’t care	1	0	0	2048000		Don’t care	Don’t care	1	0	0	3333333
14	0	0	0	0	0	437500	25	0	0	0	0	0	781250
		1	0	0	0	875000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	3125000
	Don’t care	Don’t care	1	0	0	2333333		Don’t care	Don’t care	1	0	0	4166666
30	0	0	0	0	0	937500	50	0	0	0	0	0	1562500
		1	0	0	0	1875000			1	0	0	0	3125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	3750000			1	0	0	0	6250000
	Don’t care	Don’t care	1	0	0	5000000		Don’t care	Don’t care	1	0	0	8333333
33	0	0	0	0	0	1031250	60	0	0	0	0	0	1875000
		1	0	0	0	2062500			1	0	0	0	3750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	4125000			1	0	0	0	7500000
	Don’t care	Don’t care	1	0	0	5500000		Don’t care	Don’t care	1	0	0	10000000
40	0	0	0	0	0	1250000	120	0	0	0	0	0	3750000
		1	0	0	0	2500000			1	0	0	0	7500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	5000000			1	0	0	0	15000000
	Don’t care	Don’t care	1	0	0	6666666		Don’t care	Don’t care	1	0	0	20000000

Table 34.12 Maximum bit rate with external clock input in asynchronous mode (1 of 2)

		Maximum bit rate (bps)	
PCLKA (MHz)	External input clock (MHz)	SEMR.ABCS bit = 0	SEMR.ABCS bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000

Table 34.12 Maximum bit rate with external clock input in asynchronous mode (2 of 2)

		Maximum bit rate (bps)	
PCLKA (MHz)	External input clock (MHz)	SEMR.ABCS bit = 0	SEMR.ABCS bit = 1
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000
50	12.5000	781250	1562500
60	15.0000	937500	1875000
120	30.0000	1875000	3750000

Table 34.13 BRR settings for different bit rates in clock synchronous and simple SPI modes

Bit rate (bps)	Operating frequency PCLKA (MHz)																					
	8		10		16		20		25		30		33		40		50		60		120	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																						
250	3	124	—	—	3	249																
500	2	249	—	—	3	124	—	—			3	233										
1 k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	155	3	194	3	233		
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	249	3	77	3	93	3	186
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	124	2	155	3	46	3	93
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	249	2	77	2	93	3	46
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	99	1	124	1	149	2	74
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	49	1	61	1	74	1	149
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	99	0	124	0	149	1	74
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	39	0	49	0	59	1	29
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	19	0	24	0	29	1	14
1 M	0	1			0	3	0	4	—	—	—	—	—	—	0	9	—	—	0	14	0	29
2.5 M			0	0*1			0	1	—	—	0	2	—	—	0	3	0	4	0	5	0	11
5 M							0	0*1	—	—	—	—	—	—	0	1	—	—	0	2	0	5
7.5 M											0	0*1							0	1	0	3
10 M															0	0*1					0	2

Space: Setting prohibited.

—: Can be set, but an error occurs.

Note 1. Continuous transmission or reception is not possible. After transmitting or receiving one frame of data, a 1-bit period elapses before starting to transmit or receive the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. Therefore, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate. When the FIFO is selected, this setting (BRR = 00h and SMR.CKS[1:0] = 00b) is not available.

Table 34.14 Maximum bit rate with external clock input in clock synchronous and simple SPI modes (1 of 2)

PCLKA (MHz)	External input clock (MHz)	Maximum bit rate (Mbps)
8	1.3333	1.3333333
10	1.6667	1.6666667
12	2.0000	2.0000000
14	2.3333	2.3333333
16	2.6667	2.6666667
18	3.0000	3.0000000
20	3.3333	3.3333333

Table 34.14 Maximum bit rate with external clock input in clock synchronous and simple SPI modes (2 of 2)

PCLKA (MHz)	External input clock (MHz)	Maximum bit rate (Mbps)
25	4.1667	4.1666667
30	5.0000	5.0000000
33	5.5000	5.5000000
40	6.6667	6.6666667
50	8.3333	8.3333333
60	10.0000	10.0000000
120	20.0000 (clock synchronous mode)	20.00000000
	10.0000 (simple SPI mode)	10.00000000

Table 34.15 BRR settings for different bit rates in smart card interface mode (n = 0, S = 372)

	Operating frequency PCLKA (MHz)											
	7.1424			10.00			10.7136			13.00		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	-30	0	1	-25	0	1	-8.99

	Operating frequency PCLKA (MHz)											
	14.2848			16.00			18.00			20.00		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	-15.99	0	2	-6.66

	Operating frequency PCLKA (MHz)											
	25.00			30.00			33.00			40.00		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	-12.49	0	3	5.01	0	4	-7.59	0	5	-6.66

	Operating frequency PCLKA (MHz)								
	50.00			60.00			120.00		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	6	0.01	0	7	5.01	0	16	-1.17

Table 34.16 Maximum bit rate for each operating frequency in smart card interface mode (S = 32)

PCLKA (MHz)	Maximum bit rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0
50.00	781250	0	0
60.00	937500	0	0
120.00	1875000	0	0

Table 34.17 BRR settings for different bit rates in simple IIC mode

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	30	0.8	1	12	-3.8	1	15	-2.3	1	19	-2.3
25 k	0	9	0.0	0	12	-3.8	1	4	0.0	1	5	428	1	7	-2.3
50 k	0	4	0.0	0	5	4.2	1	2	-16.7	1	2	4.2	1	3	-2.3
100 k*1	0	2	-16.7	0	3	-21.9	0	4	0.0	0	6	-10.7	1	1	-2.3
250 k	0	0	0.0	0	0	25	0	1	0.0	0	2	-16.7	0	2	4.2
350 k										0	1	-10.7	0	1	11.6*2
400 k*1										0	1	-21.9	0	1	-2.3*2

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	30			33			40			50			60		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	22	1.9	1	25	-0.8	0	124	0.0	2	9	-2.3	1	46	-0.3
25 k	1	8	4.2	1	9	3.1	0	49	0.0	2	3	-2.3	0	74	0.0
50 k	1	4	-6.3	1	4	3.1	0	24	0.0	2	1	-2.3	0	37	-1.3
100 k*1	1	2	-21.9	1	2	-14.1	0	12	-3.9	1	3	-2.3	0	18	-1.3
250 k	0	3	-6.3	0	3	3.1	0	4	0.0	0	5	4.2	0	7	-6.3
350 k	0	2	-10.7	0	2	-1.8	0	3	-10.7	0	4	-10.7	0	4	7.1
400 k*1	0	2	-21.9	0	2	-14.1	0	3	-21.9	0	3	-2.3*2	0	4	-6.3

Bit rate (bps)	Operating frequency PCLKA (MHz)		
	120		
	n	N	Error (%)
10 k	1	93	-0.3
25 k	0	149	0.0
50 k	0	74	0.0
100 k*1	0	37	-1.3
250 k	0	14	0.0
350 k	0	10	-2.6
400 k*1	0	9	-6.3

Note 1. The bit rate of 100 kbps and 400 kbps indicates the set value at which the error is on the minus side.
Note 2. The minimum value of low width is smaller than 1.3 µs which is the standard value of fast mode.

Table 34.18 Minimum widths at SCL high and low levels for different bit rates in simple IIC mode (1 of 2)

Bit rate (bps)	Operating frequency PCLKA (MHz)										
	8			10			16			20	
	n	N	Minimum widths at SCL high/low levels (µs)	n	N	Minimum widths at SCL high/low levels (µs)	n	N	Minimum widths at SCL high/low levels (µs)	n	N	Minimum widths at SCL high/low levels (µs)
10 k	0	24	43.75/50.00	0	30	43.40/49.60	1	12	45.5/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.2/20.80	1	4	17.50/20.00	1	5	16.80/19.20
50 k	0	4	8.75/10.00	0	5	8.40/9.60	1	2	10.50/12.00	1	2	8.40/9.60
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.38/5.00	0	6	4.90/5.60

Table 34.18 Minimum widths at SCL high and low levels for different bit rates in simple IIC mode (2 of 2)

Bit rate (bps)	Operating frequency PCLKA (MHz)											
	8			10			16			20		
	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)
250 k	0	0	1.75/2.00	0	0	1.40/1.60	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60
400 k										0	1	1.40/1.60

Bit rate (bps)	Operating frequency PCLKA (MHz)											
	25			30			33			40		
	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)
10 k	1	19	44.80/51.20	1	22	42.93/49.60	1	25	44.12/50.42	0	124	43.75/50.00
25 k	1	7	17.92/20.48	1	8	16.80/19.20	1	9	16.97/19.39	0	49	17.50/20.00
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	4	8.48/9.70	0	24	8.75/10.00
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.82	0	12	4.55/5.20
250 k	0	2	1.68/1.92	0	3	1.86/2.13	0	3	1.70/1.94	0	4	1.75/2.00
350 k	0	1	1.12/1.28*1	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60
400 k	0	1	1.12/1.28*1	0	2	1.40/1.60	0	2	1.27 /1.45	0	3	1.40/1.60

Bit rate (bps)	Operating frequency PCLKA (MHz)								
	50			60			120		
	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)	n	N	Minimum widths at SCL high/low levels (μs)
10 k	2	9	44.80/51.20	1	46	43.87/50.13	1	93	43.87/50.13
25 k	2	3	17.92/20.48	0	74	17.50/20.00	0	149	17.50/20.00
50 k	2	1	8.96/10.24	0	37	8.87/10.13	0	74	8.75/10.00
100 k	1	3	4.48/5.12	0	18	4.43/5.07	0	37	4.43/5.07
250 k	0	5	1.68/1.92	0	7	1.87/2.13	0	14	1.75/2.00
350 k	0	4	1.40/1.60	0	4	1.17/1.33	0	10	1.28/1.47
400 k	0	3	1.12/1.28*1	0	4	1.17/1.33	0	8	1.05/1.20

Note 1. The minimum value of low width is smaller than 1.3 μs which is the standard value of fast mode. The setting values are the same as in Table 34.17.

34.2.18 Modulation Duty Register (MDDR)

Address(es): SCI0.MDDR 4007 0012h, SCI1.MDDR 4007 0032h, SCI2.MDDR 4007 0052h, SCI3.MDDR 4007 0072h, SCI4.MDDR 4007 0092h, SCI5.MDDR 4007 00B2h, SCI6.MDDR 4007 00D2h, SCI7.MDDR 4007 00F2h, SCI8.MDDR 4007 0112h, SCI9.MDDR 4007 0132h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The MDDR register corrects the bit rate adjusted by the BRR register.

When the SEMR.BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected using the settings in the MDDR register (M/256). Table 34.19 shows the relationship between the MDDR setting (M) and the bit rate (B).

The initial value of the MDDR register is FFh. Bit [7] in this register is fixed to 1. The CPU can read the MDDR register, but the MDDR register is only writable when the TE and RE bits in SCR/SCR_SMCI are 0.

Table 34.19 Relationship between MDDR setting (M) and bit rate (B) when bit rate modulation function is used

	SEMR settings				
Mode	**BGDM bit**	**ABCS bit**	**ABCSE bit**	**BRR setting**	**Error**
Asynchronous, multi-processor transfer	0	0	0	$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	1	0	0	$N = \frac{PCLKA \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKA \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	Don’t care	Don’t care	1	$N = \frac{PCLKA \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
Clock synchronous, simple SPI*1				$N = \frac{PCLKA \times 10^6}{8 \times 2^{2n-1} \times (256/M) \times B} - 1$	—
Smart card interface				$N = \frac{PCLKA \times 10^6}{S \times 2^{2n+1} \times (256/M) \times B} - 1$	$\text{Error (\%)} = \{\frac{PCLKA \times 10^6}{B \times S \times 2^{2n+1} \times (256/M) \times (N+1)} - 1\} \times 100$
Simple IIC*2				$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	—

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 255)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLKA: Operating frequency (MHz)
n and S: Determined by the SMR/SMR_SMCI and SCMR register settings as listed in Table 34.7 and Table 34.8 in section 34, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode or in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).
Note 2. Adjust the bit rate so that the widths of high and low level of the SCL output in simple IIC mode satisfy the I2C bus standard.

Table 34.20 and Table 34.21 list examples of N settings in BRR and M settings in MDDR in asynchronous mode.

Table 34.20 Examples of BRR and MDDR settings for multiple bit rates in asynchronous mode (1)

	Operating frequency PCLKA (MHz)														
	8					9.8304					16				
Bit rate (bps)	**n**	**N**	**M**	**BGDM bit**	**Error (%)**	**n**	**N**	**M**	**BGDM bit**	**Error (%)**	**n**	**N**	**M**	**BGDM bit**	**Error (%)**
38400	0	5	236	0	0.03	0	7	(256)*1	0	0.00	0	10	173	1	-0.01
57600	0	3	236	0	0.03	0	4	240	0	0.00	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0.00	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0.00	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0.00	0	0	189	1	0.14

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	12					12.288					14				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256)*1	0	0.00	0	16	191	1	0.00
57600	0	5	236	0	0.03	0	4	192	0	0.00	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0.00	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	-0.17	0	2	202	1	-0.11
460800	0	0	157	1	-0.18	0	0	154	1	-0.26	0	0	135	1	0.14

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	16					17.2032					18				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256)*1	0	0.00	0	18	166	1	-0.01
57600	0	7	236	0	0.03	0	6	192	0	0.00	0	18	249	1	-0.01
115200	0	3	236	0	0.03	0	6	192	1	0.00	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	-0.20	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	-0.20	0	0	210	0	0.14

Note 1. In this example, the ABCS and ABCSE bits in the SEMR register are 0.
SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

Table 34.21 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (2)

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	19.6608					20					25				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	15	(256)*1	0	0.00	0	10	173	0	-0.01	0	11	151	0	0.00
57600	0	9	240	0	0.00	0	9	236	0	0.03	0	7	151	0	0.00
115200	0	4	240	0	0.00	0	4	236	0	0.03	0	3	151	0	0.00
230400	0	1	192	0	0.00	0	4	236	1	0.03	0	1	151	0	0.00
460800	0	0	192	0	0.00	0	0	189	0	0.14	0	0	151	0	0.00

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	30					33					40				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	-0.01
57600	0	10	173	0	-0.01	0	9	143	0	0.01	0	38	230	1	-0.01
115200	0	10	173	1	-0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	-0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.10	0	4	236	1	0.03

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	50					60					120				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	23	151	0	0.00	0	36	194	0	0.01	0	73	194	0	0.01
57600	0	15	151	0	0.00	0	21	173	0	-0.01	0	58	232	0	0.00
115200	0	7	151	0	0.00	0	10	173	0	-0.01	0	21	173	0	-0.01

Bit rate (bps)	Operating frequency PCLKA (MHz)														
	50					60					120				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
230400	0	3	151	0	0.00	0	10	173	1	-0.01	0	10	173	0	-0.01
460800	0	1	151	0	0.00	0	6	220	1	-0.09	0	10	173	1	-0.09

Note 1. In this example, the ABCS and ABCSE bits in the SEMR register are 0.
SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

34.2.19 Serial Extended Mode Register (SEMR)

Address(es): SCI0.SEMR 4007 0007h, SCI1.SEMR 4007 0027h, SCI2.SEMR 4007 0047h, SCI3.SEMR 4007 0067h, SCI4.SEMR 4007 0087h, SCI5.SEMR 4007 00A7h, SCI6.SEMR 4007 00C7h, SCI7.SEMR 4007 00E7h, SCI8.SEMR 4007 0107h, SCI9.SEMR 4007 0127h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	ABCSE	BRME	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	BRME	Bit Rate Modulation Enable	0: Disable bit rate modulation function 1: Enable bit rate modulation function.	R/W[*1]
b3	ABCSE	Asynchronous Mode Extended Base Clock Select 1	Valid only in asynchronous mode with SCR.CKE[1] = 0: 0: Clock cycles for 1-bit period determined by combination of the BGDM and ABCS bits in the SEMR register 1: Baud rate is 6 base clock cycles for 1-bit period.	R/W[*1]
b4	ABCS	Asynchronous Mode Base Clock Select	Valid only in asynchronous mode: 0: Select 16 base clock cycles for 1-bit period 1: Select 8 base clock cycles for 1-bit period.	R/W[*1]
b5	NFEN	Digital Noise Filter Function Enable	• In asynchronous mode: 0: Disable noise cancellation function for RXDn input signal 1: Enable noise cancellation function for RXDn input signal. • In simple IIC mode: 0: Disable noise cancellation function for SCLn and SDAn input signals 1: Enable noise cancellation function for SCLn and SDAn input signals. The NFEN bit must be 0 in all other modes.	R/W[*1]
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	Valid only in asynchronous mode with SCR.CKE[1] = 0. 0: Output clock from baud rate generator with single frequency 1: Output clock from baud rate generator with double frequency.	R/W[*1]
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	Valid only in asynchronous mode: 0: Detect low level on RXDn pin as start bit 1: Detect falling edge of RXDn pin as start bit.	R/W[*1]

Note 1. Writable only when the TE and RE bits in SCR/SCR_SMCI are 0 (both serial transmission and reception are disabled).

The SEMR register selects the clock source for the 1-bit period in asynchronous mode.

BRME bit (Bit Rate Modulation Enable)

The BRME bit enables or disables the bit rate modulation function. The bit rate generated by the on-chip baud rate generator is evenly corrected when this function is enabled.

ABCSE bit (Asynchronous Mode Extended Base Clock Select 1)

The ABCSE bit sets the pulse number for the base clock in a 1-bit period to 6, and the double-frequency clock is output from the baud rate generator. When the bit rate is set to 6 while dividing the bus clock frequency, use this bit and set SMR.CKS[1:0] = 00b and BRR = 0. Set this bit to 0 except in asynchronous mode.

ABCS bit (Asynchronous Mode Base Clock Select)

The ABCS bit selects the number of clock cycles for a 1-bit period. Set this bit to 0 except in asynchronous mode.

NFEN bit (Digital Noise Filter Function Enable)

The NFEN bit enables or disables the digital noise filter function.

When the digital noise filter function is enabled:

- Noise cancellation is applied to the RXDn input signal in asynchronous mode
- Noise cancellation is applied to the SDAn and SCLn input signals in simple IIC mode.

In all other modes, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as received.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit selects whether or not to double the base clock frequency output from the baud rate generator.

The BGDM bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.

Set this bit to 0 in modes other than asynchronous mode.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

The RXDSEL bit selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data reception operation depends on the setting of this bit. Set this bit to 1 when reception must be stopped while a break occurs or when reception must be started without keeping the RXDn pin input at the high level for the period of one data frame or longer after completion of the break.

Set this bit to 0 in modes other than asynchronous mode.

34.2.20 Noise Filter Setting Register (SNFR)

Address(es): SCI0.SNFR 4007 0008h, SCI1.SNFR 4007 0028h, SCI2.SNFR 4007 0048h, SCI3.SNFR 4007 0068h, SCI4.SNFR 4007 0088h, SCI5.SNFR 4007 00A8h, SCI6.SNFR 4007 00C8h, SCI7.SNFR 4007 00E8h, SCI8.SNFR 4007 0108h, SCI9.SNFR 4007 0128h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, selects the standard setting for the base clock: b2 b0 0 0 0: Use clock signal divided by 1 with noise filter. In simple IIC mode, selects the standard settings for the clock source of the on-chip baud rate generator selected in the SMR.CKS[1:0] bits: b2 b0 0 0 1: Use clock signal divided by 1 with noise filter 0 1 0: Use clock signal divided by 2 with noise filter 0 1 1: Use clock signal divided by 4 with noise filter 1 0 0: Use clock signal divided by 8 with noise filter. Other settings are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in SCR/SCR_SMCI are 0 (serial reception and transmission disabled).

The SNFR register sets the digital noise filter clock.

NFCS[2:0] bits (Noise Filter Clock Select)

The NFCS[2:0] bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple IIC mode, set the bits to a value in the range from 001b to 100b.

34.2.21 IIC Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 4007 0009h, SCI1.SIMR1 4007 0029h, SCI2.SIMR1 4007 0049h, SCI3.SIMR1 4007 0069h, SCI4.SIMR1 4007 0089h, SCI5.SIMR1 4007 00A9h, SCI6.SIMR1 4007 00C9h, SCI7.SIMR1 4007 00E9h, SCI8.SIMR1 4007 0109h, SCI9.SIMR1 4007 0129h

	b7	b6	b5	b4	b3	b2	b1	b0
	IICDL[4:0]					—	—	IICM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IICM	Simple IIC Mode Select	SMIF IICM 0 0: Asynchronous mode (including multi-processor mode), clock synchronous mode, or simple SPI mode 0 1: Simple IIC mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SDA Delay Output Select	SDA signal output delay in cycles of the clock signal from the on-chip baud rate generator: b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

SIMR1 selects simple IIC mode and the number of delay stages for the SDAn output.

IICM bit (Simple IIC Mode Select)

In combination with the SCMR.SMIF bit, the IICM bit selects the operating mode.

IICDL[4:0] bits (SDA Delay Output Select)

The IICDL[4:0] bits specify an output delay on the SDAn pin relative to the falling edge of the output on the SCLn pin.

The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLKA by the divisor set in SMR.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple IIC mode. In simple IIC mode, set the bits to a value in the range from 00001b to 11111b.

34.2.22 IIC Mode Register 2 (SIMR2)

Address(es): SCI0.SIMR2 4007 000Ah, SCI1.SIMR2 4007 002Ah, SCI2.SIMR2 4007 004Ah, SCI3.SIMR2 4007 006Ah, SCI4.SIMR2 4007 008Ah, SCI5.SIMR2 4007 00AAh, SCI6.SIMR2 4007 00CAh, SCI7.SIMR2 4007 00EAh, SCI8.SIMR2 4007 010Ah, SCI9.SIMR2 4007 012Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IICINTM	IIC Interrupt Mode Select	0: Use ACK/NACK interrupts 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: Do not synchronize with clock signal 1: Synchronize with clock signal.	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and ACK/NACK reception.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

SIMR2 selects how reception and transmission are controlled in simple IIC mode.

IICINTM bit (IIC Interrupt Mode Select)

The IICINTM bit selects the sources of interrupt requests in simple IIC mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCL clock signal is to be synchronized when the SCLn pin is driven low because a wait was inserted by another other device.

The SCL clock signal is not synchronized if the IICCSC bit is 0. The SCL clock signal is generated according to the rate selected in the BRR register regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set the IICACKT bit to 1 when ACK and NACK bits are received.

34.2.23 IIC Mode Register 3 (SIMR3)

Address(es): SCI0.SIMR3 4007 000Bh, SCI1.SIMR3 4007 002Bh, SCI2.SIMR3 4007 004Bh, SCI3.SIMR3 4007 006Bh, SCI4.SIMR3 4007 008Bh, SCI5.SIMR3 4007 00ABh, SCI6.SIMR3 4007 00CBh, SCI7.SIMR3 4007 00EBh, SCI8.SIMR3 4007 010Bh, SCI9.SIMR3 4007 012Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: Do not generate start condition 1: Generate start condition.*1, *3, *5, *6	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: Do not generate restart condition 1: Generate restart condition.*2, *3, *5, *6	R/W

Bit	Symbol	Bit name	Description	R/W
b2	IICSTPREQ	Stop Condition Generation	0: Do not generate stop condition 1: Generate stop condition.*2, *3, *5, *6	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: No requests are being made for generating conditions, or a condition is being generated 1: Generation of start, restart, or stop condition is complete. When 0 is written to IICSTIF, it is cleared to 0.*4	R/W*4
b5, b4	IICSDAS[1:0]	SDA Output Select	b5 b4 0 0: Output serial data 0 1: Generate start, restart, or stop condition 1 0: Output low on SDAn pin 1 1: Drive SDAn pin to high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SCL Output Select	b7 b6 0 0: Output serial clock 0 1: Generate start, restart, or stop condition 1 0: Output low on SCLn pin 1 1: Drive SCLn pin to high-impedance state.	R/W

Note 1. Only generate a start condition after checking the bus state and confirming that the bus is free.
Note 2. Generate a restart or stop condition after checking the bus state and confirming that the bus is busy.
Note 3. Do not set more than one of the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Write only 0. When 1 is written, the value is ignored.
Note 5. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 6. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICSTAREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of start condition generation.

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICRSTAREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of restart condition generation.

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICSTPREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of stop condition generation.

IICSTIF flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, the IICSTIF flag indicates that the condition generation is complete. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- On completion of a start, restart, or stop condition generation.

  If the setting condition conflicts with any of the clearing conditions for the flag, the clearing condition takes precedence.

[Clearing conditions]

- On writing 0 to the bit. After writing 0 to the IICSTIF bit, read the bit to check that it is actually set to 0.
- On writing 0 to the SIMR1.IICM bit when operation is not in simple IIC mode
- On writing 0 to the SCR.TE bit.

IICSDAS[1:0] bits (SDA Output Select)

The IICSDAS[1:0] bits control output from the SDAn pin. Set IICSDAS[1:0] and IICSCLS[1:0] to the same value.

IICSCLS[1:0] bits (SCL Output Select)

The IICSCLS[1:0] bits control output from the SCLn pin. Set IICSCLS[1:0] and IICSDAS[1:0] to the same value.

34.2.24 IIC Status Register (SISR)

Address(es): SCI0.SISR 4007 000Ch, SCI1.SISR 4007 002Ch, SCI2.SISR 4007 004Ch, SCI3.SISR 4007 006Ch, SCI4.SISR 4007 008Ch, SCI5.SISR 4007 00ACh, SCI6.SISR 4007 00CCh, SCI7.SISR 4007 00ECh, SCI8.SISR 4007 010Ch, SCI9.SISR 4007 012Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IICACKR
Value after reset:	0	0	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received.	R
b1	—	Reserved	This bit is read as 0.	R
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0.	R
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0.	R

SISR monitors the state in simple IIC mode.

IICACKR flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from the IICACKR flag. The IICACKR flag is updated on the rising edge of the SCLn clock for the received ACK/NACK bit.

34.2.25 SPI Mode Register (SPMR)

Address(es): SCI0.SPMR 4007 000Dh, SCI1.SPMR 4007 002Dh, SCI2.SPMR 4007 004Dh, SCI3.SCI3 4007 006Dh, SCI4.SPMR 4007 008Dh, SCI5.SPMR 4007 00ADh, SCI6.SPMR 4007 00CDh, SCI7.SCI7 4007 00EDh, SCI8.SPMR 4007 010Dh, SCI9.SPMR 4007 012Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SSE	SSn Pin Function Enable	0: Disable SSn pin function 1: Enable SSn pin function.	R/W*1
b1	CTSE	CTS Enable	0: Disable CTS function (enable RTS output function) 1: Enable CTS function.	R/W*1
b2	MSS	Master Slave Select	0: Transmit through TXDn pin and receive through RXDn pin (master mode) 1: Receive through TXDn pin and transmit through RXDn pin (slave mode).	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error.	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Do not invert clock polarity 1: Invert clock polarity.	R/W*1
b7	CKPH	Clock Phase Select	0: Do not delay clock 1: Delay clock.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

Note 2. Only 0 can be written to this bit, to clear the flag.

The SPMR register selects settings for simple SPI mode.

SSE bit (SSn Pin Function Enable)

Set the SSE bit to 1 to use the SSn pin to control transmission and reception in simple SPI mode. Set this bit to 0 in all other modes. In simple SPI mode, when master mode is selected (SCR.CKE[1:0] = 00b and SPMR.MSS = 0) and there is a single master, the SSn pin on the master side is not required to control reception and transmission. In such a case, set the SSE bit to 0. Do not set both the SSE and CTSE bits to 1. If this setting is made, operation is the same as that when these bits are set to 0.

CTSE bit (CTS Enable)

Set the CTSE bit to 1 if the SSn pin is to be used for inputting the CTS control signal to control transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple IIC mode. Do not set both the CTSE and SSE bits to 1. If this setting is made, operation is the same as that when these bits are set to 0.

MSS bit (Master Slave Select)

The MSS bit selects master or slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when this bit is set to 1, so that data is received through the TXDn pin and transmitted through the RXDn pin. Set this bit to 0 in modes other than simple SPI mode.

MFF flag (Mode Fault Flag)

The MFF flag indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading this flag.

[Setting condition]

- When input on the SSn pin is low during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0).

[Clearing condition]

- On writing 0 to the bit after it is read as 1.

CKPOL bit (Clock Polarity Select)

The CKPOL bit selects the polarity of the clock signal output through the SCKn pin. See Figure 34.70 for details. Set the CKPOL bit to 0 in all modes other than simple SPI mode and clock synchronous mode.

CKPH bit (Clock Phase Select)

The CKPH bit selects the phase of the clock signal output through the SCKn pin. See Figure 34.70 for details. Set the CKPH bit to 0 in all modes other than simple SPI mode and clock synchronous mode.

34.2.26 FIFO Control Register (FCR)

Address(es): SCI0.FCR 4007 0014h, SCI1.FCR 4007 0034h, SCI2.FCR 4007 0054h, SCI3.FCR 4007 0074h, SCI4.FCR 4007 0094h, SCI5.FCR 4007 00B4h, SCI6.FCR 4007 00D4h, SCI7.FCR 4007 00F4h, SCI8.FCR 4007 0114h, SCI9.FCR 4007 0134h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RSTRG[3:0]				RTRG[3:0]				TTRG[3:0]				DRES	TFRST	RFRST	FM
Value after reset:	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FM	FIFO Mode Select	Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode: 0: Non-FIFO mode. Selects TDR/RDR or TDRHL/RDRHL for communication. 1: FIFO mode. Selects FTDRHL/FRDRHL for communication.	R/W*1
b1	RFRST	Receive FIFO Data Register Reset	Valid only when FCR.FM = 1: 0: Do not reset FRDRHL 1: Reset FRDRHL.	R/W
b2	TFRST	Transmit FIFO Data Register Reset	Valid only when FCR.FM = 1: 0: Do not reset FTDRHL 1: Reset FTDRHL.	R/W
b3	DRES	Receive Data Ready Error Select Bit	Selects the interrupt requested when detecting receive data ready: 0: Receive data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI).	R/W
b7 to b4	TTRG[3:0]	Transmit FIFO Data Trigger Number	Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode: 0000: Trigger number 0 : 1111: Trigger number 15.	R/W
b11 to b8	RTRG[3:0]	Receive FIFO Data Trigger Number	Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode: 0000: Trigger number 0 : 1111: Trigger number 15.	R/W
b15 to b12	RSTRG[3:0]	RTS Output Active Trigger Number Select	Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1, SPMR.CTSE = 0, and SPMR.SSE = 0: 0000: Trigger number 0 : 1111: Trigger number 15.	R/W

Note 1. Writable only when TE = 0 and RE = 0.

FCR selects FIFO mode, resets FTDRHL and FRDRHL, selects the FIFO data trigger number for transmission or reception, and selects the RTS output active trigger number.

FM bit (FIFO Mode Select)

When the FM bit is set to 1, FTDRHL and FRDRHL are selected for communication. When the FM bit is set to 0, TDR and RDR, or TDRHL and RDRHL are selected for communication.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the FRDRHL register is reset and the received data count resets to 0. When 1 is written to the RFRST bit, it clears to 0 after 1 PCLKA.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the FTDRHL register is reset and the transmit data count resets to 0. When 1 is written to the TFRST bit, it clears to 0 after 1 PCLKA.

DRES bit (Receive Data Ready Error Select Bit)

When detecting a receive data ready error, the selection can be made from an SCIn_RXI interrupt request or an SCIn_ERI interrupt request. When starting DMAC or DTC and reading from the FRDRH and FRDRL registers, set the DRES bit to 1.

TTRG[3:0] bits (Transmit FIFO Data Trigger Number)

The TDFE flag is set to 1 when the amount of transmit data in FTDRHL is equal to or less than the transmit triggering number specified in the TTRG[3:0] bits, and software can write data to FTDRHL. If SCR.TIE = 1, an SCIn_TXI interrupt request occurs.

RTRG[3:0] bits (Receive FIFO Data Trigger Number)

The RDF flag is set to 1 when the amount of receive data in FRDRHL is equal to or greater than the receive triggering number specified in the RTRG[3:0] bits, and software can read data from FRDRHL. If SCR.RIE = 1, an SCIn_RXI interrupt request occurs.

When RTRG[3:0] is 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0, and an SCIn_RXI interrupt does not occur.

RSTRG[3:0] bits (RTS Output Active Trigger Number Select)

When the amount of receive data stored in FRDRHL is equal to or greater than the receive triggering number specified in the RSTRG[3:0] bits, the RTS signal goes high.

When RSTRG[3:0] is 0, the RTS signal does not go high even when the amount of data in FRDRHL is equal to 0.

34.2.27 FIFO Data Count Register (FDR)

Address(es): SCI0.FDR 4007 0016h, SCI1.FDR 4007 0036h, SCI2.FDR 4007 0056h, SCI3.FDR 4007 0076h,
SCI4.FDR 4007 0096h, SCI5.FDR 4007 00B6h, SCI6.FDR 4007 00D6h, SCI7.FDR 4007 00F6h,
SCI8.FDR 4007 0116h, SCI9.FDR 4007 0136h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	T[4:0]					—	—	—	R[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	R[4:0]	Receive FIFO Data Count	Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1. Indicates the amount of receive data stored in FRDRHL.	R
b7 to b5	—	Reserved	These bits are read as 0.	R

Bit	Symbol	Bit name	Description	R/W
b12 to b8	T[4:0]	Transmit FIFO Data Count	Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1. Indicates the amount of non-transmitted data stored in FTDRHL.	R
b15 to b13	—	Reserved	These bits are read as 0.	R

The FDR register indicates the amount of data stored in FRDRHL and FTDRHL.

R[4:0] bits (Receive FIFO Data Count)

The R[4:0] bits indicate the amount of receive data stored in FRDRHL. 00h means no receive data, and 10h means that the maximum received data is stored in FRDRHL.

T[4:0] bits (Transmit FIFO Data Count)

The T[4:0] bits indicate the amount of non-transmitted data stored in FTDRHL. 00h means no transmit data, and 10h means that all (maximum amount) of the data to be transmitted is stored in FTDRHL.

34.2.28 Line Status Register (LSR)

Address(es): SCI0.LSR 4007 0018h, SCI1.LSR 4007 0038h, SCI2.LSR 4007 0058h, SCI3.LSR 4007 0078h,
SCI4.LSR 4007 0098h, SCI5.LSR 4007 00B8h, SCI6.LSR 4007 00D8h, SCI7.LSR 4007 00F8h,
SCI8.LSR 4007 0118h, SCI9.LSR 4007 0138h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PNUM[4:0]					—	FNUM[4:0]					—	ORER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ORER	Overrun Error Flag	Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, and when FIFO is selected: 0: No overrun error occurred 1: Overrun error occurred.	R*[1]
b1	—	Reserved	This bit is read as 0.	R
b6 to b2	FNUM[4:0]	Framing Error Count	Indicates the amount of data with a framing error in the receive data stored in FRDRHL.	R
b7	—	Reserved	This bit is read as 0.	R
b12 to b8	PNUM[4:0]	Parity Error Count	Indicates the amount of data with a parity error in the receive data stored in FRDRHL.	R
b15 to b13	—	Reserved	These bits are read as 0.	R

Note 1. Write 0 to SSR_FIFO.ORER to clear the flag.

The LSR register indicates the receive error status.

ORER bit (Overrun Error Flag)

The ORER bit reflects the value in SSR_FIFO.ORER.

FNUM[4:0] bits (Framing Error Count)

The FNUM[4:0] value indicates the amount of data with a framing error stored in the FRDRHL register.

PNUM[4:0] bits (Parity Error Count)

The PNUM[4:0] value indicates the amount of data with a parity error stored in the FRDRHL register.

34.2.29 Compare Match Data Register (CDR)

Address(es): SCI0.CDR 4007 001Ah, SCI1.CDR 4007 003Ah, SCI2.CDR 4007 005Ah, SCI3.CDR 4007 007Ah, SCI4.CDR 4007 009Ah, SCI5.CDR 4007 00BAh, SCI6.CDR 4007 00DAh, SCI7.CDR 4007 00FAh, SCI8.CDR 4007 011Ah, SCI9.CDR 4007 013Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPD[8:0]	Compare Match Data	Holds compare data pattern for address match wakeup function.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CDR register sets the compare data for the address match function.

CMPD[8:0] bits (Compare Match Data)

The CMPD[8:0] bits set the data to be compared to receive data for the address match function, when the address match function is enabled (DCCR.DCME = 1).

Three bit lengths are available:

- CMPD[6:0] with 7-bit length
- CMPD[7:0] with 8-bit length
- CMPD[8:0] with 9-bit length.

34.2.30 Data Compare Match Control Register (DCCR)

Address(es): SCI0.DCCR 4007 0013h, SCI1.DCCR 4007 0033h, SCI2.DCCR 4007 0053h, SCI3.DCCR 4007 0073h, SCI4.DCCR 4007 0093h, SCI5.DCCR 4007 00B3h, SCI6.DCCR 4007 00D3h, SCI7.DCCR 4007 00F3h, SCI8.DCCR 4007 0113h, SCI9.DCCR 4007 0133h

	b7	b6	b5	b4	b3	b2	b1	b0
	DCME	IDSEL	—	DFER	DPER	—	—	DCMF
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DCMF	Data Compare Match Flag	0: Not matched 1: Matched.	R/(W)*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b3	DPER	Data Compare Match Parity Error Flag	0: No parity error occurred 1: Parity error occurred.	R/(W)*1
b4	DFER	Data Compare Match Framing Error Flag	0: No framing error occurred 1: Framing error occurred.	R/(W)*1
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	IDSEL	ID Frame Select	Valid only in asynchronous mode, including multi-processor mode: 0: Always compare data regardless of the MPB bit value 1: Only compare data when MPB bit = 1 (ID frame).	R/W
b7	DCME	Data Compare Match Enable	Valid only in asynchronous mode, including multi-processor mode: 0: Disable address match function 1: Enable address match function.	R/W

Note 1. Only 0 can be written, to clear the flag after reading 1.

The DCCR register controls the address match function.

DCMF flag (Data Compare Match Flag)

The DCMF flag indicates that the SCI detected a receive data match with the comparison data (CDR.CMPD).

[Setting condition]

- On match of the comparison data (CDR.CMPD) with the receive data when DCCR.DCME = 1.

[Clearing condition]

- When 0 is written after 1 is read from DCMF.

Clearing the SCR.RE bit to 0 does not affect the DCMF flag, which retains its previous value.

DPER flag (Data Compare Match Parity Error Flag)

The DPER flag indicates that a parity error occurred on address match detection (receive data match detection).

[Setting condition]

- When a parity error is detected in a frame in which an address match is detected.

[Clearing conditions]

- When 0 is written after 1 is read from DPER.

When the SCR.RE bit is set to 0 (serial reception is disabled), the DPER flag is not affected and retains its previous value.

DFER flag (Data Compare Match Framing Error Flag)

The DFER flag indicates that a framing error occurred on address match detection (receive data match detection).

[Setting conditions]

- When a stop bit of a frame in which an address match is detected is 0.

  When in 2-stop-bit mode, only the first bit of the stop bits is checked for a value of 1 (the second stop bit is not checked).

[Clearing conditions]

- When 0 is written after 1 is read from DFER.

When the SCR.RE bit is set to 0 (serial reception is disabled), the DFER flag is not affected and retains its previous value.

IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data regardless of the MPB bit value or to compare data only when MPB = 1 (ID frame), when the address match function is enabled.

DCME bit (Data Compare Match Enable)

The DCME bit enables or disables the address match function (data compare match function).

If the SCI detects a match to the comparison data (CDR.CMPD) with the receive data, the DCME bit clears automatically, after which SCI operation mode is in receive mode without data compare match function. See section 34.3.6, Address Match (Receive Data Match Detection) Function.

The write value must be 0 for all modes other than asynchronous mode.

34.2.31 Serial Port Register (SPTR)

Address(es): SCI0.SPTR 4007 001Ch, SCI1.SPTR 4007 003Ch, SCI2.SPTR 4007 005Ch, SCI3.SPTR 4007 007Ch, SCI4.SPTR 4007 009Ch, SCI5.SPTR 4007 00BCh, SCI6.SPTR 4007 00DCh, SCI7.SPTR 4007 00FCh, SCI8.SPTR 4007 011Ch, SCI9.SPTR 4007 013Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPB2IO	SPB2DT	RXDMON
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b0	RXDMON	Serial Input Data Monitor	Indicates the state of the RXDn pin: 0: RXDn pin is low 1: RXDn pin is high.	R
b1	SPB2DT	Serial Port Break Data Select	Selects the output level of the TXDn pin when SCR.TE = 0: 0: Output low level on TXDn pin 1: Output high level on TXDn pin.	R/W
b2	SPB2IO	Serial Port Break I/O	Selects whether the value of SPB2DT is output to TXDn pin: 0: Do not output value of SPB2DT bit on TXDn pin 1: Output value of SPB2DT bit on TXDn pin.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPTR register provides confirmation of the serial reception pin (RXDn pin) status and sets the transmission pin (TXDn pin) status.

This register can only be used in asynchronous mode.

The TXDn pin status is determined by the combination of SCR.TE, SPTR.SPB2IO, and SPTR.SPB2DT settings, as shown in Table 34.22.

Table 34.22 TXDn pin status

Value of SCR.TE	Value of SPTR.SPB2IO	Value of SPTR.SPB2DT	TXDn pin status
0	0	x	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	x	x	Serial transmit data is output

x: Don't care.

Note: Use the SPTR register in asynchronous mode only. Using this register in any other mode is not guaranteed.

34.3 Operation in Asynchronous Mode

Figure 34.2 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit or receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communications line is held in the mark state (high level) when not communicating.

The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

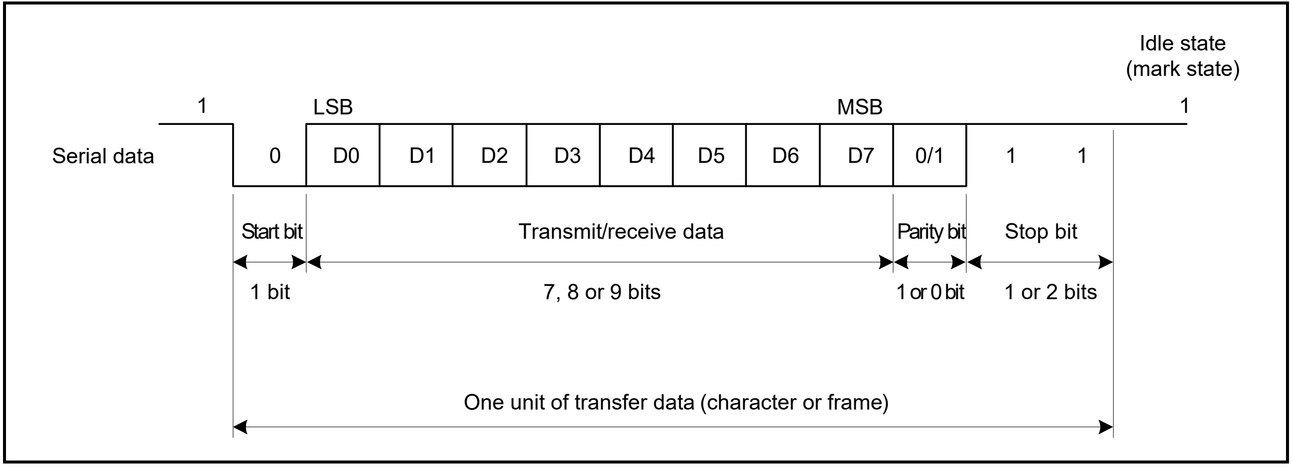


Figure 34.2 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

34.3.1 Serial Data Transfer Format

Table 34.23 lists the serial data transfer formats that can be used in asynchronous mode. Any of 18 transfer formats can be selected with the SMR and SCMR register settings. For details on the multi-processor function, see section 34.4, Multi-Processor Communication Function.

Table 34.23 Serial transfer formats in asynchronous mode (1 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP

Table 34.23 Serial transfer formats in asynchronous mode (2 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length												
CHR1	CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	0	0	S	8-bit data								P	STOP		
1	0	1	0	1	S	8-bit data								P	STOP	STOP	
1	1	0	0	0	S	7-bit data							STOP				
1	1	0	0	1	S	7-bit data							STOP	STOP			
1	1	1	0	0	S	7-bit data							P	STOP			
1	1	1	0	1	S	7-bit data							P	STOP	STOP		
0	0	—	1	0	S	9-bit data									MPB	STOP	
0	0	—	1	1	S	9-bit data									MPB	STOP	STOP
1	0	—	1	0	S	8-bit data								MPB	STOP		
1	0	—	1	1	S	8-bit data								MPB	STOP	STOP	
1	1	—	1	0	S	7-bit data							MPB	STOP			
1	1	—	1	1	S	7-bit data							MPB	STOP	STOP		

S: Start bit
STOP: Stop bit
P: Parity bit

MPB: Multi-processor bit

34.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs synchronization.

Because receive data is sampled on the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in Figure 34.3. The reception margin in asynchronous mode is determined by the following formula (1):

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)\, F - \frac{\left| D - 0.5 \right|}{N} (1 + F) \right| \times 100\ [\%] \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SEMR.ABCSE = 0 and SEMR.ABCS = 0. N = 8 when SEMR.ABCS = 1. N = 6 when SEMR.ABCSE = 1.)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\%$$

This represents the computed value. Renesas recommends a margin of 20% to 30% in system design.

Note 1. In this example, the SEMR.ABCS bit is 0 and the SEMR.ABCSE is 0. When the ABCS bit is 1 and the ABCSE bit is 0, a frequency of 8 times the bit rate is used as a base clock, and receive data is sampled on the rising edge of the 4th pulse of the base clock.
When the ABCSE bit is 1, a sextuple frequency of a bit rate is used as a base clock, and receive data is sampled on the rising edge of the 3rd pulse of the base clock.

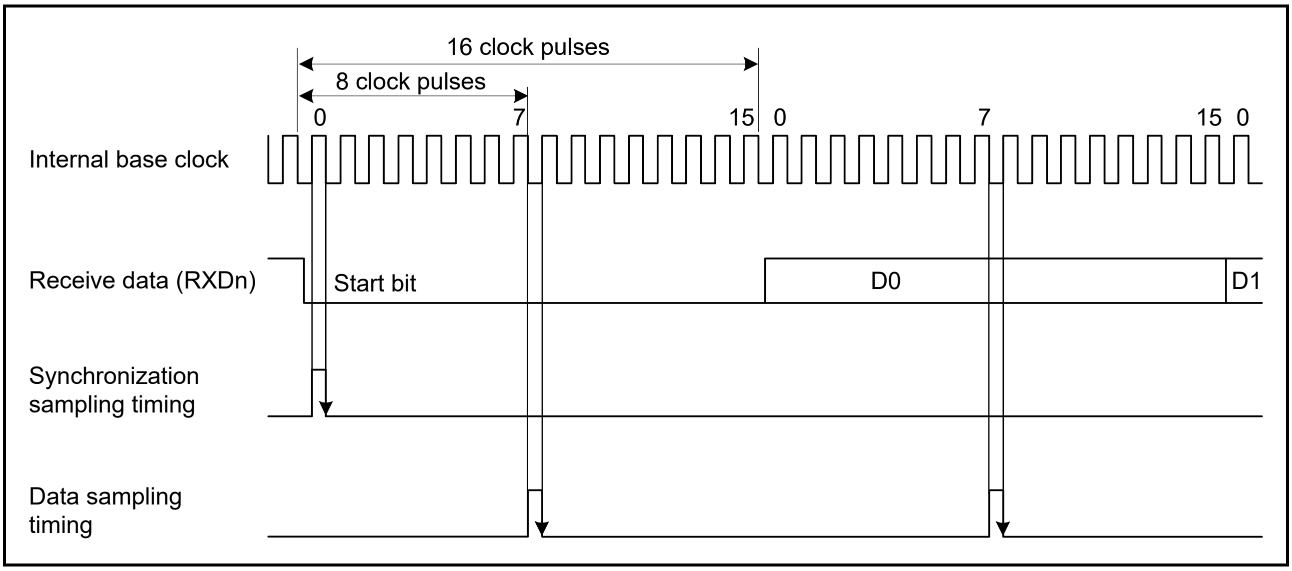


Figure 34.3 Receive data sampling timing in asynchronous mode

34.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the transfer clock of the SCI, based on the SMR.CM and SCR.CKE[1:0] settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when SEMR.ABCS = 0) or 8 times the bit rate (when SEMR.ABCS = 1).

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is configured so that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 34.4.

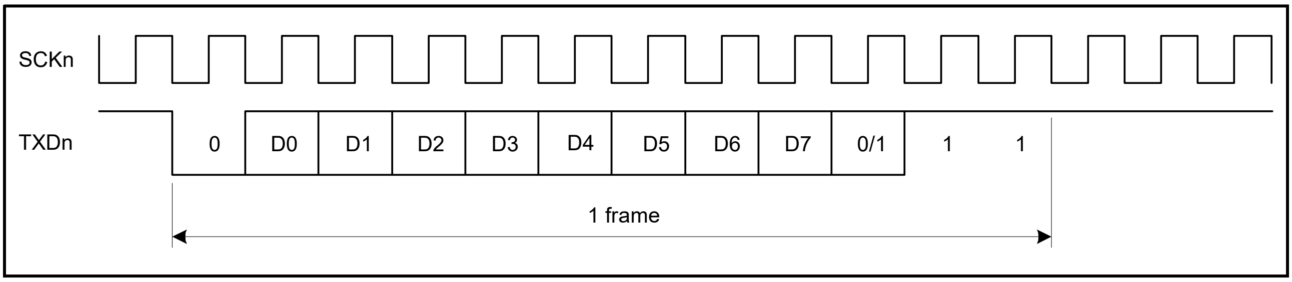


Figure 34.4 **Phase relationship between output clock and transmit data in asynchronous mode when SMR.CHR = 0, PE = 1, MP = 0, and STOP = 1**

34.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate

When the SEMR.ABCS bit is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0. When the SEMR.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double that of when BGDM is set to 0. When the SCR.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate four times that when the ABCS and BGDM bits are set to 0. When the SEMR.ABCSE bit is set to 1, the number of basic clock pulses is 6 during a period of 1 bit, and the SCI operates at a bit rate 16/3 times that when SEMR.ABCS = 0, SEMR.BGDM = 0, and SMER.ABCSE = 0.

As shown by Formula (1) in section 34.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when the SEMR.ABCS or SEMR.ABCSE bit in SEMR is set to 1. Therefore, if the target bit rate can be obtained with ABCS or ABCSE set to 0, it is recommended that you use the SCI with ABCS and ABCSE set to 0.

34.3.5 CTS and RTS Functions

The CTS function uses input on the CTSn_RTSn pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing a low level on the CTSn_RTSn pin causes transmission to start.

Driving the CTSn_RTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function, which uses output on the CTSn_RTSn pin, a low level is output when reception becomes possible. Conditions for output of the low and high levels are shown in this section.

[Conditions for low-level output]

(a) Non-FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit is 1
- Reception is not in progress
- There is no received data yet to be read
- The ORER, FER, and PER flags in the SSR register are all 0.

(b) FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit is 1
- The amount of receive data written in FRDRHL is equal to or less than the specified receive triggering number
- The ORER bit in the SSR_FIFO register (ORER in FRDRH) is 0.

[Condition for high-level output]

(a Non-FIFO selected

- The conditions for low-level output are not satisfied
- When reception is terminated with SCR.RE = 0 without reading the RDR register after reception is complete, RTS remains high. At this time, read the SCR register for dummy values after writing 0 to SCR.RE.

(b) FIFO selected

- The conditions for low-level output are not satisfied.

34.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in asynchronous mode.

If the DCCR.DCME bit is set to 1*4, when one frame of data is received, the SCI compares that received data with the data set in CDR.CMPD. If the SCI detects a match to the comparison data (CDR.CMPD*3) with the received data, the SCI can issue the SCIn_RXI interrupt request.

If the SMR.MP bit is set to 0, comparison occurs only for valid data in receive format. In multi-processor mode (SMR.MP bit = 1), if the DCCR.IDSEL bit is set to 1, receive data where the MPB bit is 1 is subject to comparison for address match and receive data where the MPB bit is 0 is always treated as a mismatch.

If the DCCR.IDSEL bit is set to 0, SCI performs address match detection regardless of the MPB bit value of the received data. Until SCI detects a match between the comparison data (CDR.CMPD*3) and the receive data, the received data is skipped (discarded), and the SCI cannot detect a parity error or framing error.

When SCI detects a match, the DCCR.DCME bit is automatically cleared, and the DCCR.DCMF flag is set to 1. If the DCCR.IDSEL bit is set to 1, the SCR.MPIE bit is automatically cleared. If DCCR.IDSEL is set to 0, the value of the SCR.MPIE bit is retained. If the SCR.RIE bit is set to 1, the SCI issues an SCIn_RXI interrupt request.

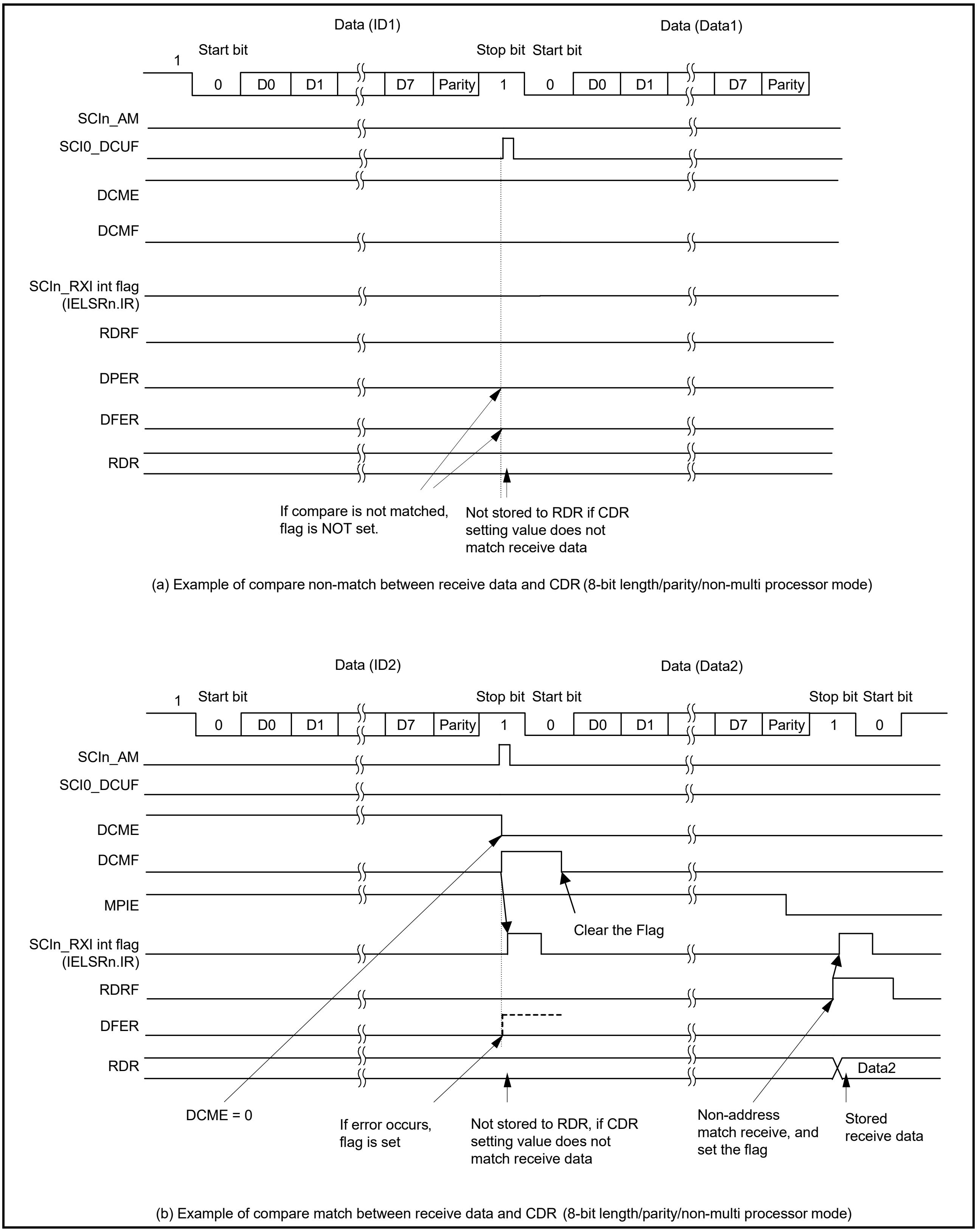
If the SCI detects a framing error in the receive data for which a match is detected, the DCCR.DFER bit is set to 1, and if the SCI detects a parity error in that frame, the DCCR.DPER bit is set to 1. The compared receive data is not stored in the RDR register*1, and SSR.RDRF remains 0.*2

After the SCI detects a match, and DCCR.DCME is automatically cleared, the SCI receives the next data continuously based on the current register setting.

When the DCCR.DFER or DCCR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the DCCR.DFER and DCCR.DPER flags to 0.

Examples of the address match function are shown in Figure 34.5 and Figure 34.6.

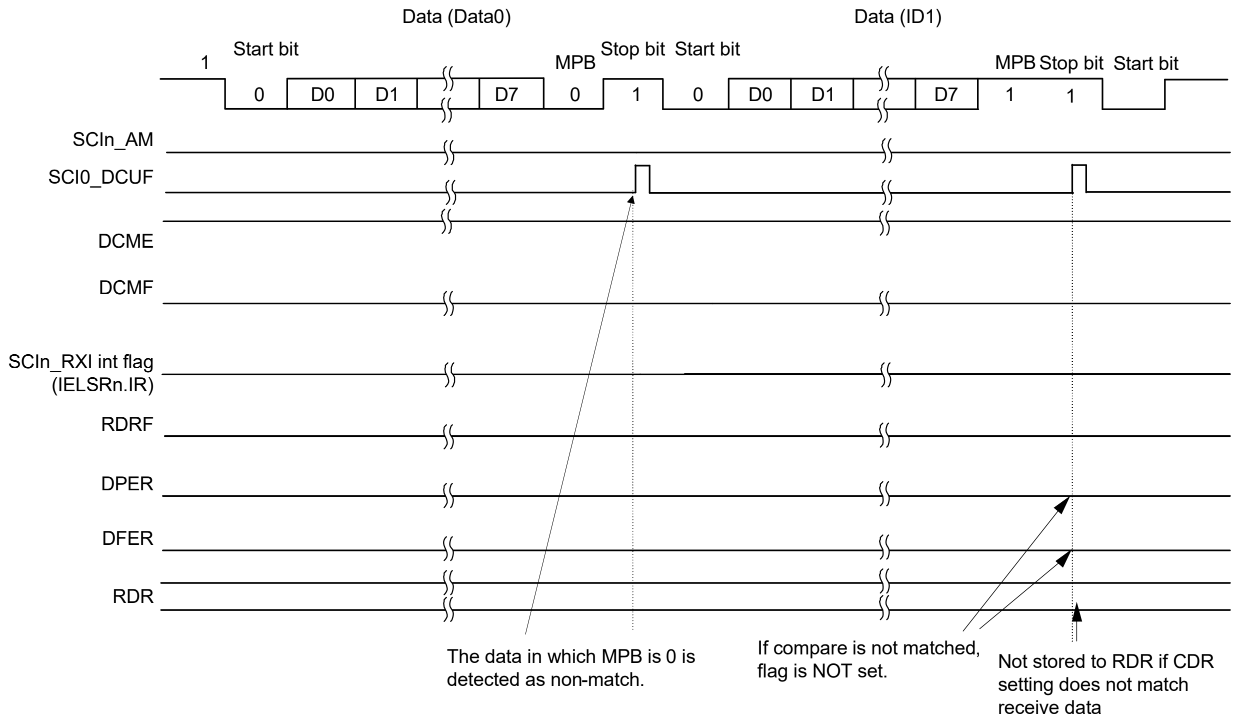
Note 1. When FCR.FM = 1, this refers to the FRDRHL register.
Note 2. When FCR.FM = 1, this refers to the SSR_FIFO.RDF flag.
Note 3. This comparative target can select one length of 3 types: CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.
Note 4. Set the DCCR.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.



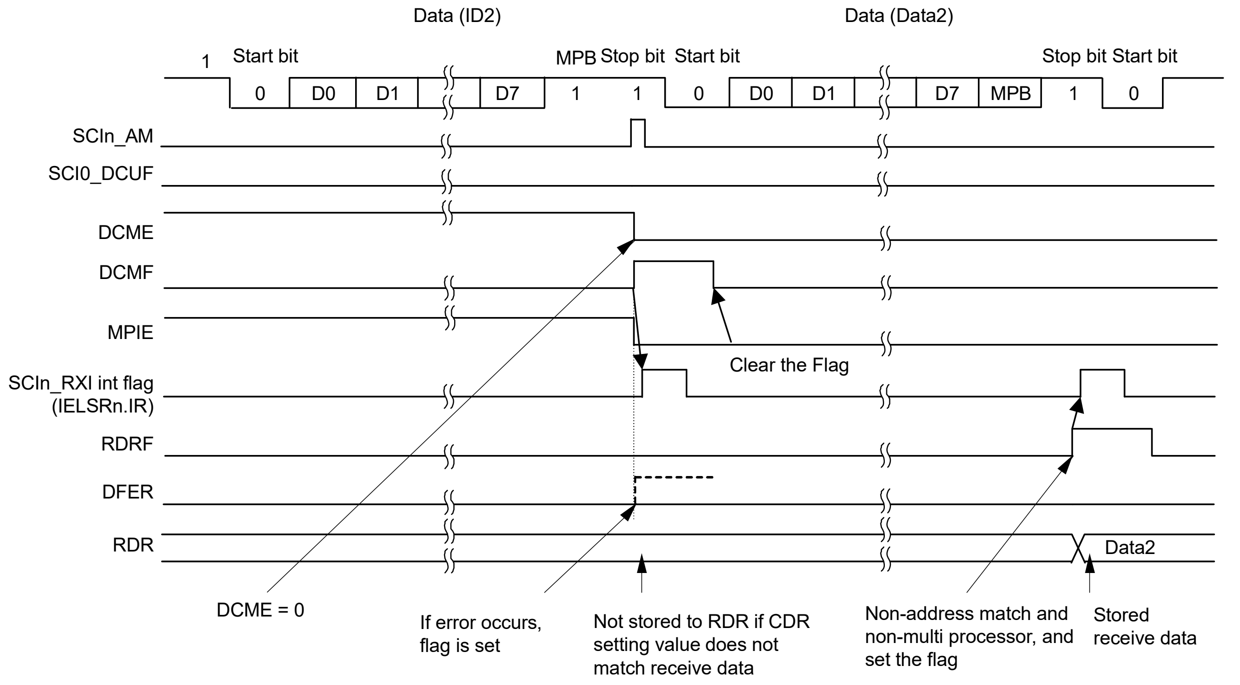
(a) Example of compare non-match between receive data and CDR (8-bit length/parity/non-multi processor mode)

(b) Example of compare match between receive data and CDR (8-bit length/parity/non-multi processor mode)

Figure 34.5 **Example of address match (1) non-multi processor mode**



(a) Example of compare non-match between receive data and CDR (8-bit length/IDSEL = 1/multi-processor mode)



(b) Example of compare match between receive data and CDR (8-bit length/IDSEL = 1/multi-processor mode)

Figure 34.6 Example of address match (2) multi-processor mode

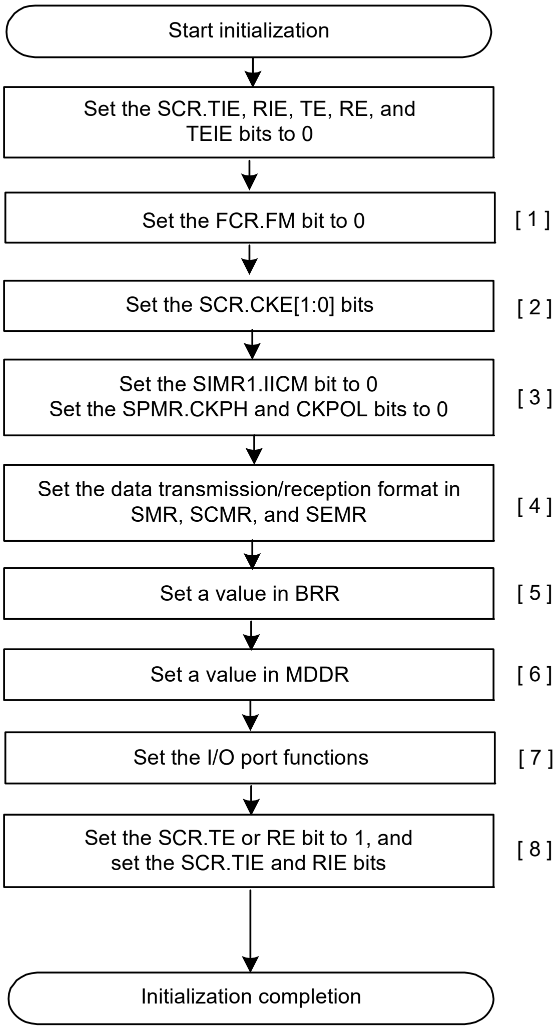
34.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register, then continue through the SCI initialization procedure (select non-FIFO or FIFO) shown in Figure 34.7 and Figure 34.8. Whenever the operating mode or transfer format is to be changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: When the SCR.RE bit is set to 0, the ORER, FER, RDRF, RDF, PER, and DR flags in SSR/SSR_FIFO, and the RDR and RDRHL registers are not initialized. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: Switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.



[1] Set the FCR.FM bit to 0.

[2] Set the clock selection in SCR.
When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits to 0.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

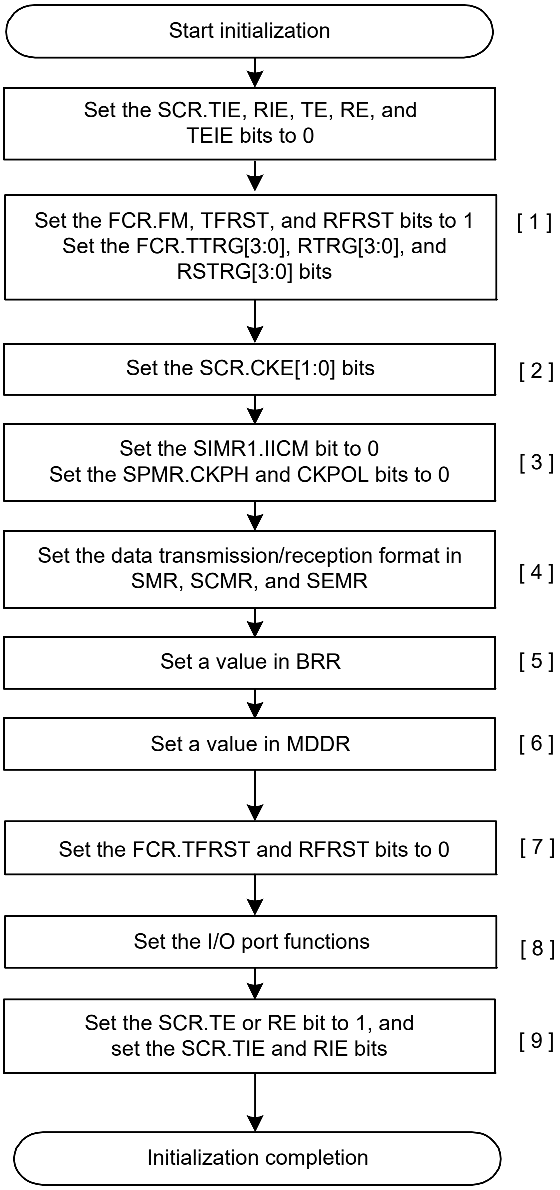
[5] Write a value corresponding to the bit rate to BRR.
This step is not necessary if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Figure 34.7 Example flow of SCI initialization in asynchronous mode with non-FIFO selected



[1] Set the FCR.FM, TFRST, and RFRST bits to 1 (FIFO mode enabled, transmit/receive FIFOs empty). Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.

[2] Set the clock selection in SCR. When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.

[3] Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0. Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.

[7] Set the FCR.TFRST and RFRST bits to 0.

[8] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[9] Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn to be used.

Figure 34.8 Example flow of SCI initialization in asynchronous mode with FIFO selected

34.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

Figure 34.9, Figure 34.10, and Figure 34.11 show examples of serial transmission in asynchronous mode.

In serial transmission, the SCI operates as described in this section. When the SCR.TE bit is set to 1, the high level for one frame (preamble) is output to TXD.

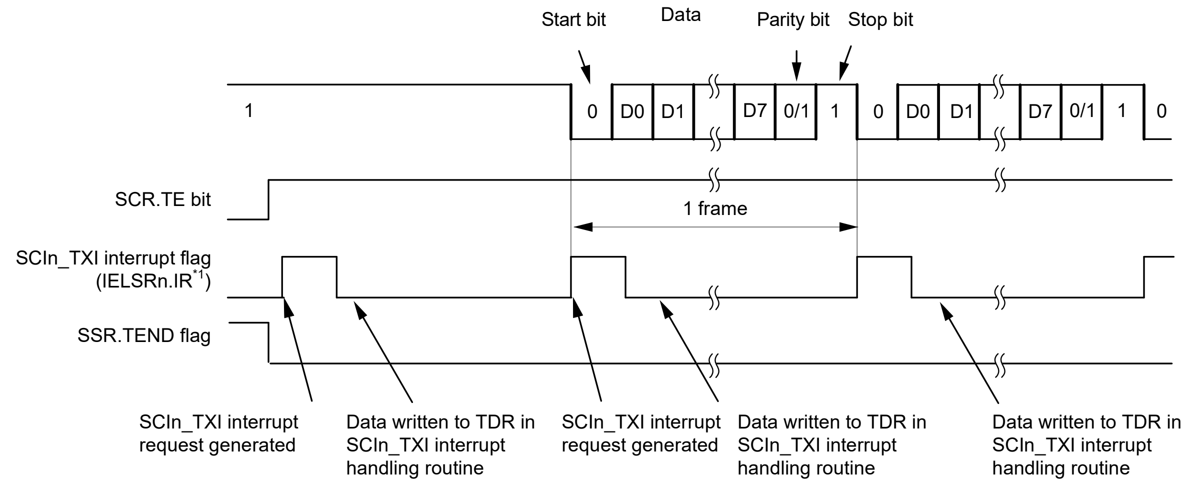
1. The SCI transfers data from the TDR^{*1} register to the TSR register when data is written to TDR^{*1} in the SCIn_TXI interrupt handling routine.
   The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE and SCR.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR^{*1} register to the TSR register. If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to the TDR^{*1} register in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is

complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR*1 register from the handling routine for SCIn_TXI requests.

3. Data is sent from the TXDn pin in the following order:
    - Start bit
    - Transmit data
    - Parity bit or multi-processor bit (can be omitted depending on the format)
    - Stop bit.
4. The SCI checks for update of the TDR register on output of the stop bit.
5. When the TDR register is updated, setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from the TDR*1 register to the TSR register and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, in which 1 is output. If the SCR.TEIE bit is 1, the SSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

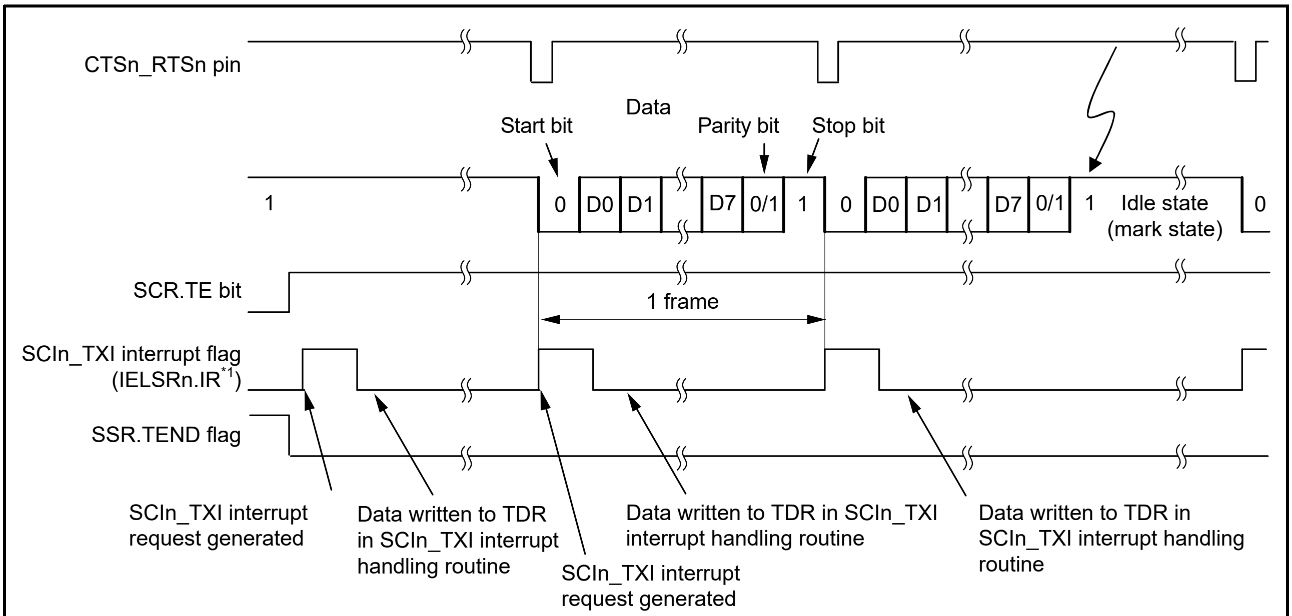
Note 1. Only write data to the TDRHL register when 9-bit data length is selected.

Figure 34.9, Figure 34.10, and Figure 34.11 show examples of serial transmission in asynchronous mode.



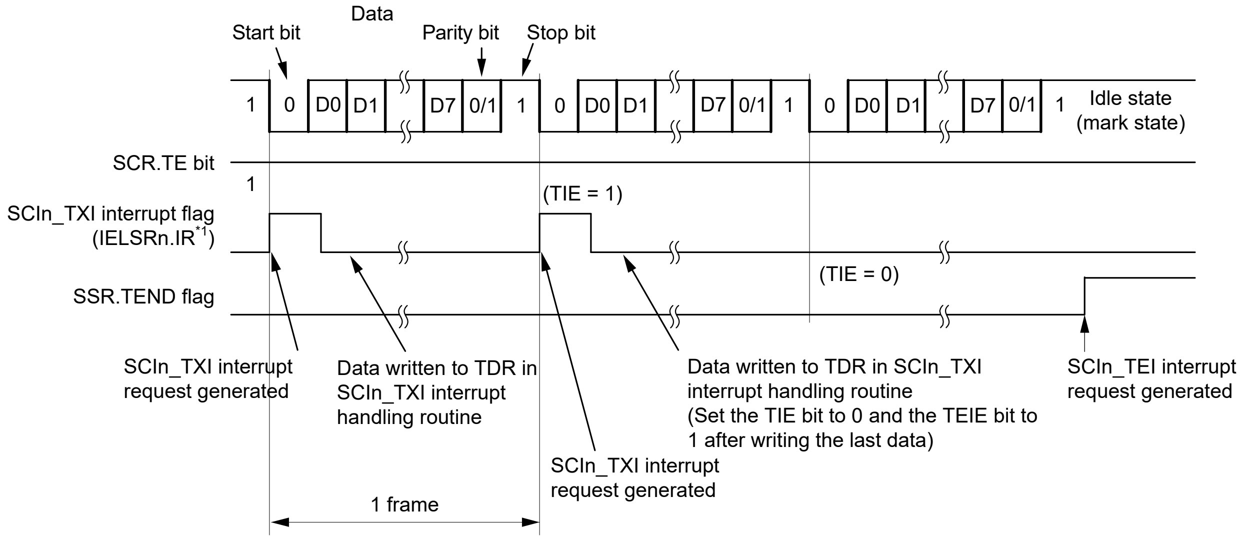
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.9 Example operation for serial transmission in asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission



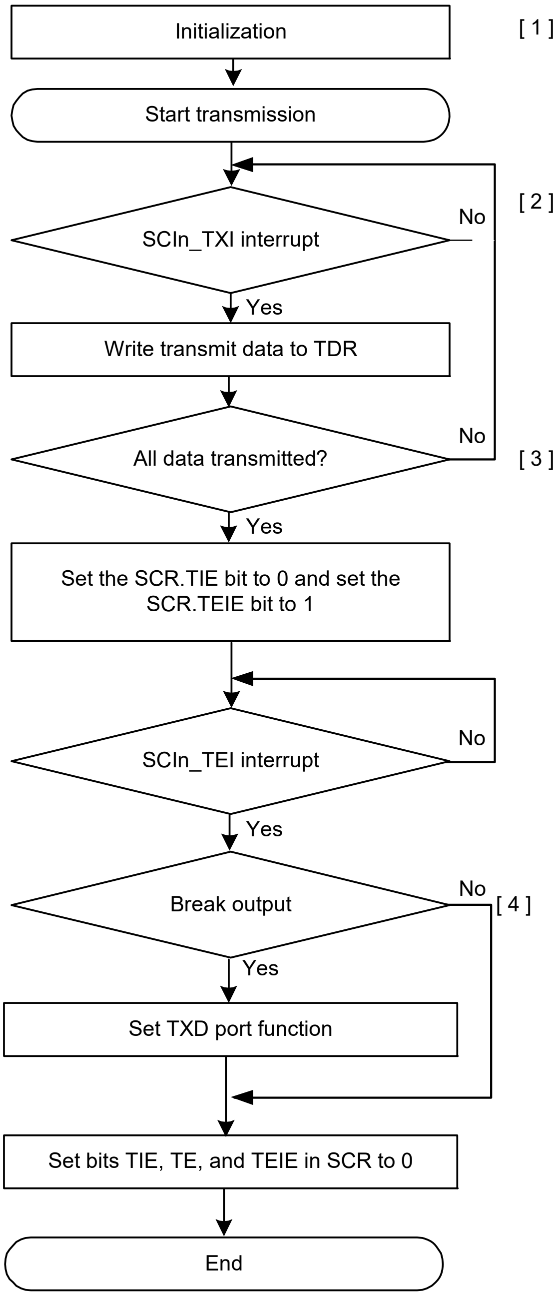
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.10 Example operation for serial transmission in asynchronous mode (2) with 8-bit data, parity bit, one stop bit, CTS function used, and at the beginning of transmission



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.11 Example operation for serial transmission in asynchronous mode (3) with 8-bit data, parity bit, one stop bit, CTS function not used, and from the middle of transmission until transmission completion



[1] SCI Initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a Frame (preamble), and transmission is enabled.

[2] Transmit data write to TDR by an SCIn_TXI interrupt request. When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (SCIn_TXI) request is generated. Write transmit data to TDR once in the SCIn_TXI interrupt processing routine.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR once using an SCIn_TXI interrupt request.
Transmit data can also be written to TDR by activating the DMAC or DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXDn, pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

Note: The TDR register becomes the TDRHL register when 9-bit data length is selected.

Figure 34.12 Example flow of serial transmission in asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 34.13 shows an example of a data format that is written to FTDRH and FTDRL in asynchronous mode.

Data corresponding to the data length is set to FTDRH and FTDRL. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
			FTDRH								FTDRL							
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	—	—	—	—	—	—	—	—	7-bit transmit data						
8 bits	1	1	—	—	—	—	—	—	—	—	8-bit transmit data							
9 bits	0	Don't care	—	—	—	—	—	—	—	9-bit transmit data								

—: Invalid. The write value should be 0.

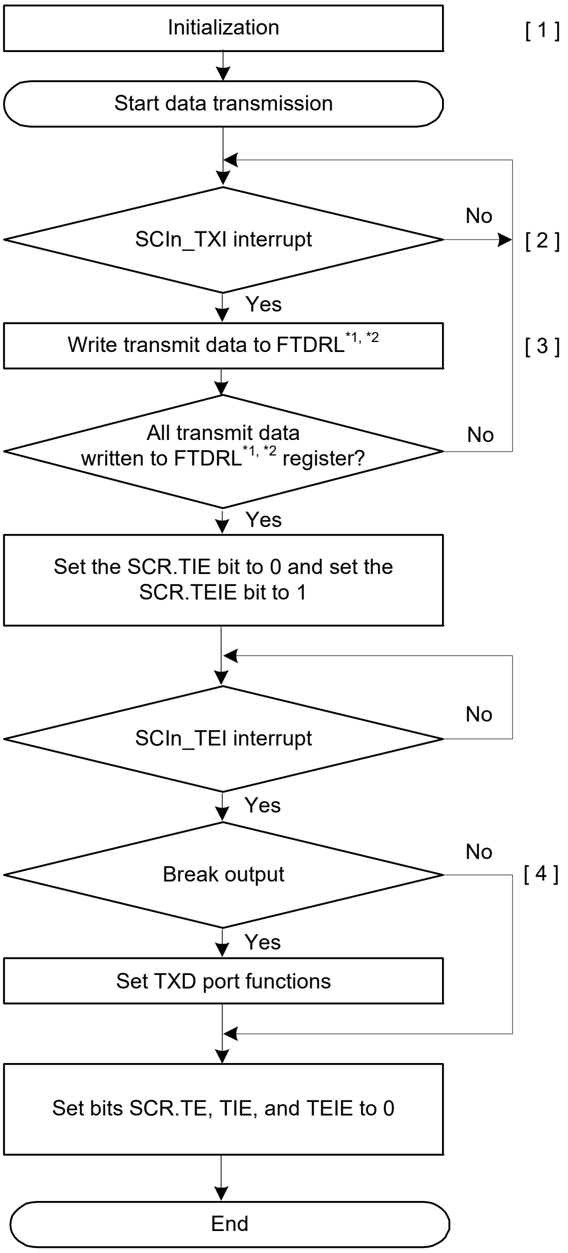
Figure 34.13 Data format written to FTDRH and FTDRL with FIFO selected

In serial transmission, the SCI operates as described in this section. When the TE bit is set to 1, the high level is output to TXD for one frame (preamble).

1. The SCI transfers data from the FTDRL*1 register to the TSR register when data is written to FTDRL*1 in the SCIn_TXI interrupt handling routine. The amount of data that can be written to FTDRL is 16 minus FDR.T[4:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE and SCR.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the FTDRL*1 register to the TSR register. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, SSR_FIFO.TDFE is set to 1. If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to FTDRL*1 in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the FTDRL*1 *2 register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
    - Start bit
    - Transmit data
    - Parity bit or multi-processor bit (can be omitted depending on the format)
    - Stop bit.
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in the FTDRL*3 register.
5. When data is set to FTDRL*3, setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from FTDRL*1 to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in FTDRL*3, the TEND flag in SSR_FIFO is set to 1, the stop bit is sent, and the mark state is entered in which 1 is output. If the SCR.TEIE bit is 1, the SSR_FIFO.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Note 1. Write data not to FTDRL but to the FTDRH and FTDRL registers.
Note 2. Write data in order from FTDRH to FTDRL when 9-bit data length is selected.
Note 3. The SCI only checks for update to the FTDRL register and not the FTDRH register when 9-bit data length is selected.

Figure 34.14 shows an example flow of serial transmission in asynchronous mode with FIFO selected.



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame (preamble), and transmission is enabled.

[2] Transmit data write to FTDRL*1 by an SCIn_TXI interrupt request:
When transmit data is transferred from FTDRL to TSR, when the quantity of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, a transmit data FIFO empty interrupt (SCIn_TXI) request is generated. Write transmit data to FTDRL*1, *2 once in the SCIn_TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to FTDRL using an SCIn_TXI interrupt request and clear the SSR_FIFO.TDFE flag to 0. The number of transmission data it is possible to write in is 16 - (the number of stored transmit FIFO data).
Transmit data can also be written to FTDRL by activating the DMAC or DTC. When writing the data to FTDRL by DMAC or DTC, the TDFE flag is cleared automatically, so do not write to the TDFE flag.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to FTDRL.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXDn pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

Note 1. When data length is 9 bits, this refers to FTDRH and FTDRL registers.

Note 2. When data length is 9 bits, write in order from FTDRH to FTDRL.

Figure 34.14 Example flow of serial transmission in asynchronous mode with FIFO selected

34.3.9 Serial Data Reception in Asynchronous Mode

(1) Non-FIFO selected

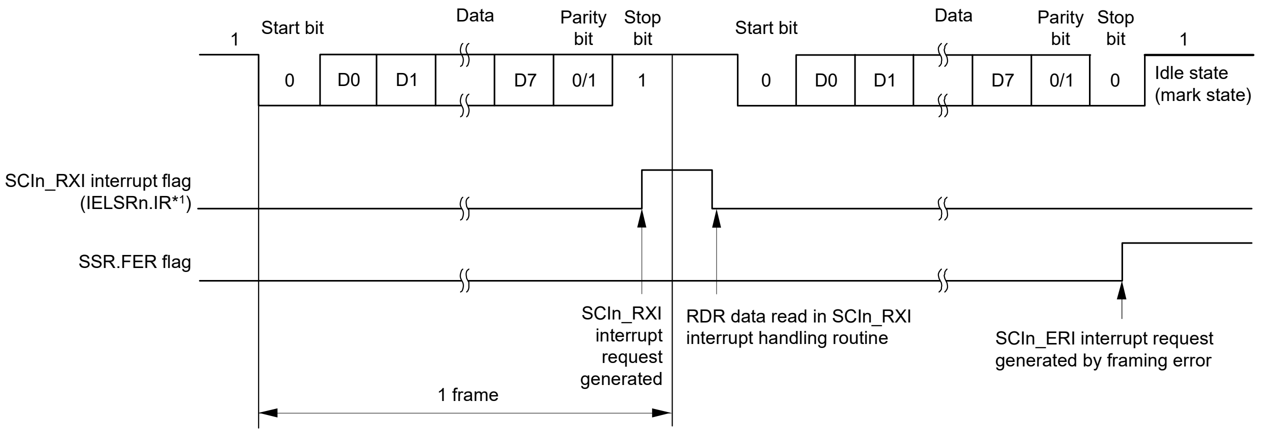
Figure 34.15 and Figure 34.16 show an example of the operation for serial data reception in asynchronous mode.

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and when it detects a start bit, the SCI performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR*1 register.

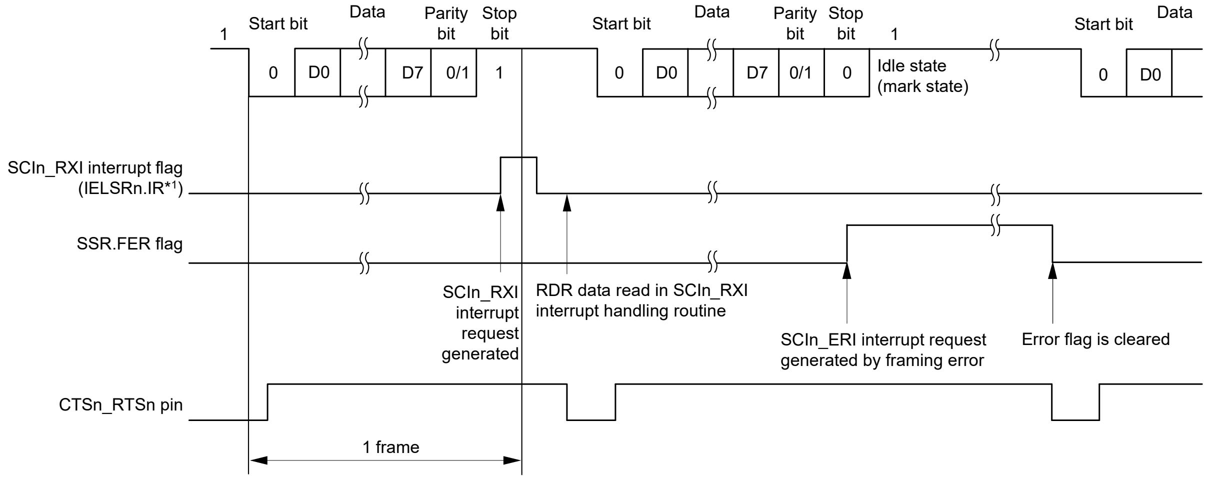
4. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR*1 register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.
5. If a framing error is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR*1 register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR*1 register. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that was transferred to the RDR register causes the CTSn_RTSn pin to output low.

Note 1. Only read data in the RDRHL register when 9-bit data length is selected.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.15 Example of SCI operation for serial reception in asynchronous mode (1) when the RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.16 Example of SCI operation for serial reception in asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

Table 34.24 lists the states of the flags in the SSR register and receive data handling when a receive error is detected.

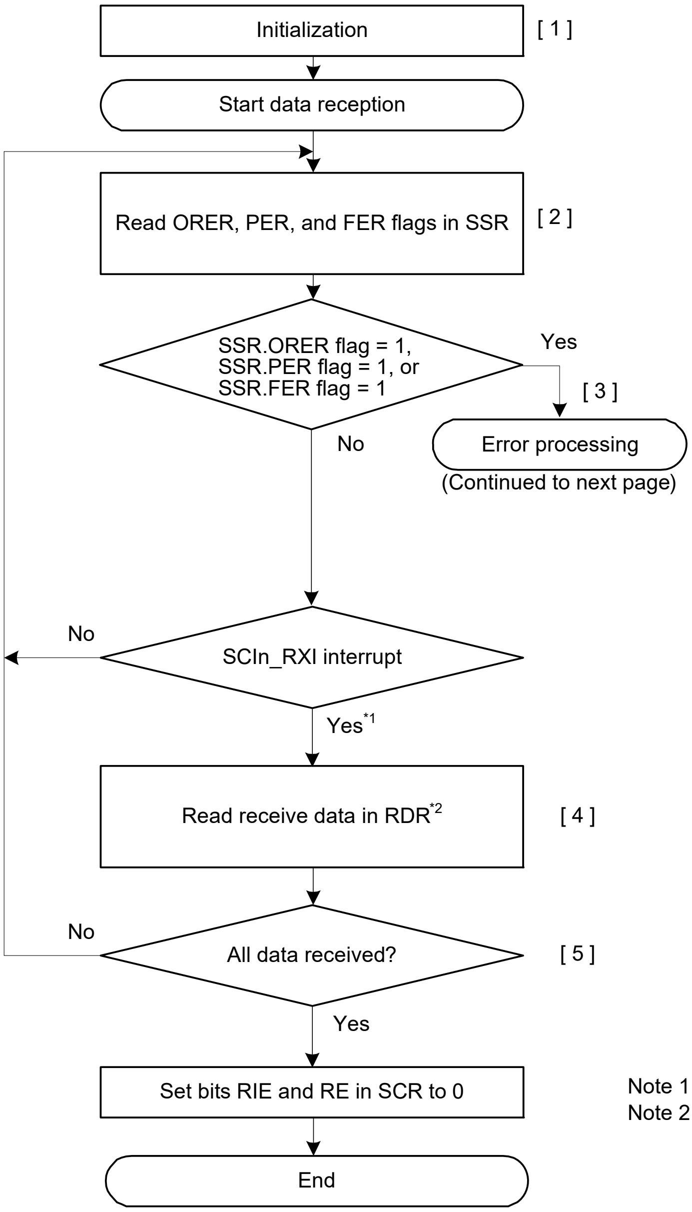
If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. In addition, be sure to read the RDR or RDRHL register during overrun error processing. When a reception is forced to terminate by setting the SCR.RE bit to 0 during operation, read the RDR or RDRHL register because received data that is not yet read might be left in the RDR or RDRHL.

Figure 34.17 and Figure 34.18 show example flows of serial data reception.

Table 34.24 Flags in SSR Status Register and receive data handling

Flags in the SSR Status Register				
ORER	**FER**	**PER**	**Receive data**	**Receive error type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Only read data in the RDRHL register when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, always set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. For a framing error, a break can be detected by reading the value of the input port associated with the RXDn pin.

[4] Read the receive data in RDR once in the SCIn_RXI interrupt handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR in the SCIn_RXI interrupt processing routine. The RDR data can also be read by activating the DMAC or DTC.

Note 1. Do not set RE to 0 before reading RDR.
Note 2. The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 34.17 Example flow of serial reception in asynchronous mode with non-FIFO selected (1)

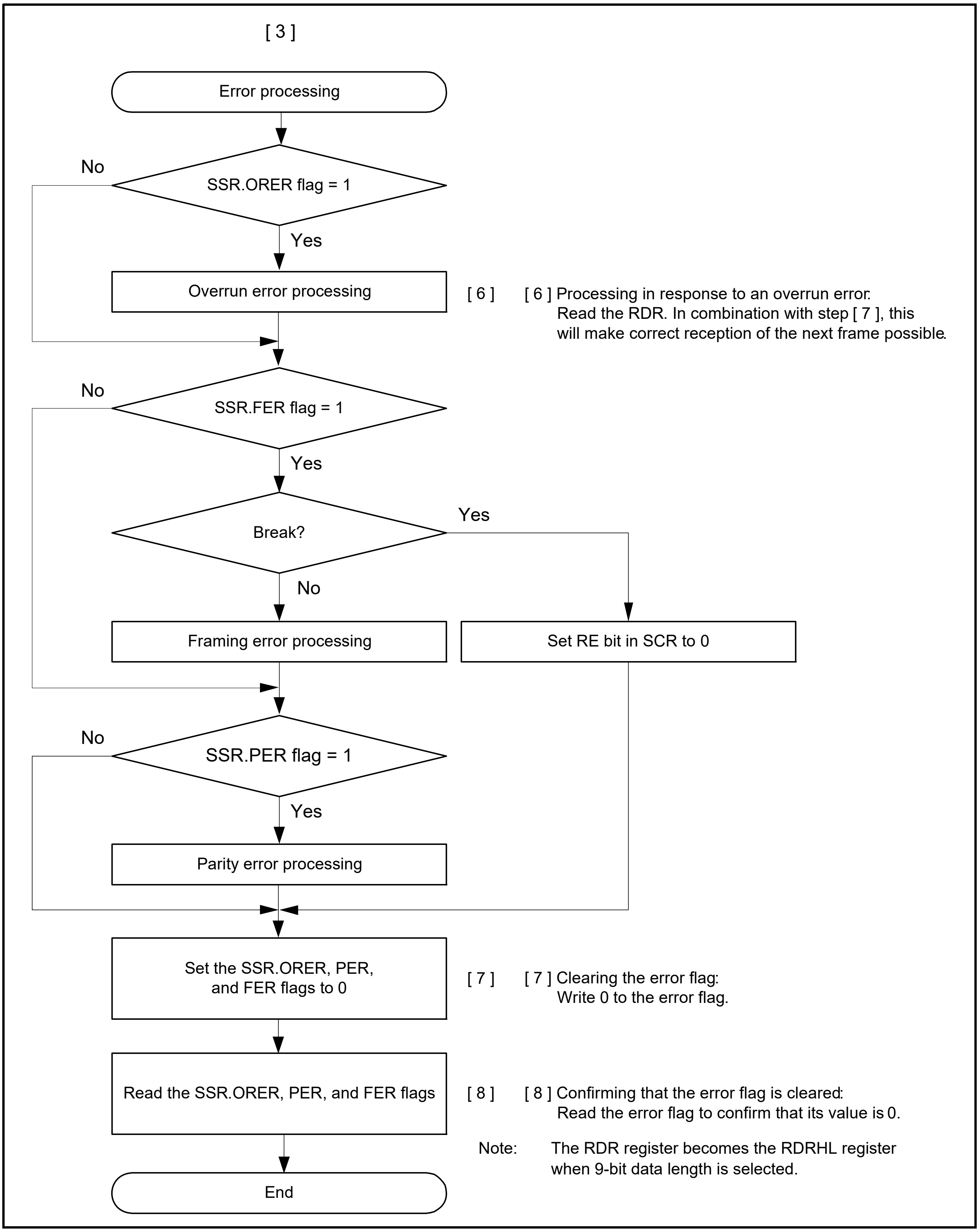


Figure 34.18 Example flow of serial reception in asynchronous mode with non-FIFO selected (2)

(2) FIFO selected

Figure 34.19 shows an example of a data format that is written to FRDRH and FRDRL in asynchronous mode.

In asynchronous mode, 0 is written to the MPB flag in the FRDRH register. Data that corresponds to the data length is written to FRDRH and FRDRL. Unused bits are written as 0. Read in order from FRDRH to FRDRL. If software reads

FRDRL, the SCI updates FER, PER, and receive data (RDAT[8:0]) in the FRDRL register with the next data. The RDF, ORER, and DR flags in the FRDRH register always reflect the associated flags in the SSR_FIFO register.

Data Length	Register Setting		Receive data in FRDRH, FRDRL															
			FRDRHL															
			FRDRH								FRDRL							
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	RDF	ORER	FER	PER	DR	0	0	0	7-bit receive data						
8 bits	1	1	—	RDF	ORER	FER	PER	DR	0	0	8-bit receive data							
9 bits	0	Don't care	—	RDF	ORER	FER	PER	DR	0	9-bit receive data								

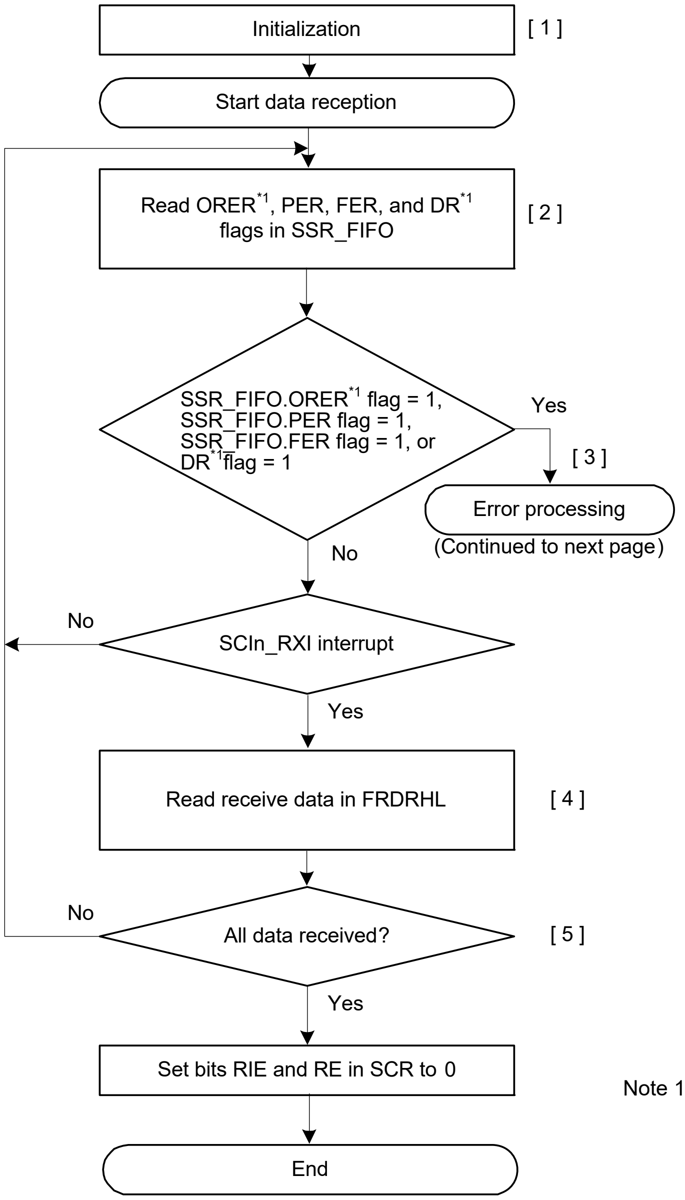
Note: 0 is always read for MPB flag (FRDRH[1]).
When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7].
When data length is 8 bits, 0 is always read for FRDRH[0].
FRDRH[7] bit is read as an indefinite value.

Figure 34.19 Data format stored in FRDRH and FRDRL with FIFO selected

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and, when it detects a start bit, the SCI performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR_FIFO.ORER flag is set to 1. If the SCR.RIE bit in SCR is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the FRDRL*1 register.
4. If a parity error is detected, the PER flag and receive data are transferred to the FRDRL*1 register. If the RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
5. If a framing error is detected, the FER flag and receive data are transferred to the FRDRL*1 register. If the RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
6. After a framing error is detected and when SCI detects that the continuous receive data is for one frame, reception stops.
7. When the amount of data stored in the FRDRL register falls below the specified receive triggering number, and the next data is not received after 15 ETUs from the last stop bit in asynchronous mode, the SSR_FIFO.DR bit is set to 1. When the RIE bit is 1 and the FCR.DRES bit is 0, the SCI generates an SCIn_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn_ERI interrupt request.
8. When reception finishes successfully, receive data is transferred to the FRDRL*1 register. The RDF bit is set to 1 when the amount of receive data written to FRDRHL is equal to or greater than the specified receive triggering number. If the SCR.RIE bit in SCR is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the FRDRL*2 register in the SCIn_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to FRDRL*3 is less than the RTS trigger number, the CTSn_RTSn pin outputs low.

Note 1. Only read data in the FRDRH and FRDRL registers when 9-bit data length is selected.
Note 2. Read data in order from FRDRH to FRDRL when 9-bit data length is selected.
Note 3. The SCI only checks for update to the FRDRL register and not to the FRDRH register when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

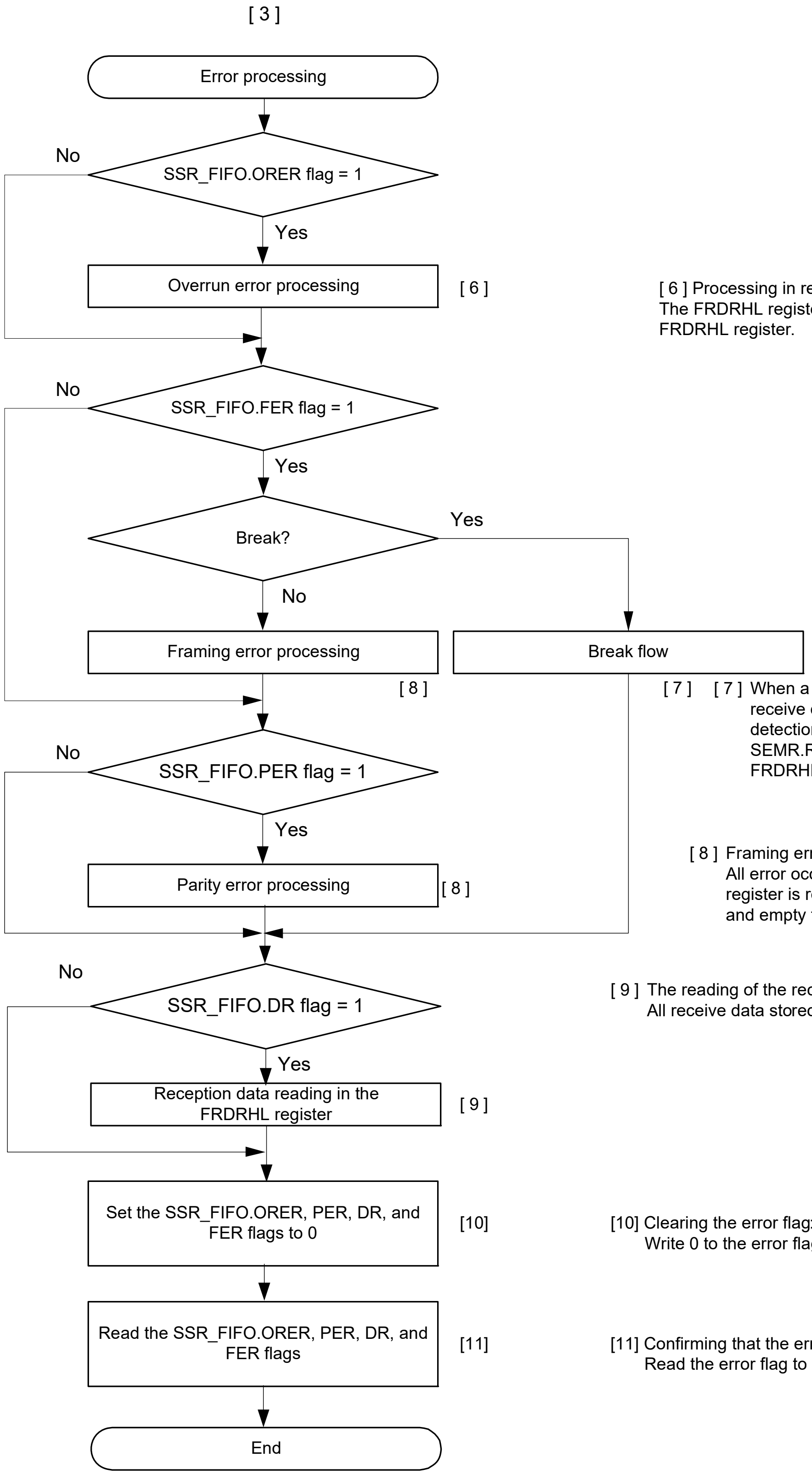
[2] [3] Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. A break can be detected by reading the SPTR.RXDMON bit. An error is identified by reading the ORER^{*1}, PER, DR^{*1}, and FER flags in SSR_FIFO. After performing the appropriate error processing, be sure to set the ORER^{*1} flag to 0. Reception cannot be resumed if ORER^{*1} flag is set to 1. The reception operation is continuous, even if the FER = 1 or PER = 1 or DR^{*1} = 1.

[4] Read the receive data in FRDRHL in the SCIn_RXI interrupt handling routine. The receive data stored in the FRDRHL register is read until the number of stored data is below the FCR.RTRG value. Confirm the number of the reception data in the FIFO by reading FDR.R.

[5] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, read data from FRDRHL in the SCIn_RXI interrupt handling routine and clear RDF and DR flags to 0.
The FRDRHL data can also be read by activating the DMA or DTC. The RDF flag is cleared automatically in this case, so do not write to the RDF flag.

Note 1. Can be read in the FRDRHL.ORER and DR flags.

Figure 34.20 Example flow of serial reception in asynchronous mode with FIFO selected (1)



[6] Processing in response to an overrun error.
The FRDRHL register is read and a space is made in the FRDRHL register.

[7] When a break is detected, transfer of the receive data to FRDRHL stops after the detection. When the break ends at SEMR.RXDESEL = 0, the last stored data of FRDRHL is break error frame (All 0 data).

[8] Framing error processing / parity error processing:
All error occurrence data stored in the FRDRHL register is read. (Or write 1 to the FCR.RFRST bit and empty the FRDRHL register.)

[9] The reading of the receive data (when FCR.DRES is 1):
All receive data stored in the FRDRHL register is read.

[10] Clearing the error flag:
Write 0 to the error flag.

[11] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 34.21 Example flow of serial reception in asynchronous mode with FIFO selected (2)

34.4 Multi-Processor Communication Function

The multi-processor communication function enables the SCI to transmit and receive data between multiple processors by sharing an asynchronous serial communication line that has an added multi-processor bit. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station.

The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle:

- When the multi-processor bit is set to 1, the transmission cycle is the ID transmission cycle
- When the multi-processor bit is set to 0, the transmission cycle is the data transmission cycle.

Figure 34.22 shows an example of communication between processors using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits communication data in which the multi-processor bit set to 0 is added to the transmit data. After receiving communication data with the multi-processor bit set to 1, the receiving station compares the received ID with the ID of the receiving station itself. If the two match, the receiving station receives communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until it receives data in which the multi-processor bit is set to 1.

(1) Non-FIFO selected

To support this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR register (the RDRHL register when 9-bit data length is selected)
- Detection of a receive error
- Setting of the respective RDRF, ORER, and FER status flags in the SSR register.

When the SCI receives a character in which the multi-processor bit is set to 1, the SSR.MPBT bit is set to 1 and the SCR.MPIE bit is automatically cleared, returning the SCI to non-multi processor reception operation. If the SCR.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi processor asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in non-multi processor asynchronous mode.

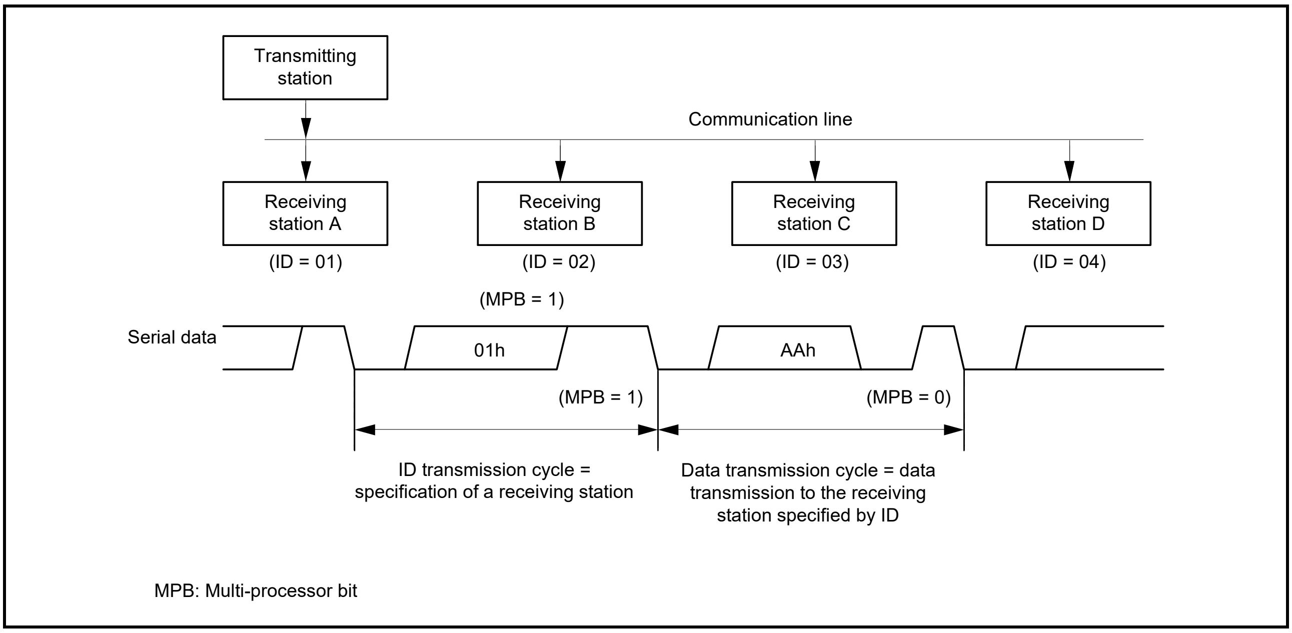


Figure 34.22 Example of communication using multi-processor format with transmission of data AAh to receiving station A

(2) FIFO selected

For data transmission, software must write data to FTDRHL.MPBT that corresponds to transmit data in FTDRHL.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to FRDRHL.MPB and receive data is written to FRDRL.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the FRDRHL register
- Detection of a receive error
- Break
- Setting of the respective RDF, ORER, and FER status flags in the SSR_FIFO register.

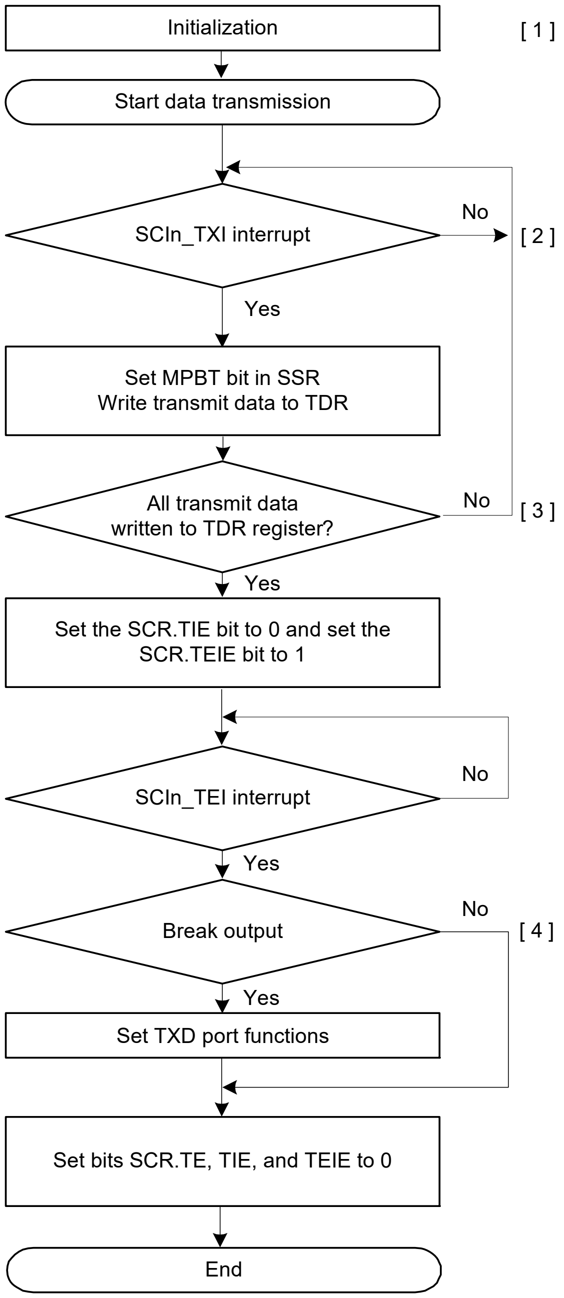
When the SCI receives an 8-bit character in which the multi-processor bit is set to 1, the FRDRHL.MPB bit is set to 1 and receive data is written to FRDRHL.RDAT. The SCR.MPIE bit is automatically cleared, returning the SCI to non-multi processor reception operation. If the SCR.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi processor asynchronous mode with non-FIFO selected.

34.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 34.23 shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode.



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame, and transmission is enabled.

[2] SCIn_TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (SCIn_TXI) request is generated.
Set the MPBT bit in SSR to 0 or 1, and write transmit data to TDR once in the SCIn_TXI interrupt processing routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using an SCIn_TXI interrupt request.
Transmit data can also be written to TDR by activating the DMAC or DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXDn pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

Figure 34.23 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

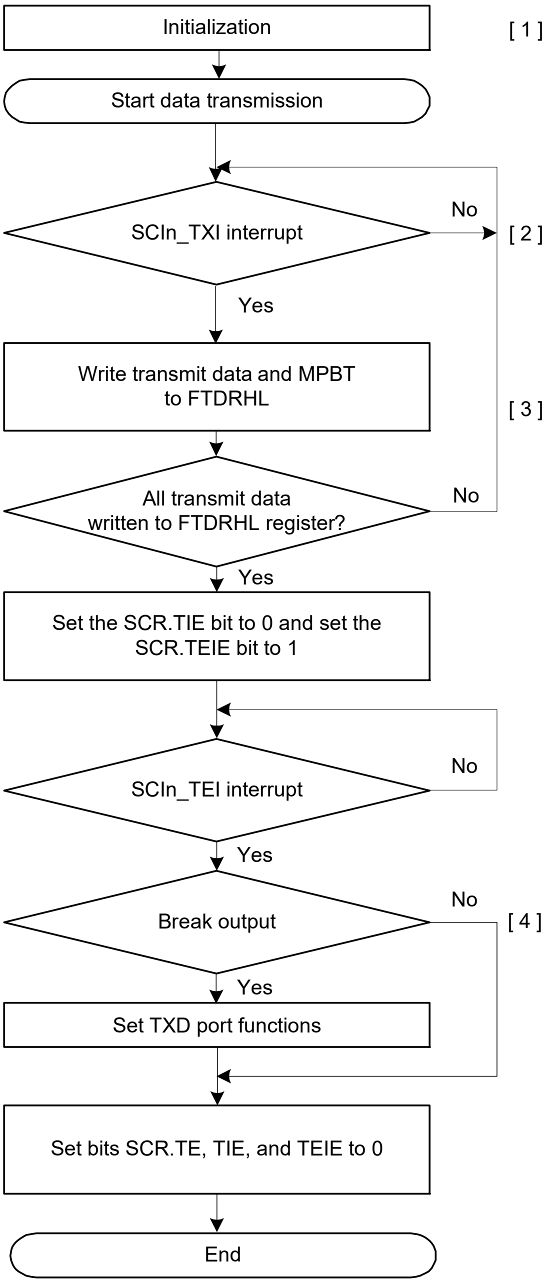
Figure 34.24 shows an example of data format that is written to FTDRH and FTDRL in multi-processor mode. The FTDRH.MPBT bit is set to 1. Data is set to FTDRH and FTDRL with the correct data length. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
			FTDRH								FTDRL							
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	—	—	—	—	—	MPBT	—	—	7-bit transmit data						
8 bits	1	1	—	—	—	—	—	—	MPBT	—	8-bit transmit data							
9 bits	0	Don’t care	—	—	—	—	—	—	MPBT	9-bit transmit data								

—: Invalid. The write value should be 0.

Figure 34.24 Data format written to FTDRH and FTDRL in multi-processor mode with FIFO selected

Figure 34.25 shows an example flow of multi-processor data transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the FTDRH.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode with non-FIFO selected.



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to FTDRHL by an SCIn_TXI interrupt request:
When transmit data is transferred from FTDRHL to TSR, when the quantity of receive data written in FTDRHL is equal to or less than the specified transmit triggering number, a transmit data FIFO empty interrupt (SCIn_TXI) request is generated.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data and MPBT to FTDRHL once using an SCIn_TXI interrupt request. Transmit data can also be written to FTDRHL by activating the DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRHL.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXDn pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

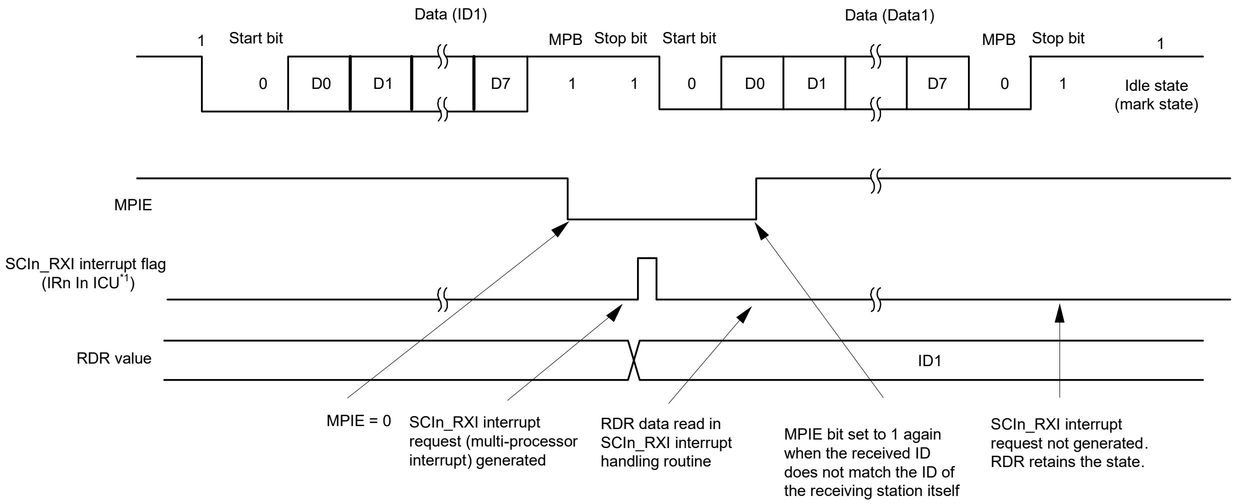
Figure 34.25 Example flow of serial transmission in multi-processor mode with FIFO selected

34.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

Figure 34.27 and Figure 34.28 are example flows of multi-processor data reception. When the SCR.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register (the RDRHL register when 9-bit data length is selected), and the SCIn_RXI interrupt request is generated. The rest of the operations are the same as operations in asynchronous mode.

Figure 34.26 shows an example operation for data reception.



MPIE = 0

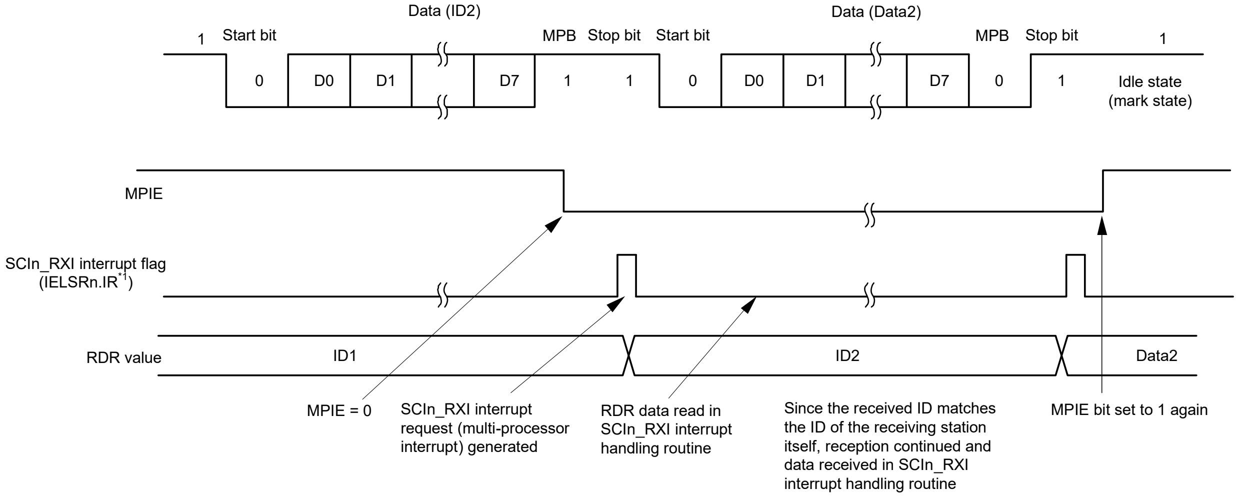
SCIn_RXI interrupt request (multi-processor interrupt) generated

RDR data read in SCIn_RXI interrupt handling routine

MPIE bit set to 1 again when the received ID does not match the ID of the receiving station itself

SCIn_RXI interrupt request not generated. RDR retains the state.

(a) When the received ID does not match the ID of the receiving station itself



MPIE = 0

SCIn_RXI interrupt request (multi-processor interrupt) generated

RDR data read in SCIn_RXI interrupt handling routine

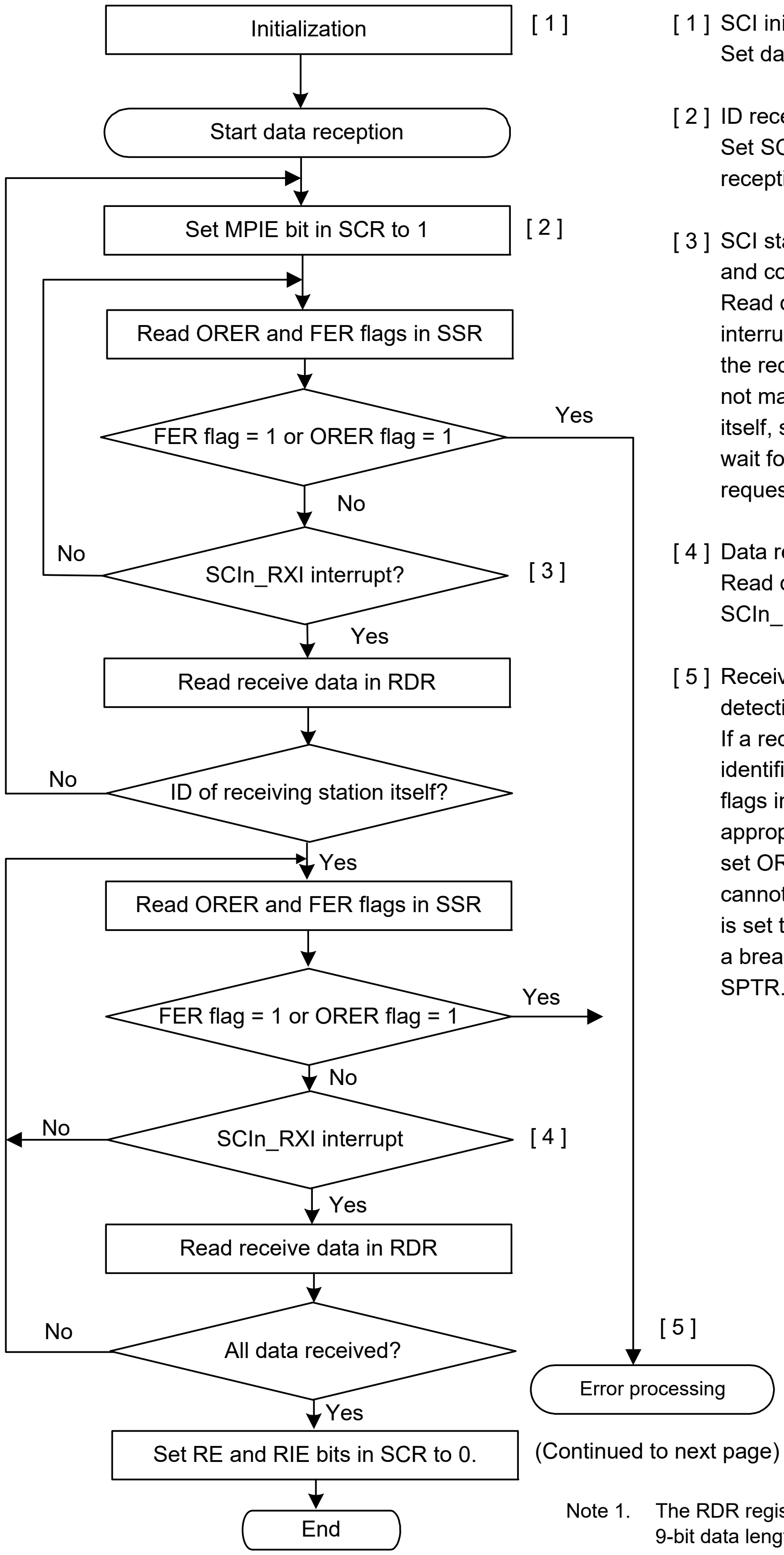
Since the received ID matches the ID of the receiving station itself, reception continued and data received in SCIn_RXI interrupt handling routine

MPIE bit set to 1 again

(b) When the received ID matches the ID of the receiving station itself

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.26 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit



[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set SCR.MPIE bit to 1 and wait for ID reception.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR[*1] at the first SCIn_RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in RDR[*1] once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in SSR. After performing the appropriate error processing, be sure to set ORER and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading SPTR.RXDMON bit.

(Continued to next page)

Note 1. The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 34.27 Example flow of multi-processor serial reception with non-FIFO selected (1)

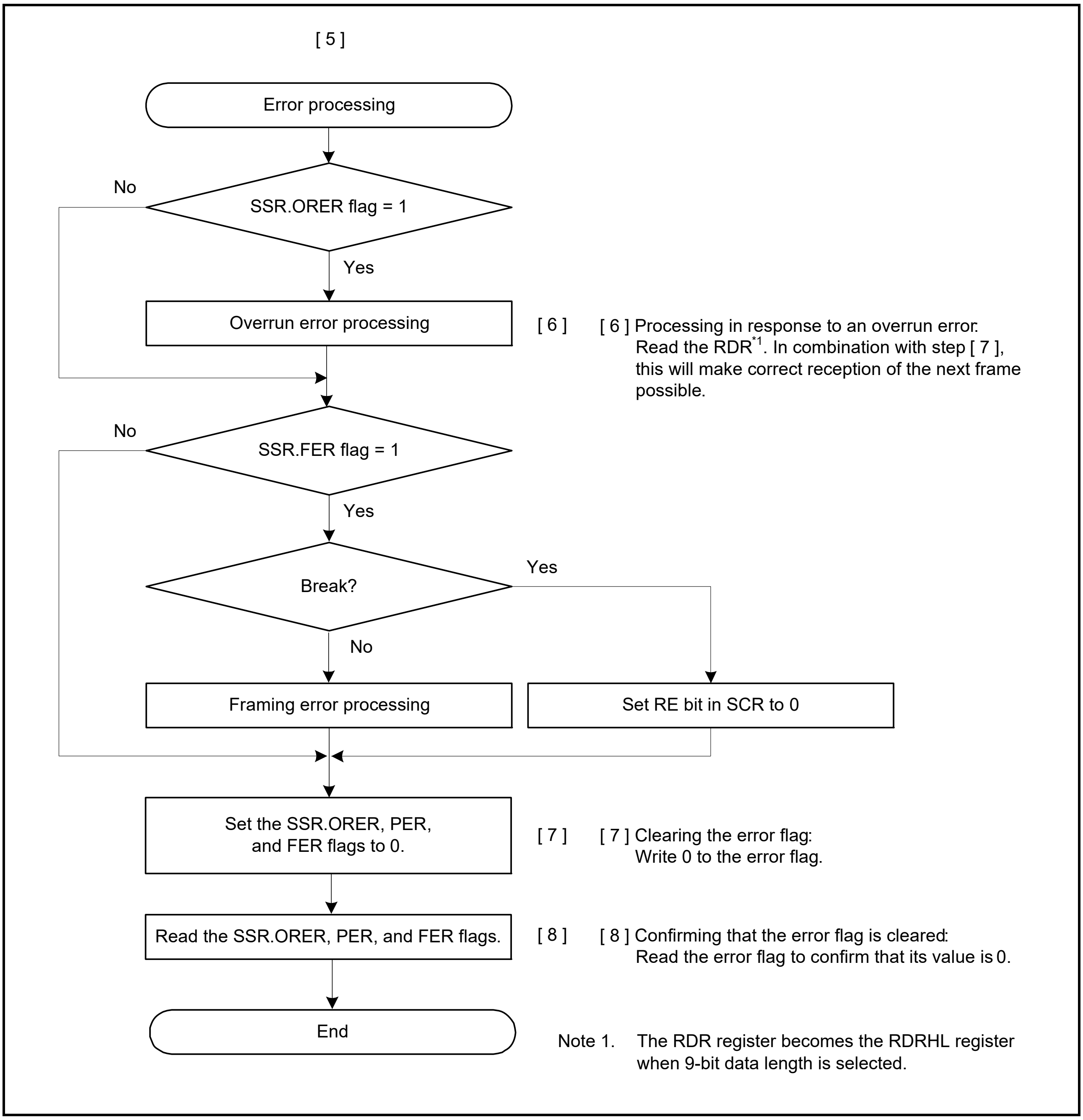


Figure 34.28 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 34.29 shows an example of a data format that is written to FRDRH and FRDRL in multi-processor mode.

In multi-processor mode, the MPB value that is a part of the receive data is written to the FRDRH.MPB flag. A value of 0 is written to the FRDRH.PER flag. Data is written to FRDRH and FRDRL with the correct data length. Unused bits are written with 0. Read in order from FRDRH to FRDRL. When software reads the FRDRL register, the SCI updates FER, MPB, and receive data (RDAT[8:0]) in FRDRL with the next data. The RDF, ORER and DR flags in the FRDRH register always reflect the associated flags in the SSR_FIFO register.

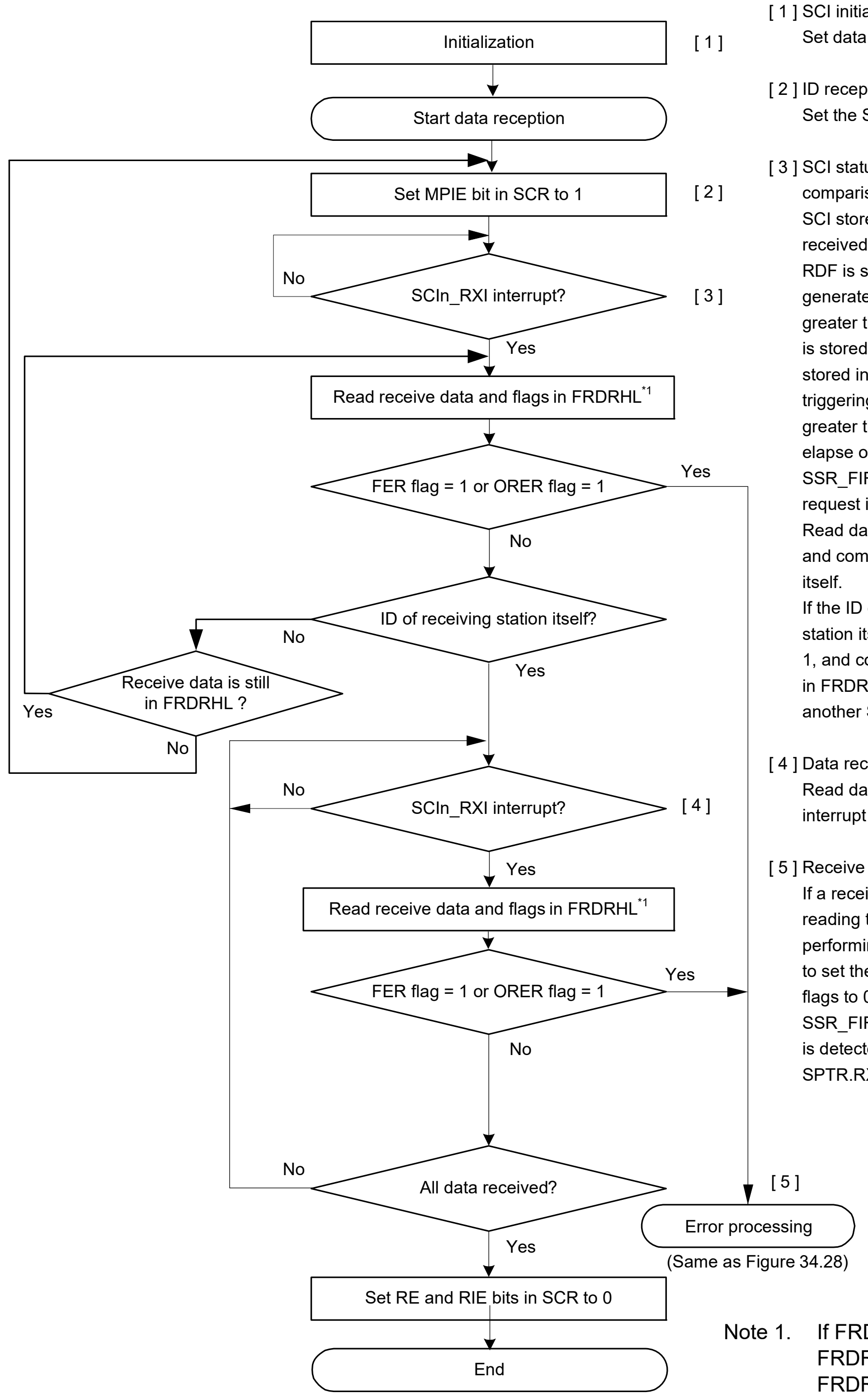
Data Length	Register Setting		Receive data in FRDRH, FRDRL															
			FRDRHL															
			FRDRH								FRDRL							
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	RDF	ORER	FER	0	DR	MPB	0	0	7-bit receive data						
8 bits	1	1	—	RDF	ORER	FER	0	DR	MPB	0	8-bit receive data							
9 bits	0	Don't care	—	RDF	ORER	FER	0	DR	MPB	9-bit receive data								

Note: When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7]
When data length is 8 bits, 0 is always read for FRDRH[0]
FRDRH[7] bit is read as an indefinite value.

Figure 34.29 Data format stored in FRDRH and FRDRL in multi-processor mode with FIFO selected

Figure 34.30 shows an example flow of multi-processor data reception with FIFO selected. When the SCR.MPIE is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to the FRDRHL register. The SCR.MPIE bit is automatically cleared and non-multi processor reception continues.

If a framing error occurs and the SSR_FIFO.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as operations in asynchronous mode with non-FIFO selected.



[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set the SCR.MPIE bit to 1 and wait for ID reception.

[3] SCI status confirmation and reception and comparison of ID:
SCI stores first data (MPB = 1) and subsequent received data in FRDRHL.
RDF is set to 1 and an SCIn_RXI interrupt request is generated when a quantity of receive data equal to or greater than the specified receive triggering number is stored in FRDRHL. When the quantity of data stored in FRDRHL falls the specified receive triggering number and received data is equal to or greater than 1, and no next data is received after the elapse of 15 ETUs from the last stop bit, SSR_FIFO.DR is set to 1. An SCIn_RXI interrupt request is generated when FCR.DRES bit is 0.
Read data in FRDRHL at the first SCIn_RXI interrupt, and compare it with the ID of the receiving station itself.
If the ID does not match the ID of the receiving station itself, the SCI reads until the data with MPB = 1, and compares next ID. If it is no data with MPB = 1 in FRDRHL, set the MPIE bit to 1 again and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in FRDRHL once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading the ORER and FER flags in SSR_FIFO. After performing the appropriate error processing, be sure to set the SSR_FIFO.ORER and SSR_FIFO.FER flags to 0. Reception cannot be resumed if the SSR_FIFO.ORER flag is set to 1. When framing error is detected, a break can be detected by reading the SPTR.RXDMON bit.

Note 1. If FRDRH and FRDRL are used instead of FRDRHL, read in order from FRDRH to FRDRL.

Figure 34.30 Example flow of serial reception in multi-processor mode with FIFO selected

34.5 Operation in Clock Synchronous Mode

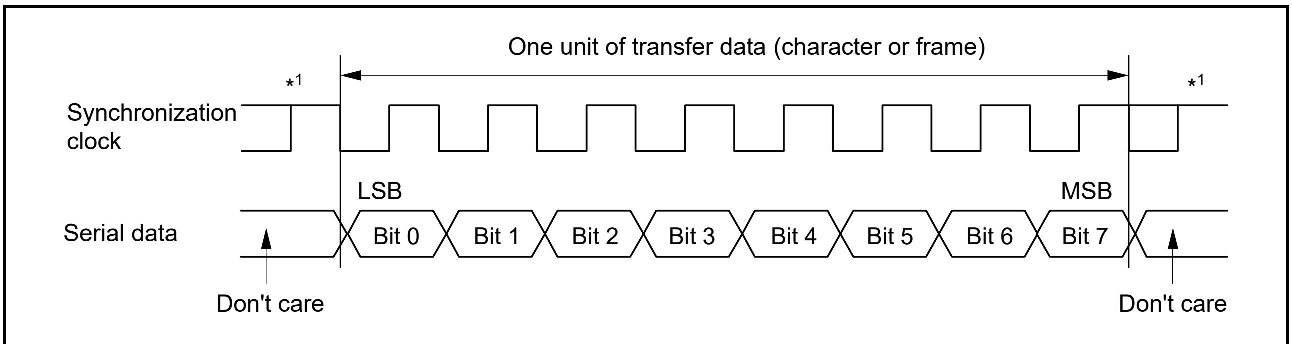
Figure 34.31 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.

In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the last bit as output state. When the SPMR.CKPH bit is 1 in slave mode, the transmission line holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a shared clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

However, it is not possible to perform continuous transfer in the fastest bit rate setting (BRR[7:0] = 00h and SMR.CKS[1:0] = 00b). Therefore, when the FIFO is selected, this setting (BRR[7:0] = 00h and SMR.CKS[1:0] = 00b) is not available.



Note 1. Holds a high level except during continuous transfer.

Figure 34.31 Data format in clock synchronous serial communications with LSB-first order

34.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected based on the SCR.CKE[1:0] setting.

When the SCI operates on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character. When no transfer is performed, the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output starts when the SCR.RE bit set to 1. The synchronization clock stops when it goes high[*1] and an overrun error occurs or the SCR.RE bit is set to 0.

When only data reception is performed and the CTS function is enabled, the clock output does not start when the SCR.RE bit set to 1 and the CTSn_RTSn pin input is high. The synchronization clock output starts when the SCR.RE bit is set to 1 and the CTSn_RTSn pin input is low. Following that, when the CTSn_RTSn pin input is high on completion of the frame reception, the synchronization clock output stops when it goes high. If the CTSn_RTSn pin input continues to be low, the synchronization clock stops when it goes high[*1] and an overrun error occurs or the SCR.RE bit is set to 0.

Note 1. The signal is held high while (SPMR.CKPH = 0 && SPMR.CKPOL = 0) or (SPMR.CKPH = 1 && SPMR.CKPOL = 1). It is held low while (SPMR.CKPH = 0 && SPMR.CKPOL = 1) or (SPMR.CKPH = 1 && SPMR.CKPOL = 0).

34.5.2 CTS and RTS Functions

In the CTS function, the CTSn_RTSn pin input controls the start of data reception or transmission when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, setting the CTSn_RTSn pin low causes data reception or transmission to start.

Setting the CTSn_RTSn pin high while the data transmission or reception is in progress does not affect transmission or reception of the current frame.

In the RTS function, the CTSn_RTSn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The CTSn_RTSn output goes low when serial communication becomes possible. Conditions for output of the CTSn_RTSn low and high are shown as follows:

[Conditions for low output]

Satisfaction of all the following conditions:

(a) Non-FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit or the SCR.TE bit is 1
- When serial communication is enabled
- There is no received data available to be read when the SCR.RE bit is 1
- Data is available for transmission in the TSR register when SCR.TE bit is 1
- The SSR.ORER flag is 0.

(b) FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit or the SCR.TE bit is 1
- When serial communication is enabled
- The amount of receive data written in FRDRHL is less than the specified CTSn_RTSn output triggering number when SCR.RE = 1
- Data that has not been transmitted is available in FTDRHL when SCR.TE bit is 1 and SCR.CKE[1] bit is 0
- Data is available for transmission in the TSR register when SCR.TE bit is 1 and SCR.CKE[1] bit is 1
- The SSR_FIFO.ORER flag is 0.

[Condition for high output]

(a) Non-FIFO selected

- The conditions for low output are not satisfied
- When reception is terminated with SCR.RE = 0 without reading the RDR register after reception is complete, RTS remains high. At this time, read the SCR register for dummy values after writing 0 to SCR.RE.

(b) FIFO selected

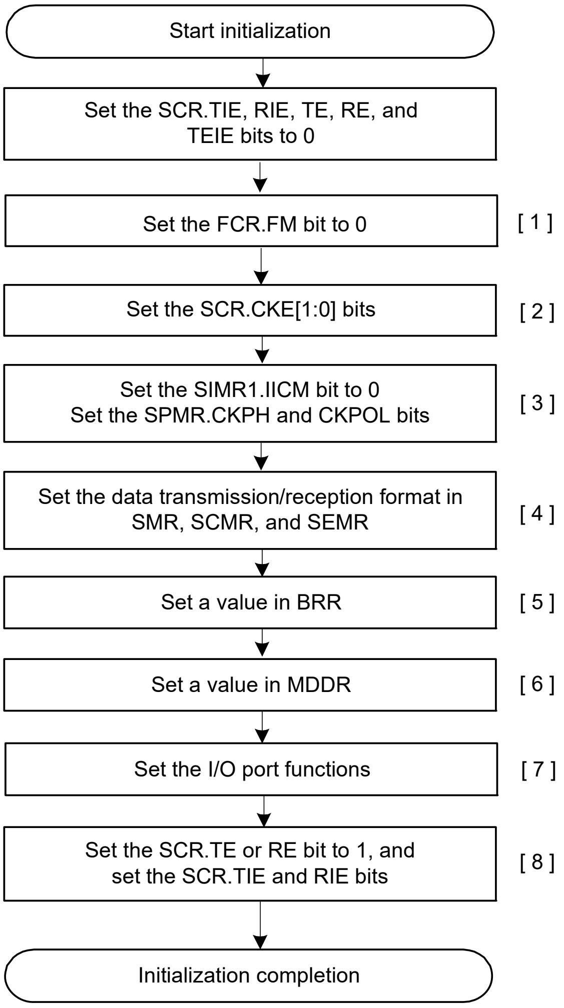
- The conditions for low output are not satisfied.

34.5.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register, then continue through the SCI initialization procedure given in the sections describing non-FIFO and FIFO selection in 34.5.2 CTS and RTS Functions. Anytime the operating mode or transfer format is to be changed, the SCR register must be initialized before the change can be made.

Note: Setting the SCR.RE bit to 0 initializes neither the ORER, FER, RDRF, RDF, PER, and DR flags in SSR/SSR_FIFO nor the RDR and RDRHL register. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: Switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 when the SCR.TIE bit is 1 generates an SCIn_TXI interrupt request.



[1] Set the FCR.FM bit to 0.

[2] Set the clock selection in SCR.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write a value corresponding to the bit rate to BRR.
This step is not necessary if an external clock is used.

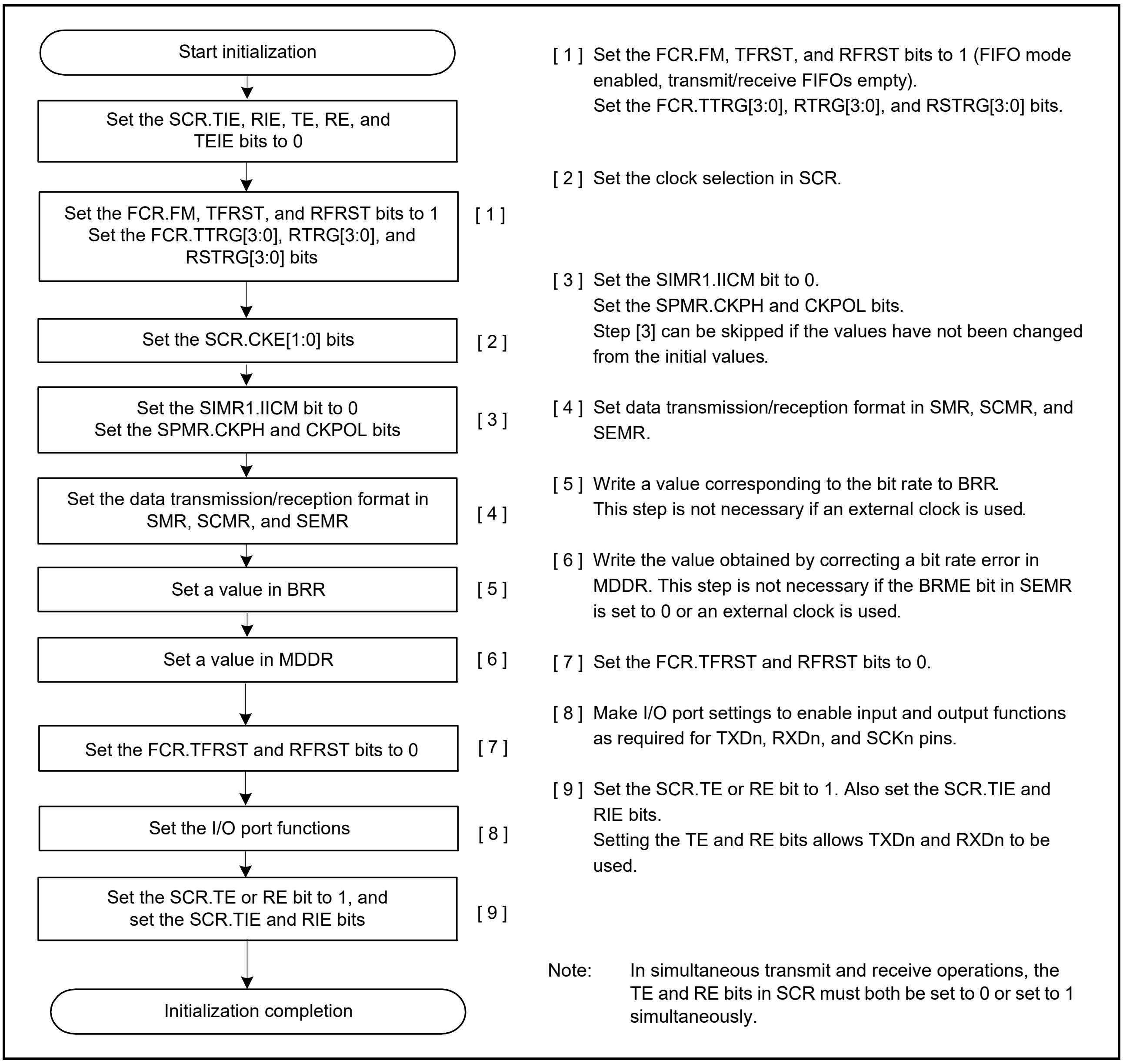
[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR must both be set to 0 or set to 1 simultaneously.

Figure 34.32 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected



[1] Set the FCR.FM, TFRST, and RFRST bits to 1 (FIFO mode enabled, transmit/receive FIFOs empty).
Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.

[2] Set the clock selection in SCR.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set data transmission/reception format in SMR, SCMR, and SEMR.

[5] Write a value corresponding to the bit rate to BRR.
This step is not necessary if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.

[7] Set the FCR.TFRST and RFRST bits to 0.

[8] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[9] Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR must both be set to 0 or set to 1 simultaneously.

Figure 34.33 Example flow of SCI initialization in clock synchronous mode with FIFO selected

34.5.4 Serial Data Transmission in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 34.34, Figure 34.35, and Figure 34.36 show examples of serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as follows:

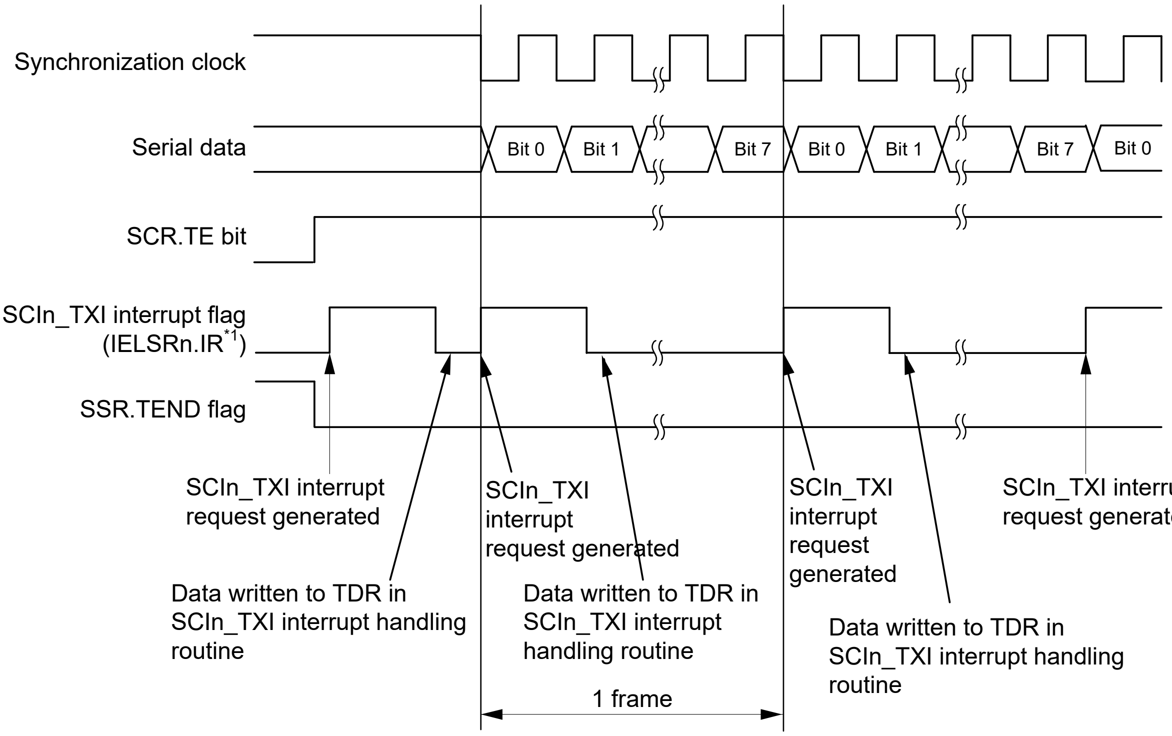
1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. The SCIn_TXI interrupt request at the beginning of transmission is generated when the TE bit is set to 1 but only after the TIE bit in the SCR is also set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.

3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the SPMR.CTSE bit is 1.
4. The SCI checks for update to the TDR register on output of the last bit.
5. When the TDR register is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame starts.
6. If TDR is not updated, the SSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 34.34, Figure 34.35, and Figure 34.36 show examples of serial data transmission.

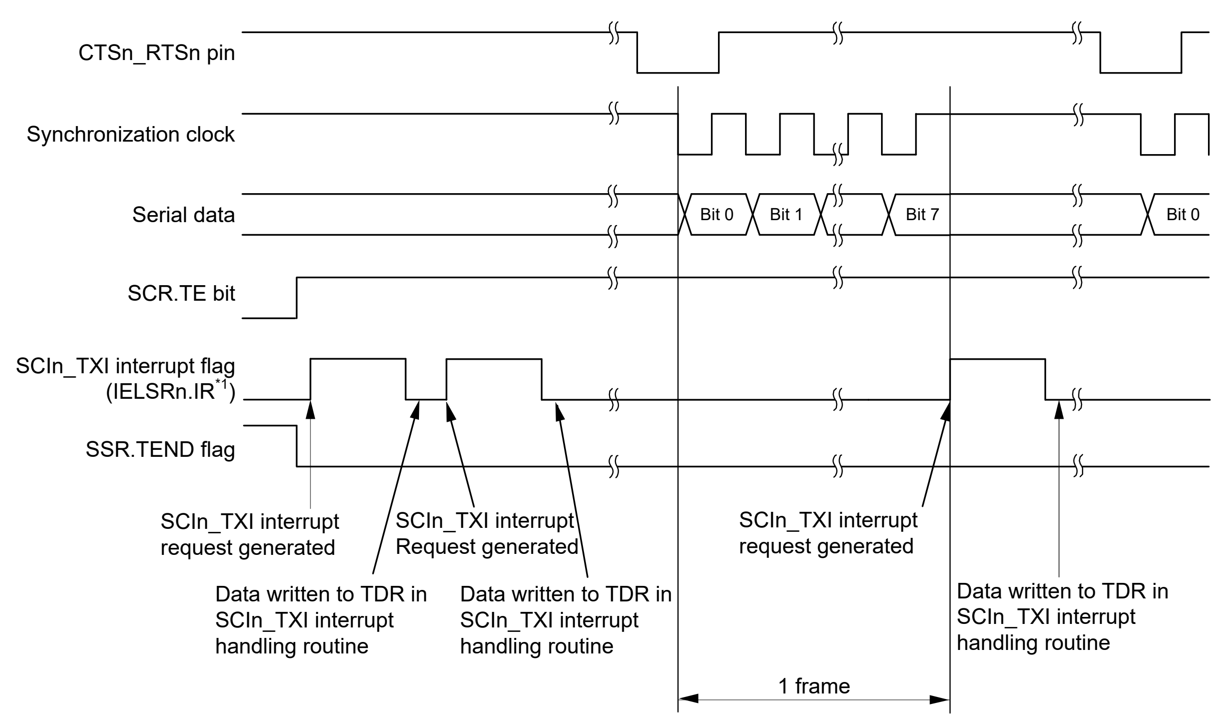
Transmission does not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Always set the receive error flags to 0 before starting transmission.

Note: Setting the SCR.RE bit to 0 does not clear the receive error flags.



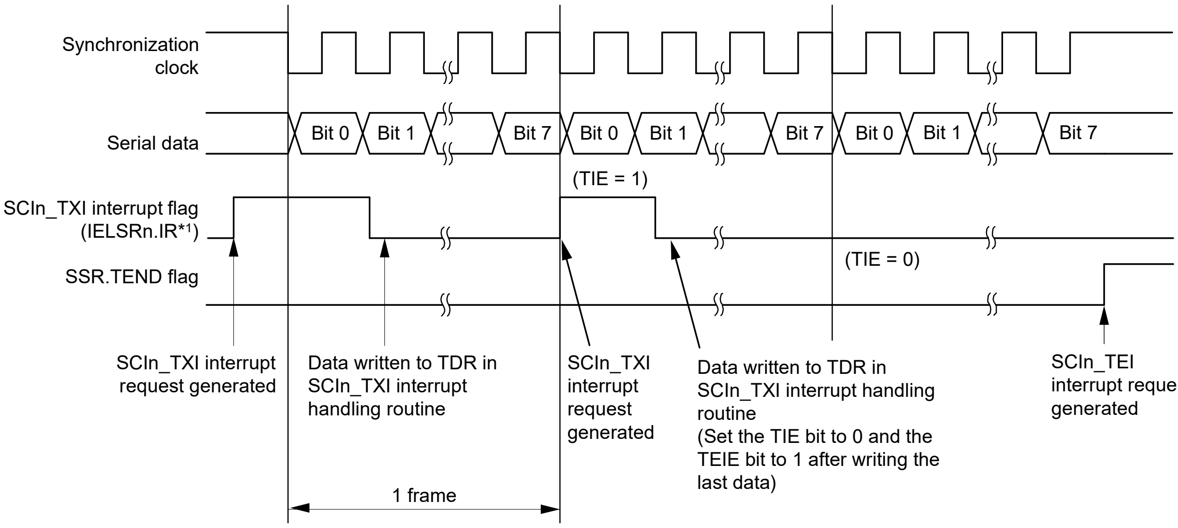
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.34 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



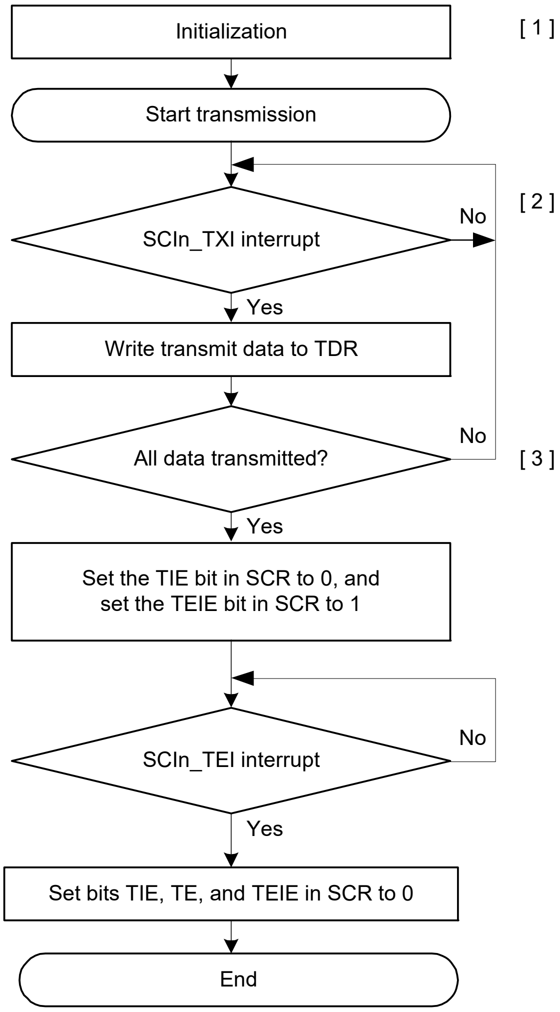
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.35 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.36 Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



[1] SCI initialization:
Set data transmission.

[2] Write transmit data to TDR by an SCIn_TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (SCIn_TXI) request is generated.
Transmit data is written to TDR once from the handling routine for SCIn_TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR on accepting a transmit data empty interrupt (SCIn_TXI) request.
Transmit data can also be written to TDR by activating the DMAC or DTC by the SCIn_TXI interrupt request.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to TDR.

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this might lead to insufficient received-data hold time on the receiver side.

Figure 34.37 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

(2) FIFO selected

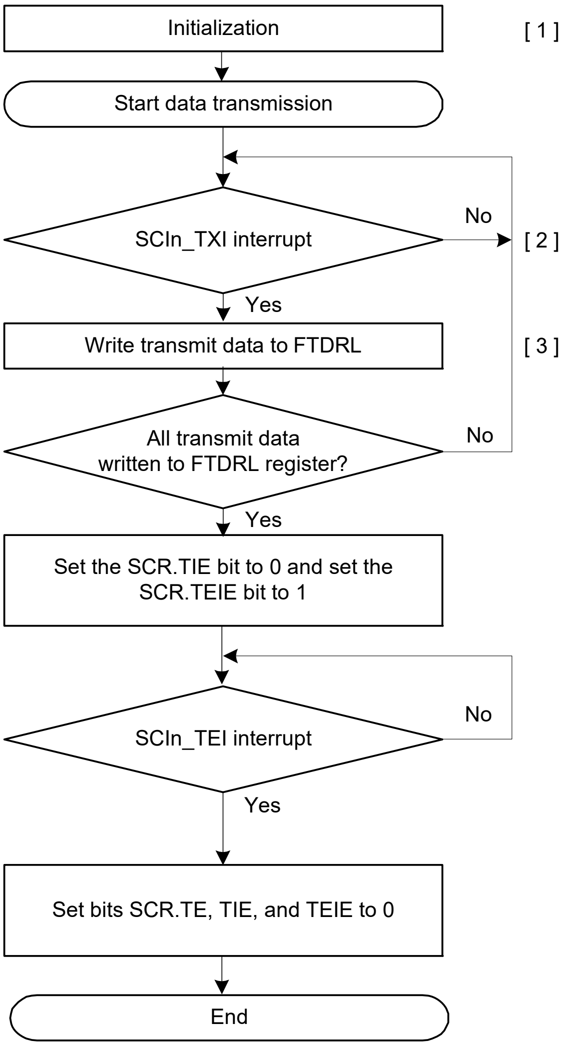
Figure 34.38 shows an example of serial transmission in clock synchronous mode with FIFO selected.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the FTDRL*1 register to the TSR register when data is written to FTDRL*1 in the SCIn_TXI interrupt handling routine. The amount of data that can be written to FTDRL is 16 minus FDR.T[4:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 but only after the SCR.TIE bit is also set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from FTDRL to TSR, the SCI starts transmission. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, the SSR_FIFO.TDFE is set to 1. When the SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine before transmission of the current transmit data has finished. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL from the handling routine for SCIn_TXI requests.

3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the SPMR.CTSE bit is 1.
4. The SCI checks whether non-transmitted data remains in FTDRL on output of the stop bit.
5. When FTDRL is updated, the next transmit data is transferred from FTDRL to TSR and serial transmission of the next frame starts.
6. If FTDRL is not updated, the SSR_FIFO.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Note 1. In clock synchronous mode, FTDRH is not used.



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to FTDRL by an SCIn_TXI interrupt request:
When transmit data is transferred from FTDRL to TSR, when the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, a transmit data FIFO empty interrupt (SCIn_TXI) request is generated. Write transmit data to FTDRL once in the SCIn_TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine and clear the SSR_FIFO.TDFE flag to 0 before transmission of the current transmit data has finished.
Transmit data can also be written to FTDRL by activating the DMAC or DTC. The TDFE flag is cleared automatically in this case. Do not write to the TDFE flag.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL.

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR_FIFO.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this might lead to insufficient received-data hold time on the receiver side.

Figure 34.38 Example flow of serial transmission in clock synchronous mode with FIFO selected

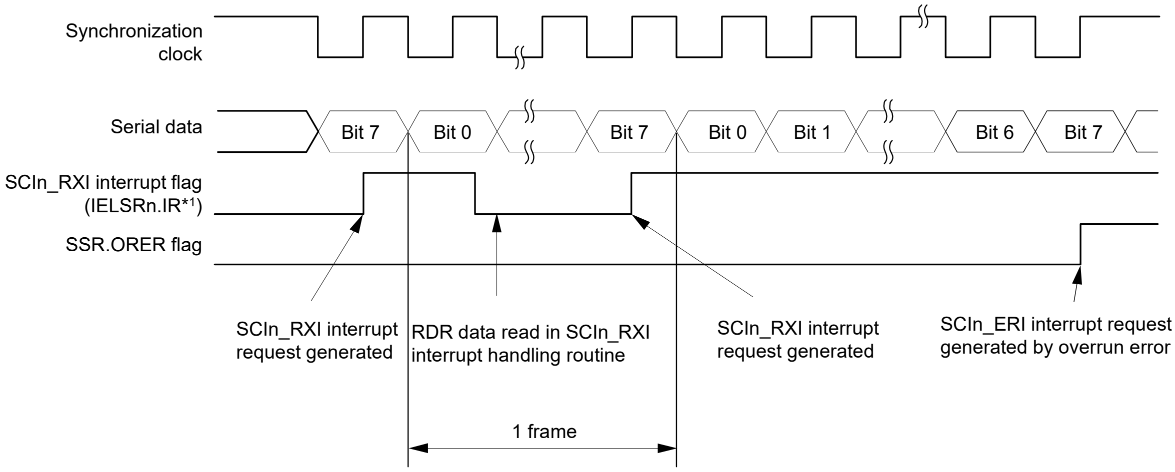
34.5.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 34.39 and Figure 34.40 show examples of SCI operation for serial reception in clock synchronous mode.

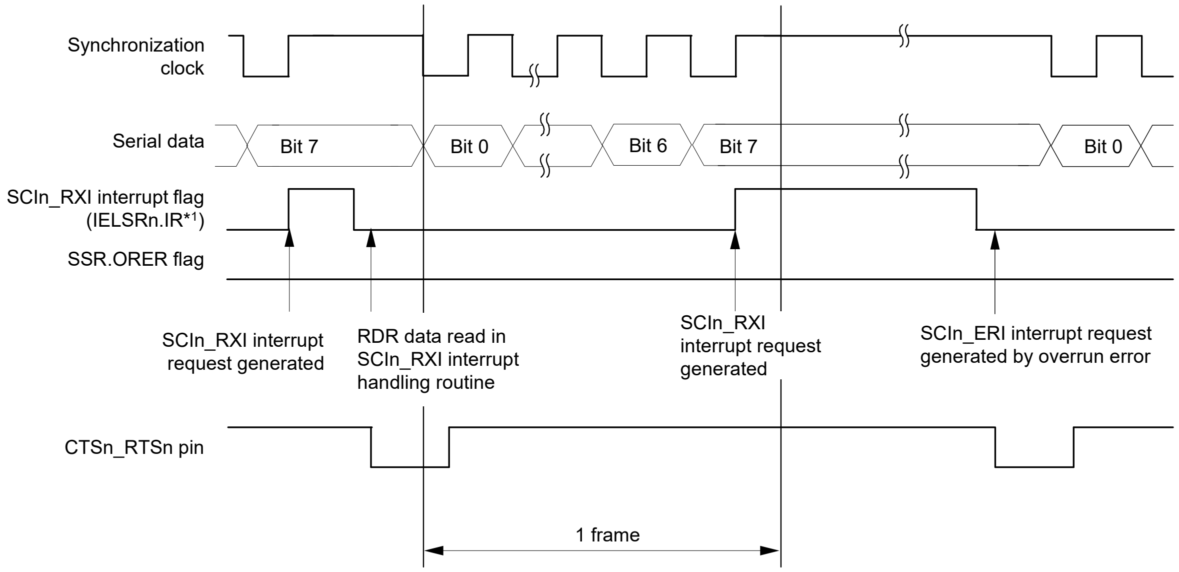
In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the CTSn_RTSn pin goes low.
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR.ORER bit is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception completes successfully, receive data is transferred to the RDR register. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that is transferred to RDR causes the CTSn_RTSn pin to output low.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.39 Example operation for serial reception in clock synchronous mode (1) when the RTS function is not used

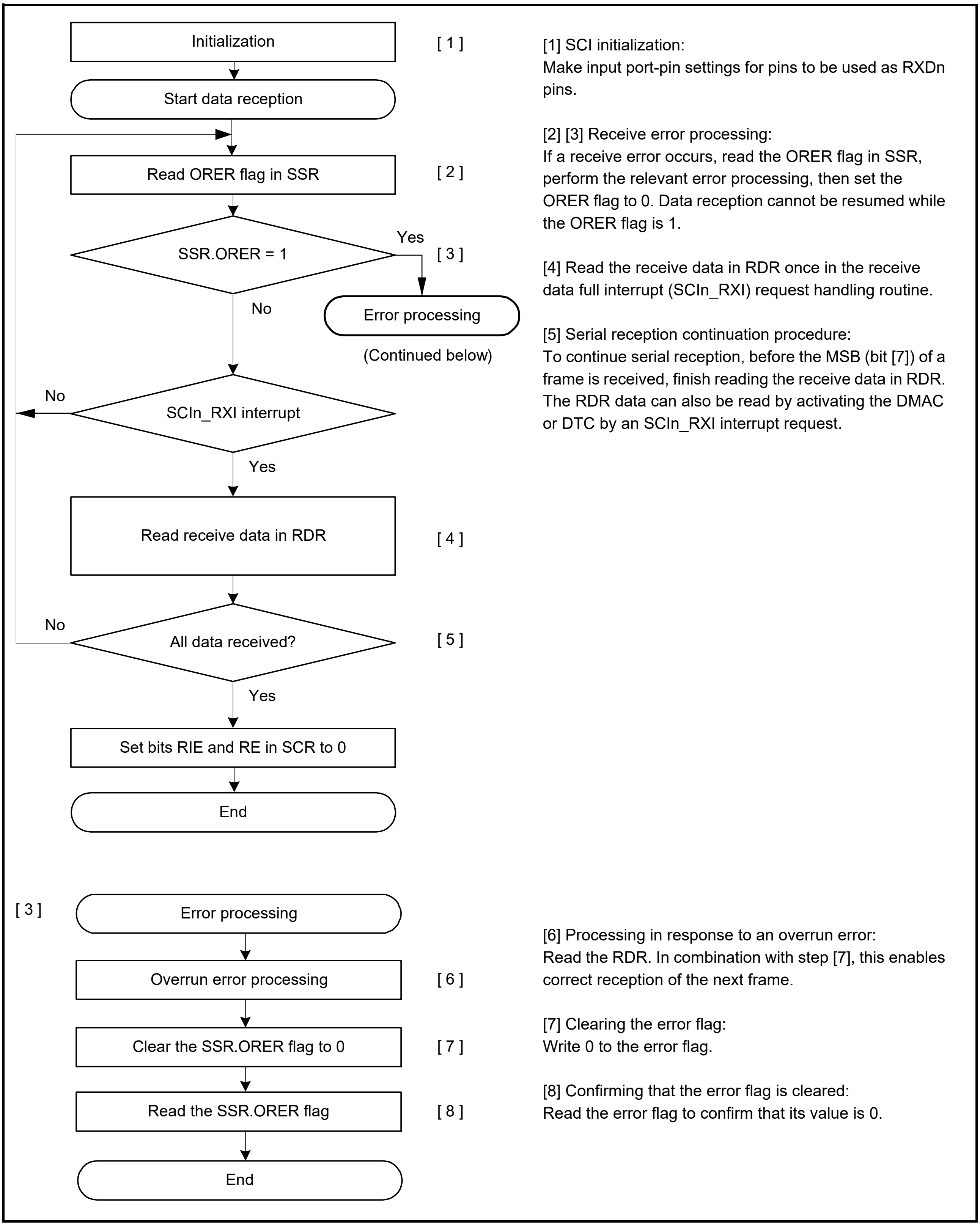


Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.40 Example operation for serial reception in clock synchronous mode (2) when RTS function is used

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER bits in the SSR register to 0 before resuming data reception. Additionally, always read the RDR register during overrun error processing. When a data reception is forced to terminate by a 0 write to the SCR.RE bit during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

Figure 34.41 shows an example flow of serial data reception.



[1] SCI initialization:
Make input port-pin settings for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in RDR once in the receive data full interrupt (SCIn_RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit [7]) of a frame is received, finish reading the receive data in RDR. The RDR data can also be read by activating the DMAC or DTC by an SCIn_RXI interrupt request.

[6] Processing in response to an overrun error:
Read the RDR. In combination with step [7], this enables correct reception of the next frame.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 34.41 Example flow of serial reception in clock synchronous mode with non-FIFO selected

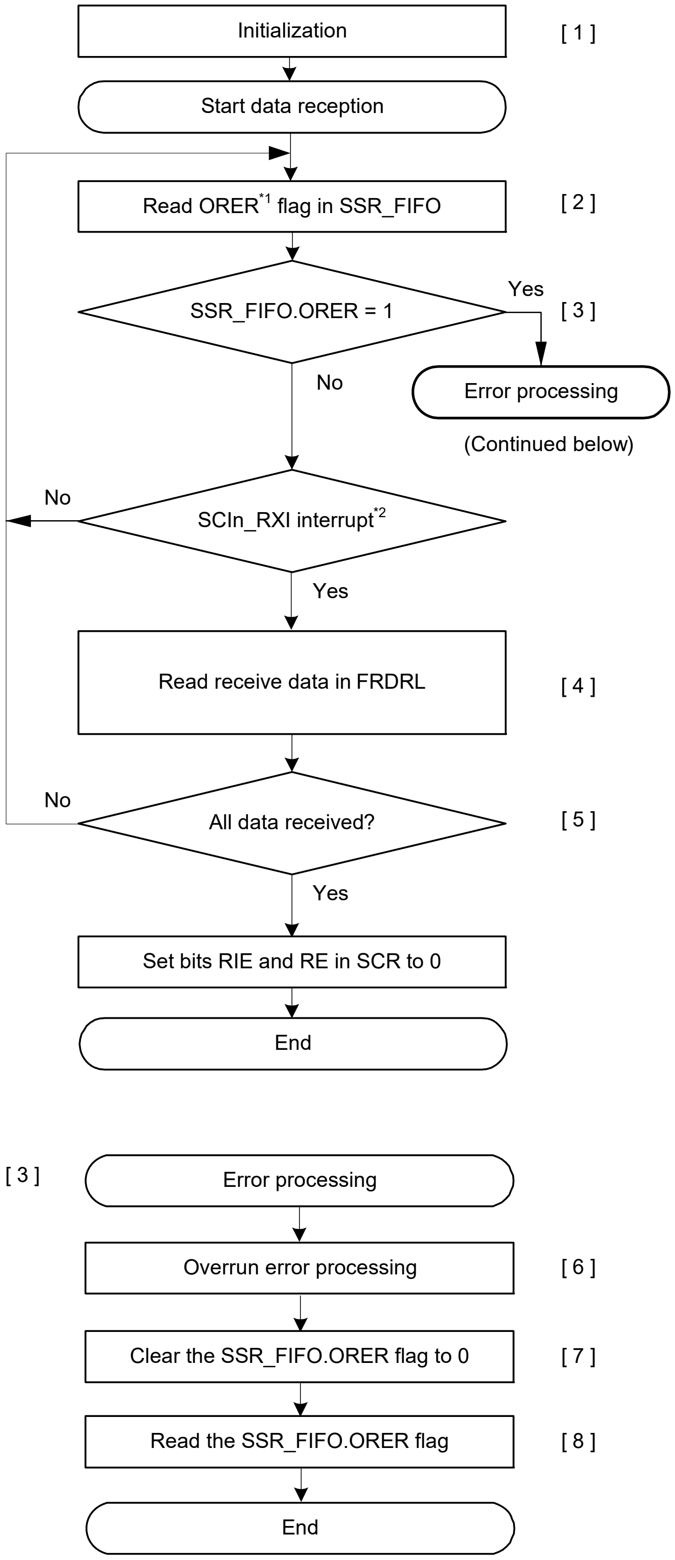
(2) FIFO selected

Figure 34.42 shows an example of serial reception in clock synchronous mode with FIFO selected.

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the CTSn_RTSn pin goes low.
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR_FIFO.ORER bit is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Received data is not transferred to the FRDRL*[1] register.
4. When data reception completes successfully, the receive data is transferred to the FRDRL*[1] register. The RDF bit is set to 1 when the amount of the receive data stored in FRDRL is equal to or greater than the specified receive triggering number. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to FRDRL*[2] in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to FRDRL is less than the RTS trigger number, the CTSn_RTSn pin goes low.

Note 1. In clock synchronous mode, FRDRH is not used.
Note 2. Read data in order from FRDRH to FRDRL when RDF and ORER are read with receive data.



[1] SCI initialization:
Make input port-pin settings for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR_FIFO, perform the relevant error processing, then set the SSR_FIFO.ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] The receive data stored in FRDRL register is read until the number of stored data is less than the FCR.RTRG value. Software can check readable data in FDR.R[4:0].

[5] Serial reception continuation procedure:
To continue serial reception, before overrun error occurs, finish reading the receive data in FRDRL and clear the SSR_FIFO.RDF flag to 0. The FRDRL data can also be read by activating the DMAC or DTC by an SCIn_RXI interrupt request. The RDF flag is cleared automatically in this case. Do not write to the RDF flag.

[6] Processing in response to an overrun error:
Read the FRDRL. In combination with step [7], this enables correct reception of the next frame.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Note 1. It can also be read from FRDRH.ORER. However, to clear the ORER flag, write 0 to the associated bit in the SSR_FIFO register.

Note 2. It should be all receive data and is an integer times the FIFO triggering number.

Figure 34.42 Example flow of serial reception in clock synchronous mode with FIFO selected

34.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

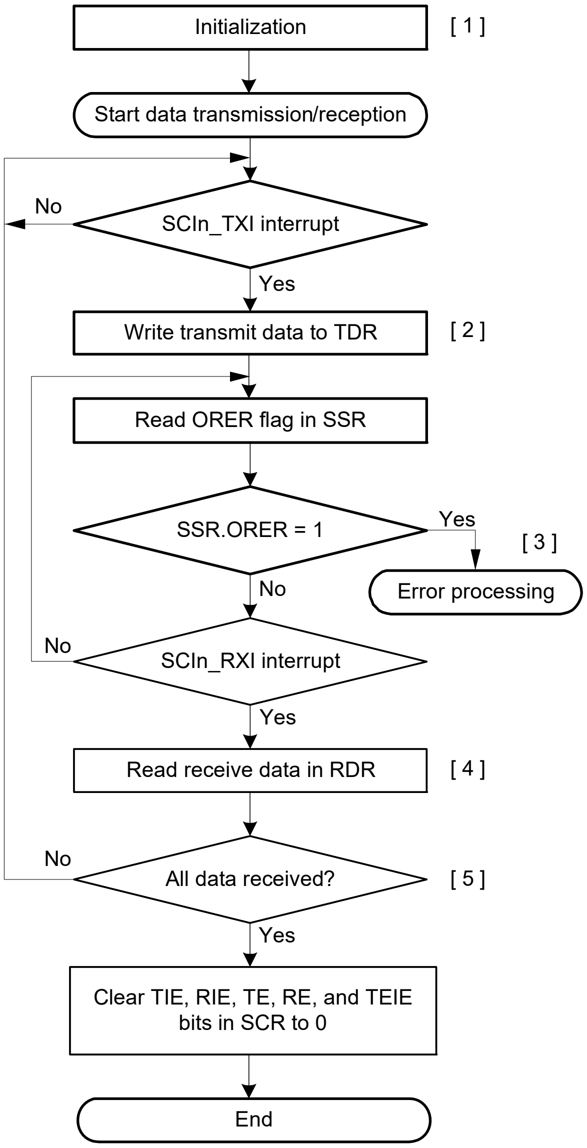
Figure 34.43 shows an example flow of simultaneous serial transmission and reception operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmission and reception operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the SSR.TEND flag is set to 1.
2. Initialize the SCR register, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data reception.
2. Set the RIE and RE bits to 0, and then check that the receive error flag ORER in the SSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to TDR once in the SCIn_TXI interrupt request handling routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in RDR once in the SCIn_RXI interrupt request handling routine.

[5] Serial transmission or reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit [7]) of the current frame is received, finish reading the receive data in RDR by the SCIn_RXI interrupt. Also, before the MSB (bit [7]) of the current frame is transmitted, write data to TDR by the SCIn_TXI interrupt.
Transmit data can also be written to TDR by activating the DMAC or DTC by a transmit data empty interrupt (SCIn_TXI) request. Similarly, the RDR data can also be read by activating the DMAC or DTC by a receive data full interrupt (SCIn_RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first set the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 34.43 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 34.44 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

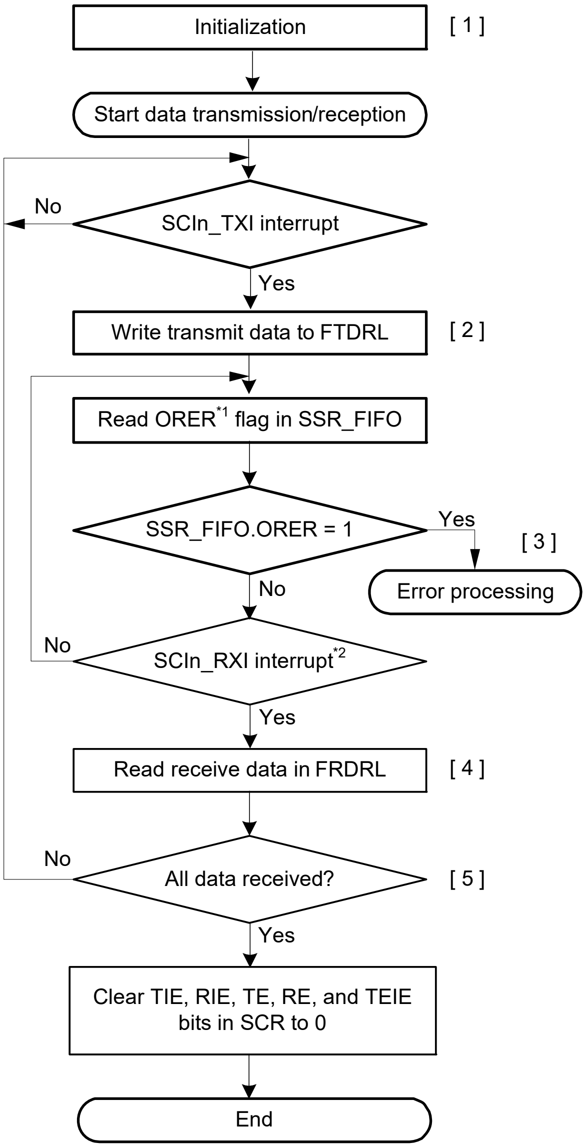
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the SSR_FIFO.TEND flag is set to 1.
2. Initialize the SCR register, then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.

2. Set the RIE and RE bits to 0, then check that the receive error flag ORER in SSR_FIFO is 0.
3. Set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to FTDRL in the SCIn_TXI Interrupt request handling routine. The amount of transmit data it is possible to write in is 16 minus the amount of stored transmit FIFO data.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR_FIFO, perform the relevant error processing, then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
The receive data stored in FRDRL register is read until the number of stored data is less than the FCR.RTRG value. Software can check readable data in FDR.R[4:0].

[5] Serial transmission or reception continuation procedure:
To continue serial reception, before overrun error occurs, finish reading the receive data in FRDRL and clear the SSR_FIFO.RDF flag to 0. To continue serial transmission, before transmission of the current transmit data has finished, write the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine and clear the SSR_FIFO.TDFE flag to 0.
Transmit data can also be written to FTDRL by activating the DMAC or DTC by a transmit FIFO data empty interrupt (SCIn_TXI) request. Similarly, the FRDRL data can also be read by activating the DMAC or DTC by a receive FIFO data full interrupt (SCIn_RXI) request. The RDF and TDFE flags are cleared automatically in this case. Do not write to the RDF and TDFE flags.

Note 1. ORER can also read from FRDRH.ORER. To clear the ORER flag, write 0 to SSR_FIFO.ORER.
Note 2. The total receive data amount must be an integral multiple of the FIFO triggering number.

Figure 34.44 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

34.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

34.6.1 Example Connection

Figure 34.45 shows an example connection between a smart card (IC card) and the MCU. As shown in Figure 34.45, because the MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the SCR_SMCI.TE and SCR_SMCI.RE bits to 1 with an IC card disconnected enables closed-loop transmission or reception, allowing self-diagnosis. To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

An output port of the MCU can be used to output a reset signal.

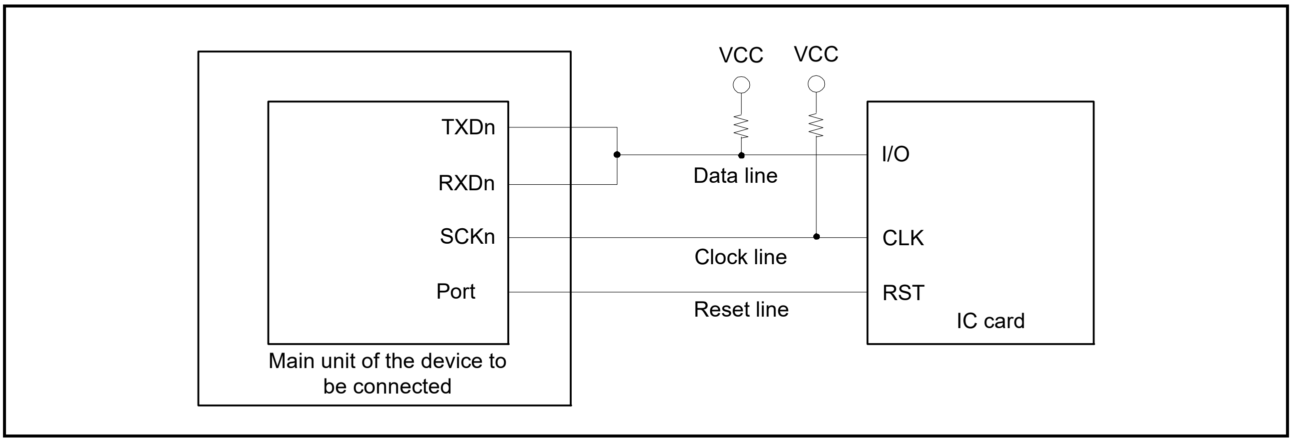


Figure 34.45 Example connection with a smart card (IC card)

34.6.2 Data Format (Except in Block Transfer Mode)

Figure 34.46 shows the data transfer formats in smart card interface mode:

- One frame consists of 8-bit data and a parity bit in asynchronous mode
- During transmission, at least 2 ETUs (elementary time unit — the time required for transferring 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame
- If a parity error is detected during reception, a low error signal is output for 1 ETU after 10.5 ETUs elapse from the start bit
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 ETUs.

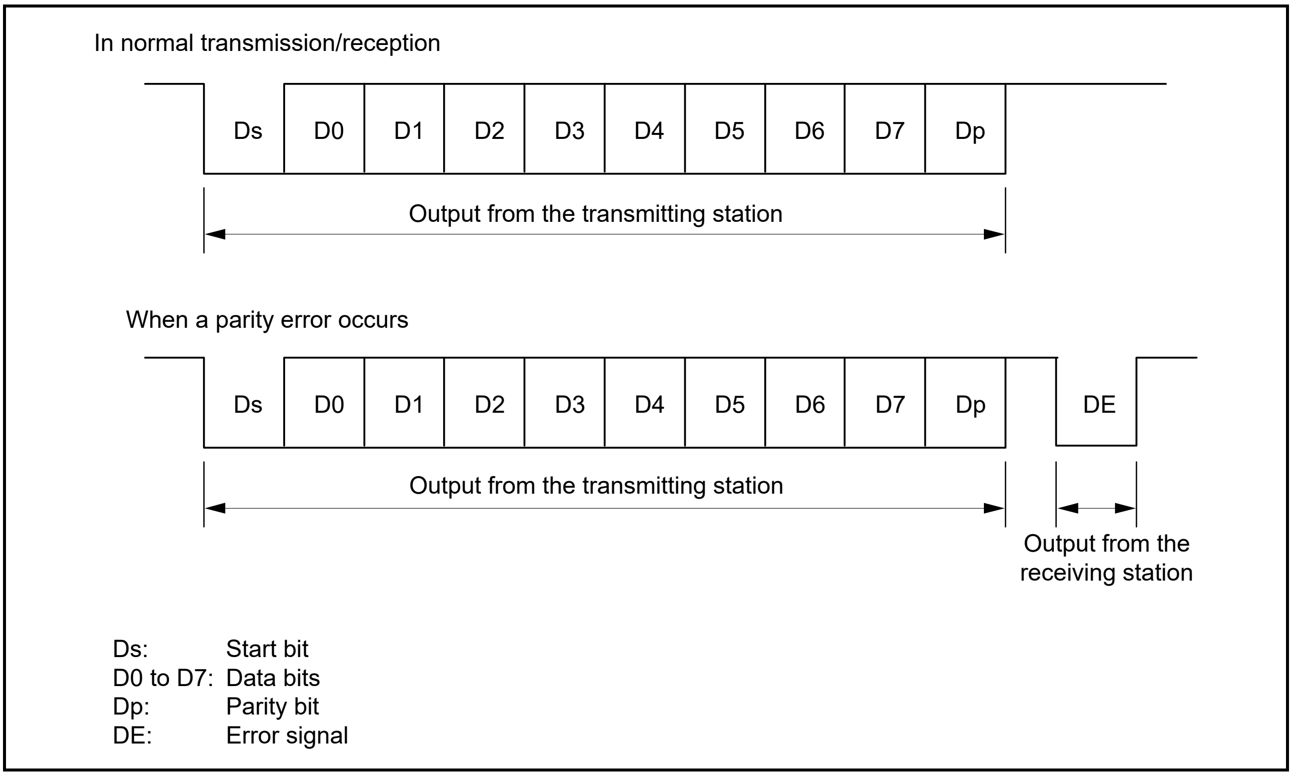


Figure 34.46 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedures in this section.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 indicate the Z and A states, respectively, and data is transferred with LSB-first for the start character, as shown in Figure 34.47. Therefore, data in the start character in the figure is 3Bh.

When using the direct convention type, write 0 to both the SCMR.SDIR and SCMR.SINV bits. Write 0 to the SMR_SMCI.PM bit to use even parity, which is required by the smart card standard.

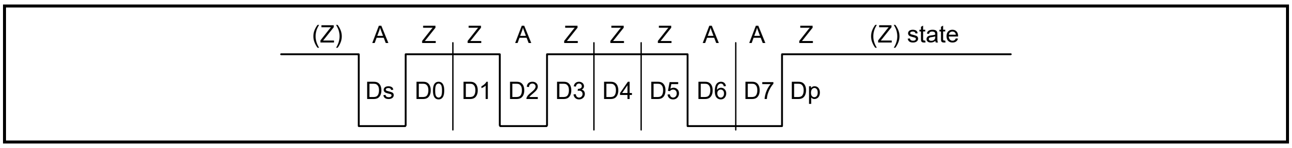


Figure 34.47 Direct convention with SDIR in SCMR = 0, SINV in SCMR = 0, and PM in SMR_SMCI = 0

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 indicate the A and Z states, respectively, and data is transferred with MSB-first for the start character, as shown in Figure 34.48. Therefore, data in the start character in the figure is 3Fh.

When using the inverse convention type, write 1 to both the SCMR.SDIR and SCMR.SINV bits. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to the Z state. Because the SINV bit of the MCU only inverts data bits D7 to D0, write 1 to the PM bit in SMR_SMCI to invert the parity bit for both transmission and reception.

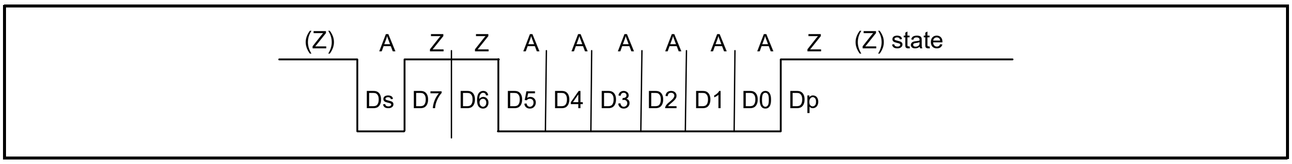


Figure 34.48 Inverse convention with SDIR in SCMR = 1, SINV in SCMR = 1, and PM in SMR_SMCI = 1

34.6.3 Block Transfer Mode

Block transfer mode differs from non-block transfer mode of smart card interface mode as follows:

- Even if a parity error is detected during reception, no error signal is output. Because the PER bit in SSR_SMCI is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 ETU is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in SSR_SMCI is set to 11.5 ETUs after transmission starts
- In block transfer mode, the ERS flag in SSR_SMCI indicates the error signal status as in non-block transfer mode of smart card interface mode, but the flag is read as 0 because no error signal is transferred.

34.6.4 Receive Data Sampling Timing and Reception Margin

Only the clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate set up in the SCMR.BCP2 and the SMR_SMCI.BCP[1:0] bits.

For data reception, the falling edge of the start bit is sampled with the base clock to perform synchronization.

Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 34.49. The reception margin is determined by the following formula:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ [\%]$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined using the following formula:

$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$

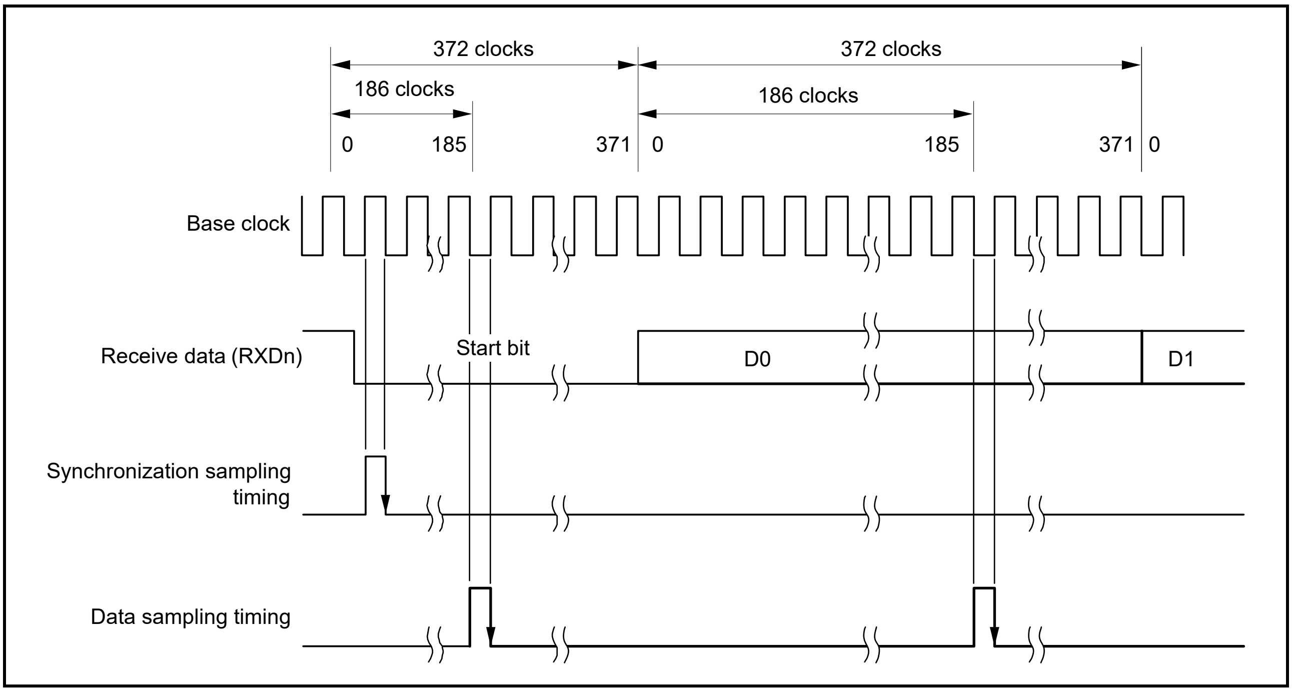


Figure 34.49 **Receive data sampling timing in smart card interface mode when clock frequency is 372 times the bit rate**

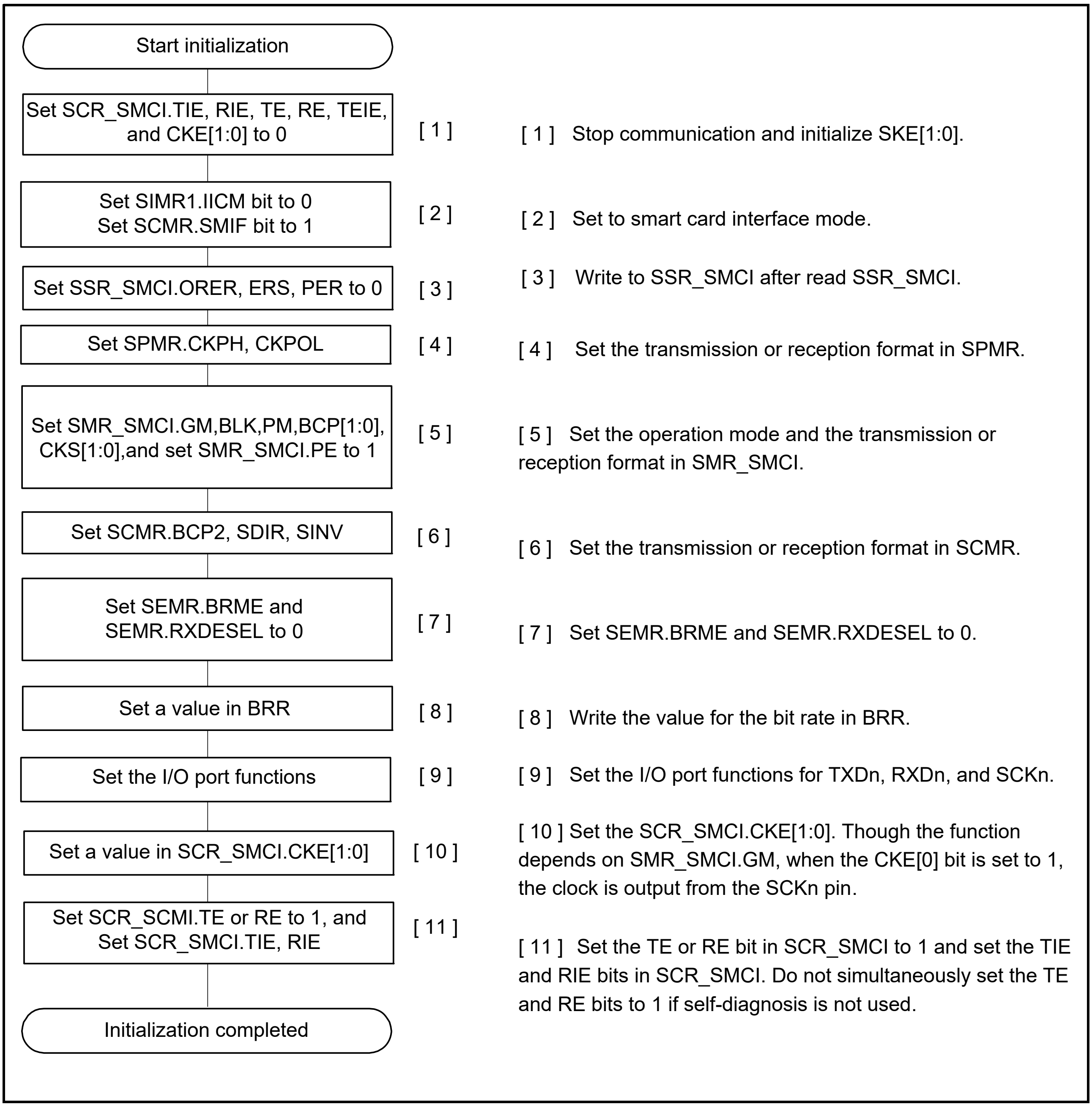
34.6.5 Initialization of the SCI

Before transmitting and receiving data, write the initial value 00h in the SCR_SMCI register and initialize the SCI following the example flow shown in Figure 34.50.

Always set the initial value in the TIE, RIE, TE, RE, TEIE bits in the SCR_SMCI register before switching from transmission to reception mode or from reception to transmission mode. When SCR_SMCI.RE is set to 0, the RDR register is not initialized.

To change from reception mode to transmission mode, first check that reception has completed, then initialize the SCI. At the end of initialization, set SSR_SMCI.TE = 1 and SSR_SMCI.RE = 0. Reception completion can be verified by reading the SCIn_RXI request, ORER, or PER flag in SSR_SMCI.

To change transmission mode to reception mode, first check that transmission has completed, then initialize the SCI. At the end of initialization, set SSR_SMCI.TE = 0 and SSR_SMCI.RE = 1. Transmission completion can be verified by reading the TEND flag in SSR_SMCI.



[1] Stop communication and initialize SKE[1:0].

[2] Set to smart card interface mode.

[3] Write to SSR_SMCI after read SSR_SMCI.

[4] Set the transmission or reception format in SPMR.

[5] Set the operation mode and the transmission or reception format in SMR_SMCI.

[6] Set the transmission or reception format in SCMR.

[7] Set SEMR.BRME and SEMR.RXDESEL to 0.

[8] Write the value for the bit rate in BRR.

[9] Set the I/O port functions for TXDn, RXDn, and SCKn.

[10] Set the SCR_SMCI.CKE[1:0]. Though the function depends on SMR_SMCI.GM, when the CKE[0] bit is set to 1, the clock is output from the SCKn pin.

[11] Set the TE or RE bit in SCR_SMCI to 1 and set the TIE and RIE bits in SCR_SMCI. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis is not used.

Figure 34.50 Example flow of SCI initialization in smart card interface mode

Figure 34.51 shows a timing diagram when data transmission is performed by transitioning to smart card interface mode according to the flow in Figure 34.50. Figure 34.51 shows when the GM bit in SMR_SMCI is set to 0. The timing in Figure 34.51 shows when the port is connected as SCKn pin and TXDn pin, the pins are Hi-Z because CKE[0] bit in SCR_SMCI is 0.

Start the clock output to the SCK pin by setting CKE[0] bit in SCR_SMCI to 1, then start data transmission by writing transmit data after setting TE bit in SCR_SMCI to 1. When the TE bit in SCR_SMCI changes from 0 to 1, there is a preamble period for one frame before data transmission starts. In smart card interface mode, the TXDn pin is Hi-Z during the preamble period. Pull-up or pull-down for the SCKn and TXDn pins is required outside the MCU.

In the smart card interface mode, even when the TE and RE bits in SCR_SMCI are 0, the clock is continuously output if the clock output setting is used.

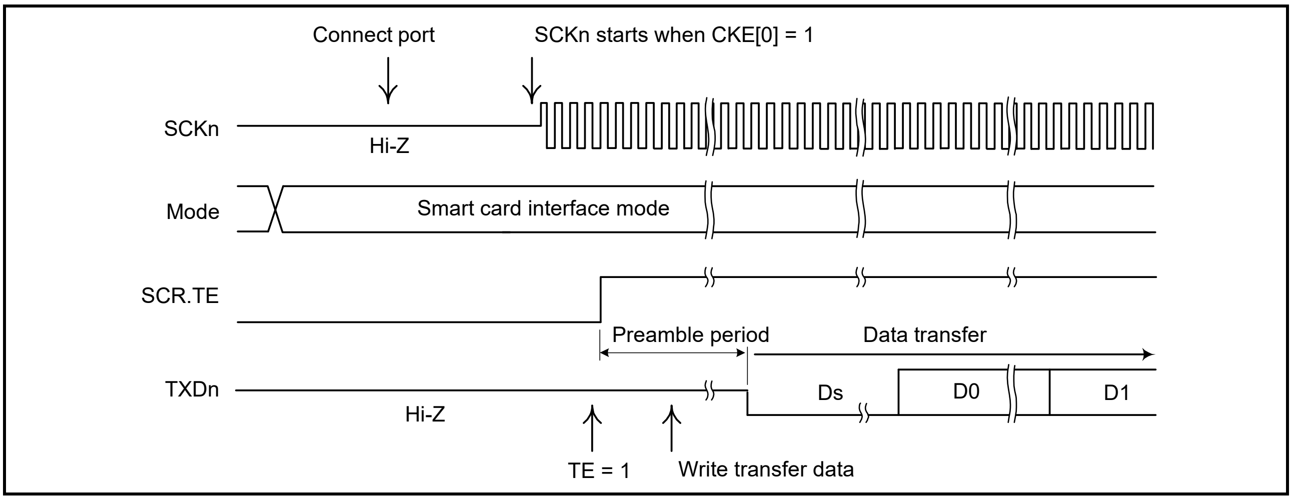


Figure 34.51 Example timing of data transmission in smart card interface mode

34.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in non-smart card interface mode, in that an error signal is sampled and data can be re-transmitted in smart card mode. Figure 34.52 shows the data re-transfer operation during transmission.

- [1] indicates when an error signal from the receiver end is sampled after 1-frame data is transmitted, the SSR_SMCI.ERS flag is set to 1. If the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
- [2] indicates for a frame in which an error signal is received, the SSR_SMCI.TEND flag is not set. Data is re-transferred from TDR to TSR, allowing automatic data retransmission.
- [3] indicates if no error signal is returned from the receiver, the ERS flag is not set to 1.
- [4] indicates the SCI determines that transmission of 1-frame data, including the re-transfer, is complete, and the TEND flag is set. If the SCR_SMCI.TIE bit is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 34.54 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DTC or DMAC.

When the SSR_SMCI.TEND flag is set to 1 in transmission and when the SCR_SMCI.TIE bit is 1, an SCIn_TXI interrupt request is generated.

The DTC or DMAC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 before enabling an SCIn_ERI interrupt request to be generated if an error occurs, and clear the ERS flag to 0.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings.

For DTC or DMAC settings, see section 17, DMA Controller (DMAC) and section 18, Data Transfer Controller (DTC).

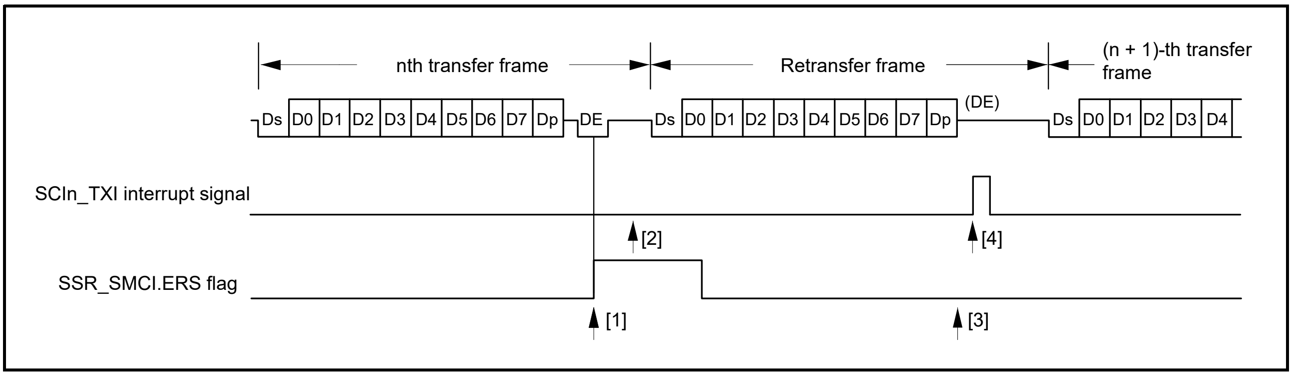
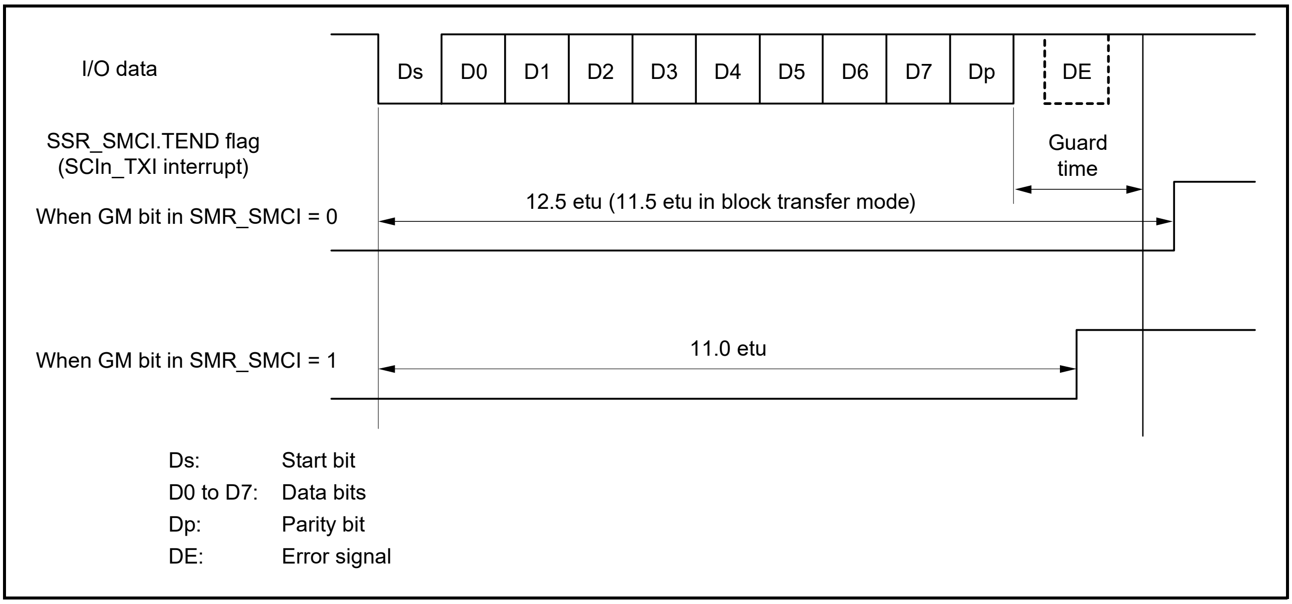


Figure 34.52 Data retransfer operation in SCI transmission mode

The SSR_SMCI.TEND flag is set at different timings depending on the SMR_SMCI.GM bit setting. Figure 34.53 shows the TEND flag generation timing.



Ds:	Start bit
D0 to D7:	Data bits
Dp:	Parity bit
DE:	Error signal

Figure 34.53 SSR.TEND flag generation timing during transmission

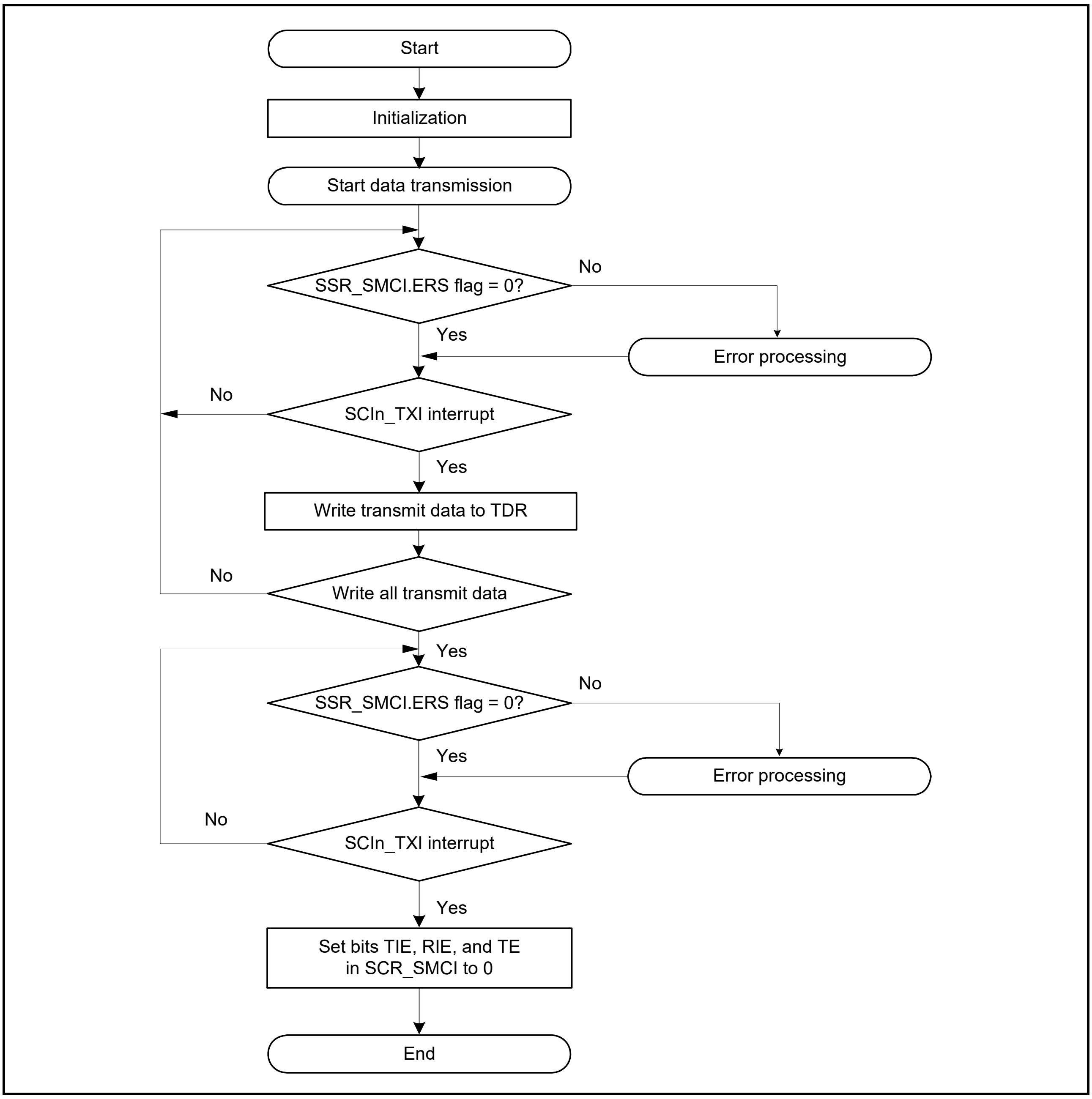


Figure 34.54 Example flow of smart card interface transmission

34.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 34.55 shows the data re-transfer operation in reception mode.

- [1] indicates if a parity error is detected in the receive data, the SSR_SMCI.PER flag is set to 1. When the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
- [2] indicates for a frame in which a parity error is detected, no SCIn_RXI interrupt is generated.
- [3] indicates when no parity error is detected, the SCR_SMCI.PER flag is not set to 1.
- [4] indicates the data is determined to be received successfully. When the SCR_SMCI.RIE bit is 1, an SCIn_RXI interrupt request is generated.

Figure 34.56 shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the DTC or DMAC.

In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DTC or DMAC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in SSR_SMCI is set to 1, a receive error interrupt (SCIn_ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, allowing the data to be read.

When a reception is forced to terminate by setting SCR_SMCI.RE to 0 during operation, read the RDR register because the received data that is not yet read might be left in the RDR.

Note: For operations in block transfer mode, see section 34.3.9, Serial Data Reception in Asynchronous Mode.

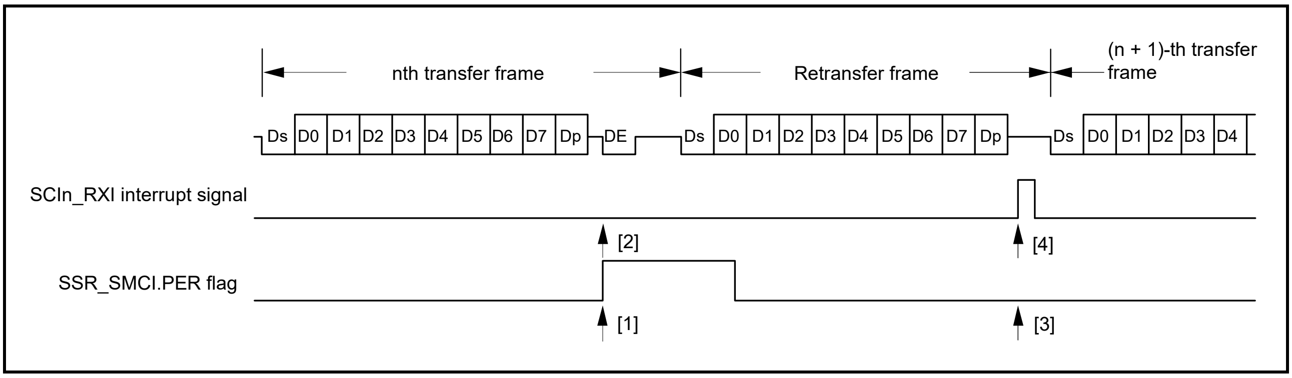


Figure 34.55 Data retransfer operation in SCI reception mode

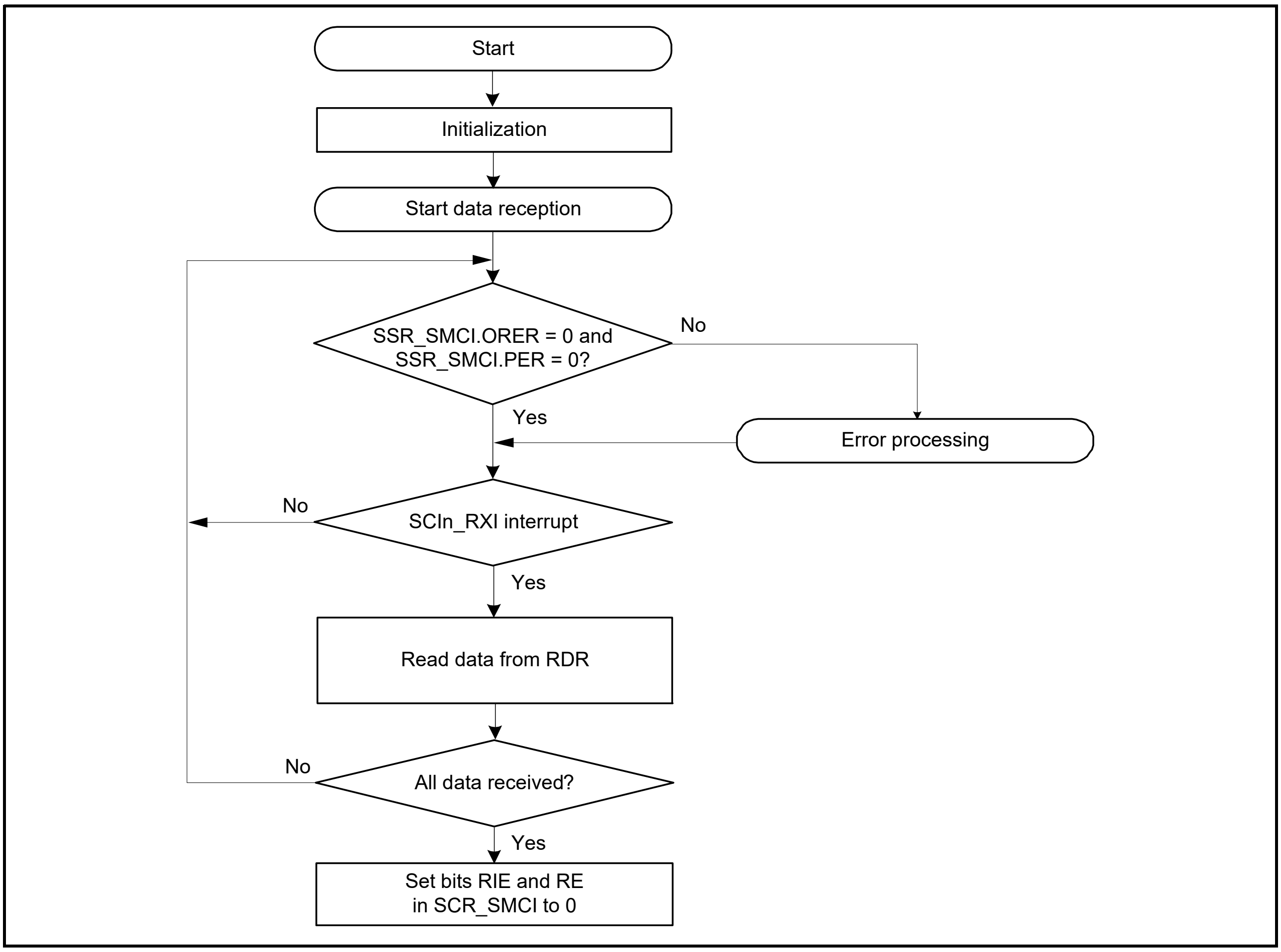


Figure 34.56 Example flow of smart card interface reception

34.6.8 Clock Output Control

When the GM bit in SMR_SMCI is set to 1, the clock output can be controlled by the CKE[1: 0] bits in SCR_SMCI. For details on the CKE[1:0] bits, see section 34.2.12, Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1). When setting the clock output, the base clock described in section 34.6.4, Receive Data Sampling Timing and Reception Margin is output.

Figure 34.57 shows an example timing for the clock output control when the CKE[1] bit in SCR_SMCI is set to 0 and the CKE[0] bit in SCR_SMCI is controlled.

When the GM bit in SMR_SMCI is 0, output control by the CKE[0] bit in SCR_SMCI is immediately reflected on the SCK pin, so there is a possibility that pulses with an unintended width might be output from the SCK pin.

When the GM bit in SMR_SMCI is 1, the clock with the same pulse width as the base clock is output even if the CKE[0] bit in SCR_SMCI is changed.

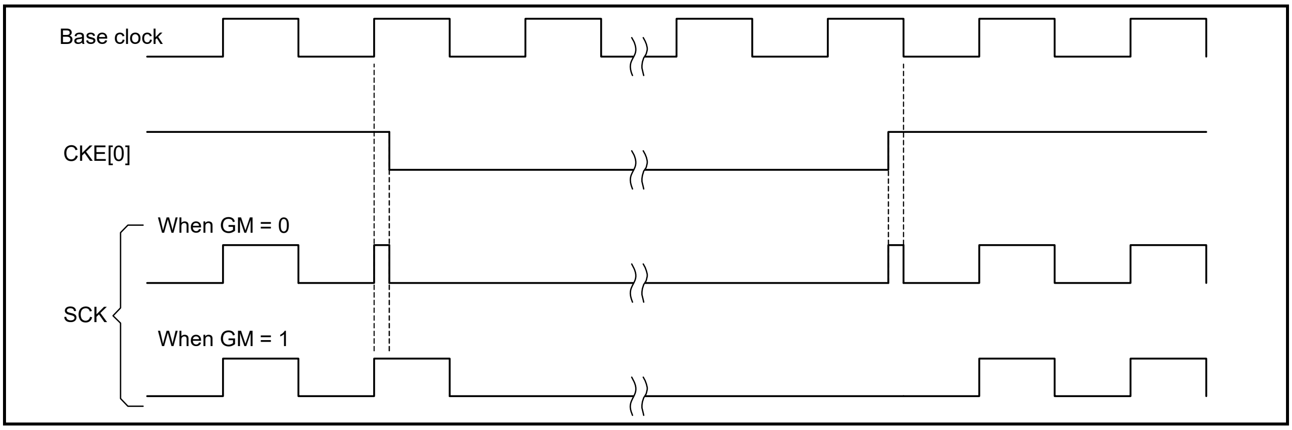


Figure 34.57 Clock output control

34.7 Operation in Simple IIC Mode

Simple IIC mode format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device can specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I^2C bus format and timing of the I^2C bus are shown in Figure 34.58 and Figure 34.59.

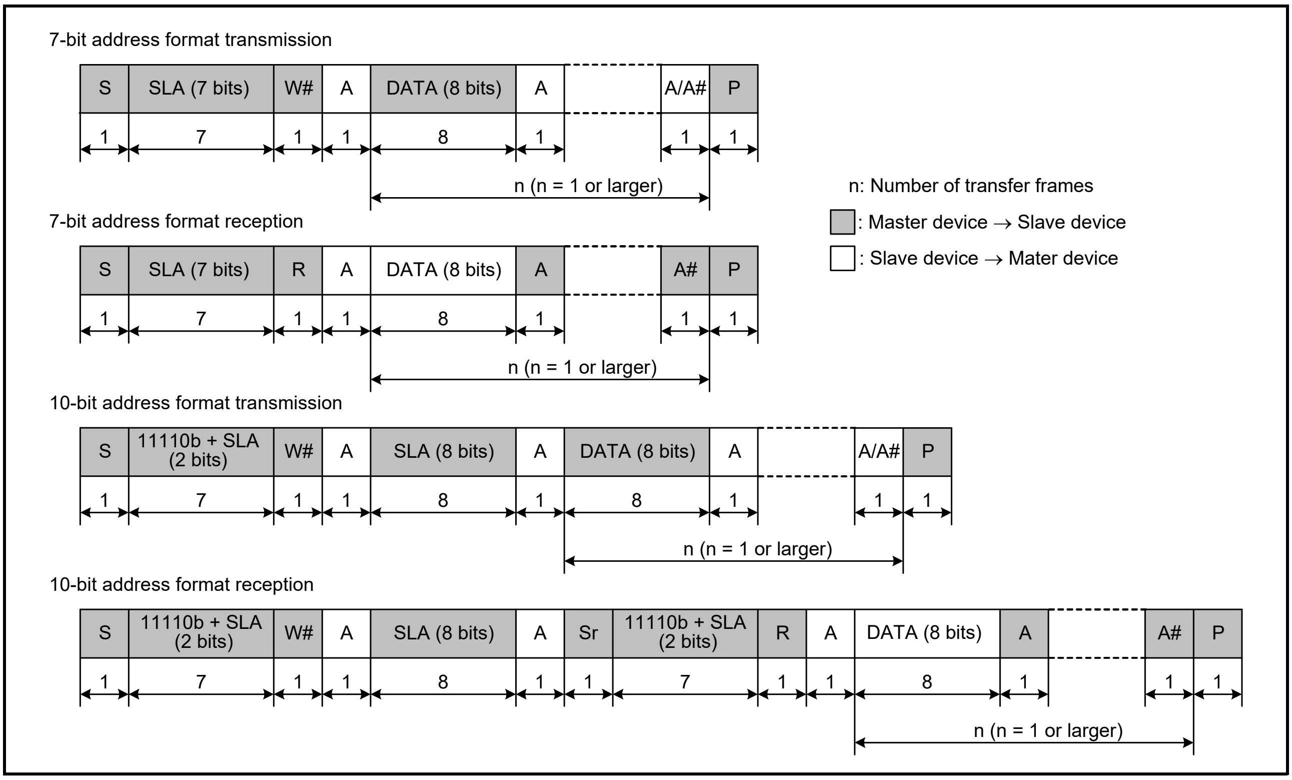


Figure 34.58 I^2C bus format

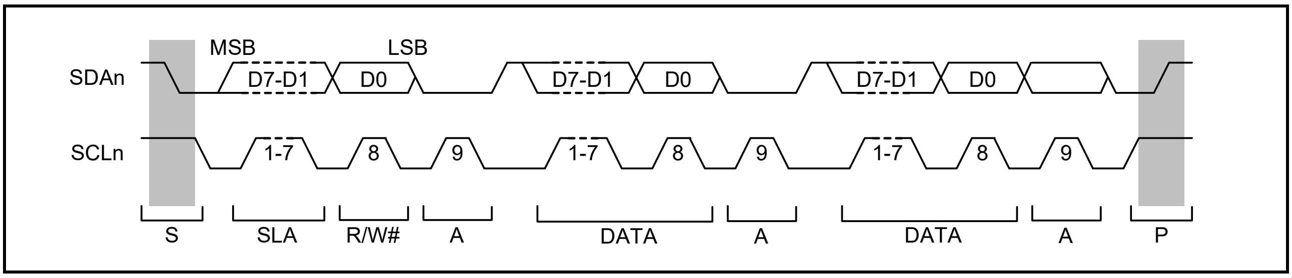


Figure 34.59 I2C bus timing when SLA is 7 bits

- S: Indicates a start condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high
- SLA: Indicates a slave address, by which the master device selects a slave device
- R/W#: Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.
- A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return low indicates ACK and return high indicates NACK.
- Sr: Indicates a restart condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high and after the setup time elapses
- DATA: Indicates the data being received or transmitted
- P: Indicates a stop condition, when the master device changes the level on the SDAn line from low to high while the SCLn line is high.

34.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the SIMR3.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state
- The hold time for the start condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SCLn line falls (from the high level to the low level), the IICSTAREQ bit in SIMR3 is set to 0, and a start-condition generated interrupt is output.

Writing 1 to the IICRSTAREQ bit in SIMR3 causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SDAn line is released and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the restart condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SDAn line falls (from the high level to the low level)
- The hold time for the restart condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SCLn line falls (from the high level to the low level), the SIMR3.IICRSTAREQ bit is set to 0, and a restart-condition generated interrupt is output.

Writing 1 to the SIMR3.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the BRR setting

- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the stop condition is set as half of a bit period at the bit rate determined by the BRR setting
- The SDAn line is released (transition from the low to the high level), the SIMR3.IICSTPREQ bit is set to 0, and a stop-condition generated interrupt is output.

Figure 34.60 shows the timing of operations in the generation of start, restart, and stop conditions.

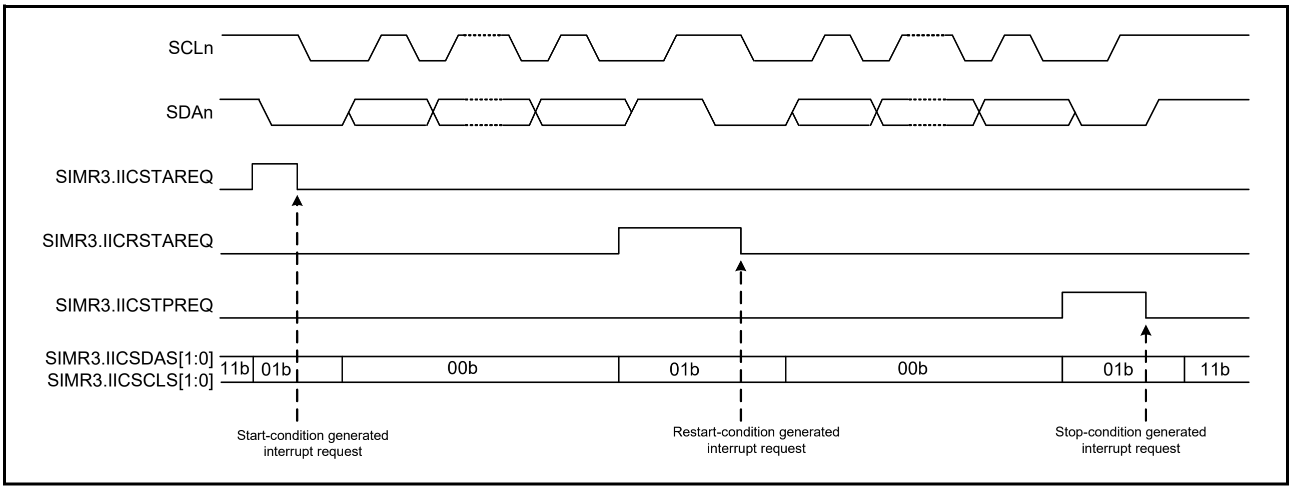


Figure 34.60 Timing of operations in generation of start, restart, and stop conditions

34.7.2 Clock Synchronization

The SCLn line can be driven low if a wait is inserted by a slave device at the other side of the transfer. Setting the SIMR2.IICCSC bit to 1 applies control to obtain synchronization when a difference arises between the levels of the internal SCLn clock signal and the level being input on the SCLn pin.

When the SIMR2.IICCSC bit is set to 1, the level of the internal SCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is being input on the SCLn pin. Counting to determine the period at a high level starts after the transition of the input on the SCLn pin to the high level.

The interval from this time until counting to determine the period at high level starts on the transition of the SCLn pin to the high level, is the total of the delay of SCLn output, delay for noise filtering of the input on the SCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLKA). The period at high level of the internal SCLn clock is extended even when other devices do not place the low level on the SCLn line.

If the SIMR2.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the SIMR2.IICCSC bit is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a wait period into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a wait period after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the waiting period, generation of the condition itself is not guaranteed. Figure 34.61 shows an example operation for synchronizing the clocks.

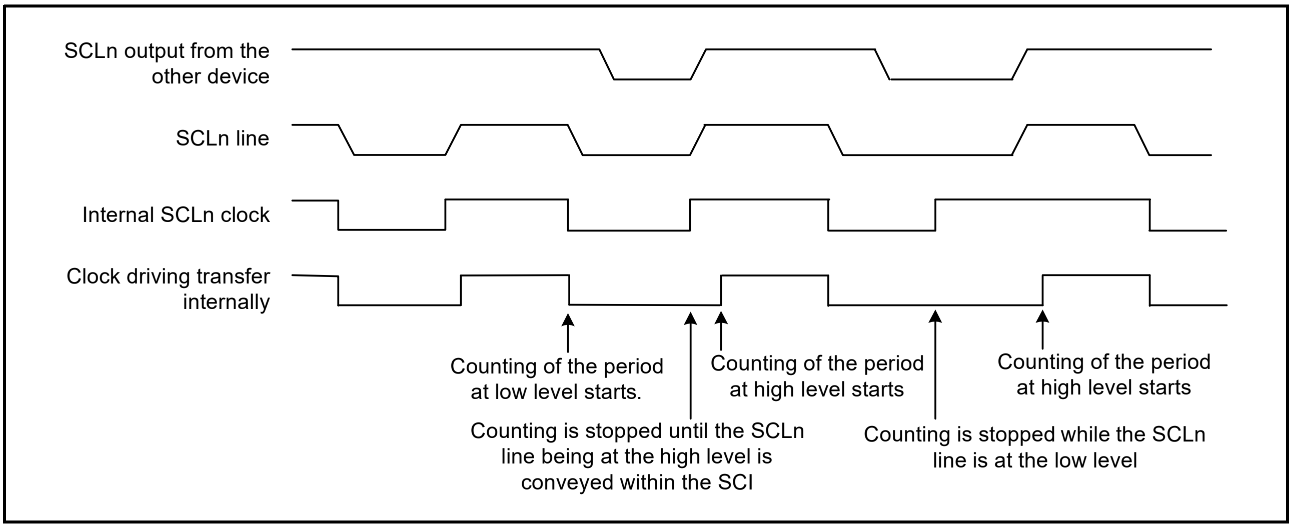


Figure 34.61 Example operations for clock synchronization

34.7.3 SDA Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. Delay settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLKA, by the divisor selected in the SMR.CKS[1:0] bits). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and acknowledge bit.

If the SDA output delay is shorter than the time for the level on the SCLn pin to fall, the change of the output on the SDAn pin starts while the output level on the SCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SDAn pin specify times greater than the time output on the SCLn pin takes to fall (300 ns for IIC in standard mode and fast mode).

Figure 34.62 shows the timing of delays in SDA output.

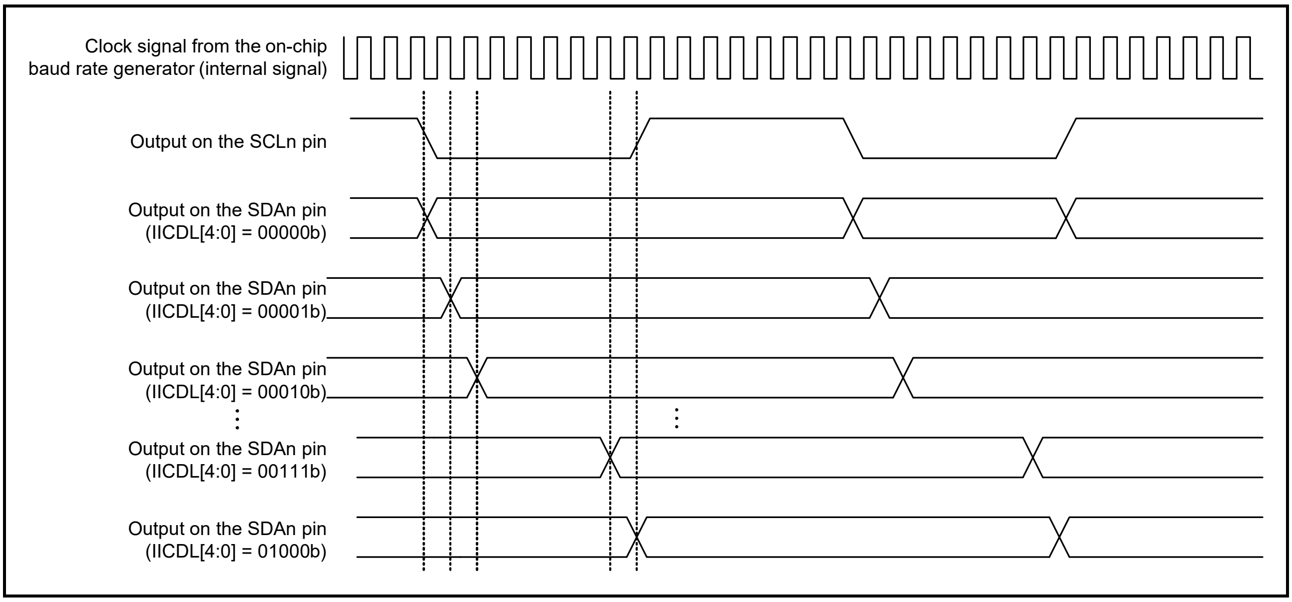


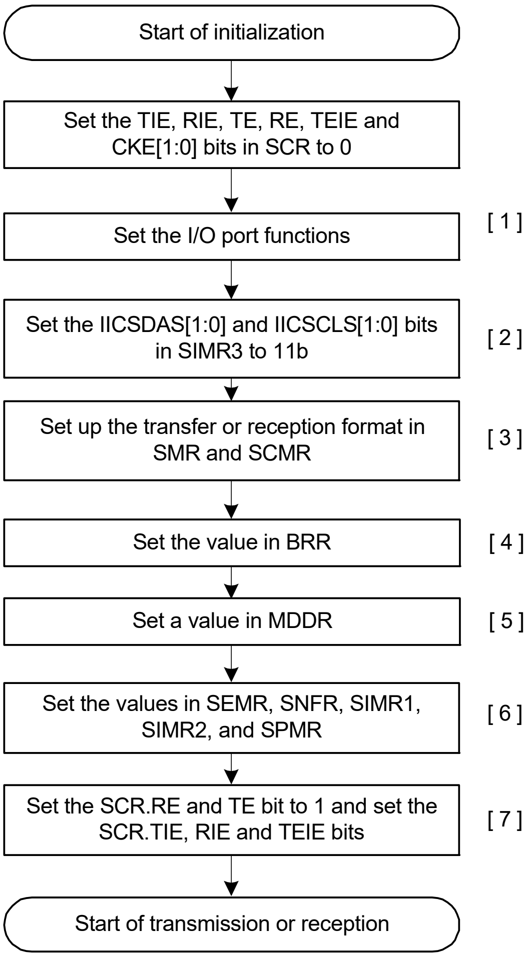
Figure 34.62 Timing of delays in SDA output

34.7.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 00h to SCR and initialize the interface following the example shown in Figure 34.63.

Always set SCR to its initial value before making any changes to the operating mode or transfer format.

In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.



[1] Make I/O port settings that allow use (on N-channel open-drain output pins) of the SCLn and SDAn pin functions.

[2] Place the SCLn and SDAn pins in the high-impedance state until a start condition is to be generated.

[3] Set the format for transmission and reception in SMR and SCMR.
In SMR, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In SCMR, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[4] Write the value for the desired bit rate to BRR.

[5] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is cleared to 0.

[6] Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR.
Set the NFEN and BRME bits in SEMR.
In SNFR, set the NFCS[2:0] bits.
In SIMR1, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In SIMR2, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In SPMR, set all the bits to 0.

[7] Set the RE and TE bits in the SCR to 1. Then, set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0). Setting the TE and RE bits to 1 makes the SCLn and SDAn pin functions available.

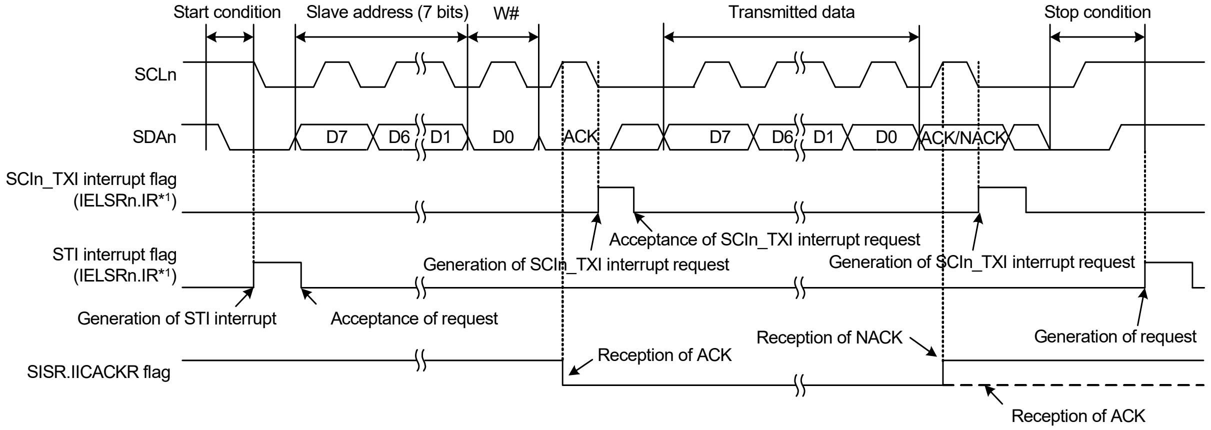
Figure 34.63 Example flow of SCI initialization in simple IIC mode

34.7.5 Operation in Master Transmission in Simple IIC Mode

Figure 34.64 and Figure 34.65 show examples of master transmission and Figure 34.66 shows an example flow of data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and the value of the SCR.RIE bit is assumed to be 0 (SCIn_RXI and SCIn_ERI interrupt requests are disabled). See Table 34.29 for more information on the STI interrupt.

When 10-bit slave addresses are in use, steps [3] and [4] in Figure 34.66 are repeated twice.

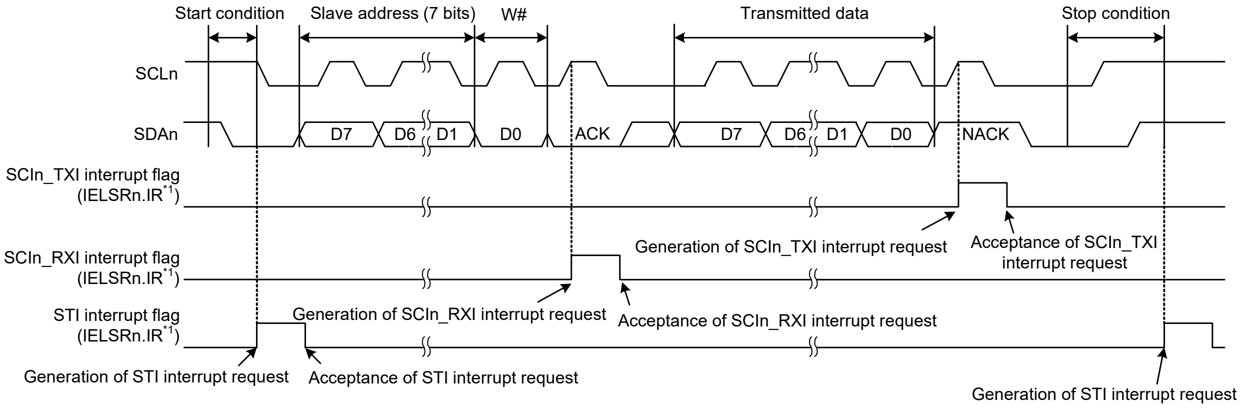
In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.64 **Example 1 of operations for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts**

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When the NACK is received, error processing such as transmission stop and retransmission is performed using the NACK interrupt as the trigger.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.65 **Example 2 of operations for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts**

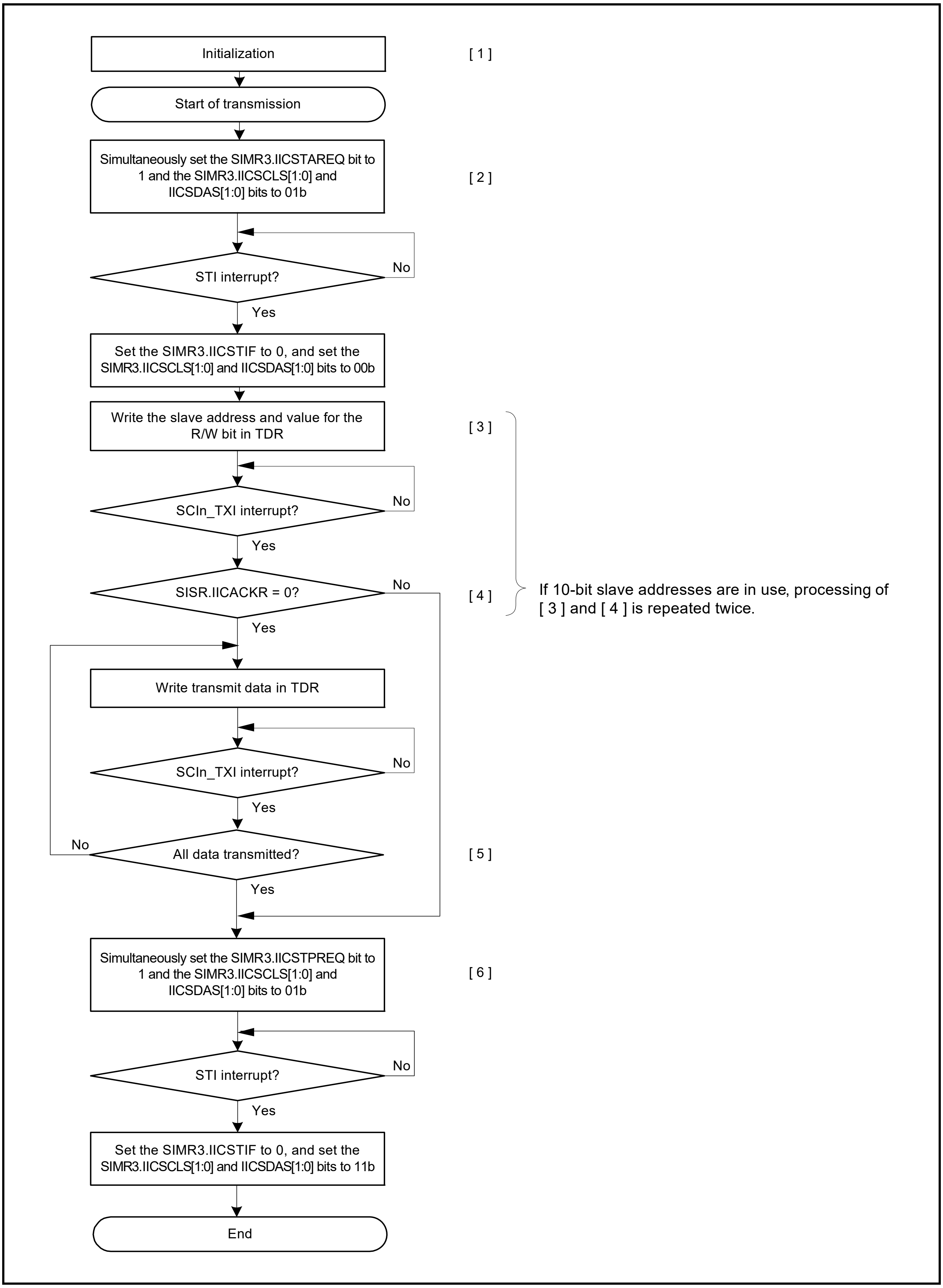


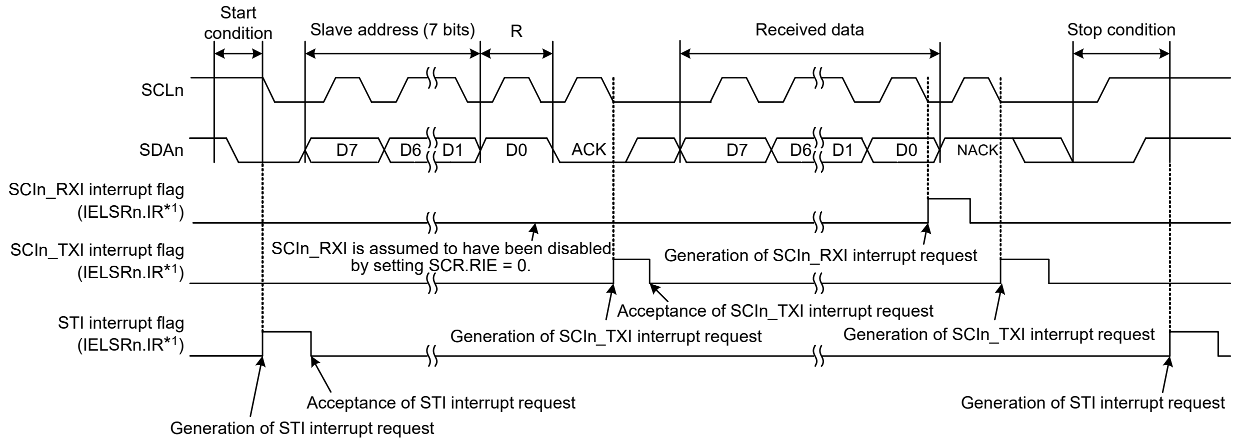
Figure 34.66 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

34.7.6 Master Reception in Simple IIC Mode

Figure 34.67 shows an example operation in simple IIC mode master reception and Figure 34.68 shows an example flow of master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.67 Example operations for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts

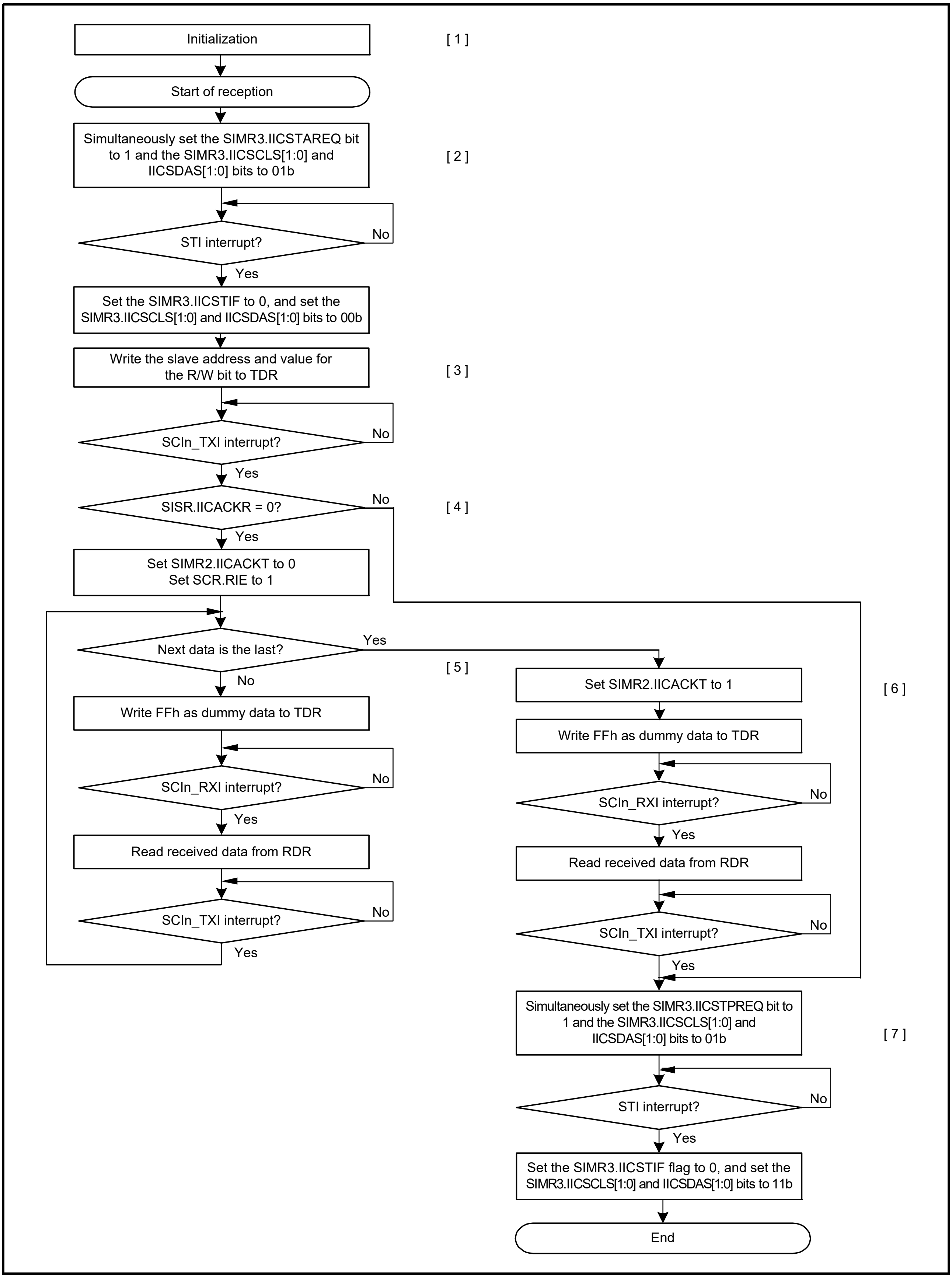


Figure 34.68 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts

34.8 Operation in Simple SPI Mode

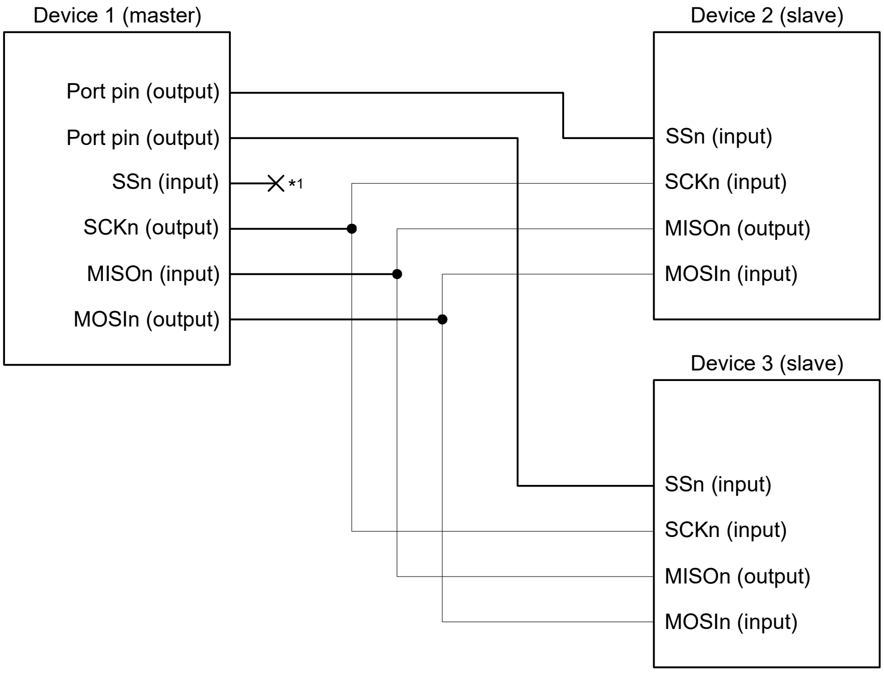
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Using the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) and setting the SPMR.SSE bit to 1 places the SCI in simple SPI mode. However, the SSn pin function on the master side is not required for connection of the device used as the master in simple SPI mode when the configuration only has a single master. Therefore, set the SPMR.SSE bit to 0 in such cases.

Figure 34.69 shows an example of connections for simple SPI mode. Control a general port pin to produce the SSn output signal from the master.

In simple SPI mode, data is transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the SCMR.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a shared clock signal. Additionally, because both the transmitter and receiver have a buffered structure, writing the next transmit data while transmission is in progress and reading previously received data while reception is in progress are both possible. This enables continuous transfer.



Note 1. The SSn input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 34.69 Example connections using simple SPI mode in single master mode with SPMR.SSE bit = 0

34.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1).

Table 34.25 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 34.25 States of pins by mode and input level on SSn pin

Mode	Input on SSn pin	State of TXDn pin	State of RXDn pin	State of SCKn pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn pin. This is equivalent to input of a high level on the SSn pin. Because the SSn pin function is not required, the pin is available for other purposes.

Note 2. The MOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).

Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

34.8.2 SS Function in Master Mode

Setting the SCR.CKE[1:0] bits to 00b and the SPMR.MSS bit to 0 selects master operation.

In single-master configurations (SPMR.SSE = 0), the SSn pin is not used, and so transmission or reception can proceed regardless of the value of the SSn pin.

When the level on the SSn pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission.

When the level on the SSn pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. The MOSIn output and SCKn pins are placed in the high-impedance state and starting transmission or reception is not possible. Additionally, the value of the SPMR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Even if a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn pin outputs are placed in the high-impedance state after completion of the transfer. Use a general port pin to produce the SS output signal from the master.

34.8.3 SS Function in Slave Mode

Setting the SCR.CKE[1:0] bits to 10b and the SPMR.MSS bit to 1 selects slave operation. When the SSn pin is high, the MISOn output pin is in the high-impedance state and clock input through the SCKn pin is ignored. When the SSn pin is low, clock input through the SCKn pin is valid and transmission or reception can proceed.

If the input on the SSn pin changes from low to high during transmission or reception, the MISOn output pin is placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception continues at the rate of the clock input through the SCKn pin until processing for the character being transmitted or received is complete, after which it stops, and the appropriate interrupt (SCIn_TXI, SCIn_RXI, or SCIn_TEI) is generated.

34.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 34.70. The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

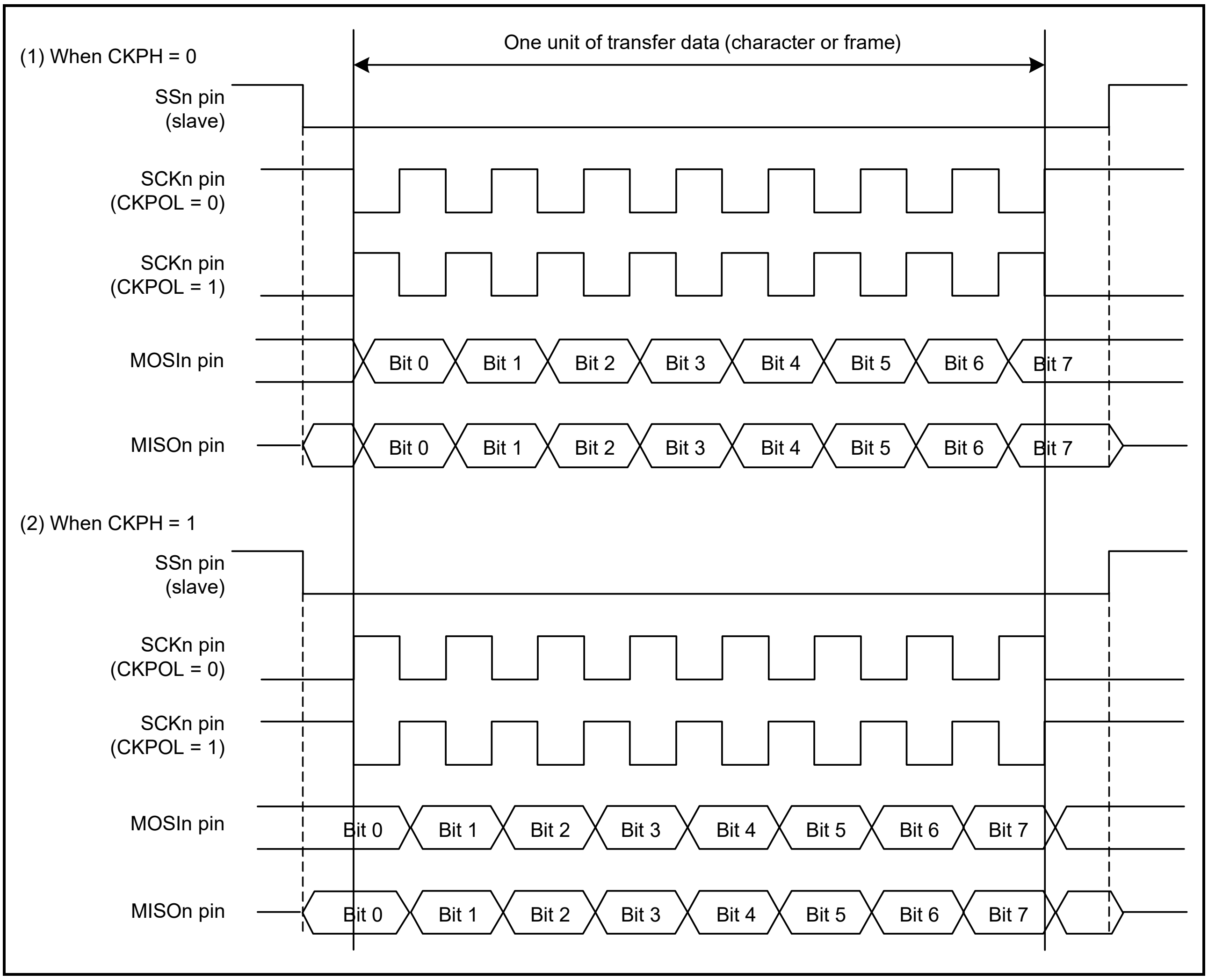


Figure 34.70 Relation between clock signal and transmit or receive data in simple SPI mode

34.8.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See Figure 34.32 for an example initialization flow. The CKPOL and CKPH bits in the SPMR register must be set to ensure that the clock signal is suitable for both master and slave devices.

Always initialize the SCR register before making any changes to the operating mode or transfer format.

Note 1. Only the RE bit is set to 0. The SSR.ORER, FER, PER, and RDR flags are not initialized.
Note 2. Changing the value of the TE bit from 1 to 0 or from 0 to 1 leads to the generation of a transmit data empty interrupt (SCIn_TXI) if the value of the SCR.TIE bit is 1.

34.8.6 Transmission and Reception of Serial Data in Simple SPI Mode

In master operation, ensure that the SSn pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

34.9 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in the MDDR register when the PCLKA is selected in the CKS[1:0] bits in SMR/SMR_SMCI.

Figure 34.71 shows an example where the PCLKA is selected in the CKS[1:0] bits in SMR/SMR_SMCI, the BRR bit is set to 0, and the MDDR is set to 160 in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256).

Note: Enabling an internal clock causes bias, and expansion and contraction are generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

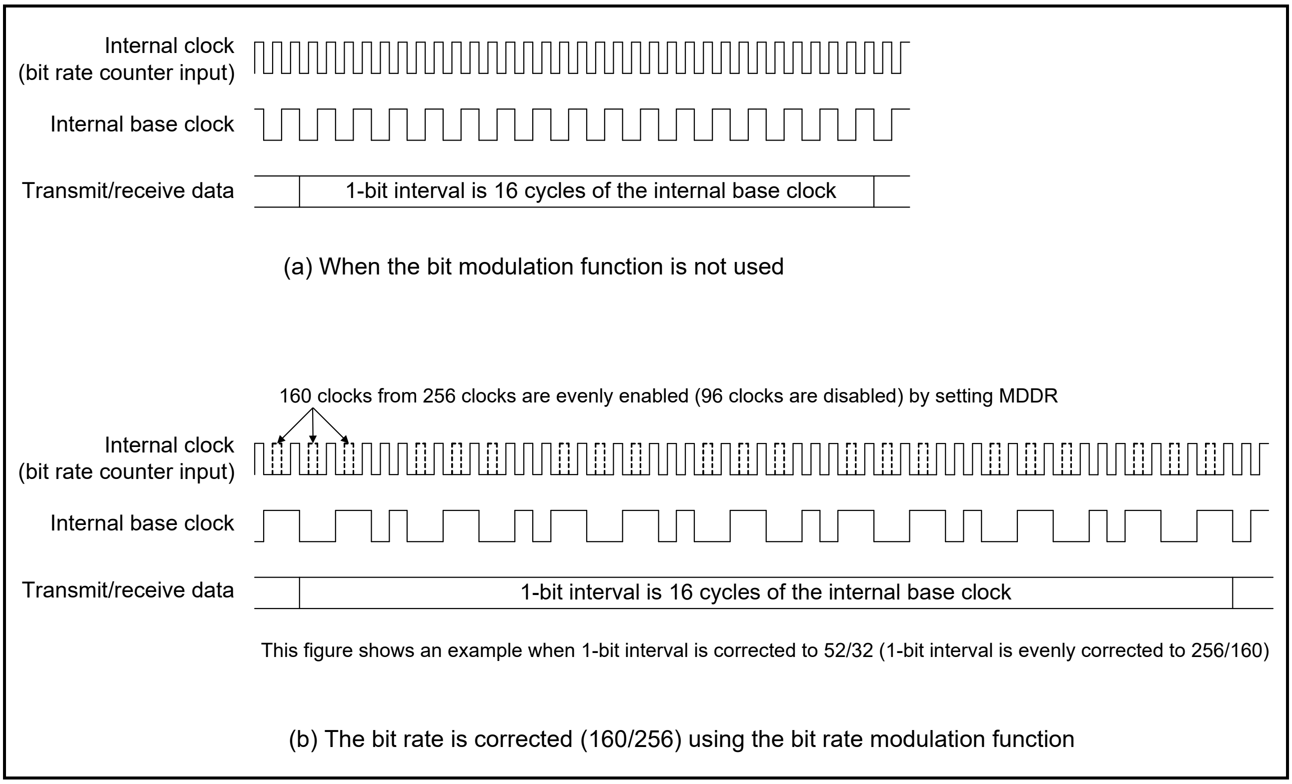


Figure 34.71 Example internal base clock when bit rate modulation function is used

34.10 Interrupt Sources

34.10.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (non-FIFO selected)

If the conditions for an SCIn_TXI and SCIn_RXI interrupt are satisfied while the interrupt status flag in the Interrupt Controller Unit (ICU) is 1, the ICU does not output the interrupt request but retains it internally (with a capacity for retention of one request per source).

When the interrupt status flag in the ICU is set to 0, the interrupt request retained within the ICU is output. The internally retained interrupt request is automatically discarded when the actual interrupt is output. Clearing of the associated interrupt enable bit (the TIE or RIE bit in the SCR/SCR_SMCI) can also be used to discard an internally retained interrupt request.

34.10.2 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (FIFO selected)

Even when an interrupt status flag in the ICU is set to 1, the SCIn_TXI and SCIn_RXI interrupts do not output interrupt requests to the ICU. When an interrupt status flag of the ICU is cleared to 0, and if the conditions for an SCIn_TXI and SCIn_RXI interrupts are satisfied, an interrupt request is generated.

34.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 34.26 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from the TDR or TDRHL register*1 to the TSR register. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the SCR.TE and SCR.TIE bits to 1 at the same time. An SCIn_TXI interrupt request can activate the DTC or DMAC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting the SCR.TE bit to 1 when SCR.TIE is 0 or by setting the SCR.TIE bit to 1 when the SCR.TE is 1.*2

When new data is not written by the time of transmission of the last bit of the current transmit data and SCR.TEIE is 1, the SSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated. Additionally, when SCR.TE is 1, the SSR.TEND flag retains the value 1 until more transmit data is written to the TDR or TDRHL register*1, and setting SCR.TEIE to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR or TDRHL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the SCIn_TEI interrupt request.

If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated when received data is stored in the RDR register. An SCIn_RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting any of the ORER, FER, and PER flags in the SSR register to 1 while the SCR.RIE bit is 1 leads to the generation of an SCIn_ERI interrupt request. An SCIn_RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the SCIn_ERI interrupt request.

(2) FIFO selected

Table 34.27 lists interrupt sources in FIFO selected mode.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when the stored amount of data in the FTDRL register becomes the threshold value indicated in FCR.TTRG or below. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the SCR.TE and SCR.TIE bits to 1 at the same time.

An SCIn_TXI interrupt request is not generated by setting SCR.TE to 1 when SCR.TIE is 0 or by setting SCR.TIE to 1 when SCR.TE is 1.

If SCR.TEIE is 1 and if the next data is not written to the FTDRL register by the time the last bit of the transmit data is sent, the SSR_FIFO.TEND flag is set to 1 and the SCIn_TEI interrupt request is generated.

If SCR.RIE is 1, the SCIn_RXI interrupt request is generated when the stored amount of data in the FRDRL register is equal to or greater than the threshold value indicated in FCR.RTRG. When RTRG is 0, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the SCR.RIE bit is 1, when the SSR_FIFO.ORER flag is set to 1 or data with a framing error or a parity error is stored in the FRDRL register, the SCIn_ERI interrupt request is generated. When the amount of data stored in the FRDRL register is at the threshold value or above, the SCIn_RXI interrupt request is also generated. The SCIn_ERI interrupt request can be canceled, in which case SSR_FIFO.ORER, FER, and PER flags are all cleared.

Note 1. When asynchronous mode and 9-bit data length are selected.

Note 2. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the interrupt request enable bit in the ICU rather than using the SCR.TIE bit. This approach can prevent the suppression of SCIn_TXI interrupt requests in the transfer of new data.

Table 34.26 SCI interrupt sources with non-FIFO selected (1 of 2)

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_ERI	Receive error *1	ORER, FER, PER, DFER, DPER	RIE	Not possible	Not possible
SCIn_RXI	Receive data full	RDRF	RIE	Possible	Possible
	Address match	DCMF	RIE	Possible	Possible
SCIn_AM	Address match	DCMF	—	Possible	Possible

Table 34.26 SCI interrupt sources with non-FIFO selected (2 of 2)

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_TXI	Transmit data empty	TDRE	TIE	Possible	Possible
SCIn_TEI	Transmit end	TEND	TEIE	Not possible	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

Table 34.27 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_ERI	Receive error*1	ORER, FER, PER, DFER, DPER	RIE	Not possible	Not possible
		DR (when FCR.DRES = 1)	RIE	Not possible	Not possible
SCIn_RXI	Receive data full	RDF	RIE	Possible	Possible
	Receive data ready	DR (when FCR.DRES = 0)	RIE	Possible	Possible
	Address match	DCMF	RIE	Possible	Possible
SCIn_AM	Address match	DCMF	—	Possible	Possible
SCIn_TXI	Transmit data empty	TDFE	TIE	Possible	Possible
SCIn_TEI	Transmit end	TEND	TEIE	Not possible	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

34.10.4 Interrupts in Smart Card Interface Mode

Table 34.28 lists interrupt sources in smart card interface mode. A transmit end interrupt (SCIn_TEI) request and an address match (SCIn_AM) request cannot be used in this mode.

Table 34.28 SCI Interrupt sources

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_ERI	Receive error or error signal detection	ORER, FER, ERS	RIE	Not possible	Not possible
SCIn_RXI	Receive data full	RDRF	RIE	Possible	Possible
SCIn_TXI	Transmit end	TEND	TIE	Possible	Possible

Data transmission or reception using the DTC or DMAC is also possible in smart card interface mode. In transmission, when the SSR_SMCI.TEND flag is set to 1, an SCIn_TXI interrupt request is generated. This SCIn_TXI interrupt request activates the DTC or DMAC, allowing transfer of transmit data if the SCIn_TXI request is previously specified as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission after an error occurrence. However, the SSR_SMCI.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR_SMCI.RIE bit to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings. For DTC or DMAC settings, see section 17, DMA Controller (DMAC) and section 18, Data Transfer Controller (DTC).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to the RDR register. This SCIn_RXI interrupt request activates the DTC or DMAC, allowing transfer of the receive data if the SCIn_RXI request is previously specified as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

34.10.5 Interrupts in Simple IIC Mode

Table 34.29 lists the interrupt sources in simple IIC mode. The STI interrupt is allocated to the transmit end interrupt (SCIn_TEI) request. The receive error interrupt (SCIn_ERI) and the address match (SCIn_AM) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple IIC mode.

When the SIMR2.IICINTM bit is 1:

- An SCIn_RXI request is generated on the falling edge of the SCLn signal for the 8th bit. If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.
- An SCIn_TXI request is generated on the falling edge of the SCLn signal for the 9th bit (acknowledge bit). If SCIn_TXI is previously set up as an activation source for the DTC or DMAC, the SCIn_TXI request activates the DTC or DMAC to handle transfer of the transmit data.

When the SIMR2.IICINTM bit is 0:

- An SCIn_RXI request (ACK detection) is generated if the input on the SDAn pin is low on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- An SCIn_TXI request (NACK detection) is generated if the input on the SDAn pin is high on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- If SCIn_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn_RXI request activates the DTC or DMAC to handle transfer of the received data.

If the DTC or DMAC is used for data transfer in reception or transmission, always set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 34.29 SCI interrupt sources

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_RXI	Reception, ACK detection	—	RIE	Possible	Possible
SCIn_TXI	Transmission, NACK detection	—	TIE	Possible	Possible
STIn	Completion of generation of a start, restart, or stop condition	IICSTIF	TEIE	Not possible	Not possible

Note 1. Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

34.11 Event Linking

By using interrupt request signals as event signals, the SCI can provide linked operation through the ELC for modules selected in advance.

Event signals can be output regardless of the values of the associated interrupt request enable bits.

(1) Error event output (receive error or error signal detected)

- Indicates abnormal termination because of a parity error during reception in asynchronous mode
- Indicates abnormal termination because of a framing error during reception in asynchronous mode
- Indicates abnormal termination because of an overrun error during reception
- Indicates detection of the error signal during transmission in smart card interface mode
- The SSR_FIFO.FER and PER flags are 0, and receive data less than the receive FIFO data trigger number is set in a reception FIFO buffer, and it indicates that 15 ETUs elapse when FIFO is selected and the FCR.DRES bit is 1.

(2) Receive data full event output

- Indicates that ACK is detected if the SIMR2.IICINTM bit is 0 in simple IIC mode
- Indicates that the 8th-bit SCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode
- When the SIMR2.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used.

(a) Non-FIFO selected

- Indicates that received data is set in the Receive Data Register (RDR or RDRHL).

(b) FIFO selected

- Using this event output is prohibited.

(3) Transmit data empty event output

- Indicates that the SCR/SCR_SMCI.TE bit is changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if the SIMR2.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th-bit SCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode.

(a) Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR or TDRHL) to the Transmit Shift Register (TSR).

(b) FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode
- When FIFO is selected, using this event output is prohibited.

(5) Address match event output

- Indicates a match of the comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode.

34.12 Address Mismatch Event Output (SCI0_DCUF)

SCI0_DCUF indicates the mismatch of comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode. This event can be used for Snooze end request only.

34.13 Noise Cancellation Function

Figure 34.72 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a 2-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the 2-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. When the same level is retained for 3 cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for 3 cycles or shorter is considered as noise, not as a receive signal.

When SEMR.ABCS = 0 and SEMR.ABCSE = 0, the cycle is 1/16 the period of 1 transfer bit.

When SEMR.ABCS = 1 and SEMR.ABCSE = 0, the cycle is 1/8 the period of 1 transfer bit.

When SEMR.ABCSE = 1, the cycle is 1/6 the period of 1 transfer bit.

In asynchronous mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The receive level of the RXDn is taken in the flip-flop circuit of the noise filter on the base clock of asynchronous mode.

In simple IIC mode, this function can be used for each input on SDAn and SCLn. The sampling clock for the noise cancellation function is selected in the SNFR.NFCS bit by dividing the baud rate generator source clock by 1, 2, 4, or 8.

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is conveyed as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles.

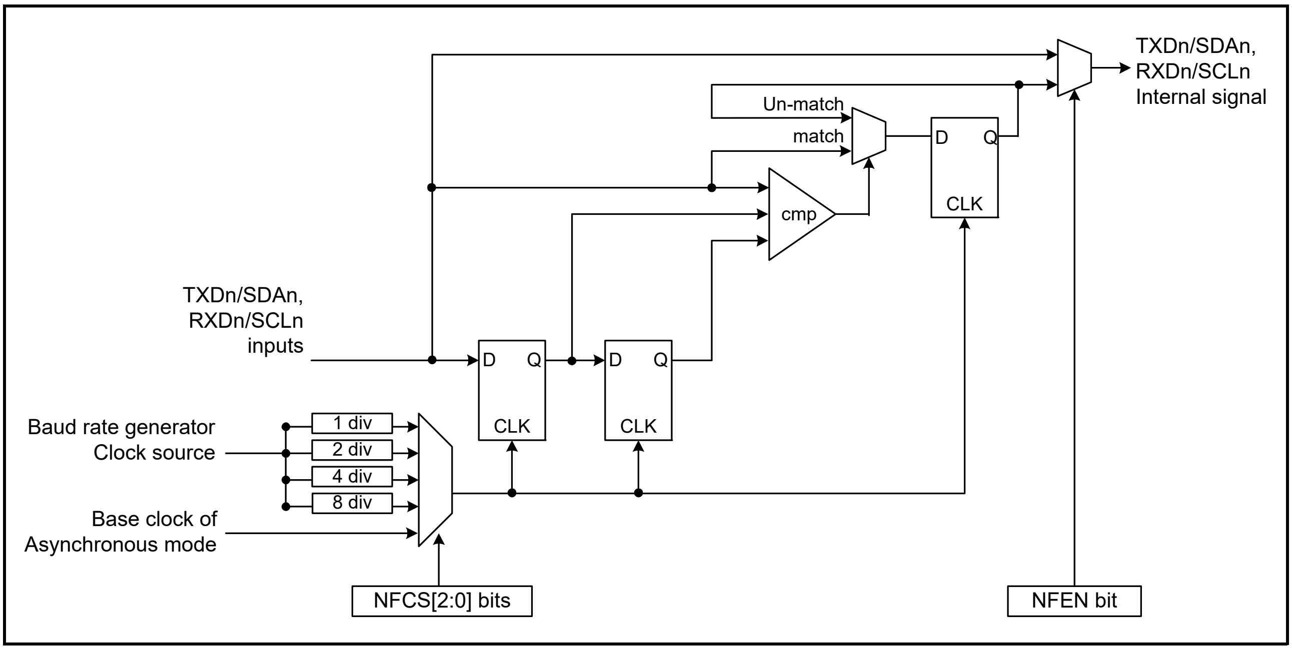


Figure 34.72 Digital noise filter circuit block diagram

34.14 Usage Notes

34.14.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SCI operation. The SCI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

34.14.2 SCI Operation during Low Power State

(1) Transmission

When setting the module to the stopped state or in transitions to Software Standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR/SCR_SMCI to 0) after switching the TXDn pin to the general I/O port pin function. When setting I/O port as an SCI connection, the SPTR register can control the state of the TXDn pin. Setting the TE bit to 0 initializes the TSR register and the TEND bit in the SSR/SSR_SMCI is initialized to 1 with non-FIFO selected. The value is saved with FIFO selected. Depending on the port settings and SPTR register settings, output pins might output the level before a transition to the low power state is made after release from the module-stop state or Software Standby mode. When transitions to these states are made during transmission, the transmitted data becomes indeterminate.

To transmit data in the same transmission mode after cancellation of the low power state:

1. Set the TE bit to 1.
2. Read SSR/SSR_FIFO/SSR_SMCI.
3. Write data to TDR sequentially to start data transmission.

To transmit data with a different transmission mode, initialize the SCI first.

Figure 34.73 shows an example flow of transition to Software Standby mode during transmission. Figure 34.74 and Figure 34.75 show the port pin states during transition to Software Standby mode.

Before specifying the module-stop state or making a transition to Software Standby mode from the transmission mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit to 1. The SCIn_TXI interrupt flag is set to 1 and transmission starts using the DTC.

(2) Reception

(a) When address match function is not used as wakeup condition

Before specifying the module-stop state or making a transition to Software Standby mode, stop the receive operations (RE = 0 in SCR/SCR_SMCI). If transition is made during data reception, the received data is invalid.

Figure 34.76 shows an example flow of transition to Software Standby mode during reception.

(b) When address match function is used as wakeup condition

Before specifying the module-stop state or making a transition to Software Standby mode:

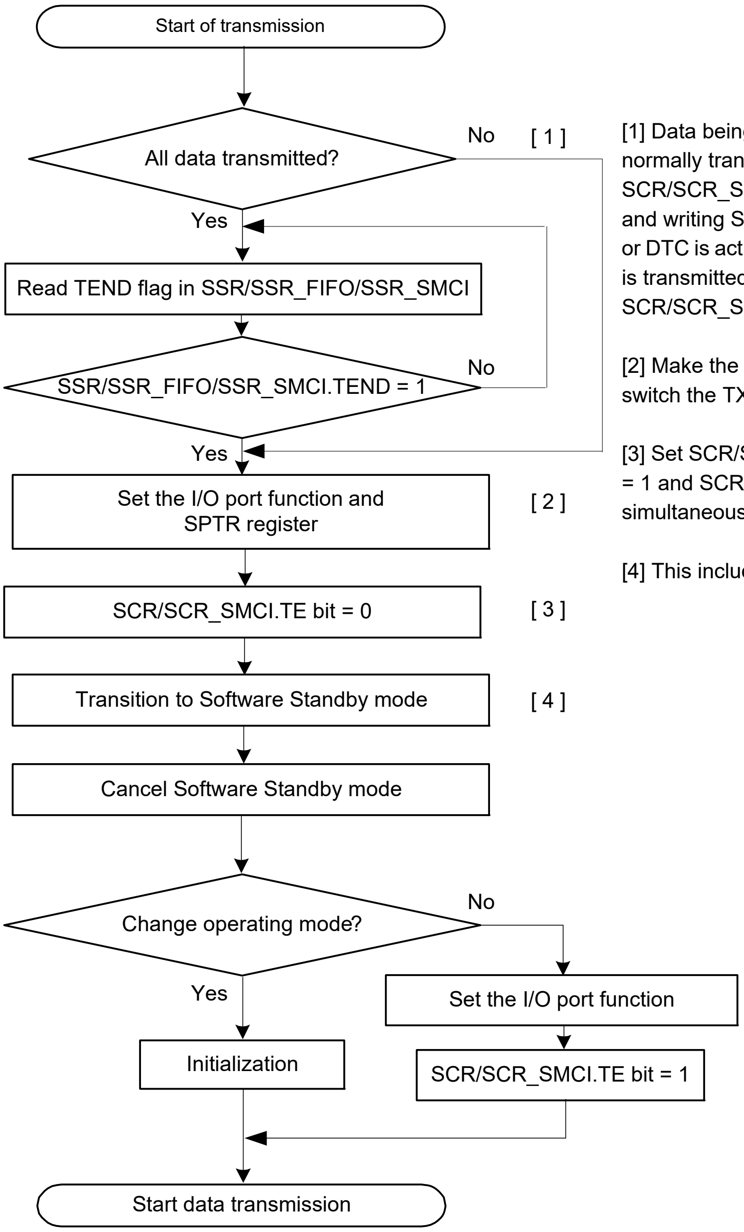
1. Set the operations after cancellation of the low power state.
2. Set CDR.CMPD and DCCR.DCME to 1.
3. Set the receive operations (RE = 1 in SCR/SCR_SMCI).
4. Set the module-stop state or Software Standby mode.

When SCI transfers to low power mode, if the receive data pin (RXD) is at the low level, set SEMR.RXDESEL = 0. When setting SEMR.RXDESEL = 1, there is a possibility that a start bit (falling edge of RXDn pin) cannot be detected on release of the low power mode.

Figure 34.77 shows an example flow of transition to Software Standby mode during reception with address match.

(c) When using SCI0 in Snooze mode

When using SCI0 in Snooze mode, some restrictions apply, including maximum bit rates. For details, see section 11, Low Power Modes.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the TE bit in SCR/SCR_SMCI to 1, reading SSR/SSR_FIFO/SSR_SMCI, and writing Software Standby mode. However, if the DMAC or DTC is activated, the data remaining in the DMAC or DTC is transmitted when both the TE and TIE bits in SCR/SCR_SMCI are set to 1.

[2] Make the I/O port function and SPTR register settings to switch the TXDn pin to operate as a general I/O port.

[3] Set SCR/SCR_SMCI.TE bit to 0. If SCR/SCR_SMCI.TIE = 1 and SCR/SCR_SMCI.TEIE = 1, these are set to 0 simultaneously with the SCR.TE bit.

[4] This includes the setting for the module-stop state.

Figure 34.73 Example flow of transition to Software Standby mode during transmission

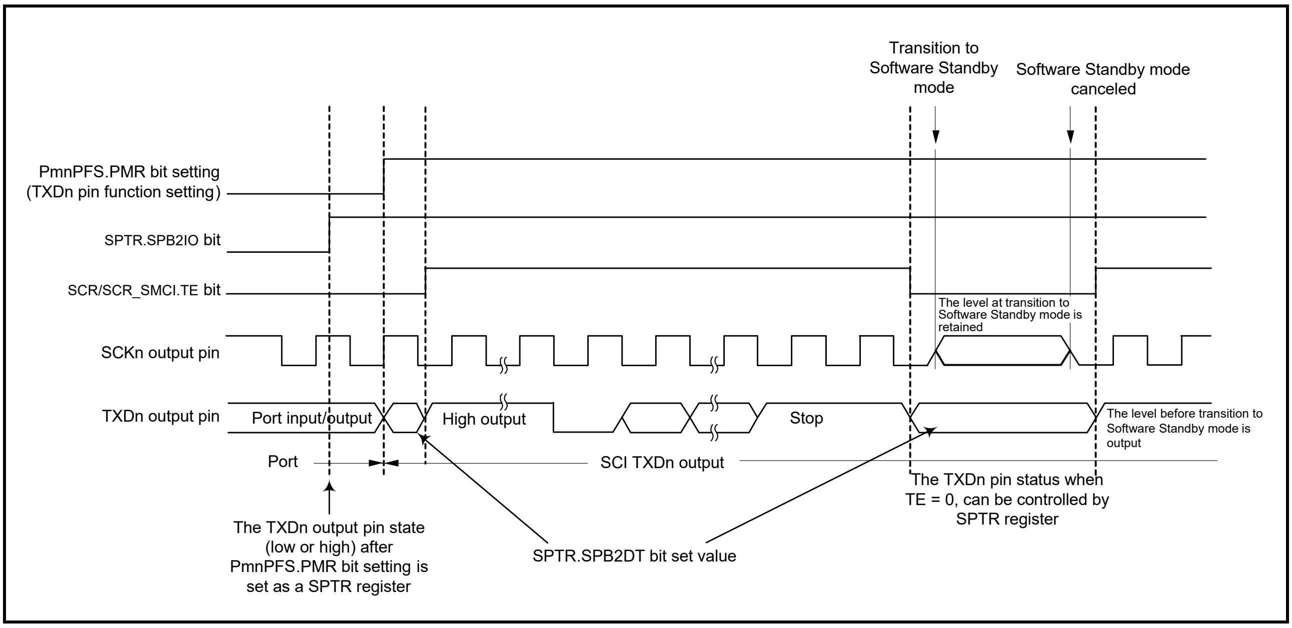


Figure 34.74 Port pin states during transition to Software Standby mode with internal clock and asynchronous transmission

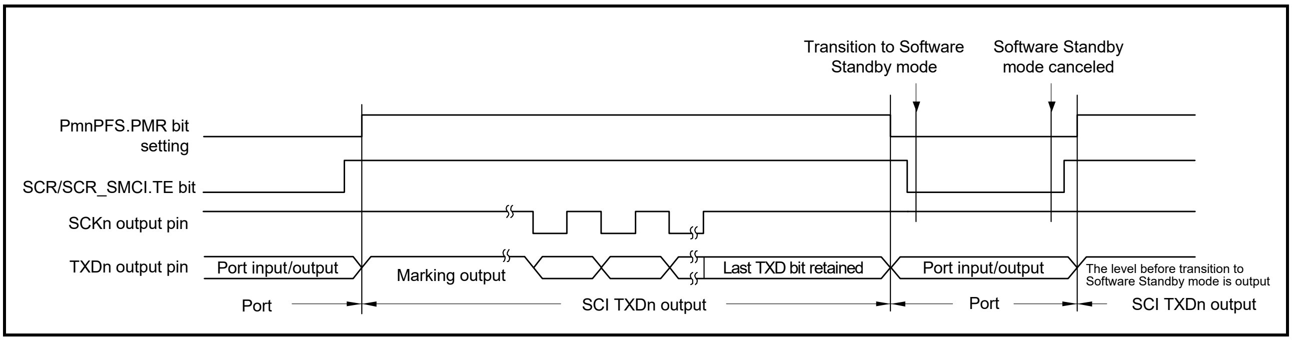
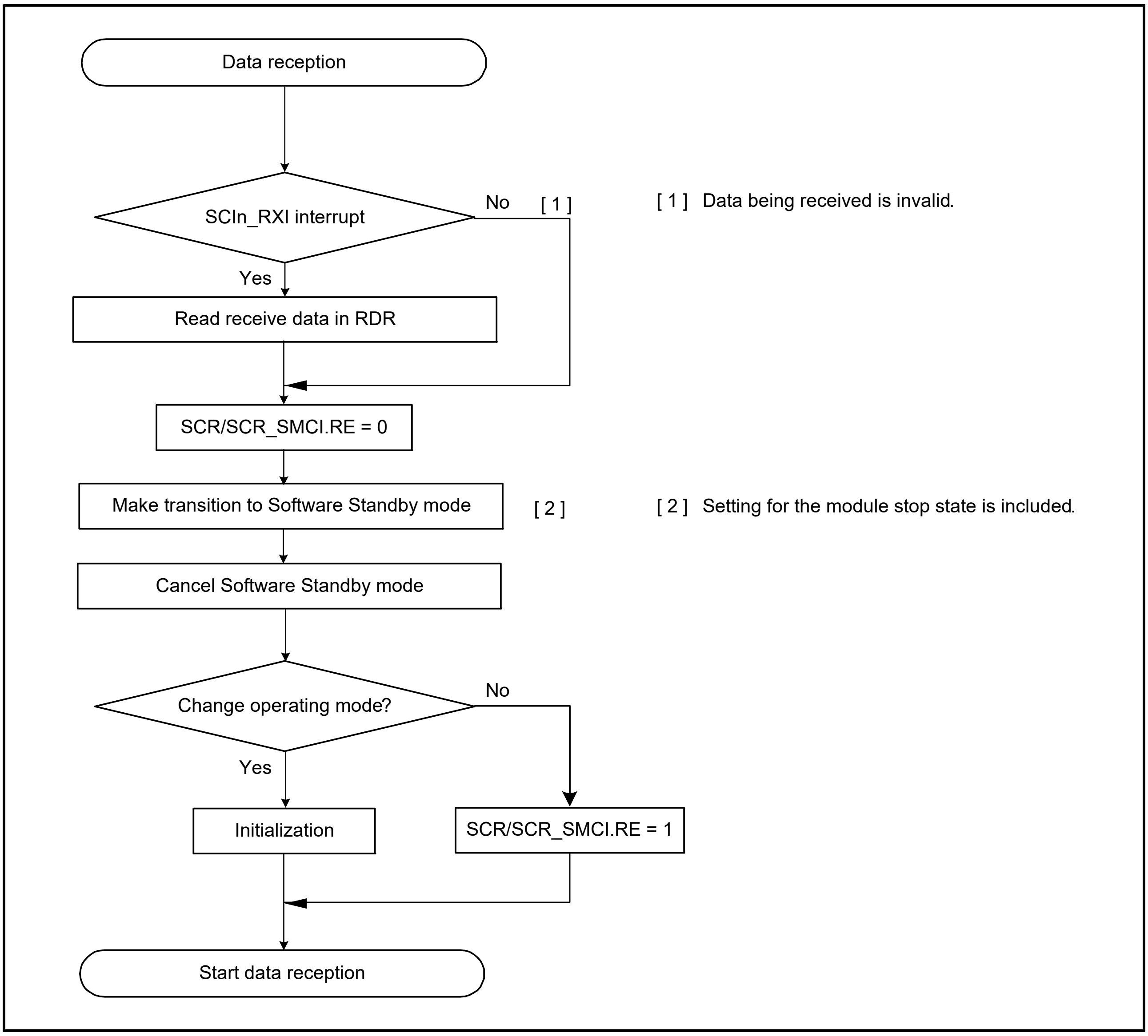


Figure 34.75 Port pin states during transition to Software Standby mode with internal clock and clock synchronous transmission



[1] Data being received is invalid.

[2] Setting for the module stop state is included.

Figure 34.76 Example flow of transition to Software Standby mode during reception

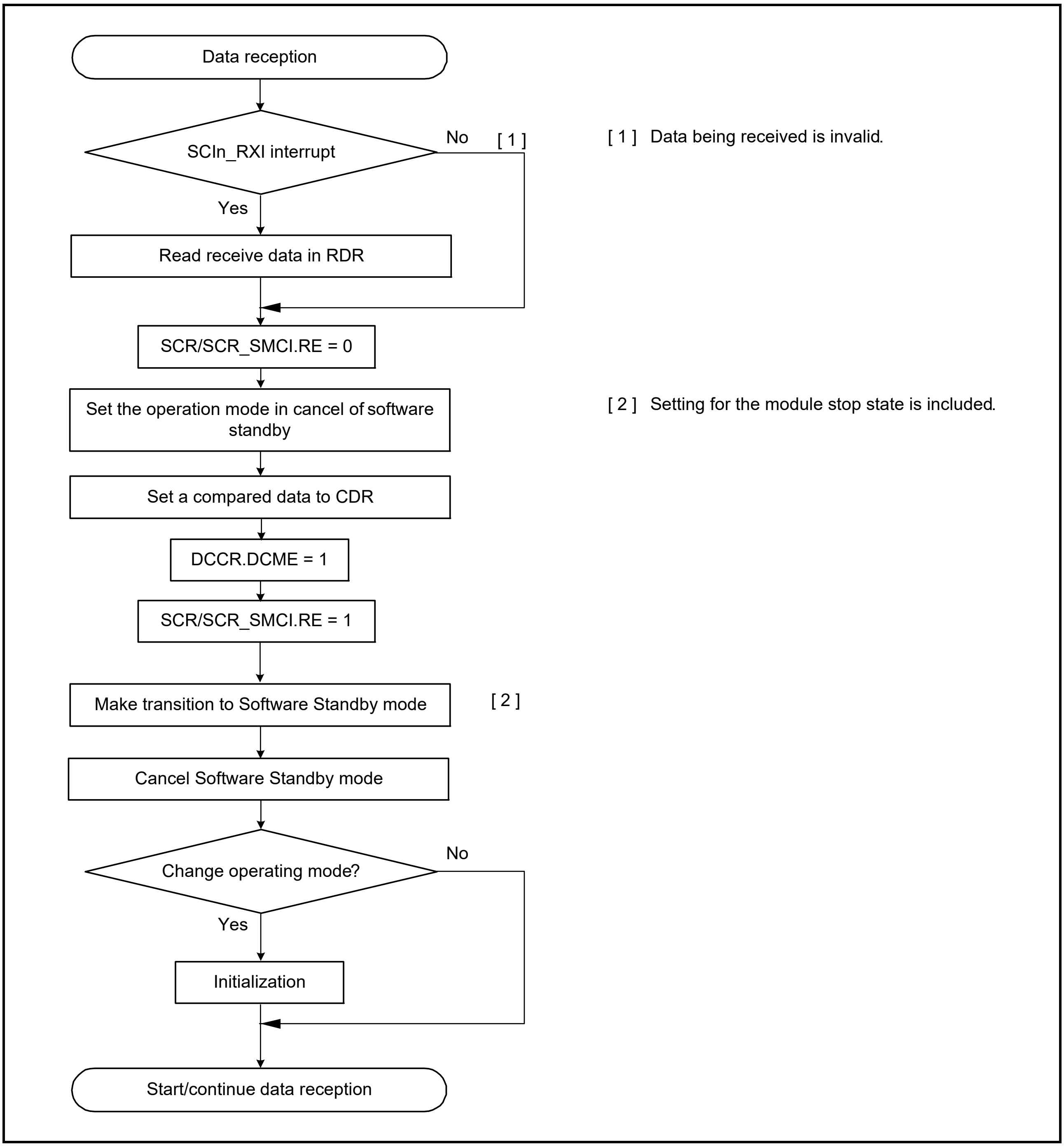


Figure 34.77 Example flow of transition to Software Standby mode during reception with address match

34.14.3 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and the SSR.FER flag is set to 1 to indicate a framing error, and the SSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, even if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0, the SSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the

RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for 1 frame, reception stops. When a framing error is detected, a break can be detected by reading the SPTR.RXDMON bit value. After the RXD signal is in the mark state and the break is finished, data reception to the FRDRHL register resumes.

34.14.4 Mark State and Production of Breaks

When the SCR/SCR_SMCI.TE bit is 0, disabling serial transmission, the state of the TXDn pin can be set using the SPTR.SPB2IO and SPTR.SPB2DT bits. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the SCR/SCR_SMCI.TE bit to 1, enabling serial transmission, set the SPB2IO and SPB2DT bits to put the communication line in the mark state (the state of 1), and change the TxDn pin using I/O port function. To output a break on data transmission, after setting the TXDn pin to output 0 by setting the SPB2IO and SPB2DT bits, change the TXDn pin using the I/O port function and set the SCR/SCR_SMCI.TE bit to 0. When the SCR/SCR_SMCI.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

34.14.5 Receive Error Flags and Transmit Operation in Clock Synchronous and Simple SPI Modes

Transmission cannot start when a receive error flag (ORER) in SSR/SSR_FIFO is set to 1, even when data is written to TDR or FTDRL*2. Always set the receive error flags to 0 before starting transmission.

Note 1. The receive error flags cannot be set to 0 when serial reception is disabled by setting the RE bit in SCR/SCR_SMCI to 0.

Note 2. Do not use the FTDRH register in simple SPI mode.

34.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous and Simple SPI Modes

When the external clock source is used as a synchronization clock, the following restrictions apply.

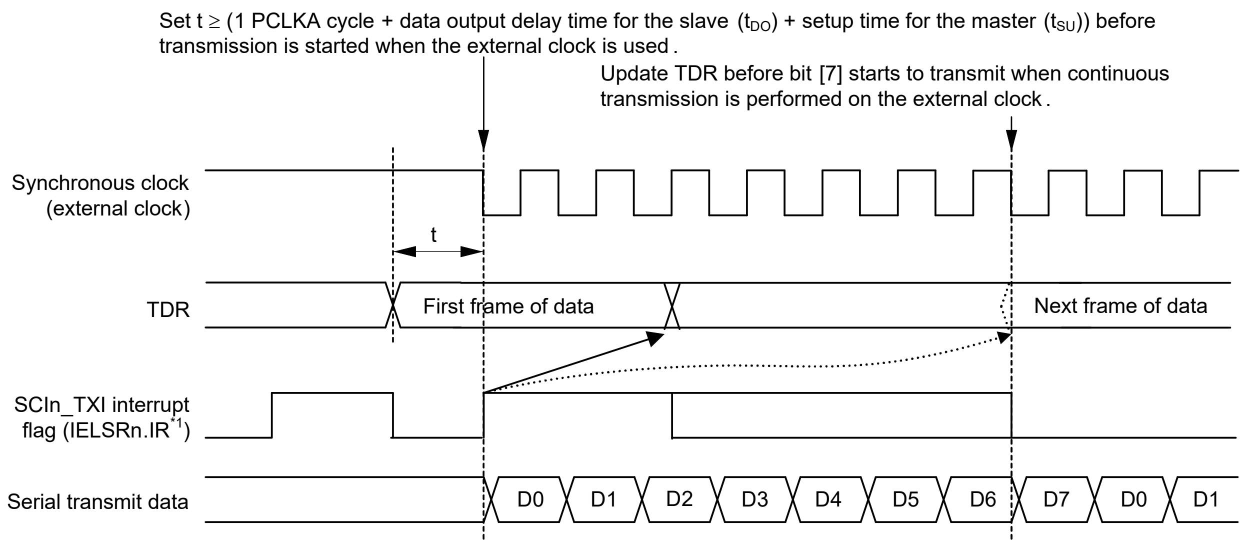
(1) Start of transmission

Wait at least the following time from writing transmit data to TDR to the start of the external clock input:
1 PCLKA cycle + data output delay time for the slave (t_{DO}) + setup time for the master (t_{SU}). See Figure 34.78.

(2) Continuous transmission

Write the next transmit data to TDR or TDRHL before the falling edge of the transmit clock for bit [7] (see Figure 34.78).

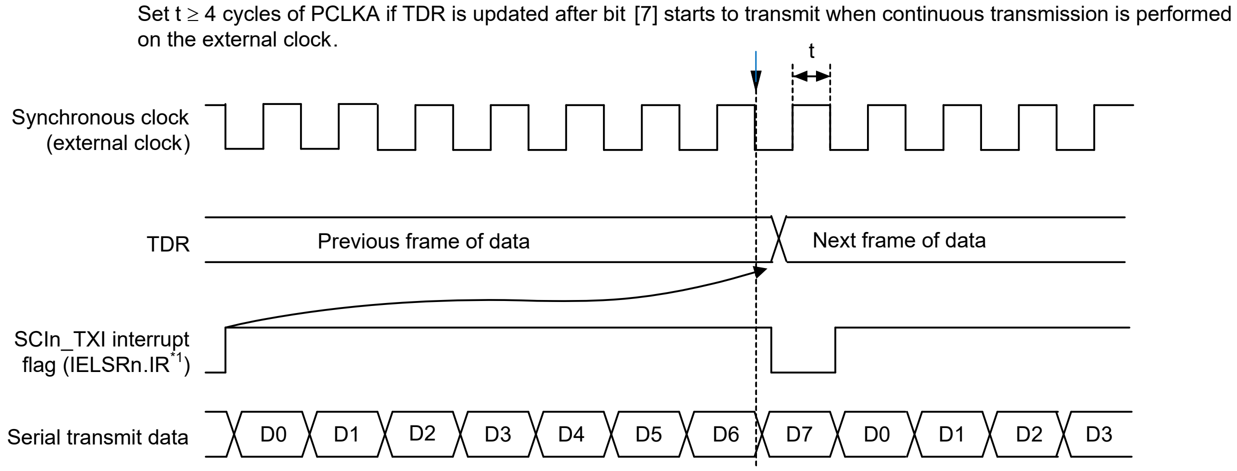
When updating TDR after bit [7] has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit [7]) to 4 PCLKA cycles or longer (see Figure 34.78).



Set t ≥ (1 PCLKA cycle + data output delay time for the slave (t_{DO}) + setup time for the master (t_{SU})) before transmission is started when the external clock is used.

Update TDR before bit [7] starts to transmit when continuous transmission is performed on the external clock.

(1) Start of transmission and (2) Continuous transmission (a)



Set t ≥ 4 cycles of PCLKA if TDR is updated after bit [7] starts to transmit when continuous transmission is performed on the external clock.

(2) Continuous transmission (b)

Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 34.78 Restrictions on use of external clock in clock synchronous transmission

34.14.7 Restrictions on Using DMAC or DTC

During transmission or reception operations using the DMAC or DTC, do not set transfer data for the DMAC/DTC.

(1) Writing data to TDR (FTDRHL)

(a) Non-FIFO selected

Data can be written to TDR and TDRHL. However, if new data is written to TDR or TDRHL when transmit data remains in TDR or TDRHL, the previous data in TDR and TDRHL is lost because it was not transferred to TSR yet. When using DTC or DMAC, always write transmit data to TDR or TDRHL in the SCIn_TXI interrupt request handling routine.

(b) FIFO selected

It is possible to write data to the FTDRH and FTDRL registers when SCR.TE is 1. Confirm the amount of writable data using the FDR.T[4:0] bits.

(2) Reading data from RDR (FRDRHL)

When using the DMAC or DTC to read RDR and RDRHL, always set the receive data full interrupt (SCIn_RXI) as the activation source of the relevant SCI.

34.14.8 Notes on Starting Transfer

When transfer starts while the Interrupt Status flag (IELSRn.IR) in the ICU is 1, follow the procedure in this section to clear interrupt requests before permitting operations (by setting the SCR/SCR_SMCI.TE or SCR/SCR_SMCI.RE bit to 1). For details on the Interrupt Status flag, see section 14, Interrupt Controller Unit (ICU).

- Confirm that transfer has stopped (the SCR/SCR_SMCI.TE or SCR/SCR_SMCI.RE bit is 0)
- Set the associated interrupt enable bit (SCR/SCR_SMCI.TIE or SCR/SCR_SMCI.RIE) to 0
- Read the associated interrupt enable bit (SCR/SCR_SMCI.TIE or SCR/SCR_SMCI.RIE bit) to check that it actually becomes 0
- Set the Interrupt Status flag, IELSRn.IR, in the ICU to 0.

34.14.9 External Clock Input in Clock Synchronous and Simple SPI Modes

In clock synchronous mode and simple SPI mode, the external clock (SCKn) must be input as follows:

High-pulse period, low-pulse period = 2 PCLKA cycles or more, period = 6 PCLKA cycles or more.

34.14.10 Limitations on Simple SPI Mode

(1) Master mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set in the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.

  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit changes from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not required because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- For the clock delay setting (SPMR.CKPH bit is 1), the receive data full interrupt (SCIn_RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 34.79. If the TE and RE bits in the SCR register become 0 before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Additionally, an SCIn_RXI interrupt might lead to the input signal on the SSn pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, the SCKn pin output goes to high-impedance while the input on the SSn pin is at the low level if a mode fault error occurs while a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

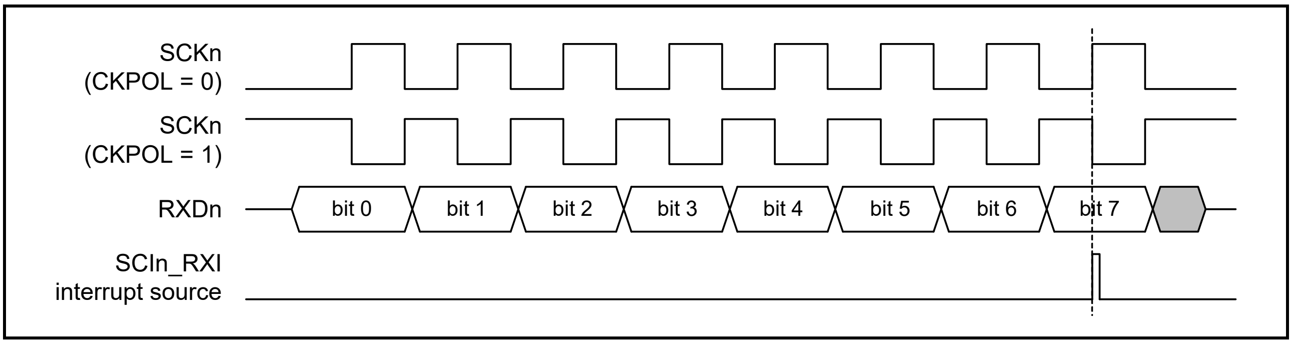


Figure 34.79 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave mode

- Wait at least the following time from writing transmit data in the TDR register to the start of the external clock input.
  1 PCLKA cycle + data output delay for the slave (tDO) + setup time for the master (tSU)

  Also wait at least 5 PCLKA cycles from the input of the low level on the SSn pin to the start of the external clock input.
- Provide an external clock signal to the master the same as the data length for transfer
- Control the input on the SSn pin before the start and after the end of data transfer
- When the input level on the SSn pin is to be changed from low to high while a character is being transferred, set the TE and RE bits in the SCR register to 0 and, after restoring the settings, restart transfer of the first byte.

35. IrDA Interface

35.1 Overview

The IrDA interface sends and receives IrDA data communication waveforms in cooperation with the SCI1 based on the IrDA (Infrared Data Association) standard 1.0.

Enabling the IrDA function in the IRE bit in the IRCR register allows encoding and decoding of the TXD1 and RXD1 signals of the SCI1 to the waveforms conforming to the IrDA standard 1.0 (IRTXD1 and IRRXD1 pins). Connecting the waveforms to an infrared transmitter/receiver implements infrared data communication conforming to the IrDA standard 1.0 system.

With the IrDA standard 1.0 system, data transfer can be started at 9,600 bps and the transfer rate can be changed whenever necessary. Because the IrDA interface cannot change the transfer rate automatically, the transfer rate must be changed through the software.

Figure 35.1 shows the cooperation between the IrDA interface and SCI1.

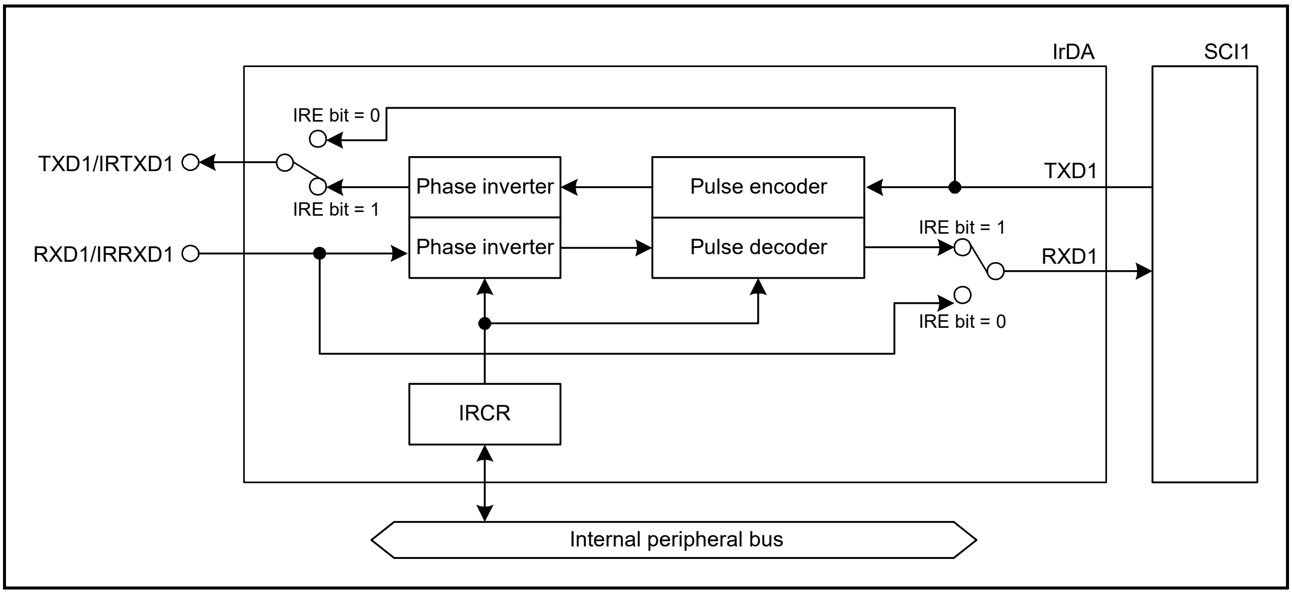


Figure 35.1 Cooperation between the IrDA interface and SCI1

Table 35.1 IrDA interface I/O pins

Pin name	I/O	Function
IRTXD1	Output	Data to be transmitted
IRRXD1	Input	Received data

35.2 Register Descriptions

35.2.1 IrDA Control Register (IRCR)

Address(es): IRDA.IRCR 4007 0F00h

	b7	b6	b5	b4	b3	b2	b1	b0
	IRE	—	—	—	IRTXINV	IRRXINV	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	IRRXINV	IRRXD Polarity Switching	0: Use IRRXD input as received data as-is 1: Use IRRXD input as received data after the polarity is inverted.	R/W
b3	IRTXINV	IRTXD Polarity Switching	0: Output data to be transmitted to IRTXD as-is 1: Output data to be transmitted IRTXD after the polarity is inverted.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	IRE	IrDA Enable	0: Use serial input/output pins for normal serial communication 1: Use serial input/output pins for IrDA data communication.	R/W

Note: The IRCR register values are retained in Sleep, Software Standby, and Deep Software Standby modes.

IRRXINV bit (IRRXD Polarity Switching)

The IRRXINV bit inverts the logic level of the IRRXD input. When inverted, the high-level pulse width is applied to the low-level pulse width.

IRTXINV bit (IRTXD Polarity Switching)

The IRTXINV bit inverts the logic level of the IRTXD output. When inverted, the high-level pulse width is applied to the low-level pulse width.

IRE bit (IrDA Enable)

The IRE bit configures the I/O pins for normal communication mode or IrDA data communication mode.

35.3 Operation

35.3.1 IrDA Interface Setup Procedure

To set up IrDA interface operation:

1. Set the associated pins to IRTXD1 and IRRXD1 in the Pin Function Control Register (PmnPFS.PSEL = 00101b) of the I/O ports function.
2. Specify the peripheral function in the Pin Function Control Register (PmnPFS.PMR = 1) of the I/O ports function.
3. Specify the IrDA function in the IRCR register.
4. Set the SCI1-related registers of the Serial Communications Interface (SCI).

35.3.2 Transmission

During transmission, the signals output from the SCI1 (UART frames) are converted to the IR frame data through the IrDA interface (see Figure 35.2). When the IRCR.IRTXINV bit is 0 and serial data is 0, high-level pulses with 3/16 the width of the bit rate (1-bit width period) are output (initial setting). The standard prescribes that the minimum high-level pulse width must be 1.41 µs and the maximum high-level pulse width must be (3/16 + 2.5%) × bit rate or (3/16 × bit rate) + 1.08 µs. When the serial data is 1, no pulses are output.

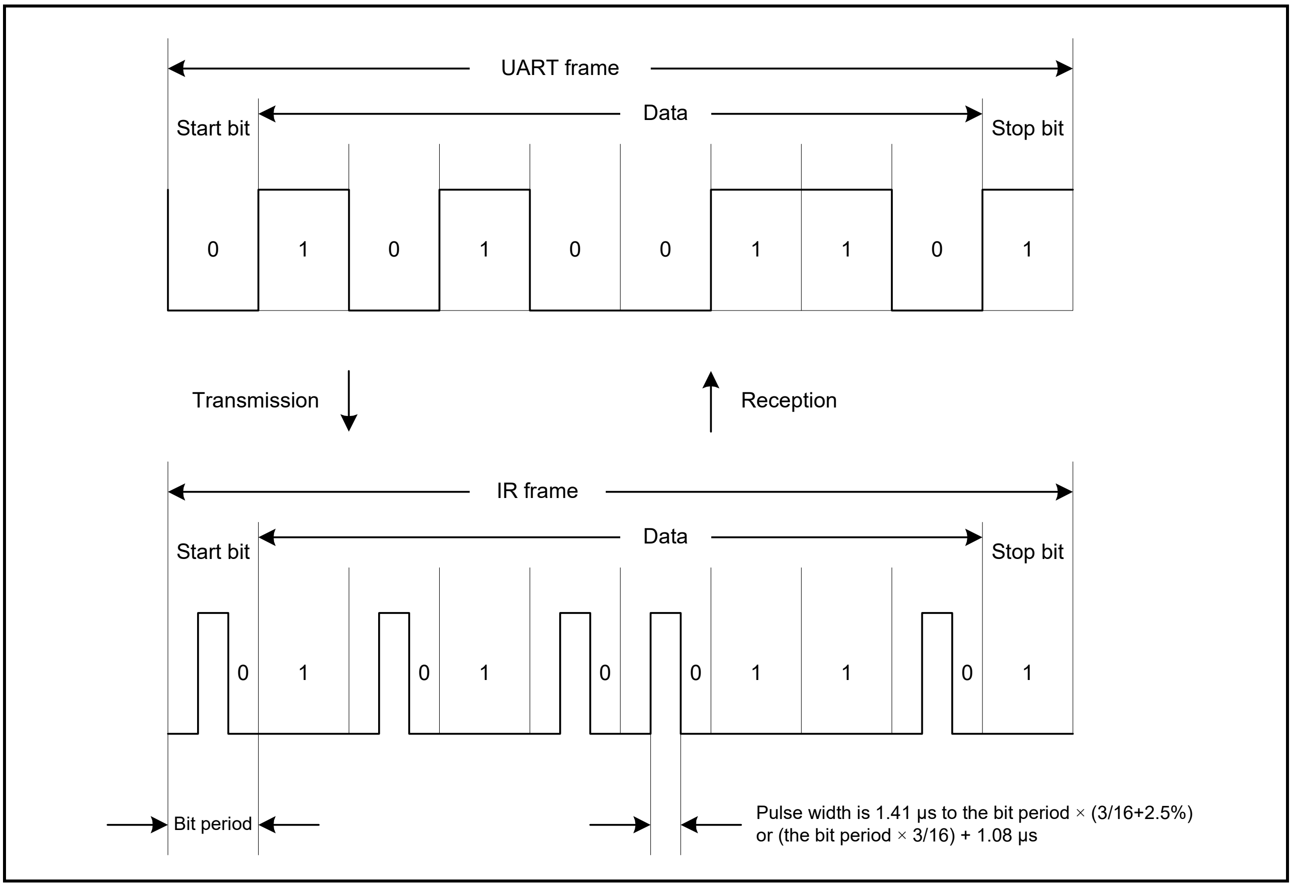


Figure 35.2 IrDA transmission and reception

35.3.3 Reception

During reception, the IR frame data is converted to the UART frame data through the IrDA interface and is input to the SCI1. Low-level data is input to SCI1 when the IRCR.IRRXINV bit is 0 and a high-level pulse is detected. High-level data is input to SCI1 when no pulse is detected for a 1-bit period.

35.4 Usage Notes

35.4.1 Settings for the Module-Stop Function

IrDA operation can be disabled or enabled using the Module Stop Control Register. The IrDA is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

35.4.2 Asynchronous Reference Clock for SCI1

The IrDA receives a clock with a frequency 16 times the bit rate from SCI1 and operates in conjunction with SCI1. When using the IrDA, set the SCI1.SEMR.ABCS bit to 0.

36. I^{2}C Bus Interface (IIC)

36.1 Overview

The 3-channel I^{2}C Bus Interface (IIC) module conforms with and provides a subset of the NXP I^{2}C bus (inter-integrated circuit bus) interface functions. Table 36.1 lists the IIC specifications, Figure 36.1 shows a block diagram, and Figure 36.2 shows an example of I/O pin connections to external circuits, with an I^{2}C bus configuration. Table 36.2 lists the I/O pins.

Table 36.1 IIC specifications (1 of 2)

Parameter	Specifications
Communications format	• I^{2}C-bus format or SMBus format • Master or slave mode selectable • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode Plus supported, up to 1 Mbps
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated • Start conditions (including restart conditions) and stop conditions are detectable
Slave address	• Configurable for up to three different slave addresses • 7- and 10-bit address formats supported, including simultaneous use • General call addresses, device ID addresses, and SMBus host addresses detectable
Acknowledgment	• For transmission, automatic loading of the acknowledge bit Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. • For reception, automatical transmission of the acknowledge bit If a wait between the eighth and ninth clock cycles is selected, the software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: • Waiting between the eighth and ninth clock cycles • Waiting between the ninth clock cycle and the first clock cycle of the next transfer
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed
Arbitration	• For multi-master operation: - SCL clock synchronization is possible when conflict occurs with the SCL signal from another master - When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line - In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line • Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions • Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match • Loss of arbitration because non-matching of internal and line levels for data is detectable in slave transmission
Timeout function	Internal detection of long-interval stops of the SCL clock
Noise cancellation	• Digital noise filters for both the SCL and SDA signals • Programmable window for noise cancellation by the filters
Interrupt sources	• Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) • Receive data full, including matching with a slave address • Transmit data empty, including matching with a slave address • Transmit end
Module-stop function	Module-stop state can be set to reduce power consumption
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive

Table 36.1 IIC specifications (2 of 2)

Parameter	Specifications
Event link function (output)	• Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) • Receive data full, including matching with a slave address • Transmit data empty, including matching with a slave address • Transmit end
Wakeup function*1	CPU can return from Software Standby mode using a wakeup event

Note 1. Only supported for IIC0. IIC1 and IIC2 are not supported.

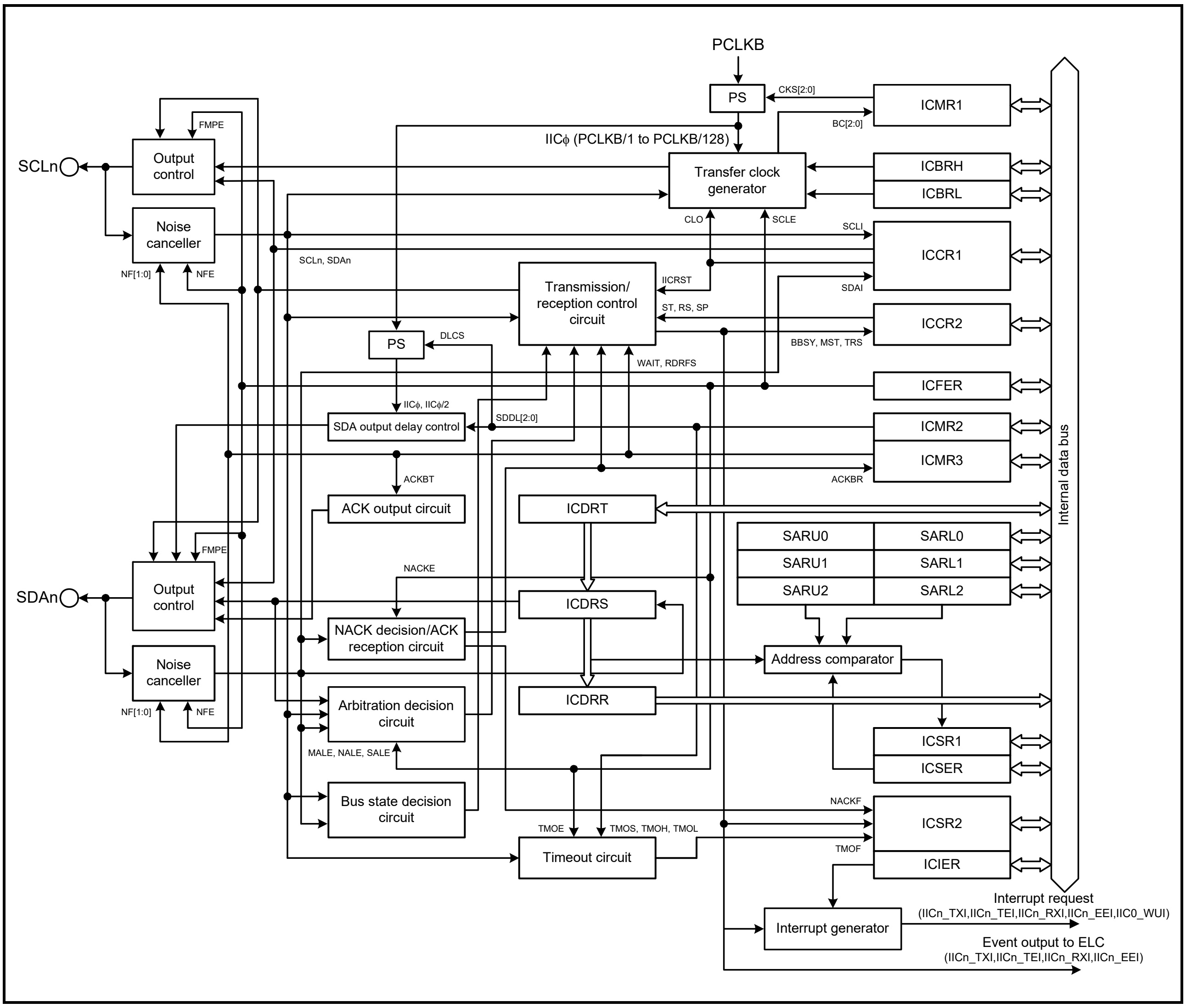


Figure 36.1 IIC block diagram

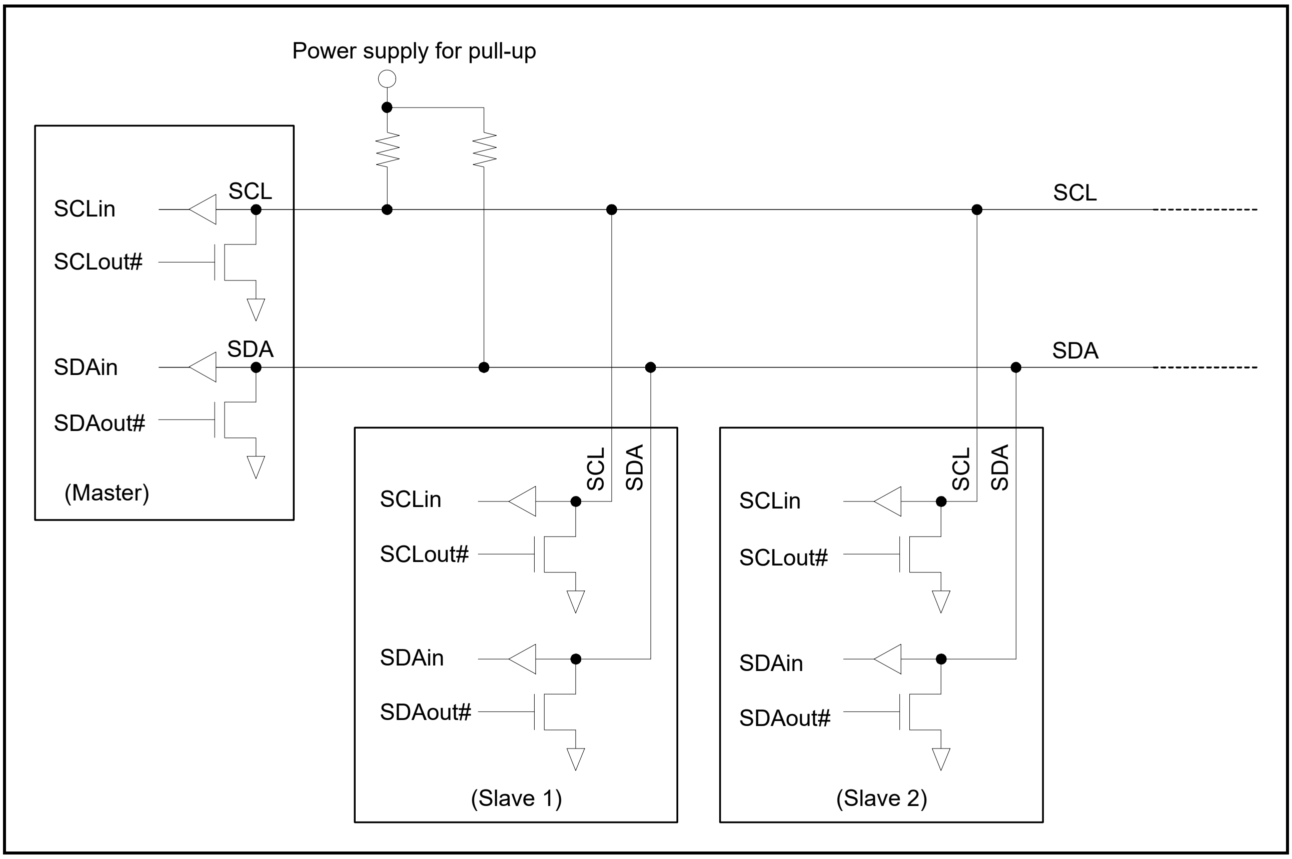


Figure 36.2 I/O pin connection to an external circuit (I^{2}C bus configuration example)

The input level of the signals for IIC is CMOS when I^{2}C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

Table 36.2 IIC I/O pins

Channel	Pin name	I/O	Function
IIC0	SCL0	I/O	IIC0 serial clock input/output pin
	SDA0	I/O	IIC0 serial data input/output pin
IIC1	SCL1	I/O	IIC1 serial clock input/output pin
	SDA1	I/O	IIC1 serial data input/output pin
IIC2	SCL2	I/O	IIC2 serial clock input/output pin
	SDA2	I/O	IIC2 serial data input/output pin

36.2 Register Descriptions

36.2.1 I^{2}C Bus Control Register 1 (ICCR1)

Address(es): IIC0.ICCR1 4005 3000h, IIC1.ICCR1 4005 3100h, IIC2.ICCR1 4005 3200h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDAn line is low 1: SDAn line is high.	R
b1	SCLI	SCL Line Monitor	0: SCLn line is low 1: SCLn line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: IIC drove SDAn pin low 1: IIC released SDAn pin. • Write: 0: Drive SDAn pin low through IIC 1: Release SDAn pin through IIC.	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: IIC drove SCLn pin low 1: IIC released SCLn pin. • Write: 0: Drive SCLn pin low through IIC 1: Release SCLn pin through IIC. Use an external pull-up resistor to drive the signal high.	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits. This bit is read as 1.	R/W
b5	CLO	Extra SCL Clock Cycle Output	0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle. This bit clears automatically after one clock cycle is output.	R/W
b6	IICRST	IIC-Bus Interface Internal Reset	0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset. This setting clears the bit counter and the SCLn/SDAn output latch.	R/W
b7	ICE	IIC-Bus Interface Enable	0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state). Combined with the IICRST bit to select either IIC or internal reset.	R/W

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO bit directly controls the SDAn and SCLn signals output from the IIC. When writing to these bits, also write 0 to the SOWP bit. Setting these bits results in input to the IIC by the input buffer. When slave mode is selected, a start condition might be detected and the bus might be released, depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, transmission, or reception. Operation after rewriting under these conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows output of an extra SCL clock cycle for debugging or error processing. Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error. For details on this function, see section 36.12.2, Extra SCL Clock Cycle Output Function.

IICRST bit (IIC-Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset.

Whether an IIC reset or internal reset is initiated is determined by the settings of this bit in combination with the ICE bit. Table 36.3 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits, and internal states of the IIC.

The internal reset initializes the following in addition to the internal states of the IIC:

- Bit counter (ICMR1.BC[2:0] bits)
- I^{2}C Bus Shift Register (ICDRS)
- I^{2}C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits)
- I^{2}C Bus Control Register 2 (except ICCR2.BBSY bit).

For the reset conditions for each register, see section 36.15, State of Registers when Issuing each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC when the bus or IIC hangs up because of a communication error. If the IIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCLn pin and SDAn pin at high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up that occurs during communication with the master device in slave mode, the slave and master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is necessary because the IIC hangs up with the SCLn line in a low level output state in slave mode, initiate an internal reset, and then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 36.3 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.ICE and ICCR1.IICRST bits, and internal states of the IIC
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, the ICSR1, ICSR2, ICDRS registers, and SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits), I2C Bus Control Register 2 (except ICCR2.BBSY bit) and the internal states of the IIC

ICE bit (IIC-Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 36.3 for the reset descriptions.

Set the ICE bit to 1 when using the IIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1. Set the ICE bit to 0 when the IIC is not used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

36.2.2 I^{2}C Bus Control Register 2 (ICCR2)

Address(es): IIC0.ICCR2 4005 3001h, IIC1.ICCR2 4005 3101h, IIC2.ICCR2 4005 3201h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Issuance Request	0: Do not issue a start condition request 1: Issue a start condition request.	R/W
b2	RS	Restart Condition Issuance Request	0: Do not issue a restart condition request 1: Issue a restart condition request.	R/W
b3	SP	Stop Condition Issuance Request	0: Do not issue a stop condition request 1: Issue a stop condition request.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode.	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode.	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: I^{2}C bus released (bus free state) 1: I^{2}C bus occupied (bus busy state).	R

Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition. When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on this function, see section 36.11, Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Only set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode. When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 36.11, Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.

Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode. When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 36.11, Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition is issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).

Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode. The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of this bit and the MST bit indicates the operating mode of the IIC.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit appended to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When the R/W# bit appended to the slave address is set to 1 in master mode
- In slave mode, on a match between the received address and the address enabled in ICSER when the value of the

received R/W# bit is 0, including when the received address is the general call address

- In slave mode, when a restart condition is detected (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is 0 and is in master mode when the bit is 1. The combination of this bit and the TRS bit indicates the operating mode of the IIC.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued or a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I^{2}C bus is occupied (bus busy state) or released (bus free state). The flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued. The flag then is set to 0 if a start condition is not detected for the bus free time (ICBRL setting), assuming that a stop condition was issued.

[Setting condition]

- When a start condition is detected.

[Clearing conditions]

- When a start condition is not detected for the bus free time (ICBRL setting) after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

36.2.3 I^{2}C Bus Mode Register 1 (ICMR1)

Address(es): IIC0.ICMR1 4005 3002h, IIC1.ICMR1 4005 3102h, IIC2.ICMR1 4005 3202h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits.	R/W*1
b3	BCWP	BC Write Protect	0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits. This bit is read as 1.	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock source (IICϕ) for the IIC. b6 b4 0 0 0: PCLKB clock 0 0 1: PCLKB/2 clock 0 1 0: PCLKB/4 clock 0 1 1: PCLKB/8 clock 1 0 0: PCLKB/16 clock 1 0 1: PCLKB/32 clock 1 1 0: PCLKB/64 clock 1 1 1: PCLKB/128 clock.	R/W
b7	MTWP	MST/TRS Write Protect	0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2.	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter indicating the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although BC[2:0] are read/write bits, it is not normally necessary to access these bits.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCLn line is at a low level. The value in the BC[2:0] bits returns to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

36.2.4 I^{2}C Bus Mode Register 2 (ICMR2)

Address(es): IIC0.ICMR2 4005 3003h, IIC1.ICMR2 4005 3103h, IIC2.ICMR2 4005 3203h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Select long mode 1: Select short mode.	R/W

Bit	Symbol	Bit name	Description	R/W
b1	TMOL	Timeout L Count Control	0: Disable count while SCLn line is low 1: Enable count while SCLn line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Disable count while SCLn line is high 1: Enable count while SCLn line is high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles. • When ICMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 1 1: 13 or 14 IICϕ cycles.	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: Select internal reference clock (IICϕ) as clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICϕ/2) as clock source for SDA output delay counter.[*1]	R/W

Note 1. The setting DLCS = 1 (IICϕ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICϕ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1). When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14-bit counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source. For details on this function, see section 36.12.1, Timeout Function.

TMOL bit (Timeout L Count Control)

The TMOL bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 1).

TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held high and the timeout function is enabled (ICFER.TMOE = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. This setting can be used for all types of SDA output, including transmission of the acknowledge bit.

Set the SDA output delay to meet the I^{2}C bus standard for the data enable time/acknowledge enable time,[*1] or the SMBus standard, within [data hold time (300 ns or more + the SCL-clock low-level period) - the data setup time (250 ns)]. If a value outside the standard is set, communication between devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see section 36.5, SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time

3,450 ns for up to 100 kbps: Standard-mode (Sm)
900 ns for up to 400 kbps: Fast-mode (Fm)
450 ns for up to 1 Mbps: Fast-mode Plus (Fm+)

36.2.5 I^{2}C Bus Mode Register 3 (ICMR3)

Address(es): IIC0.ICMR3 4005 3004h, IIC1.ICMR3 4005 3104h, IIC2.ICMR3 4005 3204h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Filter out noise of up to 1 IICϕ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IICϕ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IICϕ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IICϕ cycles (4-stage filter).	R/W
b2	ACKBR	Receive Acknowledge	0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Write protect ACKBT bit 1: Write enable ACKBT bit.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: Set the RDRF flag on the rising edge of the ninth SCL clock cycle (no low-hold on the SCLn line on the falling edge of the eighth clock cycle) 1: Set the RDRF flag on the rising edge of the eighth SCL clock cycle (low-hold on the SCLn line low on the falling edge of the eighth clock cycle). Low-hold is released by writing to ACKBT.	R/W*2
b6	WAIT	WAIT	0: No wait (no low-hold between ninth clock cycle and first clock cycle) 1: Wait (low-hold between ninth clock cycle and first clock cycle). Low-hold is released by reading ICDRR.	R/W*2
b7	SMBS	SMBus/IIC-Bus Select	0: Select I^{2}C bus 1: Select SMBus.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to the ACKWP and ACKBT bits at the same time, the ACKBT bit does not set to 1.
Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see section 36.6, Digital Noise Filter Circuits.

Note: Set the noise range to be filtered within a range less than the SCLn line high- or low-level period. If the noise range is set to a value of [SCL clock width: high- or low-level period, whichever is shorter] - [1.5 internal reference clock (IICϕ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise, which might prevent the IIC from operating normally.

ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1

- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode.

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition issuance is detected (when a stop condition is detected with the SP bit in ICCR2 set to 1)
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCLn line low on the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low on the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the eighth SCL clock cycle, and the SCLn line is held low on the falling edge of the eighth SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

WAIT bit (WAIT)

The WAIT bit controls whether to force a low-hold between the ninth SCL clock cycle and the first SCL clock cycle, until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without a low-hold between the ninth and the first SCL clock cycle. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, always read ICDRR first.

SMBS bit (SMBus/IIC-Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

36.2.6 I^{2}C Bus Function Enable Register (ICFER)

Address(es): IIC0.ICFER 4005 3005h, IIC1.ICFER 4005 3105h, IIC2.ICFER 4005 3205h

	b7	b6	b5	b4	b3	b2	b1	b0
	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOE	Timeout Function Enable	0: Disable 1: Enable.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost.	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: Disable 1: Enable.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Disable 1: Enable.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension).	R/W
b5	NFE	Digital Noise Filter Circuit Enable	0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit.	R/W
b6	SCLE	SCL Synchronous Circuit Enable	0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit.	R/W
b7	FMPE*1	Fast-Mode Plus Enable	0: Do not use the Fm+ slope control circuit for the SCLn and SDAn pins 1: Use the Fm+ slope control circuit for the SCLn and SDAn pins.	R/W

Note 1. The Fast-mode Plus enable bit (FMPE) is supported only by IIC0 (SCL0-A, SDA0-A). Bit [7] is reserved in IIC1 and IIC2.

TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see section 36.12.1, Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode, for example when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is

0, the next transfer operation is continued regardless of the received acknowledge content.

For details, see section 36.9.2, NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock at the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I^{2}C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuance of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

FMPE bit (Fast-Mode Plus Enable)

The FMPE bit specifies whether to use a slope control circuit for Fast-mode Plus (Fm+).

When this bit is set to 1, a slope control circuit conforming to the I^{2}C bus Fast-mode Plus (Fm+) standard (tof) is selected. When this bit is set to 0, a slope control circuit conforming to the I^{2}C bus Standard-mode (Sm) and Fast-mode (Fm) standards (tof) is selected.

Set this bit to 1 when using transmission rates up to 1 Mbps (Fast-mode Plus (Fm+) standard). Set it to 0 when using other transmission rates (up to 100 kbps (Sm) or up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps).

36.2.7 I^{2}C Bus Status Enable Register (ICSER)

Address(es): IIC0.ICSER 4005 3006h, IIC1.ICSER 4005 3106h, IIC2.ICSER 4005 3206h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2.	R/W
b3	GCAE	General Call Address Enable	0: Disable general call address detection 1: Enable general call address detection.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Disable device-ID address detection 1: Enable device-ID address detection.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Disable host address detection 1: Enable host address detection.	R/W

SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE bit (Device-ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device-ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device-ID address, and it recognizes the first frame as a normal slave address.

For details on this function, see section 36.7.3, Device-ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

36.2.8 I^{2}C Bus Interface Enable Register (ICIER)

Address(es): IIC0.ICIER 4005 3007h, IIC1.ICIER 4005 3107h, IIC2.ICIER 4005 3207h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Disable timeout interrupt (TMOI) request 1: Enable timeout interrupt (TMOI) request.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Disable arbitration-lost interrupt (ALI) request 1: Enable arbitration-lost interrupt (ALI) request.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Disable start condition detection interrupt (STI) request 1: Enable start condition detection interrupt (STI) request.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Disable stop condition detection interrupt (SPI) request 1: Enable stop condition detection interrupt (SPI) request.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: Disable NACK reception interrupt (NAKI) request 1: Enable NACK reception interrupt (NAKI) request.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Disable receive data full interrupt (IICn_RXI) request 1: Enable receive data full interrupt (IICn_RXI) request.	R/W
b6	TEIE	Transmit End Interrupt Request Enable	0: Disable transmit end interrupt (IICn_TEI) request 1: Enable transmit end interrupt (IICn_TEI) request.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Disable transmit data empty interrupt (IICn_TXI) request 1: Enable transmit data empty interrupt (IICn_TXI) request.	R/W

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt (TMOI) requests when the TMOF flag in ICSR2 is 1. To cancel a TMOI interrupt request, set the TMOF flag or the TMOIE bit to 0.

ALIE bit (Arbitration-Lost Interrupt Request Enable)

The ALIE bit enables or disables arbitration-lost interrupt (ALI) requests when the AL flag in ICSR2 is 1. To cancel an ALI interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt (STI) requests when the START flag in ICSR2 is 1. To cancel an STI interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt (SPI) requests when the STOP flag in ICSR2 is 1. To cancel an SPI interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt (NAKI) requests when the NACKF flag in ICSR2 is 1. To cancel an NAKI interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IICn_RXI) requests when the RDRF flag in ICSR2 is 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IICn_TEI) requests when the TEND flag in ICSR2 is 1. To cancel an IICn_TEI interrupt request, set the TEND flag or the TEIE bit to 0.

TIE bit (Transmit Data Empty Interrupt Request Enable)

The TIE bit enables or disables transmit data empty interrupt (IICn_TXI) requests when the TDRE flag in ICSR2 is 1.

36.2.9 I^{2}C Bus Status Register 1 (ICSR1)

Address(es): IIC0.ICSR1 4005 3008h, IIC1.ICSR1 4005 3108h, IIC2.ICSR1 4005 3208h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 not detected 1: Slave address 0 detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 not detected 1: Slave address 1 detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 not detected 1: Slave address 2 detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address not detected 1: General call address detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command not detected 1: Device-ID command detected. This bit is set to 1 when the first frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]).	R/(W) *1

Bit	Symbol	Bit name	Description	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address not detected 1: Host address detected. This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

Note 1. Only 0 can be written, to clear the flag.

AASy flag (Slave Address y Detection Flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy), with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address does not match the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

GCA flag (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled)

The GCA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

DID flag (Device-ID Address Detection Flag)

The DID flag indicates whether the device-ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled).
  The DID flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start or restart condition is detected does not match a value of (device ID (1111 100b)), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

HOA flag (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).
  The HOA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled)
  The HOA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

36.2.10 I^{2}C Bus Status Register 2 (ICSR2)

Address(es): IIC0.ICSR2 4005 3009h, IIC1.ICSR2 4005 3109h, IIC2.ICSR2 4005 3209h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout not detected 1: Timeout detected.	R/(W) *1

Bit	Symbol	Bit name	Description	R/W
b1	AL	Arbitration-Lost Flag	0: Arbitration not lost 1: Arbitration lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition not detected 1: Start condition detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition not detected 1: Stop condition detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK not detected 1: NACK detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: ICDRR contains no receive data 1: ICDRR contains receive data.	R/(W) *1
b6	TEND	Transmit End Flag	0: Data being transmitted 1: Data transmit complete.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: ICDRT contains transmit data 1: ICDRT contains no transmit data.	R

Note 1. Only 0 can be written, to clear the flag.

TMOF flag (Timeout Detection Flag)

The TMOF flag is set to 1 when the IIC detects a timeout because the SCLn line state remains unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

AL flag (Arbitration-Lost Flag)

The AL flag indicates that bus mastership was lost in arbitration because of a bus conflict or some other reason when a start condition was issued or an address and data was transmitted. The IIC monitors the level on the SDAn line during transmission and, if the level on the line does not match the value of the bit being output, is set the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The IIC can also set the flag to indicate the detection of arbitration loss during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock except for the ACK period during data transmission in master transmit mode
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDAn line level
- When the ST bit in ICCR2 is 1 (start condition requested), with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled (ICFER.NALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Table 36.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	**NALE**	**SALE**	**AL**	**Error**	**Arbitration-lost generation source**
1	×	×	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected, while the ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START flag (Start Condition Detection Flag)

The START flag indicates whether a start condition was detected.

[Setting condition]

- When a start (or restart) condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

STOP flag (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition was detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

NACKF flag (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates whether the IDCRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR
  The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start (or restart) condition is detected with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TEND flag (Transmit End Flag)

The TEND flag indicates whether data transmission is still being transmitted or is complete.

[Setting condition]

- On the rising edge of the ninth SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the IIC suspends data transmission and reception. Here, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the ninth clock cycle, but the TDRE flag does not set to 1.

36.2.11 I^{2}C Bus Wakeup Unit Register (ICWUR)

Address(es): IIC0.ICWUR 4005 3016h

	b7	b6	b5	b4	b3	b2	b1	b0
	WUE	WUIE	WUF	WUACK	—	—	—	WUAFA
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	WUAFA	Wakeup Analog Filter Additional Selection	0: Do not add the wakeup analog filter 1: Add the wakeup analog filter.	R/W
b3 to b1	—	Reserved	These bit are read as 0. The write value should be 0.	R/W
b4	WUACK	ACK Bit for Wakeup Mode	Choice of four response modes in combination with ICCR1.IICRST and WUACK. See Table 36.5.	R/W
b5	WUF	Wakeup Event Occurrence Flag	0: Slave address not matching during wakeup 1: Slave address matching during wakeup.	R/W
b6	WUIE	Wakeup Interrupt Request Enable	0: Disable wakeup interrupt request (IIC0_WUI) 1: Enable wakeup interrupt request (IIC0_WUI).	R/W
b7	WUE	Wakeup Function Enable	0: Disable wakeup function 1: Enable wakeup function.	R/W

Table 36.5 Wakeup mode

IICRST	WUACK	Operation mode	Description
0	0	Normal wakeup mode 1	ACK response on ninth SCL, and SCL low-hold after ninth SCL.
0	1	Normal wakeup mode 2	No ACK response immediately and SCL low-hold between eight and ninth SCL. SCL low-hold release and ACK response on ninth SCL.
1	0	Command recovery mode	ACK response on ninth SCL and no SCL low-hold.
1	1	EEP response mode	NACK response on ninth SCL and no SCL low-hold.

WUF flag (Wakeup Event Occurrence Flag)

The WUF flag indicates whether the slave address is matching during wakeup.

[Setting condition]

- When PCLKB is supplied after a slave-address match in the first eighth SCL low during wakeup mode.

[Clearing conditions]

- When 0 is written to the WUF flag after reading WUF = 1
- When ICE = 0 and IICRST = 1.

36.2.12 I2C Bus Wakeup Unit Register 2 (ICWUR2)

Address(es): IIC0.ICWUR2 4005 3017h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	WUSYF	WUASYF	WUSEN
Value after reset:	1	1	1	1	1	1	0	1

Bit	Symbol	Bit name	Description	R/W
b0	WUSEN	Wakeup Function Synchronous Enable	0: IIC asynchronous circuit enable 1: IIC synchronous circuit enable.	R/W
b1	WUASYF	Wakeup Function Asynchronous Operation Status Flag	0: IIC synchronous circuit enable condition 1: IIC asynchronous circuit enable condition.	R
b2	WUSYF	Wakeup Function Synchronous Operation Status Flag	0: IIC asynchronous circuit enable condition 1: IIC synchronous circuit enable condition.	R
b7 to b3	—	Reserved	These bits are read as1. The write value should be 1.	R/W

WUSEN bit (Wakeup Function Synchronous Enable)

It combines with the WUASYF flag (or WUSYF flag) at wakeup effective function (ICWUR.WUE = 1), and the PCLKB synchronous operation and the PCLKB asynchronous operation are switched.

[When switching from the PCLKB synchronous operation to the PCLKB asynchronous operation]

It changes into the PCLKB asynchronous operation when the ICCR2.BBSY flag is 0 if the WUASYF flag writes 0 in the WUSEN bit in the state of 0. The reception can operate without depending on the state of operation of PCLKB (with PCLKB stopped) after it switches to the PCLKB asynchronous operation (wakeup event detection operation).

[When switching from the PCLKB asynchronous operation to the PCLKB synchronous operation]

It changes into the PCLKB synchronization and the WUASYF flag becomes 0 at once after writing of 1 when the wakeup event is detected if 1 is written in the WUSEN bit when the WUASYF flag is 1. At the same time, WUASYF flag becomes 0. In other case, it changes into the PCLKB synchronous operation when the stop condition is detected at the wakeup event undetected.

WUASYF flag (Wakeup Function Asynchronous Operation Status Flag)

It is shown that IIC is in the PCLKB asynchronous operation at wakeup effective function (ICWUR.WUE = 1).

[Setting condition]

- When the ICCR2.BBSY flag is 0 with the ICWUR.WUE bit set to 1 after writing 0 to the WUSEN bit.

[Clearing conditions]

- When 1 is written to the WUSEN bit after detecting the wakeup event with ICWUR.WUE bit set to 1.
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wakeup event with WUASY flag set to 1 with ICWUR.WUE bit set to 1.
- When you write 1 in the WUSEN bit with the WUASYF flag detected 1 and the wakeup event in the state of ICWUR.WUE = 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

WUSYF flag (Wakeup Function Synchronous Operation Status Flag)

It is shown that IIC is in the PCLKB synchronous operation at wakeup effective function (ICWUR.WUE = 1). This flag is a value in which the WUASYF flag is always reserved.

[Setting conditions]

- When 1 is written to the WUSEN bit after detecting the wakeup event with ICWUR.WUE bit set to 1 with WUSYF flag cleared to 0 with ICWUR.WUE bit set to 1.
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wakeup event with WUSYF flag cleared to 0 with ICWUR.WUE bit set to 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

[Clearing condition]

- When the ICCR2.BBSY flag is 0 with the ICWUR.WUE bit set to 1 after writing 0 to the WUSEN bit.

36.2.13 Slave Address Register L y (SARLy) (y = 0 to 2)

Address(es): IIC0.SARL0 4005 300Ah, IIC1.SARL0 4005 310Ah, IIC2.SARL0 4005 320Ah,
IIC0.SARL1 4005 300Ch, IIC1.SARL1 4005 310Ch, IIC2.SARL1 4005 320Ch,
IIC0.SARL2 4005 300Eh, IIC1.SARL2 4005 310Eh, IIC2.SARL2 4005 320Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SVA0	10-Bit Address LSB	Slave address setting.	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	Slave address setting.	R/W

SVA0 bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

SVA[6:0] bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combine with the SVA0 bit to form the lower 8 bits of a 10-bit address.

When the SARyE bit in ICSER is 0, the setting in these bits is ignored.

36.2.14 Slave Address Register U y (SARUy) (y = 0 to 2)

Address(es): IIC0.SARU0 4005 300Bh, IIC1.SARU0 4005 310Bh, IIC2.SARU0 4005 320Bh,
IIC0.SARU1 4005 300Dh, IIC1.SARU1 4005 310Dh, IIC2.SARU1 4005 320Dh,
IIC0.SARU2 4005 300Fh, IIC1.SARU2 4005 310Fh, IIC2.SARU2 4005 320Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: Select 7-bit address format 1: Select 10-bit address format.	R/W

Bit	Symbol	Bit name	Description	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	Slave address setting.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FS bit (7-Bit/10-Bit Address Format Select)

The FS bit selects 7- or 10-bit format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

SVA[1:0] bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

36.2.15 I^{2}C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): IIC0.ICBRL 4005 3010h, IIC1.ICBRL 4005 3110h, IIC2.ICBRL 4005 3210h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	BRL[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low-Level Period	Low-level period of SCL clock.	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register that sets the low-level period of the SCL clock. ICBRL also works to generate the data setup time for the automatic SCL low-hold operation (see section 36.9, Automatic Low-Hold Function for SCL). When the IIC is used only in slave mode, this register must be set to a value longer than the data setup time.*1 ICBRL counts the low-level period with the internal reference clock source (IICϕ) specified in the CKS[2:0] bits in ICMR1. If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns for up to 100 kbps: Standard-mode (Sm)
100 ns for up to 400 kbps: Fast-mode (Fm)
50 ns for up to 1 Mbps: Fast-mode plus (Fm+)

36.2.16 I^{2}C Bus Bit Rate High-Level Register (ICBRH)

Address(es): IIC0.ICBRH 4005 3011h, IIC1.ICBRH 4005 3111h, IIC2.ICBRH 4005 3211h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High-Level Period	High-level period of SCL clock.	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register that sets the high-level period of the SCL clock. ICBRH is valid in master mode. If the IIC is used only in slave mode, no setting is required in this register. ICBRH counts the high-level period with the internal reference clock source (IICϕ) specified in the CKS[2:0] bits in ICMR1. If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

1. ICFER.SCLE = 0
   Transfer rate = $1/\{[(BRH + 1) + (BRL + 1)]/IIC\varphi^{*1} + tr^{*2} + tf^{*2}\}$
   Duty cycle = $\{tr + [(BRH + 1)/IIC\varphi]\}/\{tr + tf + [(BRH + 1) + (BRL + 1)]/IIC\varphi\}$
2. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = $1/\{[(BRH + 3) + (BRL+ 3)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 3)/IIC\varphi]\}/\{tr + tf + [(BRH + 3) + (BRL + 3)]/IIC\varphi\}$
3. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = $1/\{[(BRH + 3 + nf^{*3}) + (BRL + 3 + nf)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 3 + nf)/IIC\varphi]\}/\{tr + tf + [(BRH + 3 + nf) + (BRL + 3 + nf)]/IIC\varphi\}$
4. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
   Transfer rate = $1/\{[(BRH + 2) + (BRL + 2)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 2)/IIC\varphi]\}/\{tr + tf + [(BRH + 2) + (BRL + 2)]/IIC\varphi\}$
5. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
   Transfer rate = $1/\{[(BRH + 2 + nf) + (BRL + 2 + nf)]/IIC\varphi + tr + tf\}$
   Duty cycle = $\{tr + [(BRH + 2 + nf)/IIC\varphi]\}/\{tr + tf + [(BRH + 2 + nf) + (BRL + 2 + nf)]/IIC\varphi\}$

Note 1. IICφ = PCLKB × division ratio
Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I^{2}C bus standard from NXP Semiconductors.
Note 3. nf = Number of digital noise filters selected in the ICMR3.NF bit.

Table 36.6 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0](ICBRH)	BRL[4:0](ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	14 (EEh)	17 (F1h)	60	—	(1)
400	010b	8 (E8h)	19 (F3h)	60	—	(1)
1000	000b	15 (EFh)	29 (FDh)	60	—	(1)

Table 36.7 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0](ICBRH)	BRL[4:0](ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	13 (EDh)	16 (F0h)	60	—	(4)
400	010b	7 (E7h)	18 (F2h)	60	—	(4)
1000	000b	13 (EDh)	27 (FBh)	60	—	(2)

Table 36.8 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0]	BRH[4:0](ICBRH)	BRL[4:0](ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	11 (EBh)	14 (EEh)	60	01b	(5)
400	010b	5 (E5h)	16 (F0h)	60	01b	(5)
1000	000b	11 (EBh)	25 (F9h)	60	01b	(3)

36.2.17 I^{2}C Bus Transmit Data Register (ICDRT)

Address(es): IIC0.ICDRT 4005 3012h, IIC1.ICDRT 4005 3112h, IIC2.ICDRT 4005 3212h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

When ICDRT detects a space in the IIC-Bus Shift Register (ICDRS), it transfers the transmit data that was written to ICDRT to ICDRS and starts transmitting data in transmit mode. The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IICn_TXI) request is generated.

36.2.18 I^{2}C Bus Receive Data Register (ICDRR)

Address(es): IIC0.ICDRR 4005 3013h, IIC1.ICDRR 4005 3113h, IIC2.ICDRR 4005 3213h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

When 1 byte of data is received, the received data is transferred from the IIC-Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received. The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR once when a receive data full interrupt (IICn_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the IIC automatically holds the SCL clock low 1 cycle before the RDRF flag is set to 1 next.

36.2.19 I²C Bus Shift Register (ICDRS)

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—

ICDRS is an 8-bit shift register for data transmit and receive. During transmission, transmit data is transferred from ICDRT to ICDRS and is transmitted from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

36.3 Operation

36.3.1 Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start or restart condition is an address frame that specifies a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 36.3 shows the I²C bus format, and Figure 36.4 shows the I²C bus timing.

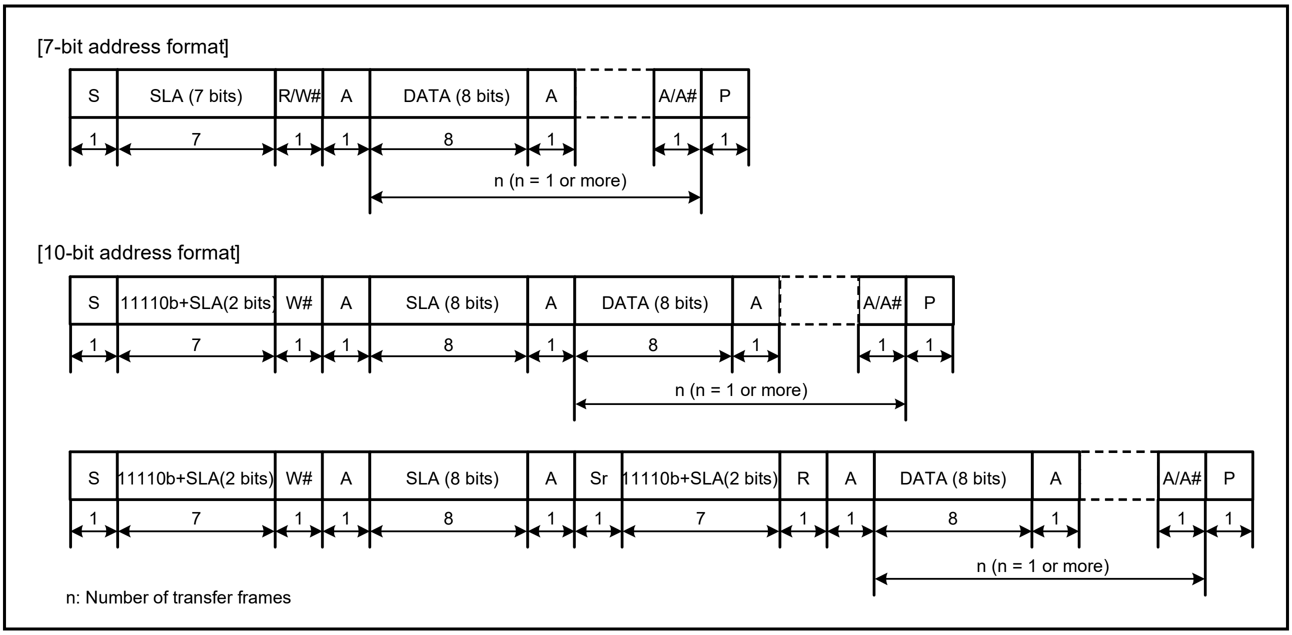


Figure 36.3 I²C bus format

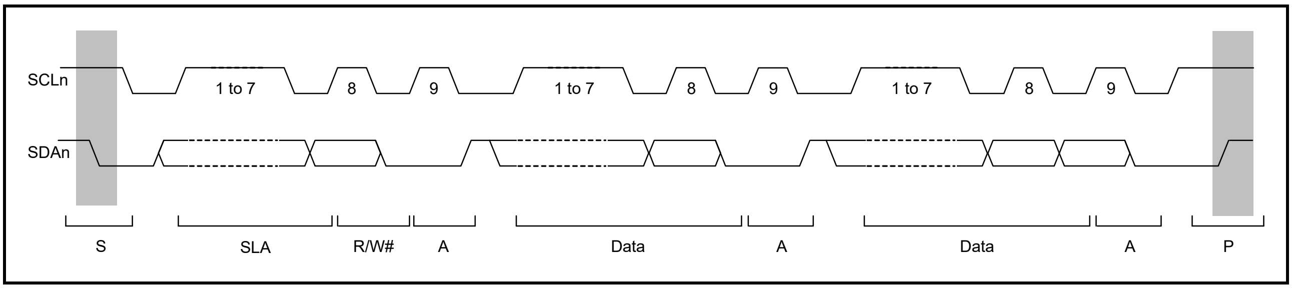


Figure 36.4 I²C bus timing when the SLA setting = 7 bits

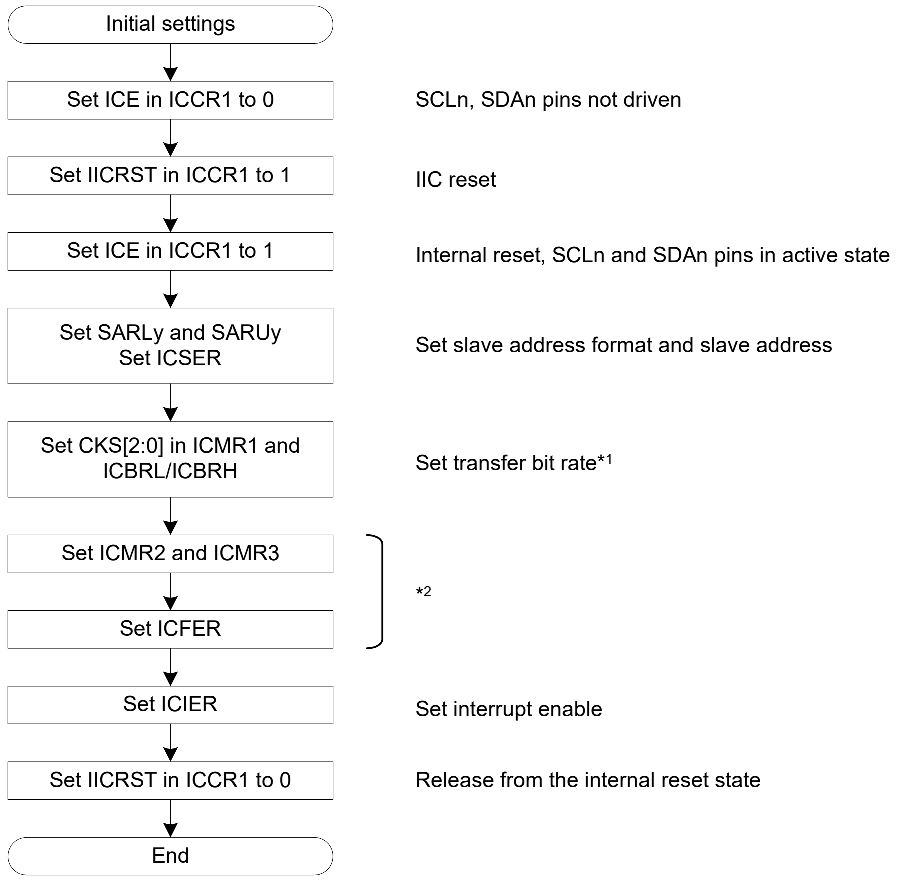
S: Start condition. The master device drives the SDAn line low from high while the SCLn line is high.

SLA: Slave address, by which the master device selects a slave device.

R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.
A:	Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
A#:	Not Acknowledge. The receive device drives the SDAn line high.
Sr:	Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
DATA:	Transmitted or received data.
P:	Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

36.3.2 Initial Settings

Before starting data transmission and reception, initialize the IIC using the procedure shown in Figure 36.5. Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (IIC reset) with the ICCR1.ICE bit set to 0 (SCLn and SDAn pins in inactive state). This internal reset procedure initializes the flags and the internal state of the ICSR1 register. Next, set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2), and set the other registers as required (for the initial IIC settings, see Figure 36.5). When the required register settings are complete, set the ICCR1.IICRST bit to 0, releasing the IIC reset. This step is not necessary if initialization of the IIC is already complete.



y = 0 to 2

Note 1. When the IIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.

Note 2. Set these registers as required.

Figure 36.5 Example IIC initialization flow

36.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 36.6 shows an example of master transmission, and Figure 36.7 to Figure 36.9 show the operation timing in master transmission.

To set up and perform master transmission:

1. Process initial settings. For details, see section 36.3.2, Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit automatically is set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the IIC continues in master transmit mode.
   Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
   For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.
4. Check that the TDRE flag in ICSR2 is 1, and then write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition. For details on issuing a stop condition, see section 36.11.3, Issuing a Stop Condition.
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

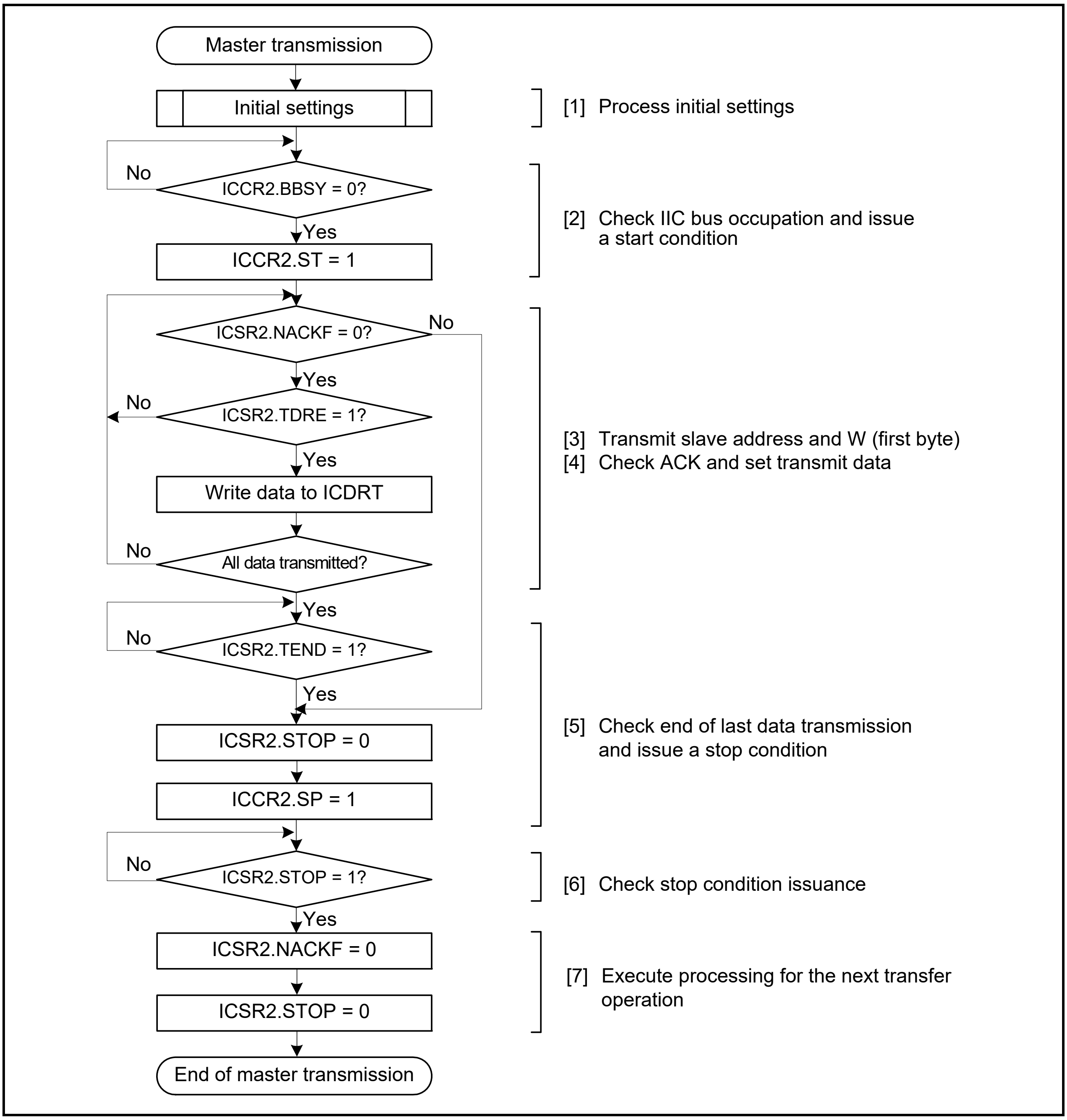


Figure 36.6 Example master transmission flow

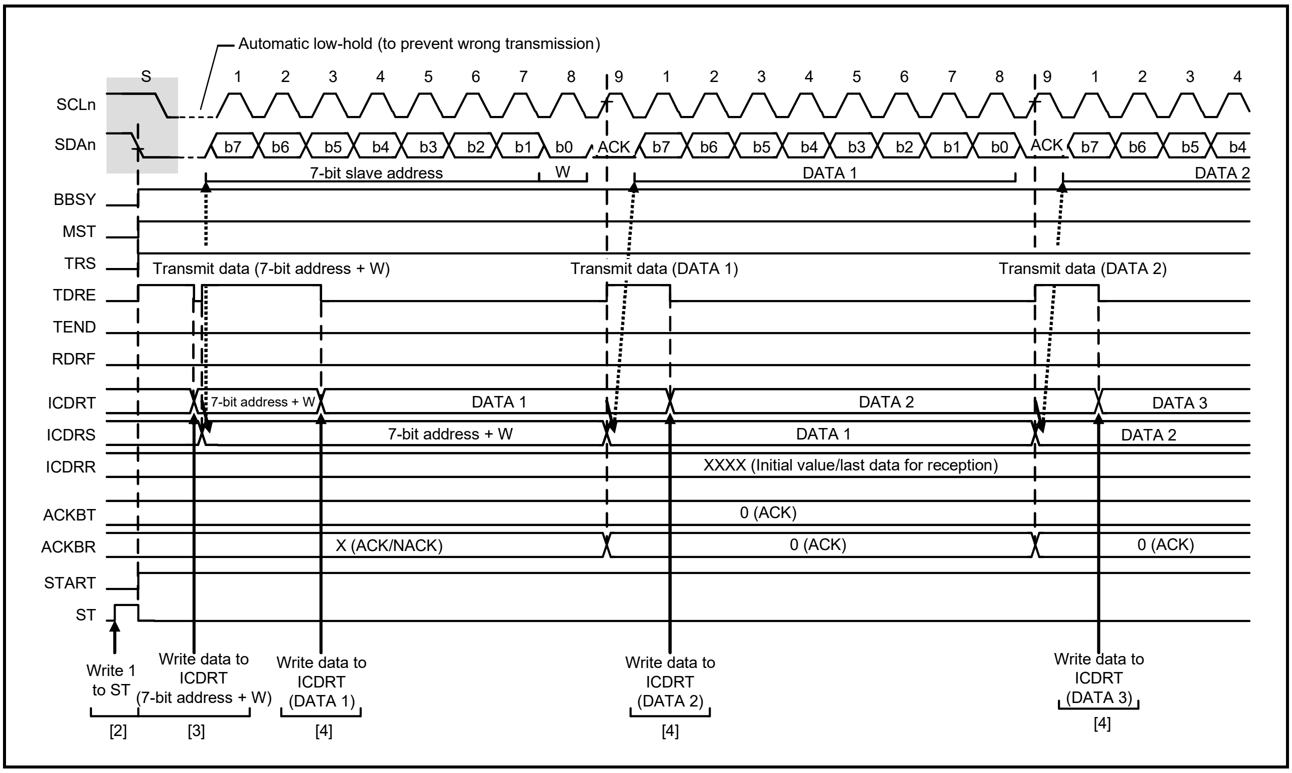


Figure 36.7 Master transmit operation timing (1) with 7-bit address format

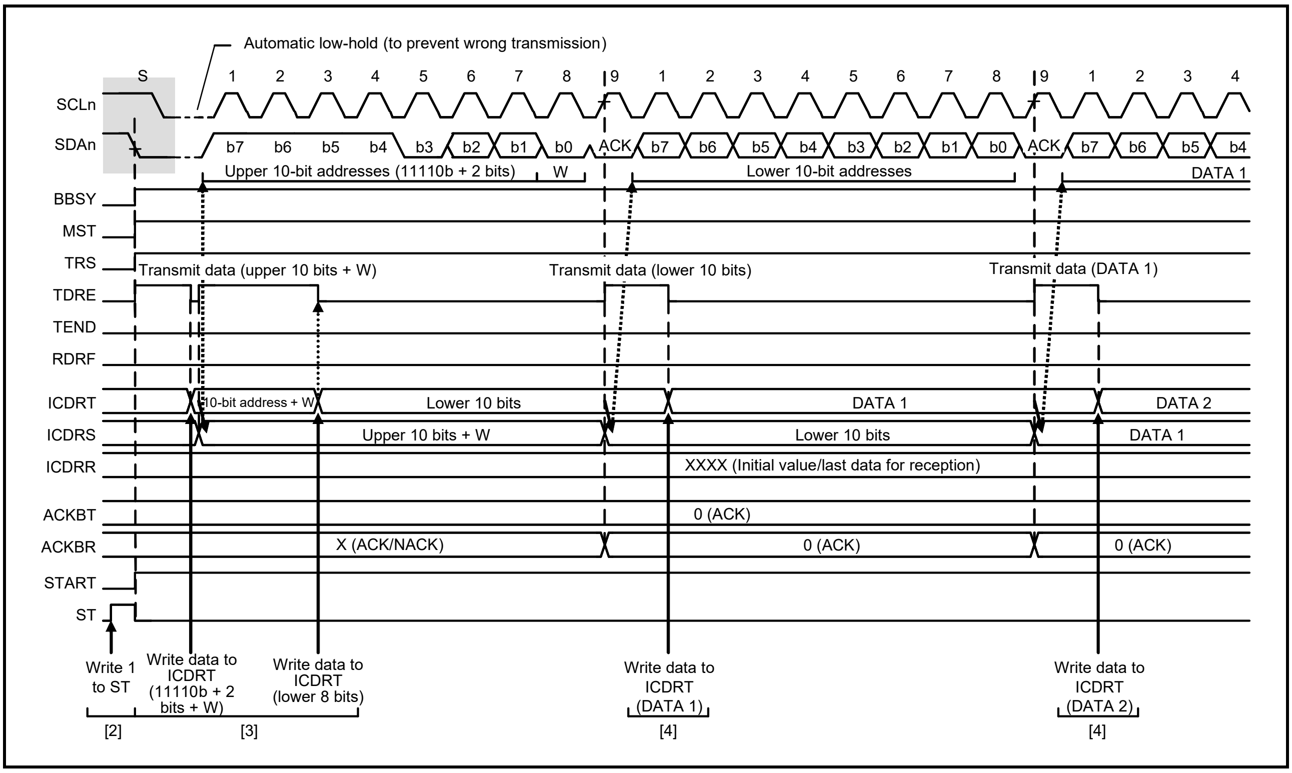


Figure 36.8 Master transmit operation timing (2) with 10-bit address format

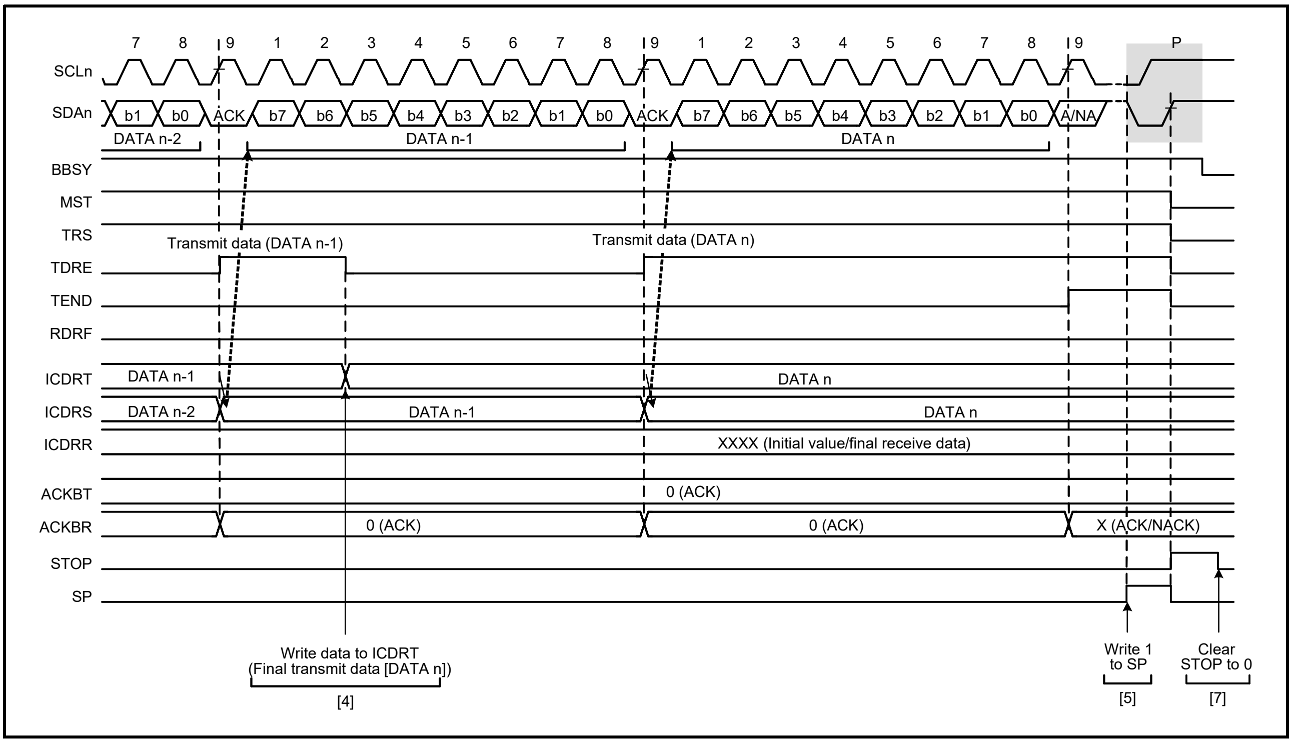


Figure 36.9 Master transmit operation timing (3)

36.3.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 36.10 and Figure 36.11 show examples of master reception (7-bit address format), and Figure 36.12 to Figure 36.14 show the operation timing in master reception.

To set up and perform master reception:

1. Process initial settings. For details, see section 36.3.2, Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically is set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag automatically is set to 1.
   Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
   For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the IIC in master receive mode.
4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. This makes the IIC start output of the SCL

clock and start data reception.

5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the eighth or ninth cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR at this time produces the received data, and the RDRF flag automatically is set to 0 at the same time. Additionally, the value of the acknowledgment field received during the ninth cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1, for wait insertion, before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

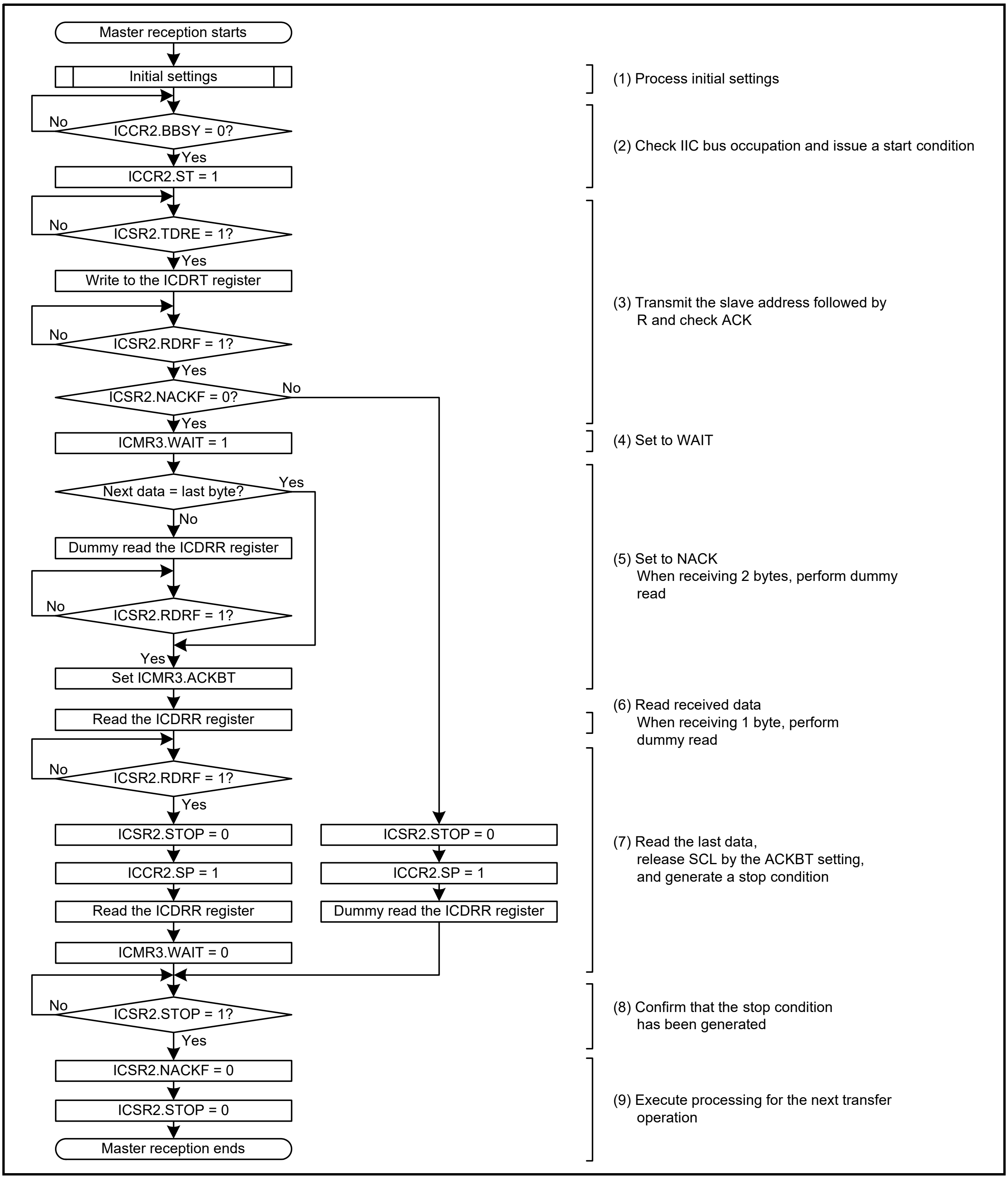


Figure 36.10 Example master reception flow with 7-bit address format and 1 or 2 bytes

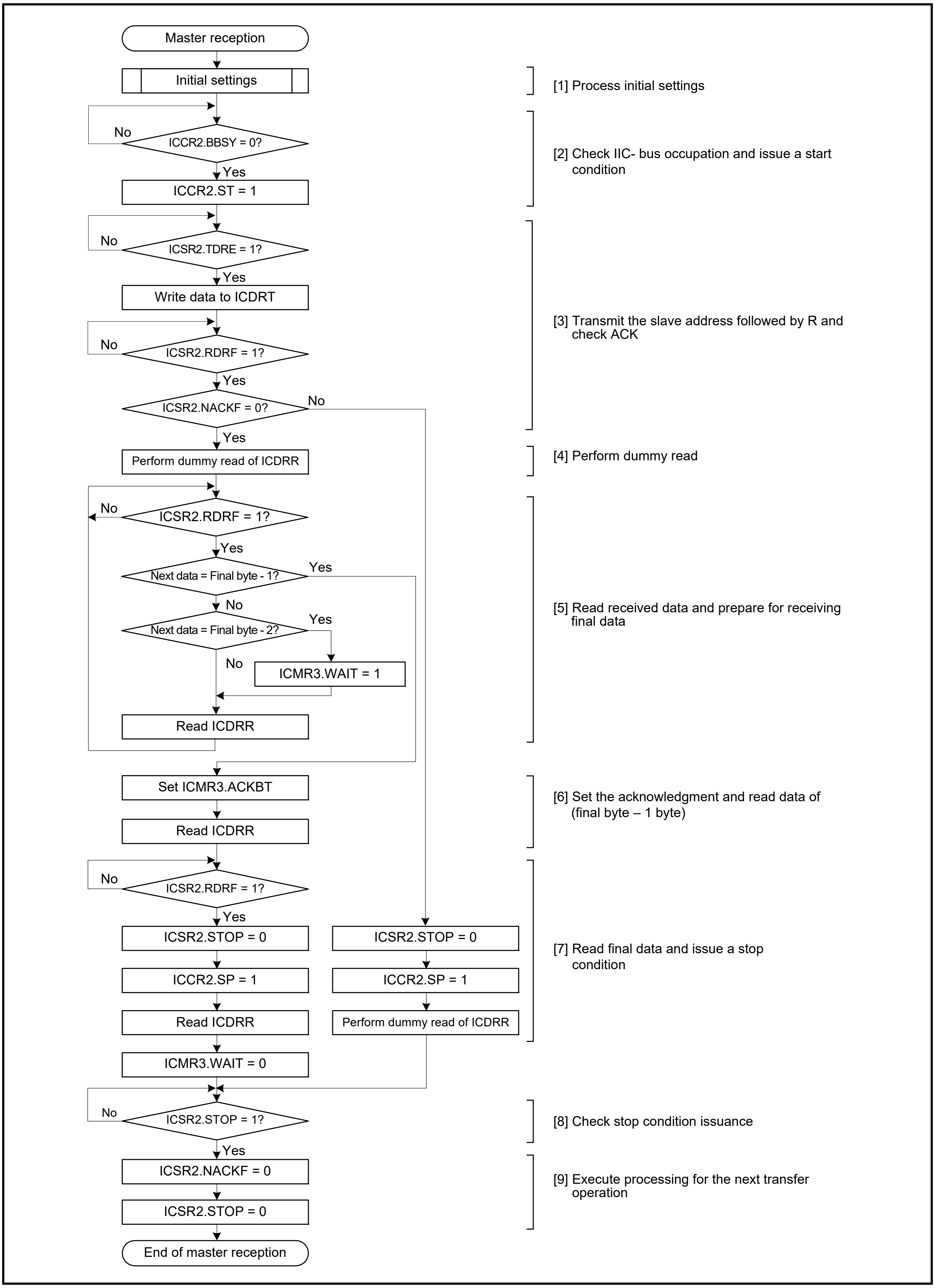


Figure 36.11 Example master reception flow with 7-bit address format and 3 or more bytes

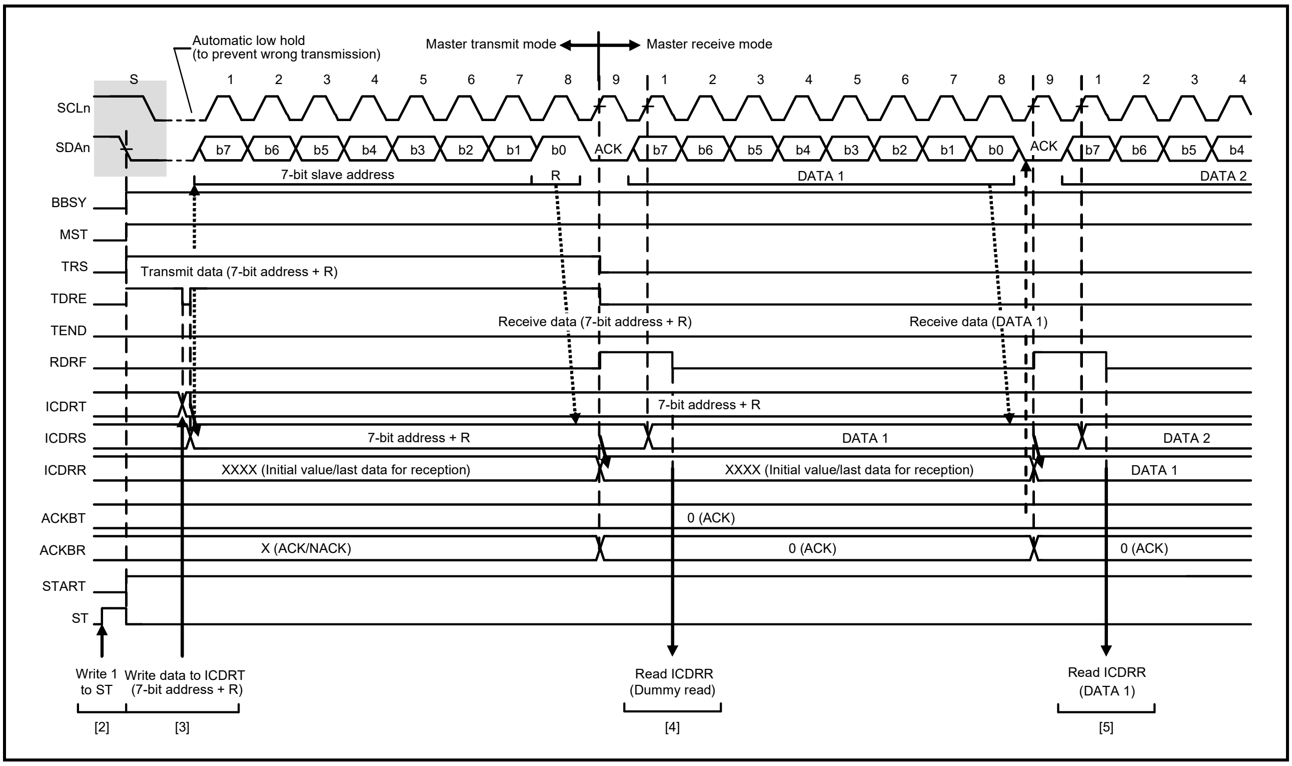


Figure 36.12 Master receive operation timing (1) with 7-bit address format, when RDRFS = 0

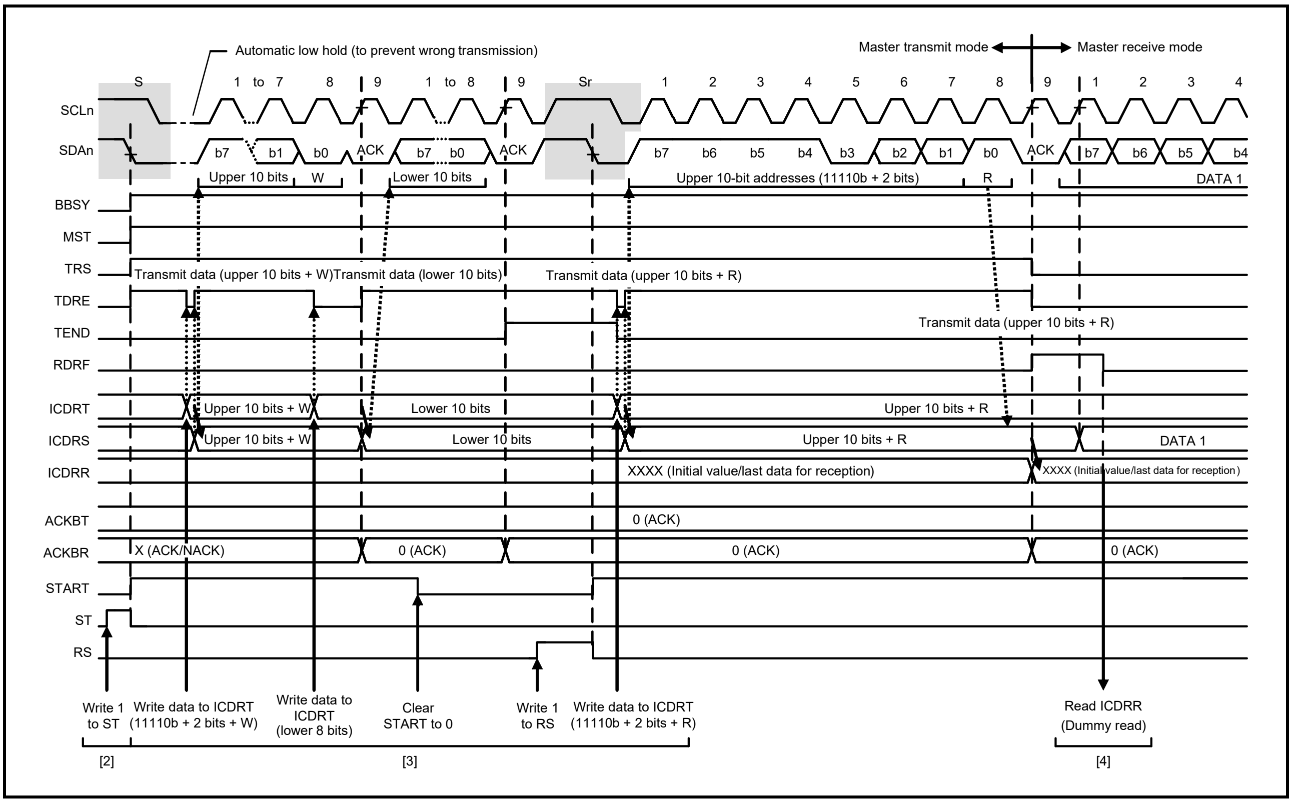


Figure 36.13 Master receive operation timing (2) with 10-bit address format, when RDRFS = 0

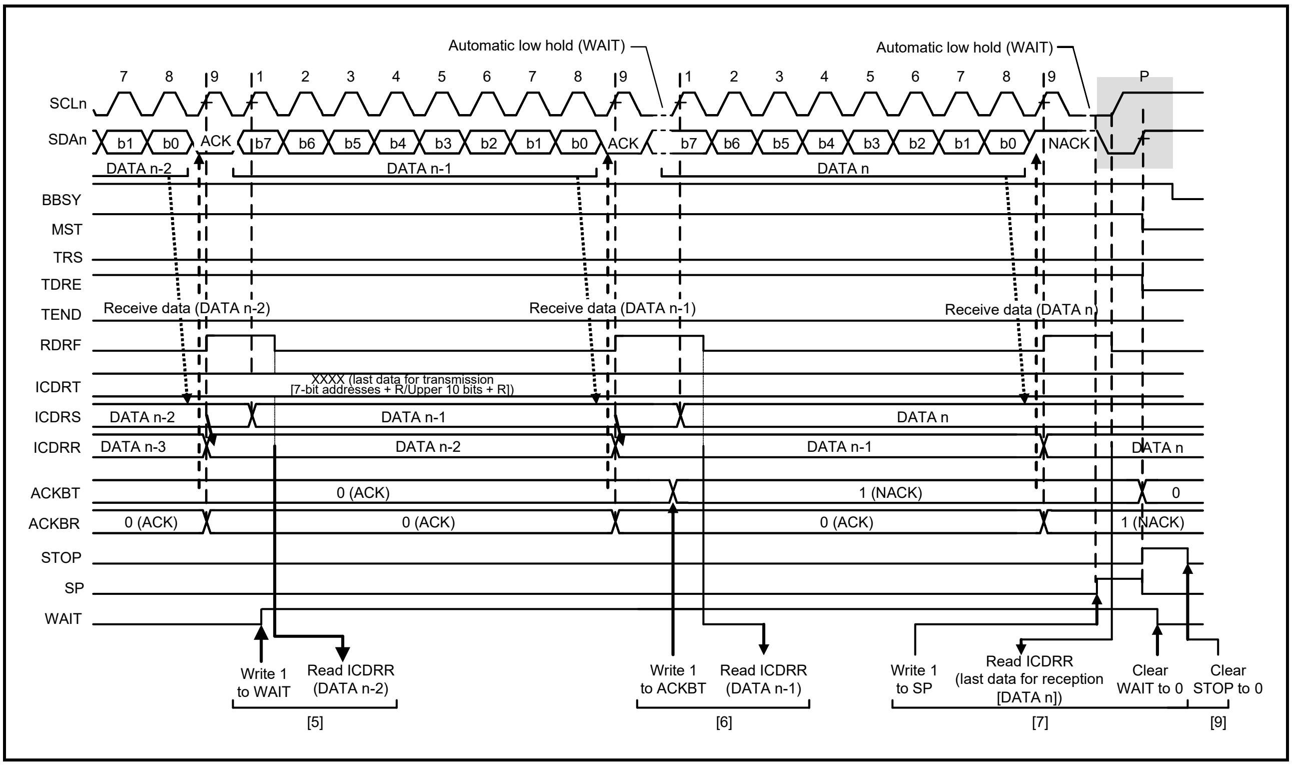


Figure 36.14 Master receive operation timing (3) when RDRFS = 0

36.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 36.15 shows an example of slave transmission, and Figure 36.16 and Figure 36.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. Process initial settings. For details, see section 36.3.2, Initial Settings.
   After the initial settings, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit that was also received at this time is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TEND flag is 1, and then write the transmit data to the ICDRT register. At this time, if the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the ninth falling edge of the SCL clock.
5. When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2), the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

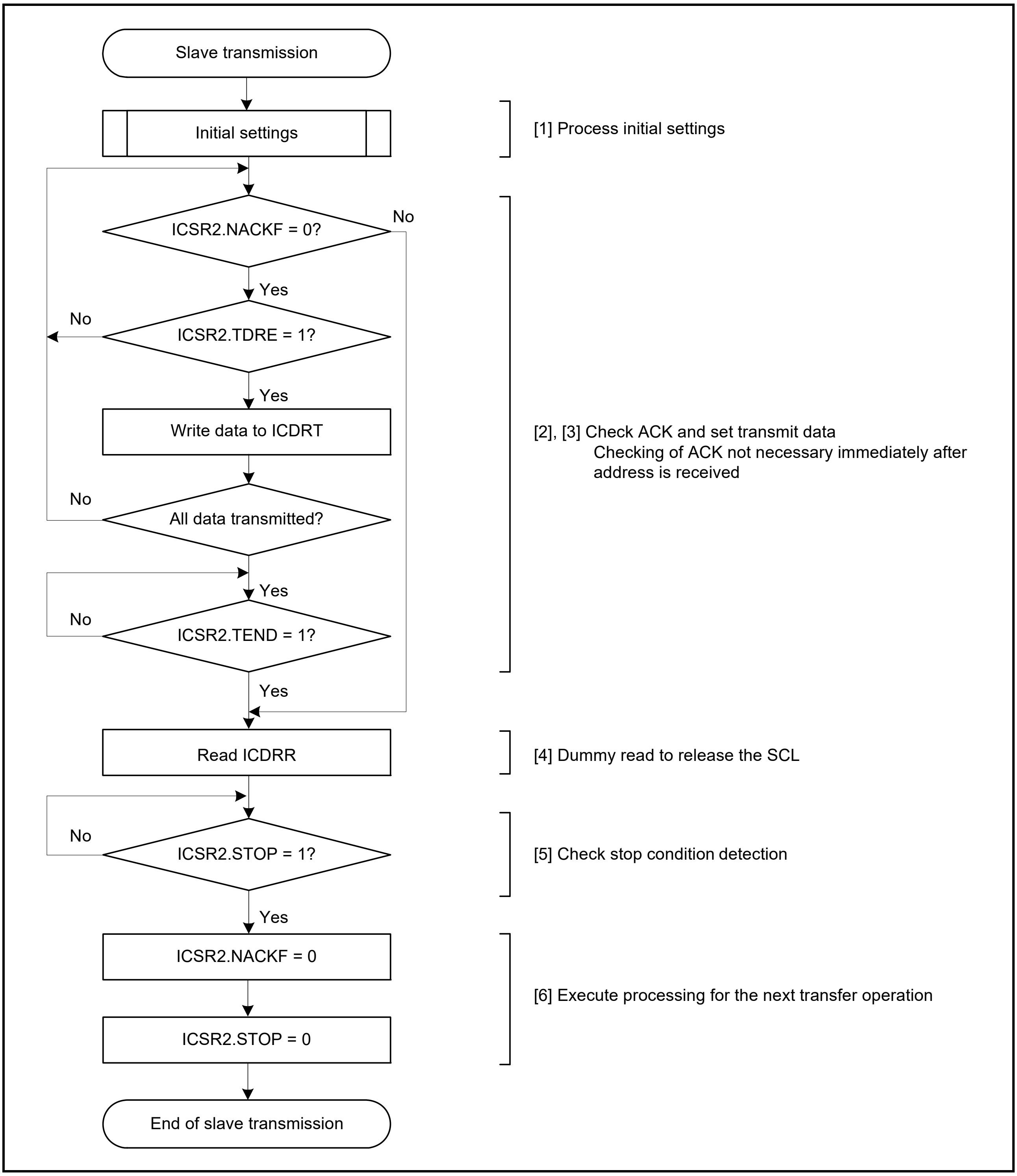


Figure 36.15 Example slave transmission flow

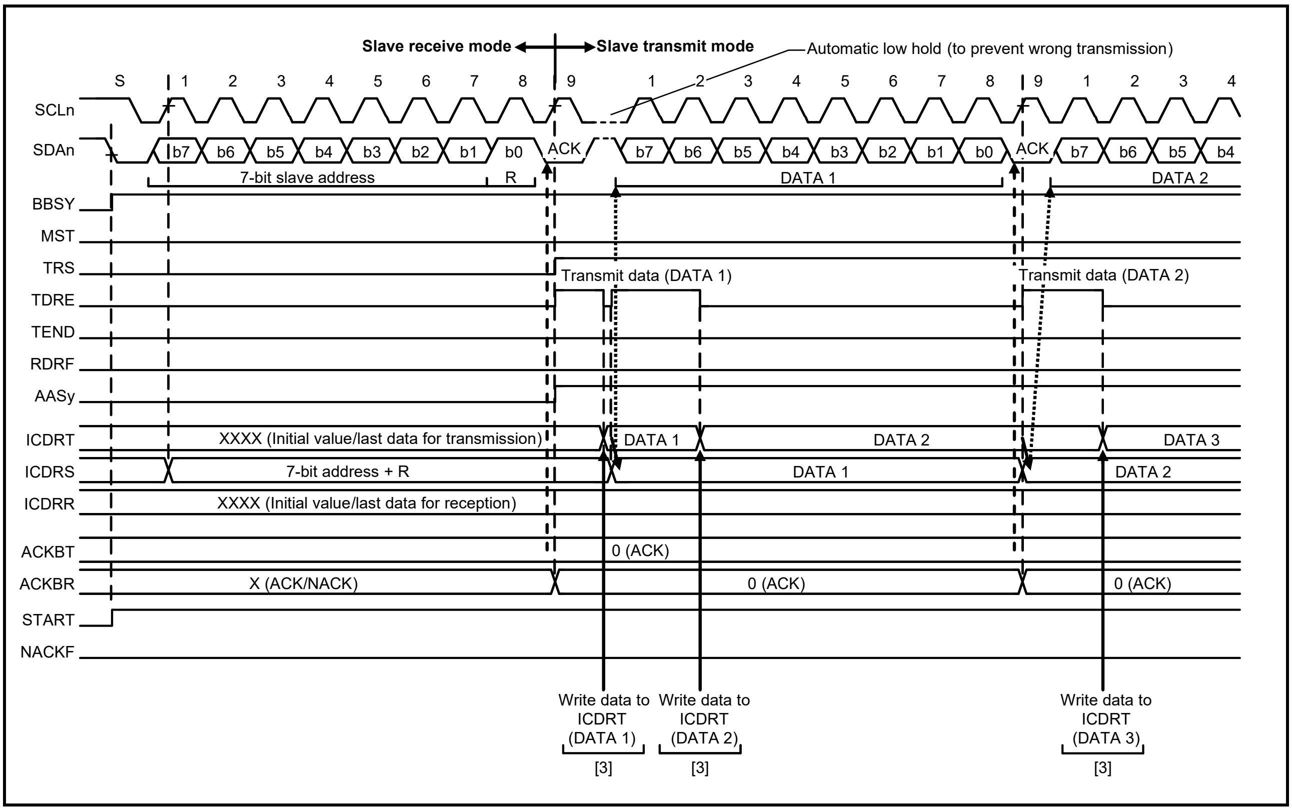


Figure 36.16 Slave transmit operation timing (1) with 7-bit address format

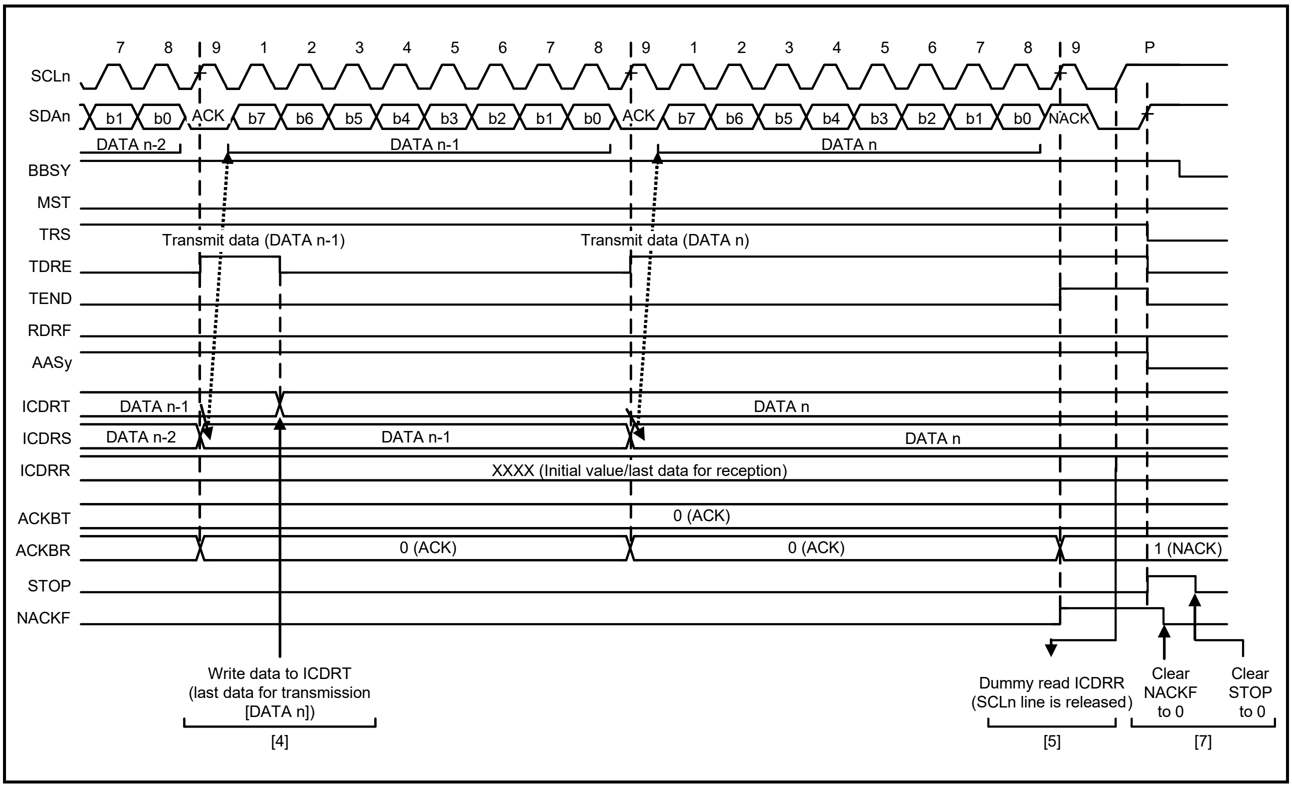


Figure 36.17 Slave transmit operation timing (2)

36.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 36.18 shows an example of slave reception, and Figure 36.19 and Figure 36.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. Process initial settings. For details, see section 36.3.2, Initial Settings.
   After the initial settings, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit that was also received at this time is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, and then dummy read ICDRR. The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until one SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held at the low level.
   When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 0.
6. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.STOP flag to 0 for the next transfer operation.

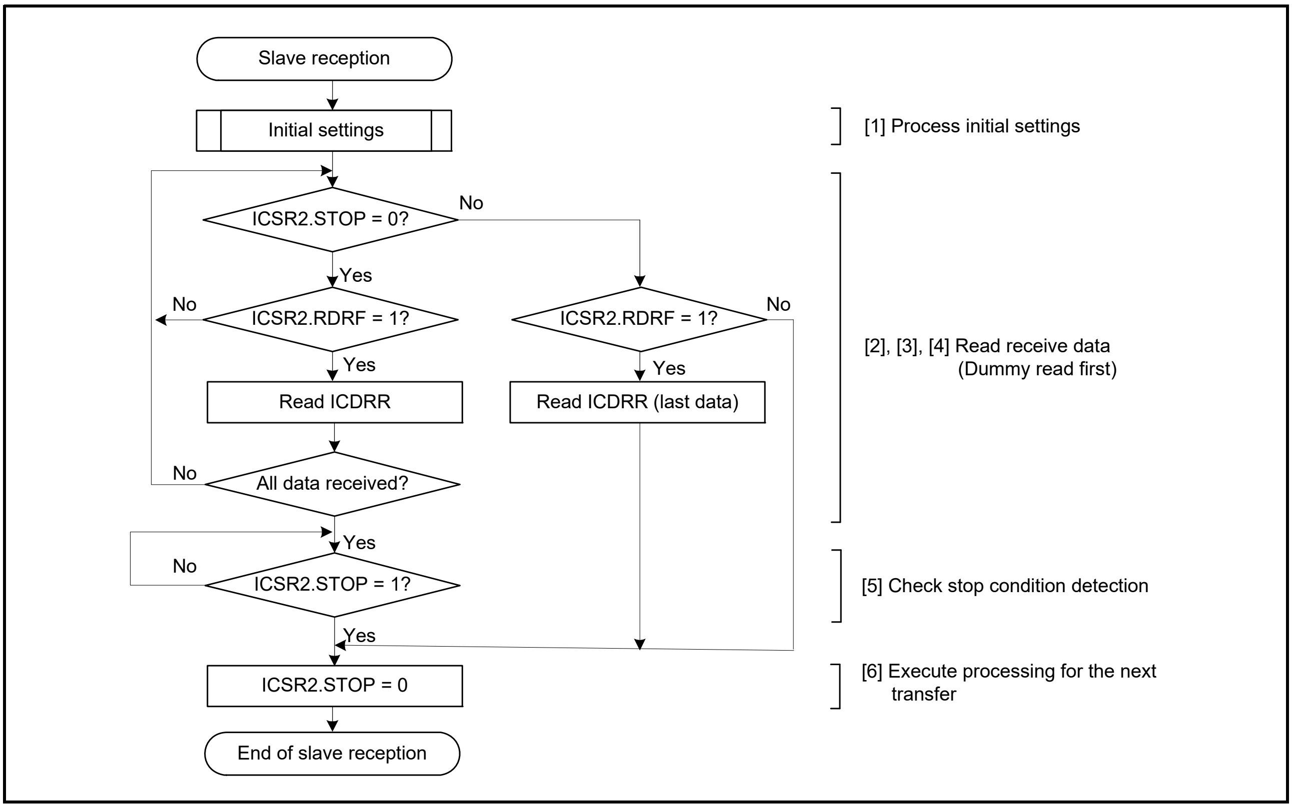


Figure 36.18 Example slave reception flow

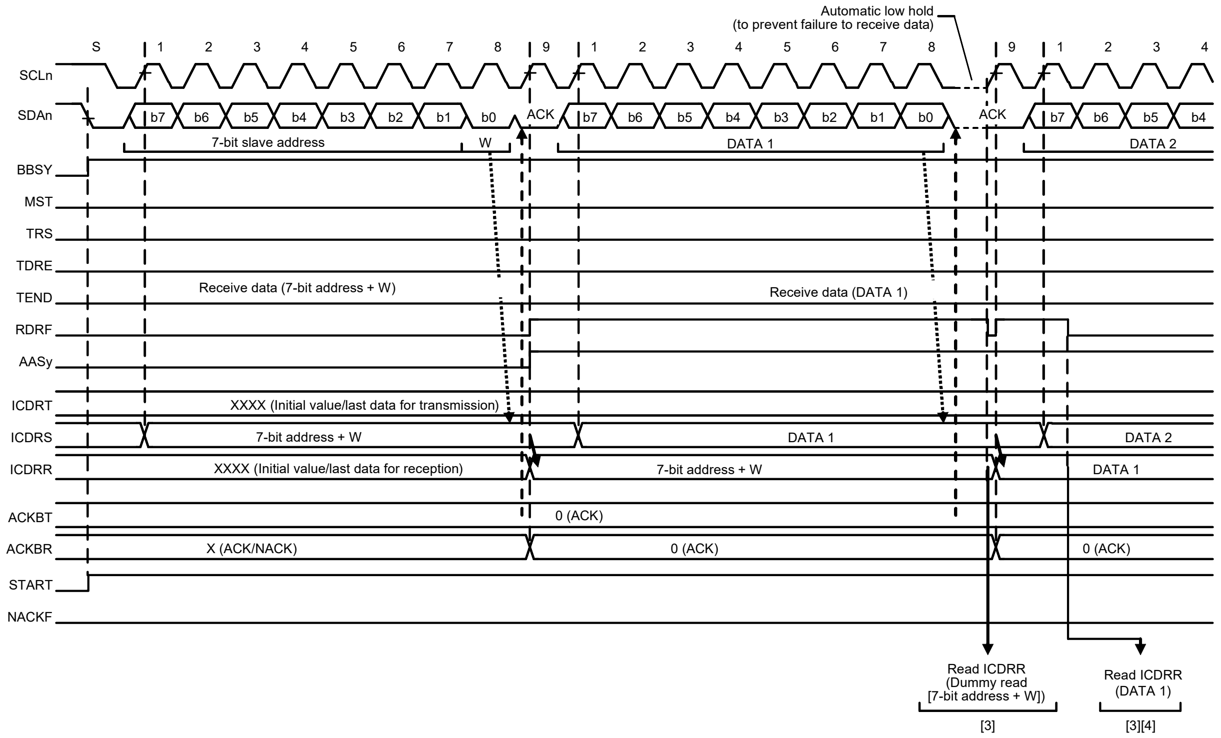


Figure 36.19 Slave receive operation timing (1) with 7-bit address format, when RDRFS = 0

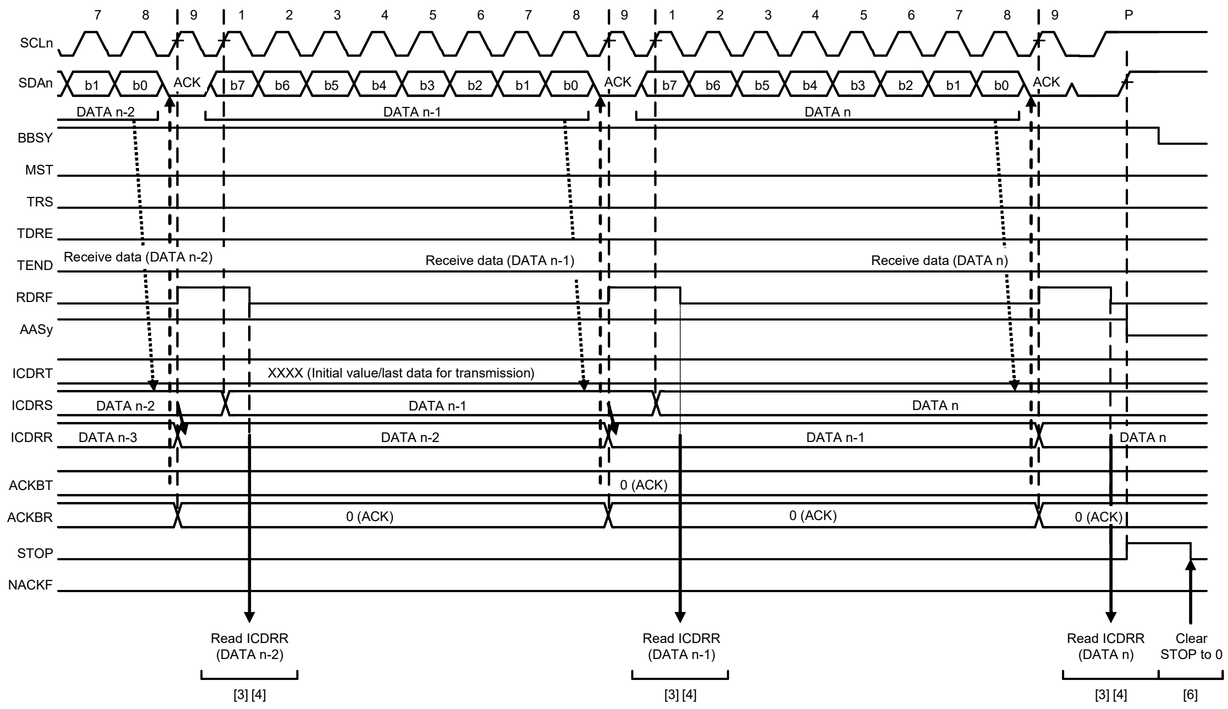


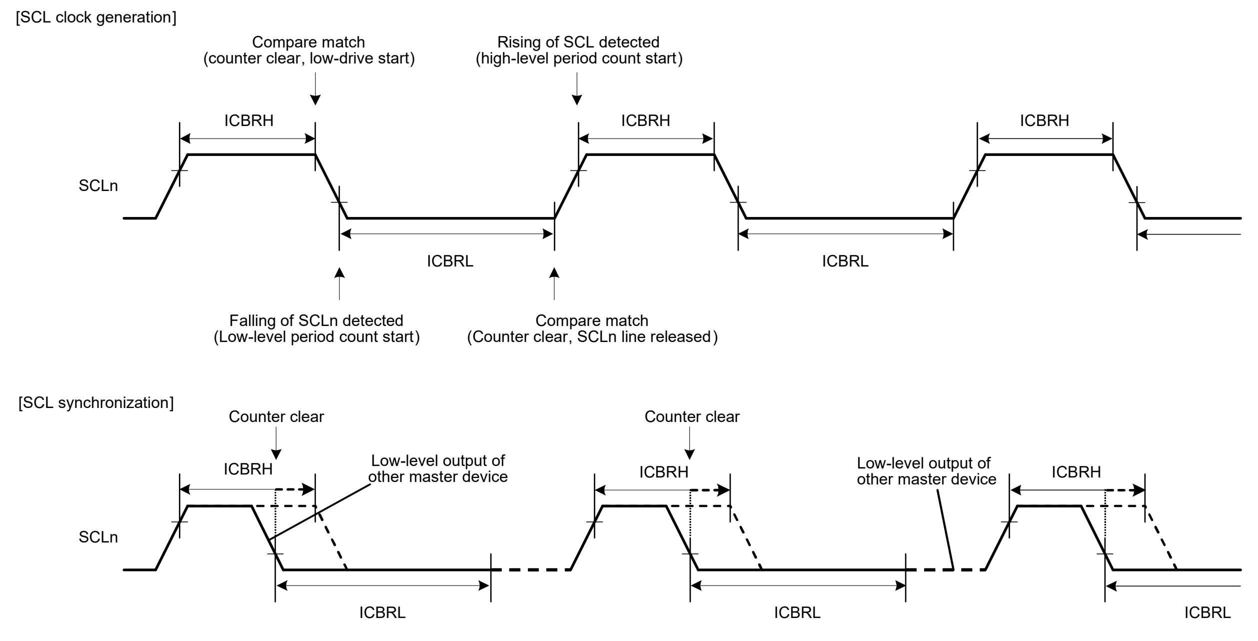
Figure 36.20 Slave receive operation timing (2) when RDRFS = 0

36.4 SCL Synchronization Circuit

For generation of the SCL clock, the IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and it drives the SCLn line low when it completes counting. When the IIC detects the falling edge of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then it stops driving the SCLn line, releasing the line, when it completes counting. The IIC repeats this process to generate the SCL clock.

If multiple master devices are connected to the I2C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit by bit, the IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the IIC detects a rising edge on the SCLn line and so starts counting out the high-level period specified in ICBRH, and the level on the SCLn line falls because an SCL signal is being generated by another master device, the IIC stops counting when it detects the falling edge, drives the level on the SCLn line low, and starts counting the low-level period specified in ICBRL. When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. At this time, if the low-level period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device has ended, the SCL signal rises because the SCLn line was released. When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. This synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.



ICBRH: IIC-Bus Bit Rate High-Level Register (SCL clock high-level period counter)
ICBRL: IIC-Bus Bit Rate Low-Level Register (SCL clock low-level period counter)

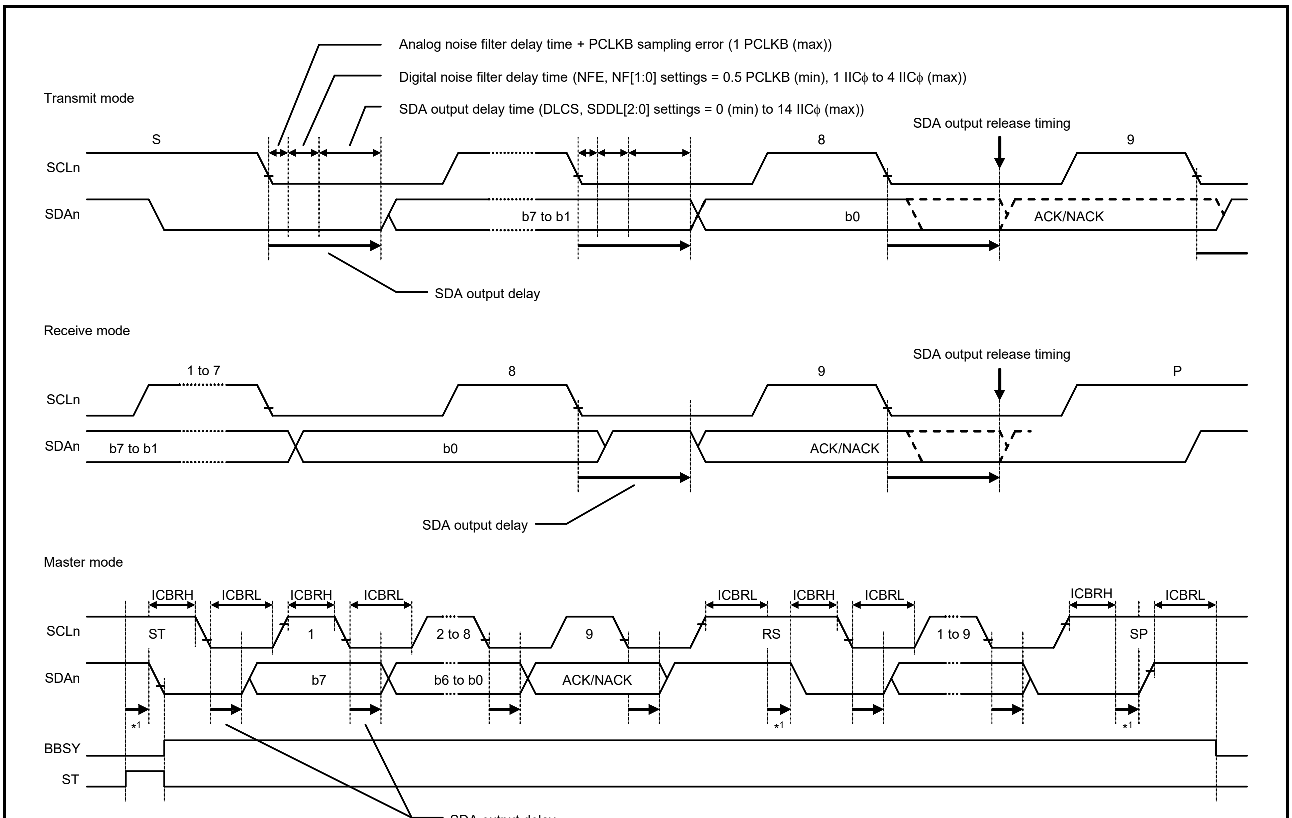
Figure 36.21 Generation and synchronization of SCL signal from IIC

36.5 SDA Output Delay Function

The IIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output on the SDA line, including issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300-ns minimum data-hold time requirement of the SMBus specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay function is enabled, for example, while the SDDL[2:0] bits in ICMR2 are set to any value other than 000b, the DLCS bit in ICMR2 selects the clock source for counting by the SDA output delay counter either as the internal base clock (IICϕ) for the IIC module or as the internal base clock divided by 2 (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. After the delay cycles count is complete, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note 1. The output delay function is set by the DLCS and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 36.22 SDA output delay function

36.6 Digital Noise Filter Circuits

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through analog and digital noise-filter circuits. Figure 36.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series and a match-detection circuit. The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for example, if data transfer is 400 kbps with PCLKB = 4 MHz, the characteristics of the digital noise filter might lead to the elimination of required signals as noise. In such cases, it is possible to disable the digital noise-filter circuit, by setting the ICFER.NFE bit to 0, and use only the analog noise filter circuit.

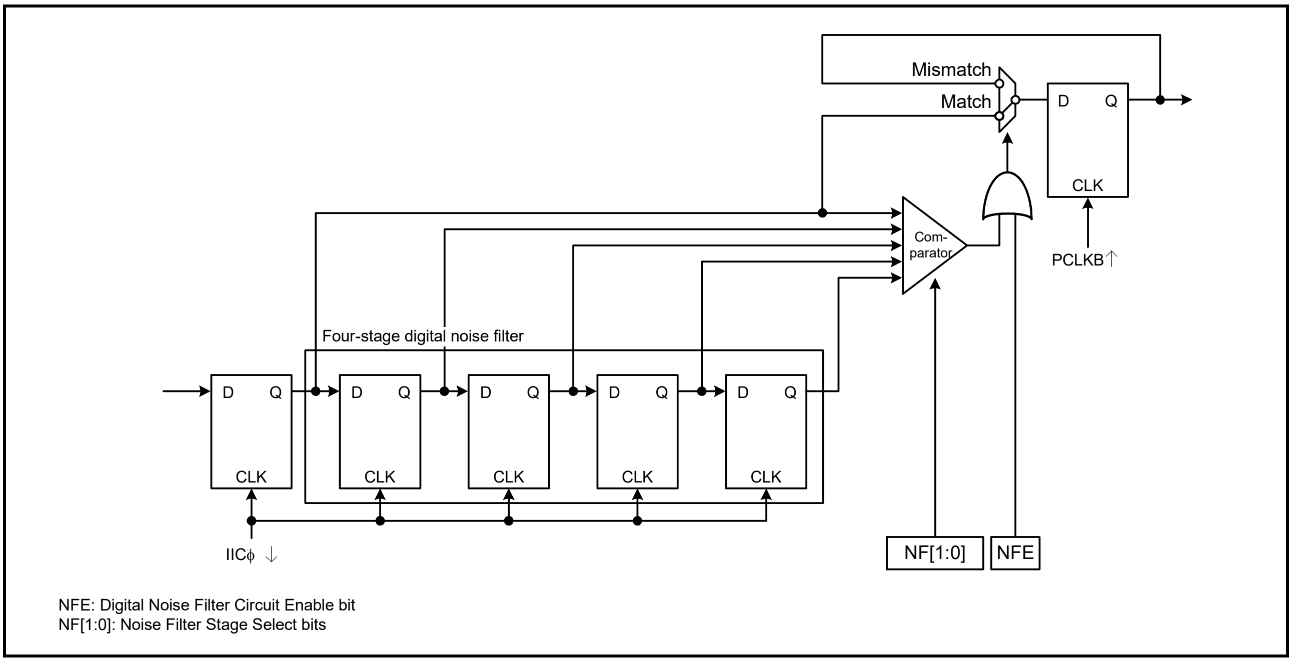


Figure 36.23 Digital noise filter circuit block diagram

36.7 Address Match Detection

The IIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7- or 10-bit slave addresses.

36.7.1 Slave-Address Match Detection

The IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

When the IIC detects a match of the set slave address, the associated AASy flag (y = 0 to 2) in ICSR1 is set to 1 on the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IICn_RXI) or transmit data empty interrupt (IICn_TXI) to be generated. The AASy flag identifies which slave address is specified.

Figure 36.24 to Figure 36.26 show the AASy flag set timing in three cases.

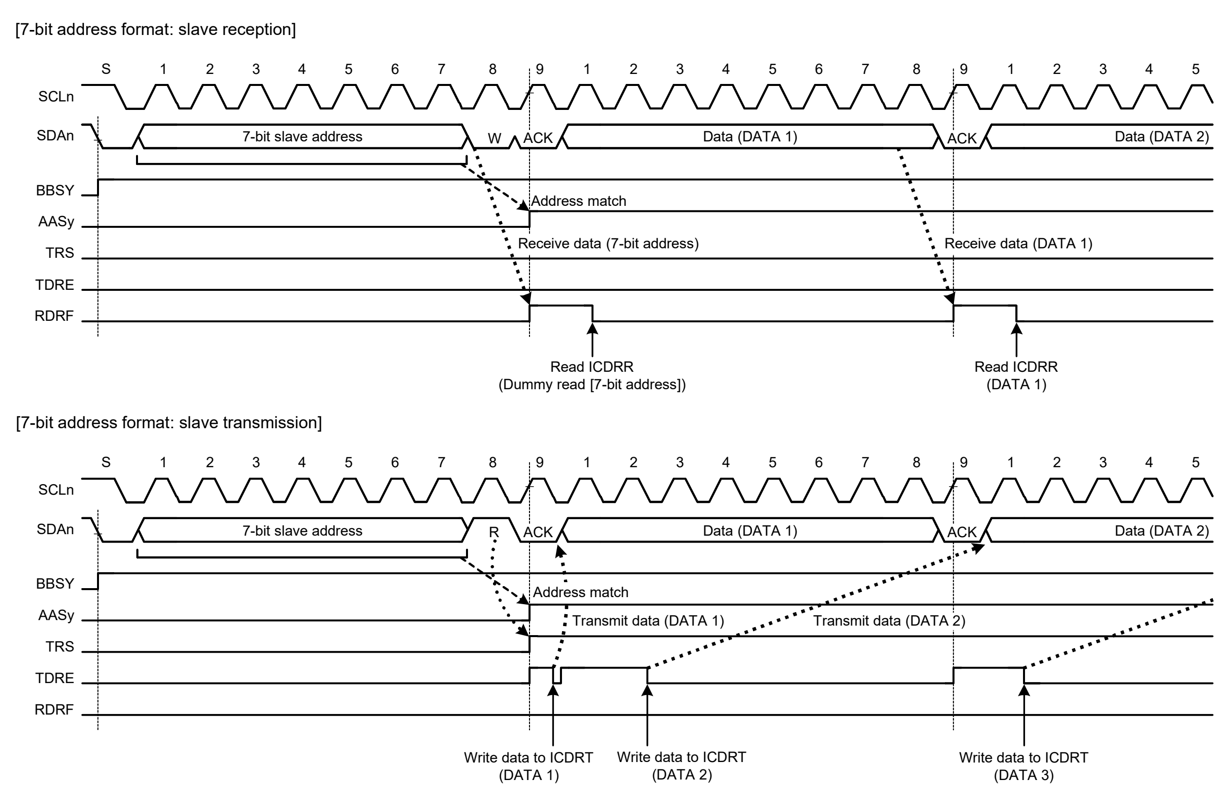


Figure 36.24 AASy flag set timing with 7-bit address format

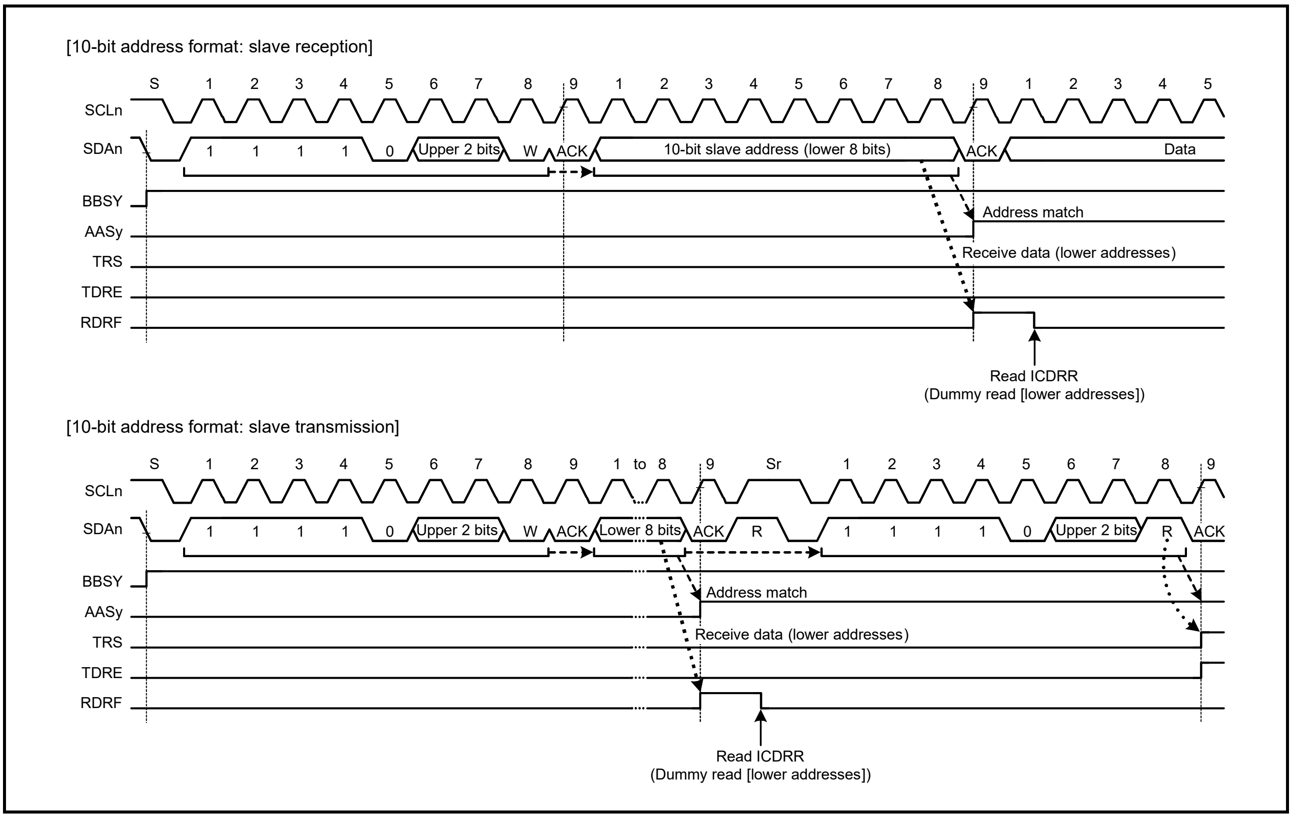


Figure 36.25 AASy flag set timing with 10-bit address format

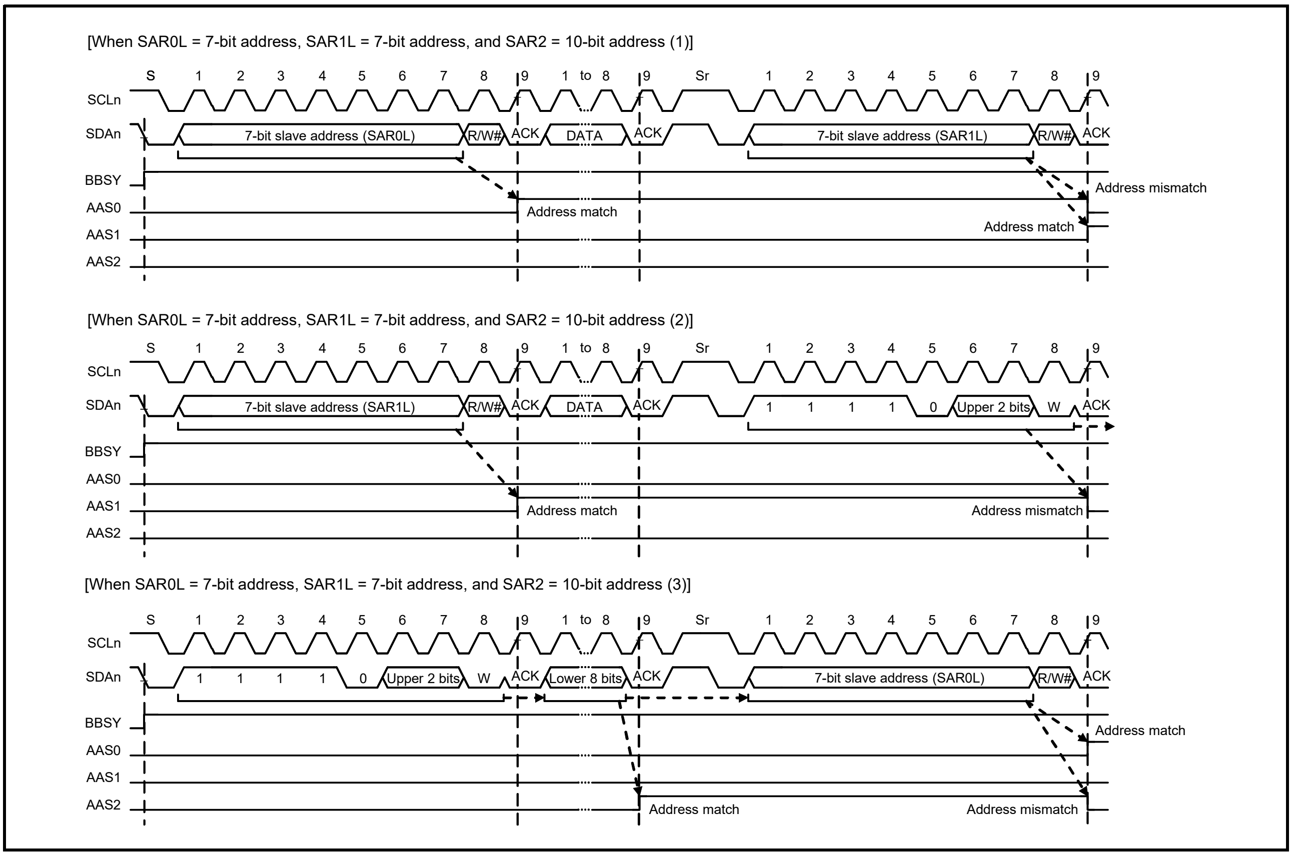


Figure 36.26 AASy flag set and clear timing with 7-bit and 10-bit address formats mixed

36.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the ninth cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address was transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

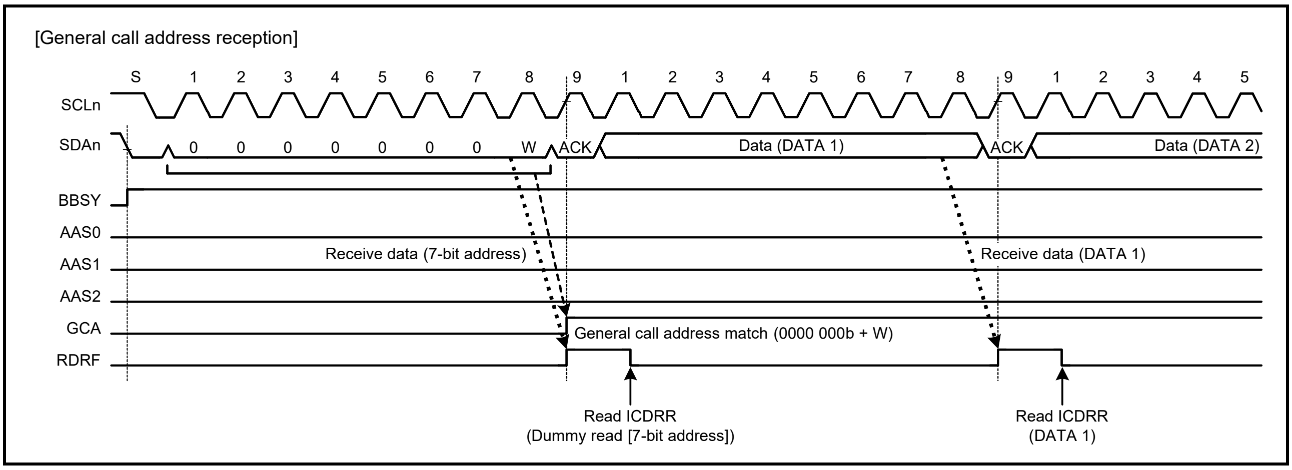


Figure 36.27 Timing of GCA flag setting during reception of general call address

36.7.3 Device-ID Address Detection

The IIC module provides detection of device-ID address compliant with the I2C bus specification (revision 03). When the IIC receives 1111 100b as the first byte after a start or restart condition was issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the eighth SCL clock cycle when the subsequent R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy flag (y = 0 to 2) in ICSR1 to 1.

After that, when the first byte received after issuance of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, the IIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device-ID address detection function, the IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained or a match with the device ID address is not obtained after a match with the IIC slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device-ID address (1111 100b), and the R/W# bit is 0, the IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of the IIC. If the R/W# bit is 1, the DID flag holds the previous value and the IIC does not compare the second and subsequent bytes. In this way, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal transmit data. For details on the information that must be included in device-ID fields, contact NXP Semiconductors.

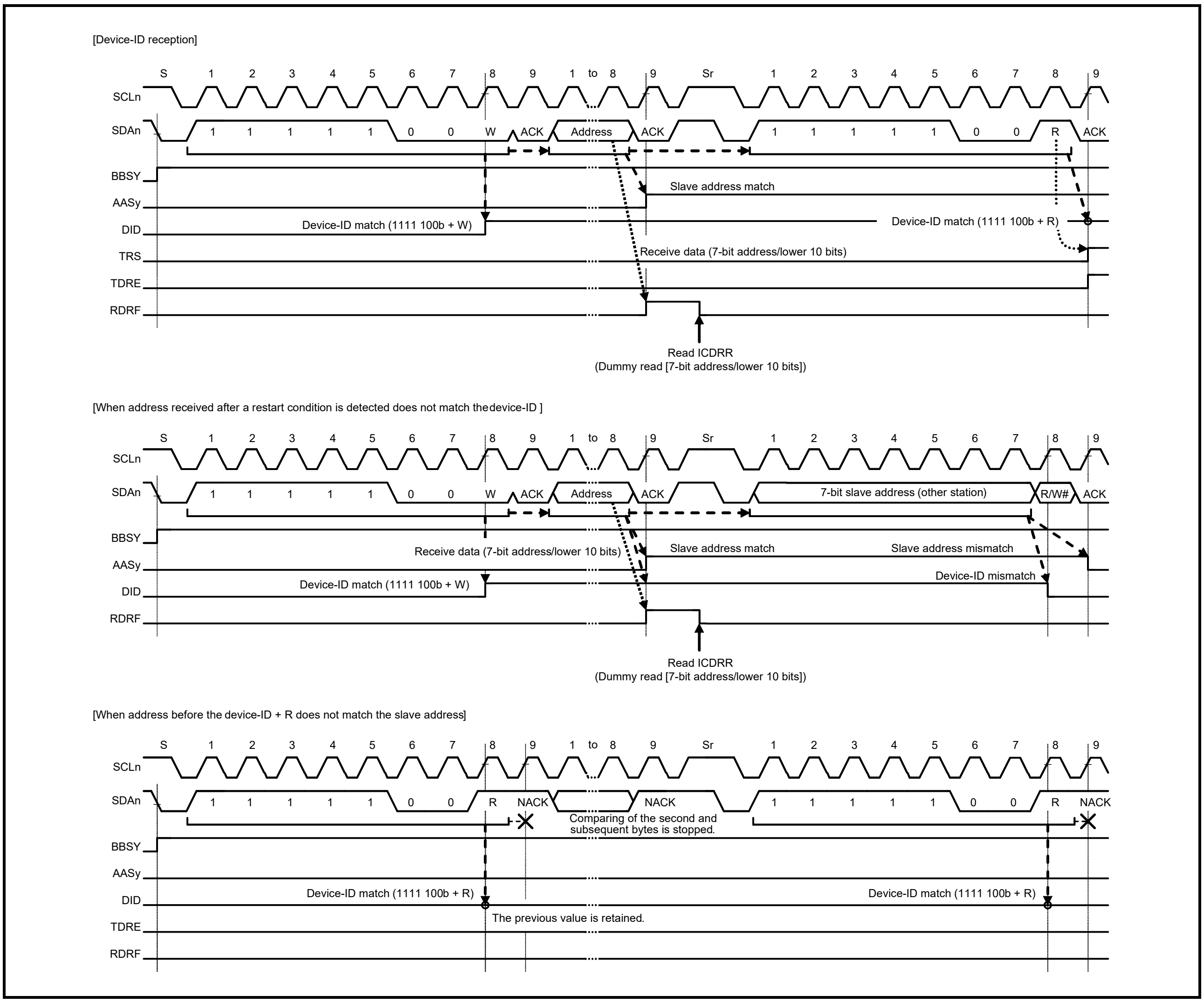


Figure 36.28 AASy and DID flag set and clear timing during reception of device-ID

36.7.4 Host Address Detection

The IIC provides host address detection while the SMBus is operating. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the ninth SCL clock cycle, and at the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.

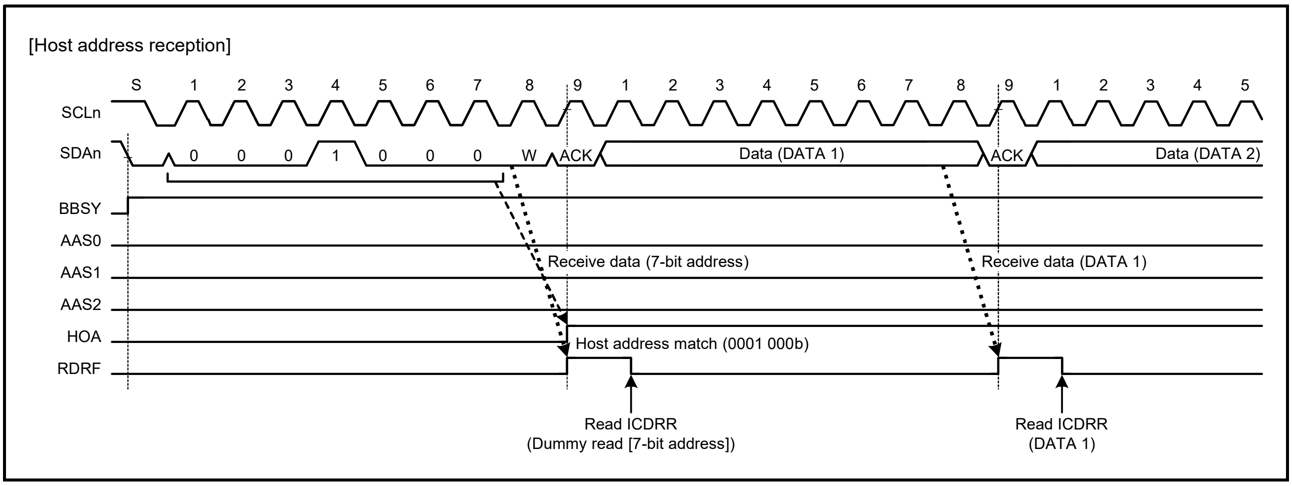


Figure 36.29 HOA flag set timing during reception of host address

36.8 Wakeup Function

The IIC provides a wakeup function that causes the MCU to transition from Software Standby mode to normal operation. The wakeup function enables the reception of data when the system clock is stopped, and it generates a wakeup interrupt signal on the match of the slave address of the received data. This interrupt signal triggers the return to normal operation.

The wakeup function has four operation modes: normal wakeup mode 1, normal wakeup mode 2, command recovery mode, and EEP response mode. Table 36.9 describes the behavior in these modes.

Table 36.9 Wakeup operation modes

Operation mode	ACK response timing	ACK response before wakeup	SCL state during wakeup
Normal wakeup mode 1	Before wakeup	ACK	Fixed low
Normal wakeup mode 2	After wakeup	Before wakeup: no response After wakeup: ACK response	Fixed low
Command recovery mode	Before wakeup	ACK	Open
EEP response mode	Before wakeup	NACK	Open

Precautions on the use of the wakeup function

1. Disable the wakeup function (WUE = 0) after a wakeup interrupt triggers the transition from Software Standby mode to normal operation.
2. Do not change the content of the IIC registers while WUF = 0, even if the wakeup interrupt recovers the system clock. Make register settings after confirming that WUF = 1.
3. Set WUE = WUIE = 1 and MST = TRS = 0 (slave reception mode) before entering Software Standby mode.
4. Do not invoke Software Standby mode while BBSY = 1.
5. The wakeup function supports the 7-bit slave address of slave address register SARL0, the general call address, and the host address. 10-bit slave addresses, SARL1 and SARL2, are not supported. Do not use them.
6. When the wakeup function is enabled, disable the interrupt selectable in the ICIER bits (TIE, TEIE, RIE, NAKIE, SPIE, STIE, ALIE, and TMOIE).
7. When the wakeup function is enabled, do not use the timeout function.
8. If the transition from Software Standby mode is triggered by the interrupt (such as IRQn) other than a wakeup interrupt. WUF is not set in this case. Follow the processing shown in Figure 36.31 and Figure 36.36.

36.8.1 Normal Wakeup Mode 1

This section describes the behavior, the timing, and an example operation of normal wakeup mode 1.

1. A wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows. Figure 36.32 provides detailed timing and Figure 36.30 shows an example operation.

   Before wakeup:ACK is sent in response to the data received with the own slave address of the IIC.

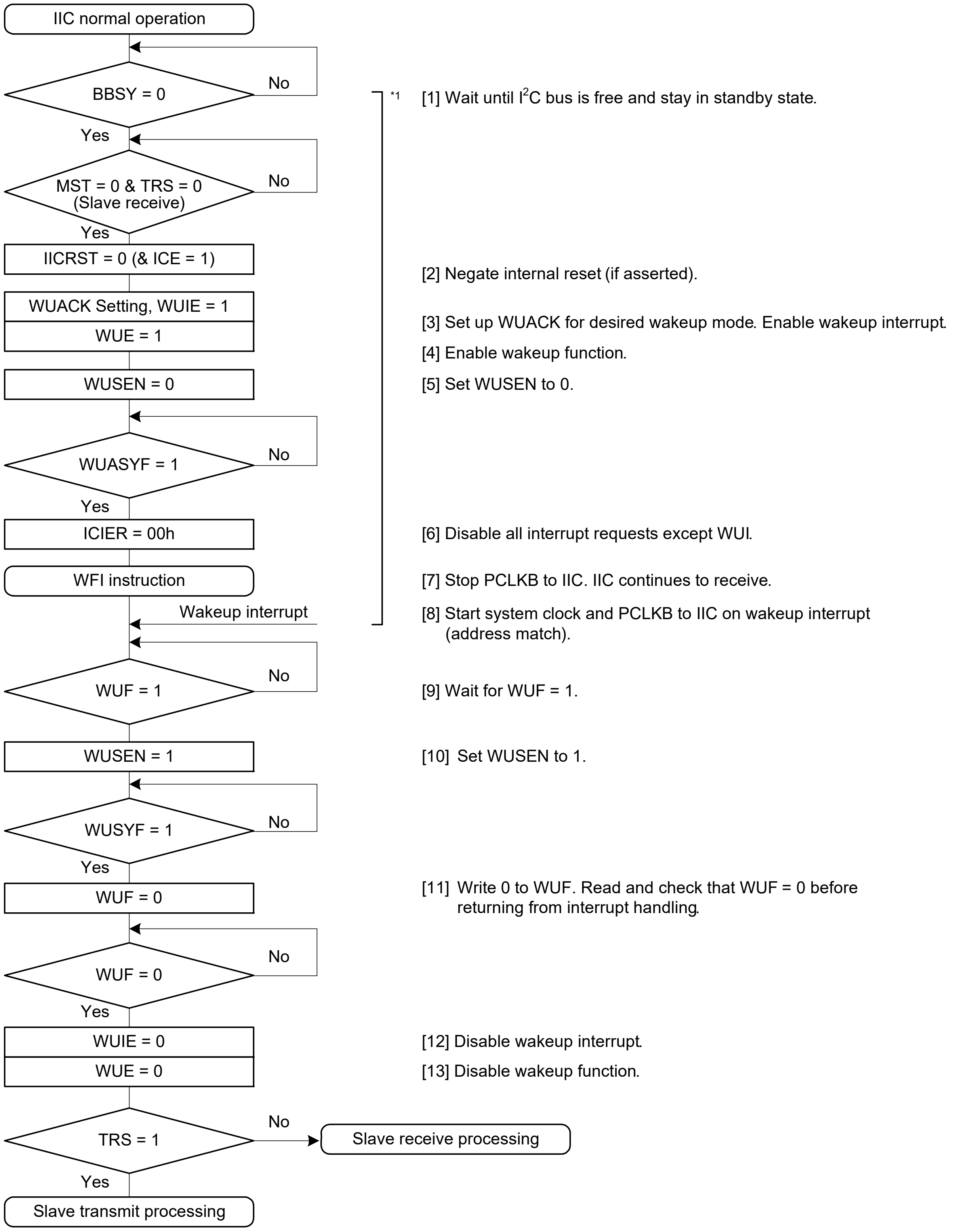
   During wakeup:ACK response is made on the ninth clock cycle of SCL, and SCL is held low afterwards.*1

   After wakeup:Normal operation continues.

   If the slave address does not match, the SCL line is not held low after the fall of the ninth clock cycle of SCL, and the slave operation continues.

Note 1. Between the ninth clock cycle and the first clock cycle during wakeup, WAIT = 1 is invalid.

2. If the transition from Software Standby mode is triggered by the interrupt (such as IRQn) other than a wakeup interrupt. WUF is not set in this case. Follow the processing shown in Figure 36.31.



[1] Wait until I^2C bus is free and stay in standby state.

[2] Negate internal reset (if asserted).

[3] Set up WUACK for desired wakeup mode. Enable wakeup interrupt.

[4] Enable wakeup function.

[5] Set WUSEN to 0.

[6] Disable all interrupt requests except WUI.

[7] Stop PCLKB to IIC. IIC continues to receive.

[8] Start system clock and PCLKB to IIC on wakeup interrupt (address match).

[9] Wait for WUF = 1.

[10] Set WUSEN to 1.

[11] Write 0 to WUF. Read and check that WUF = 0 before returning from interrupt handling.

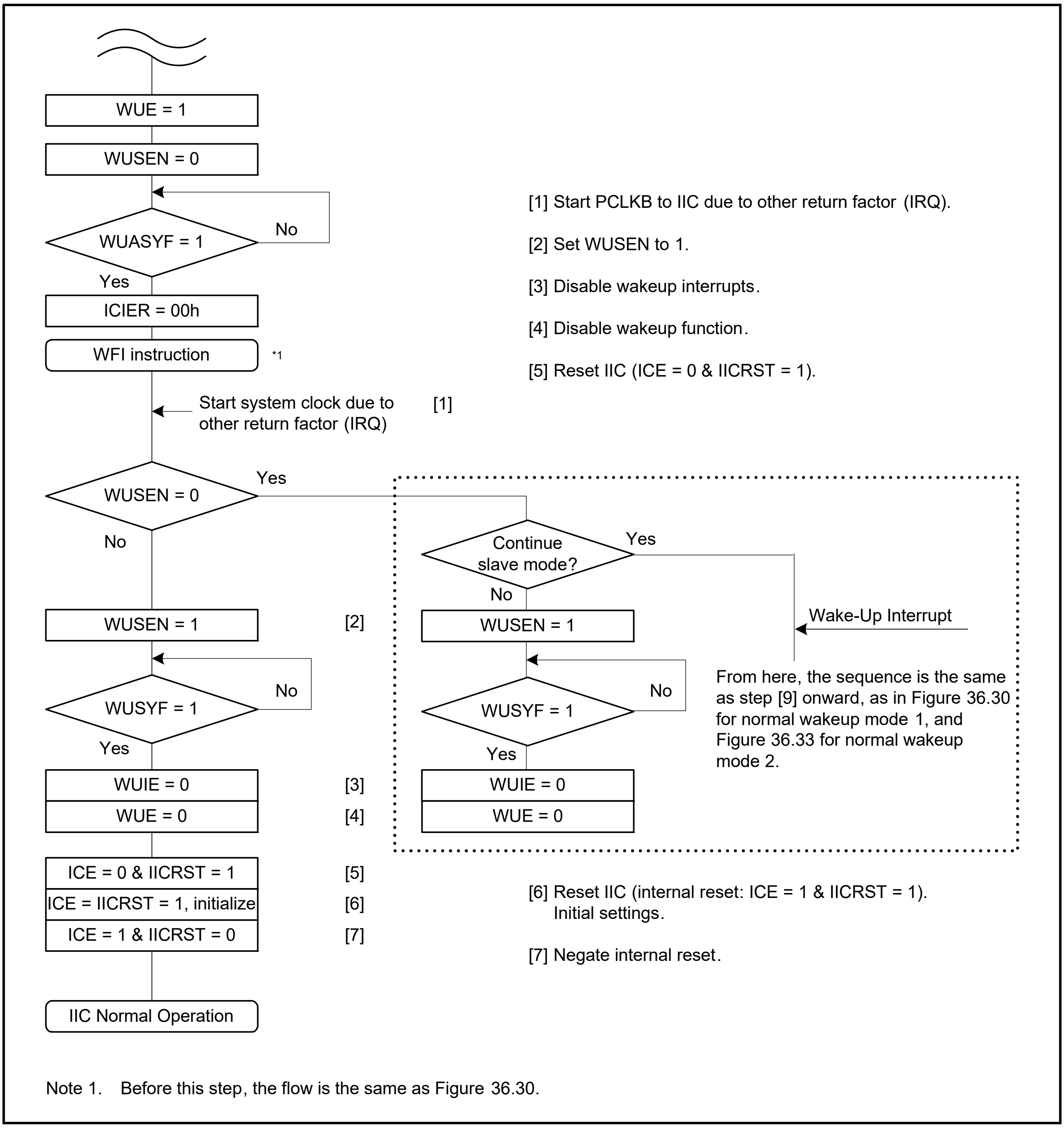
[12] Disable wakeup interrupt.

[13] Disable wakeup function.

Note 1. Do not issue start condition between BBSY = 0 and executing WFI instruction.

Figure 36.30 Example operation of normal wakeup mode 1 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



[1] Start PCLKB to IIC due to other return factor (IRQ).

[2] Set WUSEN to 1.

[3] Disable wakeup interrupts.

[4] Disable wakeup function.

[5] Reset IIC (ICE = 0 & IICRST = 1).

From here, the sequence is the same as step [9] onward, as in Figure 36.30 for normal wakeup mode 1, and Figure 36.33 for normal wakeup mode 2.

[6] Reset IIC (internal reset: ICE = 1 & IICRST = 1).
Initial settings.

[7] Negate internal reset.

Note 1. Before this step, the flow is the same as Figure 36.30.

Figure 36.31 Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example IRQn

Note: For details on the IIC initial settings, see section 36.3.2, Initial Settings.

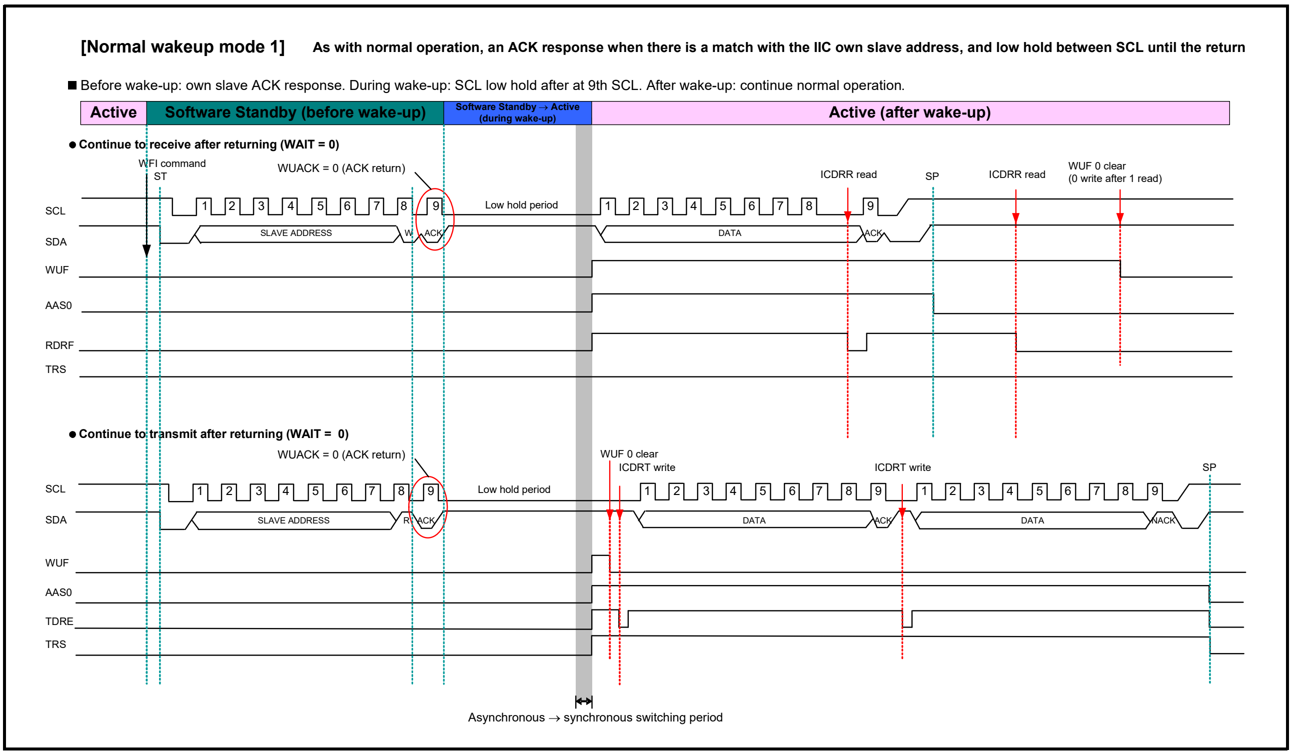


Figure 36.32 Timing of normal wakeup mode 1

36.8.2 Normal Wakeup Mode 2

This section describes the behavior, the timing, and an example operation of normal wakeup mode 2.

1. A wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows. Figure 36.34 provides detailed timing and Figure 36.33 shows an example operation.

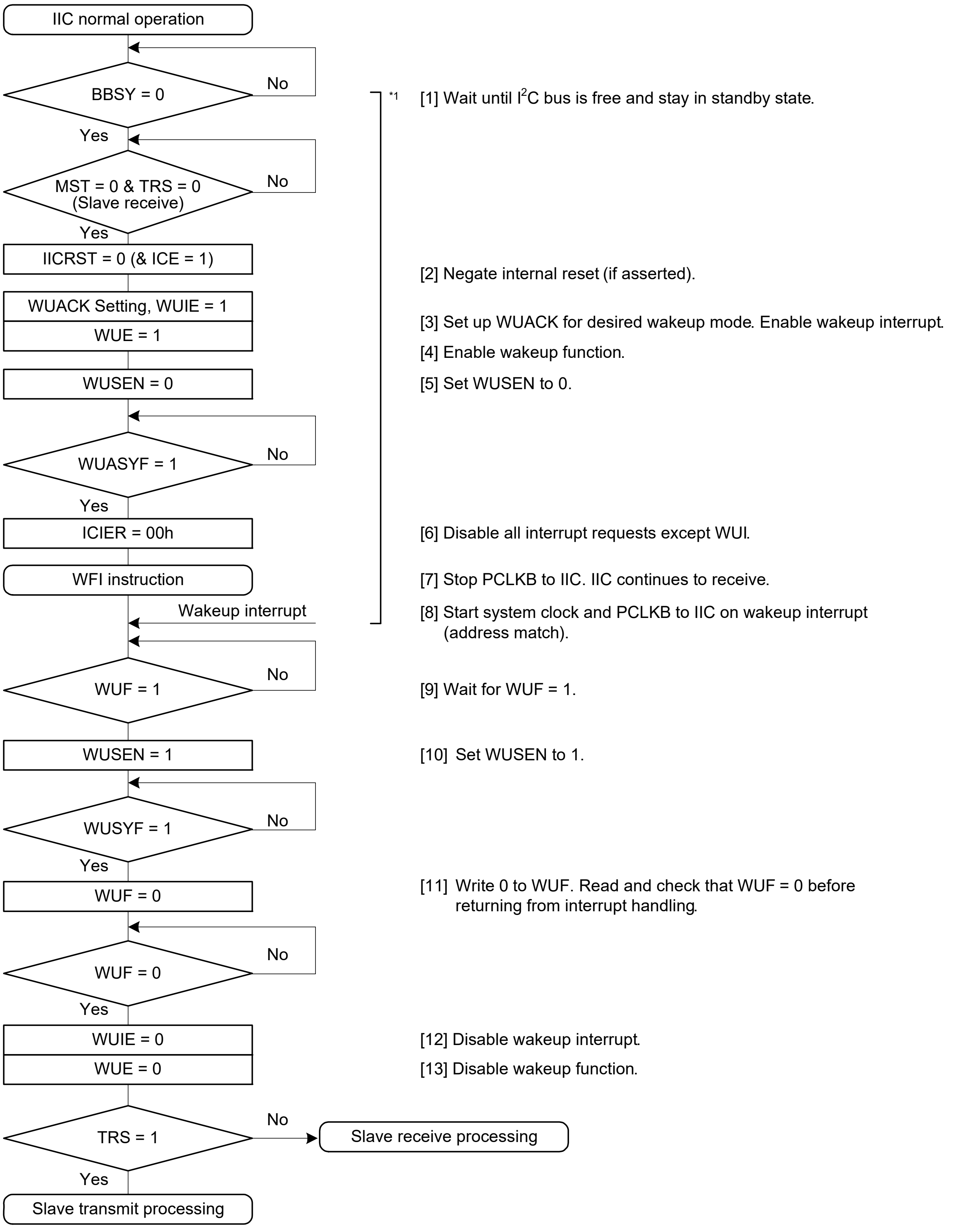
   Before wakeup:No response to the data received with the own slave address of the IIC until the end of the eighth SCL cycle.

   During wakeup:SCL line held low during the eighth and ninth clock cycles.

   After wakeup:ACK returns on the ninth clock cycle of SCL, and normal operation continues.

   If the slave address does not match, the SCL line is not held low after the fall of the eighth SCL clock cycle. The slave operation continues.

2. If the transition from Software Standby mode is triggered by the interrupt (such as IRQn) other than a wakeup interrupt. WUF is not set in this case. Follow the processing shown in Figure 36.31.



Note 1. Do not issue start condition between BBSY = 0 and executing WFI instruction.

Figure 36.33 Example operation of normal wakeup mode 2 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.

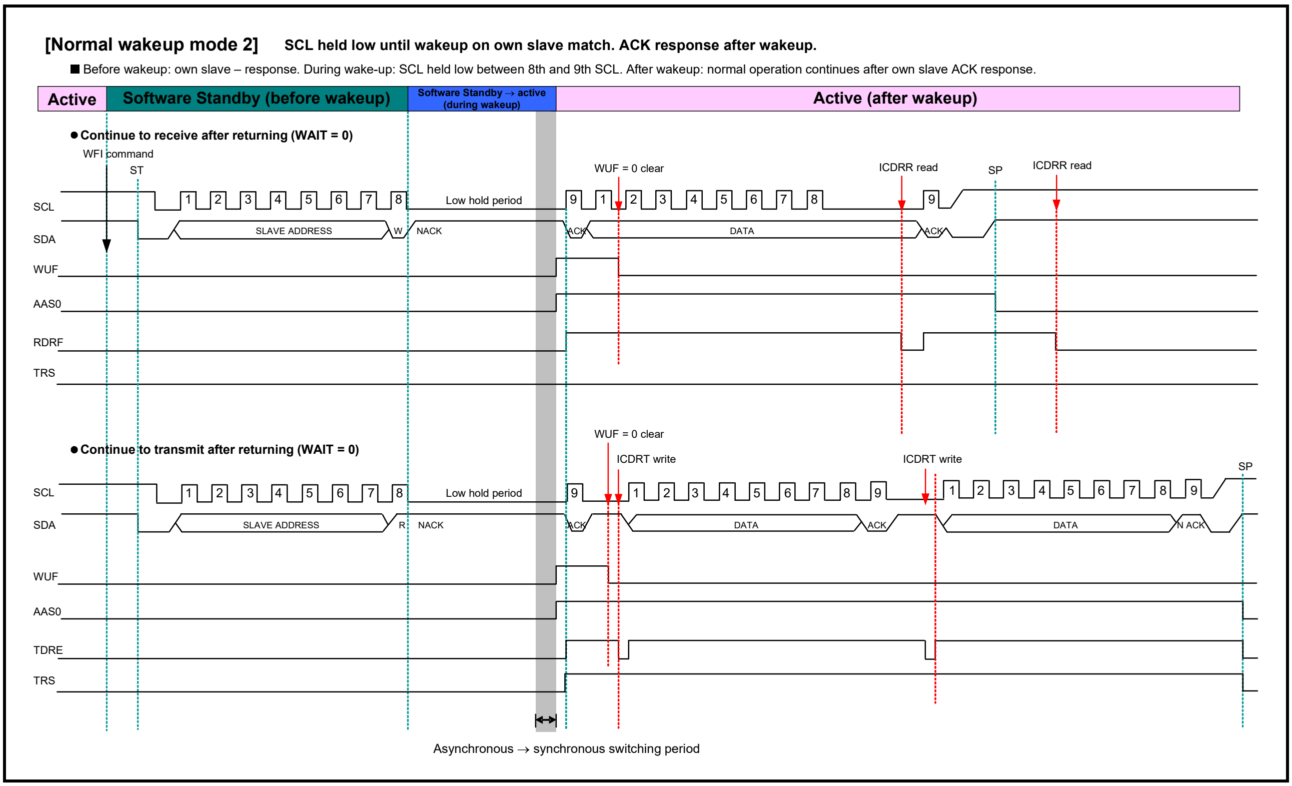


Figure 36.34 Timing of normal wakeup mode 2

36.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes)

In the command recovery and EEP response modes, the SCL line is not held low during the wakeup period (after the rise of the ninth clock cycle of SCL), so other IIC devices can use the I2C bus during this period.

This section describes the behavior, the timing, and example operations of the command recovery and EEP response modes.

1. A wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows. Figure 36.37 provides detailed timing and Figure 36.35 shows an example operation.

   Before wakeup:In response to the data received with the own slave address of the IIC, ACK (command recovery mode) or NACK (EEP response mode) is returned.

   During wakeup:The SCL line is not held low.

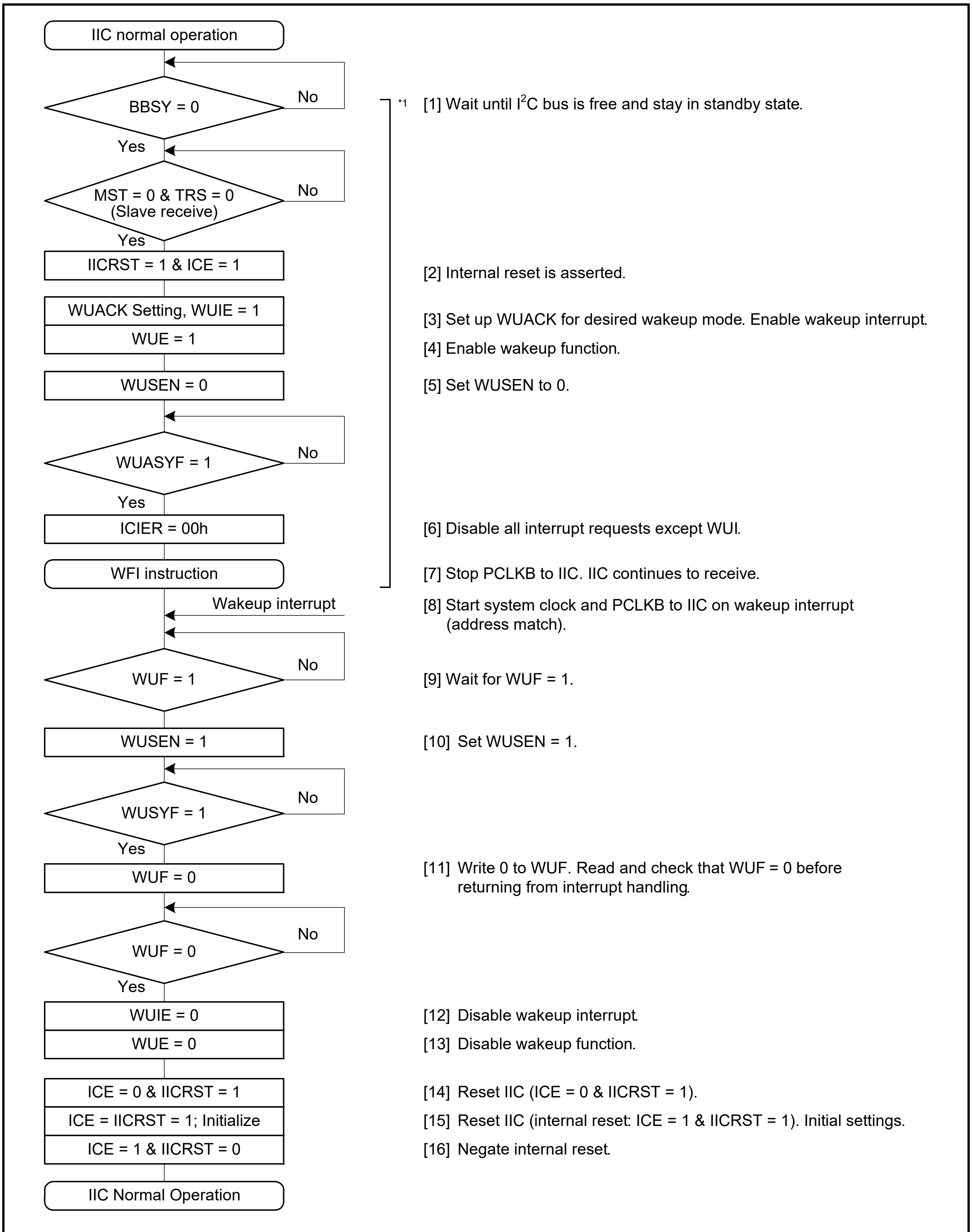
   After wakeup:Normal operation continues after the IIC initial settings.

   If the slave address does not match, the slave operation continues.

Note 1. Because the SCL line is not held low during wakeup, transmission or reception of the data that follows the slave address is not possible.

Note 2. The command recovery and EEP response modes are internal reset states (ICE = IICRST = 1). Therefore, the match of the slave address does not set the ICSR1 flags, HOA, GCA, and ASS0,ASS1,ASS2.

2. If the transition from Software Standby mode is triggered by the interrupt (such as IRQn) other than a wakeup interrupt. WUF is not set in this case. Follow the processing shown in Figure 36.36.



Note 1. Do not issue start condition between BBSY = 0 and executing WFI instruction.

Figure 36.35 Example operation of command recovery mode and EEP response mode when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.

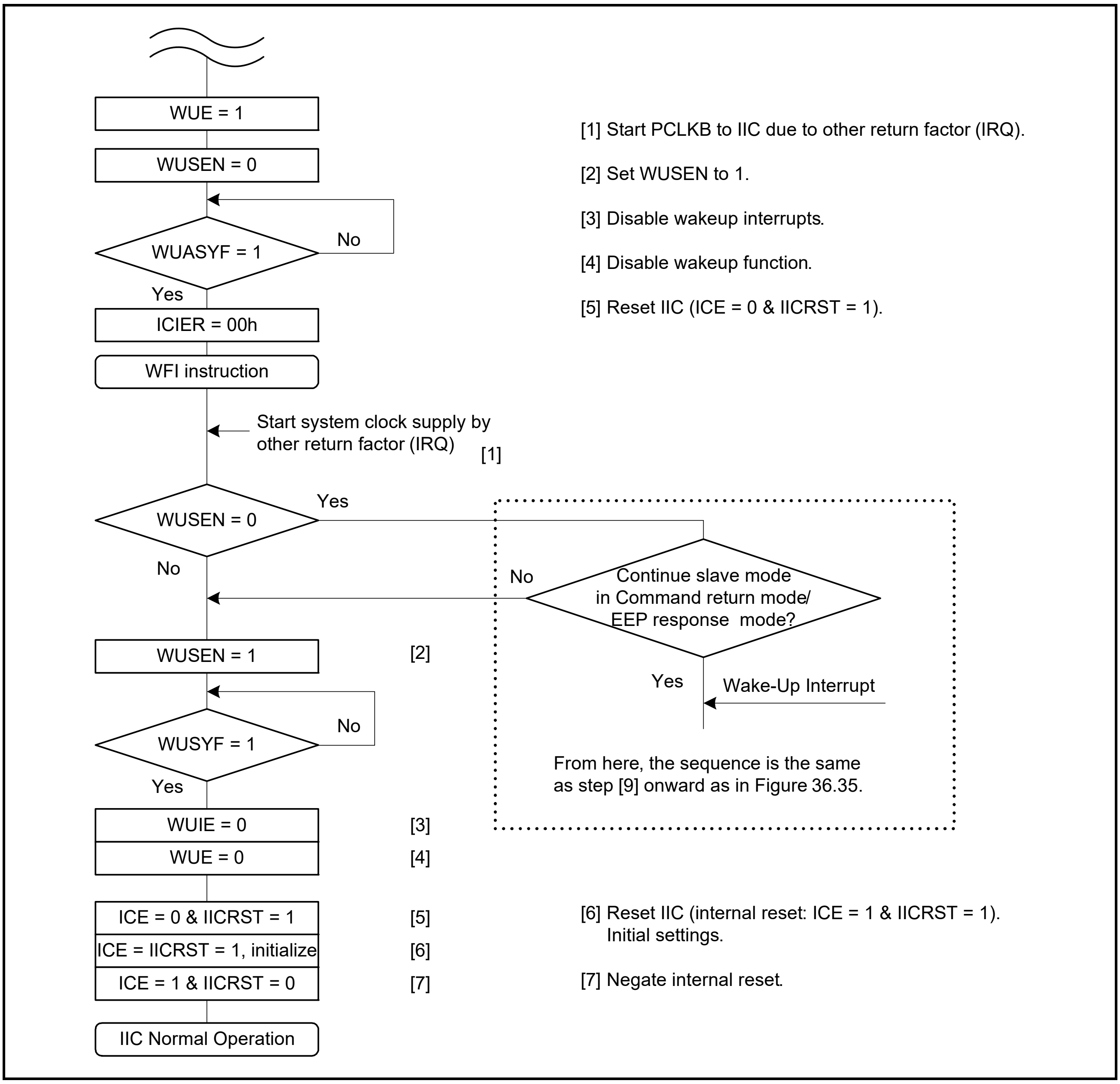


Figure 36.36 **Example operation of command recovery mode and EEP response mode when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example IRQn**

Note: For details on the IIC initial settings, see section 36.3.2, Initial Settings.

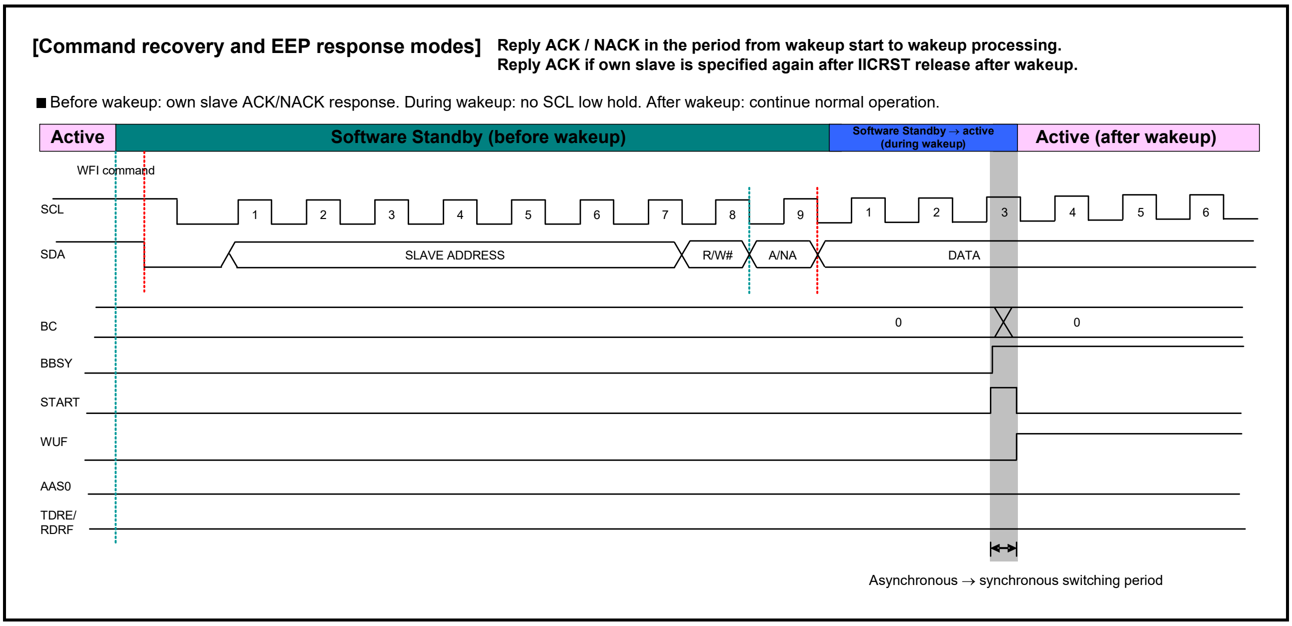


Figure 36.37 Timing of command recovery and EEP response modes

36.8.4 Precautions for WFI instruction Execution

In the example operations for the wakeup mode shown in Figure 36.30, Figure 36.33, and Figure 36.35, make sure that the start condition is not issued during the period from the setting of BBSY = 0 to the execution of the WFI instruction. When a start condition is issued during this period, NACK is returned after the reception of the first byte of the first data block. Then the detection of the start or restart condition enables the wakeup function.

36.9 Automatic Low-Hold Function for SCL

36.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I²C Bus Shift Register (ICDRS) is empty and data has not been written to the IIC-Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a start or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next.

Slave transmit mode

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next.

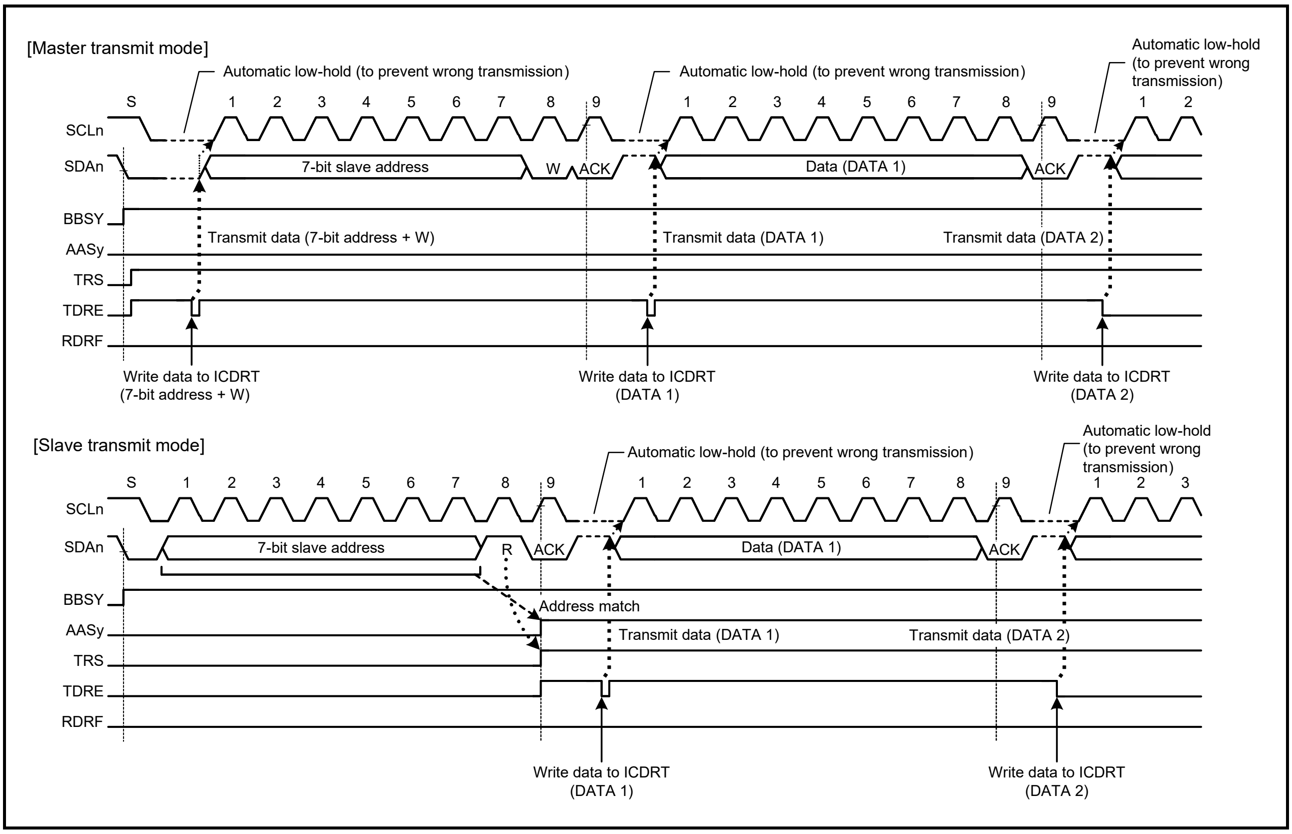


Figure 36.38 Automatic low-hold operation in transmit mode

36.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). It is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operation, you must set the NACKF flag to 0. In master transmit mode, after issue a restart or stop condition, set the NACKF flag to 0, and then issue a start condition again.

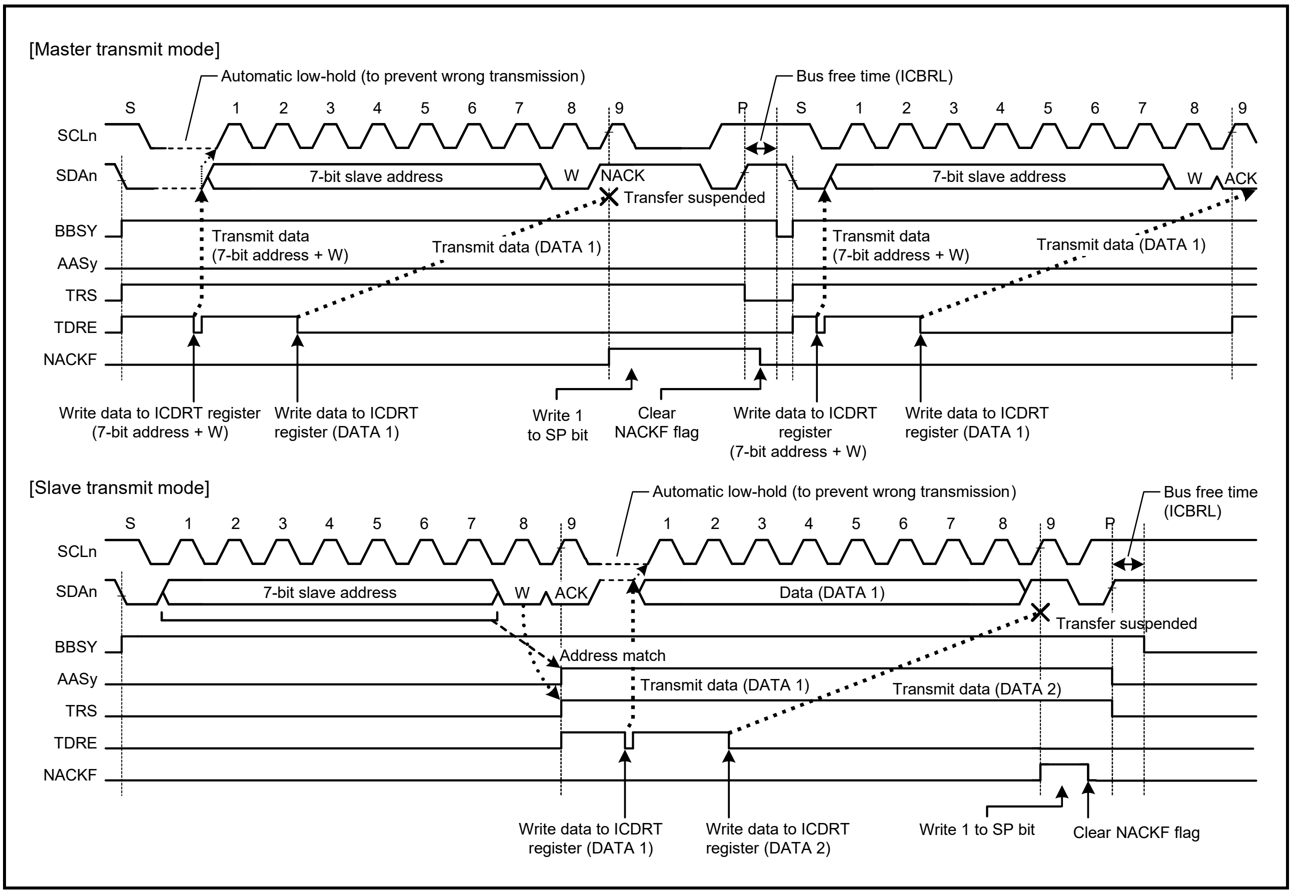


Figure 36.39 Suspension of data transfer when NACK is received, when NACKE = 1

36.9.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent a failure to receive data.

This function is enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, the IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

(2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag in ICSR2 is set to 1 (receive data full) on the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

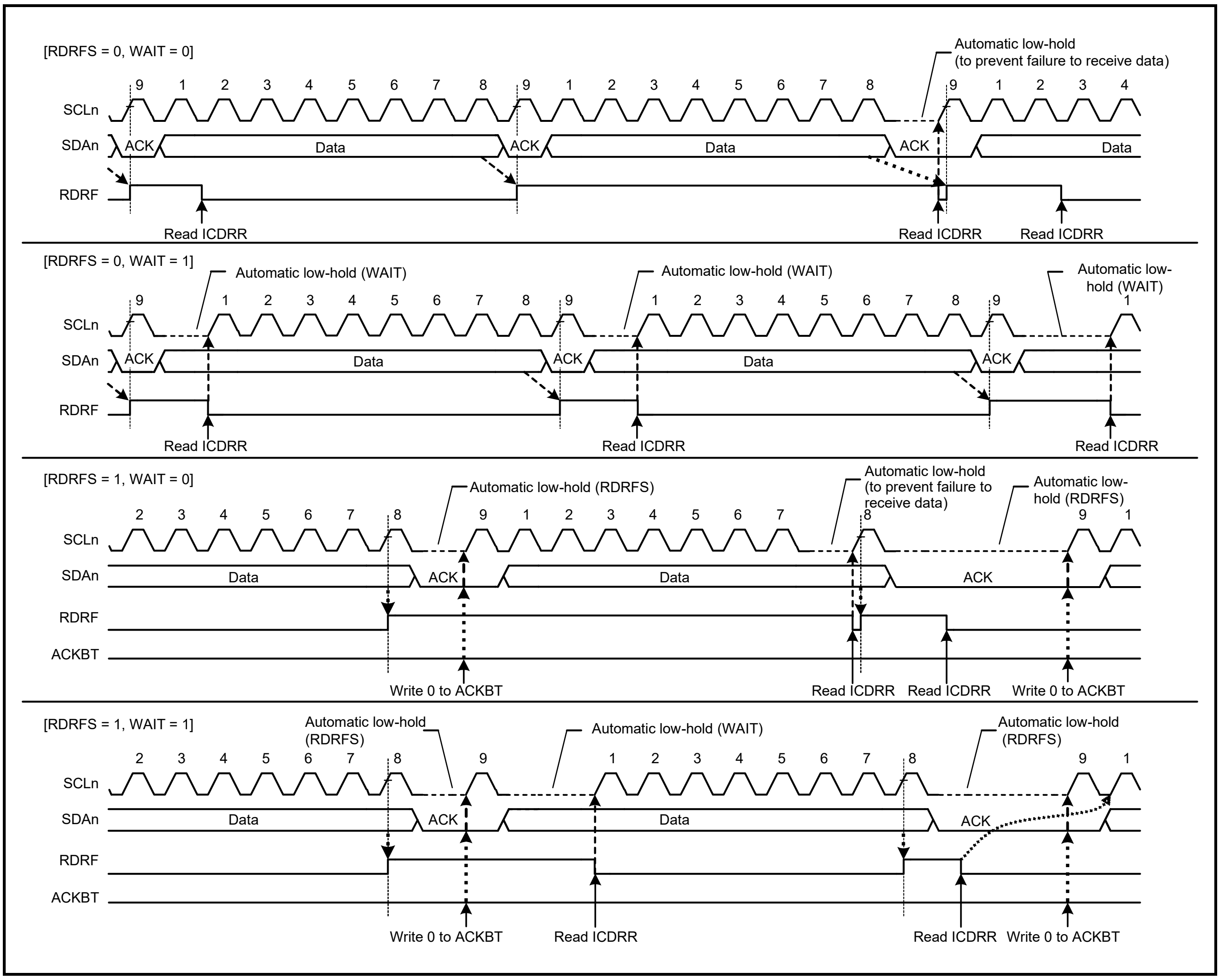


Figure 36.40 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

36.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^{2}C bus standard, the IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during transmission of NACK, and detect arbitration-lost in slave transmit mode.

36.10.1 Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, the IIC regards its own start condition as an error and considers this a loss in arbitration. Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration. This prevents a failure of transfer resulting from a start condition being issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and a low level is detected on the SDAn line, the IIC loses in arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (MST and TRS bits = 11b in ICCR2).

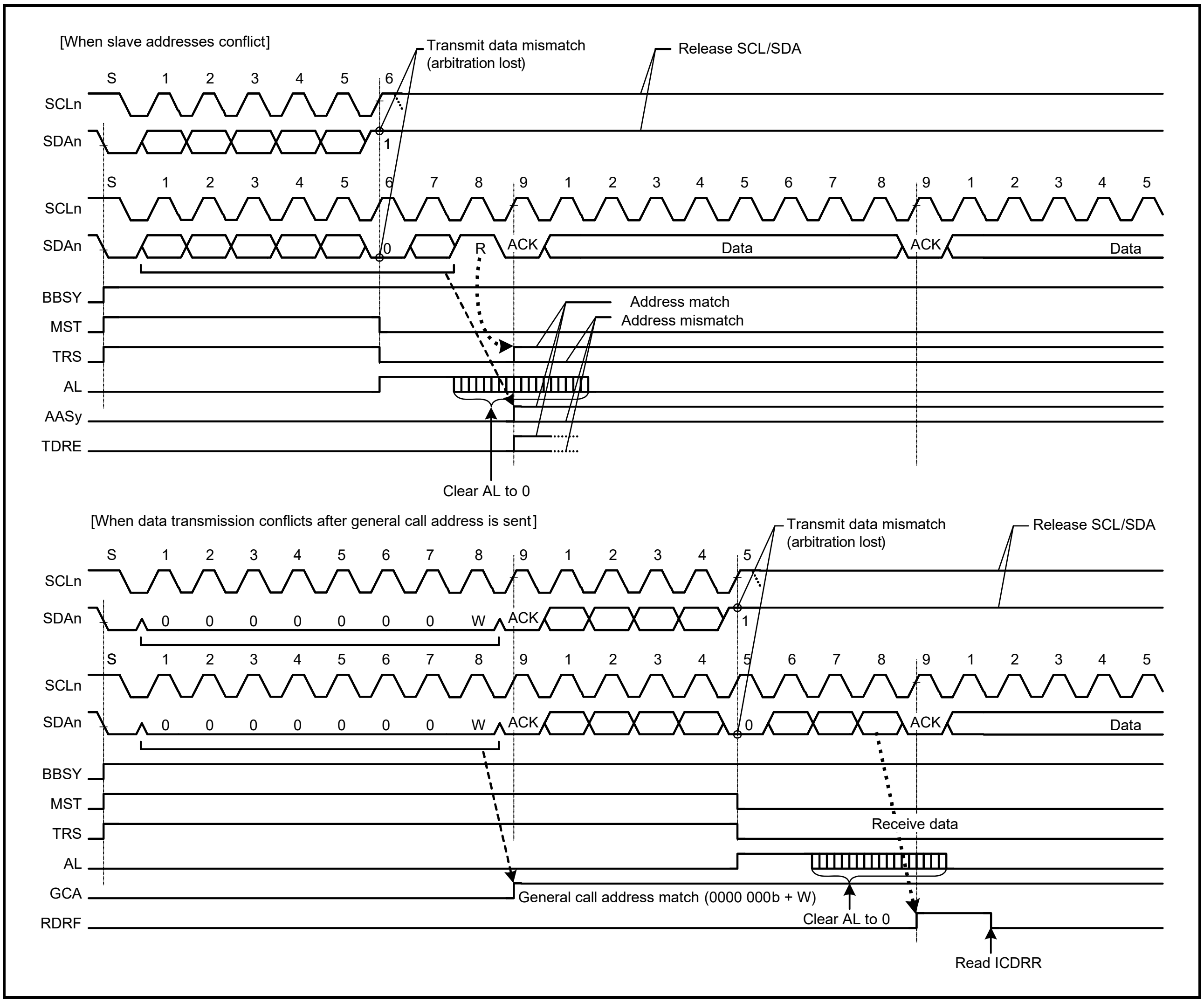


Figure 36.41 Examples of master arbitration-lost detection when MALE = 1

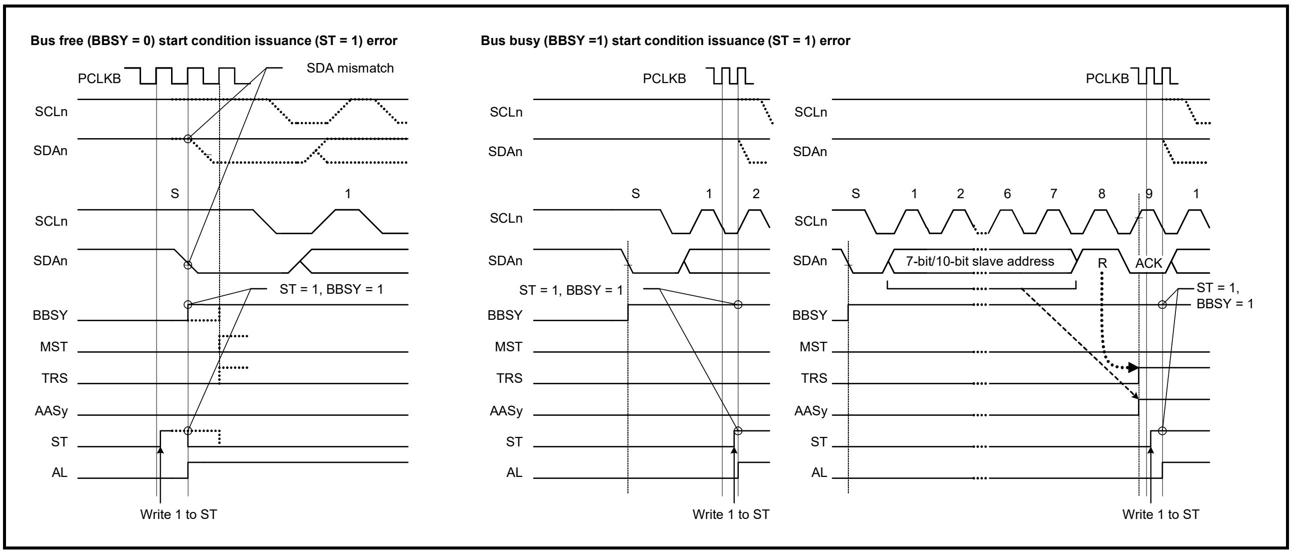


Figure 36.42 Arbitration-lost when start condition is issued when MALE = 1

36.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

This function causes arbitration to be lost if the internal SDA output level does not match the level on the SDAn line (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line during transmission of NACK in receive mode. Arbitration is lost because of a conflict of NACK and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 36.43 shows an example of arbitration-lost detection during transmission of NACK.

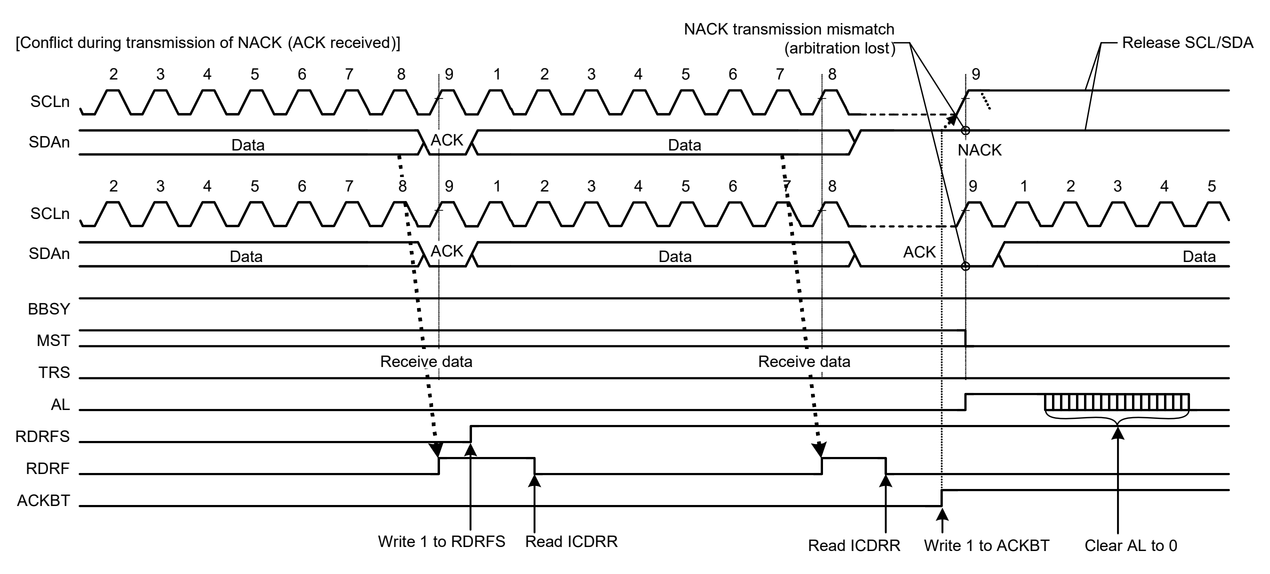


Figure 36.43 Example of arbitration-lost detection during transmission of NACK when NALE = 1

The following explains arbitration-lost detection using an example where two master devices (masters A and B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If masters A and B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or B during access to the slave device. Both masters A and B recognize that they obtained the bus mastership and operate as such. Here, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the necessary 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The issuance of the stop condition conflicts with the SCL clock output of master B, which disrupts communication.

When the IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, the IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as FFh transmission processing, necessary if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3)

36.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state), and the low level is detected on the SDAn line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When it loses slave arbitration, the IIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of FFh.

The IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

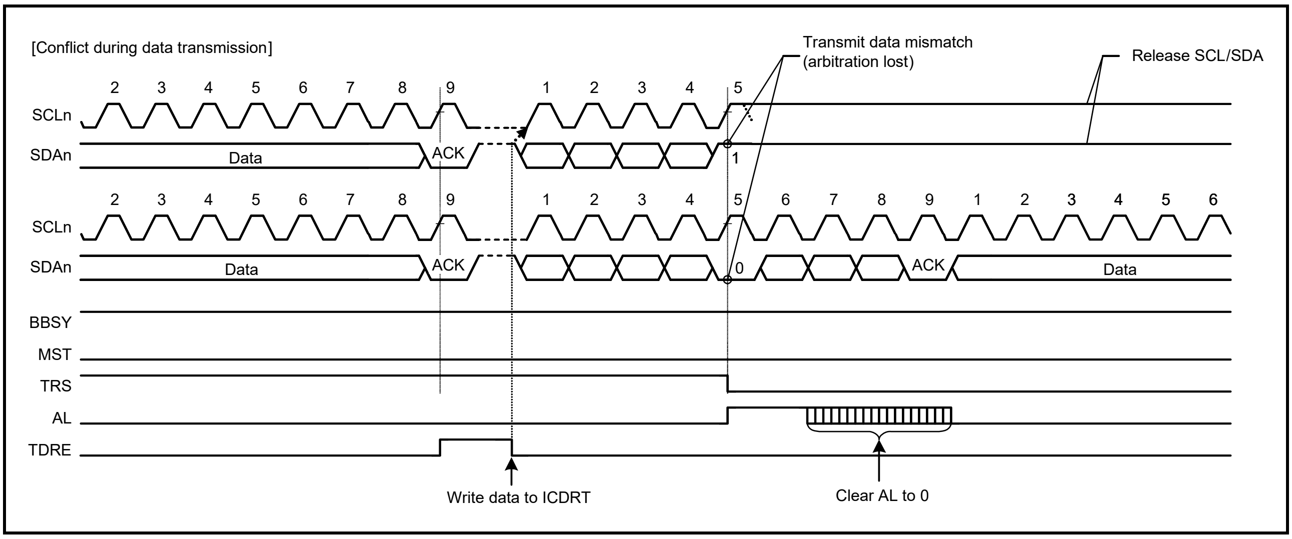


Figure 36.44 Example of slave arbitration-lost detection when SALE = 1

36.11 Start, Restart, and Stop Condition Issuing Function

36.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and the IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

36.11.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in ICCR2 is set to 1. When the RS bit is set to 1, a restart condition request is made, and the IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

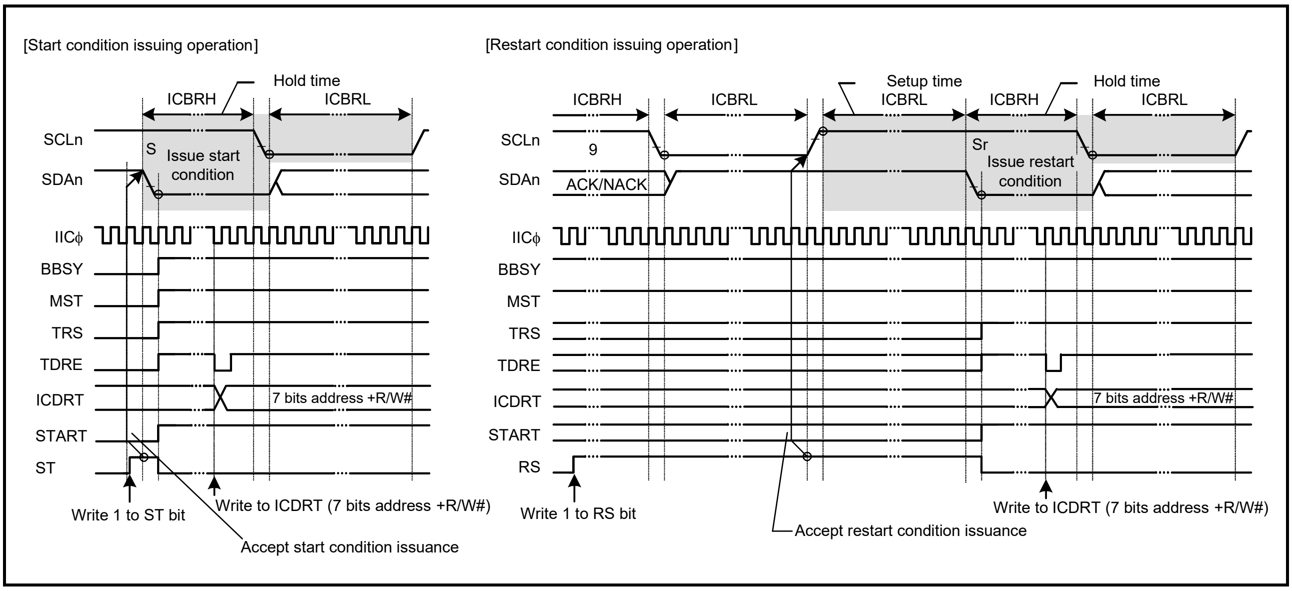


Figure 36.45 Start and restart condition issue timing using the ST and RS bits

Figure 36.46 shows the operation timing when a restart condition is issued after the master transmission.

[Restart condition issuance after the master transmission]

1. Initial setting. For details, refer to section 36.3.2, Initial Settings.
2. Read the BBSY flag in IICR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the IIC issues a start condition. At the same time, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line have matched while the ST bit is 1, the IIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and MST and TRS bits in ICCR2 are automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically set to 0, the data are transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave

address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the IIC continues in master transmit mode. Since the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.

4. After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. The IIC automatically holds the SCLn line low until the data for transmission are ready, a restart condition is issued or a stop condition is issued.
5. After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1, and then, after check that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issuance request). Upon receiving the request, the IIC issues a restart condition.
7. After check that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.

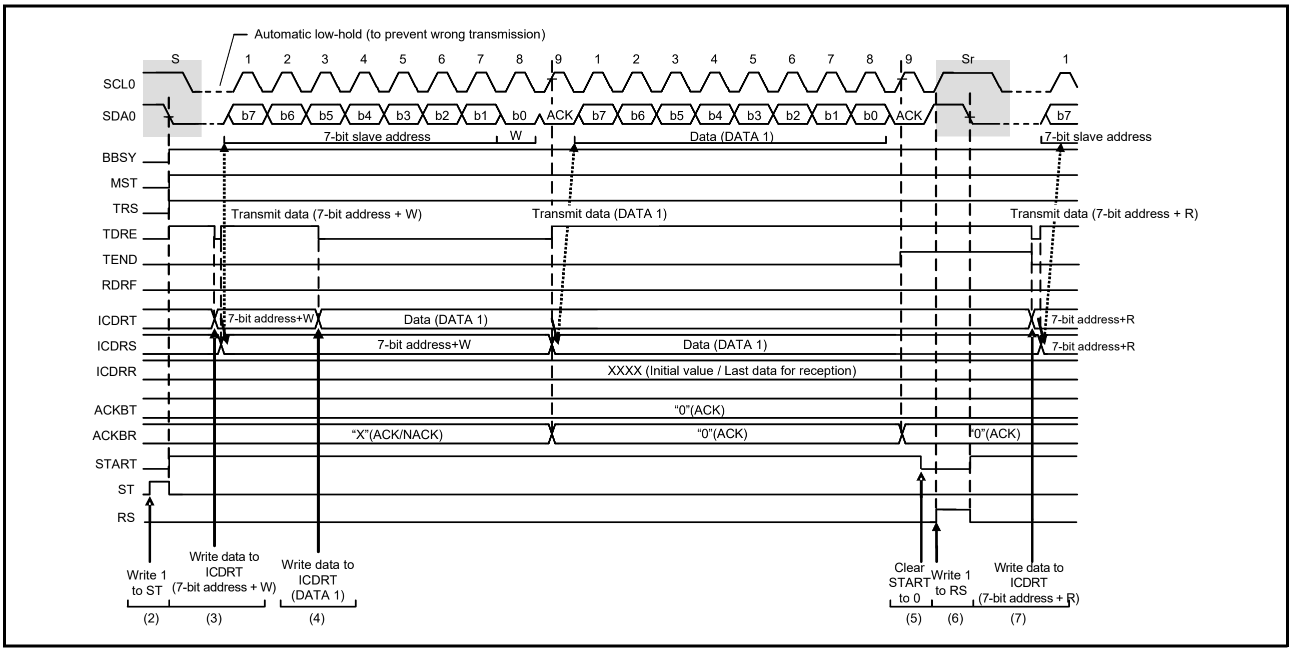


Figure 36.46 Restart condition issue timing after master transmission.

36.11.3 Issuing a Stop Condition

The IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.

7. Clear the BBSY flag to 0 to release the bus mastership.

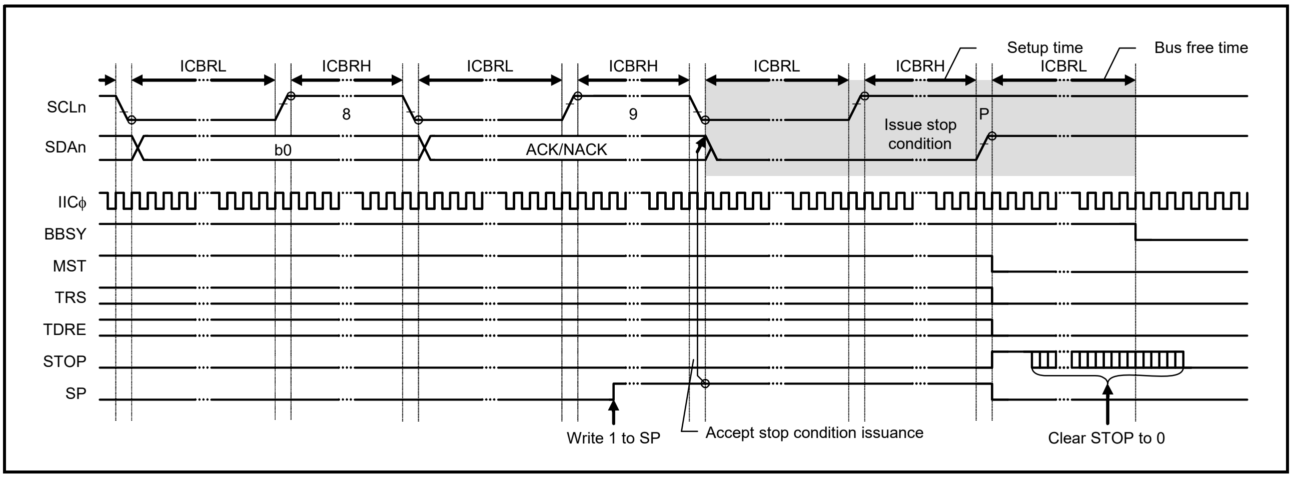


Figure 36.47 Stop condition issue timing using the SP bit

36.12 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization because of noise or other factors, the I^{2}C bus might hang with a fixed level on the SCLn or SDAn line.

To manage bus hanging, the IIC has:

- A timeout function to detect hanging by monitoring the SCLn line
- A function for the output of an extra SCL clock cycle to release the bus from a hung state because of clock signals being out of synchronization
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn line.

36.12.1 Timeout Function

The timeout function can detect when the SCLn line is stuck longer than the predetermined time. The IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

The timeout function monitors the SCLn line state and counts the low- or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rises or falls), but continues to count unless the SCLn line changes. If the internal counter overflows because no SCLn line changes, the IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- The IIC slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is free (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function uses the internal reference clock (IICϕ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter does not work.

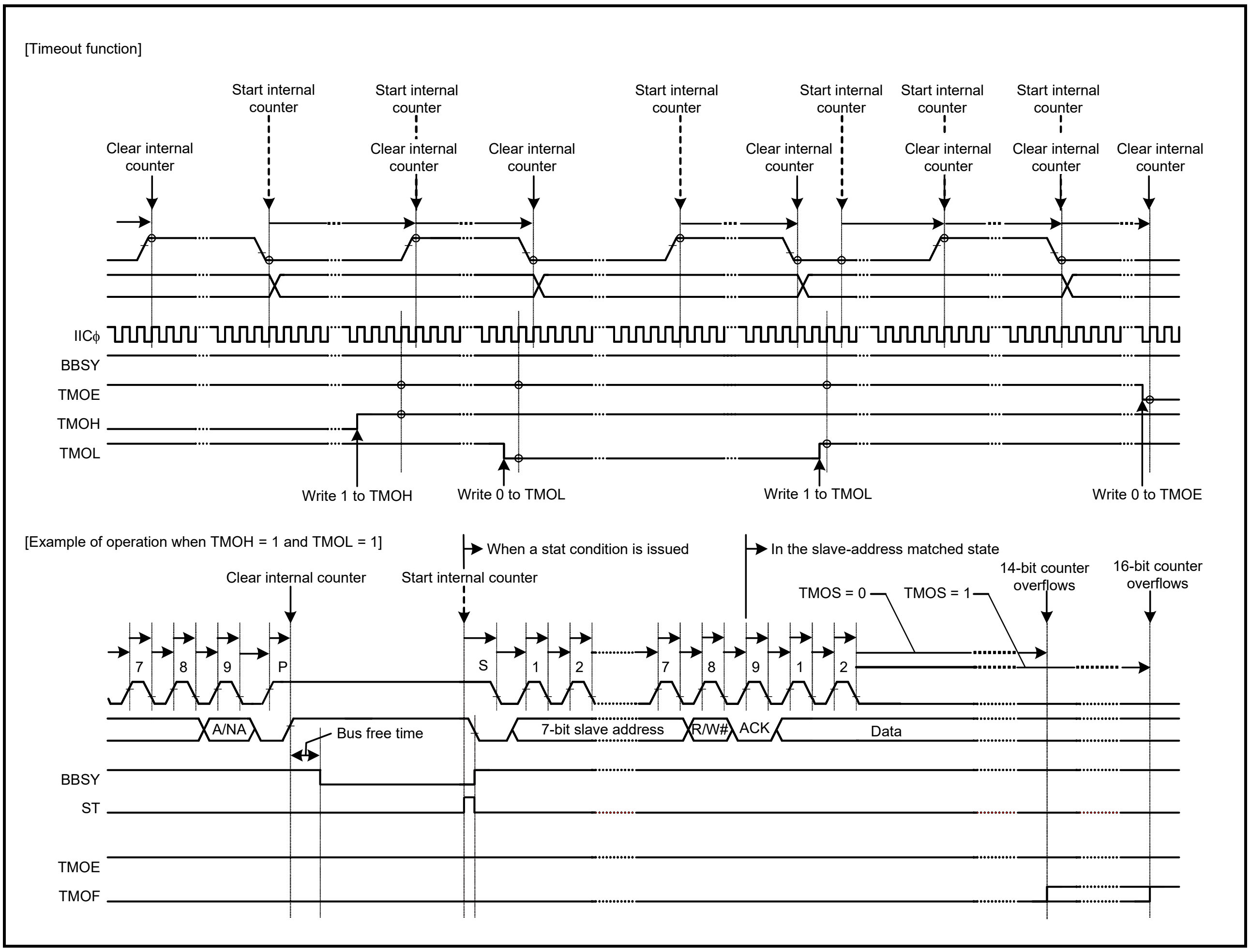


Figure 36.48 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

36.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held at the low level because the master is out of synchronization with the slave device. This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from the IIC. It uses single cycles of the SCL clock for a bus error where the IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations. Using it when communications are proceeding correctly leads to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the transfer rate specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit automatically is set to 0. More extra clock cycles can be output consecutively by software writing 1 to the CLO bit after having read CLO = 0.

When the IIC module is in master mode and the slave device is holding the SDAn line at the low level because synchronization with the slave device was lost because of noise or other effects, the output of a stop condition is not possible. This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, and so recovering the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

Use this function with the MALE bit in ICFER set to 0 (master arbitration-lost detection disabled). If the MALE bit is set to 1 (enabled), arbitration is lost when the value of the SDAO bit in ICCR1 does not match the state of the SDAn line.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)

- When the communication device does not hold the SCLn line low.

Figure 36.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

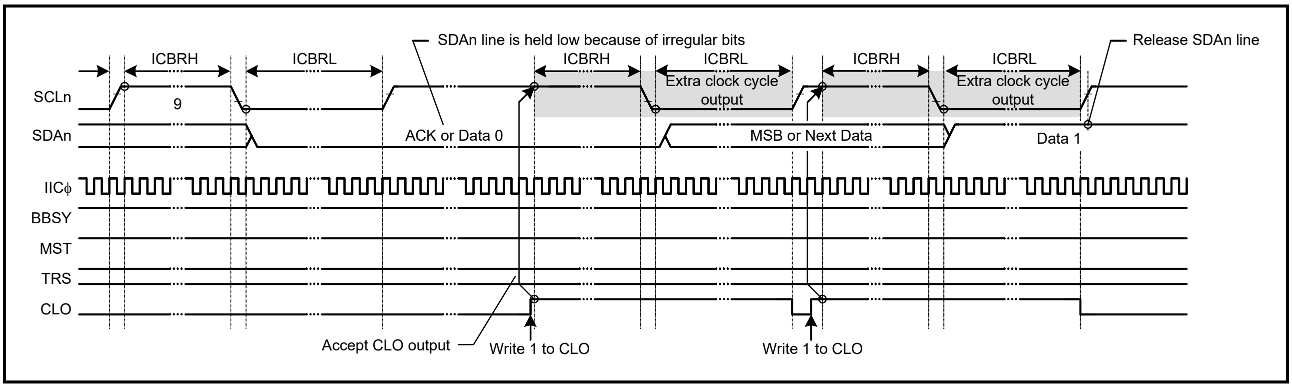


Figure 36.49 Extra SCL clock cycle output function using the CLO bit

36.12.3 IIC Reset and Internal Reset

The IIC module incorporates a function for resetting itself. It uses two types of resets: an IIC reset, which initializes all registers, including the BBSY flag in ICCR2, and an internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings. After issuing a reset, always set the IICRST bit in ICCR1 to 0.

Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see section 36.15, State of Registers when Issuing each Condition.

36.13 SMBus Operation

The IIC is available for data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, ICBRH, and ICBRL. In addition, specify the values in the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

36.13.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW: SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition.

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the GPT using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW: SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins, making them output high-impedance, which releases the bus.

(2) Measuring master device timeout

The following periods (timeout interval: $T_{LOW: MEXT}$) must be measured for master devices in SMBus communication:

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition.

To measure timeout for master devices, measure these periods with the GPT using the IIC start condition detection interrupt (STIn), stop condition detection interrupt (SPIn), transmit end interrupt (IICn_TEI), or receive data full interrupt (IICn_RXI). The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW: MEXT}$: 10 ms (maximum) of the SMBus standard, and the total of all $T_{LOW: MEXT}$ values from start condition to stop condition must be within $T_{LOW: SEXT}$: 25 ms (maximum).

For the ACK receive timing (rising edge of the ninth SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the ninth SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW: MEXT}$: 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDRT).

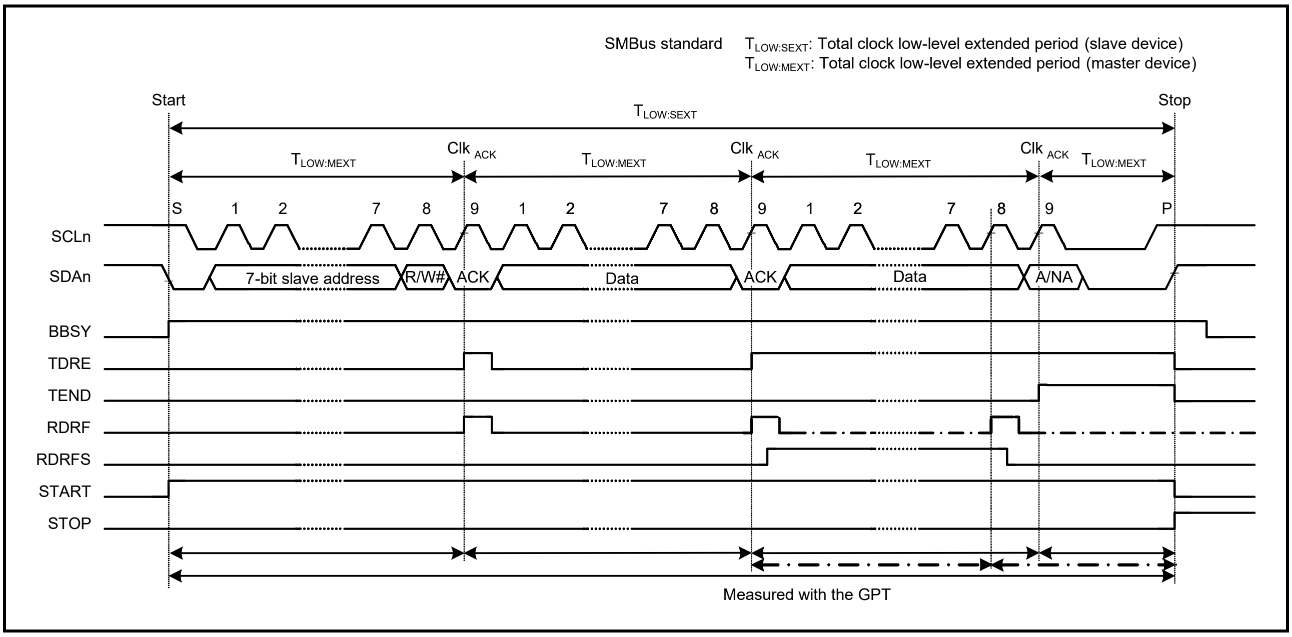


Figure 36.50 SMBus timeout measurement

36.13.2 Packet Error Code (PEC)

The MCU incorporates a CRC calculator, which enables transmission of a packet error code (PEC) or allows checking of the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see section 40, Cyclic Redundancy Check (CRC) Calculator.

In master transmit mode, the PEC data can be invoked by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the eighth SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the eighth clock cycle.

36.13.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communicating on an SMBus, a slave device can temporarily act as a master device to notify the SMBus host or ARP master of, or request the SMBus host for, its own slave address or request its own slave address from the SMBus host.

For a product using the MCU to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, so the IIC provides a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

36.14 Interrupt Sources

The IIC issues four types of interrupt requests: transfer error or event occurrence (detection of arbitration-lost, NACK, timeout, start or restart condition, or stop condition), receive data full, transmit data empty, and transmit end. Table 36.10 lists details about the interrupt requests. The receive data full and transmit data empty interrupts are both capable of activating data transfer by the DTC or DMAC.

Table 36.10 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DMAC/DTC activation	Interrupt condition
IICn_EEI*5	Transfer error or event occurrence	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI*2, *5	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI*1, *5	Transmit data empty	TDRE	Possible	TDRE = 1, TIE = 1
IICn_TEI*3, *5	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1
IIC0_WUI*4	Slave address match during wakeup function	WUF	Not possible	Slave address match Slave receive complete RWAK operation ASY0 = 1 WUIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, and then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.

Note 1. Because IICn_TXI is edge-detected, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn_TXI) automatically is set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Note 2. Because IICn_RXI is edge-detected, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) automatically is set to 0 when data is read from ICDRR.

Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

Note 4. Only channel 0 has a wakeup function, so IIC0_WUI is for channel 0 only.

Note 5. Channel number (n = 0 to 2).

Clear or mask each flag during interrupt handling.

36.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts

If the conditions for generating an IICn_TXI or IICn_RXI interrupt are satisfied while the associated IR flag is 1, the interrupt request is not output for the ICU but saved internally. One request per source can be saved internally.

An interrupt request that was being saved in the ICU is output when the ICU.IELSRn.IR flag is set to 0. Internally saved interrupt requests are automatically cleared under normal conditions. They can also be cleared by writing 0 to the interrupt enable bit within the given peripheral module.

36.15 State of Registers when Issuing each Condition

The IIC has two dedicated resets, IIC reset and internal reset. Table 36.11 shows the register states when issuing each condition.

Table 36.11 Register states when issuing each condition (1 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	In reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		In reset	In reset		
	Others			Saved		
ICCR2	BBSY	In reset	In reset	Saved	Set	Saved
	ST			In reset	Saved	Saved
	TRS,MST				Set or saved	In reset
	Others				In reset	In reset or Saved

Table 36.11 Register states when issuing each condition (2 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICMR1	BC[2:0]	In reset	In reset	In reset	In reset	Saved
	Others			Saved	Saved	
ICMR2		In reset	In reset	Saved	Saved	Saved
ICMR3		In reset	In reset	Saved	Saved	Saved
ICFER		In reset	In reset	Saved	Saved	Saved
ICSER		In reset	In reset	Saved	Saved	Saved
ICIER		In reset	In reset	Saved	Saved	Saved
ICSR1		In reset	In reset	In reset	Saved	In reset
ICSR2	TDRE, TEND	In reset	In reset	In reset	Saved	In reset
	START				Set	
	STOP				Saved	Set
	Others				Saved	Saved
ICWUR		In reset	In reset	Saved	Saved	Saved
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		In reset	In reset	Saved	Saved	Saved
ICBRH, ICBRL		In reset	In reset	Saved	Saved	Saved
ICDRT		In reset	In reset	Saved	Saved	Saved
ICDRR		In reset	In reset	Saved	Saved	Saved
ICDRS		In reset	In reset	In reset	Saved	Saved
Timeout function		In reset	In reset	Operating	Operating	Operating
Bus free time measurement		In reset	In reset	Operating	Operating	Operating
ICWUR2	WUSEN	In reset	In reset	Saved	Saved	Saved
	Others					Saved or Set or Reset

36.16 Output to the Event Link Controller (ELC)

The IIC0 to IIC2 modules handle event output for the ELC for the following sources:

(1) Transfer error event

When a transfer error event occurs, the associated event signal can be output to another module by the ELC.

(2) Receive data full

When a receive data register becomes full, the associated event signal can be output to another module by the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the associated event signal can be output to another module by the ELC.

(4) Transmit end

On completion of transfer, the associated event signal can be output to another module by the ELC.

36.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see Table 36.10) has an enable bit to control enabling and disabling of the associated interrupt signal. An interrupt request signal is output to the CPU when an interrupt source condition is satisfied while the associated enable bit is set.

The associated event link output signals are sent to other modules as event signals by the ELC when the interrupt source conditions are satisfied, regardless of the interrupt enable bit settings. For details on interrupt sources, see Table 36.10.

36.17 Usage Notes

36.17.1 Settings for the Module-Stop Function

IIC operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

36.17.2 Starting Transfer after an Interrupt Occurrence

If the IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure shown here to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally saved after transfer starts, and this can lead to unanticipated behavior of the IR flag.

To clear interrupts before starting transfer:

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits, such as ICIER.TIE, in the peripheral function to 0.
3. Read the relevant interrupt enable bits, such as ICIER.TIE, in the peripheral function and confirm that their value is 0.
4. Set the IR flag to 0.

37. Controller Area Network (CAN) Module

37.1 Overview

The Controller Area Network (CAN) module uses a message-based protocol to receive and transmit data between multiple slaves and masters in electromagnetically noisy applications. The module complies with the ISO 11898-1 (CAN 2.0A/CAN 2.0B) standard and supports up to 32 mailboxes, which can be configured for transmission or reception in normal mailbox and FIFO modes. Both standard (11-bit) and extended (29-bit) messaging formats are supported.

Table 37.1 lists the CAN module specifications and Figure 37.1 shows a block diagram. The CAN module requires an additional external CAN transceiver.

Table 37.1 CAN module specifications (1 of 2)

Parameter	Specifications
Data transfer rate	ISO11898-1-compliant for standard and extended frames
Bit rate	Data transfer rate programmable up to 1 Mbps (fCAN ≥ 8 MHz) fCAN: CAN clock source
Message box	32 mailboxes, with two selectable mailbox modes • Normal mode: 32 mailboxes independently configurable for either transmission or reception • FIFO mode: 24 mailboxes independently configurable for either transmission or reception, with remaining mailboxes used for receive and transmit 4-stage FIFOs.
Reception	• Support for data frame and remote frame reception • Reception ID format selectable to only standard ID, only extended ID, or mixed IDs • Programmable one-shot reception function • Selectable between overwrite mode (unread message overwritten) and overrun mode (unread message saved) • Reception complete interrupt independently enabled or disabled for each mailbox.
Acceptance filter	• Eight acceptance masks (one for every four mailboxes) • Masks independently enabled or disabled for each mailbox.
Transmission	• Support for data frame and remote frame transmission • Transmission ID format selectable to only standard ID, only extended ID, or mixed IDs) • Programmable one-shot transmission function • Broadcast messaging function • Priority mode selectable based on message ID or mailbox number • Support for transmission request abort, with abort completion comfirmable in status flag • Transmission complete interrupt independently enabled or disabled for each mailbox.
Mode transition for bus-off recovery	Mode transition for the recovery from the bus-off state selectable to: • ISO11898-1 specification-compliant • Automatic invoking of CAN halt mode on bus-off entry • Automatic invoking of CAN halt mode on bus-off end • Invoking of CAN halt mode through the software • Transition to error-active state through the software.
Error status monitoring	• Monitoring of CAN bus errors, including stuff error, form error, ACK error, 15-bit CRC error, bit error, and ACK delimiter error • Detection of transition to error states, including error-warning, error-passive, bus-off entry, and bus-off recovery • Supports reading of error counters.
Time stamping	• Time stamp function using a 16-bit counter • Reference clock selectable to 1-bit, 2-bit, 4-bit, and 8-bit time periods.
Interrupt function	Supports five interrupt sources: reception complete, transmission complete, receive FIFO, transmit FIFO, and error interrupts
CAN sleep mode	CAN clock stopped to reduce power consumption
Software support unit	Three software support units: • Acceptance filter support • Mailbox search support, including receive mailbox search, transmit mailbox search, and message lost search • Channel search support.
CAN clock source	PCLKB or CANMCLK

Table 37.1 CAN module specifications (2 of 2)

Parameter	Specifications
Test mode	Three test modes available for evaluation purposes: • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback).
Module-stop function	Module-stop state can be set to reduce power consumption

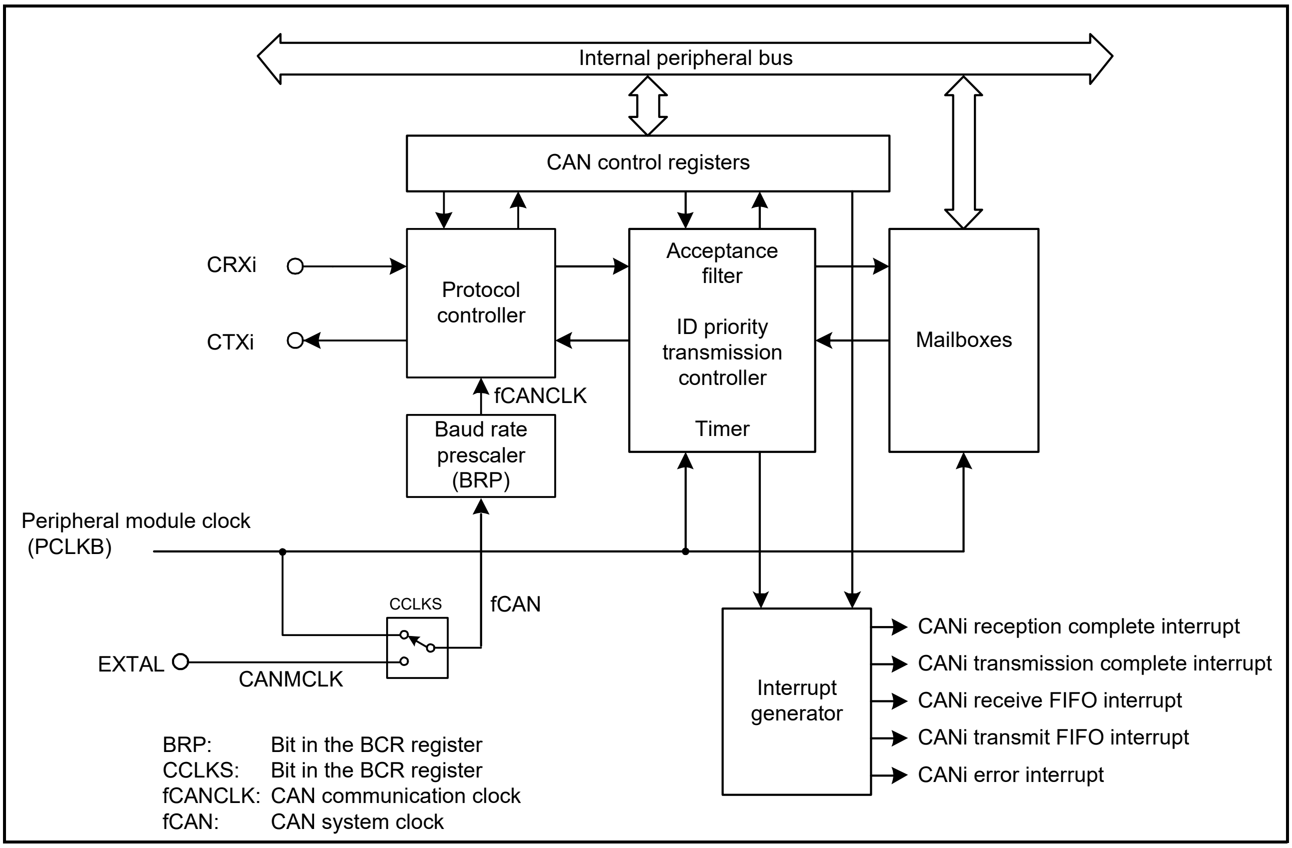


Figure 37.1 CAN module block diagram (i = 0, 1)

The CAN module includes the following blocks:

- CAN input and output pins
  CRXi and CTXi, where i = 0, 1
- Protocol controller
  Handles CAN protocol processing such as bus arbitration, bit timing during transmission and reception, stuffing, and error handling.
- Mailboxes
  Consists of 32 mailboxes, which can be configured as either transmit or receive. Each mailbox has an individual ID, data length code (DLC), data field (8 bytes), and time stamp.
- Acceptance filter
  Performs filtering of received messages. MKR0 to MKR7 are used for the filtering process.
- Timer
  Used for the time stamp function. The timer value when a message is stored in the mailbox is written as the time stamp value.
- Interrupt generator
  Generates five types of interrupts:
  - CANi reception complete interrupt

- CANi transmission complete interrupt
- CANi receive FIFO interrupt
- CANi transmit FIFO interrupt
- CANi error interrupt.

The CAN module communicates on the pins listed in Table 37.2. These pins are multiplexed with other signals on the MCU. For details, see section 20, I/O Ports.

Table 37.2 CAN module I/O pins

Pin name	I/O	Function
CRX0	Input	Data receive pin
CTX0	Output	Data transmit pin
CRX1	Input	Data receive pin
CTX1	Output	Data transmit pin

37.2 Register Descriptions

37.2.1 Control Register (CTLR)

Address(es): CAN0.CTLR 4005 0840h, CAN1.CTLR 4005 1840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RBOC	BOM[1:0]		SLPM	CANM[1:0]		TSPS[1:0]		TSRC	TPM	MLM	IDFM[1:0]		MBM
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MBM	CAN Mailbox Mode Select*1	0: Normal mailbox mode 1: FIFO mailbox mode.	R/W
b2, b1	IDFM[1:0]	ID Format Mode Select *1	b2 b1 0 0: Standard ID mode All mailboxes, including FIFO mailboxes, handle only standard IDs 0 1: Extended ID mode All mailboxes, including FIFO mailboxes, handle only extended IDs 1 0: Mixed ID mode All mailboxes, including FIFO mailboxes, handle both standard and extended IDs. In normal mailbox mode, use the associated IDE bit to differentiate standard and extended IDs. In FIFO mailbox mode, the associated IDE bits are used for mailboxes 0 to 23, the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit associated with mailbox 24 is used for the transmit FIFO. 1 1: Setting prohibited.	R/W
b3	MLM	Message Lost Mode Select*1	0: Overwrite mode 1: Overrun mode.	R/W
b4	TPM	Transmission Priority Mode Select*1	0: ID priority transmit mode 1: Mailbox number priority transmit mode.	R/W
b5	TSRC	Time Stamp Counter Reset Command*4	0: Nothing occurred 1: Reset.*3	R/W
b7, b6	TSPS[1:0]	Time Stamp Prescaler Select*1	b7 b6 0 0: Every bit time 0 1: Every 2-bit time 1 0: Every 4-bit time 1 1: Every 8-bit time.	R/W

Bit	Symbol	Bit name	Description	R/W
b9, b8	CANM[1:0]	CAN Operating Mode Select*5	b9 b8 0 0: CAN operation mode 0 1: CAN reset mode 1 0: CAN halt mode 1 1: CAN reset mode (forced transition).	R/W
b10	SLPM	CAN Sleep Mode*5,*6	0: All other modes 1: CAN sleep mode.	R/W
b12, b11	BOM[1:0]	Bus-Off Recovery Mode*1	b12 b11 0 0: Normal mode (ISO11898-1-compliant) 0 1: Enter CAN halt mode automatically on entering bus-off state 1 0: Enter CAN halt mode automatically on end of bus-off state 1 1: Enter CAN halt mode during bus-off recovery period through a software request.	R/W
b13	RBOC	Forcible Return from Bus-Off*2	0: Nothing occurred 1: Forced return from bus-off state.*3	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write to the BOM[1:0], TSPS[1:0], TPM, MLM, IDFM[1:0], and MBM bits in CAN reset mode.
Note 2. Set the RBOC bit to 1 in the bus-off state.
Note 3. This bit automatically is set to 0 after being set to 1. It should read as 0.
Note 4. Set the TSRC bit to 1 in CAN operation mode.
Note 5. When the CANM[1:0] and SLPM bits are changed, check STR to ensure that the mode is switched. Do not change the CANM[1:0] bits or SLPM bit until the mode is switched.
Note 6. Write to the SLPM bit in CAN reset mode or CAN halt mode. When changing the SLPM bit, write 0 or 1 to only the SLPM bit.

MBM bit (CAN Mailbox Mode Select)

When the MBM bit is 0 (normal mailbox mode), mailboxes 0 to 31 are configured as transmit or receive mailboxes. When the MBM bit is 1 (FIFO mailbox mode), mailboxes 0 to 23 are configured as transmit or receive mailboxes. Mailboxes 24 to 27 are configured as a transmit FIFO, and mailboxes 28 to 31 are configured as a receive FIFO. Transmit data is written into mailbox 24, the window mailbox for the transmit FIFO. Receive data is read from mailbox 28, the window mailbox for the receive FIFO.

Table 37.3 lists the mailbox configuration.

IDFM[1:0] bits (ID Format Mode Select)

The IDFM[1:0] bits specify the ID format.

MLM bit (Message Lost Mode Select)

The MLM bit specifies the operation when a new message is captured in the unread mailbox. Overwrite mode or overrun mode can be selected. In both cases, the mode applies to all mailboxes, including the receive FIFO.

When the MLM bit is 0, all mailboxes are set to overwrite mode. Any new message received overwrites the pre-existing message.

When the MLM bit is 1, all mailboxes are set to overrun mode. Any new message received does not overwrite the pre-existing message, and it is discarded.

TPM bit (Transmission Priority Mode Select)

The TPM bit specifies the priority when transmitting messages. ID priority transmit mode or mailbox number transmit mode can be selected. All mailboxes are set for either ID priority transmission or mailbox number priority transmission.

When TPM is 0, ID priority transmit mode is selected and transmission priority is arbitrated as defined in the ISO11898-1 CAN specification. In ID priority transmit mode, mailboxes 0 to 31 (in normal mailbox mode), and mailboxes 0 to 23 (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.

Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a FIFO message is being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.

When TPM is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (0 to 23).

TSRC bit (Time Stamp Counter Reset Command)

The TSRC bit resets the time stamp counter. When it is set to 1, TSR is set to 0000h. TSRC is set to 0 automatically.

TSPS[1:0] bits (Time Stamp Prescaler Select)

The TSPS[1:0] bits select the prescaler for the time stamp. The reference clock for the time stamp can be selected to 1-bit, 2-bit, 4-bit, or 8-bit time periods.

CANM[1:0] bits (CAN Operating Mode Select)

The CANM[1:0] bits select one of the following modes for the CAN module: CAN operation mode, CAN reset mode, or CAN halt mode. CAN sleep mode is set in the SLPM bit. For details, see section 37.3, Operation Modes.

When the CAN module enters CAN halt mode based on the BOM[1:0] setting, the CANM[1:0] bits are automatically set to 10b.

SLPM bit (CAN Sleep Mode)

When the SLPM bit is set to 1, the CAN module enters CAN sleep mode. When the SLPM bit is set to 0, the CAN module exits CAN sleep mode. For details, see section 37.3, Operation Modes.

BOM[1:0] bits (Bus-Off Recovery Mode)

The BOM[1:0] bits select bus-off recovery mode for the CAN module.

When the BOM[1:0] bits are 00b, the recovery from bus-off is compliant with the ISO11898-1 specification. The CAN module recovers CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request occurs when recovering from bus-off.

When the BOM[1:0] bits are 01b and the CAN module reaches the bus-off state, the CANM[1:0] bits in CTLR set 10b to enter CAN halt mode. No bus-off recovery interrupt request occurs when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 10b, the CANM[1:0] bits are set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after the recovery from the bus-off state, and after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request occurs when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 11b, the CAN module enters CAN halt mode by setting the CANM[1:0] bits to 10b while the CAN module is still in the bus-off state. No bus-off recovery interrupt request occurs when recovering from bus-off, and TECR and RECR are set to 00h. However, the interrupt does occur if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM[1:0] bits are set to 10b.

If the CPU requests an entry to CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM[1:0] bits are 01b, or at bus-off end when the BOM[1:0] bits are 10b), then the CPU request has higher priority.

RBOC bit (Forcible Return from Bus-Off)

When the RBOC bit is set to 1 in the bus-off state, the CAN module forcibly exits bus-off. It is set to 0 automatically, and the error state changes from bus-off to error-active. When the RBOC bit is set to 1, RECR and TECR clear to 00h and the BOST bit in STR is set to 0, indicating no bus-off state. The other registers remain unchanged when RBOC is set to 1. No bus-off recovery interrupt request occurs. Use the RBOC bit only when the BOM[1:0] bits are 00b (normal mode).

Table 37.3 Mailbox configuration

Mailbox	MBM bit = 0 (normal mailbox mode)	MBM bit = 1 (FIFO mailbox mode)*1 to *5
Mailboxes 0 to 23	Normal mailbox	Normal mailbox
Mailboxes 24 to 27		Transmit FIFO
Mailboxes 28 to 31		Receive FIFO

Note 1. The transmit FIFO is controlled by TFCR. The MCTL_TXj registers associated with mailboxes 24 to 27 are disabled. MCTL_TX24 to MCTL_TX27 cannot be used by the transmit FIFO.

Note 2. The receive FIFO is controlled by RFCR. The MCTL_RXj registers associated with mailboxes 28 to 31 are disabled. MCTL_RX28 to MCTL_RX31 cannot be used by the receive FIFO.
Note 3. See the MIER_FIFO description for information on the FIFO interrupts.
Note 4. The bits in MKIVLR associated with mailboxes 24 to 31 are disabled. Set 0 to these bits.
Note 5. The transmit and receive FIFOs can be used for both data frames and remote frames.

37.2.2 Bit Configuration Register (BCR)

Address(es): CAN0.BCR 4005 0844h, CAN1.BCR 4005 1844h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSEG1[3:0]				—		BRP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SJW[1:0]		—	TSEG2[2:0]			—	—	—	—	—	—	—	CCLKS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CCLKS	CAN Clock Source Selection	0: PCLKB (generated by the PLL clock) 1: CANMCLK (generated by the main clock oscillator).	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	TSEG2[2:0]	Time Segment 2 Control	b10 b8 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	SJW[1:0]	Synchronization Jump Width Control	b13 b12 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	BRP[9:0]	Baud Rate Prescaler Select *1	These bits set the frequency of the CAN communication clock (fCANCLK).	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	TSEG1[3:0]	Time Segment 1 Control	b31 b28 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq.	R/W

Tq: Time Quantum

Note 1. Do not select a value less than or equal to 1 while the SCKSCR.CKSEL[2:0] bits are 011b (selecting the main clock oscillator).

For setting the bit timing, see section 37.4, Data Transfer Rate Configuration. Set BCR before entering CAN halt or operation mode from reset mode. After the setting is made once, this register can be written to in CAN reset or CAN halt mode. 32-bit read/write accesses must be performed carefully so as not to change bits [7:0].

CCLKS bit (CAN Clock Source Selection)

When the CCLKS bit is 0, the peripheral module clock (PCLKB) produced by the PLL frequency synthesizer is used as the CAN clock source (fCAN). When the CCLKS bit is 1, CANMCLK produced externally by the EXTAL pins is used as the CAN clock source (fCAN).

TSEG2[2:0] bits (Time Segment 2 Control)

The TSEG2[2:0] bits specify the length of phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 8 Tq can be set. Set a value smaller than that of the TSEG1[3:0] bits.

SJW[1:0] bits (Synchronization Jump Width Control)

The SJW[1:0] bits specify the synchronization jump width with a Tq value. A value from 1 to 4 Tq can be set. Set a value smaller than or equal to that of the TSEG2[2:0] bits.

BRP[9:0] bits (Baud Rate Prescaler Select)

The BRP[9:0] bits set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 Tq. If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

TSEG1[3:0] bits (Time Segment 1 Control)

The TSEG1[3:0] bits specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a time quantum (Tq) value. A value from 4 to 16 Tq can be set.

37.2.3 Mask Register k (MKRk) (k = 0 to 7)

Address(es): CAN0.MKR[0] 4005 0400h to CAN0.MKR[7] 4005 041Ch, CAN1.MKR[0] 4005 1400h to CAN1.MKR[7] 4005 141Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Do not compare associated EID[17:0] bits 1: Compare associated EID[17:0] bits.	R/W
b28 to b18	SID[10:0]	Standard ID	0: Do not compare associated SID[10:0] bits 1: Compare associated SID[10:0] bits.	R/W
b31 to b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W

For the mask function in FIFO mailbox mode, see section 37.6, Acceptance Filtering and Masking Functions.

Write to MKR0 to MKR7 in CAN reset mode or CAN halt mode.

EID[17:0] bits (Extended ID)

The EID[17:0] bits are the filter mask bits associated with the CAN extended ID bits. They are used to receive extended ID messages. When an EID[17:0] bit is set to 0, the received ID is not compared with the associated mailbox ID. When the EID[17:0] bits are set to 1, the received ID is compared with the associated mailbox ID.

SID[10:0] bits (Standard ID)

The SID[10:0] bits are the filter mask bits associated with the CAN standard ID bits. They are used to receive both standard ID and extended ID messages. When the SID[10:0] bits are set to 0, the received ID is not compared with the associated mailbox ID. When the SID[10:0] bits are set to 1, the received ID is compared with the associated mailbox ID.

37.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1)

Address(es): CAN0.FIDCR0 4005 0420h, CAN0.FIDCR1 4005 0424h, CAN1.FIDCR0 4005 1420h, CAN1.FIDCR1 4005 1424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	Extended ID of the data and remote frames	R/W
b28 to b18	SID[10:0]	Standard ID	Standard ID of the data and remote frames	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame.	R/W
b31	IDE	ID Extension*1	0: Standard ID 1: Extended ID.	R/W

Note 1. The IDE bit is enabled when the CTLR.IDFM[1:0] bits are 10b (mixed ID mode). When the IDFM[1:0] bits are not 10b, only write 0 to IDE. It reads as 0.

FIDCR0 and FIDCR1 are enabled when the MBM bit in CTLR is set to 1 (FIFO mailbox mode). In this mode, the EID[17:0], SID[10:0], RTR, and IDE bits in mailbox 28 to mailbox 31 are disabled. Write to FIDCR0 and FIDCR1 in CAN reset mode or CAN halt mode. For information on using FIDCR0 and FIDCR1, see section 37.6, Acceptance Filtering and Masking Functions.

EID[17:0] bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames. They are used to receive extended ID messages.

SID[10:0] bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames. They are used to receive both standard ID and extended ID messages.

RTR bit (Remote Transmission Request)

The RTR bit sets the frame format to data frames or remote frames:

- When both RTR bits in FIDCR0 and FIDCR1 are set to 0, only data frames are received
- When both RTR bits in FIDCR0 and FIDCR1 are set to 1, only remote frames are received
- When the RTR bits in FIDCR0 and FIDCR1 are set to different values, both data frames and remote frames are received.

IDE bit (ID Extension)

The IDE bit sets the ID format to standard ID or extended ID. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode):

- When both IDE bits in FIDCR0 and FIDCR1 are set to 0, only standard ID frames are received
- When both IDE bits in FIDCR0 and FIDCR1 are set to 1, only extended ID frames are received
- When the IDE bits in FIDCR0 and FIDCR1 are set to different values, both standard ID and extended ID frames are received.

37.2.5 Mask Invalid Register (MKIVLR)

Address(es): CAN0.MKIVLR 4005 0428h, CAN1.MKIVLR 4005 1428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MB31 to MB0	Mask Invalid	0: Mask valid 1: Mask invalid.	R/W

Each bit in MKIVLR is associated with a mailbox of the same number. Bit [0] in MKIVLR corresponds to mailbox 0 (MB0), and bit [31] corresponds to mailbox 31 (MB31).*1

When a bit is set to 1, the associated acceptance mask register becomes invalid for the associated mailbox. When a mask invalid bit is set to 1, a message is received by the associated mailbox only if the receive message ID matches the mailbox ID exactly.

Write to MKIVLR in CAN reset or halt mode.

Note 1. Set bits [31:24] to 0 in FIFO mailbox mode.

37.2.6 Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31; m = 0 to 7)

Table 37.4 lists the CANi mailbox memory mapping, and Table 37.5 lists the CAN data frame configuration.

The value after reset of the CANi mailbox is undefined.

Write to MBj_ID, MBj_DL, MBj_Dm, and MBj_TS only when the related MCTL_TXj or MCTL_RXj register (j = 0 to 31) is 00h and the associated mailbox is not processing an abort request.

See Table 37.4 for detailed register addresses.

Table 37.4 CANi mailbox memory mapping (1 of 2)

Address		Message content
CAN0	**CAN1**	**Memory mapping**
4005 0200h + 16 × j + 0	4005 1200h + 16 × j + 0	IDE, RTR, SID10 to SID6
4005 0200h + 16 × j + 1	4005 1200h + 16 × j + 1	SID5 to SID0, EID17, EID16
4005 0200h + 16 × j + 2	4005 1200h + 16 × j + 2	EID15 to EID8
4005 0200h + 16 × j + 3	4005 1200h + 16 × j + 3	EID7 to EID0
4005 0200h + 16 × j + 4	4005 1200h + 16 × j + 4	—

Table 37.4 CANi mailbox memory mapping (2 of 2)

Address		Message content
CAN0	**CAN1**	**Memory mapping**
4005 0200h + 16 × j + 5	4005 1200h + 16 × j + 5	Data length code (DLC[3:0])
4005 0200h + 16 × j + 6	4005 1200h + 16 × j + 6	Data byte 0
4005 0200h + 16 × j + 7	4005 1200h + 16 × j + 7	Data byte 1
4005 0200h + 16 × j + 8	4005 1200h + 16 × j + 8	Data byte 2
4005 0200h + 16 × j + 9	4005 1200h + 16 × j + 9	Data byte 3
4005 0200h + 16 × j + 10	4005 1200h + 16 × j + 10	Data byte 4
4005 0200h + 16 × j + 11	4005 1200h + 16 × j + 11	Data byte 5
4005 0200h + 16 × j + 12	4005 1200h + 16 × j + 12	Data byte 6
4005 0200h + 16 × j + 13	4005 1200h + 16 × j + 13	Data byte 7
4005 0200h + 16 × j + 14	4005 1200h + 16 × j + 14	Time stamp upper byte
4005 0200h + 16 × j + 15	4005 1200h + 16 × j + 15	Time stamp lower byte

Table 37.5 CAN data frame configuration

SID10 to SID6	SID5 to SID0	EID17 to EID16	EID15 to EID8	EID7 to EID0	DLC3 to DLC1	DATA0	DATA1	…	DATA7

The previous value of each mailbox is retained unless a new message is received.

Address(es): CAN0.MB0_ID 4005 0200h to CAN0.MB31_ID 4005 03F0h, CAN1.MB0_ID 4005 1200h to CAN1.MB31_ID 4005 13F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b17 to b0	EID[17:0]	Extended ID*1	Extended ID of the data and remote frames	R/W
b28 to b18	SID[10:0]	Standard ID	Standard ID of the data and remote frames	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame.	R/W
b31	IDE	ID Extension*2	0: Standard ID 1: Extended ID.	R/W

Note 1. If the mailbox receives a standard ID message, the EID bits in the mailbox are undefined.

Note 2. The IDE bit is enabled when the CTLR.IDFM[1:0] bits are 10b (mixed ID mode). When the IDFM[1:0] bits are any value other than 10b, the IDE bit should be written with 0 and read as 0.

Address(es): CAN0.MB0_DL 4005 0204h to CAN0.MB31_DL 4005 03F4h, CAN1.MB0_DL 4005 1204h to CAN1.MB31_DL 4005 13F4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	DLC[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	DLC[3:0]	Data Length Code*1	b3 b0 0 0 0 0: Data length = 0 byte 0 0 0 1: Data length = 1 byte 0 0 1 0: Data length = 2 bytes 0 0 1 1: Data length = 3 bytes 0 1 0 0: Data length = 4 bytes 0 1 0 1: Data length = 5 bytes 0 1 1 0: Data length = 6 bytes 0 1 1 1: Data length = 7 bytes 1 x x x: Data length = 8 bytes.	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

x: Don't care

Note 1. If the mailbox receives a message with data length (set in DLC[3:0]) of n bytes, where n is less than 8, the data in the DATAn to DATA7 registers in the mailbox is undefined. DATA0 to DATA7 are data registers for this mailbox. For example, if data length is 6 bytes (DLC[3:0] = 6h), the data in DATA6 and DATA7 registers is undefined.

Address(es): CAN0.MB0_D0 4005 0206h to CAN0.MB31_D0 4005 03F6h, CAN1.MB0_D0 4005 1206h to CAN1.MB31_D0 4005 13F6h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA0							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D1 4005 0207h to CAN0.MB31_D1 4005 03F7h, CAN1.MB0_D1 4005 1207h to CAN1.MB31_D1 4005 13F7h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA1							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D2 4005 0208h to CAN0.MB31_D2 4005 03F8h, CAN1.MB0_D2 4005 1208h to CAN1.MB31_D2 4005 13F8h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA2							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D3 4005 0209h to CAN0.MB31_D3 4005 03F9h, CAN1.MB0_D3 4005 1209h to CAN1.MB31_D3 4005 13F9h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA3							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D4 4005 020Ah to CAN0.MB31_D4 4005 03FAh, CAN1.MB0_D4 4005 120Ah to CAN1.MB31_D4 4005 13FAh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA4							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D5 4005 020Bh to CAN0.MB31_D5 4005 03FBh, CAN1.MB0_D5 4005 120Bh to CAN1.MB31_D5 4005 13FBh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA5							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D6 4005 020Ch to CAN0.MB31_D6 4005 03FCh, CAN1.MB0_D6 4005 120Ch to CAN1.MB31_D6 4005 13FCh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA6							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D7 4005 020Dh to CAN0.MB31_D7 4005 03FDh, CAN1.MB0_D7 4005 120Dh to CAN1.MB31_D7 4005 13FDh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA7							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DATA0 to DATA7	Data Bytes 0 to 7*1,*2	DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB-first, and transmission or reception starts from bit [7].	R/W

Note 1. If the mailbox receives a message with n bytes less than 8 bytes, the values of DATAn to DATA7 in the mailbox are undefined.
Note 2. If the mailbox receives a remote frame, the previous values of DATA0 to DATA7 in the mailbox are saved.

Address(es): CAN0.MB0_TS 4005 020Eh to CAN0.MB31_TS 4005 03FEh, CAN1.MB0_TS 4005 120Eh to CAN1.MB31_TS 4005 13FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSH[7:0]								TSL[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	TSL[7:0]	Time Stamp Lower Byte	The TSH[7:0] and TSL[7:0] bits store the counter value of the time stamp when received messages are stored in the mailbox.	R/W
b15 to b8	TSH[7:0]	Time Stamp Higher Byte		R/W

EID[17:0] bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames. They are used to transmit or receive extended ID messages.

SID[10:0] bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames. They are used to transmit or receive both standard ID and extended ID messages.

RTR bit (Remote Transmission Request)

The RTR bit sets the frame format to data frames or remote frames.

- The receive mailbox only receives frames with the format specified in the RTR bit
- The transmit mailbox transmits with the frame format specified in the RTR bit
- The receive FIFO mailbox receives the data frame, remote frame, or both frames specified in the RTR bit in FIDCR0 and FIDCR1
- The transmit FIFO mailbox transmits the data frame or remote frame specified in the RTR bit in the transmit message.

IDE bit (ID Extension)

The IDE bit sets the ID format to standard IDs or extended IDs. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode).

- The receive mailbox only receives the ID format specified in the IDE bit
- The transmit mailbox transmits with the ID format specified in the IDE bit
- The receive FIFO mailbox receives messages with the standard ID and extended ID settings specified in the IDE bit in FIDCR0 and FIDCR1
- The transmit FIFO mailbox transmits messages with the standard ID or extended ID settings specified in the IDE bit in the transmit message.

DLC[3:0] bits (Data Length Code)

The DLC[3:0] bits specify the data length to be transmitted in data frames. When a remote frame is used to request data, this field specifies the requested data length.

When a data frame is received, the received data length is stored in this field. When a remote frame is received, this field stores the requested data length.

37.2.7 Mailbox Interrupt Enable Register (MIER)

Address(es): CAN0.MIER 4005 042Ch, CAN1.MIER 4005 142Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MB31 to MB0	Interrupt Enable	0: Disable interrupt 1: Enable interrupt. Bit [31] is associated with mailbox 31 (MB31), and bit [0] with mailbox 0 (MB0).	R/W

MIER can enable interrupts for each mailbox independently. This register is available in normal mailbox mode. Do not access this register in FIFO mailbox mode.

Each bit is associated with the mailbox having the same number. These bits enable or disable transmission and reception complete interrupts for the associated mailboxes:

- Bit [0] in MIER corresponds to mailbox 0 (MB0)
- Bit [31] in MIER corresponds to mailbox 31 (MB31).

Write to MIER only when the associated MCTL_TXj or MCTL_RXj register (j = 0 to 31) is 00h and the associated mailbox is not processing a transmission or reception abort request.

37.2.8 Mailbox Interrupt Enable Register for FIFO Mailbox Mode (MIER_FIFO)

Address(es): CAN0.MIER_FIFO 4005 042Ch, CAN1.MIER_FIFO 4005 142Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	MB29	MB28	—	—	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b23 to b0	MB23 to MB0	Interrupt Enable	0: Disable interrupt 1: Enable interrupt. Bit [23] is associated with mailbox 23 (MB23), and bit [0] with mailbox 0 (MB0).	R/W
b24	MB24	Transmit FIFO Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b25	MB25	Transmit FIFO Interrupt Generation Timing Control	0: Generate every time transmission completes 1: Generate when the transmit FIFO empties on transmission completion.	R/W
b27, b26	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b28	MB28	Receive FIFO Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b29	MB29	Receive FIFO Interrupt Generation Timing Control*1	0: Generate every time reception completes 1: Generate when the receive FIFO becomes a buffer warning*2 on reception completion.	R/W
b31, b30	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. No interrupt request occurs when the receive FIFO becomes buffer warning because it is full.
Note 2. "Buffer warning" indicates a state in which the third message is stored in the receive FIFO.

MIER_FIFO can individually enable interrupts for each mailbox and FIFO. This register is available in normal mailbox mode and FIFO mailbox mode. Do not access it in normal mailbox mode.

The MB0 to MB23 bits are associated with the mailbox having the same number. These bits enable or disable transmission and reception complete interrupts for the associated mailboxes:

- Bit [0] in MIER_FIFO is associated with mailbox 0 (MB0)
- Bit [23] in MIER_FIFO is associated with mailbox 23 (MB23).

MB24, MB25, MB28, and MB29 specify whether transmit and receive FIFO interrupts are enabled or disabled, and the timing of interrupt requests.

Write to MIER_FIFO only when the associated MCTL_TXj or MCTL_RXj register (j = 0 to 31) is 00h and the associated mailbox is not processing a transmission or reception abort request. In addition, change the bits in MIER_FIFO for the associated FIFO only when the TFE bit in TFCR is 0 and the TFEST bit is 1, and the RFE bit in RFCR is 0 and the RFEST bit in RFCR is 1.

37.2.9 Message Control Register for Transmit (MCTL_TXj) (j = 0 to 31)

- Transmit mode (when the TRMREQ bit is 1 and the RECREQ bit is 0)

Address(es): CAN0.MCTL_TX[0] 4005 0820h to CAN0.MCTL_TX[31] 4005 083Fh,
CAN1.MCTL_TX[0] 4005 1820h to CAN1.MCTL_TX[31] 4005 183Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SENTDATA	Transmission Complete Flag *1,*2	0: Transmission not complete 1: Transmission complete.	R/W
b1	TRMACTIVE	Transmission-in-Progress Status Flag	0: Transmission pending or not requested 1: Transmission in progress.	R
b2	TRMABT	Transmission Abort Complete Flag *1,*2	0: Transmission started, transmission abort failed because transmission completed, or transmission abort not requested 1: Transmission abort complete.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ONESHOT	One-Shot Enable *2,*3	0: Disable one-shot transmission 1: Enable one-shot transmission.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	RECREQ	Receive Mailbox Request *2,*3,*4,*5	0: Do not configure for reception 1: Configure for reception.	R/W
b7	TRMREQ	Transmit Mailbox Request *2,*4	0: Do not configure for transmission 1: Configure for transmission.	R/W

Note 1. Write 0 only. Writing 1 has no effect.
Note 2. When writing to bits of this register, write 1 to SENTDATA and TRMABT if these bits are not the write target.
Note 3. To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1. To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message is transmitted or aborted.
Note 4. Do not set both the RECREQ and TRMREQ bits to 1.
Note 5. When setting the RECREQ bit to 0, set the SENTDATA, TRMACTIVE, and TRMABT flags to 0 simultaneously.

MCTL_TXj sets mailbox j to transmit mode or receive mode. In transmit mode, MCTL_TXj also controls and indicates the transmission status. Do not access MCTL_TXj if mailbox j is in receive mode. Only write to MCTL_TXj in CAN operation or halt mode. Do not use MCTL_TX24 to MCTL_TX31 in FIFO mailbox mode.

SENTDATA flag (Transmission Complete Flag)

The SENTDATA flag is set to 1 when data transmission from the associated mailbox is complete. The SENTDATA flag is set to 0 through a software write. To set it to 0, first set the TRMREQ bit to 0. The SENTDATA and TRMREQ flags cannot be set to 0 simultaneously. To transmit a new message from the associated mailbox, set the SENTDATA flag to 0.

TRMACTIVE flag (Transmission-in-Progress Status Flag)

The TRMACTIVE flag is set to 1 when the associated mailbox of the CAN module begins transmitting a message. It is set to 0 when the CAN module loses CAN bus arbitration, a CAN bus error occurs, or data transmission completes.

TRMABT flag (Transmission Abort Complete Flag)

The TRMABT flag is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort is complete before starting transmission
- Following a transmission abort request, when the CAN module detects CAN bus arbitration-lost or a CAN bus error
- In one-shot transmission mode (RECREQ = 0, TRMREQ = 1, and ONESHOT = 1), when the CAN module detects a CAN bus arbitration-lost or a CAN bus error.

The TRMABT flag does not set to 1 when data transmission is complete. The SENTDATA flag is set to 1. The TRMABT flag is set to 0 through a software write.

ONESHOT bit (One-Shot Enable)

When the ONESHOT bit is set to 1 in transmit mode (RECREQ = 0 and TRMREQ = 1), the CAN module transmits a message only one time. The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration-lost occurs. When transmission is complete, the SENTDATA flag is set to 1. If transmission does not complete because of a CAN bus error or CAN bus arbitration-lost error, the TRMABT flag is set to 1. Set the ONESHOT bit to 0 after the SENTDATA or TRMABT flag is set to 1.

RECREQ bit (Receive Mailbox Request)

The RECREQ bit selects the receive modes listed in Table 37.10.

When the RECREQ bit is set to 1, the associated mailbox is configured for reception of a data or remote frame.

When the RECREQ bit is set to 0, the associated mailbox is not configured for reception of a data or remote frame.

Due to hardware protection, the RECREQ bit cannot be set to 0 through a software write during the following period:

- Hardware protection is started from the acceptance filter processing (the beginning of the CRC field)
- Hardware protection is released:
  - For the mailbox that is specified to receive the incoming message, after the received data is stored in the mailbox or a CAN bus error occurs. This means that the maximum period of hardware protection is from the beginning of CRC field to the end of the 7th bit of EOF.
  - For the other mailboxes, after acceptance filter processing
  - If no mailbox is specified to receive the message, after acceptance filter processing.

When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set the SENTDATA and TRMABT flags to 0 before changing to reception.

Note: MCTL_TXj.RECREQ is the mirror bit of MCTL_RXj.RECREQ.

TRMREQ bit (Transmit Mailbox Request)

The TRMREQ bit selects the transmit modes listed in Table 37.10.

When the TRMREQ bit is set to 1, the associated mailbox is configured for transmission of a data or remote frame.

When the TRMREQ bit is set to 0, the associated mailbox is not configured for transmission of a data or remote frame.

If the TRMREQ bit is changed from 1 to 0 to cancel the associated transmission request, either the TRMABT or SENTDATA flag is set to 1. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception, and then set the NEWDATA and MSGLOST bits to 0 before changing to transmission.

Note: MCTL_TXj.TRMREQ is the mirror bit of MCTL_RXj.TRMREQ.

37.2.10 Message Control Register for Receive (MCTL_RXj) (j = 0 to 31)

- Receive mode (when the TRMREQ bit is 0 and the RECREQ bit is 1)

Address(es): CAN0.MCTL_RX[0] 4005 0820h to CAN0.MCTL_RX[31] 4005 083Fh,
CAN1.MCTL_RX[0] 4005 1820h to CAN1.MCTL_RX[31] 4005 183Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALDATA	NEWDATA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	NEWDATA	Reception Complete Flag*1,*2	0: No data received, or 0 was written to the bit 1: New message being stored or was stored in the mailbox.	R/W
b1	INVALDATA	Reception-in-Progress Status Flag	0: Message valid 1: Message being updated.	R
b2	MSGLOST	Message Lost Flag*1,*2	0: Message not overwritten or overrun 1: Message overwritten or overrun.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ONESHOT	One-Shot Enable*2,*3	0: Disable one-shot reception 1: Enable one-shot reception.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	RECREQ	Receive Mailbox Request *2,*3,*4,*5	0: Do not configure for reception 1: Configure for reception.	R/W
b7	TRMREQ	Transmit Mailbox Request *2,*4	0: Do not configure for transmission 1: Configure for transmission.	R/W

Note 1. Write 0 only. Writing 1 has no effect.
Note 2. When writing to bits of this register, write 1 to the NEWDATA and MSGLOST bits if they are not the write target.
Note 3. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1. To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming that it is 0.
Note 4. Do not set both the RECREQ and TRMREQ bits to 1.
Note 5. When setting the RECREQ bit to 0, set the MSGLOST, NEWDATA, and RECREQ bits to 0 simultaneously.

MCTL_RXj sets mailbox j to transmit or receive mode. In receive mode, MCTL_RXj also controls and indicates the reception status. Do not access MCTL_RXj if mailbox j is in transmit mode. Only write to the MCTL_RXj in CAN operation or halt mode. Do not use MCTL_RX24 to MCTL_RX31 in FIFO mailbox mode.

NEWDATA flag (Reception Complete Flag)

The NEWDATA flag is set to 1 when a new message is being stored or was stored in the mailbox. Always set this bit to 1 simultaneously with the INVALDATA flag. The NEWDATA flag is cleared to 0 through a software write. The NEWDATA flag cannot be set to 0 through a software write while the associated INVALDATA flag is 1.

INVALDATA flag (Reception-in-Progress Status Flag)

After the completion of a message reception, the INVALDATA flag is set to 1 while the received message is being updated into the associated mailbox. The INVALDATA flag is set to 0 immediately after the message is stored. If the mailbox is read while the INVALDATA flag is 1, the data is undefined.

MSGLOST flag (Message Lost Flag)

The MSGLOST flag is set to 1 when the mailbox is overwritten or overrun by a new received message while the NEWDATA flag is 1. The MSGLOST flag is set to 1 at the end of the 6th bit of EOF. The MSGLOST flag is set to 0 through a software write.

In both overwrite and overrun modes, the MSGLOST flag cannot be set to 0 through a software write during the 5 PCLKB cycles following the 6th bit of EOF.

ONESHOT bit (One-Shot Enable)

When the ONESHOT bit is set to 1 in receive mode (RECREQ = 1 and TRMREQ = 0), the mailbox receives a message only one time. The mailbox does not behave as a receive mailbox after having received a message one time. The behavior of the NEWDATA and INVALDATA flags is the same as in normal receive mode. In one-shot receive mode, the MSGLOST flag does not set to 1. To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it is 0.

RECREQ bit (Receive Mailbox Request)

The RECREQ bit selects receive modes listed in Table 37.10.

When the RECREQ bit is set to 1, the associated mailbox is configured for reception of a data or remote frame.

When the RECREQ bit is set to 0, the associated mailbox is not configured for reception of a data or remote frame.

Due to hardware protection, the RECREQ bit cannot be set to 0 through a software write during the following period:

- Hardware protection is started from the acceptance filter processing (the beginning of the CRC field)
- Hardware protection is released:
  - For the mailbox that is specified to receive the incoming message, after the received data is stored in the mailbox or a CAN bus error occurs. This means that the maximum period of hardware protection is from the beginning of the CRC field to the end of the 7th bit of EOF.
  - For the other mailboxes, after acceptance filter processing
  - If no mailbox is specified to receive the message, after acceptance filter processing.

When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set the SENTDATA and TRMABT flags to 0 before changing to reception.

Note: MCTL_RXj.RECREQ is the mirror bit of MCTL_TXj.RECREQ.

TRMREQ bit (Transmit Mailbox Request)

The TRMREQ bit selects the transmit modes listed in Table 37.10.

When the TRMREQ bit is set to 1, the associated mailbox is configured for transmission of a data or remote frame.

When the TRMREQ bit is set to 0, the associated mailbox is not configured for transmission of a data or remote frame.

If the TRMREQ bit is changed from 1 to 0 to cancel the associated transmission request, either the TRMABT or SENTDATA flag is set to 1. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception and then set the NEWDATA and MSGLOST bits to 0 before changing to transmission.

Note: MCTL_RXj.TRMREQ is the mirror bit of MCTL_TXj.TRMREQ.

37.2.11 Receive FIFO Control Register (RFCR)

Address(es): CAN0.RFCR 4005 0848h, CAN1.RFCR 4005 1848h

	b7	b6	b5	b4	b3	b2	b1	b0
	RFEST	RFWST	RFFST	RFMLF	RFUST[2:0]			RFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RFE	Receive FIFO Enable	0: Disable receive FIFO 1: Enable receive FIFO.	R/W

Bit	Symbol	Bit name	Description	R/W
b3 to b1	RFUST[2:0]	Receive FIFO Unread Message Number Status	b3 b1 0 0 0: No unread message 0 0 1: 1 unread message 0 1 0: 2 unread messages 0 1 1: 3 unread messages 1 0 0: 4 unread messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved.	R
b4	RFMLF	Receive FIFO Message Lost Flag	0: Receive FIFO message not lost 1: Receive FIFO message lost.	R/W
b5	RFFST	Receive FIFO Full Status Flag	0: Receive FIFO not full 1: Receive FIFO full (4 unread messages).	R
b6	RFWST	Receive FIFO Buffer Warning Status Flag	0: Receive FIFO is not buffer warning 1: Receive FIFO is buffer warning (3 unread messages).	R
b7	RFEST	Receive FIFO Empty Status Flag	0: Unread message in receive FIFO 1: No unread message in receive FIFO.	R

Write to RFCR in CAN operation or halt mode.

RFE bit (Receive FIFO Enable)

When the RFE bit is set to 1, the receive FIFO is enabled.

When the RFE bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST bit = 1). Write 0 to the RFE bit simultaneously with setting the RFMLF flag.

Do not set this bit to 1 in normal mailbox mode (MBM bit in CTLR = 0). Due to hardware protection, the RFE bit cannot be set to 0 through a software write during the following period:

- Hardware protection is started from acceptance filter processing (the beginning of the CRC field)
- When hardware protection is released:
  - If the receive FIFO is specified to receive the incoming message, after the received data is stored in the receive FIFO or a CAN bus error occurs. This means that the maximum period of hardware protection is from the beginning of the CRC field to the end of the 7th bit of EOF.
  - If the receive FIFO is not specified to receive the message, after acceptance filter processing.

RFUST[2:0] bits (Receive FIFO Unread Message Number Status)

The RFUST[2:0] bits indicate the number of unread messages in the receive FIFO. The value of the RFUST[2:0] bits initializes to 000b when the RFE bit is set to 0.

RFMLF flag (Receive FIFO Message Lost Flag)

The RFMLF flag is set to 1 (receive FIFO message lost) when the receive FIFO receives a new message and is full. It is set to 1 at the end of the 6th bit of EOF.

The RFMLF flag is set to 0 through a software write. Writing 1 has no effect. In both overwrite and overrun modes, if the receive FIFO is full and determined to have received a message, the RFMLF flag cannot be set to 0 (receive FIFO message was not lost) through a software write because of hardware protection during the 5 PCLKB cycles following the 6th bit of EOF.

RFFST flag (Receive FIFO Full Status Flag)

The RFFST flag is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. It is 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. The flag is set to 0 when the RFE bit is 0.

RFWST flag (Receive FIFO Buffer Warning Status Flag)

The RFWST flag is set to 1 (receive FIFO is a buffer warning) when the number of unread messages in the receive FIFO is 3. It is 0 (receive FIFO is not a buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. The RFWST flag is set to 0 when the RFE bit is 0.

RFEST flag (Receive FIFO Empty Status Flag)

The RFEST flag is set to 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. It is set to 1 when the RFE bit is set to 0. The flag is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more. Figure 37.2 shows the receive FIFO mailbox operation.

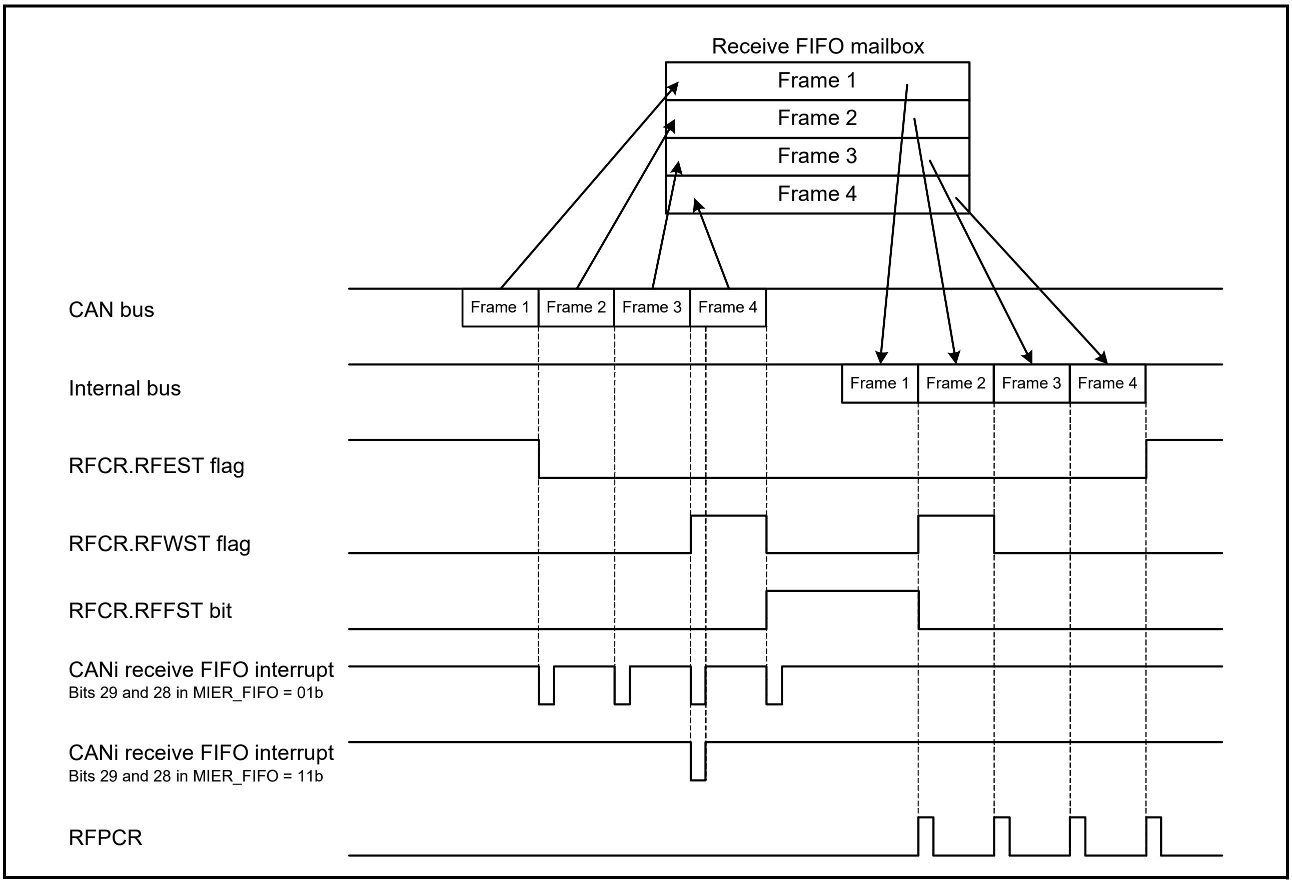


Figure 37.2 Receive FIFO mailbox operation when bits [29] and [28] in MIER_FIFO = 01b or 11b

37.2.12 Receive FIFO Pointer Control Register (RFPCR)

Address(es): CAN0.RFPCR 4005 0849h, CAN1.RFPCR 4005 1849h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU-side pointer for the receive FIFO is incremented by writing FFh to RFPCR.	W

When the receive FIFO is not empty, write FFh to RFPCR through the software to increment the CPU pointer to the next mailbox location. Do not write to RFPCR when the RFE bit in RFCR is 0 (receive FIFO disabled).

Both the CAN and CPU pointers increment when a new message is received and the RFFST flag is 1 (receive FIFO is full) in overwrite mode. When the RFMLF flag is 1 in this state, the CPU pointer does not increment on a software write to RFPCR.

37.2.13 Transmit FIFO Control Register (TFCR)

Address(es): CAN0.TFCR 4005 084Ah, CAN1.TFCR 4005 184Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	TFEST	TFFST	—	—	TFUST[2:0]			TFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TFE	Transmit FIFO Enable	0: Disable transmit FIFO 1: Enable transmit FIFO.	R/W
b3 to b1	TFUST[2:0]	Transmit FIFO Unsent Message Number Status	b3 b1 0 0 0: 0 unsent messages 0 0 1: 1 unsent message 0 1 0: 2 unsent messages 0 1 1: 3 unsent messages 1 0 0: 4 unsent messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved.	R
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	TFFST	Transmit FIFO Full Status	0: Transmit FIFO not full 1: Transmit FIFO full (4 unsent messages).	R
b7	TFEST	Transmit FIFO Empty Status	0: Unsent message in transmit FIFO 1: No unsent message in transmit FIFO.	R

Write to TFCR in CAN operation or halt mode.

TFE bit (Transmit FIFO Enable)

Setting the TFE bit set to 1 enables the transmit FIFO. Setting the TFE bit to 0 empties the transmit FIFO (TFEST bit = 1), and unsent messages in the transmit FIFO are lost in the following ways:

- Immediately if a message from the transmit FIFO is not scheduled for the next transmission or is already in transmission
- On completion of transmission, on a CAN bus error, CAN bus arbitration-lost, or entry to CAN halt mode, if a message from the transmit FIFO is scheduled for the next transmission or is already in transmission.

Before setting the TFE bit to 1 again, ensure that the TFEST bit is set to 1. After setting the TFE bit to 1, write transmit data to mailbox 24.

Do not set the TFE bit to 1 in normal mailbox mode (MBM bit in CTLR = 0).

TFUST[2:0] bits (Transmit FIFO Unsent Message Number Status)

The TFUST[2:0] bits indicate the number of unsent messages in the transmit FIFO. They are set to 000b after TFE bit is set to 0 and transmission aborts or completes.

TFFST bit (Transmit FIFO Full Status)

The TFFST bit is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. The TFFST bit is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. The TFFST bit is set to 0 when transmission from the transmit FIFO is aborted.

TFEST bit (Transmit FIFO Empty Status)

The TFEST bit is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. The TFEST bit is set to 1 when transmission from the transmit FIFO is aborted. The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 37.3 shows the transmit FIFO mailbox operation.

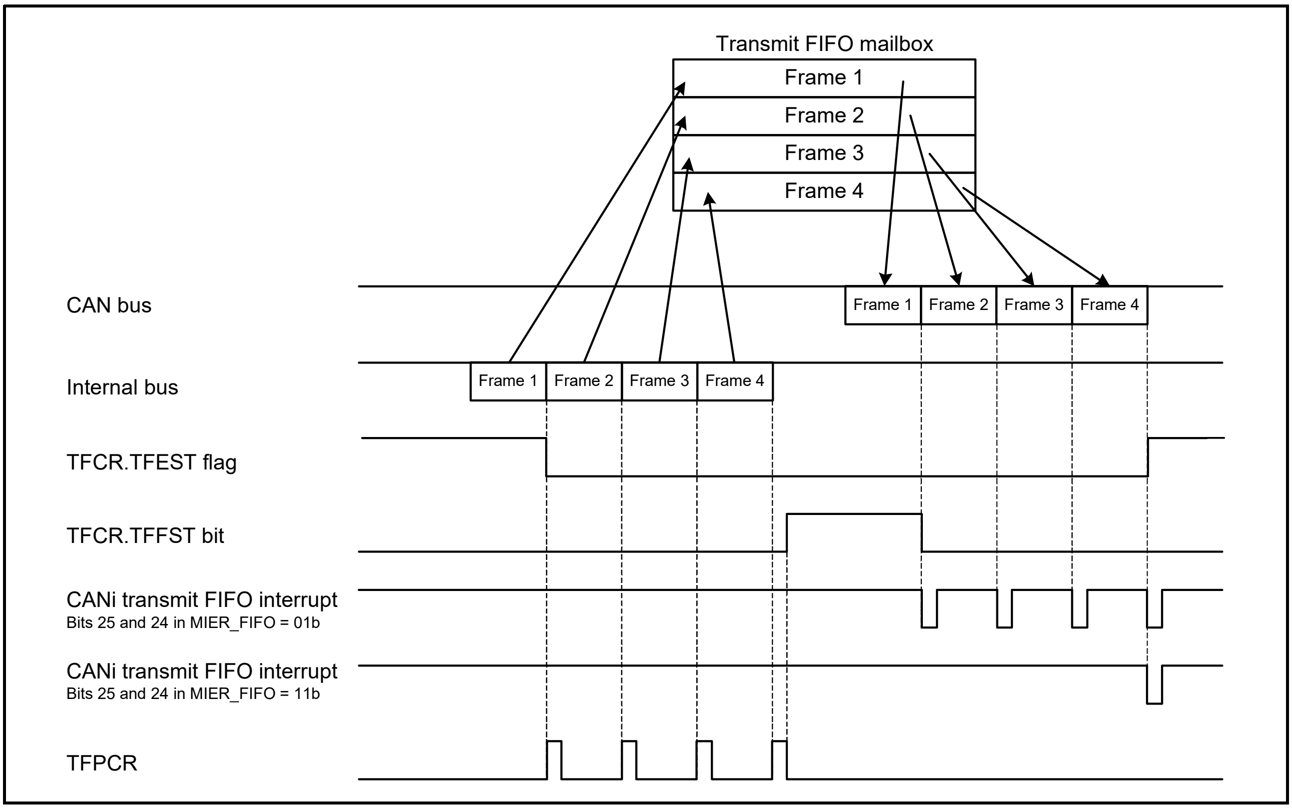


Figure 37.3 **Transmit FIFO mailbox operation when bits [25] and [24] in MIER_FIFO = 01b or 11b**

37.2.14 Transmit FIFO Pointer Control Register (TFPCR)

Address(es): CAN0.TFPCR 4005 084Bh, CAN1.TFPCR 4005 184Bh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU pointer for the transmit FIFO is incremented by writing FFh to TFPCR.	W

When the transmit FIFO is not full, write FFh to TFPCR through the software to increment the CPU pointer for the transmit FIFO to the next mailbox location.

Do not write to TFPCR when the TFE bit in TFCR is 0 (transmit FIFO disabled).

37.2.15 Status Register (STR)

Address(es): CAN0.STR 4005 0842h, CAN1.STR 4005 1842h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST	EST	TABST	FMLST	NMLST	TFST	RFST	SDST	NDST
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	NDST	NEWDATA Status Flag	0: No mailbox with NEWDATA = 1 1: 1 or more mailboxes with NEWDATA = 1.	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA = 1 1: 1 or more mailboxes with SENTDATA = 1.	R
b2	RFST	Receive FIFO Status Flag	0: Empty receive FIFO 1: Message in receive FIFO.	R
b3	TFST	Transmit FIFO Status Flag	0: Transmit FIFO is full 1: Transmit FIFO is not full.	R
b4	NMLST	Normal Mailbox Message Lost Status Flag	0: No mailbox with MSGLOST = 1 1: 1 or more mailboxes with MSGLOST = 1.	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF = 0 1: RFMLF = 1.	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT = 1 1: 1 or more mailboxes with TRMABT = 1.	R
b7	EST	Error Status Flag	0: No error occurred 1: Error occurred.	R
b8	RSTST	CAN Reset Status Flag	0: Not in CAN reset mode 1: In CAN reset mode.	R
b9	HLTST	CAN Halt Status Flag	0: Not in CAN halt mode 1: In CAN halt mode.	R
b10	SLPST	CAN Sleep Status Flag	0: Not in CAN sleep mode 1: In CAN sleep mode.	R
b11	EPST	Error-Passive Status Flag	0: Not in error-passive state 1: In error-passive state.	R
b12	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state.	R
b13	TRMST	Transmit Status Flag	0: Bus idle or reception in progress 1: Transmission in progress or module in bus-off state.	R
b14	RECST	Receive Status Flag	0: Bus idle or transmission in progress 1: Reception in progress.	R
b15	—	Reserved	This bit is read as 0.	R

NDST flag (NEWDATA Status Flag)

The NDST flag is set to 1 when at least one NEWDATA flag in MCTL_RXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. It is set to 0 when all NEWDATA flags are 0.

SDST flag (SENTDATA Status Flag)

The SDST flag is set to 1 when at least one SENTDATA flag in MCTL_TXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. It is set to 0 when all SENTDATA flags are 0.

RFST flag (Receive FIFO Status Flag)

The RFST flag is set to 1 when the receive FIFO is not empty. It is set to 0 when the receive FIFO is empty or normal mailbox mode is selected.

TFST flag (Transmit FIFO Status Flag)

The TFST flag is set to 1 when the transmit FIFO is not full. It is set to 0 when the transmit FIFO is full or normal mailbox mode is selected.

NMLST flag (Normal Mailbox Message Lost Status Flag)

The NMLST flag is set to 1 when at least one MSGLOST flag in MCTL_RXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. It is set to 0 when all MSGLOST flags are 0.

FMLST flag (FIFO Mailbox Message Lost Status Flag)

The FMLST flag is set to 1 when the RFMLF flag in RFCR is 1, regardless of the value of MIER_FIFO. It is set to 0 when the RFMLF flag is 0.

TABST flag (Transmission Abort Status Flag)

The TABST flag is set to 1 when at least one TRMABT flag in MCTL_TXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. It is set to 0 when all TRMABT flags are 0.

EST flag (Error Status Flag)

The EST flag is set to 1 when at least one error is detected by EIFR, regardless of the value of EIER. It is set to 0 when no error is detected by EIFR.

RSTST flag (CAN Reset Status Flag)

The RSTST flag is set to 1 when the CAN module is in CAN reset mode. It is 0 when the CAN module is not in CAN reset mode. It remains 1, even when the state changes from CAN reset to sleep mode.

HLTST flag (CAN Halt Status Flag)

The HLTST flag is set to 1 when the CAN module is in CAN halt mode. It is set to 0 when the CAN module is not in CAN halt mode. It remains 1, even when the state changes from CAN halt to sleep mode.

SLPST flag (CAN Sleep Status Flag)

The SLPST flag is set to 1 when the CAN module is in CAN sleep mode. It is set to 0 when the CAN module is not in CAN sleep mode.

EPST flag (Error-Passive Status Flag)

The EPST flag is set to 1 when the value of TECR or RECR exceeds 127 and the CAN module is in an error-passive state ($128 \leq TEC < 256$ or $128 \leq REC < 256$). It is set to 0 when the CAN module is not in the error-passive state.

BOST flag (Bus-Off Status Flag)

The BOST flag is set to 1 when the value of TECR exceeds 255 and the CAN module is in the bus-off state ($TEC \geq 256$). It is set to 0 when the CAN module is not in the bus-off state.

TRMST flag (Transmit Status Flag)

The TRMST flag is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. It is set to 0 when the CAN module performs as a receiver node or is in the bus-idle state.

RECST flag (Receive Status Flag)

The RECST flag is set to 1 when the CAN module performs as a receiver node. It is set to 0 when the CAN module performs as a transmitter node or is in the bus-idle state.

37.2.16 Mailbox Search Mode Register (MSMR)

Address(es): CAN0.MSMR 4005 0853h, CAN1.MSMR 4005 1853h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MBSM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	MBSM[1:0]	Mailbox Search Mode Select	b1 b0 0 0: Receive mailbox search mode 0 1: Transmit mailbox search mode 1 0: Message lost search mode 1 1: Channel search mode.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Write to MSMR in CAN operation or halt mode.

MBSM[1:0] bits (Mailbox Search Mode Select)

The MBSM[1:0] bits select the search mode for the mailbox search function.

When the MBSM[1:0] bits are 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA flag in MCTL_RXj (j = 0 to 31) for the normal mailbox and the RFEST bit in RFCR.

When the MBSM[1:0] bits are 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA flag in MCTL_TXj.

When the MBSM[1:0] bits are 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST flag in MCTL_RXj for the normal mailbox and the RFMLF flag in RFCR.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is CSSR. See section 37.2.18, Channel Search Support Register (CSSR).

37.2.17 Mailbox Search Status Register (MSSR)

Address(es): CAN0.MSSR 4005 0852h, CAN1.MSSR 4005 1852h

	b7	b6	b5	b4	b3	b2	b1	b0
	SEST	—	—	MBNST[4:0]				
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	MBNST[4:0]	Search Result Mailbox Number Status	These bits output the smallest mailbox number that is found in each mode of MSMR.	R
b6, b5	—	Reserved	These bits are read as 0.	R
b7	SEST	Search Result Status	0: Search result found 1: No search result.	R

MBNST[4:0] bits (Search Result Mailbox Number Status)

In all MSMR modes, the MBNST[4:0] bits output the smallest found mailbox number. In receive mailbox search mode, transmit mailbox search mode, and message lost search mode, the value of the mailbox (the search result to be output) is updated under the following conditions:

- When the NEWDATA, SENTDATA, or MSGLOST flag for a mailbox output by MBNST is set to 0
- When the NEWDATA, SENTDATA, or MSGLOST flag for a mailbox with a smaller number than of MBNST is

set to 1.

If the MBSM[1:0] bits are set to 00b (receive mailbox search mode) or 10b (message lost search mode), the receive FIFO (mailbox 28) is output when it is not empty and there are no unread received messages and no lost messages in any of the normal mailboxes (0 to 23). If the MBSM[1:0] bits are set to 01b (transmit mailbox search mode), the transmit FIFO (mailbox 24) is not output. Table 37.6 lists the behavior of the MBNST[4:0] bits in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] bits output the associated channel number. After MSSR is read by software, the next target channel number is output.

SEST bit (Search Result Status)

The SEST bit is set to 1 (no search result) when no associated mailbox is found after searching all mailboxes. For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA flag for the mailboxes is 1. The SEST bit is set to 0 when at least one SENTDATA flag is 1. When the SEST bit is 1, the value of the MBNST[4:0] bits is undefined.

Table 37.6 Behavior of MBNST[4:0] bits in FIFO mailbox mode

MBSM[1:0] bits	Mailbox 24 (transmit FIFO)	Mailbox 28 (receive FIFO)
00b	Mailbox 24 is not output.	Mailbox 28 is output when no MCTL_RXj.NEWDATA flag for the normal mailboxes is set to 1 (new message is being stored or was stored in the mailbox) and the receive FIFO is not empty.
01b		Mailbox 28 is not output.
10b		Mailbox 28 is output when no MCTL_RXj.MSGLOST flag for the normal mailboxes is set to 1 (message is overwritten or overrun) and the RFCR.RFMLF flag is set to 1 (receive FIFO message lost) in the receive FIFO.
11b		Mailbox 28 is not output.

37.2.18 Channel Search Support Register (CSSR)

Address(es): CAN0.CSSR 4005 0851h, CAN1.CSSR 4005 1851h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	When the value for the channel search is input, the channel number is output to MSSR.	R/W

The bits in CSSR, which are set to 1, are encoded by an 8/3 encoder (the LSB position has the higher priority) and output to the MBNST[4:0] bits in MSSR. MSSR outputs the updated value whenever MSSR is read by software.

Write to CSSR only when the MSMR.MBSM[1:0] bits are 11b (channel search mode). Write to CSSR in CAN operation mode or CAN halt mode.

Figure 37.4 shows writes to and reads from CSSR and MSSR.

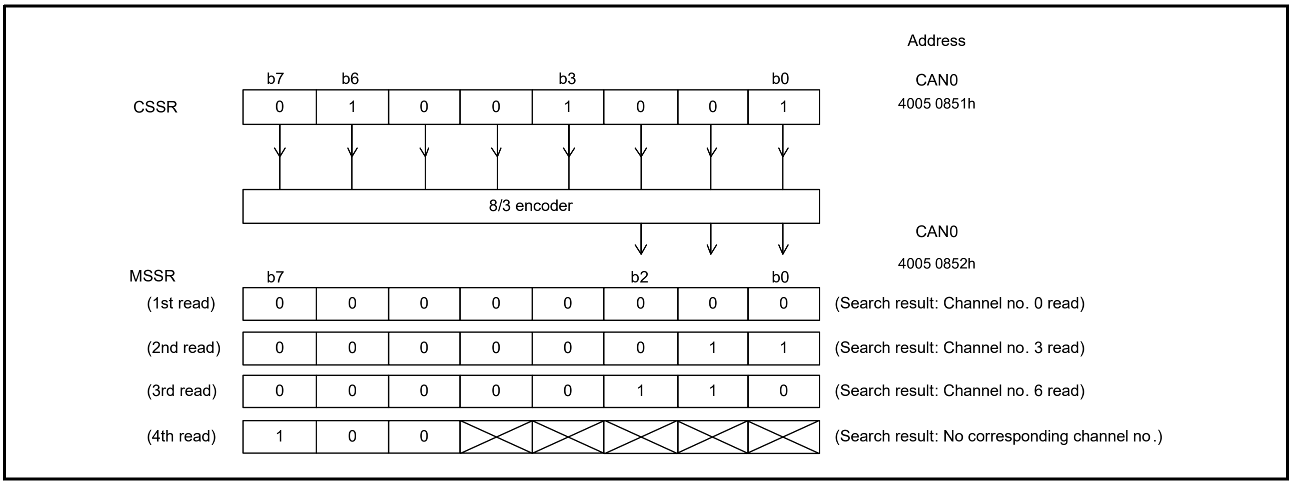


Figure 37.4 Writes to and reads from CSSR and MSSR

The value of CSSR is also updated whenever MSSR is read. On this read, the value prior to conversion by the 8/3 encoder can be read.

37.2.19 Acceptance Filter Support Register (AFSR)

Address(es): CAN0.AFSR 4005 0856h, CAN1.AFSR 4005 1856h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b15 to b0	After the standard ID of a received message is written, the value converted for data table search can be read.	R/W

Note: Write to AFSR in CAN operation mode or CAN halt mode.

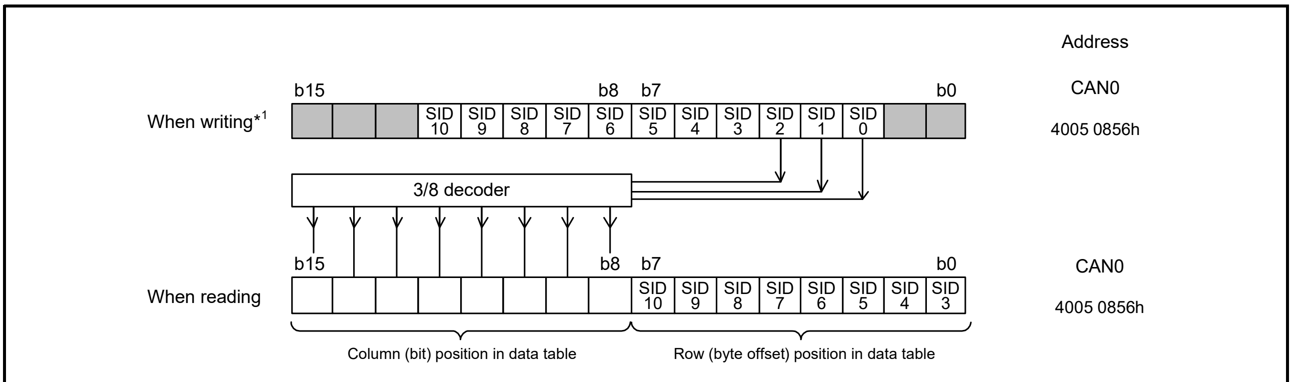
The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) searches. In the data table, all standard IDs that you create are set as valid or invalid in bit units. When AFSR is written with data in 16-bit units including the SID[10:0] bits in MBj_ID (j = 0 to 31), in which a received standard ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.

The ASU is enabled in the following cases:

- When the ID to be received cannot be masked by the acceptance filter. For example, if IDs to be received are 078h, 087h, and 111h
- When there are too many IDs to receive, and the software filtering time is expected to be shortened.

Note: AFSR cannot be set in CAN reset mode.

Figure 37.5 shows the writes to and reads from AFSR.



Note 1. Write the same value as the 16-bit unit data, including the SID[10:0] bits in MBj_ID (j = 0 to 31).

Figure 37.5 Writes to and reads from AFSR

37.2.20 Error Interrupt Enable Register (EIER)

Address(es): CAN0.EIER 4005 084Ch, CAN1.EIER 4005 184Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BEIE	Bus Error Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b1	EWIE	Error-Warning Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b2	EPIE	Error-Passive Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b3	BOEIE	Bus-Off Entry Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b4	BORIE	Bus-Off Recovery Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b5	ORIE	Overrun Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b6	OLIE	Overload Frame Transmit Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b7	BLIE	Bus Lock Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W

EIER enables or disables each error interrupt source independently in EIFR. Write to EIER in CAN reset mode.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE bit is 0, no error interrupt request occurs even if the BEIF bit in EIFR is 1. When the BEIE bit is 1, an error interrupt request occurs if the BEIF bit is set to 1.

EWIE bit (Error-Warning Interrupt Enable)

When the EWIE bit is 0, no error interrupt request occurs even if the EWIF bit in EIFR is 1. When the EWIE bit is 1, an error interrupt request is generated if the EWIF bit is set to 1.

EPIE bit (Error-Passive Interrupt Enable)

When the EPIE bit is 0, no error interrupt request occurs even if the EPIF bit in EIFR is 1. When the EPIE bit is 1, an error interrupt request occurs if the EPIF bit is set to 1.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE bit is 0, no error interrupt request occurs even if the BOEIF bit in EIFR is 1. When the BOEIE bit is 1, an error interrupt request occurs if the BOEIF bit is set to 1.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE bit is 0, no error interrupt request occurs even if the BORIF bit in EIFR is 1. When the BORIE bit is 1, an error interrupt request occurs if the BORIF bit is set to 1.

ORIE bit (Overrun Interrupt Enable)

When the ORIE bit is 0, no error interrupt request occurs even if the ORIF bit in EIFR is 1. When the ORIE bit is 1, an error interrupt request occurs if the ORIF bit is set to 1.

OLIE bit (Overload Frame Transmit Interrupt Enable)

When the OLIE bit is 0, no error interrupt request occurs even if the OLIF bit in EIFR is 1. When the OLIE bit is 1, an error interrupt request occurs if the OLIF bit is set to 1.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE bit is 0, no error interrupt request occurs even if the BLIF bit in EIFR is 1. When the BLIE bit is 1, an error interrupt request occurs if the BLIF bit is set to 1.

37.2.21 Error Interrupt Factor Judge Register (EIFR)

Address(es): CAN0.EIFR 4005 084Dh, CAN1.EIFR 4005 184Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BEIF	Bus Error Detect Flag	0: No bus error detected 1: Bus error detected.	R/W
b1	EWIF	Error-Warning Detect Flag	0: No error-warning detected 1: Error-warning detected.	R/W
b2	EPIF	Error-Passive Detect Flag	0: No error-passive detected 1: Error-passive detected.	R/W
b3	BOEIF	Bus-Off Entry Detect Flag	0: No bus-off entry detected 1: Bus-off entry detected.	R/W
b4	BORIF	Bus-Off Recovery Detect Flag	0: No bus-off recovery detected 1: Bus-off recovery detected.	R/W
b5	ORIF	Receive Overrun Detect Flag	0: No receive overrun detected 1: Receive overrun detected.	R/W
b6	OLIF	Overload Frame Transmission Detect Flag	0: No overload frame transmission detected 1: Overload frame transmission detected.	R/W
b7	BLIF	Bus Lock Detect Flag	0: No bus lock detected 1: Bus lock detected.	R/W

If an event associated with one of these bits occurs, the associated bit in EIFR is set to 1, regardless of the setting of EIER.

Clear the bits to 0 through a software write. If a bit is set to 1 at the same time that the software clears it, it becomes 1. When setting a single bit to 0 in the software, use the transfer instruction (MOV) to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

BEIF flag (Bus Error Detect Flag)

The BEIF flag is set to 1 when a bus error is detected.

EWIF flag (Error-Warning Detect Flag)

The EWIF flag is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95. It is set to 1 only when REC or TEC initially exceeds 95. If 0 is written to the EWIF flag by software while REC or TEC remains greater than 95, the EWIF flag does not set to 1 until REC or TEC goes below 95 and then exceeds 95 again.

EPIF flag (Error-Passive Detect Flag)

The EPIF flag is set to 1 when the CAN error state becomes error-passive, when the receive error counter (REC) or transmit error counter (TEC) value exceeds 127. It is set to 1 only when REC or TEC initially exceeds 127. If 0 is written to the EPIF flag by software while REC or TEC remains greater than 127, the flag does not set to 1 until REC or TEC goes below 127 and then exceeds 127 again.

BOEIF flag (Bus-Off Entry Detect Flag)

The BOEIF flag is set to 1 when the CAN error state becomes bus-off, when the transmit error counter (TEC) value exceeds 255. It also is set to 1 when the BOM[1:0] bits in CTLR are 01b (automatic entry to CAN halt mode on bus-off entry) and the CAN module becomes the bus-off state.

BORIF flag (Bus-Off Recovery Detect Flag)

The BORIF flag is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive recessive bits 128 times in the following conditions:

- When the BOM[1:0] bits in CTLR are 00b
- When the BOM[1:0] bits in CTLR are 10b
- When the BOM[1:0] bits in CTLR are 11b.

However, the BORIF flag does not set to 1 if the CAN module recovers from the bus-off state in the following conditions:

- When the CANM[1:0] bits in CTLR are set to 01b or 11b (CAN reset mode)
- When the RBOC bit in CTLR is set to 1 (forced return from bus-off)
- When the BOM[1:0] bits in CTLR are set to 01b
- When the BOM[1:0] bits in CTLR are set to 11b and the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) before normal recovery occurs.

Table 37.7 lists the behavior of the BOEIF and BORIF bits for each CTLR.BOM[1:0] bit setting.

Table 37.7 Behavior of BOEIF and BORIF flags for each CTLR.BOM[1:0] setting

BOM[1:0] bits	BOEIF bit	BORIF bit
00b	Set to 1 on entry to the bus-off state	Sets to 1 on exit from the bus-off state
01b		Does not set to 1
10b		Sets to 1 on exit from the bus-off state
11b		Sets to 1 if normal bus-off recovery occurs before the CANM[1:0] bits are set to 10b (CAN halt mode)

ORIF flag (Receive Overrun Detect Flag)

The ORIF flag is set to 1 when a receive overrun occurs. It does not set to 1 in overwrite mode.

In overwrite mode, a reception complete interrupt request occurs if an overwrite condition occurs and the ORIF bit is not set to 1.

In overrun mode with normal mailbox mode, if an overrun occurs in any of mailboxes 0 to 31, this flag is set to 1. In overrun mode with FIFO mailbox mode, if an overrun occurs in any of mailboxes 0 to 23 or the receive FIFO, this bit is set to 1.

OLIF flag (Overload Frame Transmission Detect Flag)

The OLIF flag is set to 1 if the transmitting condition of an overload frame is detected when the CAN module is transmitting or receiving.

BLIF flag (Bus Lock Detect Flag)

The BLIF flag is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode. After the BLIF flag is set to 1, 32 consecutive dominant bits are detected again under either of the following conditions:

- Recessive bits are detected after this flag changes to 0 from 1
- The CAN module enters CAN reset or halt mode and then enters CAN operation mode again after this flag changes to 0 from 1.

37.2.22 Receive Error Count Register (RECR)

Address(es): CAN0.RECR 4005 084Eh, CAN1.RECR 4005 184Eh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Description	R/W
b7 to b0	Receive error count function RECR increments or decrements the counter value based on the error status of the CAN module during reception.	R

RECR indicates the value of the receive error counter. See the CAN specification (ISO11898-1) for the increment and decrement conditions of the receive error counter. The value of RECR in the bus-off state is undefined.

37.2.23 Transmit Error Count Register (TECR)

Address(es): CAN0.TECR 4005 084Fh, CAN1.TECR 4005 184Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Description	R/W
b7 to b0	Transmit error count function TECR increments or decrements the counter value based on the error status of the CAN module during transmission.	R

TECR indicates the value of the transmit error counter. See the CAN specification (ISO11898-1) for the increment and decrement conditions of the transmit error counter. The value of TECR in the bus-off state is undefined.

37.2.24 Error Code Store Register (ECSR)

Address(es): CAN0.ECSR 4005 0850h, CAN1.ECSR 4005 1850h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SEF	Stuff Error Flag*1,*2	0: No stuff error detected 1: Stuff error detected.	R/W
b1	FEF	Form Error Flag*1,*2	0: No form error detected 1: Form error detected.	R/W
b2	AEF	ACK Error Flag*1,*2	0: No ACK error detected 1: ACK error detected.	R/W
b3	CEF	CRC Error Flag*1,*2	0: No CRC error detected 1: CRC error detected.	R/W
b4	BE1F	Bit Error (recessive) Flag*1,*2	0: No bit error (recessive) detected 1: Bit error (recessive) detected.	R/W
b5	BE0F	Bit Error (dominant) Flag*1,*2	0: No bit error (dominant) detected 1: Bit error (dominant) detected.	R/W
b6	ADEF	ACK Delimiter Error Flag*1,*2	0: No ACK delimiter error detected 1: ACK delimiter error detected.	R/W
b7	EDPM	Error Display Mode Select*3,*4	0: Output first detected error code 1: Output accumulated error code.	R/W

Note 1. Writing 1 has no effect on these bit values.
Note 2. To write 0 to the SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF bits, use the transfer (MOV) instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.
Note 3. Write to the EDPM bit in CAN reset or halt mode.
Note 4. If more than one error condition is detected simultaneously, all related bits are set to 1.

ECSR indicates whether an error occurs on the CAN bus. See the CAN specification (ISO11898-1) for the conditions when each error occurs.

Clear all of the bits except for the EDPM bit to 0 through a software write. If a bit is set to 1 at the same time that the software clears it, it becomes 1.

SEF flag (Stuff Error Flag)

The SEF flag is set to 1 when a stuff error is detected.

FEF flag (Form Error Flag)

The FEF flag is set to 1 when a form error is detected.

AEF flag (ACK Error Flag)

The AEF flag is set to 1 when an ACK error is detected.

CEF flag (CRC Error Flag)

The CEF flag is set to 1 when a CRC error is detected.

BE1F flag (Bit Error (recessive) Flag)

The BE1F flag is set to 1 when a recessive bit error is detected.

BE0F flag (Bit Error (dominant) Flag)

The BE0F flag is set to 1 when a dominant bit error is detected.

ADEF flag (ACK Delimiter Error Flag)

The ADEF flag is set to 1 when a form error is detected with the ACK delimiter during transmission.

EDPM bit (Error Display Mode Select)

The EDPM bit selects the output mode of ECSR. When the EDPM bit is set to 0, ECSR outputs the first error code. When the EDPM bit is set to 1, ECSR outputs the accumulated error code.

37.2.25 Time Stamp Register (TSR)

Address(es): CAN0.TSR 4005 0854h, CAN1.TSR 4005 1854h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Description	R/W
b15 to b0	Free-running counter value for the time stamp function	R

Note: Read TSR in 16-bit units.

When TSR is read, the value of the time stamp counter (16-bit free-running counter) at that moment is read. The time stamp counter reference clock is configured in the TSPS[1:0] bits in CTLR. The counter stops in CAN sleep and halt modes, and is initialized in CAN reset mode. Its value is stored in the TSL[7:0] and TSH[7:0] bits in MBj_TS when a received message is stored in a receive mailbox.

37.2.26 Test Control Register (TCR)

Address(es): CAN0.TCR 4005 0858h, CAN1.TCR 4005 1858h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TSTM[1:0]		TSTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TSTE	CAN Test Mode Enable	0: Disable CAN test mode 1: Enable CAN test mode.	R/W
b2, b1	TSTM[1:0]	CAN Test Mode Select	b2 b1 0 0: Not CAN test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback) 1 1: Self-test mode 1 (internal loopback).	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCR controls the CAN test mode. Write to TCR in CAN halt mode only.

(1) Listen-only mode

The CAN specification (ISO11898-1) recommends an optional bus monitoring mode. In listen-only mode, valid data frames and valid remote frames can be received. However, only recessive bits can be sent on the CAN bus. The ACK bit, overload flag, and active error flag cannot be sent.

Listen-only mode can be used for baud rate detection.

Do not request transmission from any mailboxes in listen-only mode.

Figure 37.6 shows the connection when listen-only mode is selected.

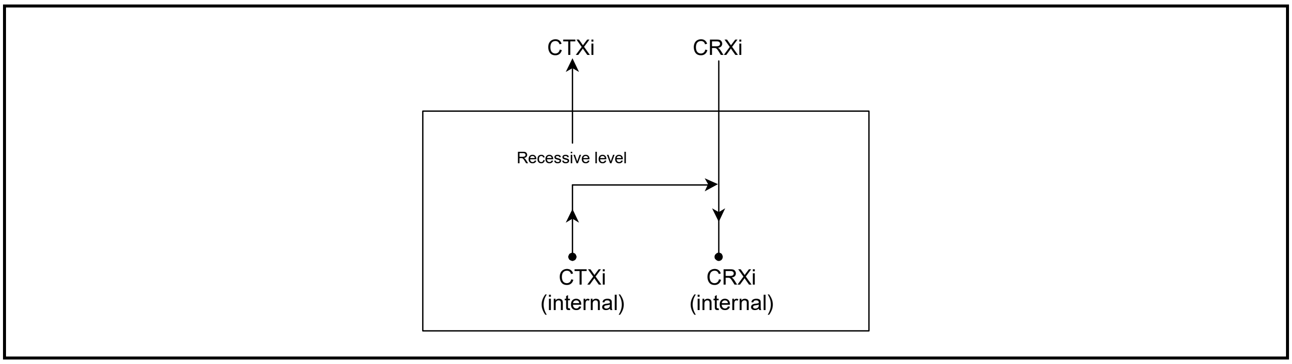


Figure 37.6 Connection when listen-only mode is selected (i = 0, 1)

(2) Self-test mode 0 (external loopback)

Self-test mode 0 is provided for CAN transceiver tests. In this mode, the protocol module treats its own transmitted messages as those received by the CAN transceiver and stores them into the receive mailbox. To be independent from external stimulation, the protocol module generates the ACK bit.

Connect the CTXi and CRXi pins to the transceiver.

Figure 37.7 shows the connection when self-test mode 0 is selected.

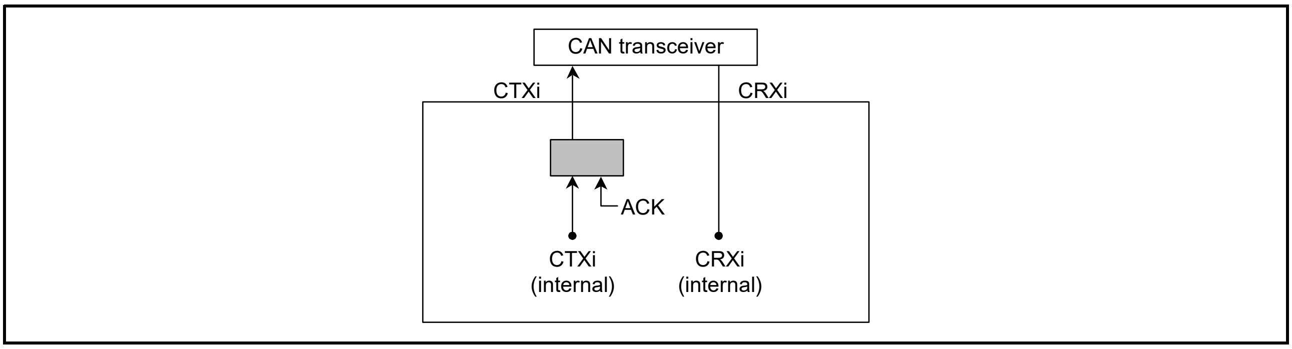


Figure 37.7 Connection when self-test mode 0 is selected (i = 0, 1)

(3) Self-test mode 1 (internal loopback)

Self-test mode 1 is provided for self-test functions.

In self-test mode 1, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

In self-test mode 1, the protocol controller performs an internal feedback from the internal CTXi pin to the internal CRXi pin. The input value of the external CRXi pin is ignored. The external CTXi pin outputs only recessive bits. The CTXi and CRXi pins are not required to be connected to the CAN bus or any external device.

Figure 37.8 shows the connection when self-test mode 1 is selected.

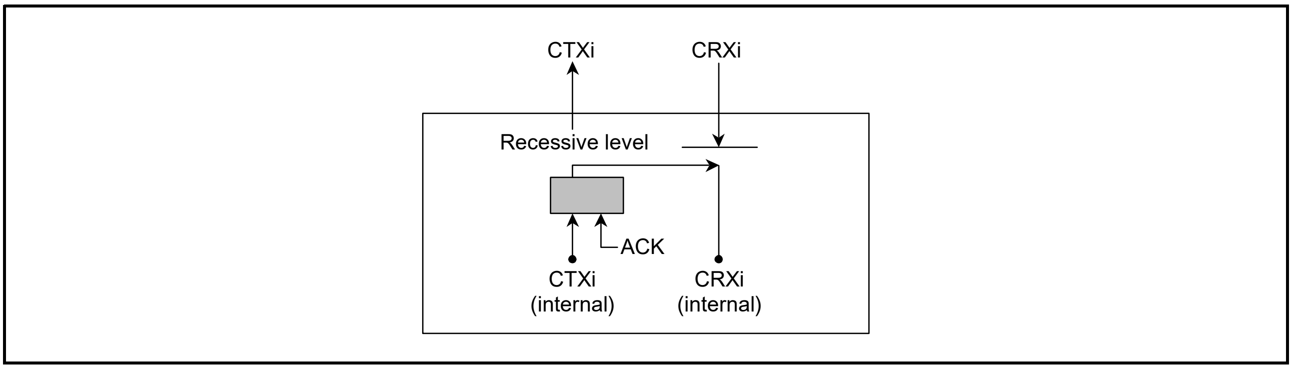


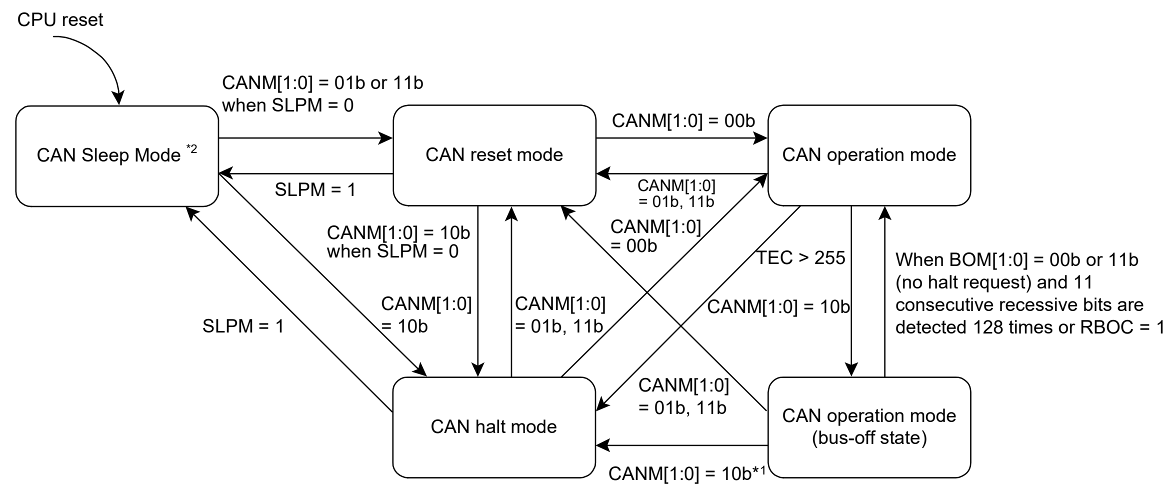
Figure 37.8 Connection when self-test mode 1 is selected (i = 0, 1)

37.3 Operation Modes

The CAN module operation modes include:

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode.

Figure 37.9 shows the transitions between the operation modes.



CANM, SLPM, BOM, RBOC: Bits in CTLR

Note 1. The transition timing from the bus-off state to CAN halt mode depends on the setting of the CTLR.BOM[1:0] bits.
When the CTLR.BOM[1:0] bits are 01b, the state transition occurs immediately after entering the bus-off state.
When the CTLR.BOM[1:0] bits are 10b, the state transition occurs at the end of the bus-off state.
When the CTLR.BOM[1:0] bits are 11b, the state transition occurs at the setting of the CTLR.CANM[1:0] bits to 10b (CAN halt mode).

Note 2. Change the CTLR.SLPM bit to set or cancel CAN Sleep mode.

Figure 37.9 Transition between different operation modes

37.3.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration. When the CTLR.CANM[1:0] bits are set to 01b or 11b, the CAN module enters CAN reset mode. The STR.RSTST bit then is set to 1. Do not change the CTLR.CANM[1:0] bits until the RSTST bit is 1. Set BCR before exiting CAN reset mode to any other modes.

The following registers are initialized to their reset values after entering CAN reset mode, and their initial values are saved during CAN reset mode:

- MCTL_TXj and MCTL_RXj
- STR (except for the SLPST and TFST bits)
- EIFR
- RECR
- TECR
- TSR
- MSSR
- MSMR
- RFCR
- TFCR
- TCR
- ECSR (except for the EDPM bit).

The following registers retain their previous values even after entering CAN reset mode:

- CTLR
- STR (only the SLPST and TFST bits)
- MIER and MIER_FIFO
- EIER
- BCR
- CSSR
- ECSR (only the EDPM bit)
- MBj_ID, MBj_DL, MBj_Dm and MBj_TS
- MKRk
- FIDCR0 and FIDCR1
- MKIVLR
- AFSR
- RFPCR
- TFPCR.

37.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.

When the CTLR.CANM[1:0] bits are set to 10b, CAN halt mode is selected. Then the STR.HLTST bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST bit is 1.

See Table 37.8 for the state transition conditions when transmitting or receiving.

All registers except for the RSTST, HLTST, and SLPST bits in STR remain unchanged when the CAN enters CAN halt mode.

Do not change CTLR (except for the CANM[1:0] and SLPM bits) and EIER in CAN halt mode. BCR can be changed in CAN halt mode only when listen-only mode is selected for automatic baud rate detection.

Table 37.8 Operation in CAN reset and halt modes

Operation mode	Receiver	Transmitter	Bus-off
CAN reset mode (forced transition) CANM[1:0] = 11b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode without waiting for the end of message transmission.	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN reset mode CANM[1:0] = 01b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode after waiting for the end of message transmission.*1,*4	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception.*2,*3	CAN module enters CAN halt mode after waiting for the end of message transmission.*1,*4	When the BOM[1:0] bits are 00b: A halt request from software is accepted only after bus-off recovery. When the BOM[1:0] bits are 01b: CAN module automatically enters CAN halt mode without waiting for the end of bus-off recovery, regardless of a halt request from the software. When the BOM[1:0] bits are 10b: CAN module automatically enters CAN halt mode after waiting for the end of bus-off recovery, regardless of a halt request from software. When the BOM[1:0] bits are 11b: CAN module enters CAN halt mode without waiting for the end of bus-off recovery, if a halt is requested by software during bus-off.

BOM[1:0] bits: Bits in CTLR

Note 1. If transmission of multiple messages is requested, a mode transition occurs on completion of the first transmission. If the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.

Note 2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the BLIF bit in EIFR.

Note 3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN module transitions to CAN halt mode.

Note 4. If a CAN bus error or arbitration-lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN module transits to the requested CAN mode.

37.3.3 CAN Sleep Mode

CAN sleep mode reduces power consumption by stopping the clock supply to the CAN module. After a reset from an MCU pin or a software reset, the CAN module starts from CAN sleep mode.

When the SLPM bit in CTLR is set to 1, the CAN module enters CAN sleep mode. Then the SLPST bit in STR is set to 1. Do not change the value of the SLPM bit until the SLPST bit is 1. The other registers remain unchanged when the CAN module enters CAN sleep mode.

Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any registers (except for the SLPM bit) during CAN sleep mode. Read operation is still allowed.

When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.

37.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication.

When the CANM[1:0] bits in CTLR are set to 00b, the CAN module enters CAN operation mode. Then the RSTST and HLTST bits in STR set to 0. Do not change the value of the CANM[1:0] bits until the RSTST and HLTST bits are 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode, the following occurs:

- The CAN module becomes an active node on the network, which enables transmission and reception of CAN messages

- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module might be in one of the following three sub-modes, depending on the status of the CAN bus.

- Idle mode: No transmission or reception occurs
- Receive mode: A CAN message sent by another node is being received
- Transmit mode: A CAN message is being transmitted. The CAN module receives a message transmitted by the local node simultaneously when self-test mode 0 (TSTM[1:0] bits in TCR = 10b) or self-test mode 1 (TSTM[1:0] bits = 11b) is selected.

Figure 37.10 demonstrates the sub-modes in CAN operation mode.

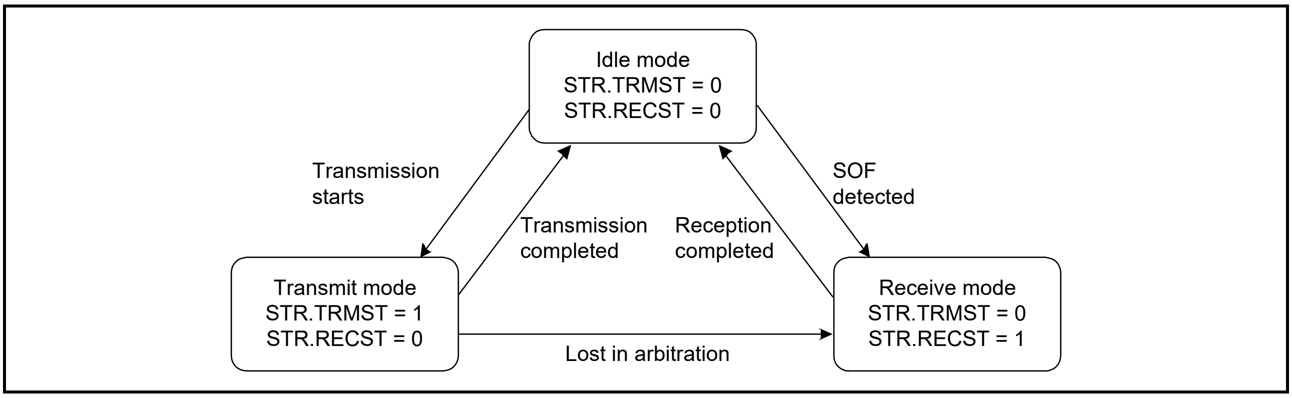


Figure 37.10 Sub-modes of CAN operation mode

37.3.5 CAN Operation Mode (Bus-Off State)

The CAN module enters the bus-off state based on the incrementing decrementing rules for the transmit and error counters defined in the CAN Specifications. The following cases apply when the CAN module is recovering from the bus-off state. When the CAN module is in the bus-off state, the values of the CAN-related registers remain unchanged, except for those in STR, EIFR, RECR, TECR, and TSR.

(1) When CTLR.BOM[1:0] = 00b (normal mode)

The CAN module enters the error-active state after it completes recovery from the bus-off state and CAN communication is enabled. The BORIF flag in EIFR is set to 1 (bus-off recovery detected) at this time.

(2) When CTLR.RBOC = 1 (forced return from bus-off)

The CAN module enters the error-active state when it is in the bus-off state and the RBOC bit is 1. CAN communication is enabled again after 11b consecutive recessive bits are detected. The BORIF bit does not set to 1 at this time.

(3) When CTLR.BOM[1:0] = 01b (automatic transition to CAN halt mode on bus-off)

The CAN module enters CAN halt mode when it reaches the bus-off state. The BORIF flag does not set to 1 at this time.

(4) When CTLR.BOM[1:0] = 10b (automatic transition to CAN halt mode on bus-off end)

The CAN module enters CAN halt mode when it completes recovery from bus-off. The BORIF flag is set to 1 at this time.

(5) When CTLR.BOM[1:0] = 11b (automatic transition to CAN halt mode through software) and CTLR.CANM[1:0] = 10b (CAN halt mode) during bus-off state

The CAN module enters CAN halt mode when it is in the bus-off state and the CANM[1:0] bits are set to 10b (CAN halt mode). The BORIF flag does not set to 1 at this time.

If the CANM[1:0] bits are not set to 10b during bus-off, the same behavior as (1) applies.

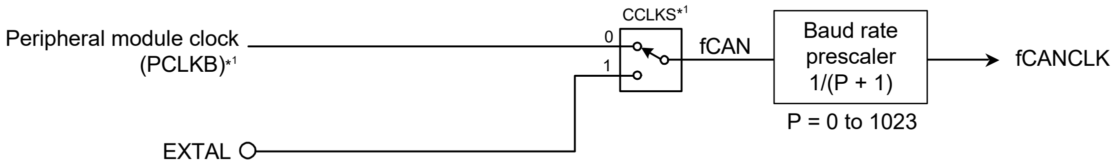
37.4 Data Transfer Rate Configuration

This section describes how to configure the data transfer rate.

37.4.1 Clock Setting

The CAN module has a CAN clock generator that can be set by the CCLKS and the BRP[9:0] bits in the BCR register.

Figure 37.11 shows a block diagram of the CAN clock generator.



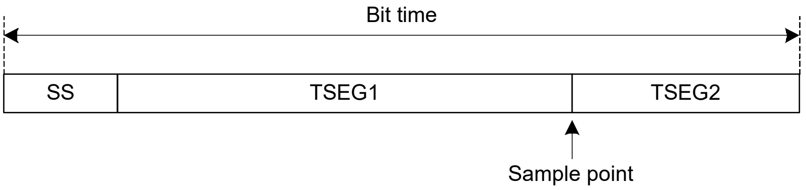
CCLKS:	Bit in the BCR register
fCAN:	CAN system clock
P:	Value selected in BRP[9:0] bits in BCR (P = 0 to 1023)
fCANCLK:	CAN communication clock (fCANCLK = fCAN/(P + 1))

Note 1. See section 9, Clock Generation Circuit. When using the CAN module while the CCLKS bit is cleared to 0, PLL must be selected as the source of the peripheral module clock.

Figure 37.11 Block diagram of CAN clock generator

37.4.2 Bit Timing Setting

The bit time consists of the following three segments shown in Figure 37.12.



The range of each segment: Bit time = 8 Tq to 25 Tq
SS = 1 Tq
TSEG1 = 4 Tq to 16 Tq
TSEG2 = 2 Tq to 8 Tq
SJW = 1 Tq to 4 Tq
Setting of TSEG1 and TSEG2: TSEG1 > TSEG2 ≥ SJW

Figure 37.12 Bit timing

37.4.3 Data Transfer Rate

The data transfer rate depends on the division value of fCAN (CAN system clock), the division value of the baud rate prescaler, and the Tq count for 1 bit time.

$$\text{Data transfer rate (bps)} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value}^{*1} \times \text{number of Tq of 1 bit time}} = \frac{\text{fCANCLK}}{\text{Tq count for 1 bit time}}$$

Note 1. Division value of baud rate prescaler = P + 1 (P: 0 to 1023), where P is the BRP[9:0] setting in BCR.

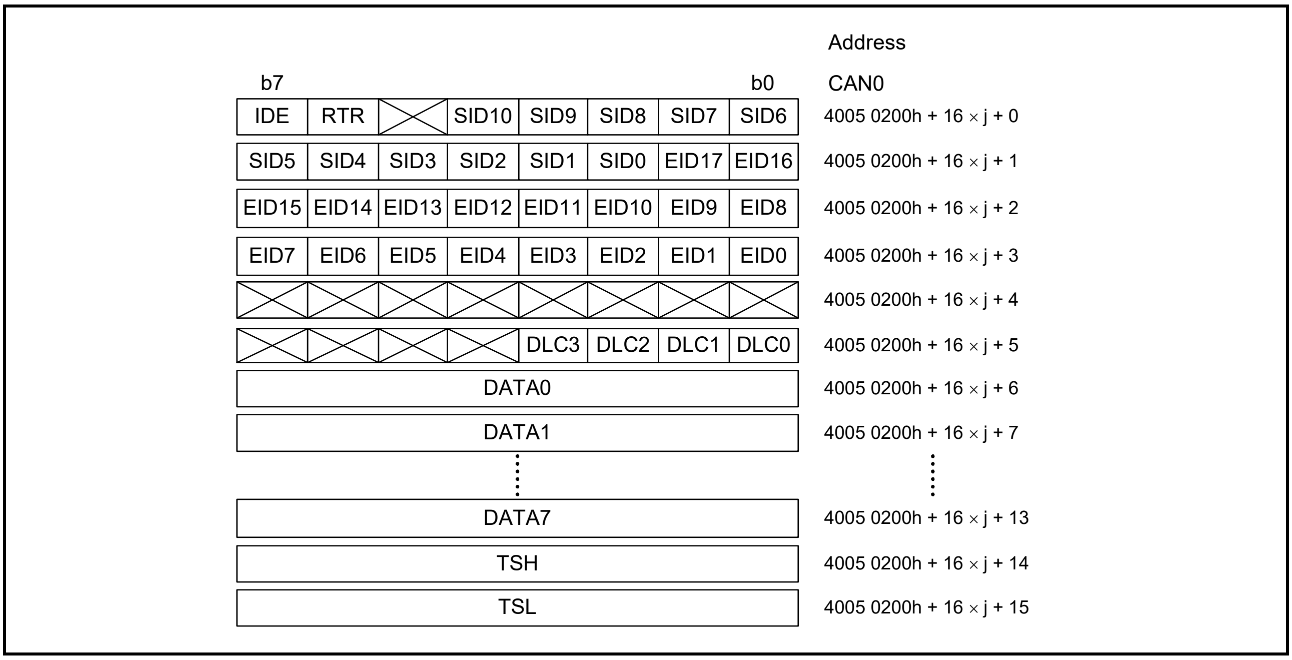
Table 37.9 lists data transfer rate examples.

Table 37.9 Data transfer rate examples

fCAN	50 MHz		48 MHz		40 MHz		32 MHz	
Data transfer rate	Tq count	P + 1	Tq count	P + 1	Tq count	P + 1	Tq count	P + 1
1 Mbps	10 Tq 25 Tq	5 2	8 Tq 12 Tq 16 Tq	6 4 3	10 Tq 20 Tq	4 2	8 Tq 16 Tq	4 2
500 kbps	10 Tq 25 Tq	10 4	8 Tq 12 Tq 16 Tq	12 8 6	10 Tq 20 Tq	8 4	8 Tq 16 Tq	8 4
250 kbps	10 Tq 25 Tq	20 8	8 Tq 12 Tq 16 Tq	24 16 12	10 Tq 20 Tq	16 8	8 Tq 16 Tq	16 8
125 kbps	10 Tq 25 Tq	40 16	8 Tq 12 Tq 16 Tq	48 32 24	10 Tq 20 Tq	32 16	8 Tq 16 Tq	32 16
83.3 kbps	10 Tq 25 Tq	60 24	8 Tq 12 Tq 16 Tq	72 48 36	8 Tq 10 Tq 16 Tq 20 Tq	60 48 30 24	8 Tq 16 Tq	48 24
33.3 kbps	10 Tq 25 Tq	150 60	8 Tq 12 Tq 16 Tq	180 120 90	8 Tq 10 Tq 20 Tq	150 120 60	8 Tq 10 Tq 16 Tq 20 Tq	120 96 60 48

37.5 Mailbox and Mask Register Structure

Figure 37.13 shows the structure of the 32 mailbox registers: MBj_ID, MBj_DL, MBj_Dm, and MBj_TS.



b7							b0	Address CAN0
IDE	RTR		SID10	SID9	SID8	SID7	SID6	4005 0200h + 16 × j + 0
SID5	SID4	SID3	SID2	SID1	SID0	EID17	EID16	4005 0200h + 16 × j + 1
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	4005 0200h + 16 × j + 2
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	4005 0200h + 16 × j + 3
								4005 0200h + 16 × j + 4
				DLC3	DLC2	DLC1	DLC0	4005 0200h + 16 × j + 5
DATA0								4005 0200h + 16 × j + 6
DATA1								4005 0200h + 16 × j + 7
⋮								⋮
DATA7								4005 0200h + 16 × j + 13
TSH								4005 0200h + 16 × j + 14
TSL								4005 0200h + 16 × j + 15

Figure 37.13 Structure of the mailbox registers (j = 0 to 31)

Figure 37.14 shows the structure of the eight mask registers: MKRk.

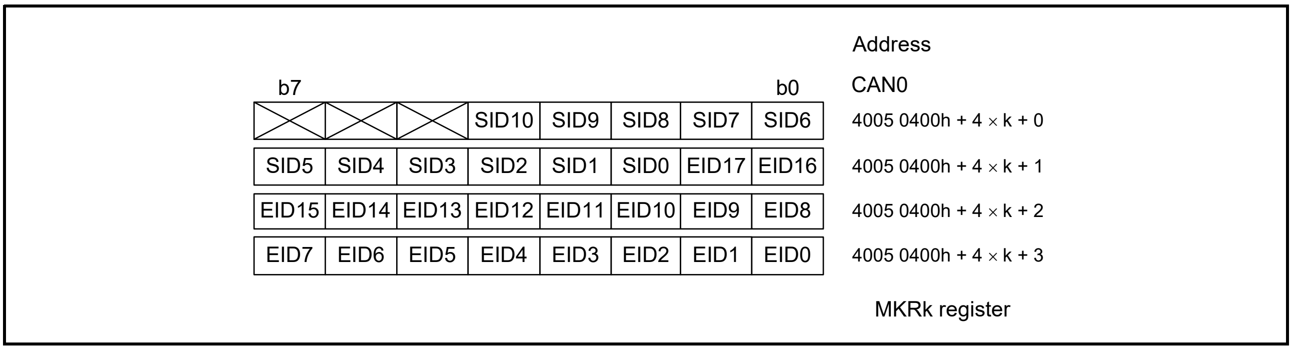


Figure 37.14 Structure of the MKRk registers (k = 0 to 7)

Figure 37.15 shows the structure of the two FIFO receive ID compare registers: FIDCR0 and FIDCR1.

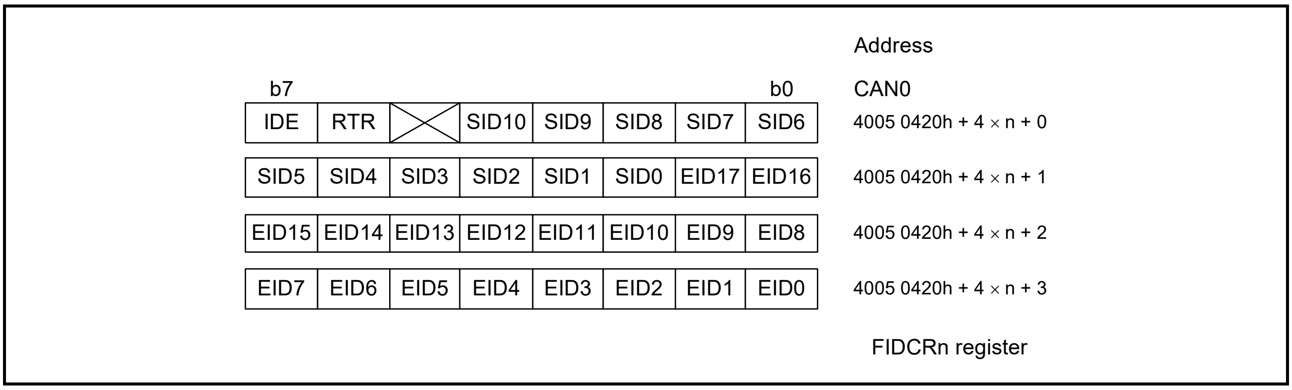


Figure 37.15 Structure of the FIDCRn registers (n = 0, 1)

37.6 Acceptance Filtering and Masking Functions

The acceptance filtering and masking functions allow you to select and receive messages with multiple IDs for mailboxes within a specified range.

The MKRk registers can mask the standard ID and the extended ID for 29 bits.

- MKR0 controls mailboxes 0 to 3
- MKR1 controls mailboxes 4 to 7
- MKR2 controls mailboxes 8 to 11
- MKR3 controls mailboxes 12 to 15
- MKR4 controls mailboxes 16 to 19
- MKR5 controls mailboxes 20 to 23
- MKR6 controls mailboxes 24 to 27 in normal mailbox mode and the receive FIFO mailboxes 28 to 31 in FIFO mailbox mode
- MKR7 controls mailboxes 28 to 31 in normal mailbox mode and the receive FIFO mailboxes 28 to 31 in FIFO mailbox mode.

MKIVLR disables acceptance filtering independently for each mailbox.

The IDE bit in MBj_ID is valid when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode).

The RTR bit in MBj_ID selects a data or remote frame.

In FIFO mailbox mode, the normal mailboxes (0 to 23) use one associated register from MKR0 to MKR5 for acceptance filtering. The receive FIFO mailboxes (28 to 31) use two registers, MKR6 and MKR7, for acceptance filtering.

The receive FIFO also uses two registers, FIDCR0 and FIDCR1, for ID comparison. The EID[17:0], SID[10:0], RTR, and IDE bits in mailbox28 to mailbox31 for the receive FIFO are disabled. As acceptance filtering depends on the result of two logic OR operations, two ranges of IDs can be received into the receive FIFO.

MKIVLR is disabled for the receive FIFO.

If different standard ID and extended ID values are set in the IDE bits in FIDCR0 and FIDCR1, both ID formats are received.

If different data frame and remote frame values are set in the RTR bits in FIDCR0 and FIDCR1, both data and remote frames are received.

When a combination of two ranges of IDs is not necessary, set the same mask value and the same ID into both the FIFO ID and mask registers.

Figure 37.16 shows the associations between the mask registers and mailboxes. Figure 37.17 shows the acceptance filtering.

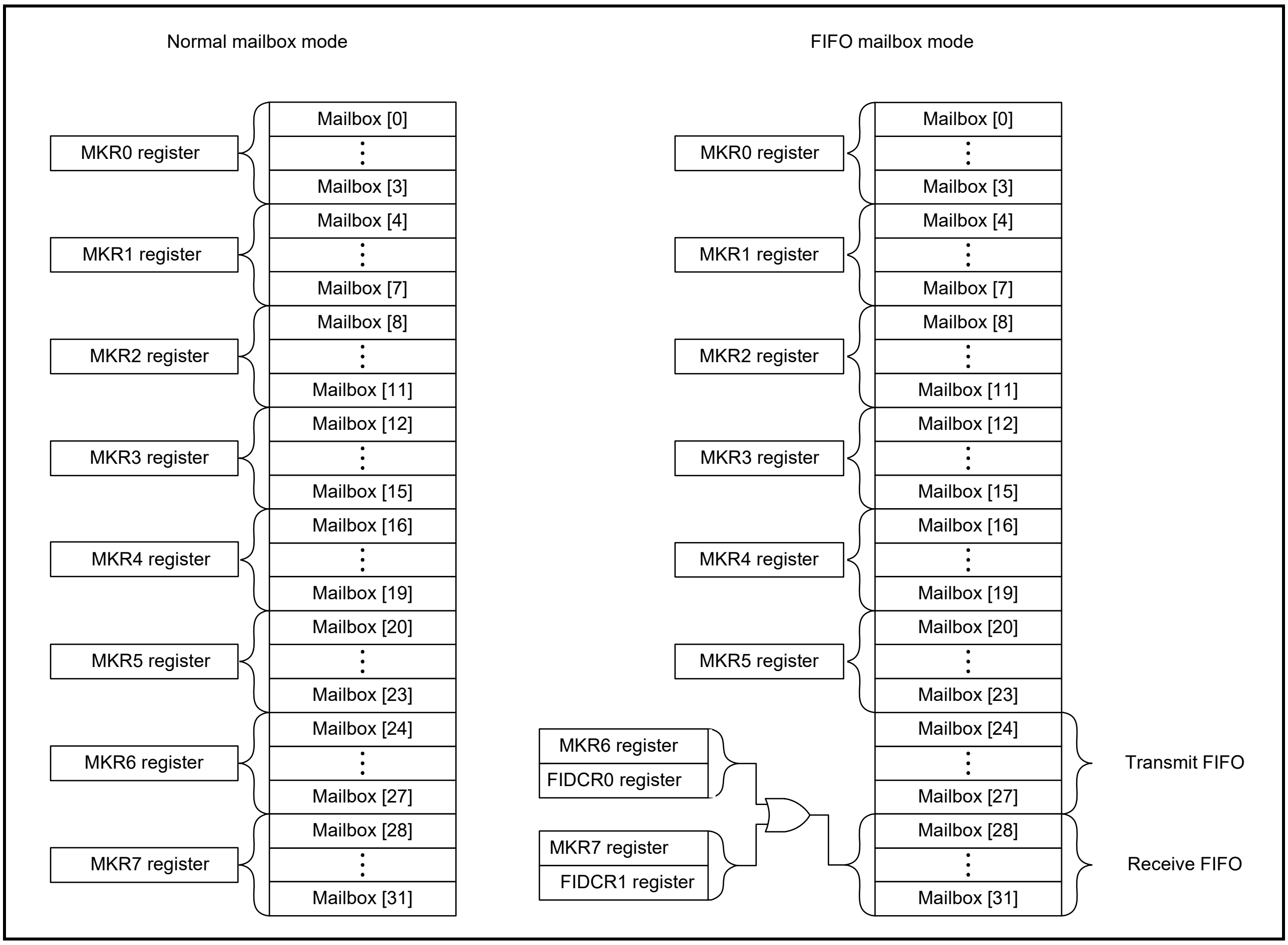
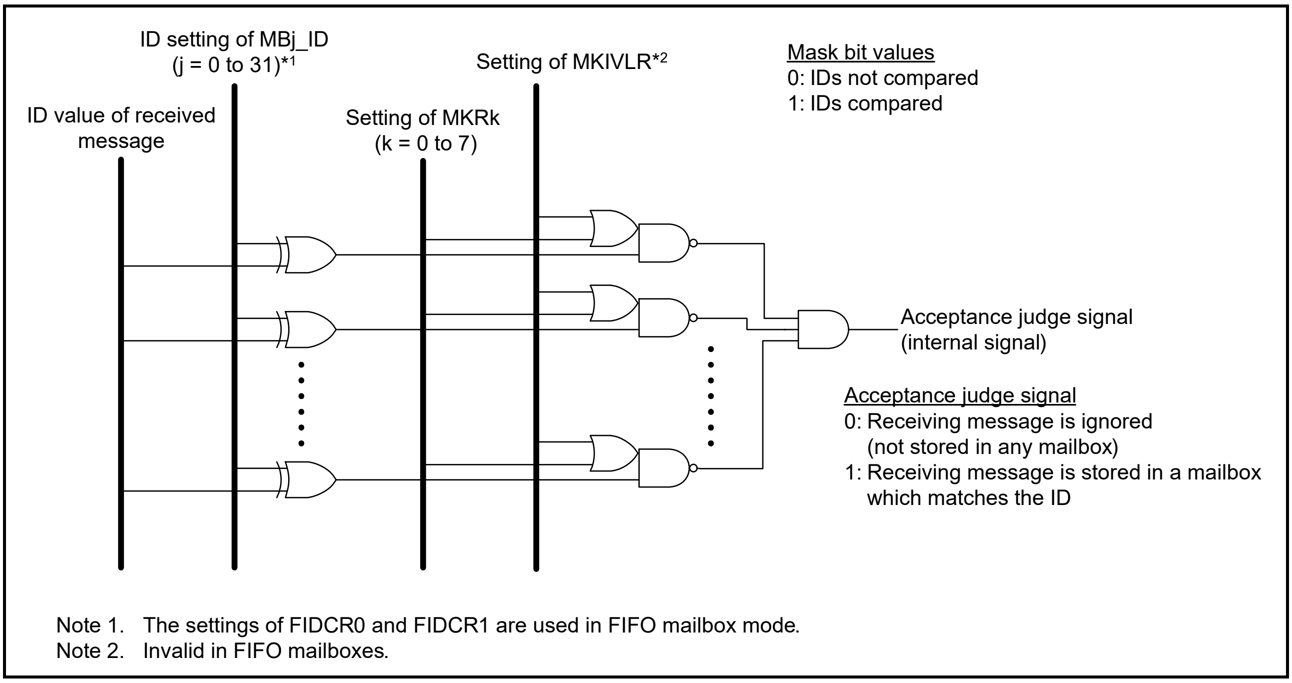


Figure 37.16 Associations between mask registers and mailboxes



Note 1. The settings of FIDCR0 and FIDCR1 are used in FIFO mailbox mode.
Note 2. Invalid in FIFO mailboxes.

Figure 37.17 Acceptance filtering

37.7 Reception and Transmission

Table 37.10 lists the CAN communication mode settings.

Table 37.10 Settings for CAN receive and transmit modes

MCTL_TXj.TRMREQ and MCTL_RXj.TRMREQ	MCTL_TXj.RECREQ and MCTL_RXj.RECREQ	MCTL_TXj.ONESHOT and MCTL_RXj.ONESHOT	Mailbox communication mode
0	0	0	Mailbox disabled or transmission being aborted
0	0	1	Can be configured only when transmission or reception from a mailbox programmed in one-shot mode is aborted
0	1	0	Configured as a receive mailbox for a data or remote frame
0	1	1	Configured as a one-shot receive mailbox for a data or remote frame.
1	0	0	Configured as a transmit mailbox for a data or remote frame.
1	0	1	Configured as a one-shot transmit mailbox for a data or remote frame.
1	1	0	Do not set.
1	1	1	Do not set.

j = 0 to 31

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox, the following restrictions apply:

- Before configuring the mailbox, set MCTL_RXj to 00h.
- A received message is stored in the first mailbox that matches the condition resulting from the receive mode settings and acceptance filtering. The matching mailbox with the smallest number takes priority for storing the received message.
- In CAN operation mode, the CAN module does not receive its own transmitted data even if the ID is a match. In self-test mode, however, the CAN module receives its own transmitted data and returns ACK.

When configuring a mailbox as a transmit mailbox or a one-shot transmit mailbox, the following constraint applies:

- Before configuring a mailbox, ensure that MCTL_TXj is 00h and that there is no pending abort process.

37.7.1 Reception

Figure 37.18 shows an operation example of data frame reception in overwrite mode. The example shows the overwriting of the first message when the CAN module receives two consecutive CAN messages that match the receiving conditions in MCTL_RXj (j = 0 to 31).

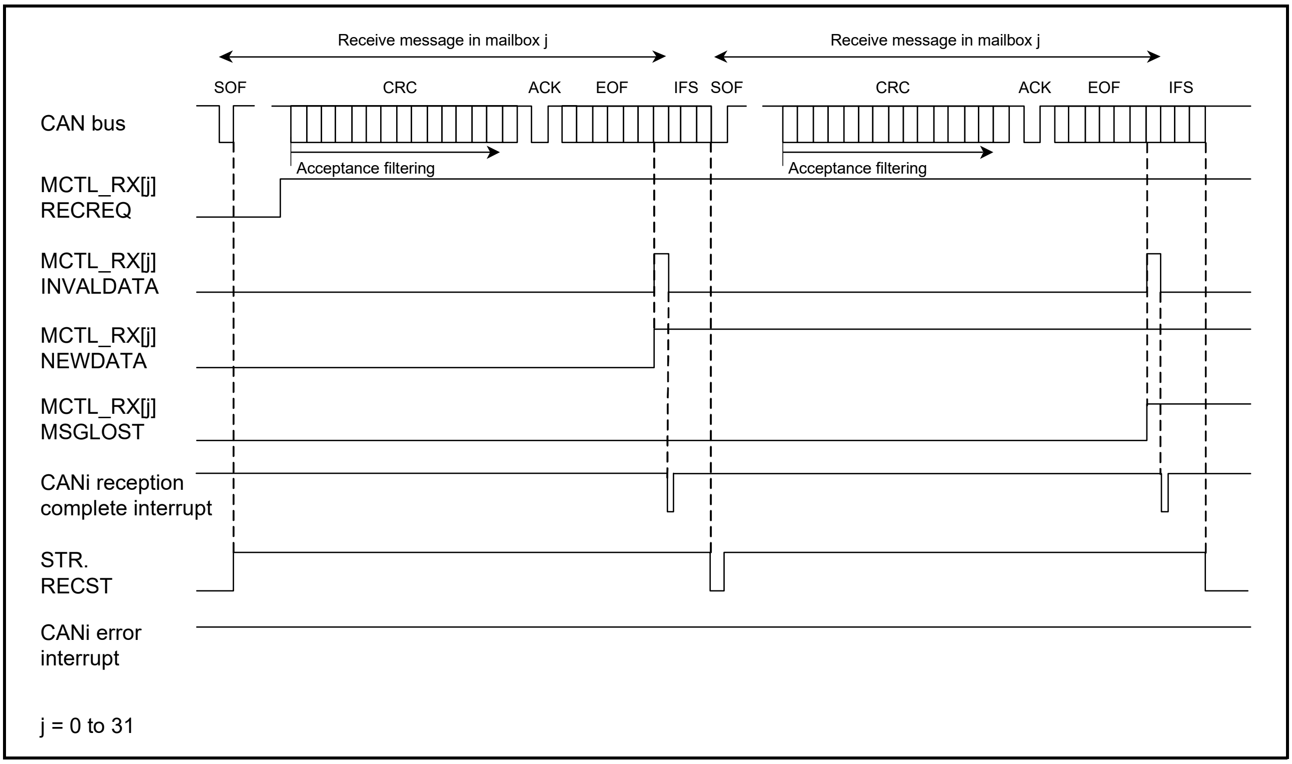


Figure 37.18 Operation example of data frame reception in overwrite mode

1. When an SOF is detected on the CAN bus, the RECST bit in STR is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
2. Acceptance filtering starts at the beginning of the CRC field to select the receive mailbox.
3. After a message is received, the NEWDATA flag in MCTL_RXj for the receive mailbox is set to 1 (new message is being stored or was stored in the mailbox). The INVALDATA flag in MCTL_RXj is set to 1 (message is being updated) at the same time, and then the INVALDATA flag is set to 0 (message valid) again after the complete message is transferred to the mailbox.
4. When the interrupt enable bit in MIER for the receive mailbox is 1 (interrupt enabled), the INVALDATA flag is set to 0, which triggers a CAN0 reception complete interrupt request.
5. After reading the message from the mailbox, the NEWDATA flag must be set to 0 by software.
6. In overwrite mode, if the next CAN message is received while the NEWDATA flag in MCTL_RXj is set to 1, the MSGLOST flag in MCTL_RXj is set to 1 (message was overwritten). The new received message is transferred to the mailbox. The CAN0 reception complete interrupt request occurs the same as in step 4.

Figure 37.19 shows an operation example of data frame reception in overrun mode. The example shows the overrunning of the second message when the CAN module receives two consecutive CAN messages that match the receiving conditions in MCTL_RXj (j = 0 to 31).

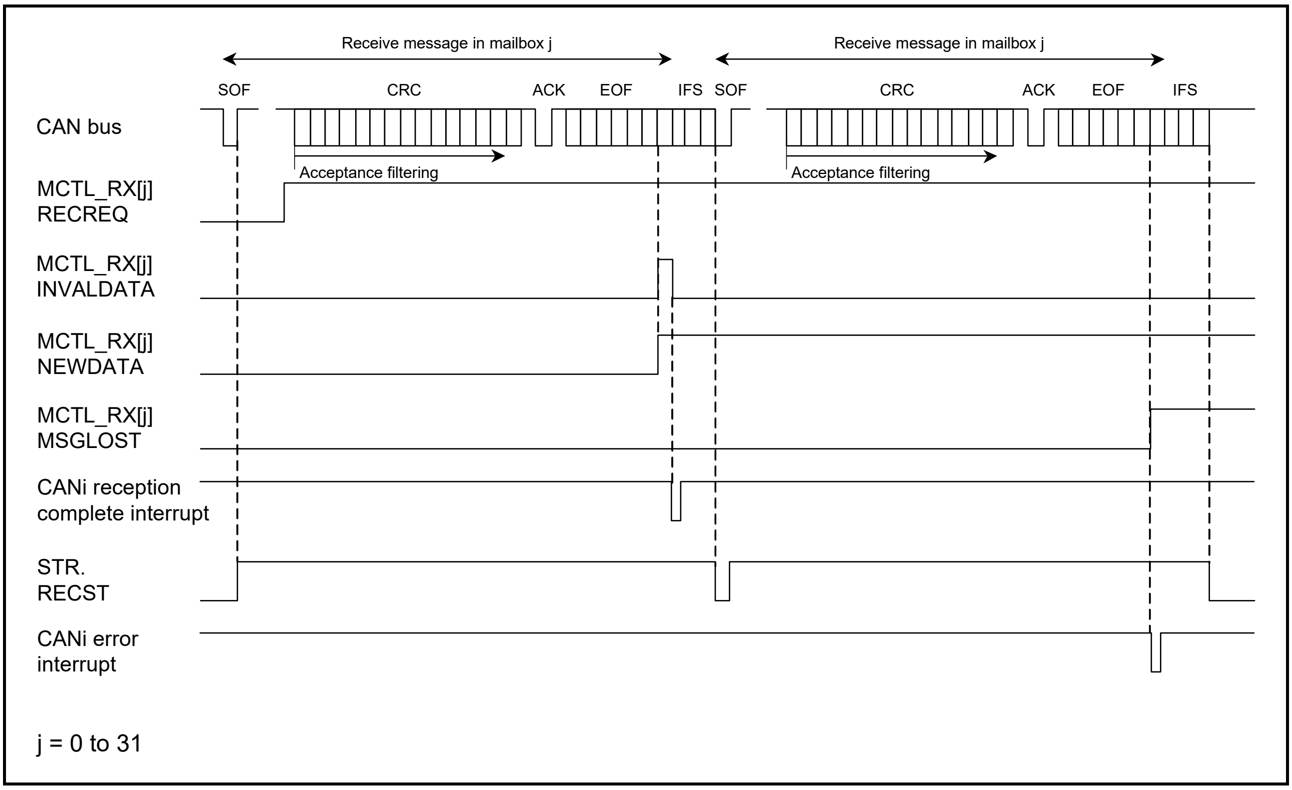


Figure 37.19 Operation example of data frame reception in overrun mode

Steps 1 to 5 are the same as in overwrite mode.

6. In overrun mode, if the next CAN message is received before the NEWDATA flag in MCTL_RXj is set to 0, the MSGLOST flag in MCTL_RXj is set to 1 (message overrun). The new received message is discarded and a CANi error interrupt request occurs if the associated interrupt enable bit in EIER is set to 1 (interrupt enabled).

37.7.2 Transmission

Figure 37.20 shows an operation example of data frame transmission.

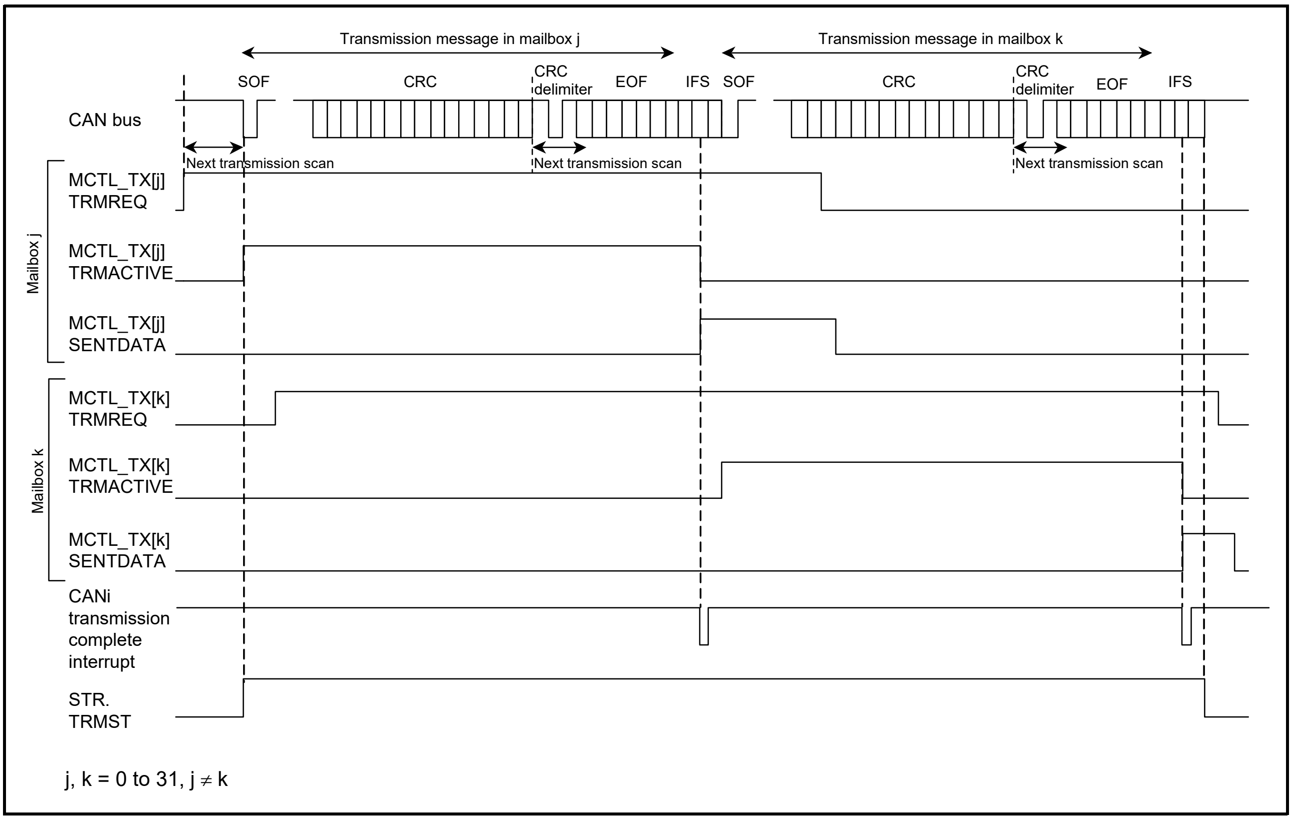


Figure 37.20 Operation example of data frame transmission

1. When a TRMREQ bit in MCTL_TXj (j = 0 to 31) is set to 1 (transmit mailbox) in the bus-idle state, mailbox scanning starts to decide the highest-priority mailbox for transmission. When the transmit mailbox is decided, the TRMACTIVE flag in MCTL_TXj is set to 1 (from acceptance of transmission request to completion of transmission, or error or arbitration-lost), the TRMST bit in STR is set to 1 (transmission in progress), and the CAN module starts transmission.*1
2. If other TRMREQ bits are set, the transmission scanning starts with the CRC delimiter for the next transmission.
3. If transmission is completed without losing arbitration, the SENTDATA flag in MCTL_TXj is set to 1 (transmission complete) and the TRMACTIVE flag is set to 0 (transmission is pending or transmission is not requested). If the interrupt enable bit in MIER is 1 (interrupt enabled), the CANi transmission complete interrupt request is generated.
4. When requesting the next transmission from the same mailbox, set the SENTDATA flag and TRMREQ bit to 0, and then set the TRMREQ bit to 1 after checking that the SENTDATA flag and TRMREQ bit are set to 0.

Note 1. If arbitration is lost after the CAN module starts transmission, the TRMACTIVE flag is set to 0. Transmission scanning is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following arbitration-lost, transmission scanning is performed again to search for the highest-priority transmit mailbox from the start of the error delimiter.

37.8 Interrupts

The CAN module provides the following interrupts for each channel. Table 37.11 lists the CAN interrupts.

- CANi reception complete interrupt for mailboxes 0 to 31 (CANi_RXM)
- CANi transmission complete interrupt for mailboxes 0 to 31 (CANi_TXM)
- CANi receive FIFO interrupt (CANi_RXF)
- CANi transmit FIFO interrupt (CANi_TXF)
- CANi error interrupt (CANi_ERS).

Eight interrupt sources are available for CANi error interrupts. Check EIFR to determine whether these sources were triggered:

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock.

Table 37.11 CAN interrupts

Module	Interrupt name	Interrupt source	Source flag
CANi i = 0, 1	CANi_ERS	Bus lock detected	EIFR.BLIF
		Overload frame transmission detected	EIFR.OLIF
		Overrun detected	EIFR.ORIF
		Bus-off recovery detected	EIFR.BORIF
		Bus-off entry detected	EIFR.BOEIF
		Error-passive detected	EIFR.EPIF
		Error-warning detected	EIFR.EWIF
		Bus error detected	EIFR.BEIF
	CANi_RXF	Receive FIFO message received (MIER_FIFO.MB29 = 0)	RFCR.RFUST[2:0]
		Receive FIFO warning (MIER_FIFO.MB29 = 1)	
	CANi_TXF	Transmit FIFO message transmission completed (MIER_FIFO.MB25 = 0)	TFCR.TFUST[2:0]
		FIFO last message transmission completed (MIER_FIFO.MB25 = 1)	
	CANi_RXM	Mailbox 0 to 31 message received	MCTL_RX0.NEWDATA to MCTL_RX31.NEWDATA
	CANi_TXM	Mailbox 0 to 31 message transmission completed	MCTL_TX0.SENTDATA to MCTL_TX31.SENTDATA

37.9 Usage Notes

37.9.1 Settings for the Module-Stop Function

CAN operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The CAN module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

37.9.2 Settings for the Operating Clock

The settings for the operating clock can be made as follows:

- The following clock constraint must be satisfied for the CAN module when the CCLKS bit is 1:

  fPCLKB ≥ fCANMCLK

- The source of the peripheral module clock must be PLL for the CAN module when the CCLKS bit is 0.

38. Serial Peripheral Interface (SPI)

38.1 Overview

The MCU provides two independent channels for the Serial Peripheral Interface (SPI). The SPI channels are capable of high-speed, full-duplex, synchronous serial communications with multiple processors and peripheral devices. Table 38.1 lists the SPI specifications, Figure 38.1 shows a block diagram, and Table 38.2 lists the I/O pins.

In this section, *n* indicates A or B, and *i* indicates 0 or 1. A lower-case letter *i* in pin and signal names indicates a value from 0 to 3, and a lower-case letter *m* in SPI Command Register m (SPCMDm) indicates a value from 0 to 7.

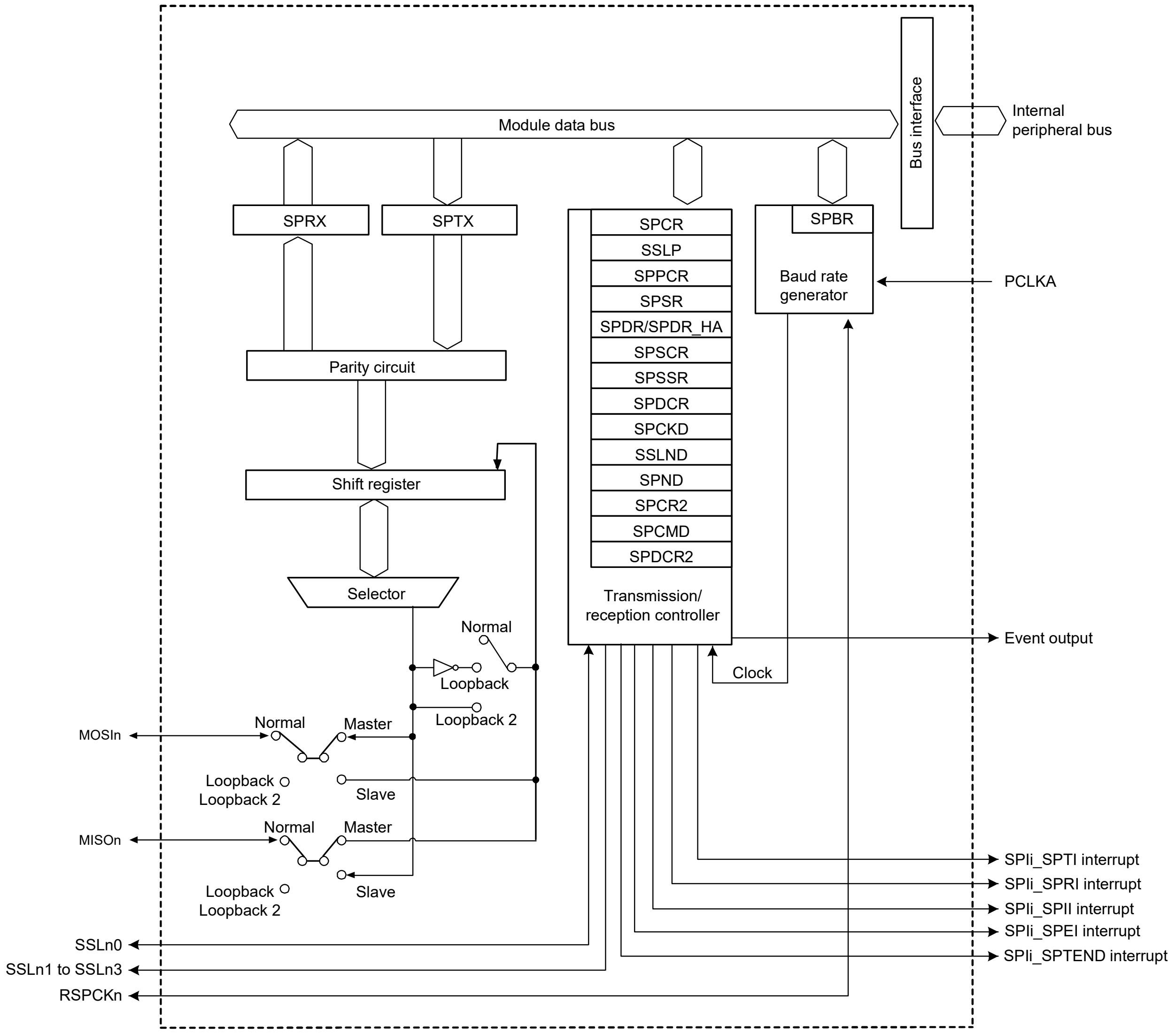
Table 38.1 SPI specifications (1 of 2)

Parameter	Specifications
Number of channels	Two channels
SPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method) • Transmit-only operation available • Communication mode selectable to full-duplex or transmit-only • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB-first or LSB-first selectable • Transfer bit length selectable to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits • 128-bit transmit and receive buffers • Up to four frames transferrable in one round of transmission or reception (each frame consisting of up to 32 bits) • Byte swap operating function
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLKA (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum PCLKA clock divided by 4 can be input as RSPCK (PCLKA divided by 4 is the maximum RSPCK frequency) Width at high level: 2 PCLKA cycles; width at low level: 2 PCLKA cycles
Buffer configuration	• Double buffer configuration for the transmit and receive buffers • 128 bits for the transmit and receive buffers
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection^{*1} • Parity error detection
SSL control function	• Four SSL pins (SSLn0 to SSLn3) for each channel • In single master mode, SSLn0 to SSLn3 pins are output. • In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused • In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• Transfers of up to eight commands each can be executed sequentially in looped execution • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay • Transfers can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function
Interrupt sources	Interrupt sources: • Receive buffer full interrupt • Transmit buffer empty interrupt • SPI error interrupt (mode fault, overrun, parity error) • SPI idle interrupt (SPI idle) • Transmission-complete interrupt

Table 38.1 SPI specifications (2 of 2)

Parameter	Specifications
Event link function (output)	The following events can be output to the Event Link Controller (ELC): • Receive buffer full signal • Transmit buffer empty signal • Mode fault, underrun, overrun, or parity error signal • SPI idle signal • Transmission-complete signal
Other functions	• Switching between CMOS output and open-drain output • SPI initialization function • Loopback mode
Module-stop function	Module-stop state can be set to reduce power consumption

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.



SPCR:	SPI Control Register
SPCR2:	SPI Control Register 2
SSLP:	SPI Slave Select Polarity Register
SPPCR:	SPI Pin Control Register
SPSR:	SPI Status Register
SPDR/SPDR_HA:	SPI Data Register
SPSCR:	SPI Sequence Control Register
SPSSR:	SPI Sequence Status Register
SPDCR:	SPI Data Control Register
SPCKD:	SPI Clock Delay Register
SSLND:	SPI Slave Select Negation Delay Register
SPND:	SPI Next-Access Delay Register
SPCMD:	SPI Command Registers 0 to 7 (eight registers)
SPBR:	SPI Bit Rate Register
SPTX:	SPI Transmit Buffer
SPRX:	SPI Receive Buffer
SPDCR2:	SPI Data Control Register 2

Figure 38.1 SPI block diagram

Table 38.2 lists the I/O pins used in the SPI. The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master and as an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISOn pins are automatically set as inputs or outputs based on the master or slave setting and the level input on the SSLn0 pin. For details, see section 38.3.2, Controlling the SPI Pins.

Table 38.2 SPI I/O pins

Channel	Pin name	I/O	Function
SPI0	RSPCKA	I/O	Clock input/output
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output
	SSLA0	I/O	Slave selection input/output
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output
SPI1	RSPCKB	I/O	Clock input/output
	MOSIB	I/O	Master transmit data input/output
	MISOB	I/O	Slave transmit data input/output
	SSLB0	I/O	Slave selection input/output
	SSLB1	Output	Slave selection output
	SSLB2	Output	Slave selection output
	SSLB3	Output	Slave selection output

38.2 Register Descriptions

38.2.1 SPI Control Register (SPCR)

Address(es): SPI0.SPCR 4007 2000h, SPI1.SPCR 4007 2100h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODF EN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SPMS	SPI Mode Select	0: Select SPI operation (4-wire method) 1: Select clock synchronous operation (3-wire method).	R/W
b1	TXMD	Communications Operating Mode Select	0: Select full-duplex synchronous serial communications 1: Select serial communications with transmit-only.	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disable detection of mode fault errors 1: Enable detection of mode fault errors.	R/W
b3	MSTR	SPI Master/Slave Mode Select	0: Select slave mode 1: Select master mode.	R/W
b4	SPEIE	SPI Error Interrupt Enable	0: Disable SPI error interrupt requests 1: Enable SPI error interrupt requests.	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disable transmit buffer empty interrupt requests 1: Enable transmit buffer empty interrupt requests.	R/W
b6	SPE	SPI Function Enable	0: Disable SPI function 1: Enable SPI function.	R/W
b7	SPRIE	SPI Receive Buffer Full Interrupt Enable	0: Disable SPI receive buffer full interrupt requests 1: Enable SPI receive buffer full interrupt requests.	R/W

If the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit is changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

SPMS bit (SPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).

The SSLn0 to SSLn3 pins are not used in clock synchronous operation. The RSPCKn, MOSIn, and MISOn pins handle communications. For clock synchronous operation in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. For clock synchronous operation in slave mode (SPCR.MSTR = 0), always set the CPHA bit to 1. Do not perform operations if the CPHA bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0).

TXMD bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit-only operations. When this bit is set to 1, the SPI only performs transmit operations and not receive operations (see section 38.3.6, Data Transfer Modes), and receive buffer full interrupt requests cannot be used.

MODFEN bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault errors (see section 38.3.8, Error Detection). In addition, the SPI determines the I/O direction of the SSLn0 to SSLn3 pins based on combination of the MODFEN and MSTR bits (see section 38.3.2, Controlling the SPI Pins).

MSTR bit (SPI Master/Slave Mode Select)

The MSTR bit selects master or slave mode for the SPI. Based on the MSTR bit settings, the SPI determines the direction of the RSPCKn, MOSIn, MISOn, and SSLn0 to SSLn3 pins.

SPEIE bit (SPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of SPI error interrupt requests when one of the following occurs:

- The SPI detects a mode fault error or underrun error and sets the SPSR.MODF flag to 1
- The SPI detects an overrun error and sets the SPSR.OVRF flag to 1
- The SPI detects a parity error and sets the SPSR.PERF flag to 1.

For details, see section 38.3.8, Error Detection.

SPTIE bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the SPI detects that the transmit buffer is empty. To generate a transmit buffer empty interrupt request when transmission starts, set the SPE and SPTIE bits to 1 at the same time or set the SPE bit to 1 after setting the SPTIE bit to 1.

When the SPTIE bit is 1, transmit buffer interrupts are generated even when the SPI function is disabled (when the SPE bit is changed to 0).

SPE bit (SPI Function Enable)

The SPE bit enables or disables the SPI function. The SPE bit cannot be set to 1 when the SPSR.MODF flag is 1. For details, see section 38.3.8, Error Detection.

Setting the SPE bit to 0 disables the SPI function and initializes a part of the module function. For details, see section 38.3.9, Initializing the SPI. In addition, a transmit buffer empty interrupt request is generated when the SPE bit is changed from 0 to 1 or from 1 to 0.

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

The SPRIE bit enables or disables the generation of an SPI receive buffer full interrupt request when the SPI detects a receive buffer full write after completion of a serial transfer.

38.2.2 SPI Slave Select Polarity Register (SSLP)

Address(es): SPI0.SSLP 4007 2001h, SPI1.SSLP 4007 2101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: Set SSL0 signal to active low 1: Set SSL0 signal to active high.	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: Set SSL1 signal to active low 1: Set SSL1 signal to active high.	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: Set SSL2 signal to active low 1: Set SSL2 signal to active high.	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: Set SSL3 signal to active low 1: Set SSL3 signal to active high.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SSLP are changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

38.2.3 SPI Pin Control Register (SPPCR)

Address(es): SPI0.SPPCR 4007 2002h, SPI1.SPPCR 4007 2102h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SPLP	SPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission).	R/W
b1	SPLP2	SPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission).	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: Set level output on MOSIn pin during MOSI idling to correspond to low 1: Set level output on MOSIn pin during MOSI idling to correspond to high.	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: Set MOSI output value to equal final data from previous transfer 1: Set MOSI output value to equal value set in the MOIFV bit.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SPPCR are changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

SPLP bit (SPI Loopback)

The SPLP bit selects the mode of the SPI pins. When this bit is set to 1, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0. The SPI then connects the input path and output path for the shift register (loopback mode).

SPLP2 bit (SPI Loopback 2)

The SPLP2 bit selects the mode of the SPI pins. When this bit is set to 1, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the

SPCR.MSTR bit is 0. The SPI then connects the input path and output path for the shift register (loopback mode).

MOIFV bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIn pin output value during the SSL negation period, including the SSL retention period during a burst transfer.

MOIFE bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIn output value when the SPI is in master mode and in an SSL negation period, including the SSL retention period during a burst transfer. When the MOIFE bit is 0, the SPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIn pin. When the MOIFE bit is 1, the SPI outputs the fixed value set in the MOIFV bit to the MOSIn pin.

38.2.4 SPI Status Register (SPSR)

Address(es): SPI0.SPSR 4007 2003h, SPI1.SPSR 4007 2103h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	—	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurred 1: Overrun error occurred.	R/(W)*1
b1	IDLNF	SPI Idle Flag	0: SPI is in the idle state 1: SPI is in the transfer state.	R
b2	MODF	Mode Fault Error Flag	0: No mode fault or underrun error occurred 1: Mode fault error or underrun error occurred.	R/(W)*1
b3	PERF	Parity Error Flag	0: No parity error occurred 1: Parity error occurred.	R/(W)*1
b4	UDRF	Underrun Error Flag	0: Mode fault error occurred (MODF = 1) 1: Underrun error occurred (MODF = 1). This bit is invalid when MODF flag is 0.	R/W*1,*2
b5	SPTEF	SPI Transmit Buffer Empty Flag	0: Data is in the transmit buffer 1: No data is in the transmit buffer.	R/(W)*3
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	SPRF	SPI Receive Buffer Full Flag	0: No valid data is in SPDR/SPDR_HA 1: Valid data is in SPDR/SPDR_HA.	R/(W)*3

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. The UDRF flag clears at the same time that the software clears the MODF flag.
Note 3. The write value should be 1.

OVRF flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (SPCR.MSTR bit = 1) and when the RSPCK clock auto-stop function is enabled (SPCR2.SCKASE bit = 1), overrun errors do not occur. This flag does not set to 1. For details, see section 38.3.8.1, Overrun errors.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When 0 is written to the OVRF flag after the OVRF flag is confirmed to be 1 by a read of SPSR.

IDLNF flag (SPI Idle Flag)

The IDLNF flag indicates the transfer status of the SPI.

[Setting conditions]

Master mode

- When conditions 1. and 2. in the master mode [Clearing conditions] are not satisfied.

Slave mode

- When the SPCR.SPE bit is 1, enabling the SPI function.

[Clearing conditions]

Master mode

- When condition 1. OR conditions 2., 3., and 4. are satisfied.

1. The SPCR.SPE bit is 0, indicating the SPI is initialized.
2. The transmit buffer (SPTX) is empty, indicating that data for the next transfer is not set.
3. The SPSSR.SPCP[2:0] bits are 000b, indicating the beginning of sequence control.
4. The SPI internal sequencer enters the idle state, indicating that operations up to the next-access delay are complete.

Slave mode

- When condition 1. is satisfied.

MODF flag (Mode Fault Error Flag)

The MODF flag indicates the occurrence of a mode fault error or an underrun error. Use the UDRF flag to identify which error occurred.

[Setting conditions]

Multi-master mode

- When the input level of the SSLni pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled). The SPI detects a mode fault error.

Slave mode

- When condition 1. OR 2. is satisfied.

1. The SSLni pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled). The SPI detects a mode fault error.
2. The serial transfer begins while the SPCR.MSTR bit is 0 (slave mode), the SPCR.SPE bit is 1, and the transmission data is not prepared. The SPI detects an underrun error.

The active level of the SSLni signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When 0 is written to the MODF flag after the MODF flag is confirmed to be 1 by a read of SPSR.

PERF flag (Parity Error Flag)

The PERF flag indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The SPI detects a parity error.

[Clearing condition]

- When 0 is written to the PERF flag after the PERF flag is confirmed to be 1 by a read of SPSR.

UDRF flag (Underrun Error Flag)

The UDRF flag indicates the occurrence of an underrun error.

[Setting condition]

- When the serial transfer begins while the SPCR.MSTR bit is 0 (slave mode), the SPCR.SPE bit is 1, and the transmission data is not prepared. The SPI detects an underrun error.

[Clearing condition]

- When 0 is written to the UDRF flag after the UDRF flag is confirmed to be 1 by a read of SPSR.

SPTEF flag (SPI Transmit Buffer Empty Flag)

The SPTEF flag indicates the status of the transmit buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting conditions]

- When condition 1. OR 2. is satisfied.

1. The SPCR.SPE bit is 0 (the SPI is initialized).
2. Transmit data was transferred from the transmit buffer to the shift register.

[Clearing condition]

- When data written to SPDR/SPDR_HA equals the number of frames set in the SPFC[1:0] bits in the SPI Data Control Register (SPDCR).

Data can only be written to SPDR/SPDR_HA when the SPTEF flag is 1. If data is written to the transmit buffer of SPDR/SPDR_HA when the SPTEF flag is 0, the data in the transmit buffer is not updated.

SPRF flag (SPI Receive Buffer Full Flag)

The SPRF flag indicates the status of the receive buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting condition]

- When a serial transfer ends while the communication operating mode select bit (TXMD) in the SPI Control Register (SPCR) is 0, the SPRF flag is 0, and the SPI transfers the receive data from the shift register to SPDR/SPDR_HA. However, when the OVRF flag is 1, the SPRF flag does not change from 0 into 1.

[Clearing condition]

- When received data is read from the SPDR/SPDR_HA.

38.2.5 SPI Data Register (SPDR/SPDR_HA)

Address(es): SPI0.SPDR 4007 2004h, SPI1.SPDR 4007 2104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

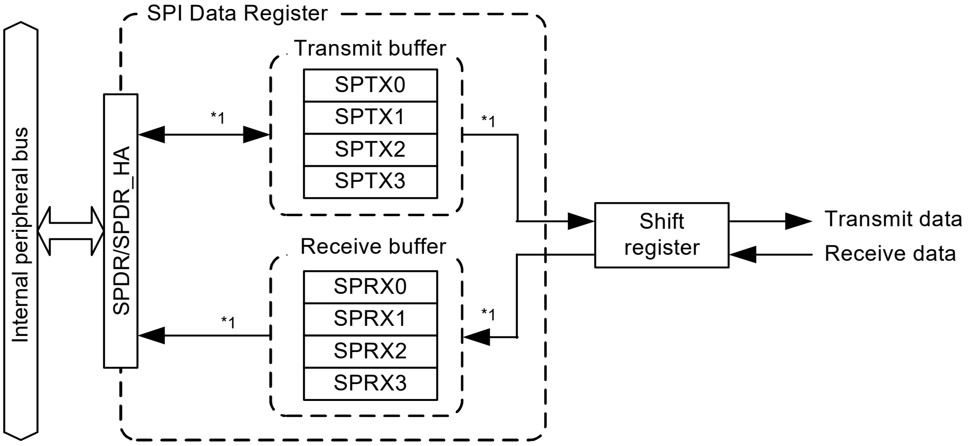
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Address(es): SPI0.SPDR_HA 4007 2004h, SPI1.SPDR_HA 4007 2104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDR/SPDR_HA is the interface with the buffers that hold data for transmission and reception by the SPI. When accessing this register in words (the SPLW bit is 1), access SPDR. When accessing it in halfwords (the SPLW bit is 0), access SPDR_HA.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR/SPDR_HA. Figure 38.2 shows the configuration.



Note 1. The destination buffer and stage for access is automatically switched by the hardware.

Figure 38.2 Configuration of SPDR/SPDR_HA

The transmit and receive buffers each have four stages. The number of stages to be used is selectable in the number of frames specification bits in the SPI Data Control Register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all mapped to the single address of SPDR/SPDR_HA.

Data written to SPDR/SPDR_HA is written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Additionally, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the associated bits in SPRXn (n = 0 to 3). For example, if the data length is 9 bits, the received data is stored in the SPRXn[8:0] bits, and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus interface

SPDR/SPDR_HA is an interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for SPDR/SPDR_HA. Additionally, the unit of access for SPDR/SPDR_HA is selected by the SPI word access/halfword access specification bit in the SPI Data Control Register (SPDCR.SPLW). Other case, make an access to SPDR with the access size specified by the SPI byte access bit in the SPI Data Control Register (SPDCR.SPBYT).

Flush transmission data at the LSB end of the register, and store received data at the LSB end.

This section describes the operations involved in writing to and reading from SPDR/SPDR_HA.

(a) Writing

Data written to SPDR/SPDR_HA is written to a transmit buffer (SPTXn). This is not affected by the value of the SPDCR.SPRDTD bit, unlike when reading from SPDR/SPDR_HA. The transmit buffer includes a transmit buffer write pointer that is automatically updated to reference the next stage each time data is written to SPDR/SPDR_HA.

Figure 38.3 shows the configuration of the bus interface with the transmit buffer when writing to SPDR.

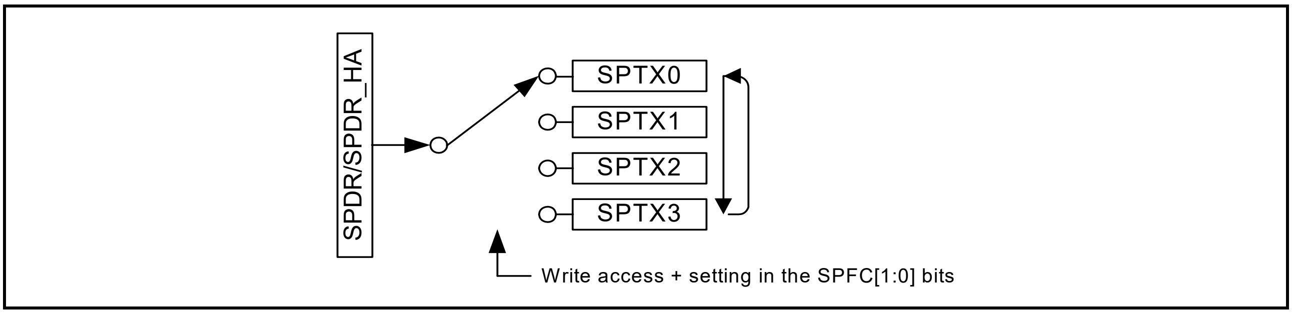


Figure 38.3 Configuration of SPDR/SPDR_HA for write access

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the SPI Data Control Register (SPDCR.SPFC[1:0]). The relationship of the SPFC[1:0] setting and the sequence of pointer switching among SPTX0 to SPTX3 is as follows:

- When SPFC[1:0] = 00b: SPTX0 → SPTX0 → SPTX0 → …
- When SPFC[1:0] = 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
- When SPFC[1:0] = 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
- When SPFC[1:0] = 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the SPI function enable bit in the SPI Control Register (SPCR.SPE) while the value of the bit is 0, SPTX0 is the destination the next time writing proceeds.

When writing to the transmit buffer (SPTXn) after generating the transmit buffer empty interrupt (when SPSR.SPTEF is 1), write the number of frames set in SPFC[1:0] in the SPI Data Control Register (SPDCR). Even when the specified number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before the next transmit buffer empty interrupt is generated (when SPTEF is 0).

(b) Reading

SPDR/SPDR_HA can be accessed to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting in the SPI receive/transmit data select bit in the SPI Data Control Register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer. The sequence of reading the SPDR/SPDR_HA register is controlled by the independent receive buffer and transmit buffer read pointers.

Figure 38.4 shows the configuration of the bus interface with the receive and transmit buffers for a read from SPDR/SPDR_HA.

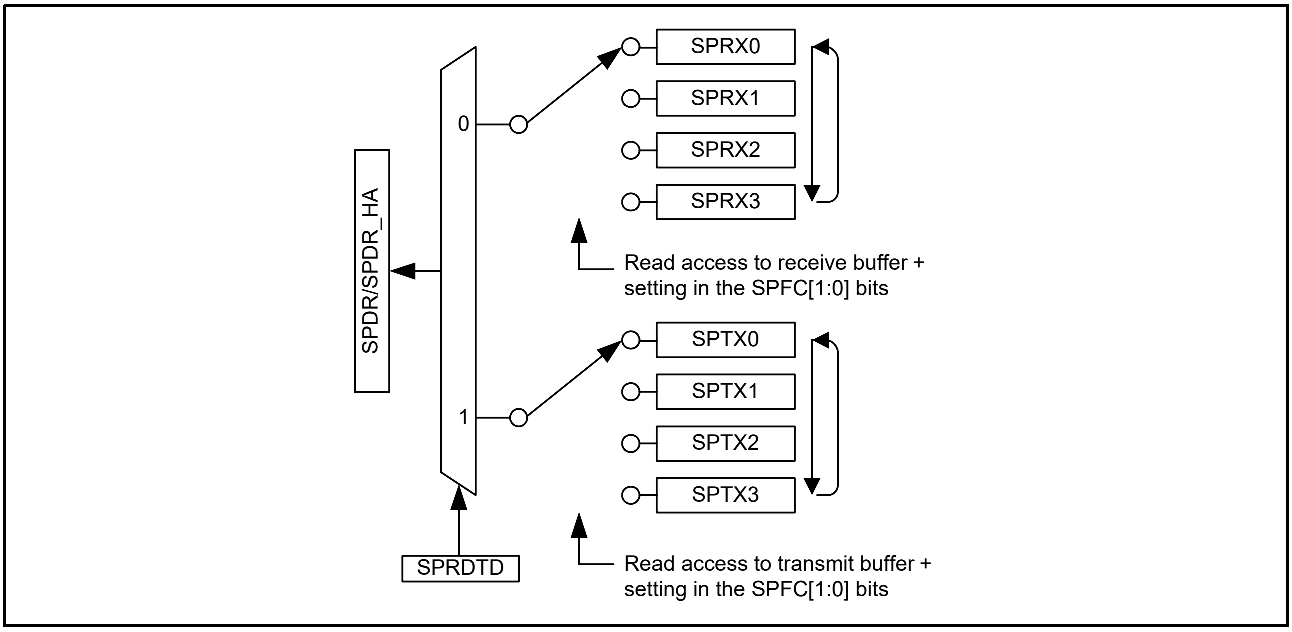


Figure 38.4 Configuration of SPDR/SPDR_HA for read access

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically. The switching sequence for the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the SPI function enable bit in the SPI Control Register (SPCR.SPE) while the value of the bit is 1, SPRX0 is referenced by the buffer read pointer the next time reading proceeds.

The transmit buffer read pointer is updated when writing to SPDR/SPDR_HA, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR/SPDR_HA is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0s in the interval after completion of writing the number of frames of data specified in the SPDCR.SPFC[1:0] bits and before generation of the next buffer empty interrupt (when SPTEF is 0).

38.2.6 SPI Sequence Control Register (SPSCR)

Address(es): SPI0.SPSCR 4007 2008h, SPI1.SPSCR 4007 2108h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SPSLN[2:0]	SPI Sequence Length Specification	b2 b0 Sequence Length Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 0→0→… 0 0 1: 2 0→1→0→… 0 1 0: 3 0→1→2→0→… 0 1 1: 4 0→1→2→3→0→… 1 0 0: 5 0→1→2→3→4→0→… 1 0 1: 6 0→1→2→3→4→5→0→… 1 1 0: 7 0→1→2→3→4→5→6→0→… 1 1 1: 8 0→1→2→3→4→5→6→7→0→… The sequence length that is set in these bits determines the order in which the SPCMD0 to SPCMD07 registers are referenced. The setting defines the relationship between the sequence length and the SPCMD0 to SPCMD7 registers referenced by the SPI. In slave mode, the SPI references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPSCR specifies the sequence length when the SPI operates in master mode. Before changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, always check that the SPSR.IDLNF flag is 0.

SPSLN[2:0] bits (SPI Sequence Length Specification)

The SPSLN[2:0] bits specify the sequence length when the SPI in master mode performs sequential operations. The SPI in master mode changes the SPCMD0 to SPCMD7 registers to be referenced, and the order in which they are referenced is based on this sequence length setting. In slave mode, SPCMD0 is referenced.

38.2.7 SPI Sequence Status Register (SPSSR)

Address(es): SPI0.SPSSR 4007 2009h, SPI1.SPSSR 4007 2109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SPCP[2:0]	SPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7.	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	SPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7.	R
b7	—	Reserved	This bit is read as 0.	R

SPSSR indicates the sequence control status when the SPI operates in master mode. Any writes to SPSSR are ignored.

SPCP[2:0] bits (SPI Command Pointer)

The SPCP[2:0] bits indicate the SPCMDm register that is referenced to by the pointer during sequence control by the SPI. For the SPI sequence control, see section 38.3.10.1, Master mode operation.

SPECM[2:0] bits (SPI Error Command)

The SPECM[2:0] bits indicate the SPCMDm register that is specified in the SPCP[2:0] bits when an error is detected during sequence control by the SPI. The SPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.

For the SPI error detection function, see section 38.3.8, Error Detection. For the SPI sequence control, see section 38.3.10.1, Master mode operation.

38.2.8 SPI Bit Rate Register (SPBR)

Address(es): SPI0.SPBR 4007 200Ah, SPI1.SPBR 4007 210Ah

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

When the SPI is used in slave mode, the bit rate depends on the bit rate of the input clock regardless of the settings in SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting). Use bit rates that satisfy the electrical characteristics.

The bit rate is determined by combinations of the SPBR and SPCMDm.BRDV[1:0] settings. The equation for calculating the bit rate is as follows:

$$\text{Bit rate} = \frac{f\,(\text{PCLKA})}{2 \times (n + 1) \times 2^N}$$

In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] setting (0, 1, 2, 3).

Table 38.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.

Table 38.3 Relationship among SPBR settings, BRDV[1:0] settings, and bit rates

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate							
			PCLKA = 32 MHz	PCLKA = 36 MHz	PCLKA = 40 MHz	PCLKA = 50 MHz	PCLKA = 60 MHz	PCLKA = 80 MHz	PCLKA = 100 MHz	PCLKA = 120 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps	Not supported		
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps	13.3 Mbps	16.7 Mbps	20.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	7.50Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	6.00 Mbps	8.00 Mbps	10.0 Mbps	12.0 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	5.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps	3.33 Mbps	4.17 Mbps	5.00 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps	1.25 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	625 kbps	833 kbps	1.04 Mbps	1.25 Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	14.6 kbps	19.5 kbps	24.4 kbps	29.3 kbps

38.2.9 SPI Data Control Register (SPDCR)

Address(es): SPI0.SPDCR 4007 200Bh, SPI1.SPDCR 4007 210Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPBYT	SPLW	SPRDTD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	SPFC[1:0]	Number of Frames Specification	b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPRDTD	SPI Receive/Transmit Data Select	0: Read SPDR/SPDR_HA values from receive buffer 1: Read SPDR/SPDR_HA values from transmit buffer, but only if the transmit buffer is empty.	R/W
b5	SPLW	SPI Word Access/Halfword Access Specification	0: Set SPDR_HA to valid for halfword access 1: Set SPDR to valid for word access.	R/W
b6	SPBYT	SPI Byte Access Specification	0: SPDR is accessed in halfword or word (SPLW is valid) 1: SPDR is accessed in byte (SPLW is invalid).	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Up to four frames can be transmitted or received in one round of transmission or reception. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.

When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, always check that the SPSR.IDLNF flag is 0.

SPFC[1:0] bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR/SPDR_HA per transfer activation. Up to four frames can be transmitted or received in one round of transmission or reception.

When the number of transmission data frames specified in the SPFC[1:0] bits is written to the SPDR/SPDR_HA register, SPI clears the SPSR.SPTEF flag to 0 and begins transmitting. After that, when the number of transmission data frames specified in the SPFC[1:0] bits is transmitted to the shift register, the SPI generates the transmit buffer empty interrupt (SPSR.SPTEF sets to 1).

When the number of data frames specified in the SPFC[1:0] bits is received, the SPI generates the receive buffer full interrupt (SPSR.SPRF sets to 1).

Table 38.4 Settable combinations of the SPSLN[2:0] and SPFC[1:0] bits (1 of 2)

Setting	SPSLN[2:0]	SPFC[1:0]	Number of frames in a single sequence	Number of frames at which transmission or receive buffer is filled
1-1	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1

Table 38.4 Settable combinations of the SPSLN[2:0] and SPFC[1:0] bits (2 of 2)

Setting	SPSLN[2:0]	SPFC[1:0]	Number of frames in a single sequence	Number of frames at which transmission or receive buffer is filled
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD bit (SPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR/SPDR_HA reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the value written to SPDR/SPDR_HA register immediately beforehand is read. When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (when SPSR.SPTEF is 1).

For details, see section 38.2.5, SPI Data Register (SPDR/SPDR_HA).

SPLW bit (SPI Word Access/Halfword Access Specification)

The SPLW bit specifies the access width for SPDR. Access to SPDR_HA in halfwords is valid when the SPLW bit is 0 and access to SPDR in words is valid when the SPLW bit is 1. Also, when this bit is 0, set the SPCMDm.SPB[3:0] bits (SPI data length setting) from 8 to 16 bits. Do not perform any operations when 20, 24, or 32 bits is specified.

SPBYT bit (SPI Byte Access Specification)

This bit is used to set the data width of access to the SPI Data Register (SPDR). When SPBYT = 0, use word or half word access to SPDR. When SPBYT = 1 (in that case, SPLW is invalid), use byte access to SPDR.

When SPBYT = 1, set the SPI data length bits (SPB[3:0]) in the SPI Command Register n (SPCMDn) to 0 bits. If SPB[3:0] are set to 9 to 16, 20, 24, or 32 bit, subsequent operation is not guaranteed.

38.2.10 SPI Clock Delay Register (SPCKD)

Address(es): SPI0.SPCKD 4007 200Ch, SPI1.SPCKD 4007 210Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPCKD specifies the RSPCK delay, the period from the beginning of SSLni signal assertion to RSPCK oscillation, when the SPCMDm.SCKDEN bit is 1. If the contents of SPCKD are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SCKDL[2:0] bits (RSPCK Delay Setting)

The SCKDL[2:0] bits specify an RSPCK delay value when the SPCMDm.SCKDEN bit is 1. When using the SPI in slave mode, set the SCKDL[2:0] bits to 000b.

38.2.11 SPI Slave Select Negation Delay Register (SSLND)

Address(es): SPI0.SSLND 4007 200Dh, SPI1.SSLND 4007 210Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SSLND specifies the SSL negation delay, the period from the transmission of a final RSPCK edge to the negation of the SSLni signal during a serial transfer by the SPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SLNDL[2:0] bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits specify an SSL negation delay value when the SPI is in master mode. When using the SPI in slave mode, set the SLNDL[2:0] bits to 000b.

38.2.12 SPI Next-Access Delay Register (SPND)

Address(es): SPI0.SPND 4007 200Eh, SPI1.SPND 4007 210Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SPNDL[2:0]	SPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 2 PCLKA 0 0 1: 2 RSPCK + 2 PCLKA 0 1 0: 3 RSPCK + 2 PCLKA 0 1 1: 4 RSPCK + 2 PCLKA 1 0 0: 5 RSPCK + 2 PCLKA 1 0 1: 6 RSPCK + 2 PCLKA 1 1 0: 7 RSPCK + 2 PCLKA 1 1 1: 8 RSPCK + 2 PCLKA.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPND specifies the next-access delay, the non-active period of the SSLni signal after termination of a serial transfer, when the SPCMDm.SPNDEN bit is 1. If the contents of SPND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SPNDL[2:0] bits (SPI Next-Access Delay Setting)

The SPNDL[2:0] bits specify a next-access delay when the SPCMDm.SPNDEN bit is 1. When using the SPI in slave mode, set the SPNDL[2:0] bits to 000b.

38.2.13 SPI Control Register 2 (SPCR2)

Address(es): SPI0.SPCR2 4007 200Fh, SPI1.SPCR2 4007 210Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SPPE	Parity Enable	0: Do not add parity bit to transmit data and do not check parity bit of receive data 1: When SPCR.TXMD = 0: Add parity bit to transmit data and check parity bit of receive data When SPCR.TXMD = 1: Add parity bit to transmit data but do not check parity bit of receive data.	R/W
b1	SPOE	Parity Mode	0: Select even parity for transmission and reception 1: Select odd parity for transmission and reception.	R/W
b2	SPIIE	SPI Idle Interrupt Enable	0: Disable idle interrupt requests 1: Enable idle interrupt requests.	R/W
b3	PTE	Parity Self-Testing	0: Disable self-diagnosis function of the parity circuit 1: Enable self-diagnosis function of the parity circuit.	R/W
b4	SCKASE	RSPCK Auto-Stop Function Enable	0: Disable RSPCK auto-stop function 1: Enable RSPCK auto-stop function.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the SPPE, SPOE, or SCKASE bit in SPCR2 is changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

SPPE bit (Parity Enable)

The SPPE bit enables or disables the parity function.

When the SPCR.TXMD bit is 0 and this bit is 1, the parity bit is added to transmit data and parity checking is performed for receive data.

When the SPCR.TXMD bit is 1 and this bit is 1, the parity bit is added to transmit data but parity checking is not performed for receive data.

SPOE bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit or receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit or receive character plus the parity bit is odd.

The SPOE bit is only valid when the SPPE bit is 1.

SPIIE bit (SPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of SPI idle interrupt requests when an idle state is detected in the SPI and the SPSR.IDLNF flag clears to 0.

PTE bit (Parity Self-Testing)

The PTE bit enables self-diagnosis of the parity circuit to check whether the parity function is operating correctly.

SCKASE bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs when data is received in master mode. For details, see section 38.3.8.1, Overrun errors.

38.2.14 SPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)

Address(es): SPI0.SPCMD0 4007 2010h, SPI0.SPCMD1 4007 2012h, SPI0.SPCMD2 4007 2014h, SPI0.SPCMD3 4007 2016h, SPI0.SPCMD4 4007 2018h, SPI0.SPCMD5 4007 201Ah, SPI0.SPCMD6 4007 201Ch, SPI0.SPCMD7 4007 201Eh, SPI1.SPCMD0 4007 2110h, SPI1.SPCMD1 4007 2112h, SPI1.SPCMD2 4007 2114h, SPI1.SPCMD3 4007 2116h, SPI1.SPCMD4 4007 2118h, SPI1.SPCMD5 4007 211Ah, SPI1.SPCMD6 4007 211Ch, SPI1.SPCMD7 4007 211Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	Bit name	Description	R/W
b0	CPHA	RSPCK Phase Setting	0: Select data sampling on leading edge, data change on trailing edge 1: Select data change on leading edge, data sampling on trailing edge.	R/W
b1	CPOL	RSPCK Polarity Setting	0: Set RSPCK low during idle 1: Set RSPCK high during idle.	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: Base bit rate 0 1: Base bit rate divided by 2 1 0: Base bit rate divided by 4 1 1: Base bit rate divided by 8.	R/W
b6 to b4	SSLA[2:0]	SSL Signal Assertion Setting	b6 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 1 x x: Setting prohibited. x: Don’t care.	R/W
b7	SSLKP	SSL Signal Level Keeping	0: Negate all SSL signals on completion of transfer 1: Keep SSL signal level from the end of transfer until the beginning of the next access.	R/W
b11 to b8	SPB[3:0]	SPI Data Length Setting	b11 b8 0100 to 0111: 8 bits 1 0 0 0: 9 bits 1 0 0 1: 10 bits 1 0 1 0: 11 bits 1 0 1 1: 12 bits 1 1 0 0: 13 bits 1 1 0 1: 14 bits 1 1 1 0: 15 bits 1 1 1 1: 16 bits 0 0 0 0: 20 bits 0 0 0 1: 24 bits 0010, 0011: 32 bits.	R/W
b12	LSBF	SPI LSB First	0: MSB first 1: LSB first.	R/W
b13	SPNDEN	SPI Next-Access Delay Enable	0: Select next-access delay of 1 RSPCK + 2 PCLKA 1: Select next-access delay equal to the setting in the SPI Next-Access Delay Register (SPND).	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: Select SSL negation delay of 1 RSPCK 1: Select SSL negation delay equal to the setting in the SPI Slave Select Negation Delay Register (SSLND).	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: Select RSPCK delay of 1 RSPCK 1: Select RSPCK delay equal to the setting in the SPI Clock Delay Register (SPCKD).	R/W

The SPCMDm registers specify the transfer format for the SPI in master mode. Each channel has eight SPI Command Registers (SPCMD0 to SPCMD7). Some of the bits in the SPCMD0 register are used to set the transfer mode for the SPI in slave mode. The SPI in master mode sequentially references the SPCMDm registers based on the settings in the

SPSCR.SPSLN[2:0] bits and executes the serial transfer that is set in the referenced SPCMDm register.

Set the SPCMDm registers while the transmit buffer is empty (SPSR.SPTEF is 1 and data for the next transfer is not set) and before the setting of the data to be transmitted when that SPCMDm register is referenced.

The SPCMDm register referenced by the SPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If the contents of SPCMDm are changed while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1, do not perform subsequent operations.

CPHA bit (RSPCK Phase Setting)

The CPHA bit selects the RSPCK phase of the SPI in master or slave mode. Data communications between SPI modules require the same RSPCK phase setting between the modules.

CPOL bit (RSPCK Polarity Setting)

The CPOL bit selects the RSPCK polarity of the SPI in master or slave mode. Data communications between SPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] bits (Bit Rate Division Setting)

The BRDV[1:0] bits determine the bit rate. The bit rate is determined by the combination of the settings in the BRDV[1:0] bits and SPBR (see section 38.2.8, SPI Bit Rate Register (SPBR)). The SPBR settings determine the base bit rate. The BRDV[1:0] setting selects the bit rate obtained by dividing the base bit rate by 1, 2, 4, or 8. Different BRDV[1:0] bit settings can be specified in the SPCMDm registers. This enables execution of serial transfers at different bit rates for each command.

SSLA[2:0] bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLni signal assertion when the SPI performs serial transfers in master mode. When an SSLni signal is asserted, its polarity is determined by the value set in the associated SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLn0 pin acts as input).

When using the SPI in slave mode, set the SSLA[2:0] bits to 000b.

SSLKP bit (SSL Signal Level Keeping)

When the SPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLni signal level for the current command is to be kept or negated between the SSL negation associated with the current command and the SSL assertion associated with the next command. Setting the SSLKP bit to 1 enables a burst transfer. For details, see section 38.3.10.1, Master mode operation. When using the SPI in slave mode, set the SSLKP bit to 0.

SPB[3:0] bits (SPI Data Length Setting)

The SPB[3:0] bits specify the transfer data length for the SPI in master or slave mode. When the SPLW bit is 0, set these bits from 8 to 16 bits.

LSBF bit (SPI LSB First)

The LSBF bit specifies the data format of the SPI in master or slave mode to MSB-first or LSB-first.

SPNDEN bit (SPI Next-Access Delay Enable)

The SPNDEN bit specifies the next-access delay, the period from the time the SPI in master mode terminates a serial transfer and sets the SSLni signal inactive until the SPI enables the SSLni signal assertion for the next access. If the SPNDEN bit is 0, the SPI sets the next-access delay to 1 RSPCK + 2 PCLKA. If the SPNDEN bit is 1, the SPI inserts a next-access delay in accordance with the SPND setting.

When using the SPI in slave mode, set the SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

The SLNDEN bit specifies the SSL negation delay, the period from the time the SPI in master mode stops RSPCK oscillation until the SPI sets the SSLni signal to inactive. If the SLNDEN bit is 0, the SPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the SPI negates the SSL signal at the SSL negation delay determined by the SSLND setting.

When using the SPI in slave mode, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

The SCKDEN bit specifies the SPI clock delay, the period from the point when the SPI in master mode asserts the SSLni signal until the RSPCK starts oscillation. If the SCKDEN bit is 0, the SPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the SPI starts the oscillation of RSPCK at the RSPCK delay determined by the SPCKD setting.

When using the SPI in slave mode, set the SCKDEN bit to 0.

38.2.15 SPI Data Control Register 2 (SPDCR2)

Address(es): SPI0.SPDCR2 4007 2020h, SPI1.SPDCR2 4007 2120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BYSW
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BYSW	Byte Swap Operating Mode Select	0: Byte Swap OFF 1: Byte Swap ON	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPI Data Control Register2 (SPDCR2) is the setting register, that is to swap a transmit/receive data in byte units. When a data of transmit buffers copies to a shift register, it is to swap in byte units. When a data of shift register copies to a receive buffers, it is to swap in byte units.

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. When byte access is valid (SPDCR.SPBYT = 1), byte swap is invalid. When byte swap is valid, parity function must be invalid (SPCR2.SPPE bit = 0). Setting change of BYSW bit must be SPCR.SPE bit = 0.

A data after byte swap is different by a data length (setting of SPCMD.SPB[3:0]).

When byte swap, A data length (setting of SPB[3:0]) must be set to 32 bit or 16bit. Other case of data length (that is 8 to 15, 20, 24 bit length), byte swap is not guaranteed. Before swap and after swap are shown below (length data (32 bit/16 bit)).

- Length data 32bit (SPB[3:0] = 0010 or 0011)

  Before swap: [31:24] [23:16] [15:8] [7:0]

  After swap: [7:0] [15:8] [23:16] [31:24]

- Length data 16bit (SPB[3:0] = 1111)

  Before swap: [31:24] [23:16]

  After swap: [23:16] [31:24].

When byte access mode (SPDCR.SPBT = 1), byte swap setting is invalid.

When byte swap is valid, set parity function to invalid (SPCR2.SPPE = 0). When the parity function set to valid, the behavior is not guaranteed.

38.3 Operation

In this section, the serial transfer period refers to the period from the beginning of driving valid data to the fetching of the final valid data.

38.3.1 Overview of SPI Operation

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation).

The SPI mode can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. Table 38.5 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 38.5 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKn signal	Input	Output	Output/Hi-Z	Input	Output
MOSIn signal	Input	Output	Output/Hi-Z	Input	Output
MISOn signal	Output/Hi-Z	Input	Input	Output	Input
SSLn0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 to SSLn3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	-	-
Max transfer rate	PCLKA/4	PCLKA/2	PCLKA/2	PCLKA/4	PCLKA/2
Clock source	RSPCKn input	On-chip baud rate generator	On-chip baud rate generator	RSPCKn input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	-	-
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer trigger	SSL input active or RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported				
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2, *4	Supported*2, *4	Supported*2	Supported*2
Parity error detection	Supported*2,*3				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Table 38.5 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
Underrun error detection	Supported	Not supported	Not supported	Supported	Not supported

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR2.SCKASE bit is 1, overrun error detection is not performed.

38.3.2 Controlling the SPI Pins

The SPI can switch pin states based on the settings in the MSTR, MODFEN, and SPMS bits in SPCR and the PmnPFS.NCODR bit for the I/O ports. Table 38.6 lists the relationship between the pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects CMOS output. Setting it to 1 selects open-drain output. The I/O port settings must follow this relationship.

Table 38.6 Relationship between pin states and bit settings

Mode	Pin	Pin state*2 PmnPFS.NCODR bit for I/O ports = 0	 PmnPFS.NCODR bit for I/O ports = 1
Single master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.
Note 3. When SSLn0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z.
Note 5. These pins are available for use as I/O port pins.

The SPI in single master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) based on the MOIFE and

MOIFV bit settings in SPPCR, as listed in Table 38.7.

Table 38.7 MOSI signal value determination during SSL negation

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

38.3.3 SPI System Configuration Examples

38.3.3.1 Single master and single slave with the MCU as a master

Figure 38.5 shows a single-master and single-slave SPI system configuration example where the MCU is a master. In the single-master/single-slave configuration, the SSLn0 to SSLn3 outputs of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in the selected state.*1

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

Note 1. In the transfer format configured when the SPCMDm.CPHA bit is 0, the SSL signal cannot be fixed to the active level for some slave devices. In situations where the SSL signal cannot be fixed, the SSLni output of the MCU must be connected to the SSL input of the slave device.

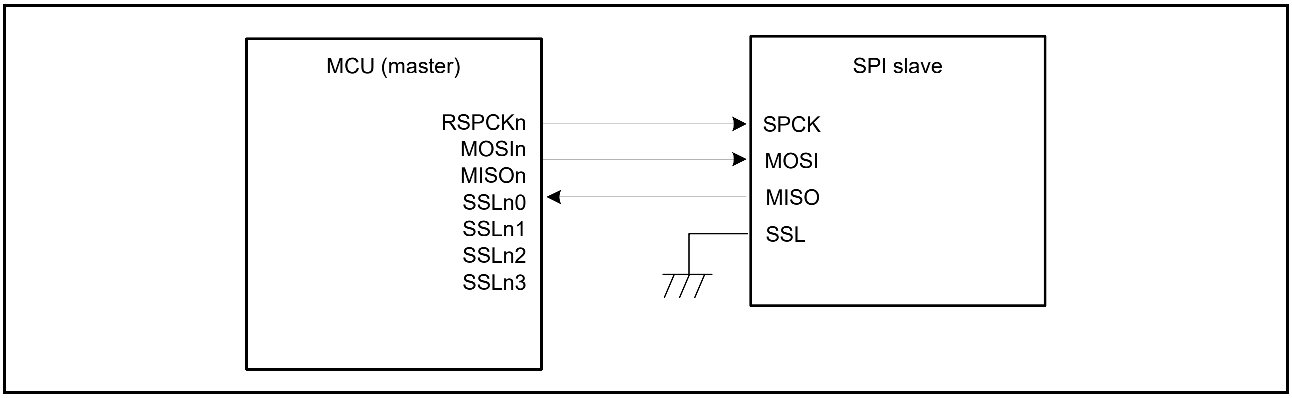


Figure 38.5 Single-master/single-slave configuration example with the MCU as a master

38.3.3.2 Single master and single slave with the MCU as a slave

Figure 38.6 shows a single-master/single-slave SPI system configuration example where the MCU is a slave. When the MCU is a slave, the SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI signals. The MCU (slave) drives the MISOn signal.*1

In the single-slave configuration when the SPCMDm.CPHA bit is set to 1, the SSLn0 input of the MCU (slave) is fixed to the low level and the MCU (slave) stays selected. This enables serial transfer execution (Figure 38.7).

Note 1. When SSLn0 is at the non-active level, the pin state is Hi-Z.

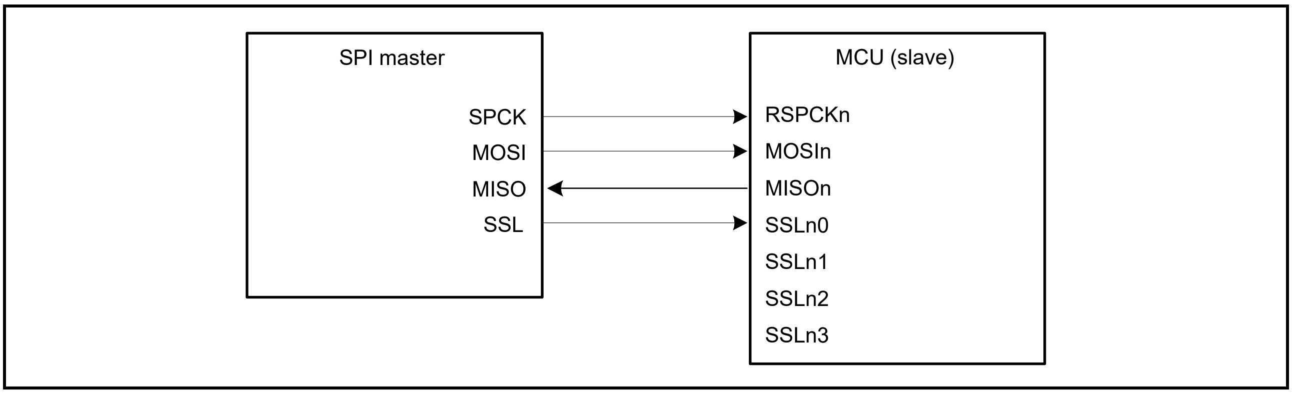


Figure 38.6 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0

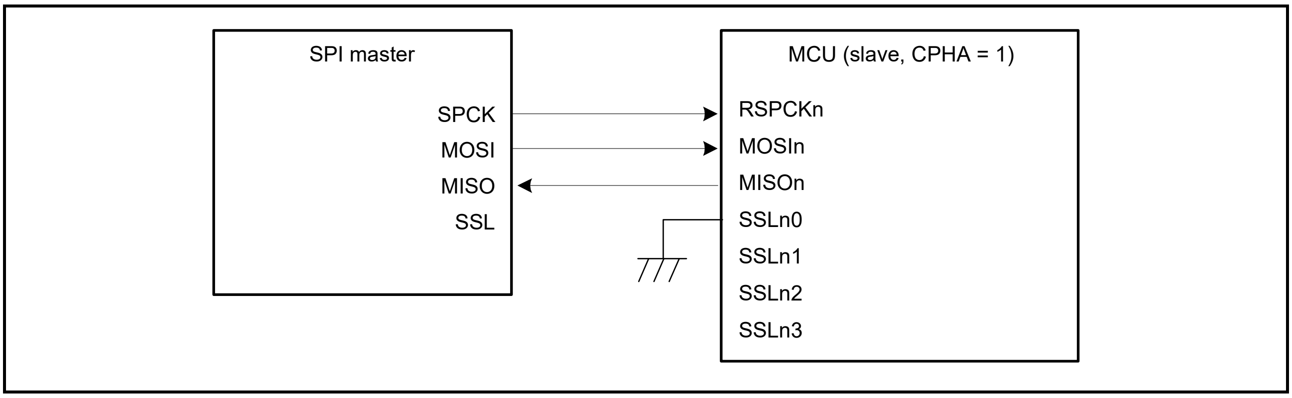


Figure 38.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

38.3.3.3 Single master and multi slave with the MCU as a master

Figure 38.8 shows a single-master/multi-slave SPI system configuration example where the MCU is a master. In this example, the SPI system includes the MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 to 3. The MISO outputs of SPI slaves 0 to 3 are all connected to the MISOn input of the MCU (master). The SSLn0 to SSLn3 outputs of the MCU (master) are connected to the SSL inputs of SPI slaves 0 to 3, respectively.

The MCU (master) drives the RSPCKn, MOSIn, and SSLn0 to SSLn3 signals. Of the SPI slaves 0 to 3, the slave that receives low-level input into the SSL input drives the MISO signal.

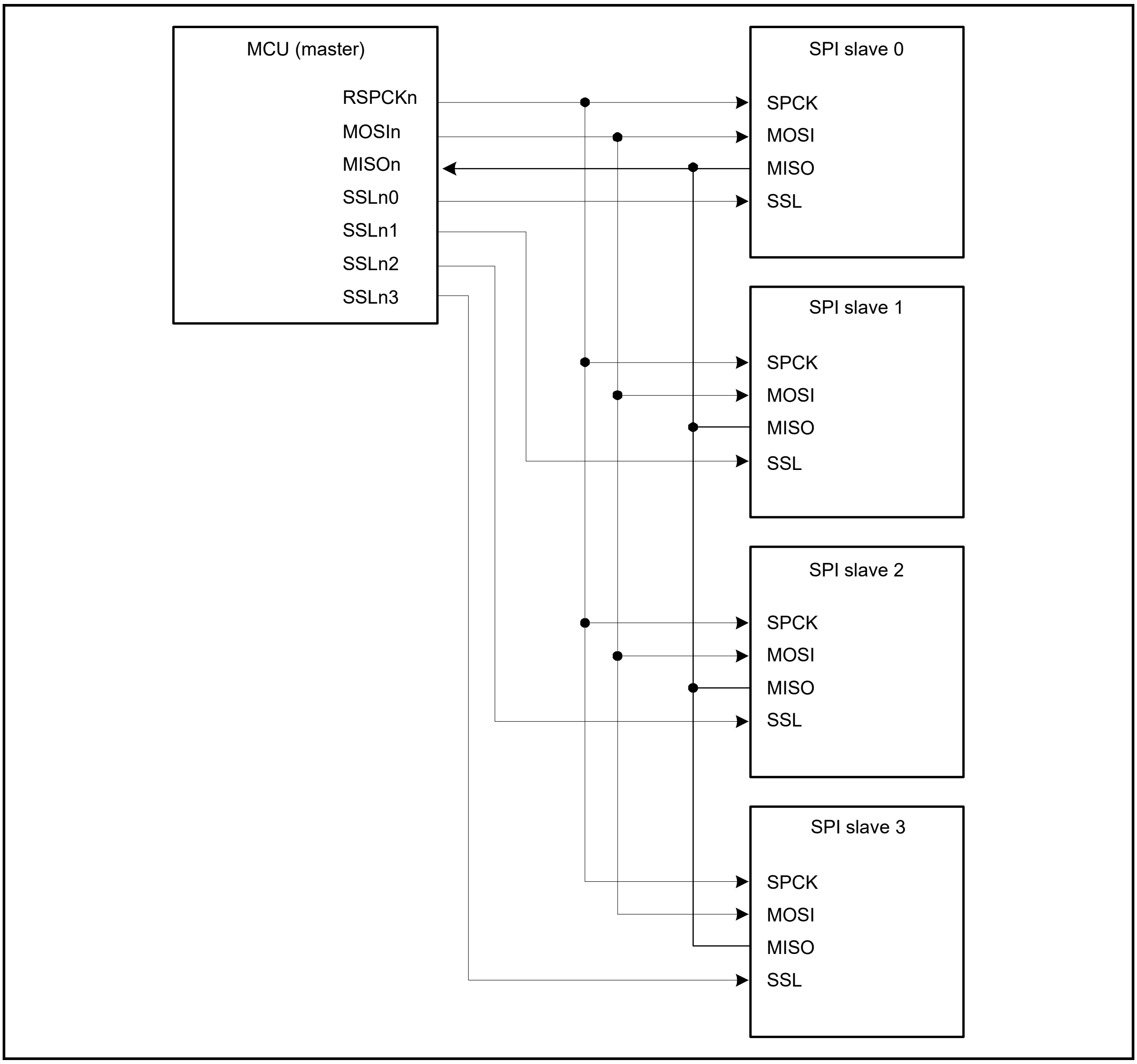


Figure 38.8 Single master/multi-slave configuration example with the MCU as a master

38.3.3.4 Single master and multi slave with the MCU as a slave

Figure 38.9 shows a single-master and multi-slave SPI system configuration example where the MCU is a slave. In this example, the SPI system includes an SPI master and two MCUs (slaves X and Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slaves X and Y). The MISOn outputs of the MCUs (slaves X and Y) are all connected to the MISO input of the SPI master. The SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slaves X and Y, respectively).

The SPI master drives the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slaves X and Y), the slave that receives low-level input into the SSLn0 input drives the MISOn signal.

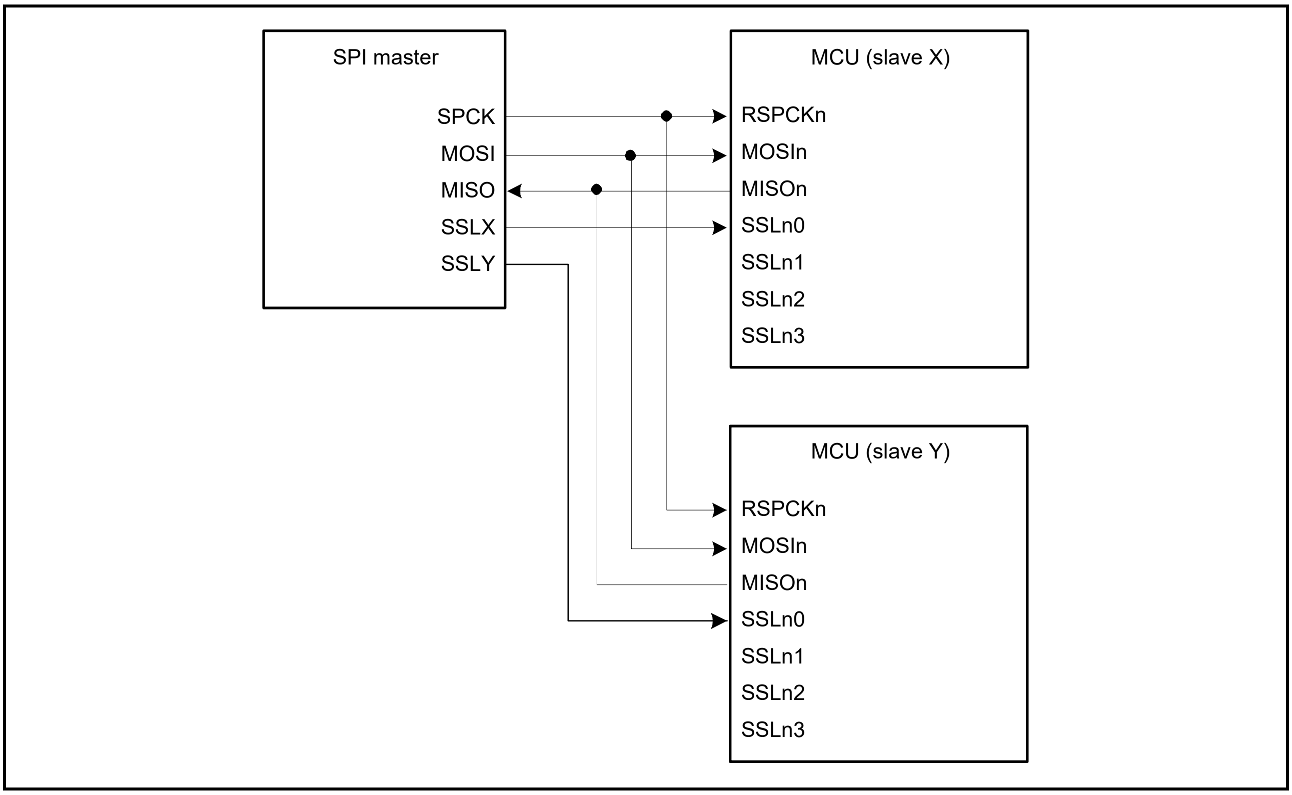


Figure 38.9 Single-master/multi-slave configuration example with the MCU as a slave

38.3.3.5 Multi master and multi slave with the MCU as a master

Figure 38.10 shows a multi-master/multi-slave SPI system configuration example where the MCU is a master. In this example, the SPI system includes two MCUs (masters X and Y) and two SPI slaves (SPI slaves 1 and 2).

The RSPCKn and MOSIn outputs of the MCUs (masters X and Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOn inputs of the MCUs (masters X and Y). Any generic port Y output from the MCU (master X) is connected to the SSLn0 input of the MCU (master Y). Any generic port X output of the MCU (master Y) is connected to the SSLn0 input of the MCU (master X). The SSLn1 and SSLn2 outputs of the MCUs (masters X and Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of the MCU is not required.

The MCU drives the RSPCKn, MOSIn, SSLn1, and SSLn2 signals when the SSLn0 input level is high. When the SSLn0 input level is low, the MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the SPI bus directly to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives the MISO signal.

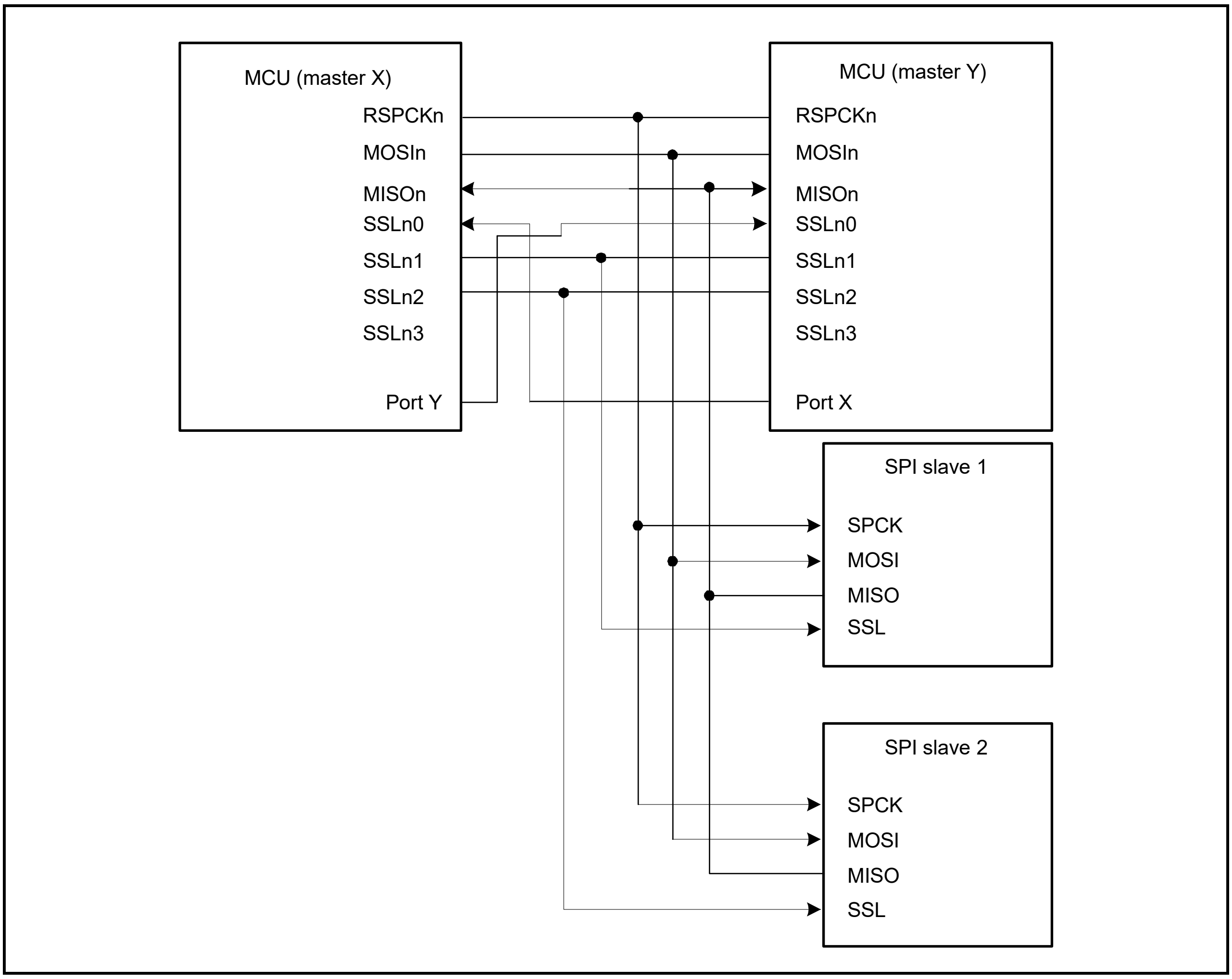


Figure 38.10 **Multi-master/multi-slave configuration example with the MCU as a master**

38.3.3.6 Master and slave in clock synchronous mode with the MCU as a master

Figure 38.11 shows master and slave in clock synchronous mode where the MCU is a master. In this configuration, SSLn0 to SSLn3 of the MCU (master) are not used.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

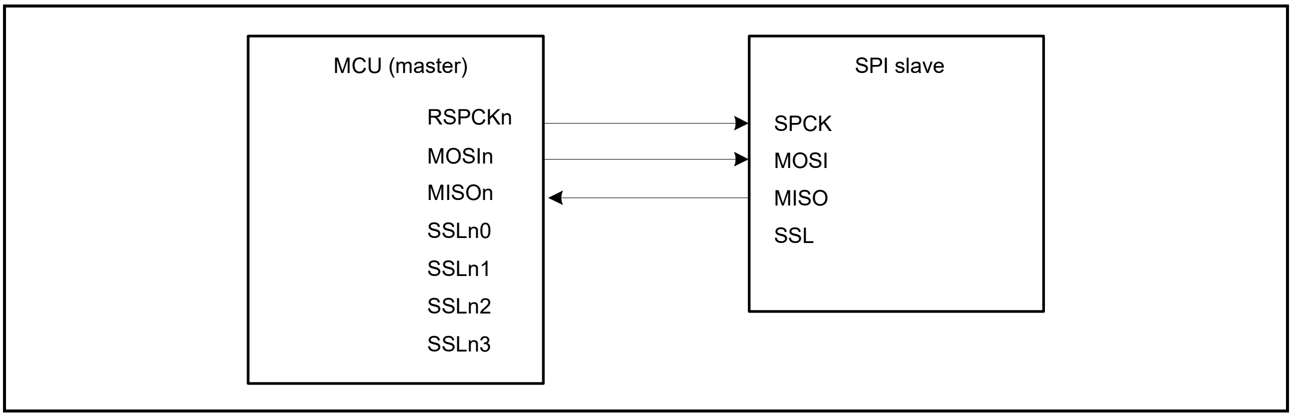


Figure 38.11 **Configuration example of master/slave in clock synchronous mode with the MCU as a master**

38.3.3.7 Master and slave in clock synchronous mode with the MCU as a slave

Figure 38.12 shows a master and slave in clock synchronous mode configuration where the MCU is a slave. When the MCU operates as a slave in clock synchronous mode, the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfers in the single slave configuration when the SPCMDm.CPHA bit is set to 1.

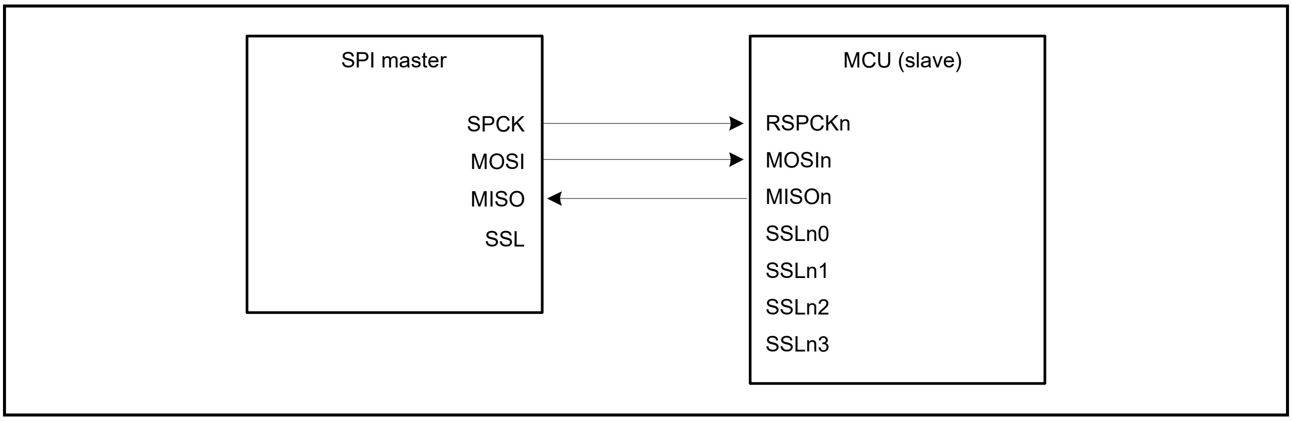


Figure 38.12 Configuration example of master and slave in clock synchronous mode with the MCU as a slave and CPHA = 1

38.3.4 Data Formats

The data format of the SPI depends on the settings in SPI Command Register m (SPCMDm) (m = 0 to 7) and the parity enable bit in SPI Control Register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the SPI treats the range from the LSB bit in the SPI Data Register (SPDR/SPDR_HA) to the bit corresponding to the selected data length, as transfer data.

This section shows the format of one frame of data before or after transfer.

(a) Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[3:0]).

(b) Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

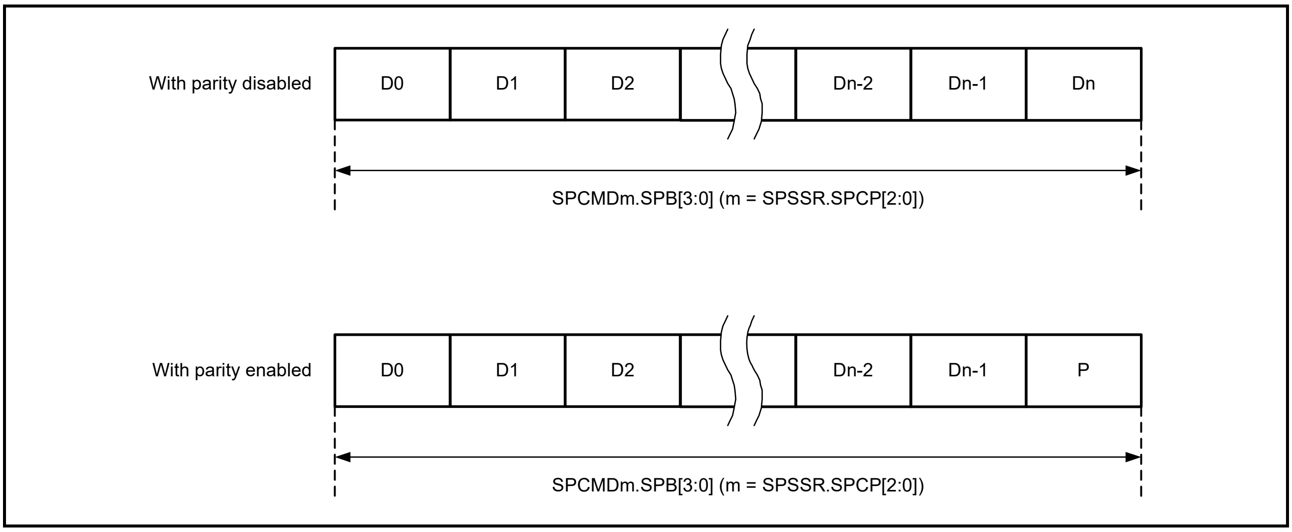


Figure 38.13 Data format with parity disabled and enabled

38.3.4.1 Operation when parity is disabled (SPCR2.SPPE = 0)

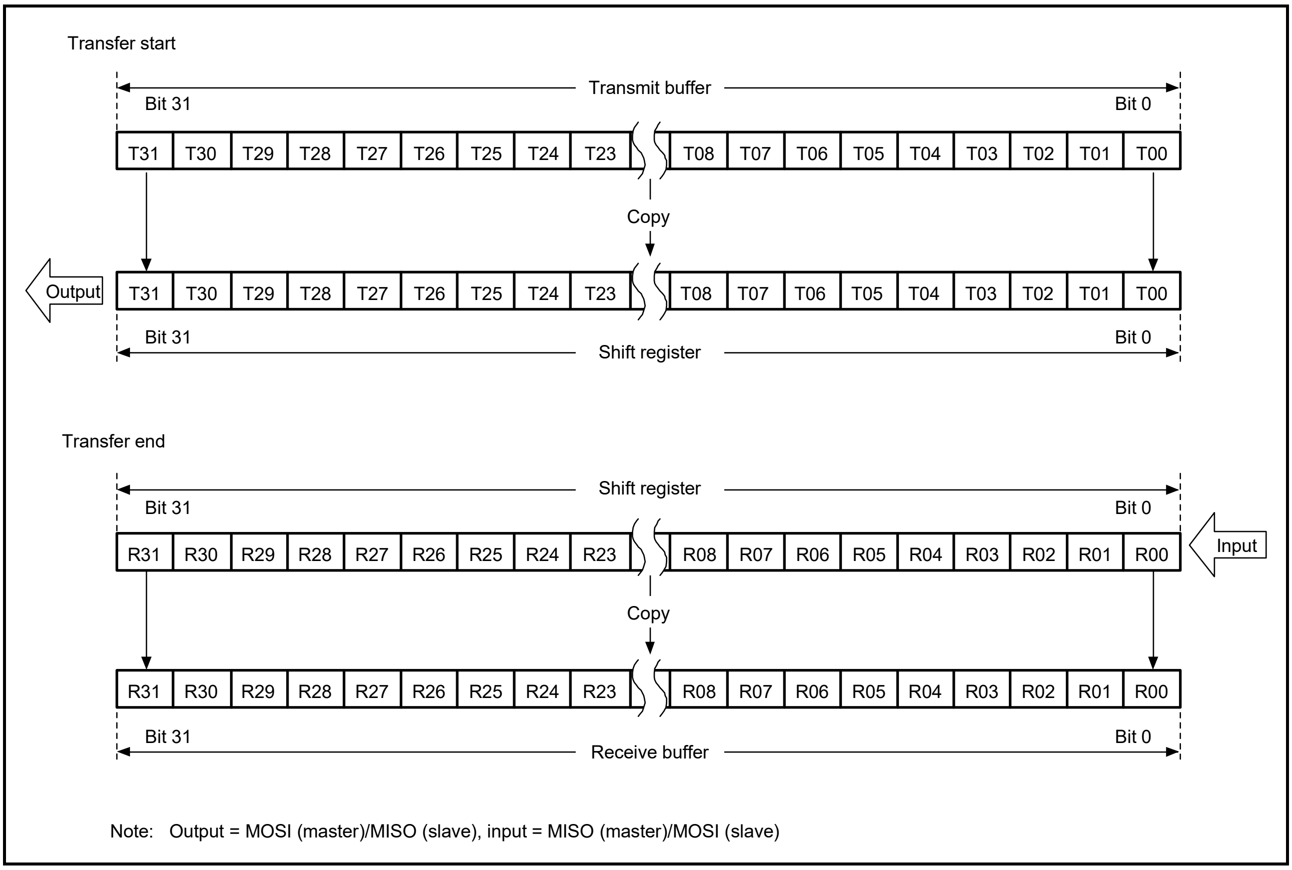
When parity is disabled, data for transmission is copied to the shift register with no pre-processing. This section describes the connection between the SPI Data Register (SPDR/SPDR_HA) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

(1) MSB-first transfer with 32-bit data

Figure 38.14 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity disabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00, in that order.

In reception, received data is shifted in bit-by-bit through bit [0] of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

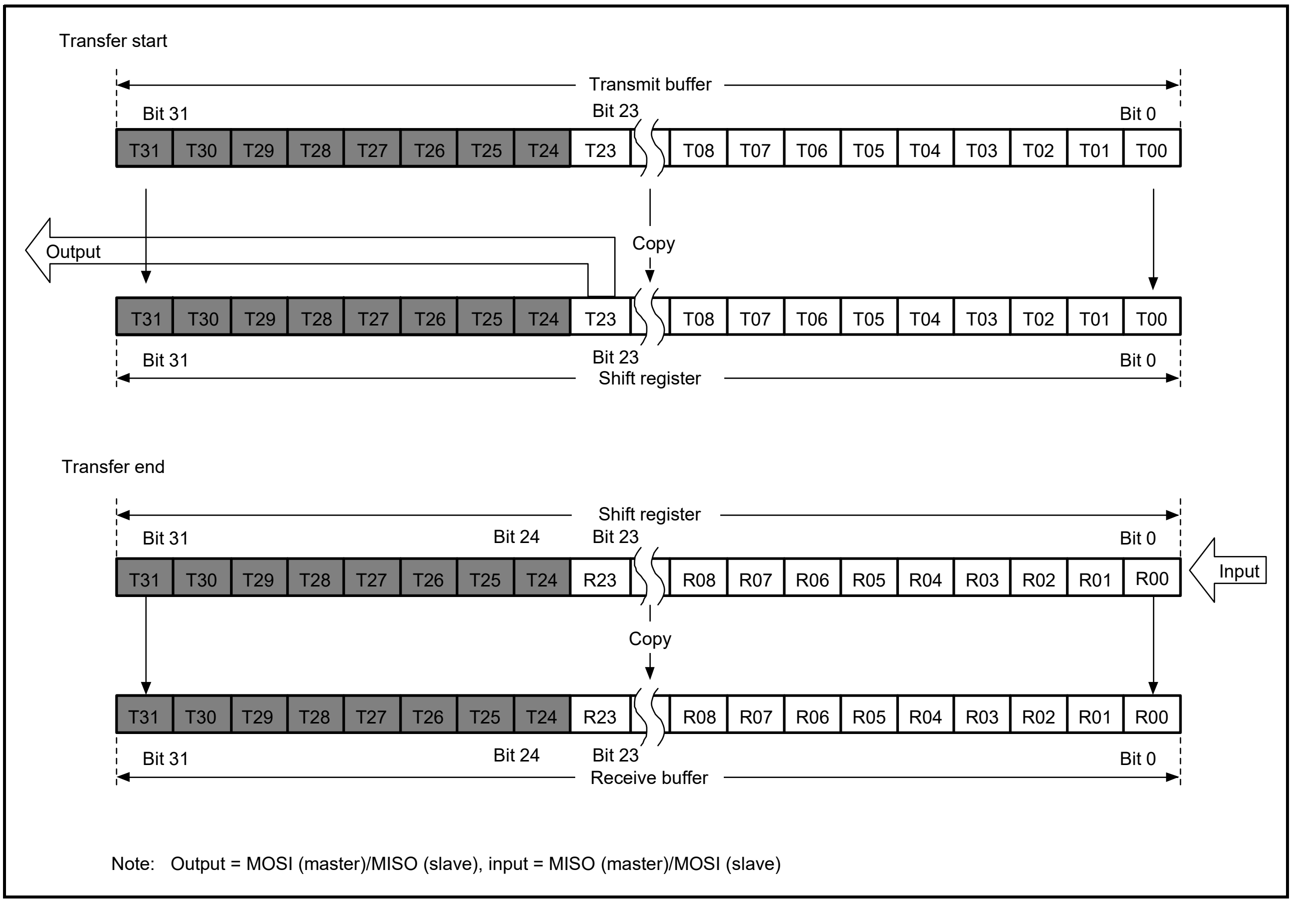
Figure 38.14 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 38.15 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity disabled, an SPI data length of 24 bits, and MSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T23 to T22, and continuing to T00, in that order.

In reception, received data is shifted in bit-by-bit through bit [0] of the shift register. When the R23 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

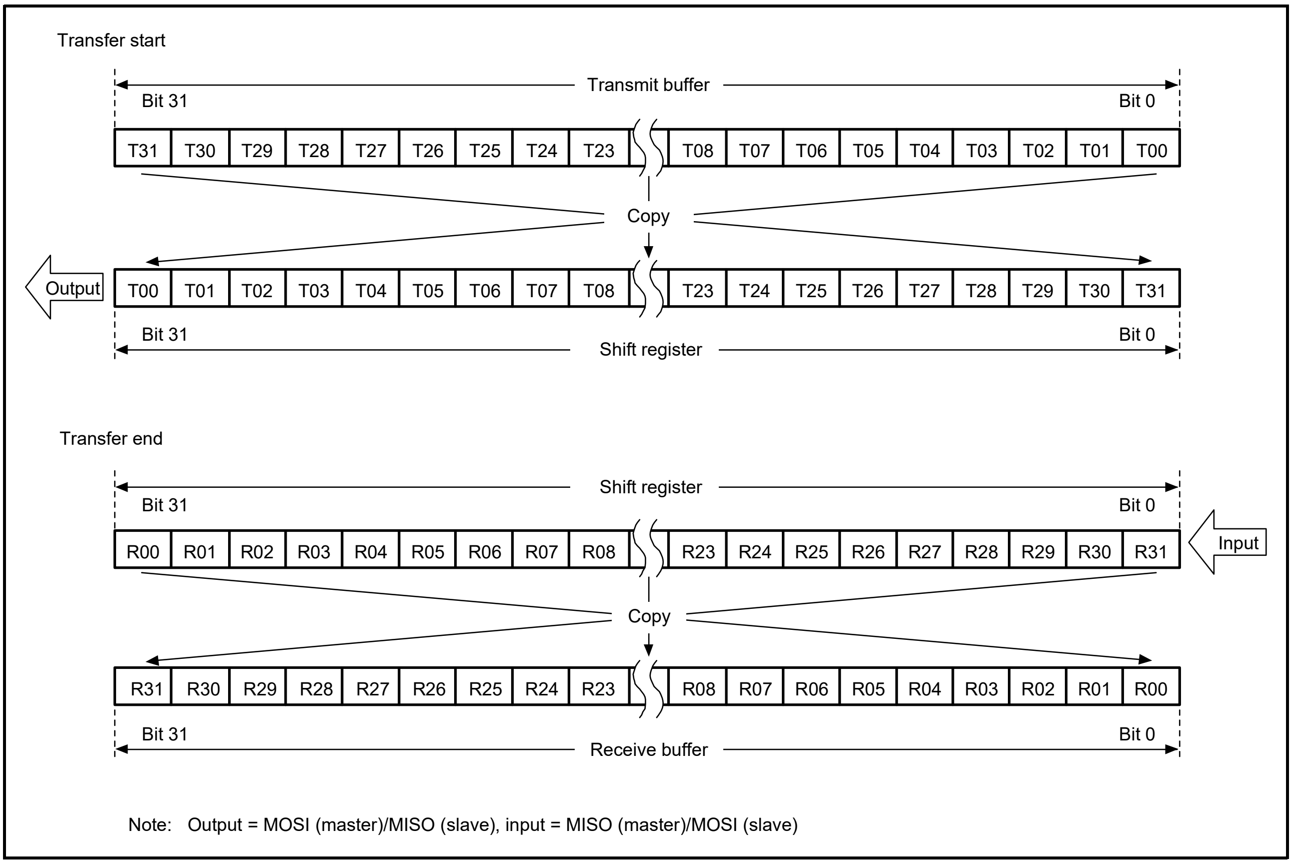
Figure 38.15 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 38.16 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity disabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T31, in that order.

In reception, received data is shifted in bit-by-bit through bit [0] of the shift register. When the R00 to R31 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 38.16 LSB-first transfer with 32-bit data and parity disabled

(4) LSB-first transfer with 24-bit data

Figure 38.17 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity disabled, an SPI data length of 24 bits, and LSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T23, in that order.

In reception, received data is shifted in bit-by-bit through bit [8] of the shift register. When the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

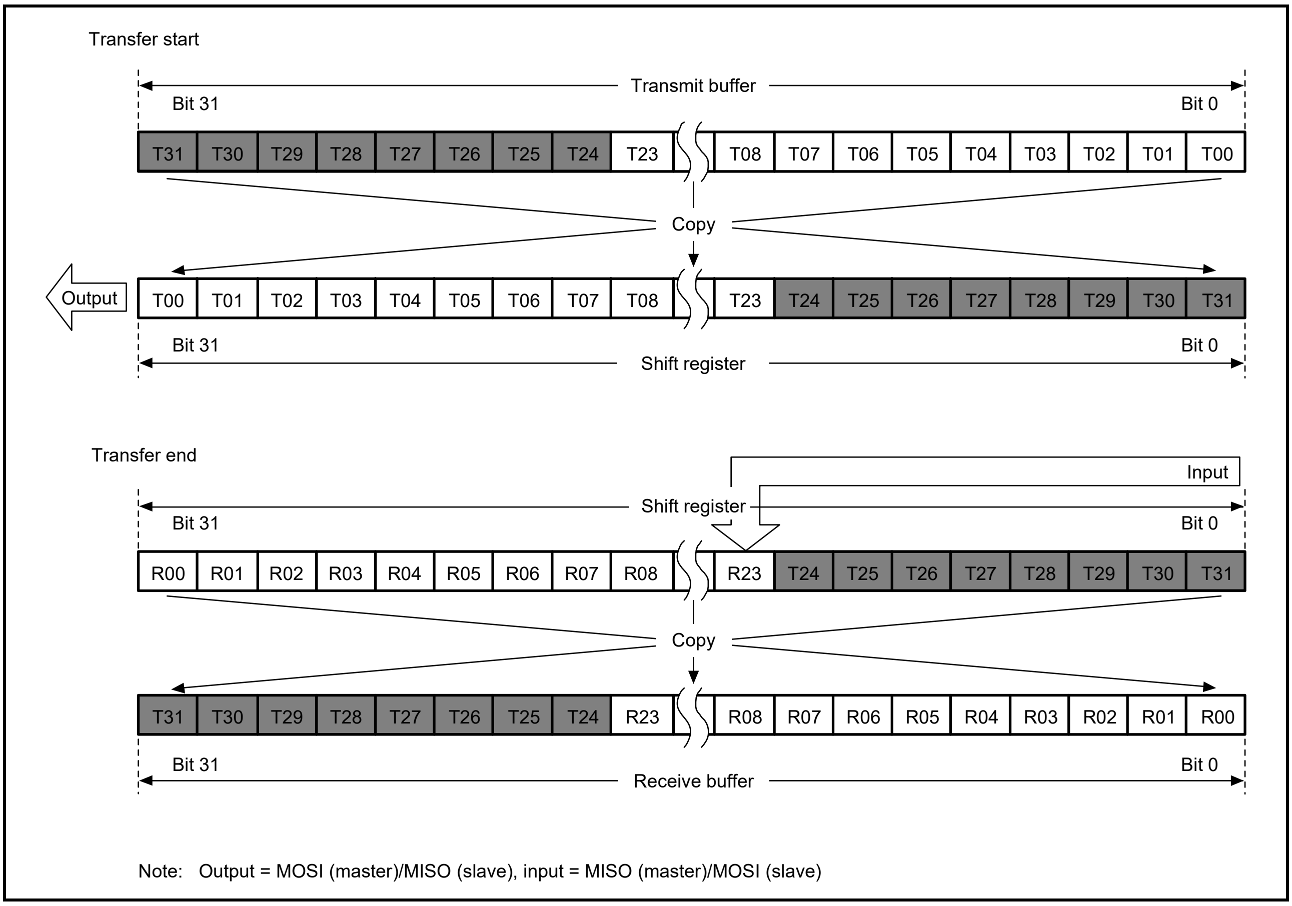


Figure 38.17 LSB-first transfer with 24-bit data and parity disabled

38.3.4.2 Operation when parity is enabled (SPCR2.SPPE = 1)

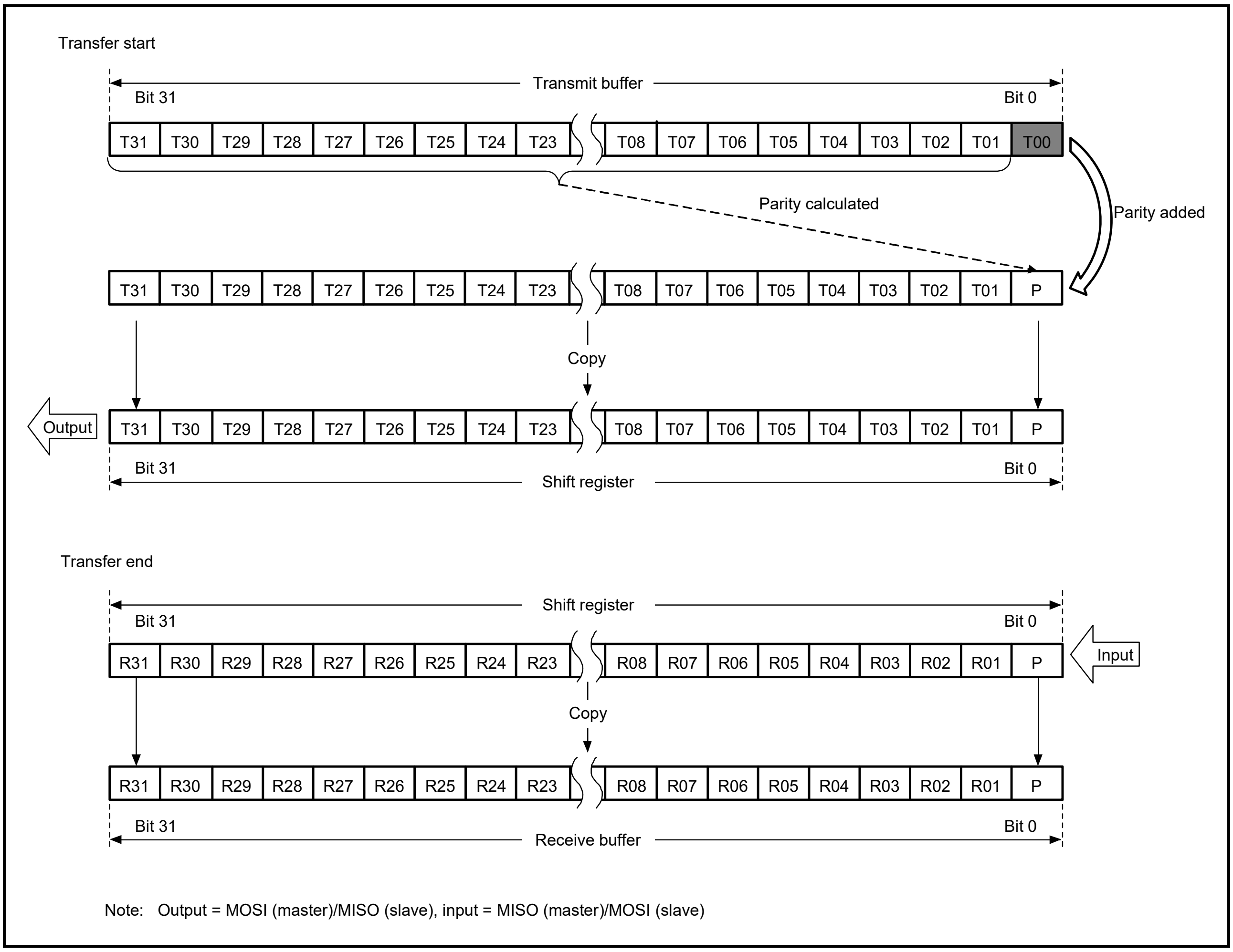
When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB-first transfer with 32-bit data

Figure 38.18 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity enabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data is transmitted in the order T31, T30, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit [0] of the shift register. When the R31 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, data from R31 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

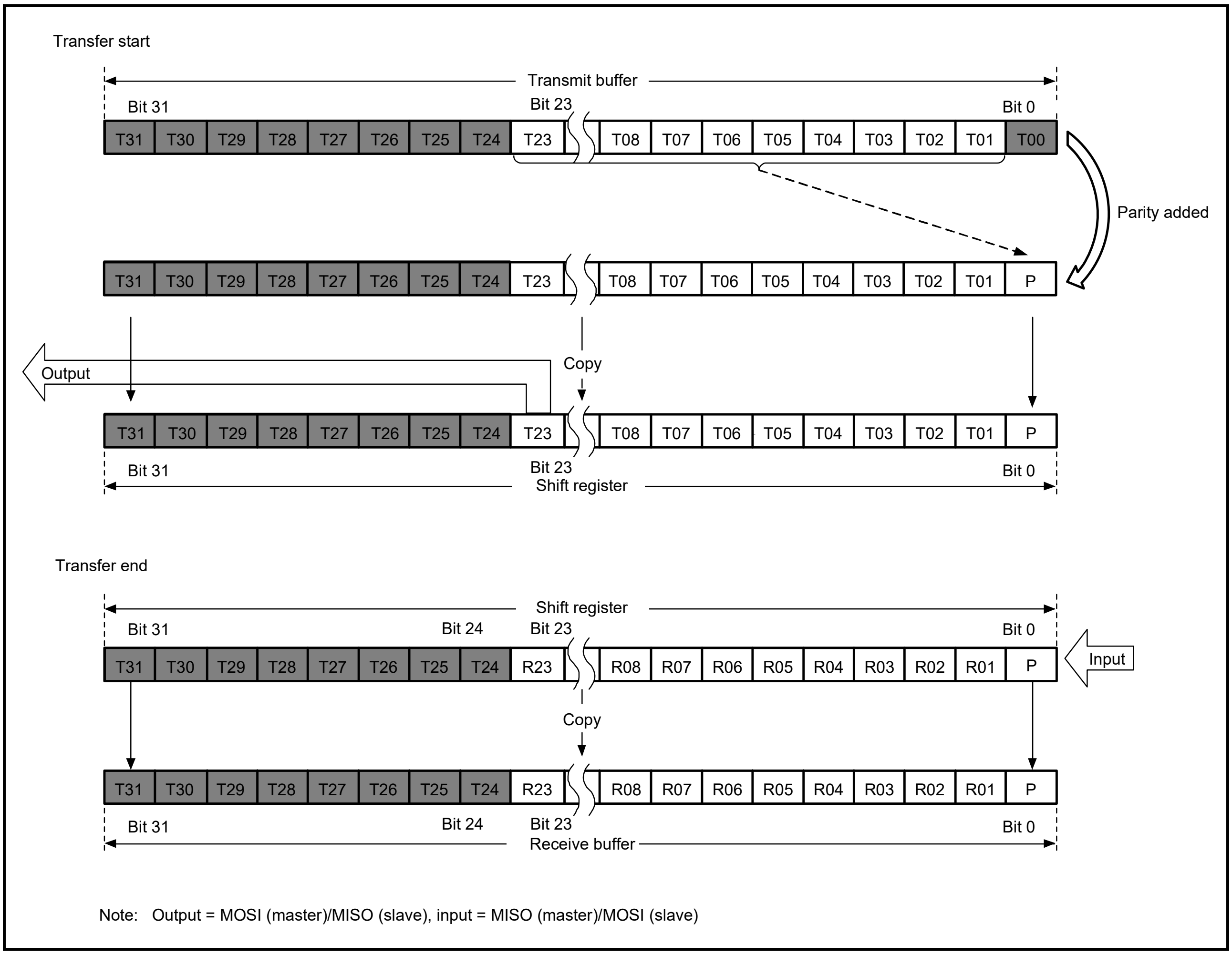
Figure 38.18 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 38.19 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity enabled, an SPI data length of 24 bits and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data is transmitted in the order T23, T22, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit [0] of the shift register. When the R23 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, data from R23 to P is checked for parity. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 38.19 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 38.20 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity enabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data is transmitted in the order T00, T01, …, T30, and P.

In reception, received data is shifted in bit-by-bit through bit [0] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, data from R00 to P is checked for parity.

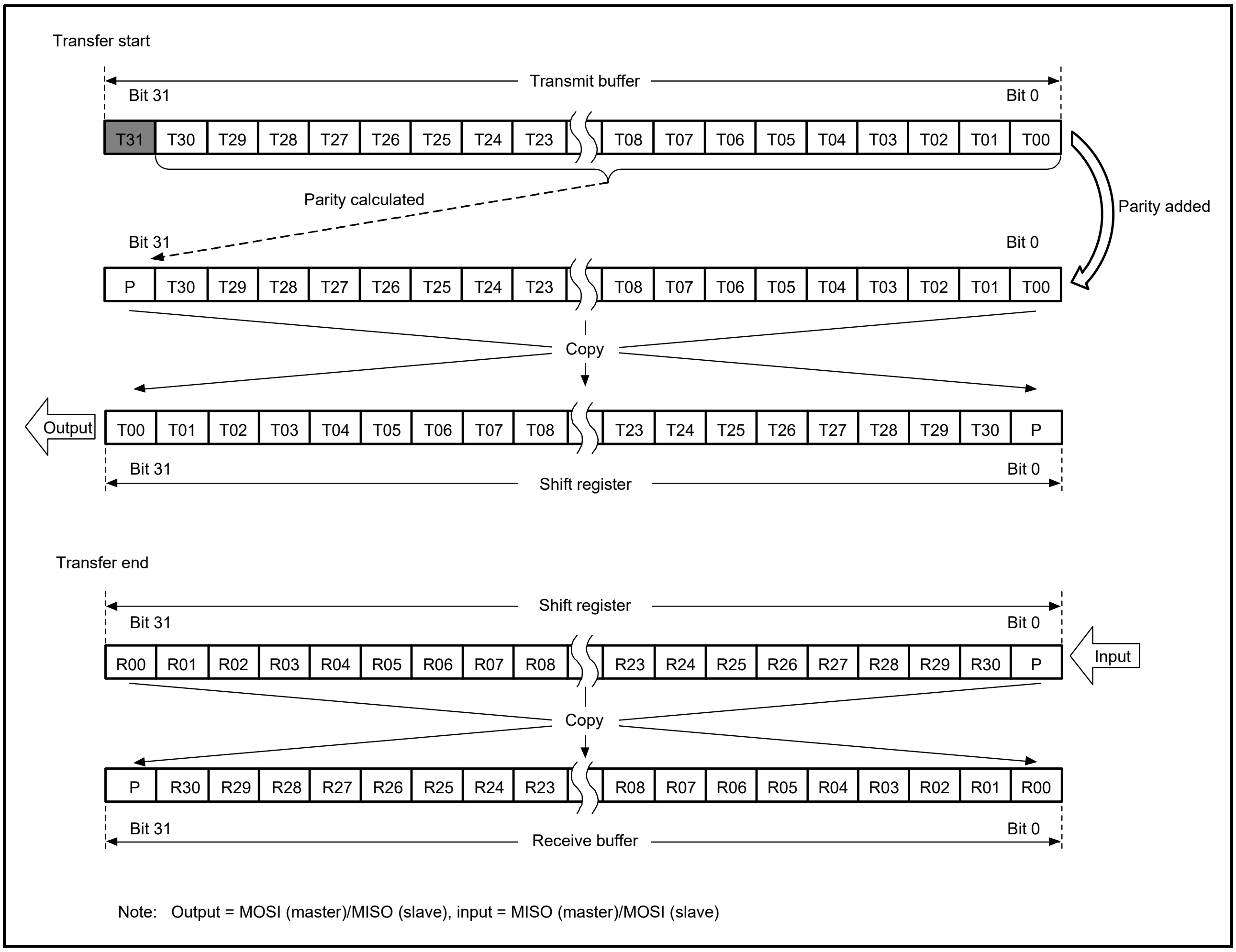


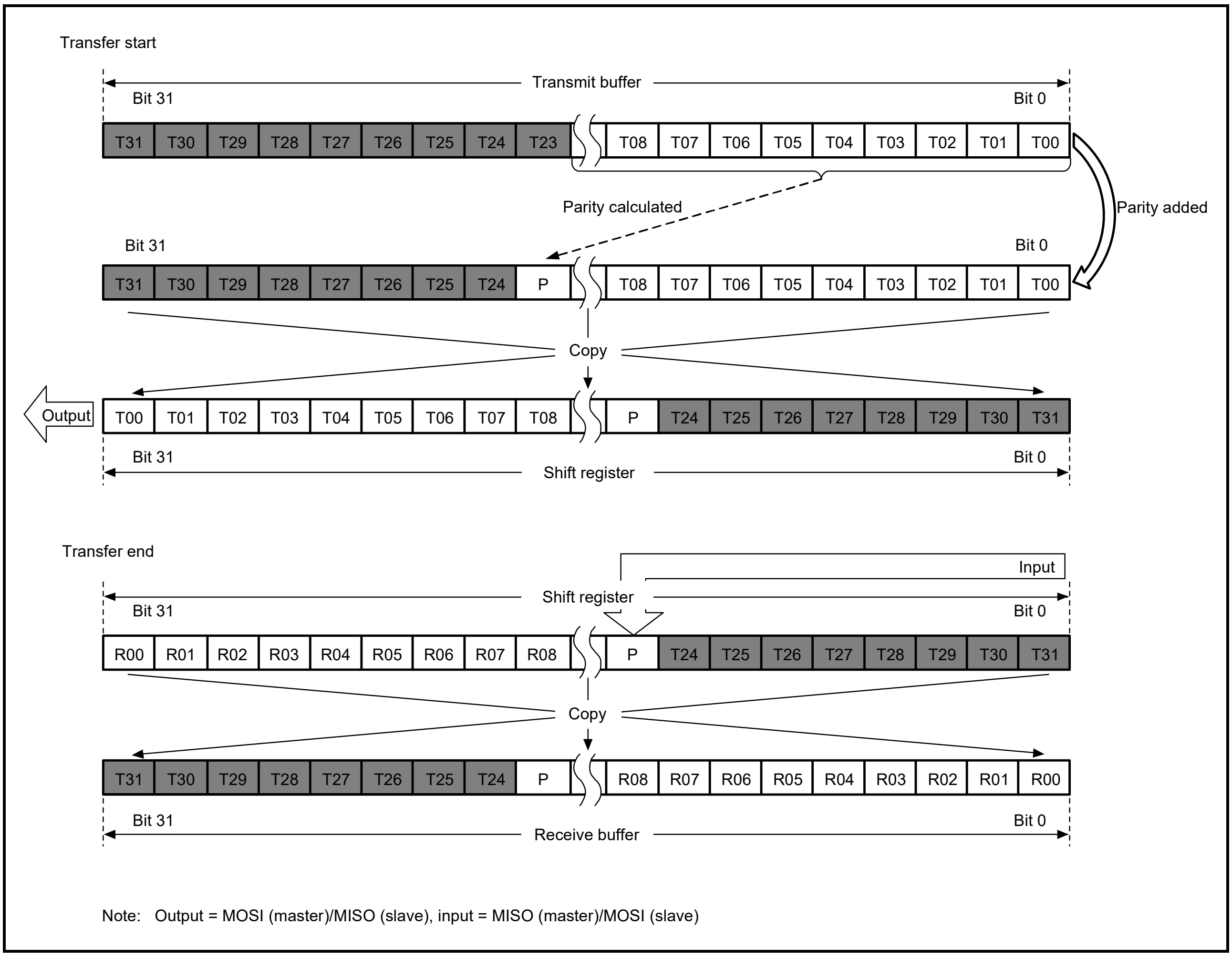
Figure 38.20 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 38.21 shows the transfer operations of the SPI Data Register (SPDR) and the shift register with parity enabled, an SPI data length of 24 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data is transmitted in the order T00, T01, …, T22, and P.

In reception, received data is shifted in bit-by-bit through bit [8] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, data from R00 to P is checked for parity. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 38.21 LSB-first transfer with 24-bit data and parity enabled

38.3.5 Transfer Formats

38.3.5.1 Transfer format when CPHA = 0

Figure 38.22 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS = 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In the figure, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0, and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 38.3.2, Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals commences on an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled every 1 RSPCK cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing as it only affects the signal polarity.

t1 denotes the period from an SSLni signal assertion to RSPCKn oscillation (RSPCK delay). t2 denotes the period from the termination of RSPCKn oscillation to an SSLni signal negation (SSL negation delay). t3 denotes the period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the SPI system. For a description of t1, t2, and t3 when the SPI is in master mode, see section 38.3.10.1, Master mode operation.

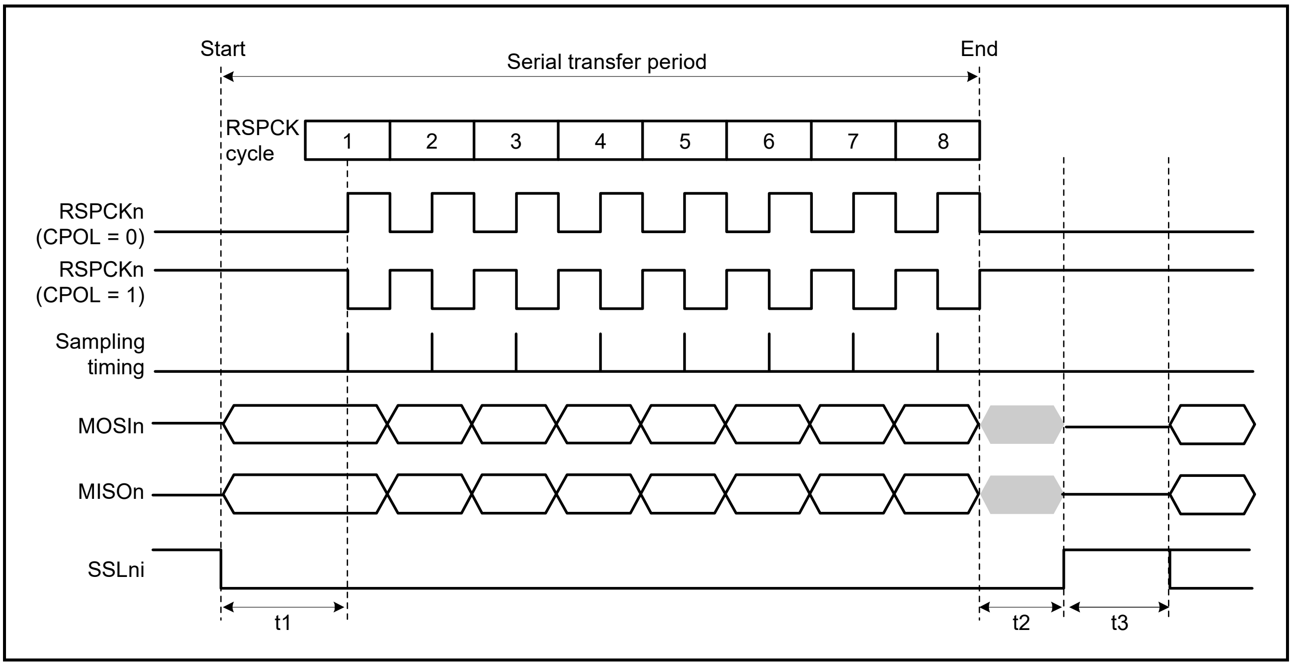


Figure 38.22 SPI transfer format when CPHA = 0

38.3.5.2 When CPHA = 1

Figure 38.23 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 38.23, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCK (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave). For details, see section 38.3.2, Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOn signal commences on an SSLni signal assertion. The output of valid data to the MOSIn and MISOn signals commences at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see section 38.3.10.1, Master mode operation.

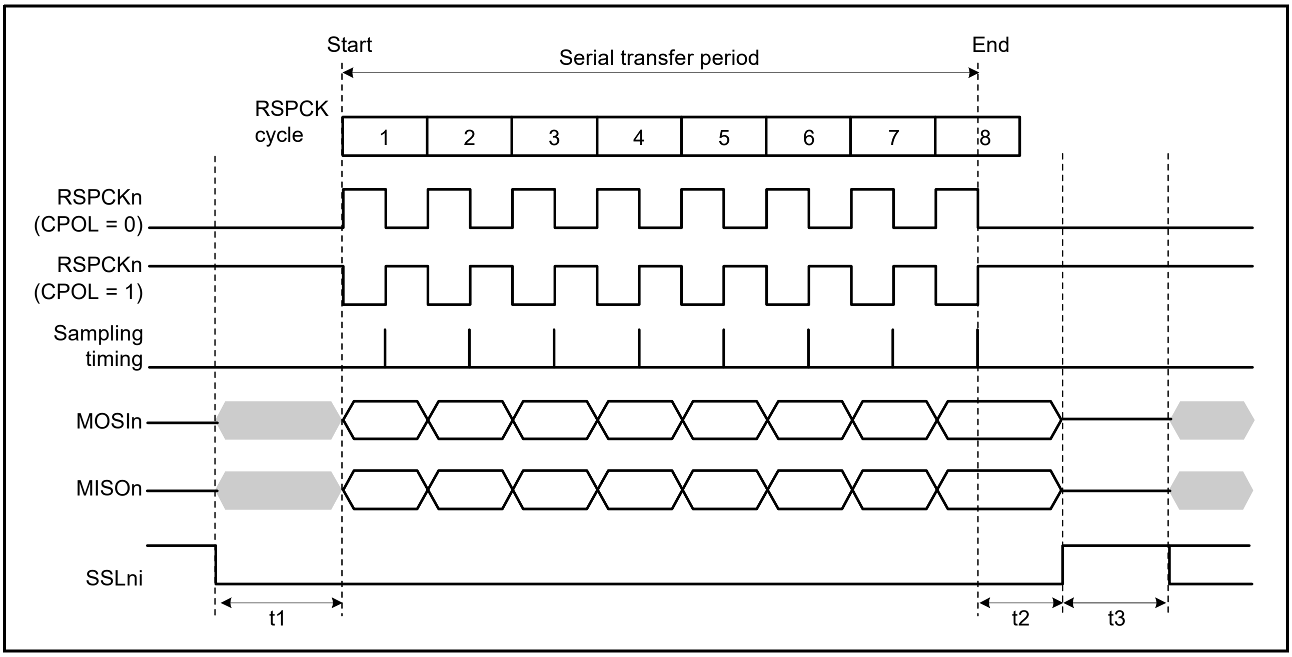


Figure 38.23 SPI transfer format when CPHA = 1

38.3.6 Data Transfer Modes

Full-duplex synchronous serial communications or transmit operations can only be selected in the communications operating mode select bit (SPCR.TXMD). The SPDR/SPDR_HA access shown in Figure 38.24 and Figure 38.25 indicate the condition of access to the register, where W denotes a write cycle.

38.3.6.1 Full-duplex synchronous serial communications (SPCR.TXMD = 0)

Figure 38.24 shows an example of operation where the communications operating mode select bit (SPCR.TXMD) is set to 0. In this example, the SPI performs an 8-bit serial transfer when SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, indicating the number of transferred bits.

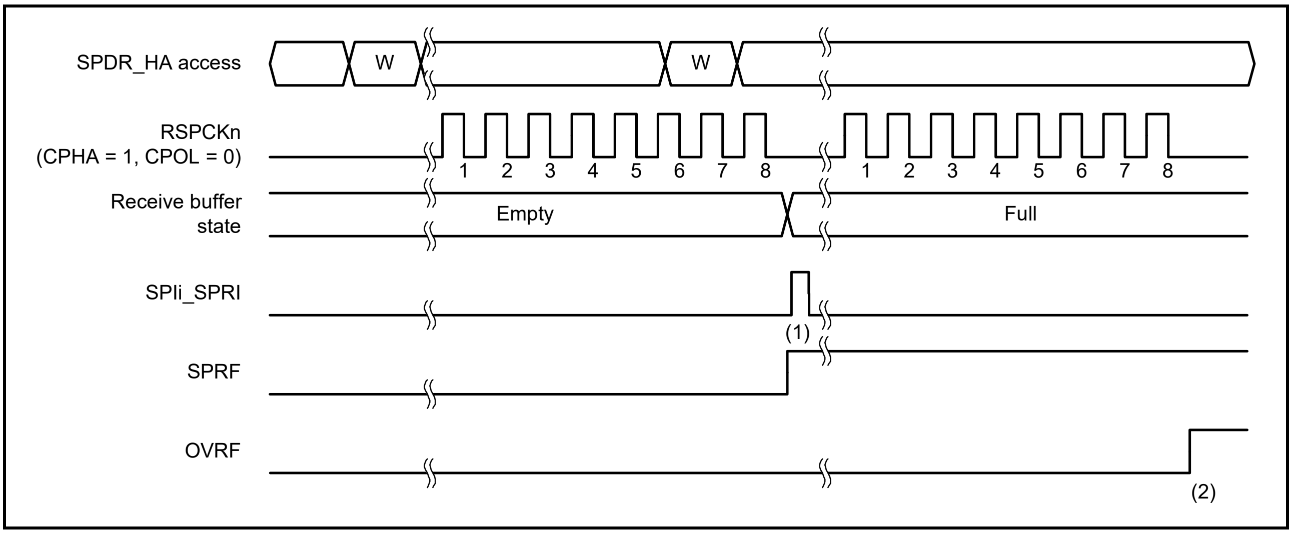


Figure 38.24 Operation example when SPCR.TXMD = 0

The operation of the flags at times (1) and (2) in the figure is as follows:

1. When a serial transfer ends with the receive buffer of SPDR_HA empty, the SPI generates a receive buffer full interrupt request (SPIi_SPRI) (the SPI sets the SPSR.SPRF flag to 1) and copies the received data in the shift

register to the receive buffer.

2. When a serial transfer ends with the receive buffer of SPDR_HA holding data that was received in the previous serial transfer, the SPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

38.3.6.2 Transmit operations only (SPCR.TXMD = 1)

Figure 38.25 shows an example of operation where the communications operating mode select bit (SPCR.TXMD) is set to 1. In this example, the SPI performs an 8-bit serial transfer when SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, indicating the number of transferred bits.

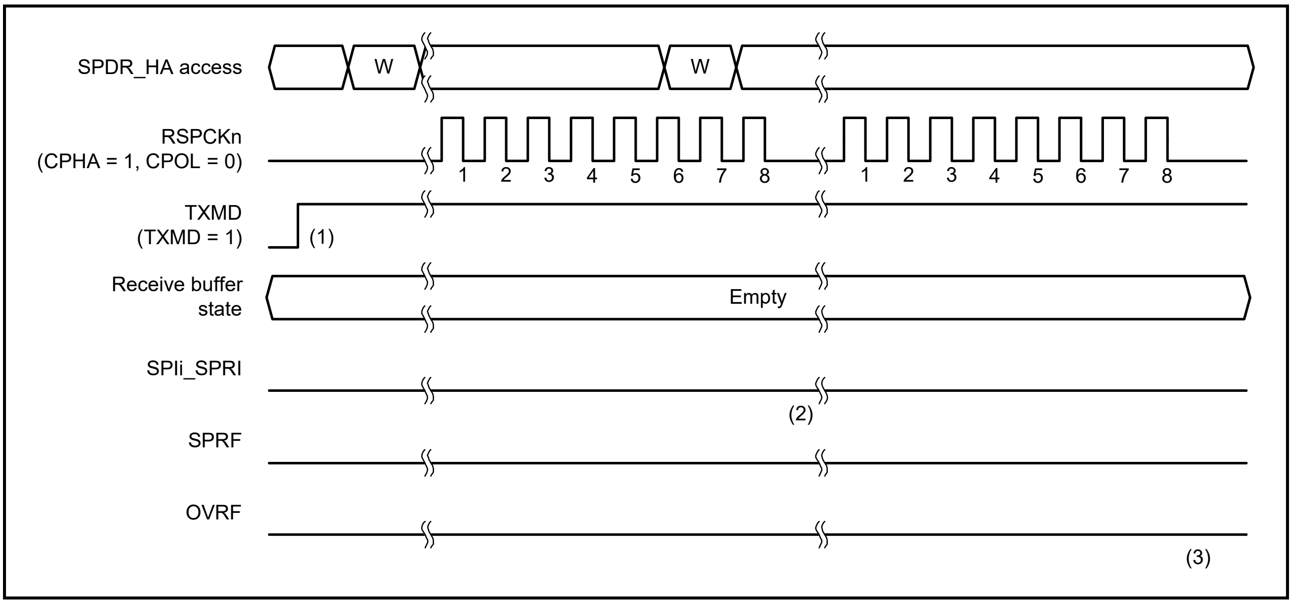


Figure 38.25 Operation example when SPCR.TXMD = 1

The operation of the flags at times (1) to (3) in the figure is as follows:

1. Make sure there is no data left in the receive buffer (the SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering transmit-only mode (SPCR.TXMD = 1).
2. When a serial transfer ends with the receive buffer of SPDR_HA empty, if the transmit-only mode is selected (SPCR.TXMD = 1), the SPSR.SPRF flag retains the value of 0, and the SPI does not copy the data in the shift register to the receive buffer.
3. Because the receive buffer of SPDR_HA does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

In transmit-only mode (SPCR.TXMD = 1), the SPI transmits data but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at times (1) to (3).

38.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 38.26 and Figure 38.27 show examples of operation of the transmit buffer empty interrupt (SPIi_SPTI) and the receive buffer full interrupt (SPIi_SPRI). The SPDR_HA register accesses shown in these figures indicate the condition of access to the register, where W denotes a write cycle and R a read cycle. In the example in Figure 38.26, the SPI performs an 8-bit serial transfer when SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 0, and the SPCMDm.CPOL bit is 0. In the example in Figure 38.27, the SPI performs an 8-bit serial transfer when SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, indicating the number of transferred bits.

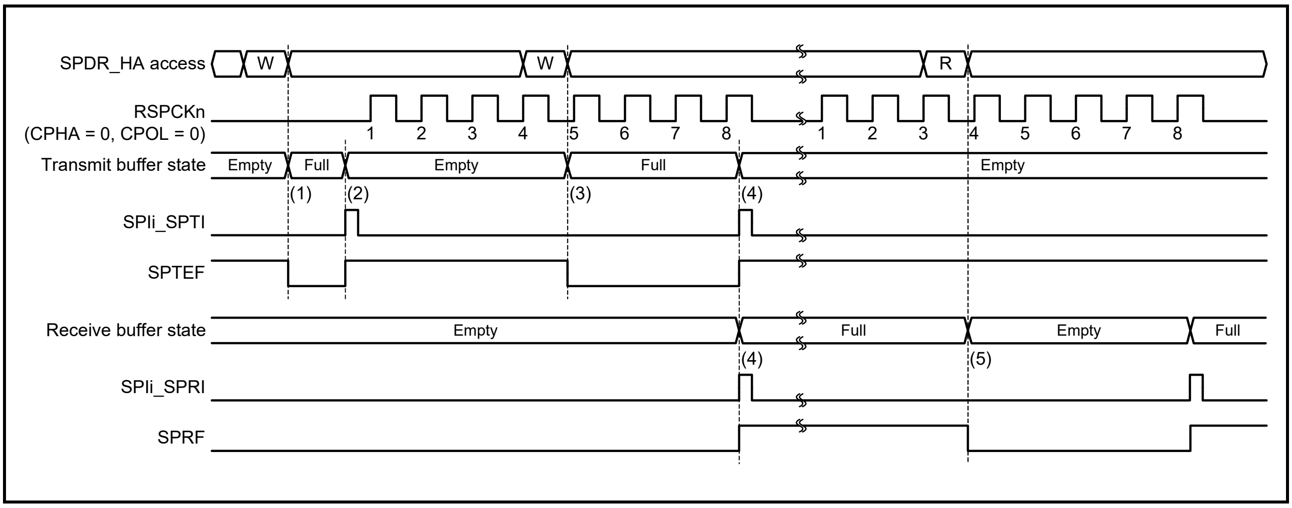


Figure 38.26 Operation example of the SPIi_SPTI and SPIi_SPRI interrupts when CPHA = 0 and CPOL = 0

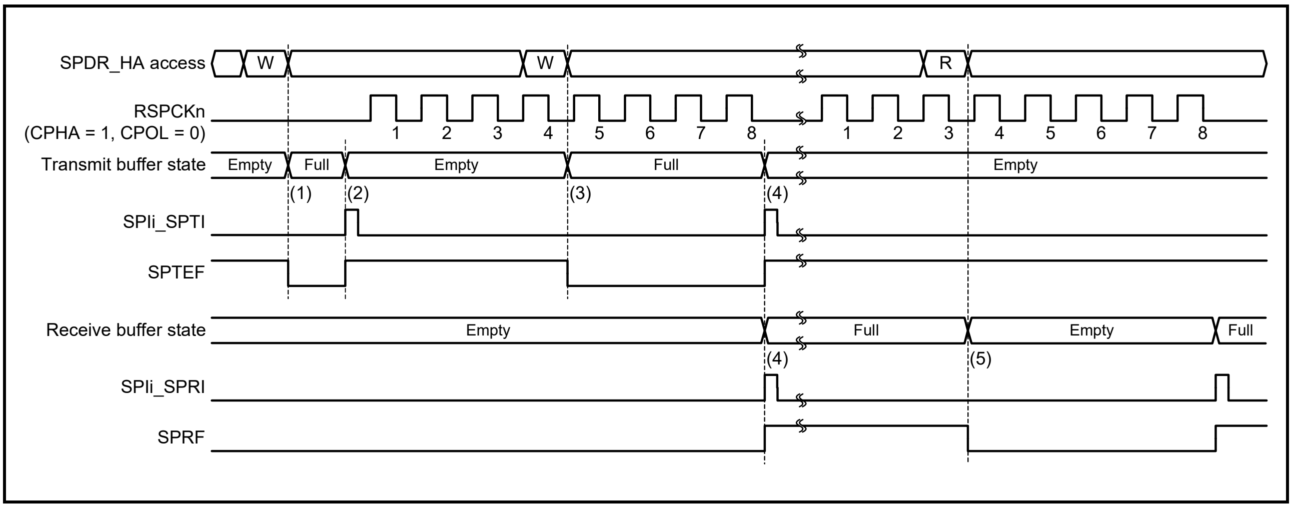


Figure 38.27 Operation example of the SPIi_SPTI and SPIi_SPRI interrupts when CPHA = 1 and CPOL = 0

The operation of the SPI at times (1) to (5) in the figure is as follows:

1. When transmit data is written to SPDR_HA and when the transmit buffer of SPDR_HA is empty (data for the next transfer is not set), the SPI writes data to the transmit buffer and clears the SPSR.SPTEF flag to 0.
2. If the shift register is empty, the SPI copies the data in the transmit buffer to the shift register, generates a transmit buffer empty interrupt request (SPIi_SPTI), and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the SPI. For details, see section 38.3.10, SPI Operation, and section 38.3.11, Clock Synchronous Operation.
3. When transmit data is written to SPDR_HA either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty state using the SPTEF flag, the SPI writes data to the transmit buffer and clears the SPTEF flag to 0. Because the data being transferred serially is stored in the shift register, the SPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends and the receive buffer of SPDR_HA is empty, the SPI copies the receive data in the shift register to the receive buffer, generates a receive buffer full interrupt request (SPIi_SPRI), and sets the SPRF flag to 1. Because the shift register becomes empty on completion of the serial transfer, when the transmit buffer is full before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.

5. When SPDR_HA is read either by the receive buffer full interrupt routine or processing of the receive buffer full state using the SPRF flag, the receive data can be read.

If SPDR_HA is written to when the transmit buffer holds data that is not yet transmitted (the SPTEF flag is 0), the SPI does not update data in the transmit buffer. When writing to SPDR_HA, always use either a transmit buffer empty interrupt request or processing of the transmit buffer empty interrupt using the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1. If the SPI function is disabled (the SPCR.SPE bit is 0), set the SPTIE bit to 0.

When serial transfer ends and the receive buffer is full (the SPRF flag is 1), the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see section 38.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an SPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the associated IELSRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers.

Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See section 14, Interrupt Controller Unit (ICU) for the interrupt vector numbers.

38.3.8 Error Detection

In normal SPI serial transfers, data written to the transmit buffer of SPDR/SPDR_HA is transmitted, and received data can be read from the receive buffer of SPDR/SPDR_HA. In some cases non-normal transfers can be executed when SPDR/SPDR_HA is accessed, depending on the status of the transmit or receive buffer or the status of the SPI at the beginning or end of serial transfer.

If a non-normal transfer operation occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 38.8 lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 38.8 Relationship between non-normal transfer operations and SPI error detection

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR/SPDR_HA is written when the transmit buffer is full.	• Keeps the contents of the transmit buffer • Missing write data.	None
2	SPDR/SPDR_HA is read when the receive buffer is empty.	Outputs the contents of the receive buffer and previously received data.	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data.	• Suspends serial transfer • Missing transmit and receive data • Stops driving of the MISOA output signal • Disables the SPI function.	Underrun error
4	Serial transfer terminates when the receive buffer is full.	• Keeps the contents of the receive buffer • Missing receive data.	Overrun error
5	An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled.	Asserts the parity error flag.	Parity error
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Stops driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals • Disables the SPI function.	Mode fault error
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	• Suspends serial transfer • Missing transmit and receive data • Stops driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals • Disables the SPI function.	Mode fault error
8	The SSLn0 input signal is negated during serial transfer in slave mode.	• Suspends serial transfer • Missing transmit and receive data • Stops driving of the MISOn output signal • Disables the SPI function.	Mode fault error

In operation 1 described in Table 38.8, the SPI does not detect an error. To prevent data omission during the writing to SPDR/SPDR_HA, write operations to SPDR/SPDR_HA must be executed using a transmit buffer empty interrupt request (when the SPSR.SPTEF flag is 1).

Similarly, the SPI does not detect an error in operation 2. To prevent extraneous data from being read, SPDR/SPDR_HA read operations must be executed with an SPI receive buffer full interrupt request (when the SPSR.SPRF flag is 1).

For the other errors in the figure, see the following sections:

- Underrun error (operation 3): section 38.3.8.4, Underrun errors
- Overrun error (operation 4): section 38.3.8.1, Overrun errors
- Parity error (operation 5): section 38.3.8.2, Parity errors
- Mode fault error (operations 6 to 8): section 38.3.8.3, Mode fault errors.

For the transmit and receive interrupts, see section 38.3.7, Transmit Buffer Empty and Receive Buffer Full Interrupts.

38.3.8.1 Overrun errors

If a serial transfer ends when the receive buffer of SPDR/SPDR_HA is full, the SPI detects an overrun error and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the SPI does not copy data from the shift register to the receive buffer, so the data prior to the error occurrence is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU reads SPSR with the OVRF flag set to 1.

Figure 38.28 shows an example of operation of the OVRF and SPRF flags. The SPSR and SPDR_HA accesses shown in the figure indicate the condition of access to the register, where W denotes a write cycle and R a read cycle. In this example, the SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, indicating the number of transferred bits.

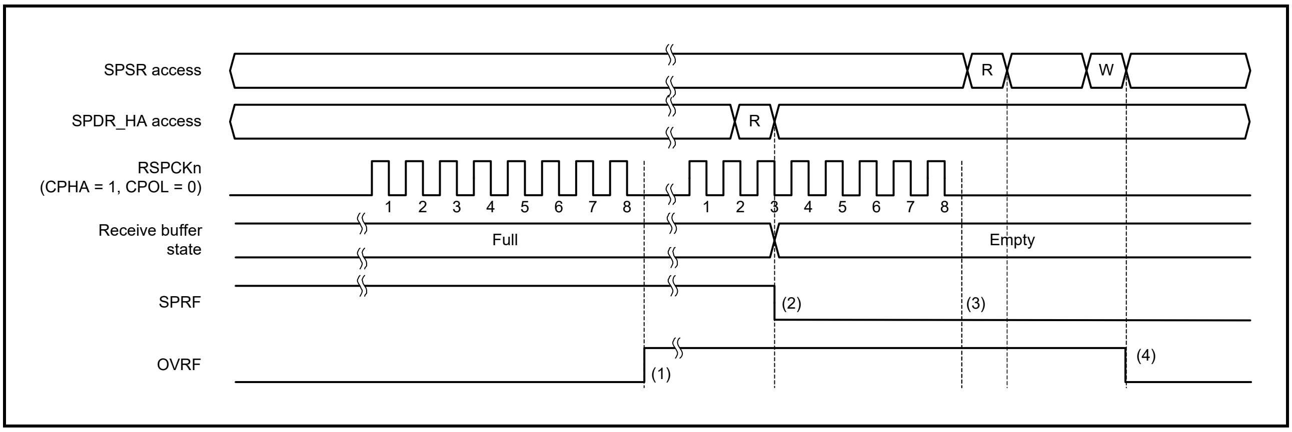


Figure 38.28 Operation example of the OVRF and SPRF flags

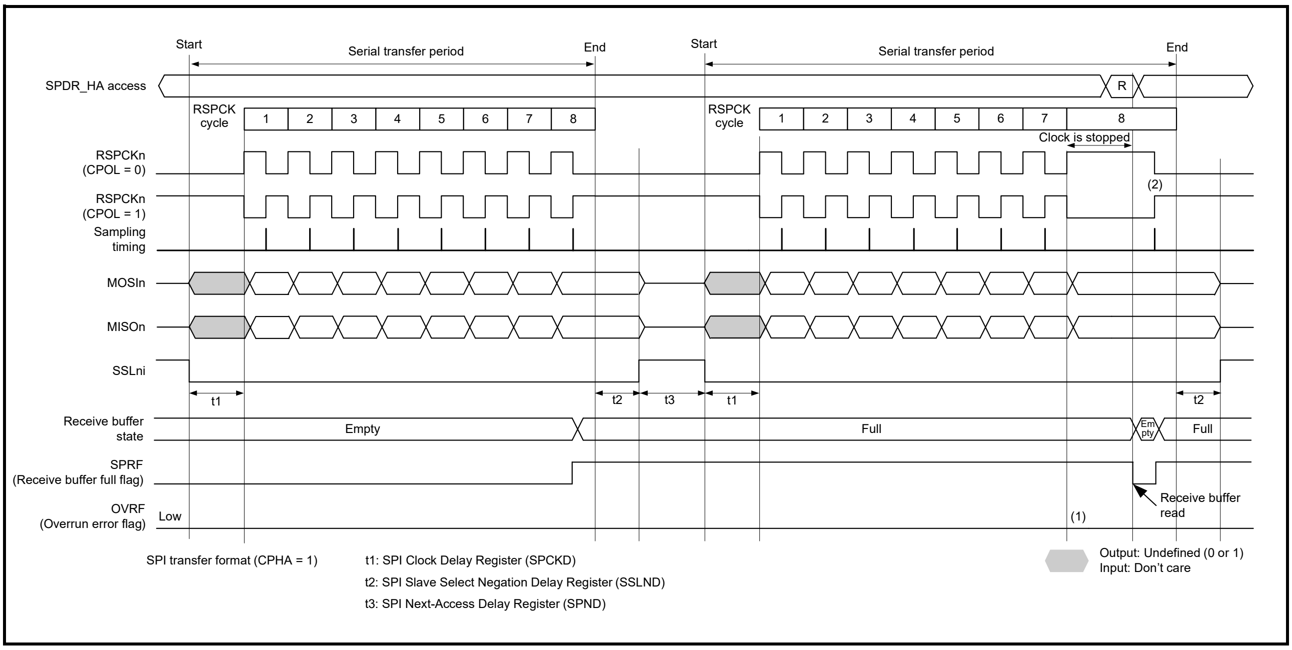
The operation of the flags at times (1) to (4) in the figure is as follows:

1. If a serial transfer terminates with the SPRF flag set to 1 (receive buffer full), the SPI detects an overrun error, and sets the OVRF flag to 1. The SPI does not copy the data in the shift register to the receive buffer. Even when the SPPE bit is 1, parity errors are not detected. In master mode, the SPI copies the value of the SPCMDm pointer to the SPSSR.SPECM[2:0] bits.
2. When SPDR_HA is read, the SPI outputs the data in the receive buffer. The SPRF flag is then set to 0. The receive buffer becoming empty does not set the OVRF flag to 0.
3. If the serial transfer ends with the OVRF flag set to 1 (overrun error occurred), the SPI does not copy data in the shift register to the receive buffer (the SPRF flag does not set to 1). A receive buffer full interrupt is not generated. Even when the SPPE bit is 1, parity errors are not detected. In master mode, the SPI does not update the SPSSR.SPECM[2:0] bits. In an overrun error state when the SPI does not copy the received data from the shift register to the receive buffer, on termination of the serial transfer, the SPI determines that the shift register is empty. This enables data transfer from the transmit buffer to the shift register.
4. If 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag clears to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, you must ensure that overrun errors are detected early, for example by reading SPSR immediately after SPDR_HA is read. In master mode, the value of the SPCMDm pointer at the error occurrence can be checked by reading the SPSSR.SPECM[2:0] bits.

If an overrun error occurs and the OVRF flag sets to 1, normal reception operations cannot be performed until the OVRF flag is cleared to 0.

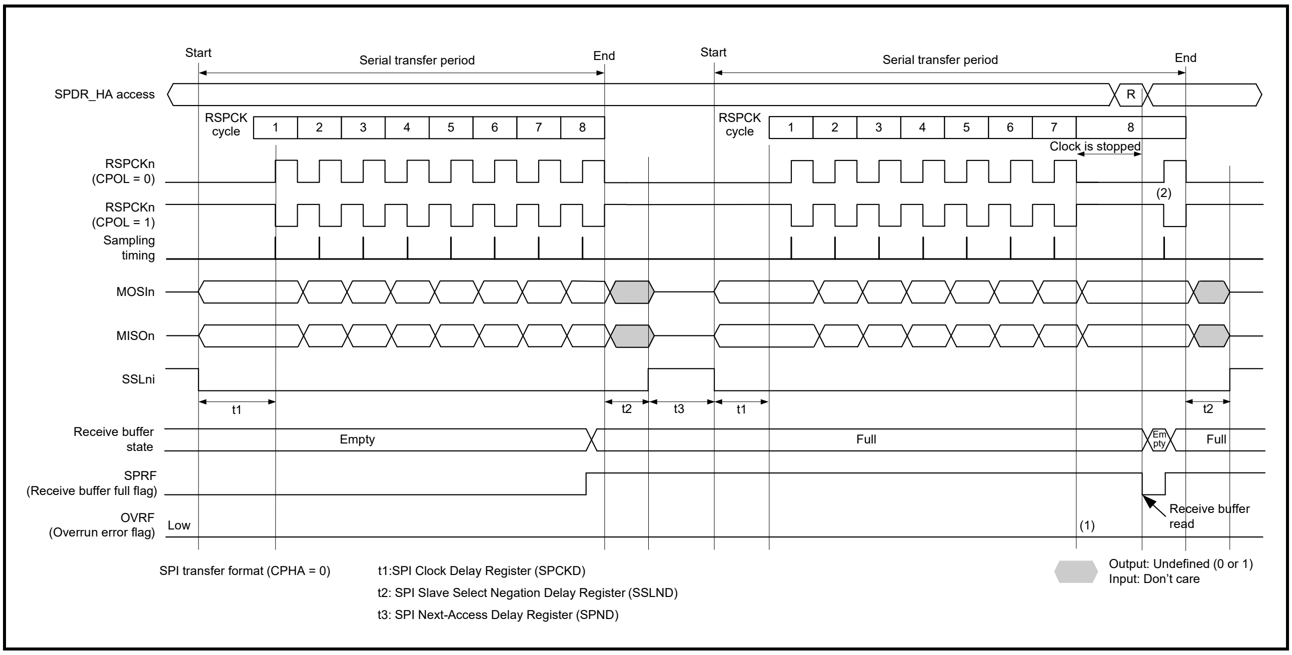
When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 38.29 and Figure 38.30 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.



SPI transfer format (CPHA = 1)

t1: SPI Clock Delay Register (SPCKD)
t2: SPI Slave Select Negation Delay Register (SSLND)
t3: SPI Next-Access Delay Register (SPND)

Figure 38.29 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 1)



SPI transfer format (CPHA = 0)

t1:SPI Clock Delay Register (SPCKD)
t2: SPI Slave Select Negation Delay Register (SSLND)
t3: SPI Next-Access Delay Register (SPND)

Figure 38.30 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 0)

The operation of the flags at times (1) and (2) in the figure is as follows:

1. When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.
2. If SPDR_HA is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPSR.SPRF flag clears to 0).

38.3.8.2 Parity errors

When full-duplex synchronous serial communication is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, the SPI checks for parity errors when serial transfer ends. On detecting a parity error in the received data, the SPI sets the SPSR.PERF flag to 1. Because the SPI does not copy data in the shift register to the receive buffer when the SPSR.OVRF flag sets to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after the SPSR register is read with the PERF flag set to 1.

Figure 38.31 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 38.31 indicates the condition of access to the register, where W denotes a write cycle and R a read cycle. In this example, full-duplex synchronous serial communication is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

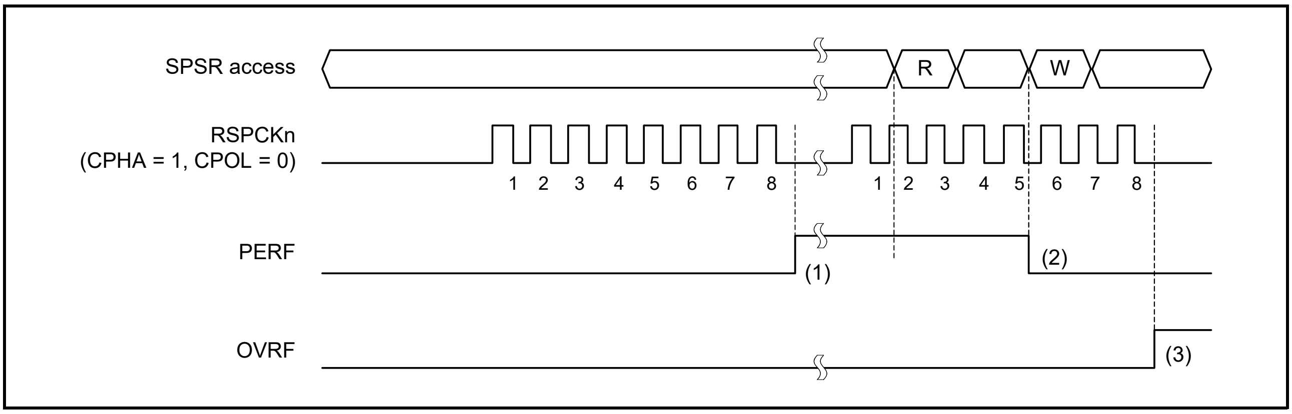


Figure 38.31 Operation example of the PERF flag

The operation of the flags at times (1) to (3) in the figure is as follows:

1. If a serial transfer terminates with the SPI not detecting an overrun error, the SPI copies the data in the shift register to the receive buffer. The SPI checks the received data at this timing and sets the PERF flag to 1 if a parity error is detected. In master mode, the SPI copies the value of the SPCMDm pointer to the SPSSR.SPECM[2:0] bits.
2. If 0 is written to the PERF flag after the SPSR register is read when the PERF flag is 1, the PERF flag clears to 0.
3. When the SPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

Parity errors can be checked for by either reading the SPSR register or using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, such checks are required to ensure early detection of parity errors. When the SPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

38.3.8.3 Mode fault errors

The SPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input for the SSLn0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of the status of the serial transfer, and sets the SPSR.MODF flag to 1. On detecting the mode fault error, the SPI copies the value of the SPCMDm pointer to the SPSSR.SPECM[2:0] bits. The active level of the SSLn0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode. The SPI detects a mode fault error if the MODFEN bit of the SPI in slave mode is 1, and the SPMS bit is 0, and if the SSLn0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

On detecting a mode fault error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see section 38.3.9, Initializing the SPI). For multi-master configuration, detection of a mode fault error is used to stop the driving of output signals and the SPI function, which allows the master to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. SPSR polling is required for detecting mode fault errors if the SPI error interrupt is not used. When using the SPI in master mode, the value of the SPCMDm pointer at the error occurrence can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of a mode fault error, the MODF flag must be set to 0.

38.3.8.4 Underrun errors

When the serial transfer begins while the SPCR.MSTR bit is 0 (slave mode), SPCR.SPE bit is 1 and the transmission data not prepared, the SPI detects an underrun error. Then, SPI sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see section 38.3.9, Initializing the SPI).

The occurrence of an underrun error can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. SPSR polling is required for detecting underrun errors if the SPI error interrupt is not used.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be cleared to 0.

38.3.9 Initializing the SPI

If 0 is written to the SPCR.SPE bit or the SPI sets the SPE bit to 0 because of the detection of a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

38.3.9.1 Initialization by clearing of the SPE bit

When the SPCR.SPE bit is set to 0, the SPI performs the following initialization:

- Suspends any serial transfer that is being executed
- Stops the driving of output signals (Hi-Z) in slave mode
- Initializes the internal state of the SPI
- Initializes the transmit buffer of the SPI (the SPSR.SPTEF flag sets to 1).

Initialization by clearing of the SPE bit does not initialize the control bits of the SPI. For this reason, the SPI can be started in the same transfer mode in use prior to initialization when the SPE bit is set to 1 again.

The SPSR.SPRF, SPSR.OVRF, SPSR.MODF, SPSR.PERF, and SPSR.UDRF flags are not initialized, and the value of the SPI Sequence Status Register (SPSSR) is not initialized. Therefore, even after the SPI is initialized, data from the receive buffer can be read to check the status of error occurrence during an SPI transfer.

The transmit buffer is initialized to an empty state (the SPSR.SPTEF flag sets to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit at the same time as writing 0 to the SPE bit.

38.3.9.2 System reset

An initialization by a system reset completely initializes the SPI by initializing all SPI control bits, status bits, and data registers, in addition to meeting the requirements described in section 38.3.9.1, Initialization by clearing of the SPE bit.

38.3.10 SPI Operation

38.3.10.1 Master mode operation

The only difference between single- and multi-master mode operation lies in mode fault error detection (see section

38.3.8, Error Detection). The SPI does not detect mode fault errors in single master mode and does in multi-master mode. This section explains operations that are the same for single- and multi-master modes.

(1) Starting a serial transfer

The SPI updates the data in the transmit buffer (SPTX) when data is written to the SPI Data Register (SPDR/SPDR_HA) and the SPI transmit buffer is empty (data for the next transfer is not set and the SPSR.SPTEF flag is 1). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR/SPDR_HA, the SPI copies data from the transmit buffer to the shift register and starts serial transfer. On copying transmit data to the shift register, the SPI changes the status of the shift register to full. On termination of the serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

The polarity of the SSLni output pins depends on the SSLP register settings. For details on the SPI transfer format, see section 38.3.5, Transfer Formats.

(2) Terminating a serial transfer

Regardless of the SPCMDm.CPHA bit setting, the SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPSR.SPRF flag is 0), on termination of the serial transfer, the SPI copies data from the shift register to the receive buffer of the SPDR/SPDR_HA register.

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLni output pin depends on the SSLP register settings. For details on the SPI transfer format, see section 38.3.5, Transfer Formats.

(3) Sequence control

The transfer format used in master mode is determined by the SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

The SPSCR register determines the sequence configuration for serial transfers that are executed by the SPI in master mode. The following Parameters are set in the SPCMDm register:

- SSLni pin output signal value
- MSB- or LSB-first
- Data length
- Some of the bit rate settings
- RSPCK polarity and phase
- Whether SPCKD is to be referenced
- Whether SSLND is to be referenced
- Whether SPND is to be referenced.

SPBR holds some of the bit rate settings, including SPCKD (SPI clock delay), SSLND (SSL negation delay), and SPND (next-access delay).

Based on the sequence length assigned in SPSCR, the SPI makes up a sequence comprised of a part or all of the SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command in the sequence, the SPI sets the pointer to SPCMD0, and in this way the sequence is executed repeatedly.

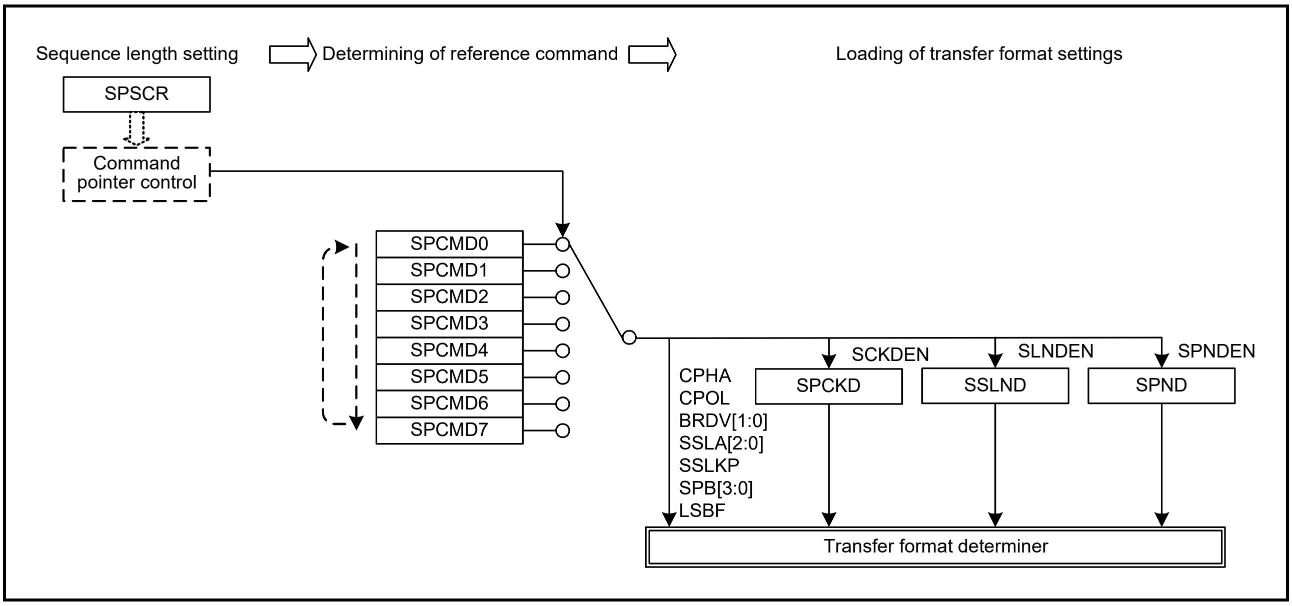


Figure 38.32 Procedure for determining the form of a serial transfer in master mode

In this section, a frame is the combination of the data in SPDR/SPDR_HA and the settings in SPCMDm.

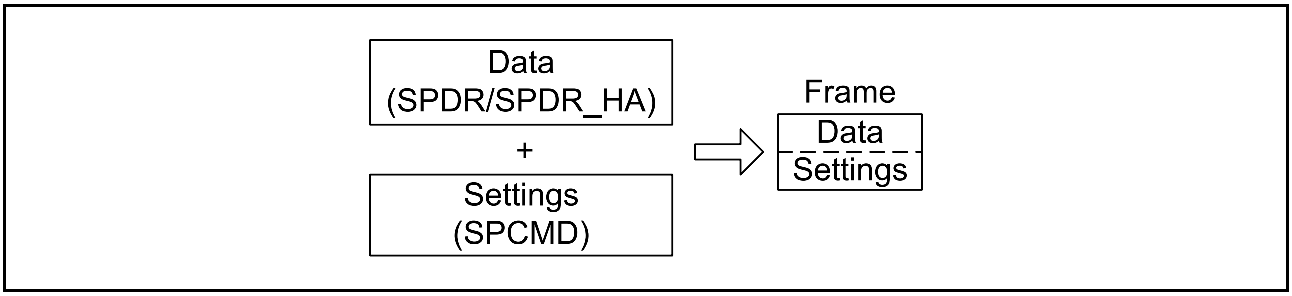
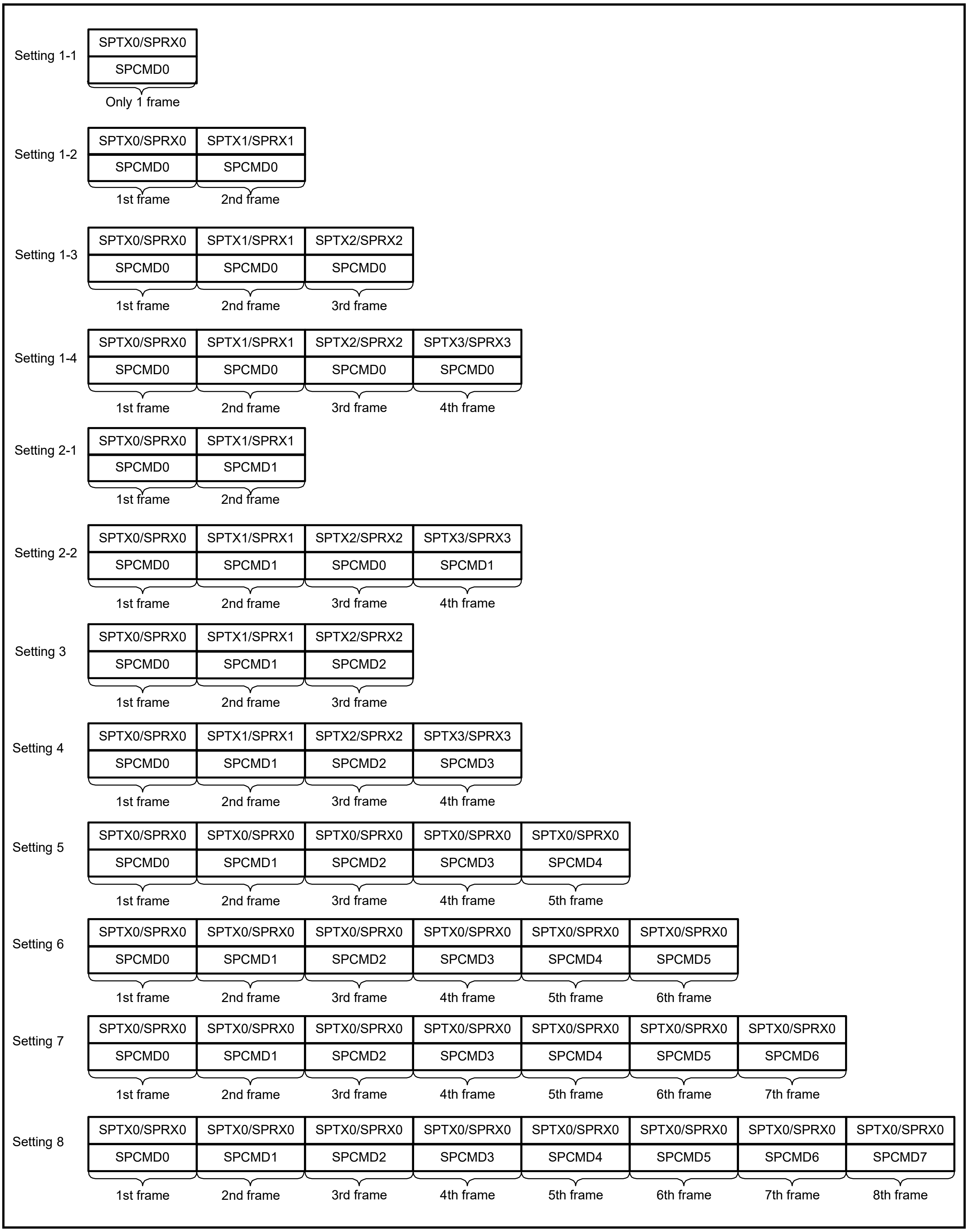


Figure 38.33 Conceptual diagram of frames

Figure 38.34 shows the correspondence between the commands and the transmit and receive buffers in the sequence of operations specified by the settings in Table 38.4.



	1st frame	2nd frame	3rd frame	4th frame	5th frame	6th frame	7th frame	8th frame
Setting 1-1	SPTX0/SPRX0 SPCMD0 Only 1 frame							
Setting 1-2	SPTX0/SPRX0 SPCMD0	SPTX1/SPRX1 SPCMD0						
Setting 1-3	SPTX0/SPRX0 SPCMD0	SPTX1/SPRX1 SPCMD0	SPTX2/SPRX2 SPCMD0					
Setting 1-4	SPTX0/SPRX0 SPCMD0	SPTX1/SPRX1 SPCMD0	SPTX2/SPRX2 SPCMD0	SPTX3/SPRX3 SPCMD0				
Setting 2-1	SPTX0/SPRX0 SPCMD0	SPTX1/SPRX1 SPCMD1						
Setting 2-2	SPTX0/SPRX0 SPCMD0	SPTX1/SPRX1 SPCMD1	SPTX2/SPRX2 SPCMD0	SPTX3/SPRX3 SPCMD1				
Setting 3	SPTX0/SPRX0 SPCMD0	SPTX1/SPRX1 SPCMD1	SPTX2/SPRX2 SPCMD2					
Setting 4	SPTX0/SPRX0 SPCMD0	SPTX1/SPRX1 SPCMD1	SPTX2/SPRX2 SPCMD2	SPTX3/SPRX3 SPCMD3				
Setting 5	SPTX0/SPRX0 SPCMD0	SPTX0/SPRX0 SPCMD1	SPTX0/SPRX0 SPCMD2	SPTX0/SPRX0 SPCMD3	SPTX0/SPRX0 SPCMD4			
Setting 6	SPTX0/SPRX0 SPCMD0	SPTX0/SPRX0 SPCMD1	SPTX0/SPRX0 SPCMD2	SPTX0/SPRX0 SPCMD3	SPTX0/SPRX0 SPCMD4	SPTX0/SPRX0 SPCMD5		
Setting 7	SPTX0/SPRX0 SPCMD0	SPTX0/SPRX0 SPCMD1	SPTX0/SPRX0 SPCMD2	SPTX0/SPRX0 SPCMD3	SPTX0/SPRX0 SPCMD4	SPTX0/SPRX0 SPCMD5	SPTX0/SPRX0 SPCMD6	
Setting 8	SPTX0/SPRX0 SPCMD0	SPTX0/SPRX0 SPCMD1	SPTX0/SPRX0 SPCMD2	SPTX0/SPRX0 SPCMD3	SPTX0/SPRX0 SPCMD4	SPTX0/SPRX0 SPCMD5	SPTX0/SPRX0 SPCMD6	SPTX0/SPRX0 SPCMD7

Figure 38.34 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Burst transfers

If the SPCMDm.SSLKP bit that the SPI references during the current serial transfer is 1, the SPI maintains the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the SPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

Figure 38.35 shows an example of an SSLni signal operation for a burst transfer that is implemented using the SPCMD0 and SPCMD1 register settings. This section describes SPI operations (1) to (7) shown in Figure 38.35.

Note: The polarity of the SSLni output signal depends on the SSLP register settings.

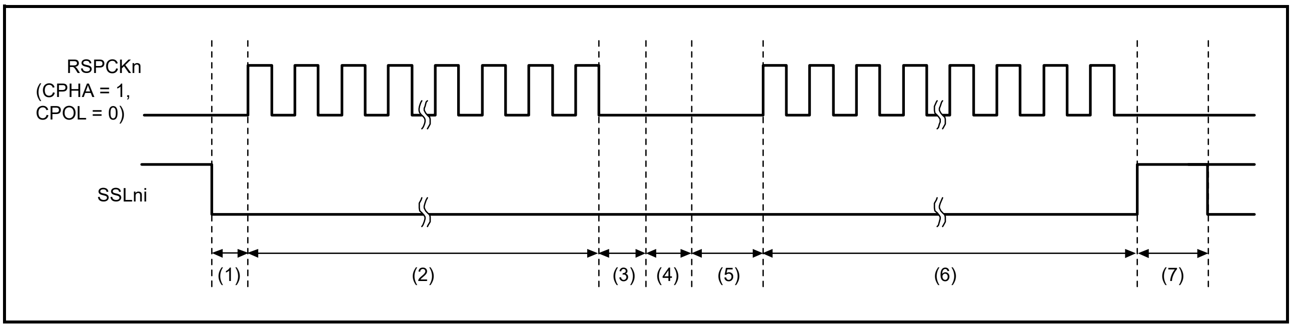


Figure 38.35 Example of burst transfer operation using the SSLKP bit

The SPI operation at times (1) to (7) in the figure is as follows:

1. Based on the SPCMD0 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
2. The SPI executes serial transfers in accordance with the SPCMD0 settings.
3. The SPI inserts an SSL negation delay.
4. Because the SPCMD0.SSLKP bit is 1, the SPI keeps the SSLni signal value specified in SPCMD0. This period is sustained at a minimum for a period equal to the next-access delay in SPCMD0. If the shift register is empty after the passage of the minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
5. Based on the SPCMD1 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
6. The SPI executes serial transfers in accordance with the SPCMD1 settings.
7. Because the SPCMD1.SSLKP bit is 0, the SPI negates the SSLni signal. In addition, a next-access delay is inserted in accordance with SPCMD1.

If the SSLni signal output settings in the SPCMDm register where 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the SPI switches the SSLni signal status to SSLni signal assertion as shown in (5) in Figure 38.35. This corresponds to the command for the next transfer.

Note: If such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels might occur.

The SPI in master mode references the SSLni signal operation within the module when the SSLKP bit is not used. When the SPCMDm.CPHA bit is 0, the SPI can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally.

(5) RSPCK delay (t1)

The RSPCK delay value of the SPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The SPI determines the SPCMDm register to be referenced during a serial transfer by pointer control, and determines the RSPCK delay value by using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 38.9. For a definition of the RSPCK delay, see section 38.3.5, Transfer Formats.

Table 38.9 Relationship among the SCKDEN bit, SPCKD register, and RSPCK delay (1 of 2)

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay
0	000b to 111b	1 RSPCK

Table 38.9 Relationship among the SCKDEN bit, SPCKD register, and RSPCK delay (2 of 2)

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The SPI determines the SPCMDm register to be referenced by pointer control during a serial transfer, and determines the SSL negation delay by using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 38.10. For a definition of the SSL negation delay, see section 38.3.5, Transfer Formats.

Table 38.10 Relationship among the SLNDEN bit, SSLND, and SSL negation delay

SPCMDm.SLNDEN bit	SSLND.SLNDL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-access delay (t3)

The next-access delay value of the SPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND register setting. The SPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines the next-access delay during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 38.11. For a definition of the next-access delay, see section 38.3.5, Transfer Formats.

Table 38.11 Relationship among the SPNDEN bit, SPND, and next-access delay

SPCMDm.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 2 PCLKA
1	000b	1 RSPCK + 2 PCLKA
	001b	2 RSPCK + 2 PCLKA
	010b	3 RSPCK + 2 PCLKA
	011b	4 RSPCK + 2 PCLKA
	100b	5 RSPCK + 2 PCLKA
	101b	6 RSPCK + 2 PCLKA
	110b	7 RSPCK + 2 PCLKA
	111b	8 RSPCK + 2 PCLKA

(8) Initialization flow

Figure 38.36 shows an example of initialization flow for SPI operation when the SPI is used in master mode. For a description of how to set up the Interrupt Controller Unit (ICU), DMAC, and I/O ports, see the descriptions given in the individual blocks.

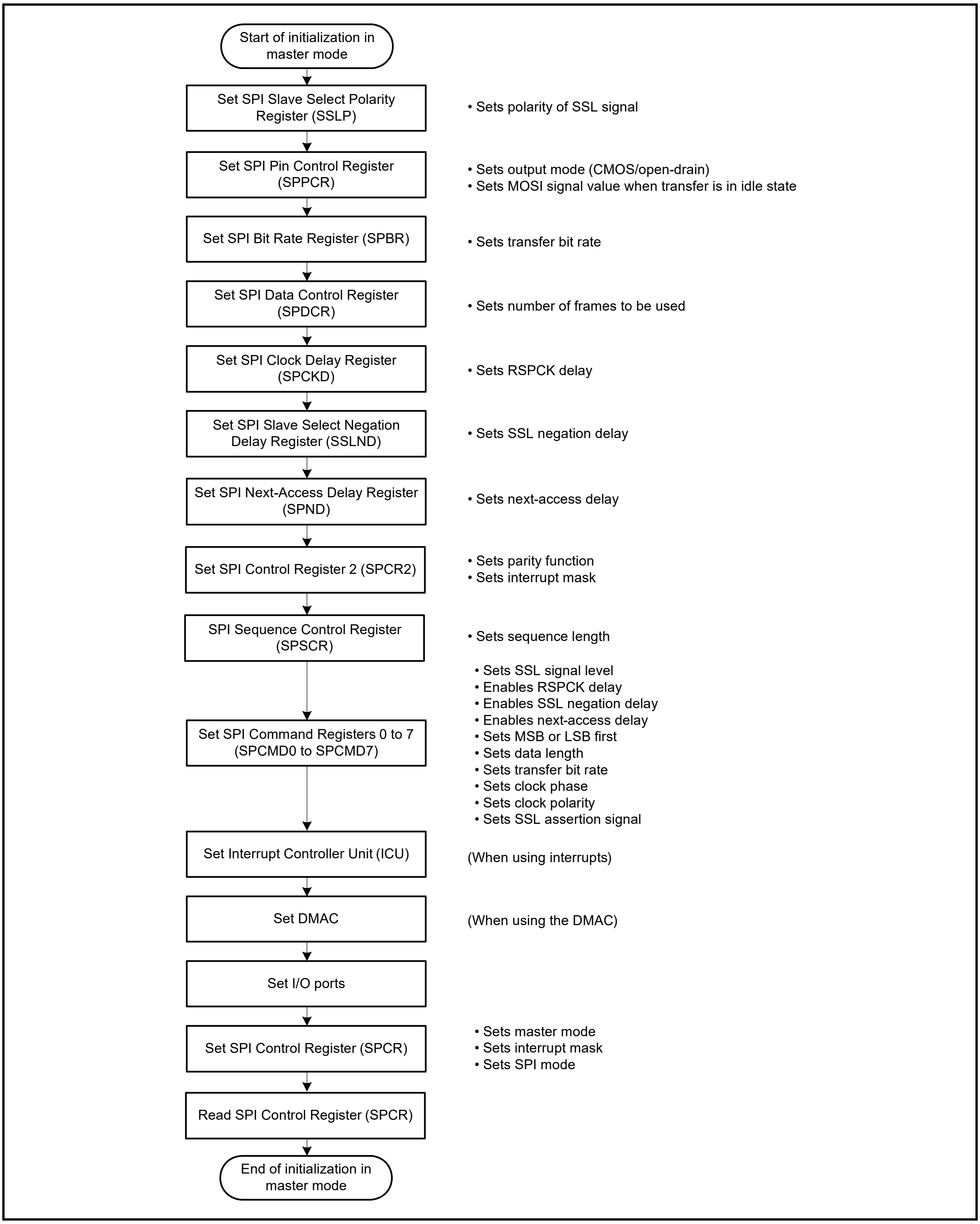


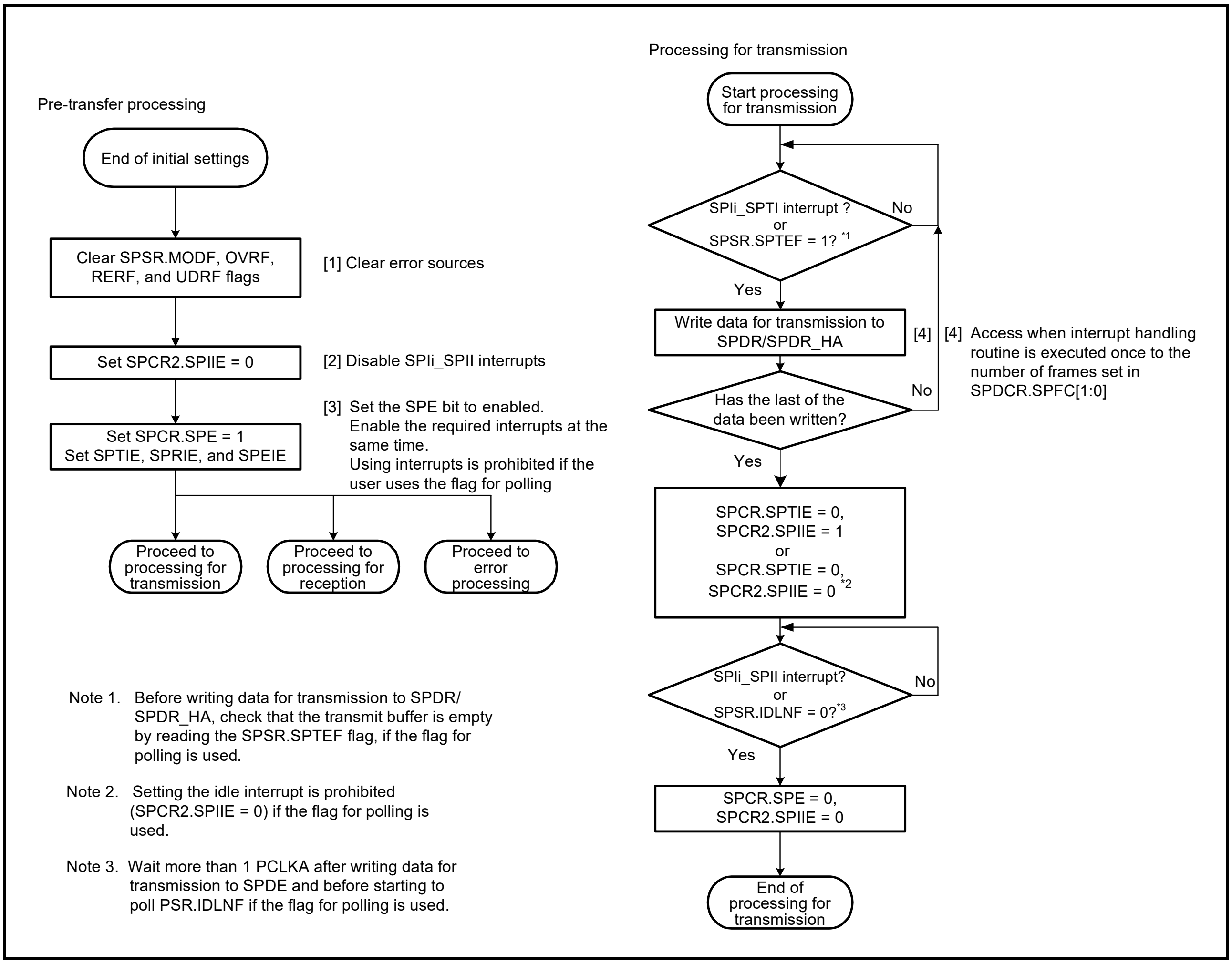
Figure 38.36 Example of initialization flow in master mode for SPI operation

(9) Software processing flow

Figure 38.37 to Figure 38.39 show examples of the software processing flow.

(a) Transmit processing flow

When transmitting data and when the SPIi_SPII interrupt is enabled, the CPU is notified of the completion of data transmission after the last writing of data for transmission.



Note 1. Before writing data for transmission to SPDR/SPDR_HA, check that the transmit buffer is empty by reading the SPSR.SPTEF flag, if the flag for polling is used.

Note 2. Setting the idle interrupt is prohibited (SPCR2.SPIIE = 0) if the flag for polling is used.

Note 3. Wait more than 1 PCLKA after writing data for transmission to SPDE and before starting to poll PSR.IDLNF if the flag for polling is used.

Figure 38.37 Transmission flow in master mode transmission

(b) Receive processing flow

The SPI does not handle receive-only operation, so processing for transmission is required.

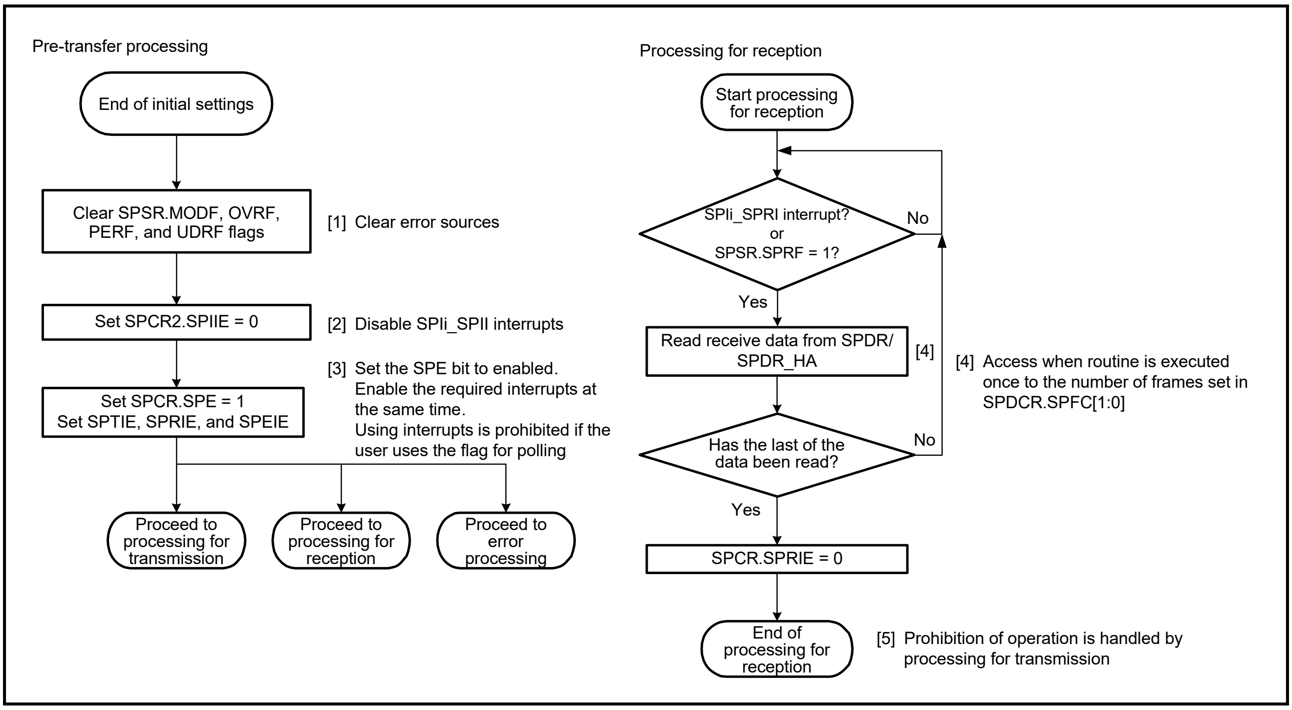


Figure 38.38 Reception flow in master mode

(c) Error processing flow

The SPI detects mode fault errors, underrun errors, overrun errors, and parity errors. When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, the SPCR.SPE bit is not cleared and operations for transmission and reception continue. Therefore, Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode fault errors. Not doing so leads to updating of the SPSSR.SPECM[2:0] bits.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

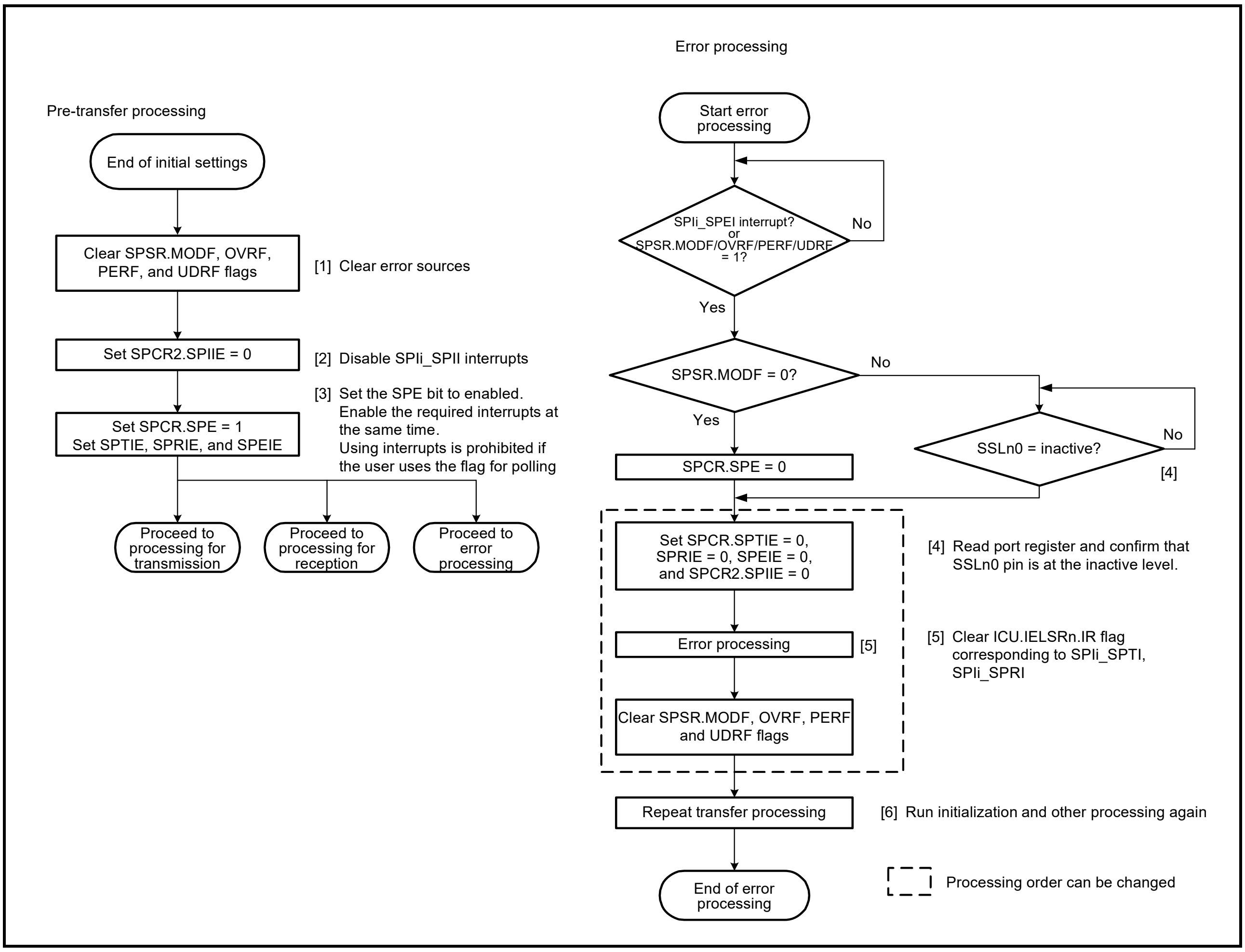


Figure 38.39 Error processing flow for master mode

38.3.10.2 Slave mode operation

(1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISOn output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

For details on the SPI transfer format, see section 38.3.5, Transfer Formats. The polarity of the SSLn0 input signal depends on the SSLP.SSL0P setting.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 38.3.8, Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLn0 input signal is determined by the SSLP.SSL0P bit setting. For details on the SPI transfer format, see section 38.3.5, Transfer Formats.

(3) Notes on single slave operations

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the type of configuration shown in Figure 38.7, for example, if the SPI is used in single slave mode, the SSLn0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. To correctly execute transmit and receive operations by the SPI in slave mode when the SSLn0 input signal is fixed at the active state, the CPHA bit must be set to 1. Do not fix the SSLn0 input signal if there is a requirement for setting the CPHA bit to 0.

(4) Burst transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. When the CPHA bit is 1, the serial transfer period is the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state. Even when the SSLn0 input signal remains at the active level, the SPI can accommodate burst transfers, because it can detect the start of an access.

When the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization flow

Figure 38.40 shows an example of initialization flow for SPI operation when the SPI is used in slave mode. For a description of how to set up the ICU, DMAC, and I/O ports, see the descriptions given in the individual blocks.

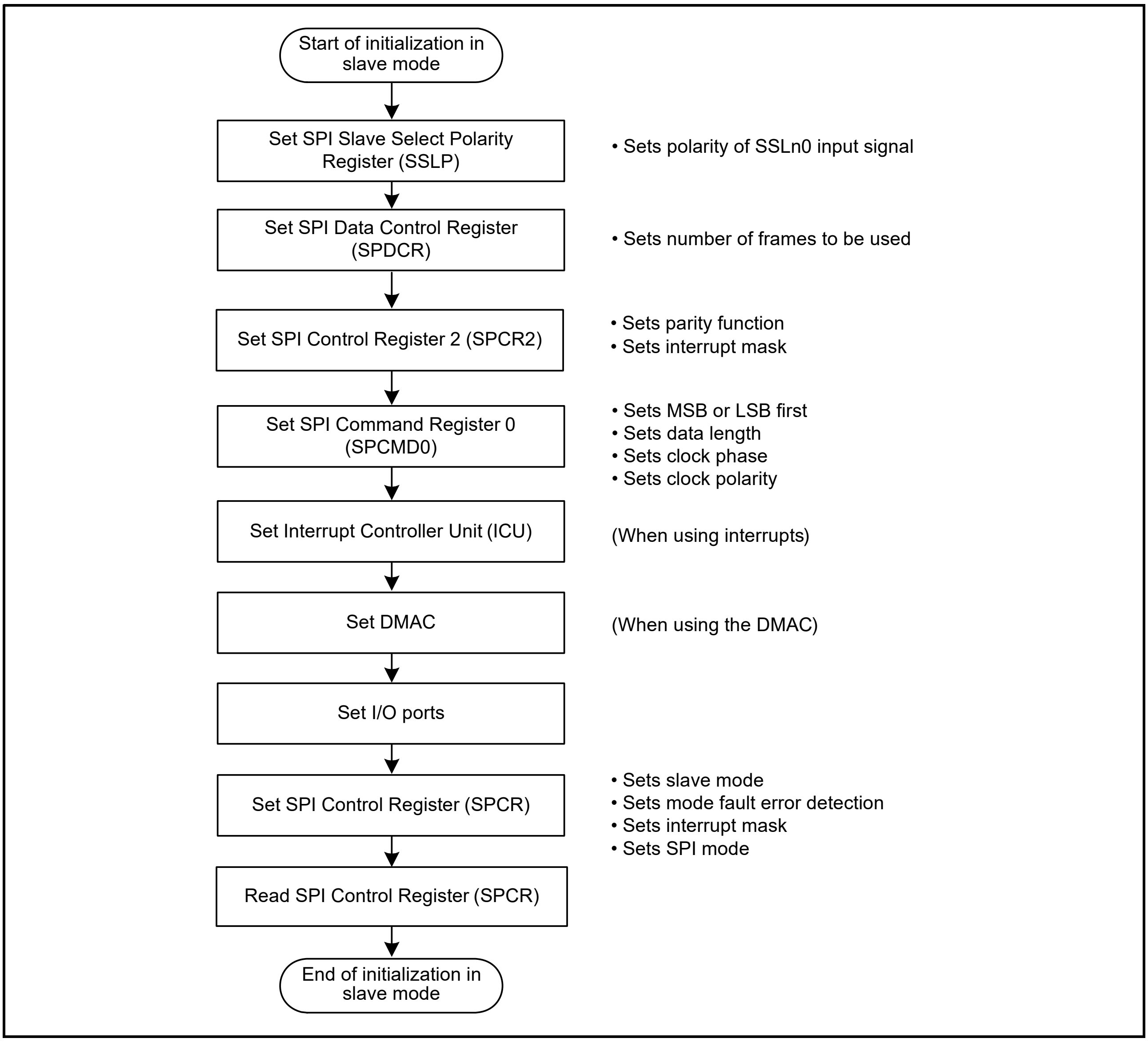


Figure 38.40 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Figure 38.41 to Figure 38.43 show examples of the flow of software processing.

(a) Transmit processing flow

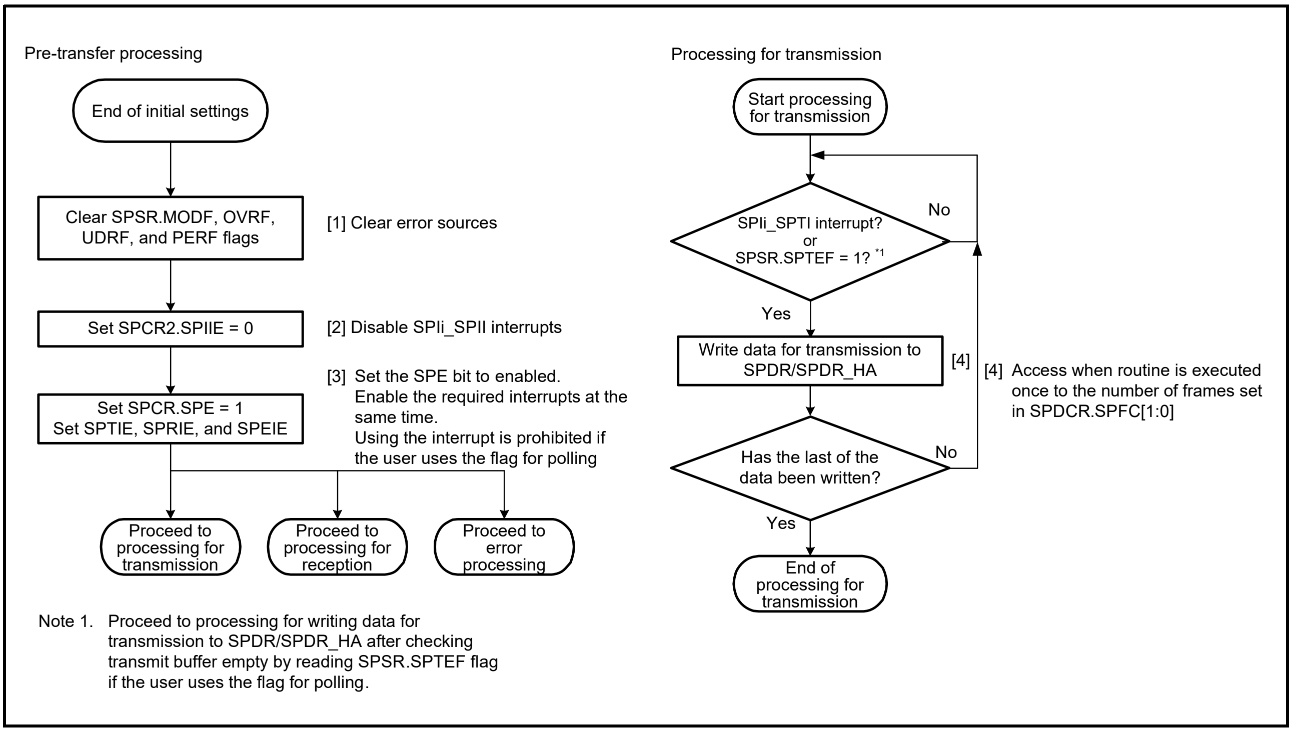


Figure 38.41 Transmission flow in slave mode

(b) Receive processing flow

The SPI does not handle receive-only operation, so processing for transmission is required.

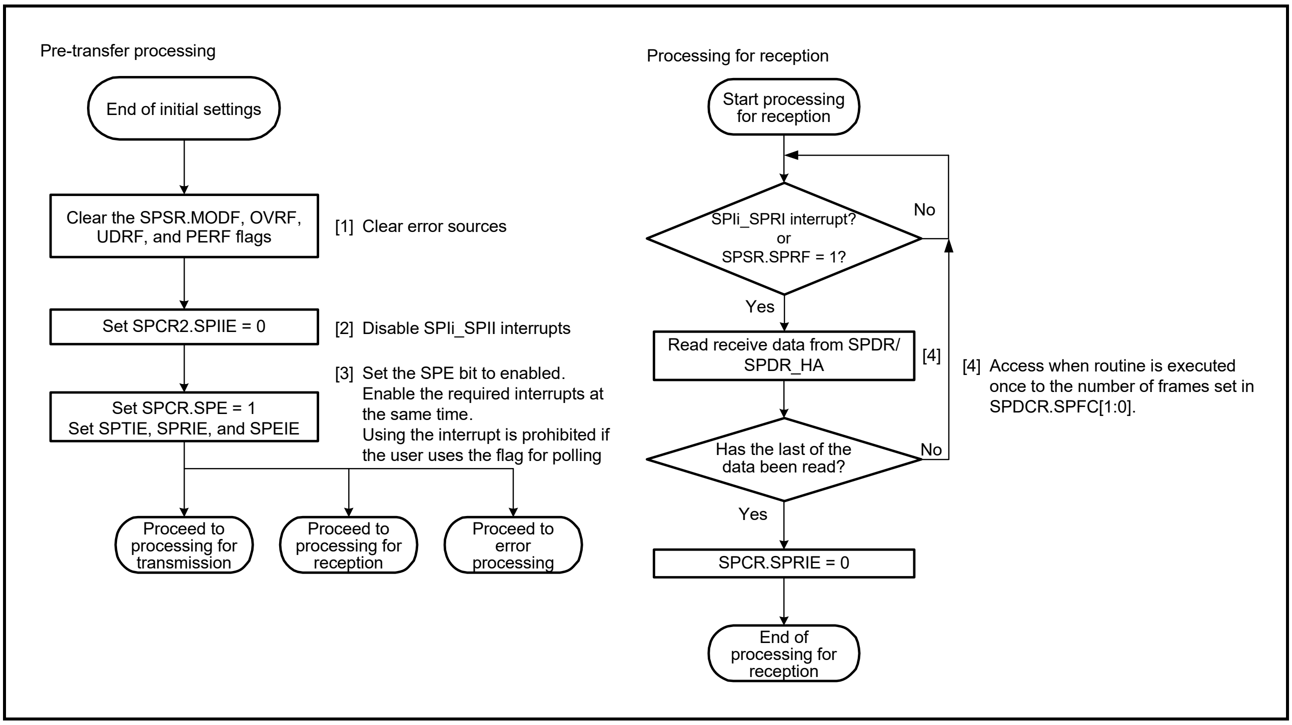


Figure 38.42 Reception flow in slave mode

(c) Error processing flow

In slave operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the state of the SSLn0 pin.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

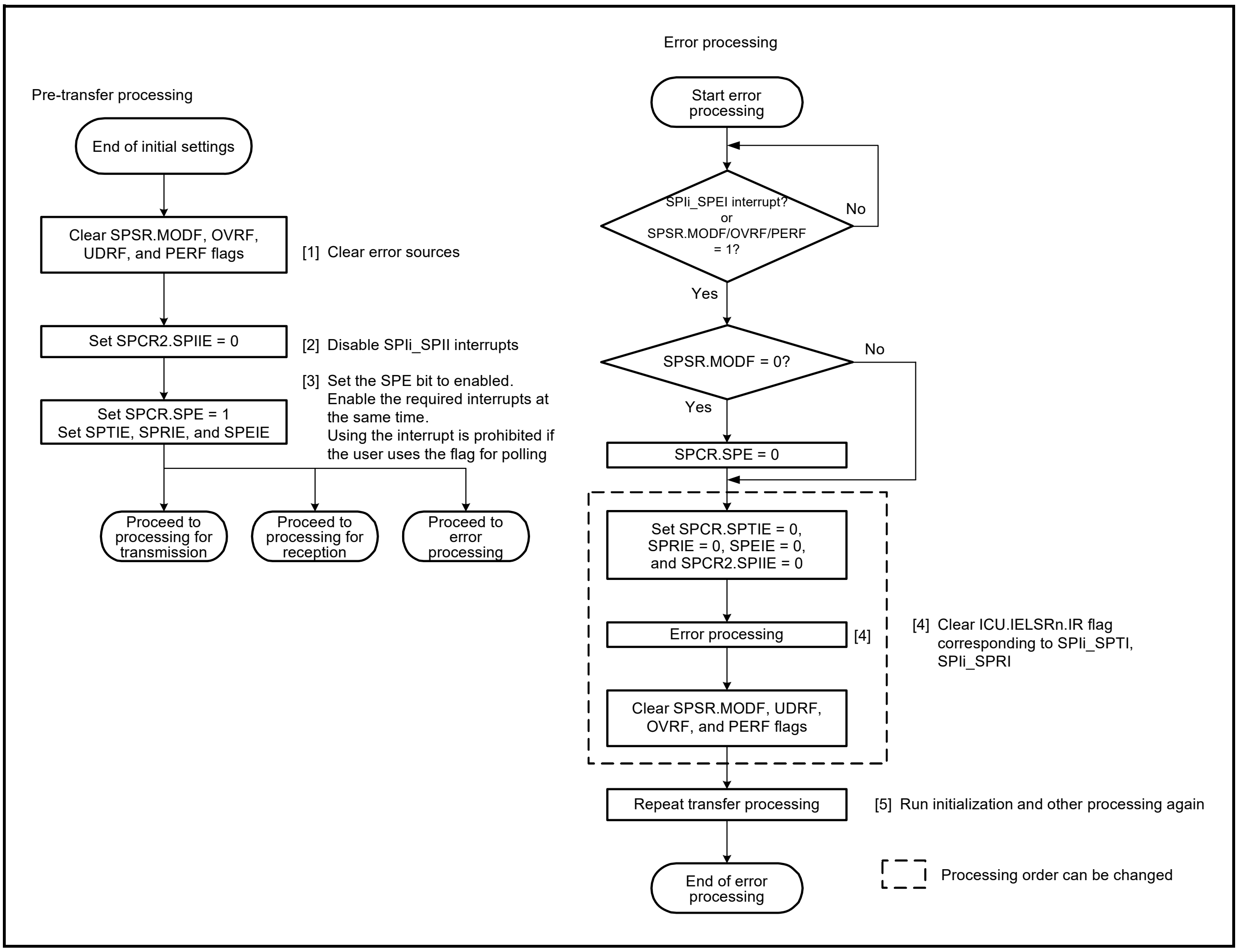


Figure 38.43 Error processing flow for slave mode

38.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSLni pin is not used, and the RSPCKn, MOSIn, and MISOn pins handle communications. All SSLni pins are available as I/O port pins.

Although clock synchronous operation does not require the use of the SSLni pin, operation of the module is the same as in SPI operation. In both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected, because the SSLni pin is not used.

Additionally, do not perform operation if clock synchronous operation enabled when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

38.3.11.1 Master mode operation

(1) Starting serial transfer

The SPI updates the data in the transmit buffer (SPTX) of SPDR/SPDR_HA when data is written to the SPDR/SPDR_HA register and the transmit buffer is empty (data for the next transfer is not set and the SPSR.SPTEF flag is 1). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR/SPDR_HA, the SPI copies data from the transmit buffer to the shift register and starts serial transmission. On copying transmit data to the shift register, the SPI changes the status of the shift register to full, and on termination of serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 38.3.5, Transfer Formats.

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR/SPDR_HA).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[3:0] bit setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 38.3.5, Transfer Formats.

(3) Sequence control

The transfer format used in master mode is determined by the SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPSCR register determines the sequence configuration for serial transfers that are executed by the SPI in master mode. The following parameters are specified in the SPCMDm register:

- SSLni output signal value
- MSB or LSB first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPCKD is to be referenced
- Whether SSLND is to be referenced
- Whether SPND is to be referenced.

SPBR holds some of the bit rate settings such as SPCKD, an SPI clock delay value, SSLND, an SSL negation delay, and SPND, a next-access delay value.

Based on the sequence length that is assigned to SPSCR, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command comprising the sequence, the SPI sets the pointer to the SPCMD0 register, and in this manner the sequence is executed repeatedly.

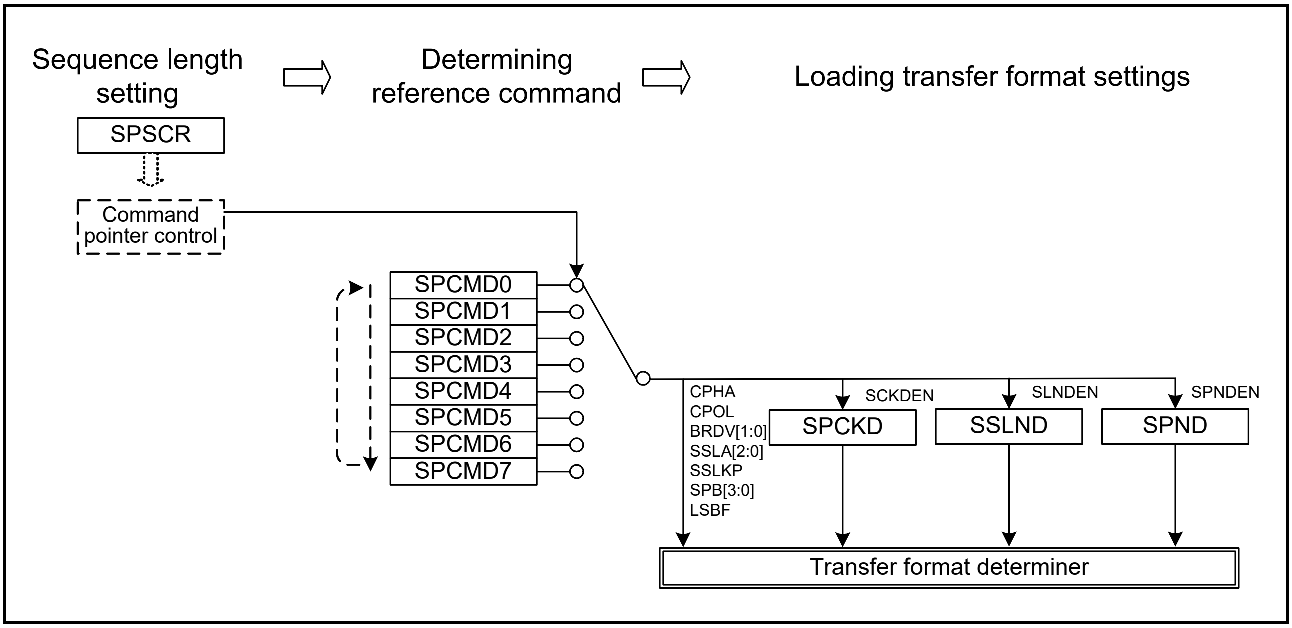


Figure 38.44 Procedure for determining the form of serial transmission in master mode

In this section, a frame is the combination of the data (SPDR/SPDR_HA) and the settings (SPCMDm).

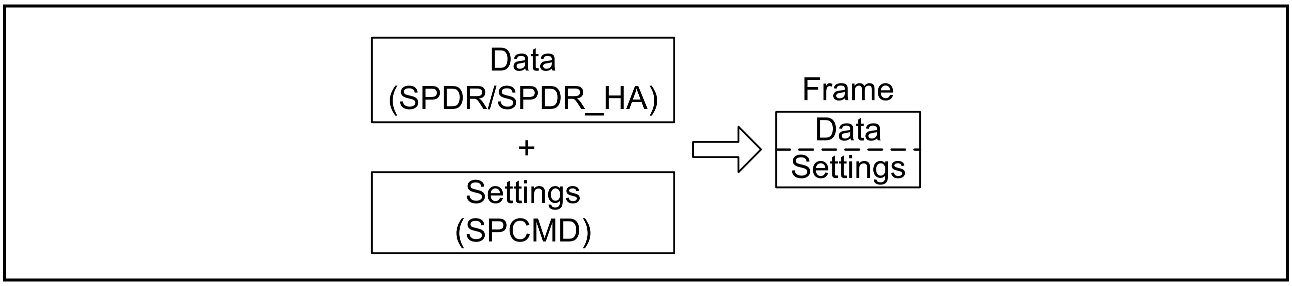


Figure 38.45 Conceptual diagram of frames

Figure 38.46 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 38.4.

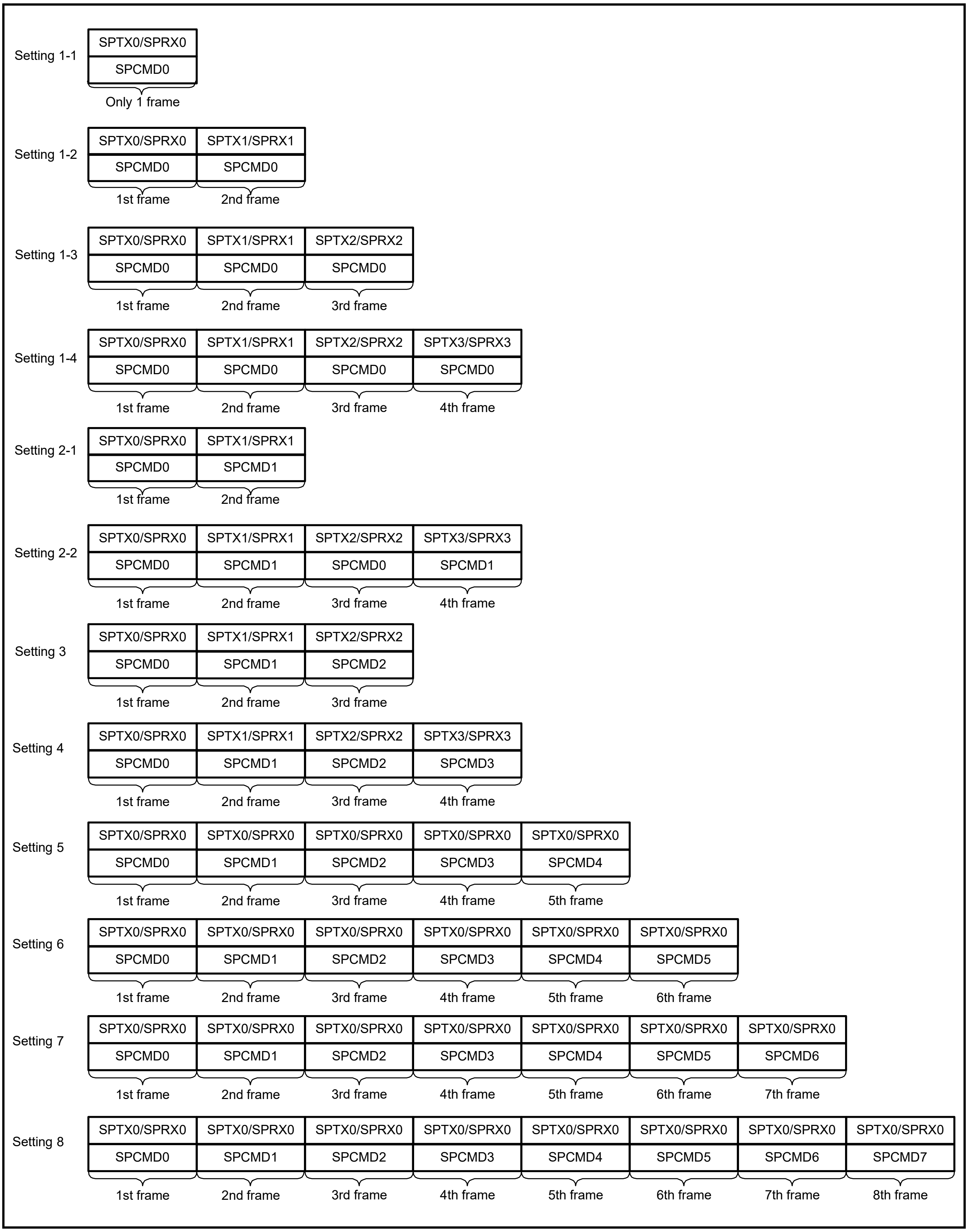


Figure 38.46 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations

(4) Initialization flow

Figure 38.47 shows an example of initialization flow for clock synchronous operation when the SPI is used in master mode. For a description of how to set up the ICU, DMAC, and I/O ports, see the descriptions given in the individual blocks.

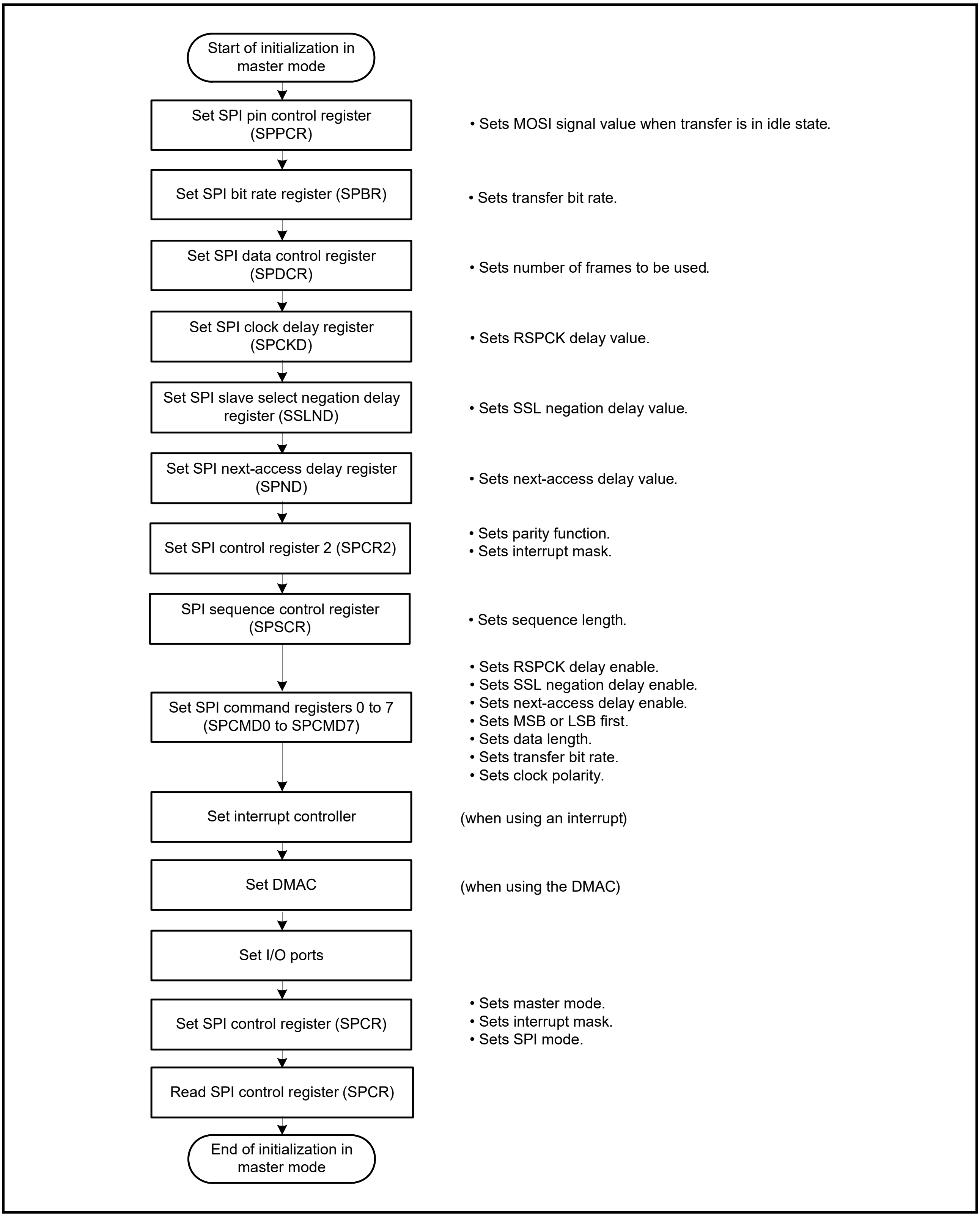


Figure 38.47 Example of initialization flow in master mode for clock synchronous operation

(5) Software processing flow

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see section 38.3.10.1, (9) Software processing flow. Mode fault errors do not occur in clock synchronous operation.

38.3.11.2 Slave mode operation

(1) Starting serial transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the SPI, and the SPI drives the MISOn output signal. The SSLn0 input signal is not used in clock synchronous operation. For details on the SPI transfer format, see section 38.3.5, Transfer Formats.

(2) Terminating serial transfer

The SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer the SPI changes the status of the shift register to empty regardless of the receive buffer.

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[3:0] bit setting. For details on the SPI transfer format, see section 38.3.5, Transfer Formats.

(3) Initialization flow

Figure 38.48 shows an example of initialization flow for clock synchronous operation when the SPI is used in slave mode. For a description of how to set up the ICU, DMAC, and I/O ports, see the descriptions given in the individual blocks.

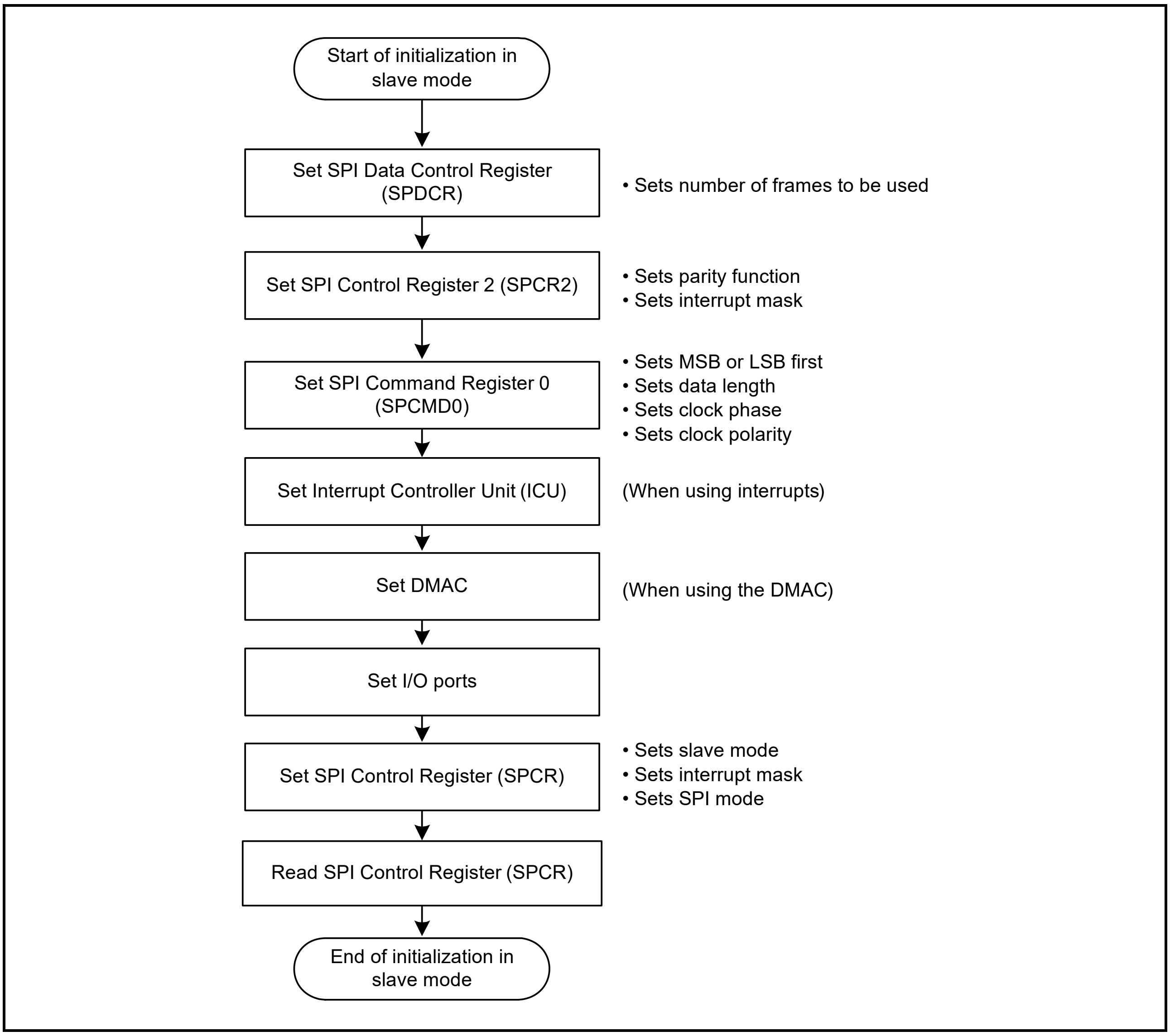


Figure 38.48 Example of initialization flow in slave mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see section 38.3.10.2, (6) Software processing flow. Mode fault errors do not occur in clock synchronous mode.

38.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, or between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input and output paths of the shift register. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, or between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the SPI or the reversed transmit data becomes the received data for the SPI.

Table 38.12 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 38.49 shows the configuration of the shift register I/O paths when the SPI in master mode is set to loopback mode (SPPCR.SPLP2 = 1, SPPCR.SPLP = 0 or 1).

Table 38.12 SPLP2 and SPLP bit settings and received data

SPPCR.SPLP2 bit	SPPCR.SPLP bit	Received data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

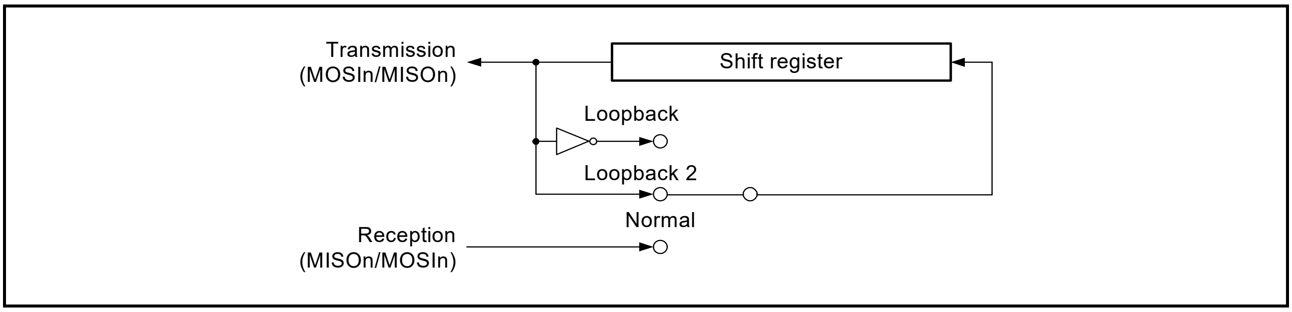


Figure 38.49 Configuration of Shift register I/O paths in loopback mode for master mode

38.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flow shown in Figure 38.50.

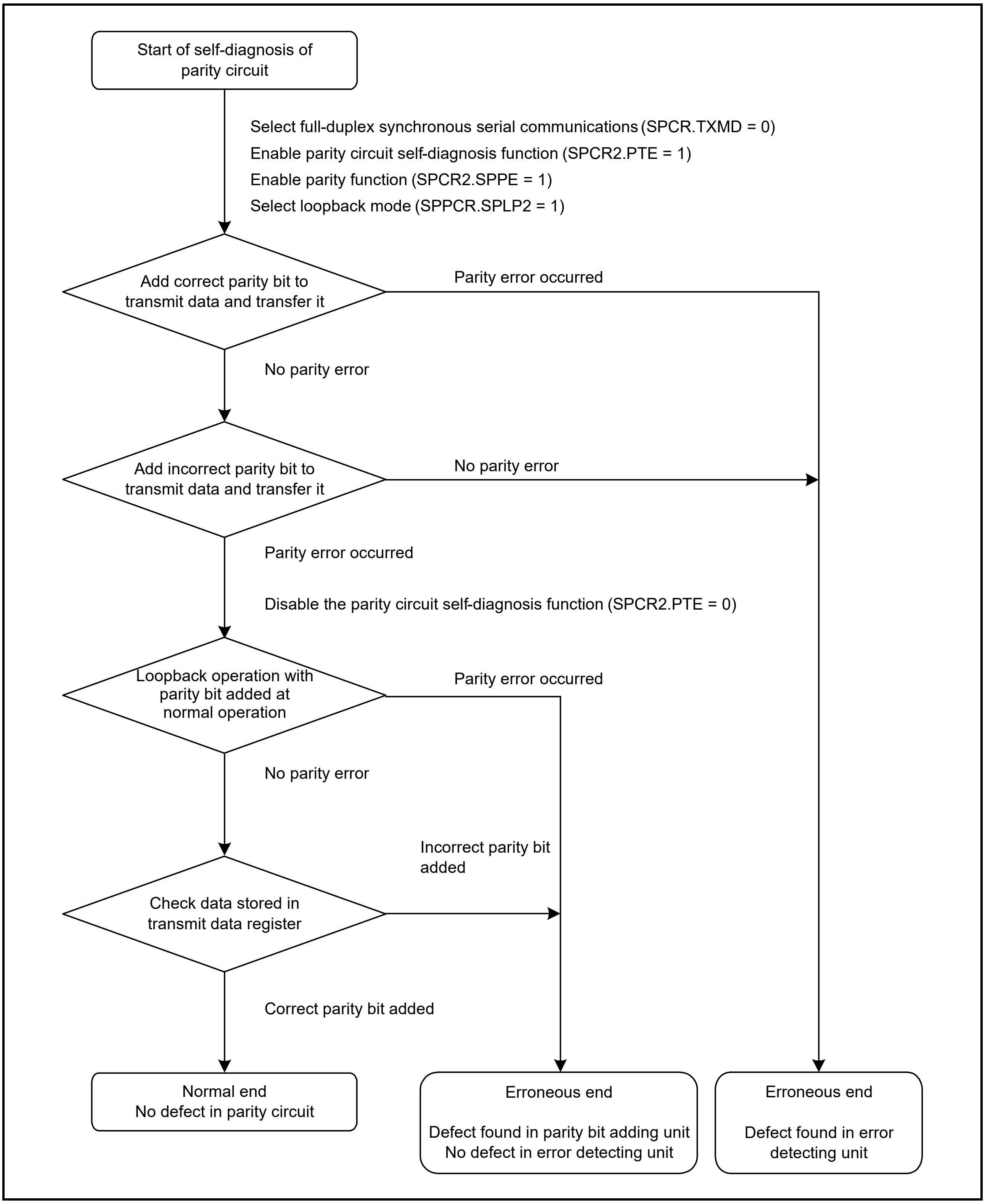


Figure 38.50 Self-diagnosis flow for parity circuit

38.3.14 Interrupt Sources

The SPI interrupt sources include:

- Receive buffer full
- Transmit buffer empty

- SPI error (mode fault, underrun, overrun, or parity error)
- SPI idle
- Transmission-complete

The DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for the SPIi_SPEI (SPI error interrupt) is allocated to interrupt requests on mode fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the SPI are listed in Table 38.13. An interrupt is generated on satisfaction of one of the interrupt conditions in Table 38.13. Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DTC or DMAC to perform data transmission and reception, the DTC or DMAC must be set up first to a transfer-enabled status before making the SPI settings. For information on setting up the DTC or DMAC, see section 17, DMA Controller (DMAC), or section 18, Data Transfer Controller (DTC).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IELSRn.IR flag clears to 0. A retained interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 38.13 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DMAC or DTC activation
Receive buffer full	SPIi_SPRI	Receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1	Possible
Transmit buffer empty	SPIi_SPTI	Transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI error (mode fault, underrun, overrun, or parity error)	SPIi_SPEI	SPSR.MODF, OVRF, PERF, or UDRF flag sets to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIi_SPII	SPSR.IDLNF flag clears to 0 while the SPCR2.SPIIE bit is 1	Impossible
Transmission-complete	SPIi_SPTEND	• Master mode: Interrupt is generated when the IDLNF flag (SPI idle flag) changes from 1 to 0 • Slave mode: interrupt occurs on conditions shown in Table 38.15	Impossible

38.4 Output to the Event Link Controller (ELC)

The ELC can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode-fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output.

The event link output signal is output regardless of the interrupt enable bit setting.

38.4.1 Receive Buffer Full Event Output

This event signal is output when received data is transferred from the shift register to the SPDR/SPDR_HA on completion of serial transfer.

38.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission is transferred from the transmit buffer to the shift register and when the value of the SPE bit changes from 0 to 1.

38.4.3 Mode-Fault, Underrun, Overrun, or Parity Error Event Output

This event signal is output when mode fault, underrun, overrun, or parity error is detected. See section 38.5.4, Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output if using this event signal.

(1) Mode-fault

Table 38.14 lists the conditions for occurrence of a mode-fault event.

Table 38.14 Conditions for mode fault occurrence

SPI mode	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR bit = 0)	1	Not active	Event is output only when the pin is deactivated during transmission

(2) Underrun

This event signal is output in response to an underrun when a serial transfer starts while the transmission data is not ready, and the value of the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags set to 1.

(3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the receive buffer contains unread data and the value of the SPCR.TXMD bit is 0. Under these conditions, the OVRF flag sets to 1.

(4) Parity error

This event signal is output in response to a parity error detected on completion of a serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

38.4.4 SPI Idle Event Output

(1) In master mode

In master mode, an event is output when the condition for setting the IDLNF flag (SPI idle flag) to 0 is satisfied.

(2) In slave mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (SPI is initialized).

38.4.5 Transmission-Completed Event Output

During both SPI operation and clock synchronous operation in master mode, an event is output when the IDLNF flag (SPI idle flag) changes from 1 to 0. Table 38.15 lists the conditions for occurrence of a transmission-completed event.

Table 38.15 Conditions for generation of transmission-complete event in slave mode

SPI mode	Transmit buffer state	Shift register state	Other
SPI operation (SPMS = 0)	Empty	Empty	Negation of SSLn0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Edge detection of the last RSPCKn

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode-fault error or the underrun error.

38.5 Usage Notes

38.5.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SPI operation. The SPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

38.5.2 Constraint on Low Power Functions

When using the module-stop function and entering a low power mode other than Sleep, set the SPCR.SPE bit to 0 before completing communication.

38.5.3 Constraints on Starting Transfer

If the ICU.IELSRn.IR flag is 1 when transfer starts, the interrupt request is internally saved, which can lead to unanticipated behavior of the ICU.IELSRn.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (the SPCR.SPE bit is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

38.5.4 Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output

Using the mode-fault, underrun, overrun or parity error event is prohibited if the SPI is in multi-master mode (when the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

38.5.5 Constraints on the SPRF and SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, interrupt usage is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupts or the flags can be used, but not both.

39. Quad Serial Peripheral Interface (QSPI)

39.1 Overview

The Quad Serial Peripheral Interface (QSPI) module is a memory controller for connecting serial ROM that has an SPI-compatible interface. This includes nonvolatile memory, such as a serial flash memory, serial EEPROM, or serial FeRAM. Table 39.1 lists the QSPI specifications, Figure 39.1 shows a block diagram, and Table 39.2 lists the I/O pins.

Table 39.1 QSPI specifications

Parameter	Specifications
Number of channels	1 channel
SPI	• Support for Extended SPI, Dual SPI, and Quad SPI protocols • Configurable to SPI mode 0 and SPI mode 3 • Address width selectable to 8, 16, 24, or 32 bits.
Timing adjustment function	Configurable to support a wide range of serial flash
Flash read function	• Support for Read, Fast Read, Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, and Fast Read Quad I/O instructions • Substitutable instruction code • Adjustable number of dummy cycles • Prefetch function • Polling processing • SPI bus cycle extension function.
Direct communication function	Flexible support for a wide variety of serial flash instructions and functions through software control, including erase, write, ID read, and power-down control
Interrupt source	Error interrupts
Module-stop function	Module-stop state can be set to reduce power consumption

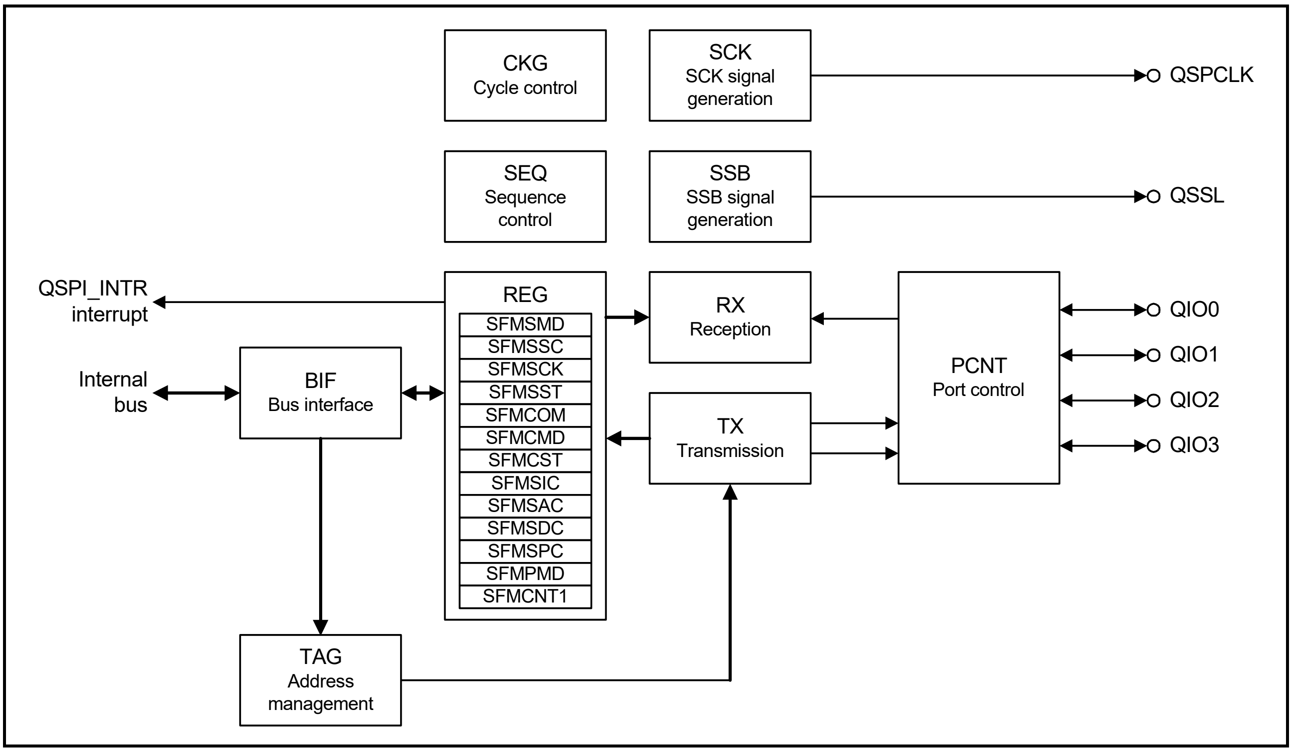


Figure 39.1 QSPI block diagram

Table 39.2 QSPI I/O pins

Pin name	I/O	Function
QSPCLK	Output	QSPI clock output pin
QSSL	Output	QSPI slave select pin
QIO0	I/O	Data 0 input/output
QIO1	I/O	Data 1 input/output
QIO2	I/O	Data 2 input/output
QIO3	I/O	Data 3 input/output

39.2 Register Descriptions

39.2.1 Transfer Mode Control Register (SFMSMD)

Address(es): QSPI.SFMSMD 6400 0000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SFMCCE	—	—	—	SFMOSW	SFMOHW	SFMOEX	SFMMD3	SFMPAE	SFMPFE	SFMSE[1:0]		—	SFMRM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b2 to b0	SFMRM[2:0]	Serial interface read mode select	b2 b0 0 0 0: Standard Read 0 0 1: Fast Read 0 1 0: Fast Read Dual Output 0 1 1: Fast Read Dual I/O 1 0 0: Fast Read Quad Output 1 0 1: Fast Read Quad I/O 1 1 0: Setting prohibited (unpredictable operation can result) 1 1 1: Setting prohibited (unpredictable operation can result).	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	SFMSE[1:0]	QSSL extension function select after SPI bus access	b5 b4 0 0: Do not extend QSSL 0 1: Extend QSSL by 33 QSPCLK 1 0: Extend QSSL by 129 QSPCLK 1 1: Extend QSSL infinitely.	R/W
b6	SFMPFE	Prefetch function select	0: Disable function 1: Enable function.	R/W
b7	SFMPAE	Function select for stopping prefetch at locations other than on byte boundaries	0: Disable function 1: Enable function.	R/W
b8	SFMMD3	SPI mode select. Initial value determined by input to CFGMD3	0: SPI mode 0 1: SPI mode 3.	R/W
b9	SFMOEX	Extension select for the I/O buffer output enable signal for the serial interface	0: Do not extend 1: Extend by 1 QSPCLK.	R/W

Bits	Symbol	Bit name	Description	R/W
b10	SFMOHW	Hold time adjustment for serial transmission	0: Do not extend high-level width of QSPCLK during transmission 1: Extend high-level width of QSPCLK by 1 PCLKA during transmission.	R/W
b11	SFMOSW	Setup time adjustment for serial transmission	0: Do not extend low-level width of QSPCLK during transmission 1: Extend low-level width of QSPCLK by 1 PCLKA during transmission.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SFMCCE	Read instruction code select	0: Set default instruction code for each instruction 1: Write instruction code in the SFMSIC register.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.2 Chip Selection Control Register (SFMSSC)

Address(es): QSPI.SFMSSC 6400 0004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	SFMSLD	SFMSHD	SFMSW			
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1

Bits	Symbol	Bit name	Description	R/W
b3 to b0	SFMSW	Minimum high-level width select for QSSL signal	b3 b0 0 0 0 0: 1 QSPCLK 0 0 0 1: 2 QSPCLK 0 0 1 0: 3 QSPCLK 0 0 1 1: 4 QSPCLK 0 1 0 0: 5 QSPCLK 0 1 0 1: 6 QSPCLK 0 1 1 0: 7 QSPCLK 0 1 1 1: 8 QSPCLK 1 0 0 0: 9 QSPCLK 1 0 0 1: 10 QSPCLK 1 0 1 0: 11 QSPCLK 1 0 1 1: 12 QSPCLK 1 1 0 0: 13 QSPCLK 1 1 0 1: 14 QSPCLK 1 1 1 0: 15 QSPCLK 1 1 1 1: 16 QSPCLK.	R/W
b4	SFMSHD	QSSL signal release timing select	0: Release QSSL 0.5 QSPCLK cycles after the last rising edge of QSPCLK 1: Release QSSL 1.5 QSPCLK cycles after the last rising edge of QSPCLK.	R/W
b5	SFMSLD	QSSL signal output timing select	0: Output QSSL 0.5 QSPCLK cycles before the first rising edge of QSPCLK 1: Output QSSL 1.5 QSPCLK cycles before the first rising edge of QSPCLK.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.3 Clock Control Register (SFMSKC)

Address(es): QSPI.SFMSKC 6400 0008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	SFMDTY	SFMDV				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	Symbol	Bit name	Description	R/W
b4 to b0	SFMDV	Serial interface reference cycle select. (Pay attention to irregularities.)	b4 b0 0 0 0 0 0: 2 PCLKA 0 0 0 0 1: 3 PCLKA (multiplied by an odd number)*1 0 0 0 1 0: 4 PCLKA 0 0 0 1 1: 5 PCLKA (multiplied by an odd number)*1 0 0 1 0 0: 6 PCLKA 0 0 1 0 1: 7 PCLKA (multiplied by an odd number)*1 0 0 1 1 0: 8 PCLKA 0 0 1 1 1: 9 PCLKA (multiplied by an odd number)*1 0 1 0 0 0: 10 PCLKA 0 1 0 0 1: 11 PCLKA (multiplied by an odd number)*1 0 1 0 1 0: 12 PCLKA 0 1 0 1 1: 13 PCLKA (multiplied by an odd number)*1 0 1 1 0 0: 14 PCLKA 0 1 1 0 1: 15 PCLKA (multiplied by an odd number)*1 0 1 1 1 0: 16 PCLKA 0 1 1 1 1: 17 PCLKA (multiplied by an odd number)*1 1 0 0 0 0: 18 PCLKA 1 0 0 0 1: 20 PCLKA 1 0 0 1 0: 22 PCLKA 1 0 0 1 1: 24 PCLKA 1 0 1 0 0: 26 PCLKA 1 0 1 0 1: 28 PCLKA 1 0 1 1 0: 30 PCLKA 1 0 1 1 1: 32 PCLKA 1 1 0 0 0: 34 PCLKA 1 1 0 0 1: 36 PCLKA 1 1 0 1 0: 38 PCLKA 1 1 0 1 1: 40 PCLKA 1 1 1 0 0: 42 PCLKA 1 1 1 0 1: 44 PCLKA 1 1 1 1 0: 46 PCLKA 1 1 1 1 1: 48 PCLKA.	R/W
b5	SFMDTY	Duty ratio correction function select for the QSPCLK signal	0: Make no correction 1: Delay the rising of the QSPCLK signal by 0.5 PCLKA cycles. (Valid when PCLKA is multiplied by an odd number.)	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When PCLKA multiplied by an odd number is selected, the high-level width of the QSPCLK signal is longer than the low-level width by 1 PCLKA before duty ratio correction.

39.2.4 Status Register (SFMSST)

Address(es): QSPI.SFMSST 6400 000Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PFOFF	PFFUL	—	PFCNT				
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b4 to b0	PFCNT	Number of bytes of prefetched data	b4 b0 0 0 0 0 0: 0 bytes 0 0 0 0 1: 1 byte 0 0 0 1 0: 2 bytes 0 0 0 1 1: 3 bytes 0 0 1 0 0: 4 bytes 0 0 1 0 1: 5 bytes 0 0 1 1 0: 6 bytes 0 0 1 1 1: 7 bytes 0 1 0 0 0: 8 bytes 0 1 0 0 1: 9 bytes 0 1 0 1 0: 10 bytes 0 1 0 1 1: 11 bytes 0 1 1 0 0: 12 bytes 0 1 1 0 1: 13 bytes 0 1 1 1 0: 14 bytes 0 1 1 1 1: 15 bytes 1 0 0 0 0: 16 bytes 1 0 0 0 1: 17 bytes 1 0 0 1 0: 18 bytes. Other settings are reserved.	R
b5	—	Reserved	This bit is read as 0.	R
b6	PFFUL	Prefetch buffer state	0: Prefetch buffer has free space 1: Prefetch buffer is full.	R
b7	PFOFF	Prefetch function operating state	0: Prefetch function operating 1: Prefetch function not enabled or not operating.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

39.2.5 Communication Port Register (SFMCOM)

Address(es): QSPI.SFMCOM 6400 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SFMD							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bits	Symbol	Bit name	Description	R/W
b7 to b0	SFMD	Port select for direct communication with the SPI bus	Input to and output from this port is converted to an SPI bus cycle. This port is only accessible in direct communication mode, when DCOM = 1. Access to this port is ignored in the ROM access mode.	R/W
b 31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.6 Communication Mode Control Register (SFMCMD)

Address(es): QSPI.SFMCMD 6400 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCOM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b0	DCOM	Mode select for communication with the SPI bus	0: ROM access mode 1: Direct communication mode.	R/W
b 31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.7 Communication Status Register (SFMCST)

Address(es): QSPI.SFMCST 6400 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	EROMR	—	—	—	—	—	—	COMBSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b0	COMBSY	SPI bus cycle completion state in direct communication	0: No serial transfer being processed 1: Serial transfer being processed.	R
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	EROMR	ROM access detection status in direct communication mode	0: ROM access not detected 1: ROM access detected.	R/(W)*1
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit.

39.2.8 Instruction Code Register (SFMSIC)

Address(es): QSPI.SFMSIC 6400 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SFMCIC							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b7 to b0	SFMCIC	Serial flash instruction code to substitute		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.9 Address Mode Control Register (SFMSAC)

Address(es): QSPI.SFMSAC 6400 0024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	SFM4BC	—	—	SFMAS	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	Symbol	Bit name	Description	R/W
b1, b0	SFMAS	Number of address bytes select for the serial interface	b1 b0 0 0: 1 byte 0 1: 2 bytes 1 0: 3 bytes 1 1: 4 bytes.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SFM4BC	Default instruction code select, when the serial interface address width is 4 bytes	0: Do not use 4-byte address read instruction code 1: Use 4-byte address read instruction code.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.10 Dummy Cycle Control Register (SFMSDC)

Address(es): QSPI.SFMSDC 6400 0028h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SFMXD								SFMXEN	SFMXST	—	—	SFMDN[3:0]			
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b3 to b0	SFMDN[3:0]	Number of dummy cycles select for Fast Read instructions	b3 b0 0 0 0 0: Default dummy cycles for each instruction: - Fast Read Quad I/O: 6 QSPCLK - Fast Read Quad Output: 8 QSPCLK - Fast Read Dual I/O: 4 QSPCLK - Fast Read Dual Output: 8 QSPCLK - Fast Read: 8 QSPCLK. 0 0 0 1: 3 QSPCLK*1 0 0 1 0: 4 QSPCLK 0 0 1 1: 5 QSPCLK 0 1 0 0: 6 QSPCLK 0 1 0 1: 7 QSPCLK 0 1 1 0: 8 QSPCLK 0 1 1 1: 9 QSPCLK 1 0 0 0: 10 QSPCLK 1 0 0 1: 11 QSPCLK 1 0 1 0: 12 QSPCLK 1 0 1 1: 13 QSPCLK 1 1 0 0: 14 QSPCLK 1 1 0 1: 15 QSPCLK 1 1 1 0: 16 QSPCLK 1 1 1 1: 17 QSPCLK.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SFMXST	XIP mode status	0: Normal (non-XIP) mode 1: XIP mode.	R
b7	SFMXEN	XIP mode permission in the QSPI	0: Prohibit XIP mode 1: Permit XIP mode.	R/W
b15 to b8	SFMXD	Mode data for serial flash. (Controls XIP mode.)		R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To avoid a conflict with the input/output switch of the serial flash pin connected to QIO0 pin, select more than 4 QSPCLK dummy cycles when the output enable signal is extended by setting the SFMOEX bit in the SFMSMD register to 1.

39.2.11 SPI Protocol Control Register (SFMSPC)

Address(es): QSPI.SFMSPC 6400 0030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	SFMSDE	—	—	SFMSPI	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b1, b0	SFMSPI	SPI protocol select	b1 b0 0 0: Extended SPI protocol 0 1: Dual SPI protocol 1 0: Quad SPI protocol 1 1: Setting prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SFMSDE	Minimum time select for input/output switch, when Dual SPI or Quad SPI protocol is selected and in standard read mode	0: Do not allocate minimum switch time 1: Allocate minimum switch time equivalent to 1 QSPCLK.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.12 Port Control Register (SFMPMD)

Address(es): QSPI.SFMPMD 6400 0034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SFMWPL	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	SFMWPL	WP pin level specification	0: Low level 1: High level.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

39.2.13 External QSPI Address Register (SFMCNT1)

Address(es): QSPI.SFMCNT1 6400 0804h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	QSPI_EXT[5:0]						—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b25 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b26	QSPI_EXT[5:0]	Bank switching address	When accessing from 6000 0000h to 63FF FFFFh, the address bus is set from QSPI_EXT[5:0] to the upper 6 bits of the internal bus address.	R/W

39.3 Memory Map

39.3.1 Internal Bus Space

The locations of the serial flash and control registers in the AHB space are determined by the address range of the configured area.

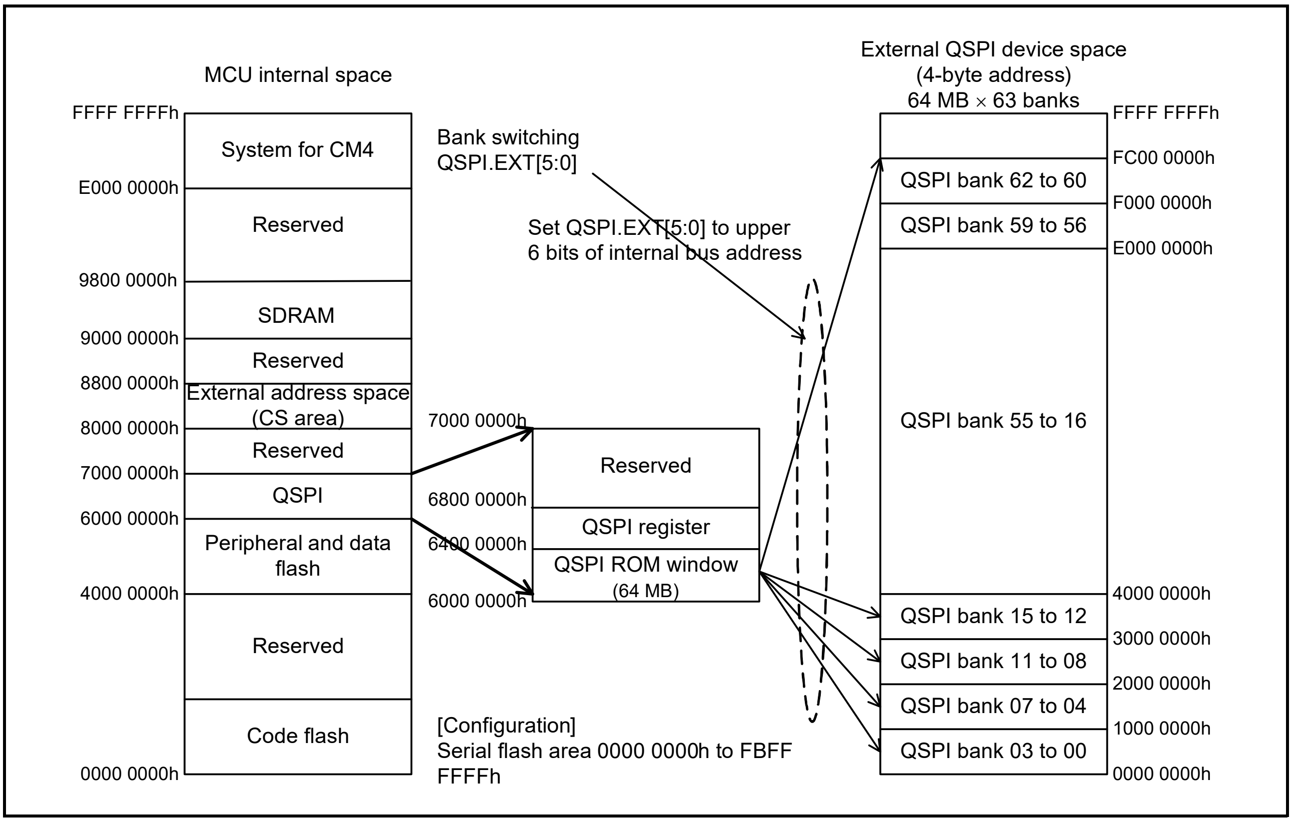


Figure 39.2 Default area setting and AHB space memory map

39.3.2 Address Width of the SPI Space and SPI Bus

The SPI space has a 32-bit address width for referencing the serial flash. When the SPI space is accessed for a read, an SPI bus cycle starts automatically, and data read from the serial flash is returned.

The address width of the SPI space is fixed at 32 bits. However, the address width of the SPI bus is selectable to 8, 16, 24, or 32 bits in the SFMAS[1:0] bits in the SFMSAC register. If 8, 16, or 24 bits is selected as the address width of the SPI bus, only the lower part of the address used to access the SPI space is posted to the serial flash through the SPI bus. As a result, the mirror image of the serial flash corresponding to the address width of the SPI bus repeatedly appears in the SPI space.

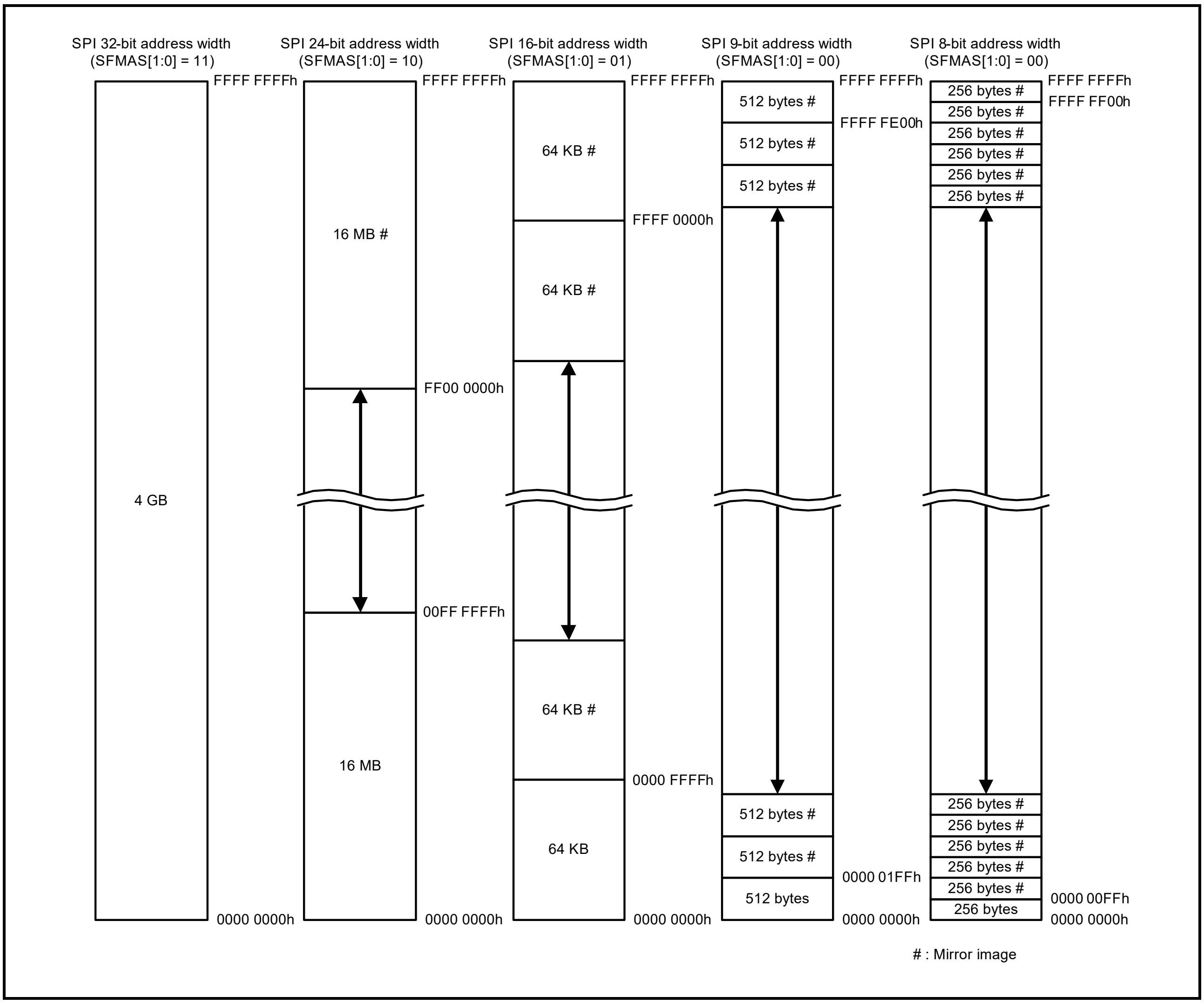


Figure 39.3 Memory map of SPI space

Note: The SPI bus address width is selectable to 8, 16, 24, or 32 bits in the SFMAS[1:0] bits in the SFMSAC register. When an 8-bit address width is selected, the address information of the ninth bit can be embedded in the Read instruction code. The address map in the figures is for the SPI 9-bit address width. For details on the Read instruction, see section 39.6.2, Standard Read Instruction.

39.4 SPI Bus

39.4.1 SPI Protocol

The QSPI supports Extended SPI, Dual SPI, and Quad SPI, in addition to the SPI protocol used for serial flash connection. The initial state is Extended SPI. To change the protocol, set the SFMSPI bit in the SFMSPC register. The Extended SPI protocol always outputs instruction codes from a single QIO0 pin. It performs subsequent address and data

I/O operations using one to four pins, depending on the instruction code format.

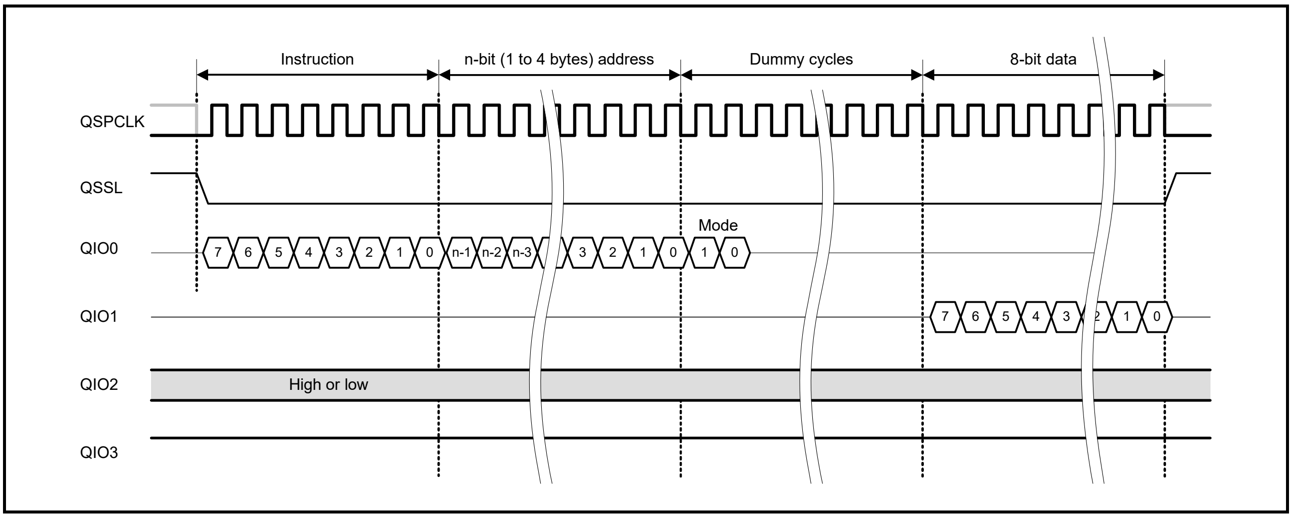


Figure 39.4 Extended SPI protocol example 1 for Fast Read

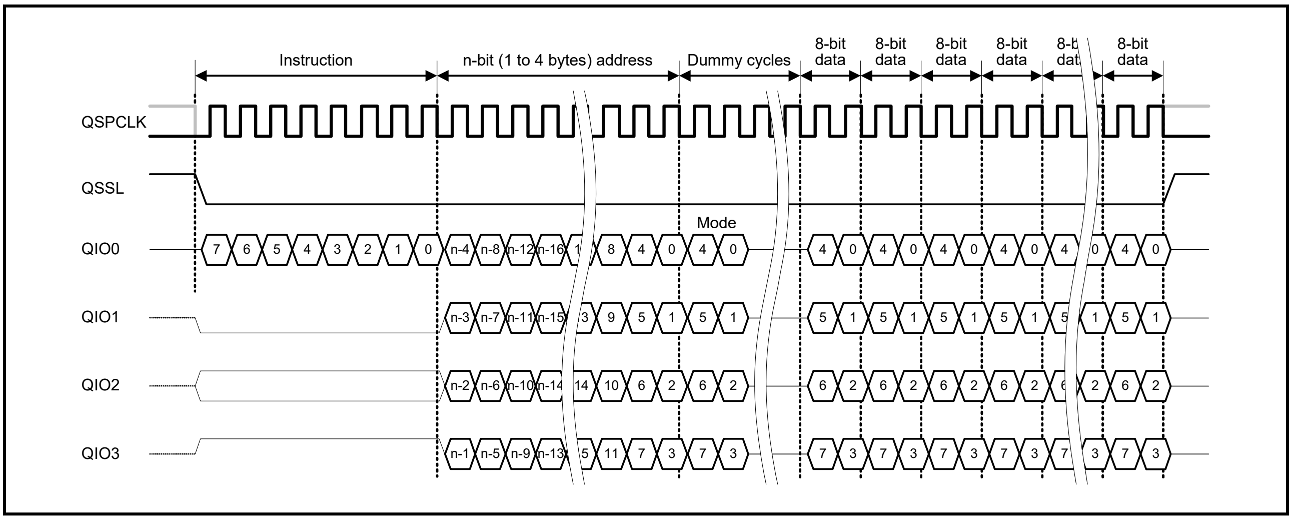


Figure 39.5 Extended SPI protocol example 2 for Fast Read Quad I/O

The Dual SPI protocol performs I/O operation of all signals such as instruction codes, addresses, and data using two pins, QIO0 and QIO1.

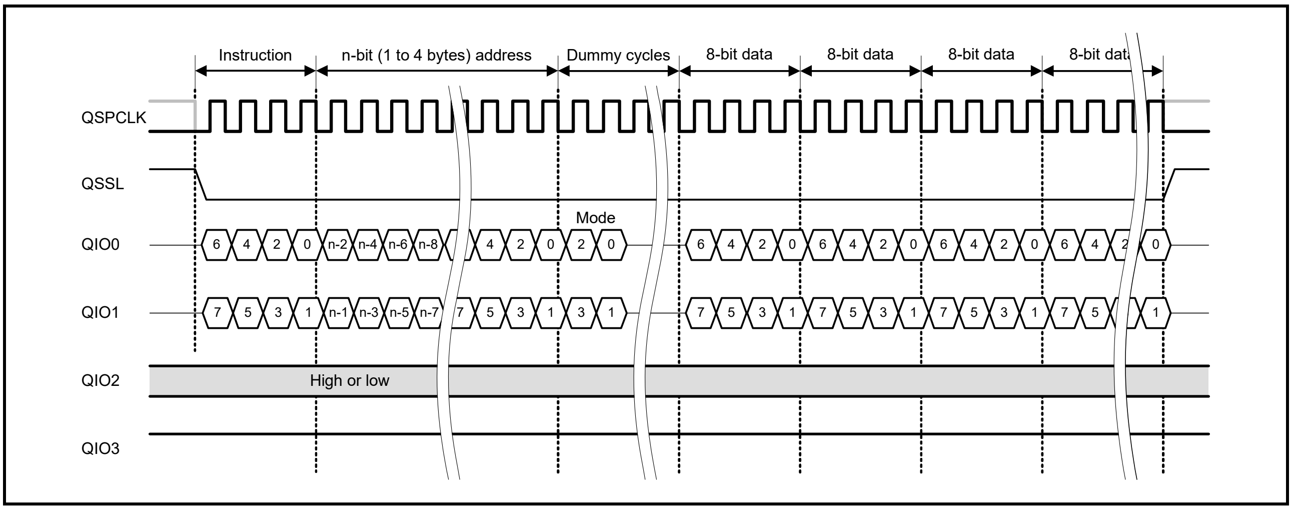


Figure 39.6 Dual SPI protocol example for Fast Read

The Quad SPI protocol performs I/O operation of all signals such as instruction codes, addresses, and data using four pins, QIO0, QIO1, QIO2, and QIO3.

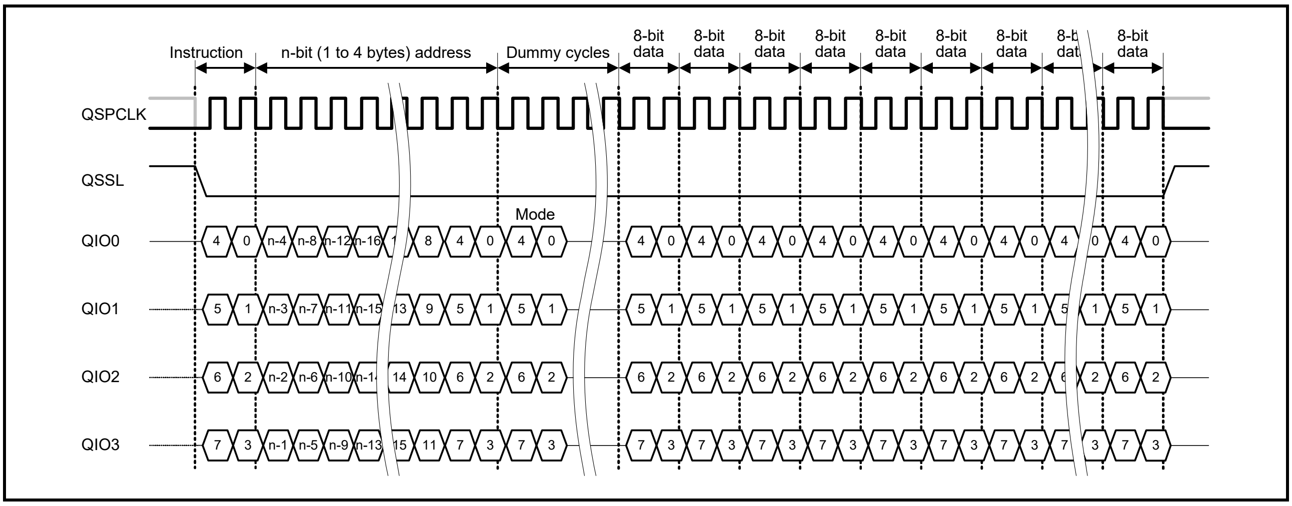


Figure 39.7 Quad SPI protocol example for Fast Read

39.4.2 SPI Mode

The initial SPI mode is set to SPI mode 0 or 3 by the CFGMD3 pin. This can be switched by changing the register setting during operation. The difference between SPI modes 0 and 3 is the standby level of the QSPCLK signal. The standby level of the QSPCLK signal in SPI mode 0 is low, and high in SPI mode 3.

Serial data is output from the QSPI on a falling edge of the serial clock and is read into the external flash on a rising edge of the serial clock. Serial data is output from the external flash on a falling edge of the serial clock and is read into the QSPI on the next falling edge of the serial clock.

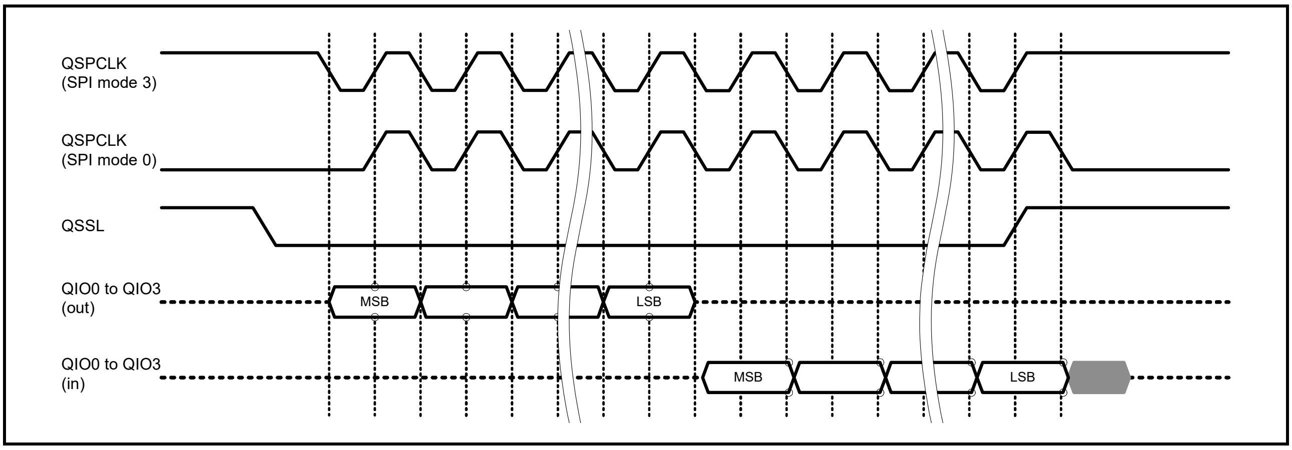


Figure 39.8 Basic serial interface timing

39.5 SPI Bus Timing Adjustment

The timing of the SPI bus signal can be adjusted in the registers. The configured timing is applied to all SPI bus accesses, for both ROM access and direct communication.

39.5.1 SPI Bus Reference Cycles

The SPI bus operates on reference cycles obtained by multiplying PCLKA by an integer. The reference cycles are selectable within the range of PCLKA multiplied by 2 to 48 in the SFMDV[4:0] bits in the SFMSKC register.

Table 39.3 Relationship among SFMDV[4:0] bits, cycle multiplier, and serial clock frequencies (1 of 2)

SFMDV[4:0]	Cycle multiplier	PCLKA frequency (MHz)
		120
11111	48	2.50
11110	46	2.61
11101	44	2.73
11100	42	2.86
11011	40	3.00
11010	38	3.16
11001	36	3.33
11000	34	3.53
10111	32	3.75
10110	30	4.00
10101	28	4.29
10100	26	4.62
10011	24	5.00
10010	22	5.45
10001	20	6.00
10000	18	6.67
01111	17	7.06
01110	16	7.50
01101	15	8.00
01100	14	8.57
01011	13	9.23
01010	12	10.00

Table 39.3 Relationship among SFMDV[4:0] bits, cycle multiplier, and serial clock frequencies (2 of 2)

SFMDV[4:0]	Cycle multiplier	PCLKA frequency (MHz)
		120
01001	11	10.91
01000	10	12.00
00111	9	13.33
00110	8	15.00
00101	7	17.14
00100	6	20.00
00011	5	24.00
00010	4	30.00
00001	3	40.00
00000	2	60.00

39.5.2 QSPCLK Signal Duty Ratio

When the reference clock is configured as PCLKA multiplied by an even number, the high- and low-level widths of the QSPCLK signal match each other. When PCLKA is multiplied by an odd number, however, the high-level width is longer than the low-level width by 1 PCLKA.

To make the duty ratio of the QSPCLK signal close to 50% when PCLKA multiplied by an odd number is the reference clock, set the SFMDTY bit in the SFMSKC register to 1. With this setting, the rising edge of the QSPCLK output signal is delayed by one-half PCLKA cycle to perform an interface operation equivalent to a duty ratio of 50%.

When the reference clock is PCLKA multiplied by an even number, the SFMDTY setting in the SFMSKC register is ignored.

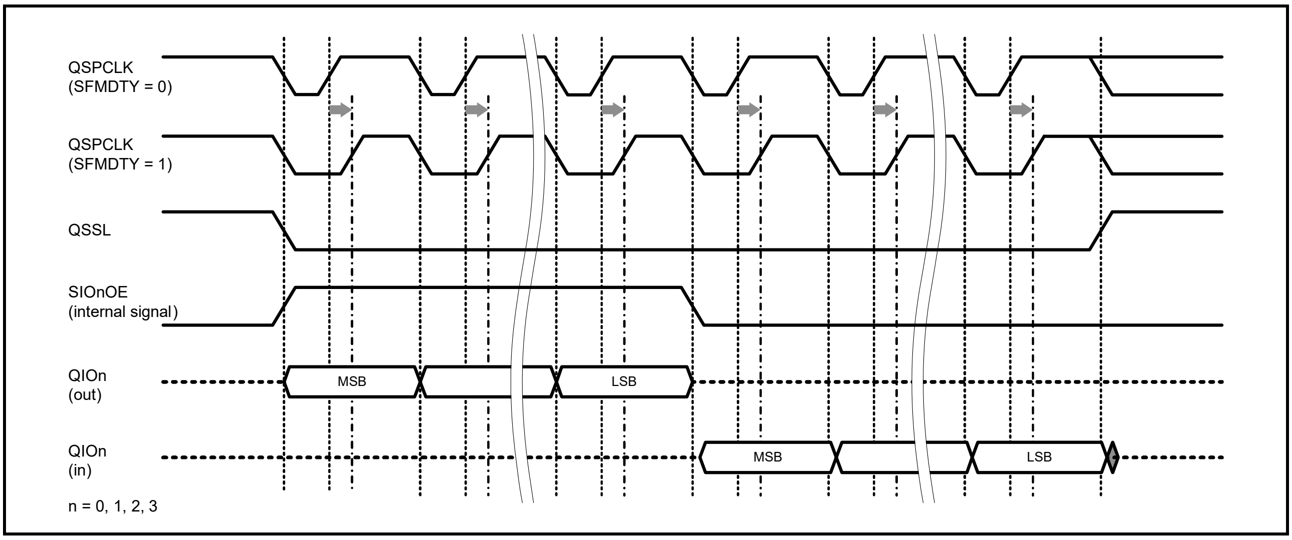


Figure 39.9 Example correction of the QSPCLK signal duty ratio using the SFMDTY bit, when PCLKA is multiplied by 3

39.5.3 Minimum High-Level Width for the QSSL Signal

Between adjacent SPI bus cycles, the QSSL signal must be held high (inactive) for a sufficient time to satisfy the deselect time required by the serial flash. The minimum high-level width of the QSSL output signal is selectable as the reference cycle multiplied by an integer from 1 to 16 in the SFMSW[3:0] bits in the SFMSSC register.

39.5.4 QSSL Signal Setup Time

When the QSPCLK signal first rises after the QSSL signal is driven low, the QSSL signal setup time can be configured to satisfy the serial flash requirements. The setup time can be selectable as 0.5 QSPCLK or 1.5 QSPCLK in the SFMSLD bit in the SFMSSC register.

The SFMSLD setting in the SFMSSC register is also applied to the allocate a setup time from the output of the serial data output enable signal (QIO0OE, QIO1OE, QIO2OE, or QIO3OE) until the first rising edge of the QSPCLK signal. Set a value that meets the most constrained timing condition for your application.

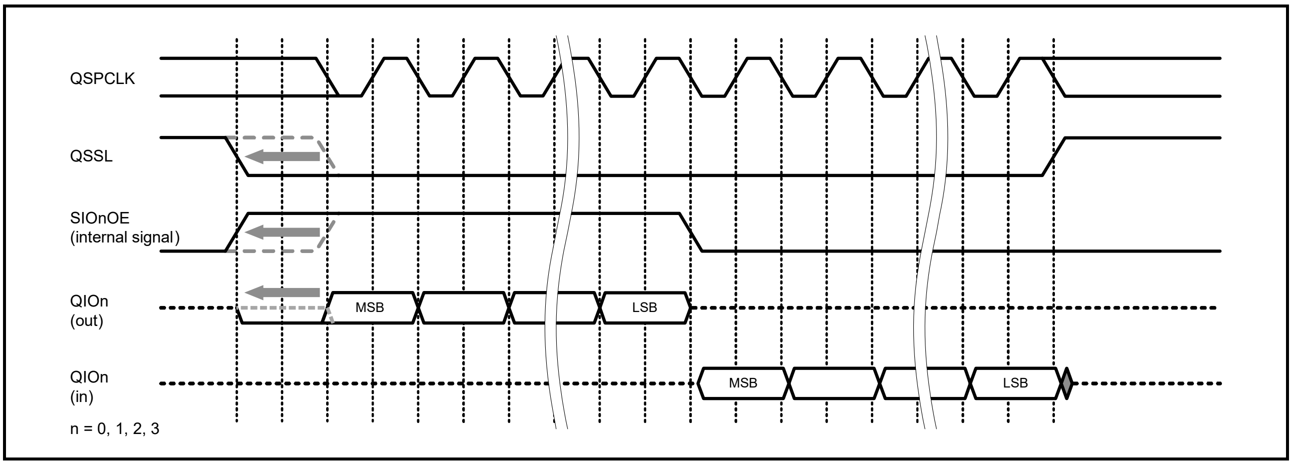


Figure 39.10 Setup time adjustment for the QSSL signal using the SFMSLD bit

39.5.5 QSSL Signal Hold Time

When the QSSL signal is driven high after the last rising edge of the QSPCLK signal, the QSSL signal hold time can be configured to satisfy the device requirements. The hold time is selectable as 0.5 QSPCLK or 1.5 QSPCLK in the SFMSHD bit in the SFMSSC register.

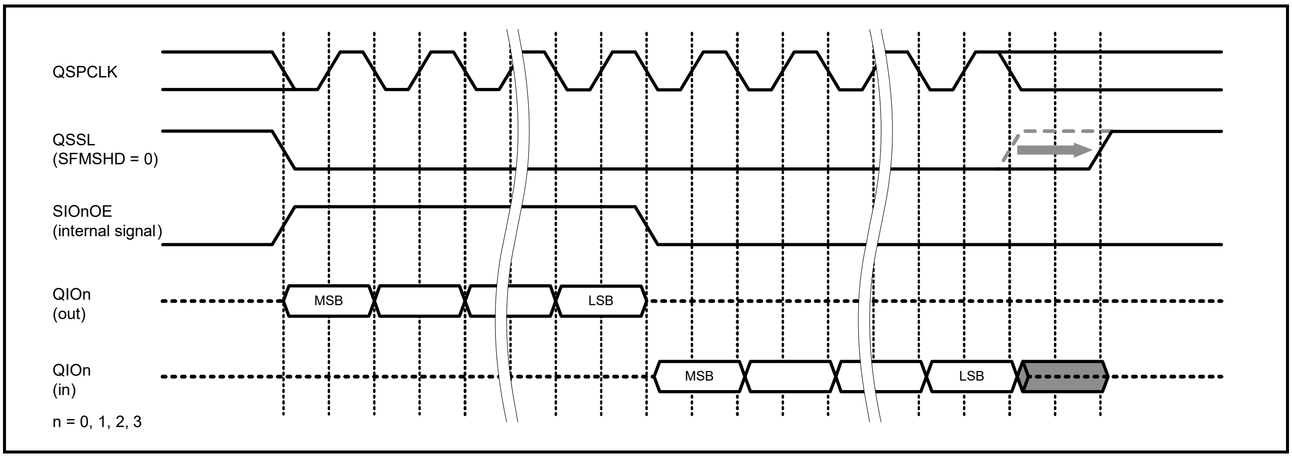


Figure 39.11 Hold time adjustment for the QSSL signal using the SFMSHD bit

39.5.6 Hold Time of the Serial Data Output Enable

The buffer output enable of the QIO0, QIO1, QIO2, or QIO3 pin can be extended by 1 QSPCLK using the SFMOEX bit in the SFMSMD register. The target extension signals include only the output enable signals: QIO0E, QIO1OE, QIO2OE, and QIO3OE. They do not include the output data signals: QIO0O, QIO1O, QIO2O, or QIO3O.

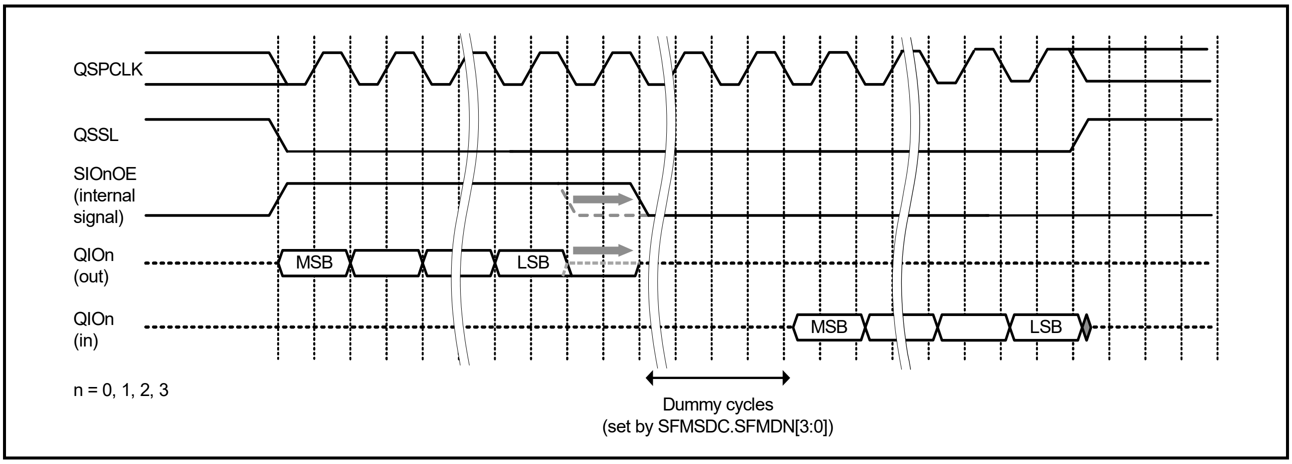


Figure 39.12 Hold time adjustment for output enable using the SFMOEX bit

39.5.7 Setup Time for Serial Data Output

When a command or address is transmitted to the serial flash, the setup time begins on serial data output and ends when the QSPCLK signal rises. If this setup time is insufficient, it can be extended by 1 PCLKA using the SFMOSW bit in the SFMSMD register. When SFMOSW is 1, the low-level width of QSPCLK during serial data transmission is extended by 1 PCLKA while data is being output from the QSPI. This function has no effect on serial data reception.

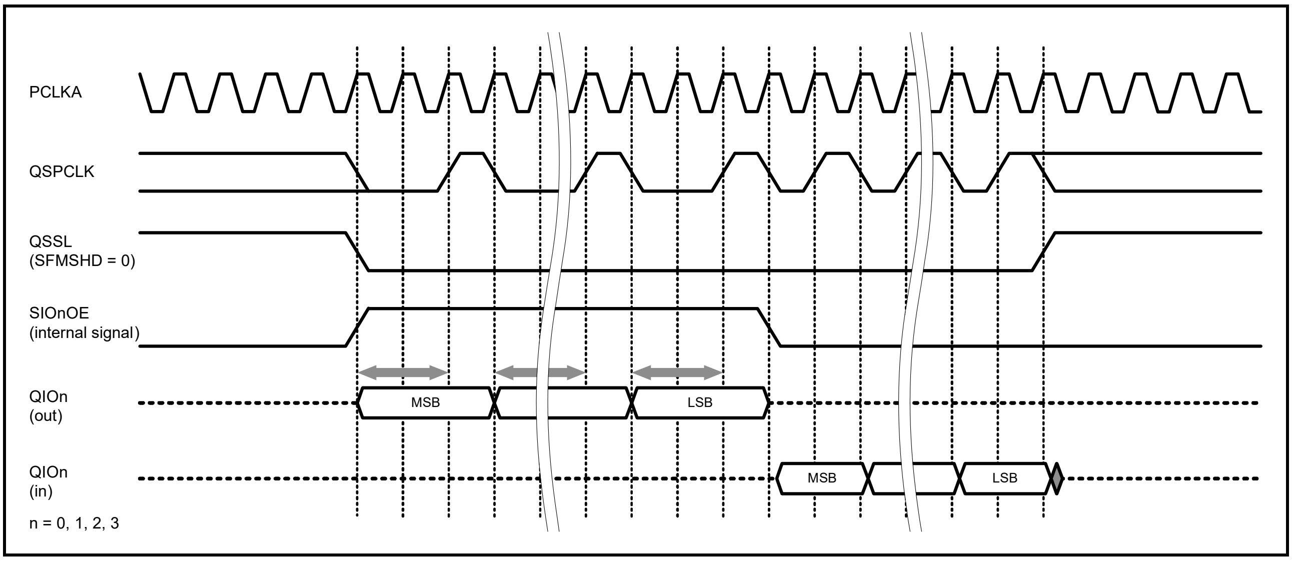


Figure 39.13 Setup time adjustment for serial data output using the SFMOSW bit

39.5.8 Hold Time for Serial Data Output

When a command or address is transmitted to the serial flash, the hold time begins on the rising edge of QSPCLK and ends when the serial data makes another transmission. If this hold time is insufficient, it can be extended by 1 PCLKA using the SFMOHW bit in the SFMSMD register. When SFMOSW is 1, the high-level width of QSPCLK during serial data transmission is extended by 1 PCLKA while data is being output from the QSPI. This function has no effect on serial data reception.

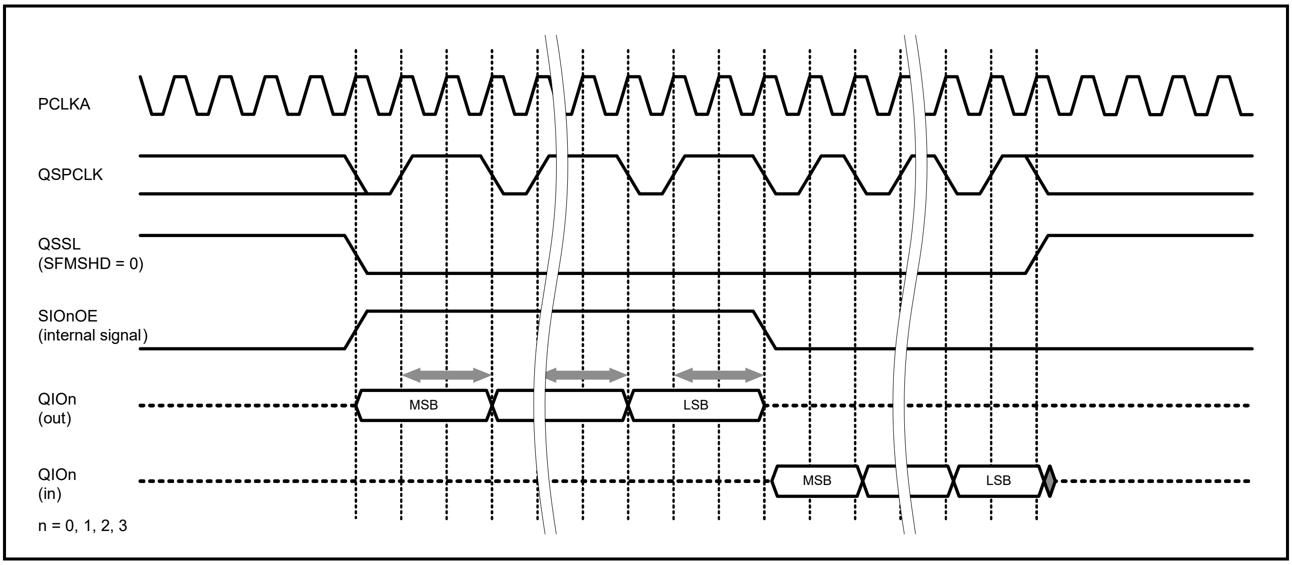


Figure 39.14 Hold time adjustment for serial data output using the SFMOHW bit

39.5.9 Serial Data Receiving Latency

The serial flash outputs data in synchronization with the falling edge of the QSPCLK signal. The QSPI receives that data in synchronization with the falling edge of the subsequent QSPCLK signal. The delay from when the serial flash starts outputting data until the QSPI receives that data is called the receiving latency. The QSPI adds a latency adjustment cycle immediately before the first data reception cycle in the SPI bus cycle. From the serial flash side, this is seen as an increase in the number of data reception cycles. This added latency adjustment cycle is not generated in the SPI bus cycle without accompanying data reception.

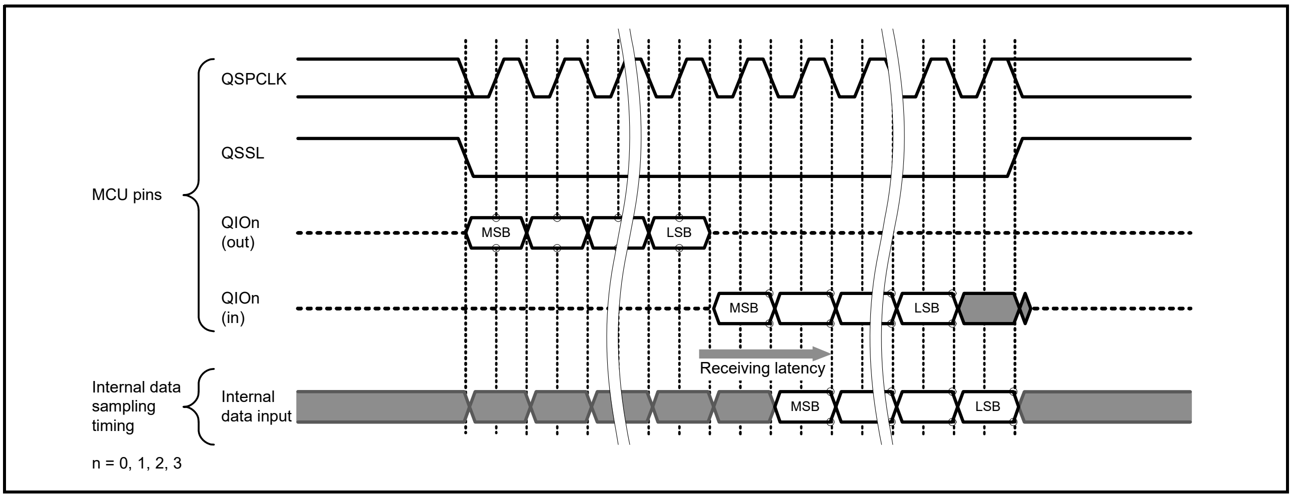


Figure 39.15 Receiving latency

39.6 SPI Instruction Set Used for Flash Access

39.6.1 SPI Instructions That Are Automatically Generated

When the serial flash is accessed, an SPI bus cycle using one of the instructions described in Table 39.4 to Table 39.8 is automatically generated based on the settings in the SFMAS[1:0] bits in the SFMSAC register and in the SFMSMD register.

Table 39.4 SPI instructions automatically generated when SFMAS[1:0] = 00b

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h[*1]	1	—	1 to ∞	Required: SFMRM[2:0] = 000, A8 = 0
	0Bh[*1]	1	—	1 to ∞	Required: SFMRM[2:0] = 000, A8 = 1

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.

Table 39.5 SPI instructions automatically generated when SFMAS[1:0] = 01b

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h[*1]	2	—	1 to ∞	Required: SFMRM[2:0] = 000

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.

Table 39.6 SPI instructions automatically generated when SFMAS[1:0] = 10b

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h[*1]	3	—	1 to ∞	Required: SFMRM[2:0] = 000
Fast Read	0Bh[*1]	3	8[*2]	1 to ∞	Selectable: SFMRM[2:0] = 001
Fast Read Dual Output	3Bh[*1]	3	8[*2]	1 to ∞	Selectable: SFMRM[2:0] = 010
Fast Read Dual I/O	BBh[*1]	3	4[*2]	1 to ∞	Selectable: SFMRM[2:0] = 011
Fast Read Quad Output	6Bh[*1]	3	8[*2]	1 to ∞	Selectable: SFMRM[2:0] = 100
Fast Read Quad I/O	EBh[*1]	3	6[*2]	1 to ∞	Selectable: SFMRM[2:0] = 101
Write Enable	06h	—	—	—	Selectable: ENEX4B[1:0] = 10
Exit 4-Byte Mode	E9h	—	—	—	Selectable: ENEX4B[1:0] = 01,10

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. The number of dummy cycles is configurable in SFMSDC.SFMDN[3:0].

Table 39.7 SPI instructions automatically generated when SFMAS[1:0] = 11b and SFM4BC = 0

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h[*1]	4	—	1 to ∞	Required: SFMRM[2:0] = 000
Fast Read	0Bh[*1]	4	8[*2]	1 to ∞	Selectable: SFMRM[2:0] = 001
Fast Read Dual Output	3Bh[*1]	4	8[*2]	1 to ∞	Selectable: SFMRM[2:0] = 010
Fast Read Dual I/O	BBh[*1]	4	4[*2]	1 to ∞	Selectable: SFMRM[2:0] = 011
Fast Read Quad Output	6Bh[*1]	4	8[*2]	1 to ∞	Selectable: SFMRM[2:0] = 100
Fast Read Quad I/O	EBh[*1]	4	6[*2]	1 to ∞	Selectable: SFMRM[2:0] = 101
Write Enable	06h	—	—	—	Selectable: ENEX4B[1:0] = 10
Enter 4-Byte Mode	B7h	—	—	—	Selectable: ENEX4B[1:0] = 01,10

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. The number of dummy cycles is configurable in SFMSDC.SFMDN[3:0].

Table 39.8 SPI instructions automatically generated when SFMAS[1:0] = 11b and SFM4BC = 1

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	13h*1	4	—	1 to ∞	Required: SFMRM[2:0] = 000
Fast Read	0Ch*1	4	8*2	1 to ∞	Selectable: SFMRM[2:0] = 001
Fast Read Dual Output	3Ch*1	4	8*2	1 to ∞	Selectable: SFMRM[2:0] = 010
Fast Read Dual I/O	BCh*1	4	4*2	1 to ∞	Selectable: SFMRM[2:0] = 011
Fast Read Quad Output	6Ch*1	4	8*2	1 to ∞	Selectable: SFMRM[2:0] = 100
Fast Read Quad I/O	ECh*1	4	6*2	1 to ∞	Selectable: SFMRM[2:0] = 101
Write Enable	06h	—	—	—	Selectable: ENEX4B[1:0] = 10
Enter 4-Byte Mode	B7h	—	—	—	Selectable: ENEX4B[1:0] = 01,10

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. The number of dummy cycles is configurable in SFMSDC.SFMDN[3:0].

39.6.2 Standard Read Instruction

The standard Read instruction is a common read instruction supported by most serial flash. When an SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (03h/13h)*1 is output. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, is transmitted. Data is then received.

This standard Read instruction is selected in the initial QSPI settings.

Note 1. Many 4-Kb serial flash devices have an address field not larger than 1 byte (A7-A0) to minimize the overhead and to receive A8 information from bit 3 of the Read instruction code. To support these devices, the QSPI only outputs A8 (address bit 8) to bit [3] of the standard Read instruction code when an address width of 1 byte is specified (SFMAS[1:0] = 00). This means that 0Bh might be output instead of 03h as the standard Read instruction code. This code duplicates the Fast Read instruction code. However, for most of the 2-Kb or smaller serial flash devices, with an address width of 1 byte, bit 3 of a command is designed to be excluded from decoding as a don’t-care bit, so such a Read instruction code is recognized correctly as the standard Read instruction code. In rare cases, some serial flash devices allow bit 3 to be decoded. When such a serial flash is connected, configure your application to avoid access resulting in A8 = 1.

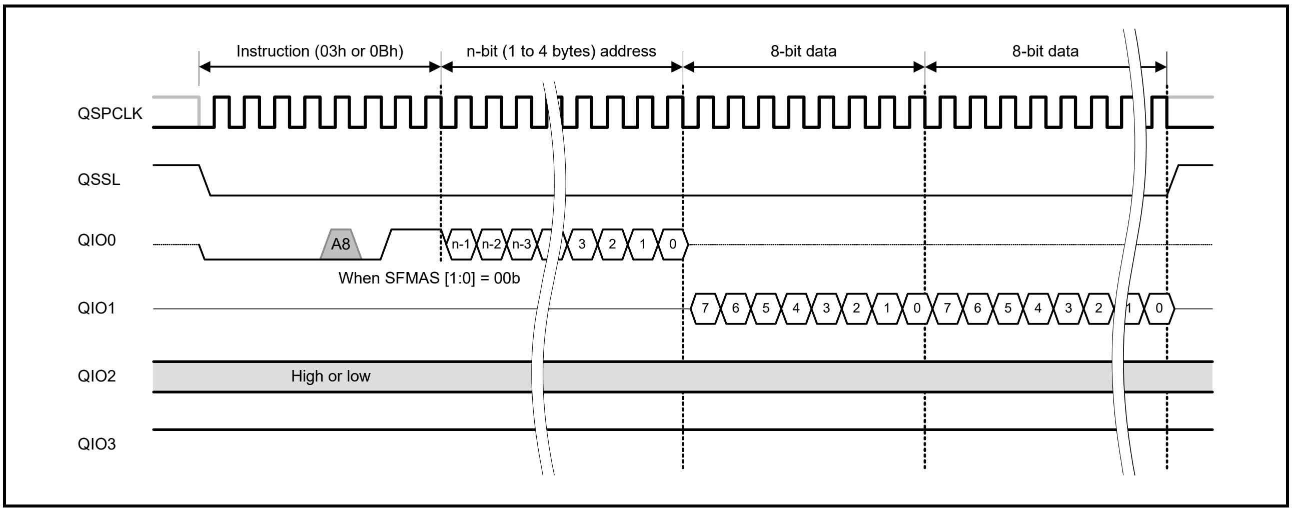


Figure 39.16 Standard Read bus cycle

39.6.3 Fast Read Instruction

The Fast Read instruction is a read instruction that supports a higher communication clock speed than the standard Read instruction. When an SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (0Bh/0Ch) is output. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in SFMSAC, and a certain number of dummy cycles, specified in the SFMSDC register, are transmitted. Data is then received.

The first two dummy cycles are used to select or deselect the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see section 39.8, XIP Control.

Switching to the Fast Read instruction is controlled in the SFMSMD register.

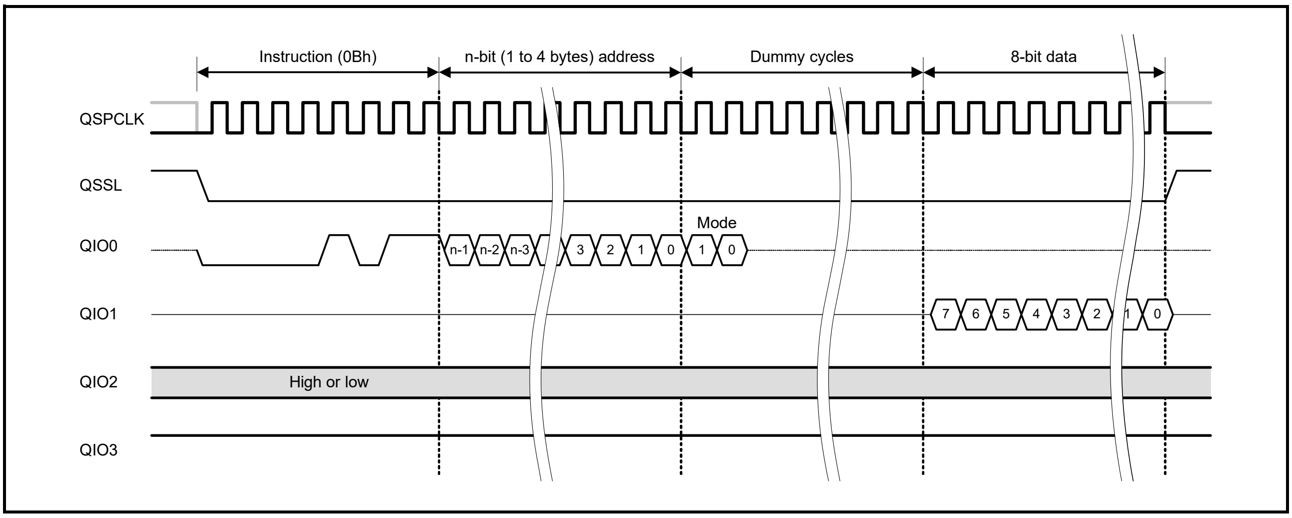


Figure 39.17 Fast Read bus cycle

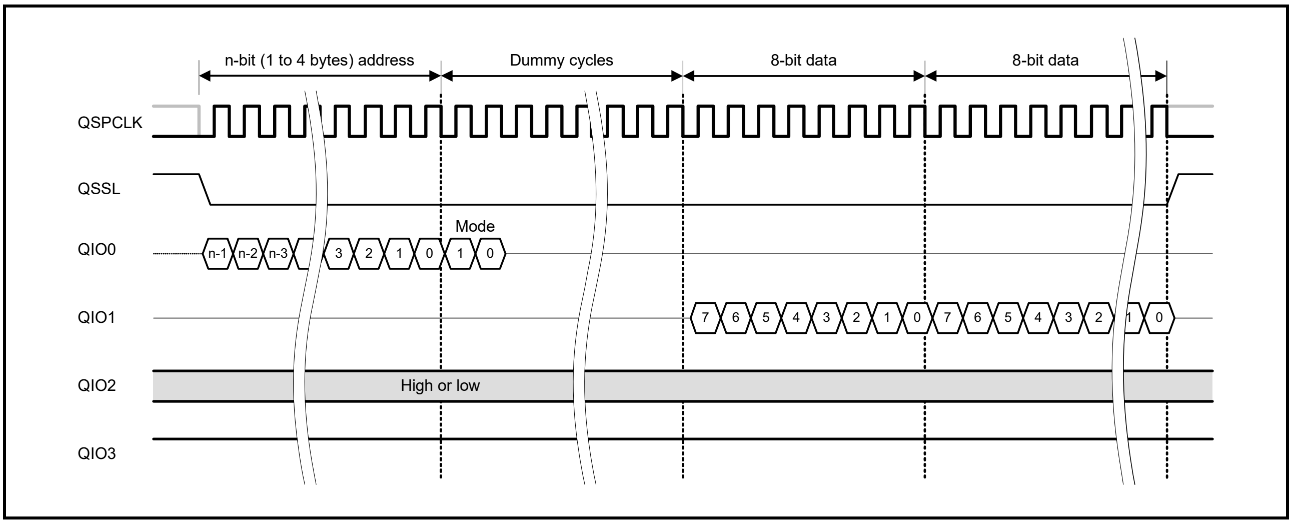


Figure 39.18 Fast Read bus cycle in XIP mode

Note: To use the Fast Read instruction, a serial flash device that supports Fast Read transfers is required.

39.6.4 Fast Read Dual Output Instruction

The Fast Read Dual Output instruction is a read instruction that uses two signal lines to receive data. When the SPI bus cycle starts, the serial flash select signal is asserted. The instruction code (3Bh/3Ch) and an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, are transmitted from the QIO0 pin. Next, a certain number of dummy cycles, specified in the SFMSDC register, is generated. Data is then received through the QIO0 and QIO1 pins. Even bit data is received from the QIO0 pin and odd bit data is received from the QIO1 pin.

The first two dummy cycles are used to select or deselect the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see section 39.8, XIP Control.

Switching to Fast Read Dual Output is controlled in the SFMSMD register.

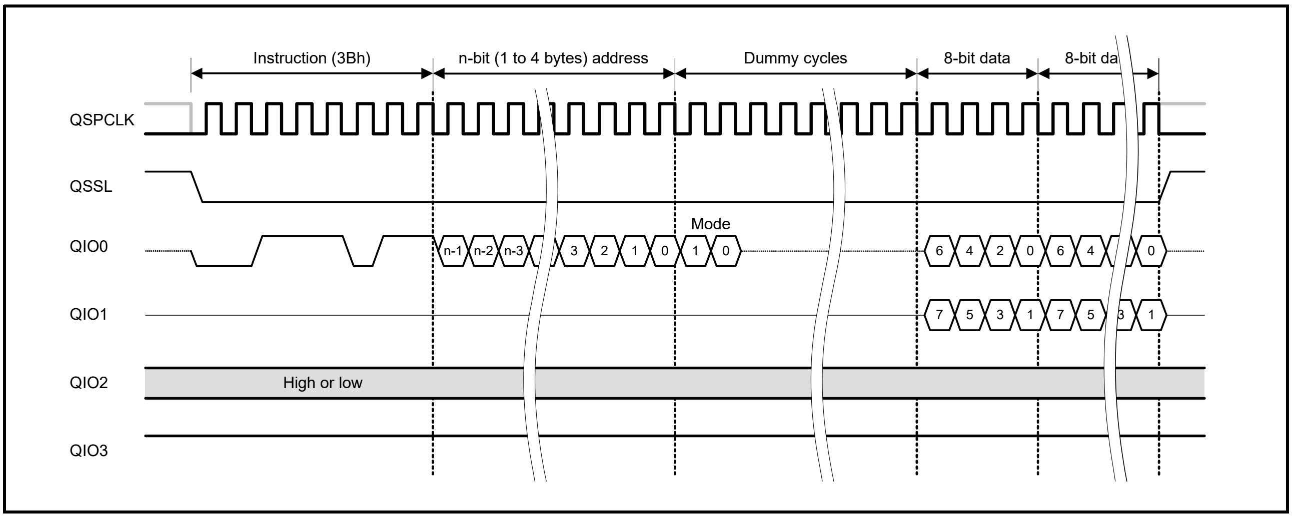


Figure 39.19 Fast Read Dual Output bus cycle

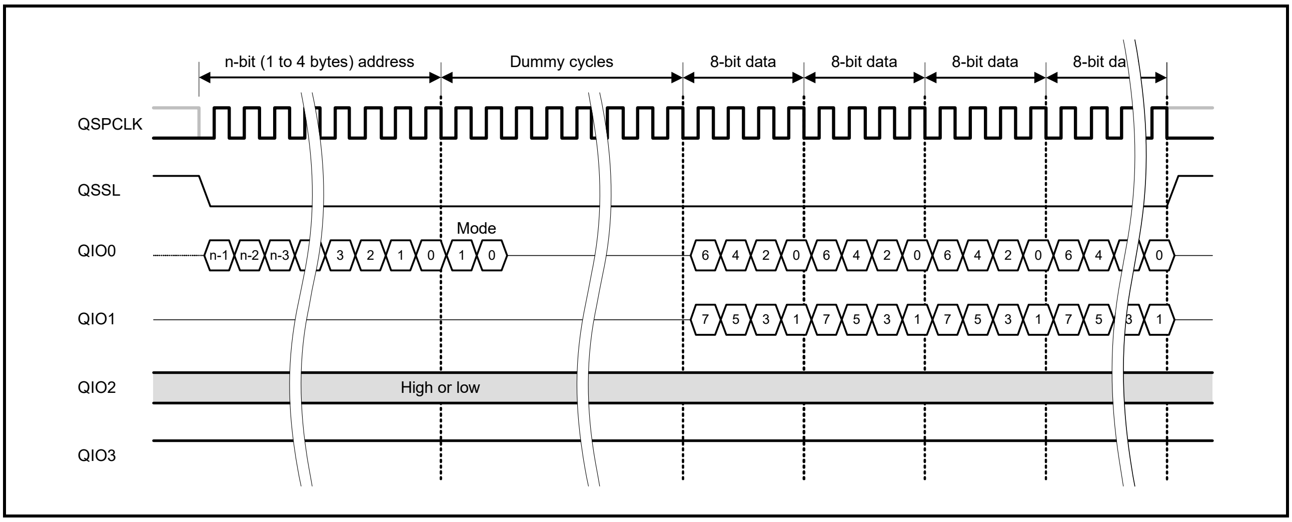


Figure 39.20 Fast Read Dual Output bus cycle in XIP mode

Note: To use the Fast Read Dual Output instruction, a serial flash device that supports Fast Read Dual Output transfers is required.

39.6.5 Fast Read Dual I/O Instruction

The Fast Read Dual I/O instruction is a read instruction that uses two signal lines to transmit an address and receive data. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (BBh/BCh) is output from the QIO0 pin. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, is transmitted through the QIO0 and QIO1 pins, and a certain number of dummy cycles, specified in the SFMSDC register, is generated. Data is then is received through the QIO0 and QIO1 pins. Address and dummy cycle transmission and data reception are performed through the QIO0 pin for even bits and through the QIO1 pin for odd bits.

The first two dummy cycles are used to select or deselect the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see section 39.8, XIP Control.

Switching to Fast Read Dual I/O is controlled in the SFMSMD register.

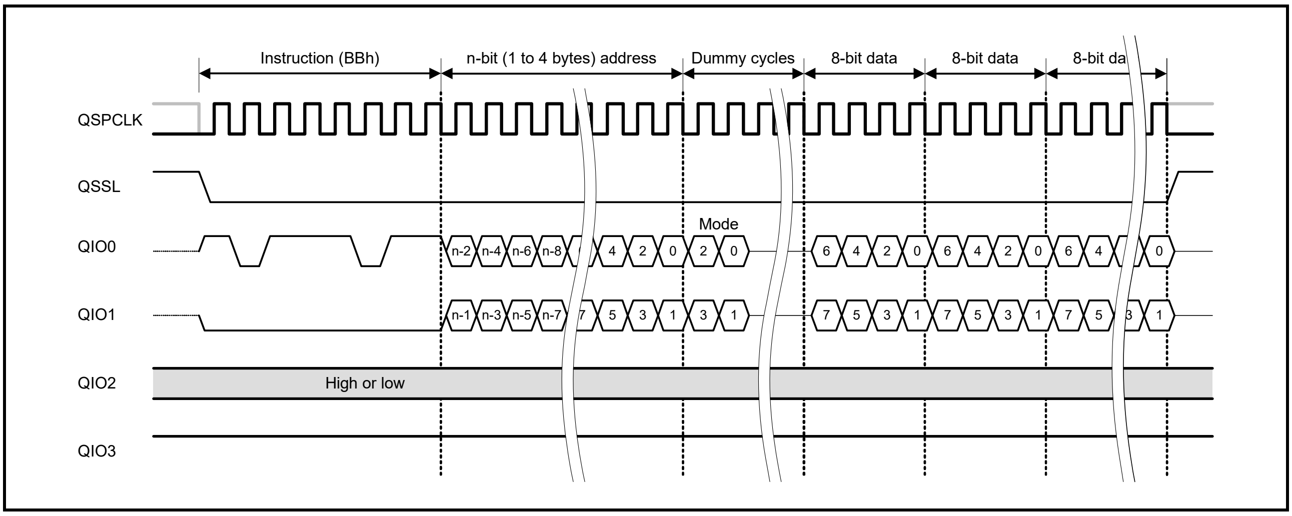


Figure 39.21 Fast Read Dual I/O bus cycle

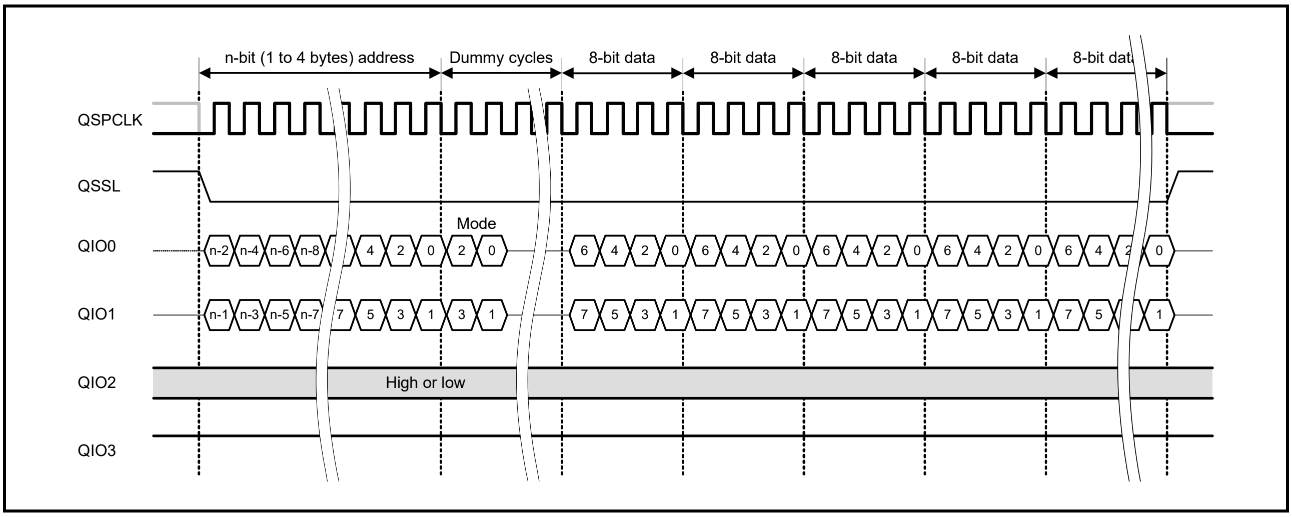


Figure 39.22 Fast Read Dual I/O bus cycle in XIP mode

Note: To use the Fast Read Dual I/O instruction, a serial flash device that supports Fast Read Dual I/O transfers is required.

39.6.6 Fast Read Quad Output Instruction

The Fast Read Quad Output instruction is a read instruction that uses four signal lines to receive data. When the SPI bus cycle starts, the serial flash select signal is asserted. The instruction code (6Bh/6Ch) and an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, are output from the QIO0 pin. Next, a certain number of dummy cycles, specified in the SFMDN[3:0] bits in the SFMSMD register, are generated. Data is then received through the QIO0, QIO1, QIO2, and QIO3 pins.

The first two dummy cycles are used to select or deselect the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see section 39.8, XIP Control.

Switching to Fast Read Quad Output is controlled in the SFMSMD register.

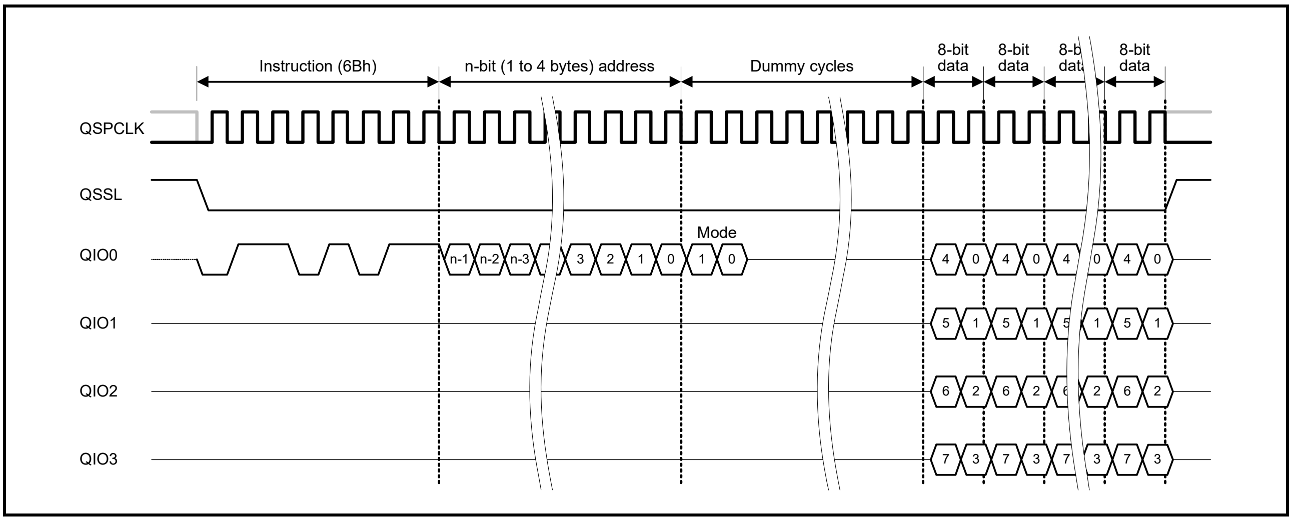


Figure 39.23 Fast Read Quad Output bus cycle

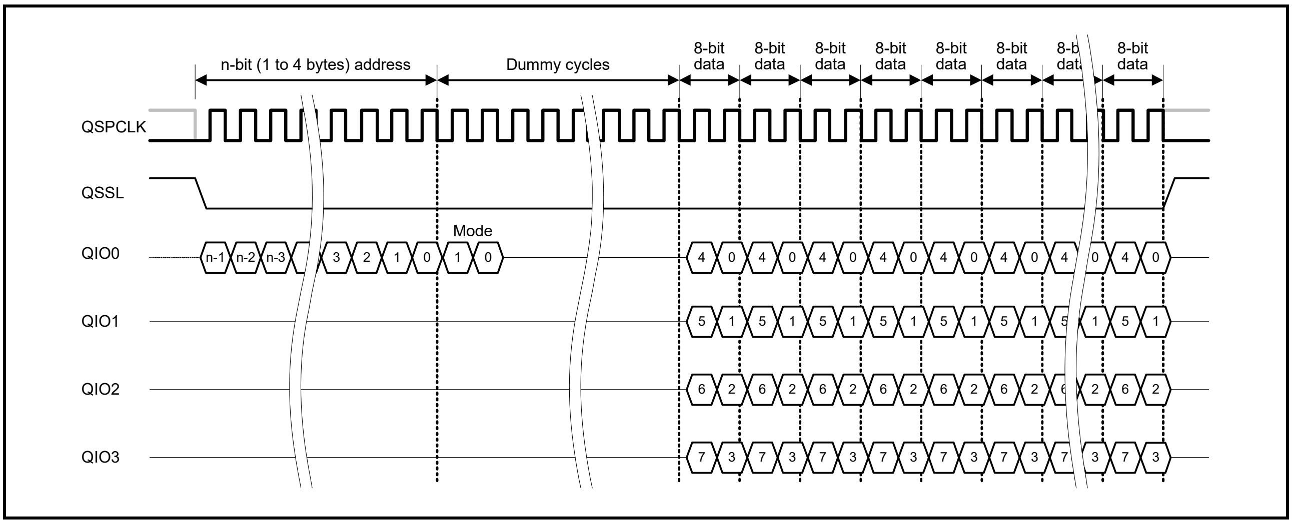


Figure 39.24 Fast Read Quad Output bus cycle in XIP mode

Note: To use Fast Read Quad Output, a serial flash that supports Fast Read Quad Output transfer is required.

39.6.7 Fast Read Quad I/O Instruction

The Fast Read Quad I/O instruction is a read instruction that uses four signal lines to transmit an address and receive data. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (EBh/ECh) is output. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits in the SFMSAC register, is transmitted through the QIO0, QIO1, QIO2, and QIO3 pins, and a certain number of dummy cycles, specified in the SFMDN[3:0] bits in the SFMSMD register, is generated. Data is then received through the QIO0, QIO1, QIO2, and QIO3 pins.

The first two dummy cycles are used to select or deselect the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and the instruction code is not output the next SPI bus cycle. For details on the XIP mode, see section 39.8, XIP Control.

Switching to Fast Read Quad I/O is controlled in the SFMSMD register.

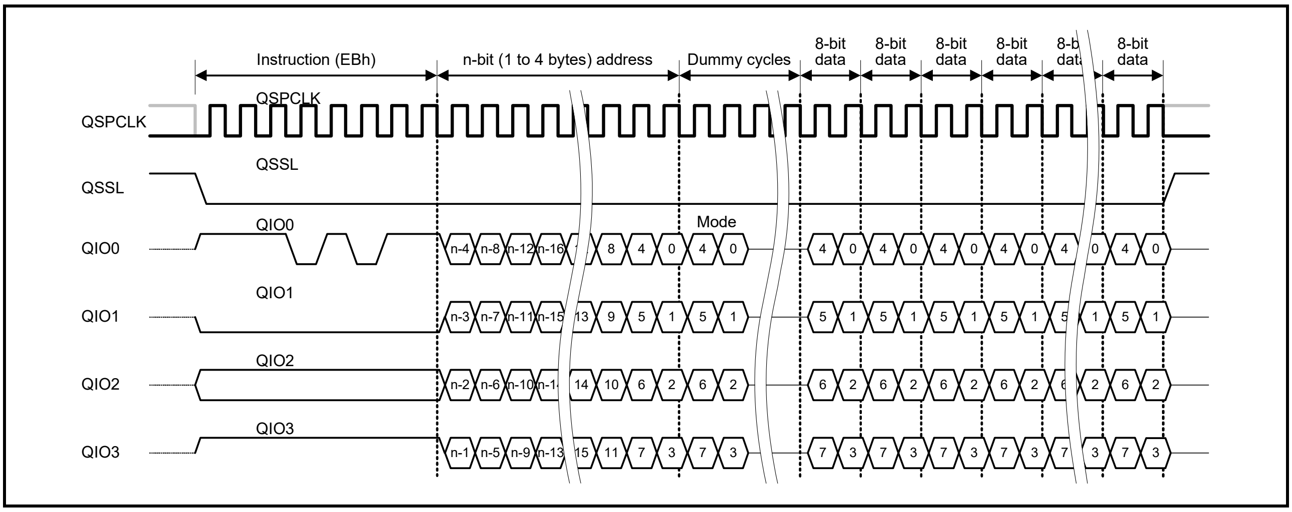


Figure 39.25 Fast Read Quad I/O bus cycle

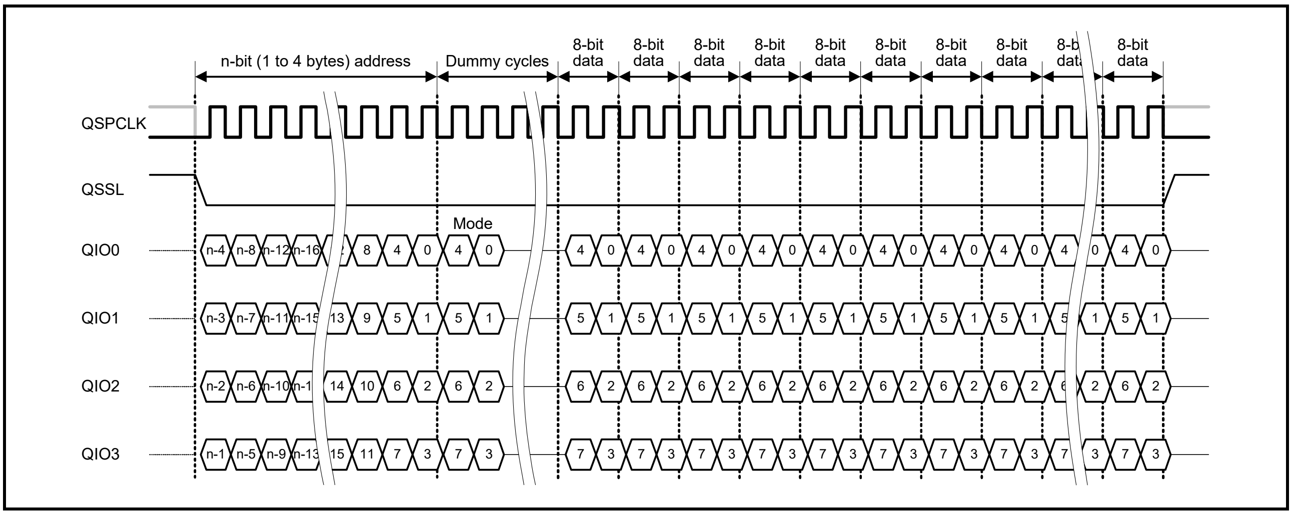


Figure 39.26 Fast Read Quad I/O bus cycle in XIP mode

Note: To use the Fast Read Quad I/O instruction, a serial flash device that supports Fast Read Quad I/O transfers is required.

39.6.8 Enter 4-Byte Mode Instruction

The Enter 4-Byte Mode instruction sets the serial flash address width to 4 bytes. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (B7h) is output.

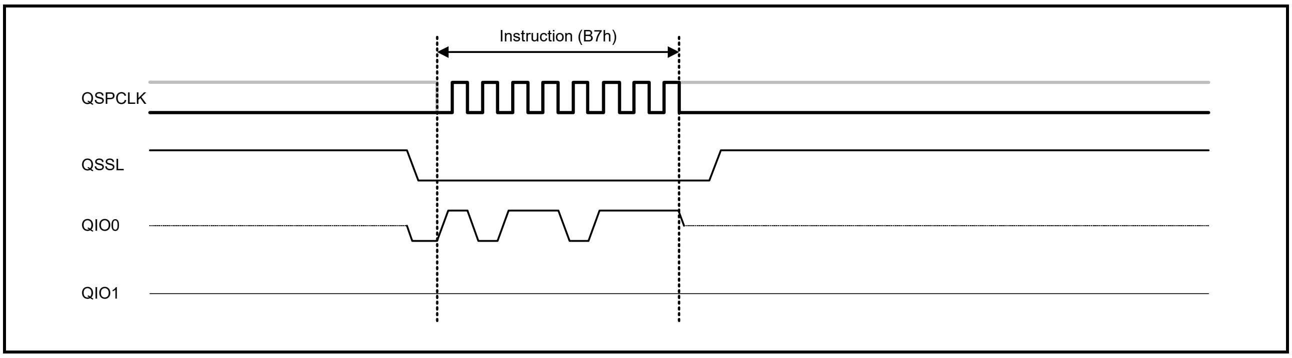


Figure 39.27 Enter 4-Byte Mode bus cycle

Note: The Enter 4-Byte Mode instruction is issued regardless of whether the serial flash is in 3- or 4-byte mode.

39.6.9 Exit 4-Byte Mode Instruction

The Exit 4-Byte Mode instruction sets the serial flash address width to 3 bytes. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (E9h) is output.

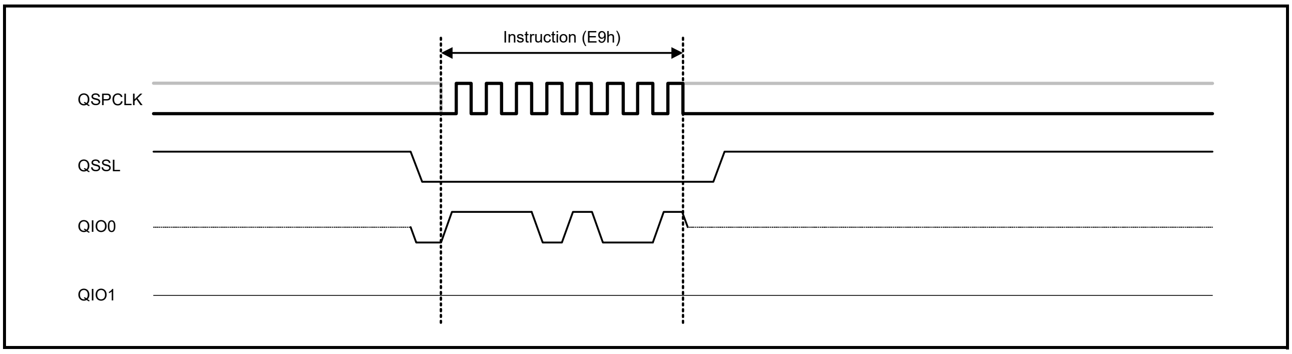


Figure 39.28 Exit 4-Byte Mode bus cycle

Note: The Exit 4-Byte Mode instruction is issued regardless of whether the serial flash is in 3- or 4-byte mode.

39.6.10 Write Enable Instruction

The Write Enable instruction enables changing of the serial flash address width. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (06h) is output.

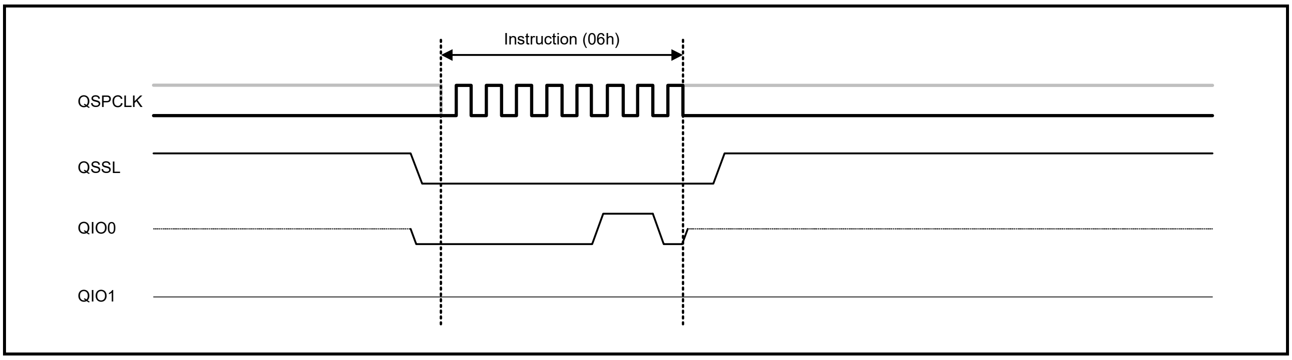


Figure 39.29 Write Enable bus cycle

39.7 SPI Bus Cycle Arrangement

39.7.1 Flash Read Based on Individual Conversion

ROM read internal bus cycles are individually converted to SPI bus cycles on a one-to-one basis. When a ROM read bus cycle is detected, the QSSL signal is asserted, and an SPI bus cycle starts. When data is received from the serial flash, the QSSL signal is deasserted, and the SPI bus cycle is complete.

When another ROM read bus cycle is detected, the QSSL signal is reasserted after ensuring the minimum high-level width of the QSSL signal is reached. Then another SPI bus cycle starts.

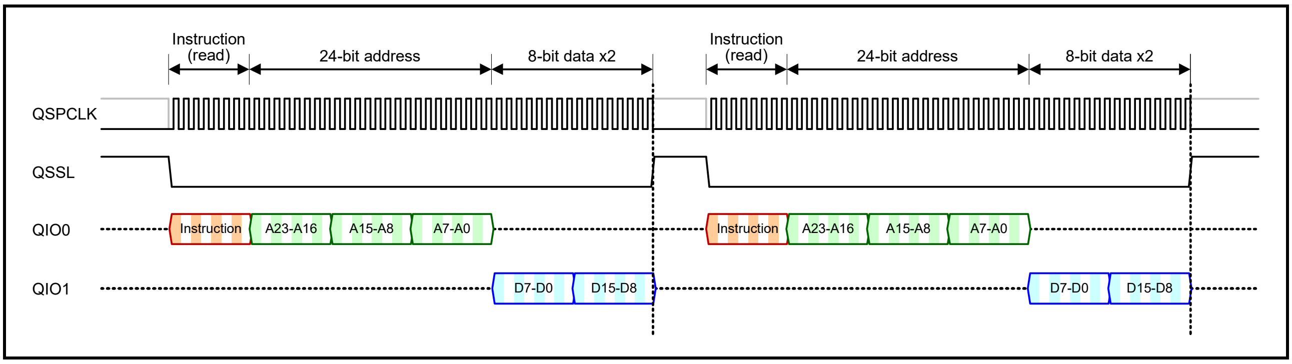


Figure 39.30 **Successive data read operations based on individual conversion**

39.7.2 Flash Read Using the Prefetch Function

In operations such as CPU instruction execution and block data transfer, data is often read in ascending order from contiguous flash addresses. Serial flash provides the ability to repeat data reception without reissuing an instruction code and address. However, if bus cycles issued by the MCU are individually converted, SPI bus cycles are separated from each other, resulting in a failure to take advantage of this feature of serial flash. The QSPI contains a prefetch function to ensure the use of this capability.

To enable the prefetch function, set the SFMPFE bit in the SFMSMD register to 1. When the prefetch function is enabled, data is received continuously and stored in the buffer, without waiting for another flash read request. When the MCU performs a flash read operation, an address check is made. If an address match is confirmed, the data in the buffer is passed to the MCU. If an address mismatch is found, the data in the buffer is discarded, and a new SPI bus cycle is issued.

The buffer for prefetching is 18 bytes long. When this buffer is full, the SPI bus cycle is ended. When the buffer data is read to create free space, a new SPI bus cycle is automatically started to resume prefetching.

The prefetch function allows for efficient transfer operations when data is read in ascending order from contiguous addresses, as in instruction fetch and block data transfer.

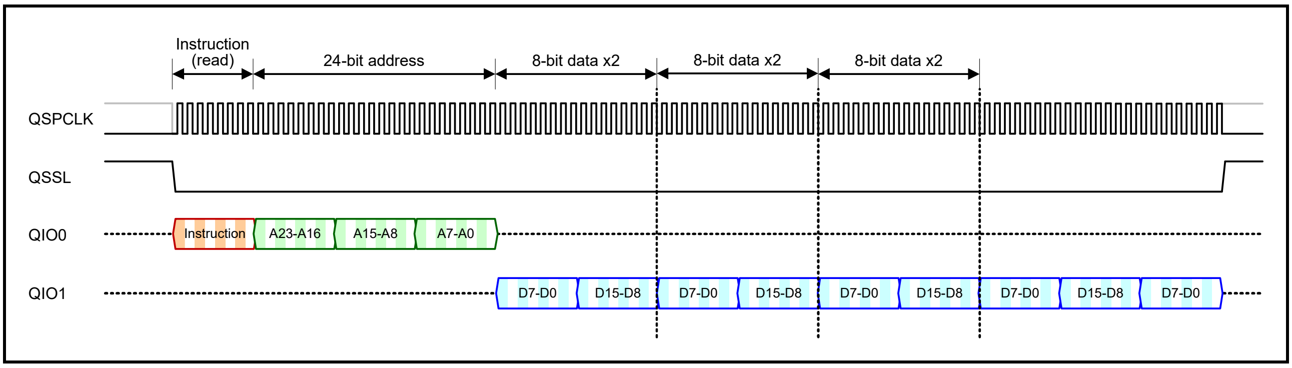


Figure 39.31 **Successive data read operations using the prefetch function**

39.7.3 Halt of Prefetching

If a ROM read bus cycle for reading from another address occurs during a serial transfer for prefetching, the unnecessary serial transfer being made is halted and a new SPI bus cycle is started. Usually, such a halt of serial transfer occurs on data reception byte boundaries. However, if the SFMPAE bit in the SFMSMD register is set to 1, the halt can occur on locations other than byte boundaries. To use this function, the serial flash device must support halts not on byte boundaries.

39.7.4 Direct Specification of Prefetch Destination

When the SFMPFE bit is set and the QSPI receives internal bus write access to the QSPI window area, the system obtains it as a prefetch address and starts to prefetch. Internal bus write access to the QSPI window area can only be used to obtain prefetch address data. Writes to serial flash cannot be performed.

Combining this function with the prefetch state polling function described in section 39, Prefetch State Polling, can reduce the load on the internal bus when data is read from a low-speed serial flash.

Note: When writing to the QSPI window area to indicate a prefetch destination, write to the first byte of the address where prefetching is to be started. Writes to the QSPI window area with a data size of 2 bytes or more return an ERROR response.

39.7.5 Prefetch State Polling

Reading data from a low-speed serial flash increases system load, because the internal bus enters a wait state until the SPI reception bus cycle is complete. The prefetch state polling function is provided to reduce this load.

The PFOFF bit in the SFMSST register indicates the state of the prefetch function, and the PFCNT[4:0] bits in the SFMSST register indicate the number of data bytes already prefetched. This allows the prefetch status to be determined with a single CPU operation.

```
//
// copy 1K byte (32bit x 256 word) data from serial flash to SDRAM
//
unsigned long *sptr;                    // pointer for the serial flash
unsigned long *dptr;                    // pointer for the SDRAM
int i;

SFMSMD |= 0x0040;                        // set SFMPFE bit to enable prefetch
*( (volatile unsigned char *) sptr ) = 0;    // make the TAG valid to start prefetch

for ( i = 0 ; i < 256 ; i++ ){
while ( ( SFMSST & 0x00FF ) < 0x04 ){};   // waiting for 4-byte data to be received
*(dptr++) = *(sptr++);
}
```

Note: When executing a polling program, place the program outside of the serial flash or enable the instruction cache. If the polling program is executed when the program is on the serial flash or is executed without using the instruction cache, the prefetch target frequently switches to an instruction code. This eliminates the effect of polling, and an infinite loop can result because the prefetch buffer is not filled.

39.7.6 Flash Read Using the SPI Bus Cycle Extension Function

If the SFMSE[1:0] bits in the SFMSMD register are set to a value other than 00b, the QSPI waits for next flash read, suspending the SPI bus cycle, while stopping the QSPCLK signal and holding the QSSL signal low even after data is obtained from the serial flash.

If the address of the next flash read is contiguous in ascending order, the toggling of the QSPCLK signal is restarted to continue reception of subsequent data. If the address of the next flash read is not contiguous in ascending order, the QSSL signal is driven high once to end the SPI bus cycle being suspended. A new SPI bus cycle is then started.

When data is read intermittently from ascending order contiguous addresses, this function enables an efficient transfer operation to be performed by reducing the overhead for instruction code and address transmission.

The SPI bus cycle extension time is selectable in the SFMSE[1:0] bits in the SFMSMD register. When the specified extension time elapses, the QSSL signal returns to the high level to automatically end the SPI bus cycle being suspended. If the SFMSE[1:0] bits are set to 11b, QSSL is extended infinitely. This increases the power consumption of the serial flash, so the system must be designed accordingly.

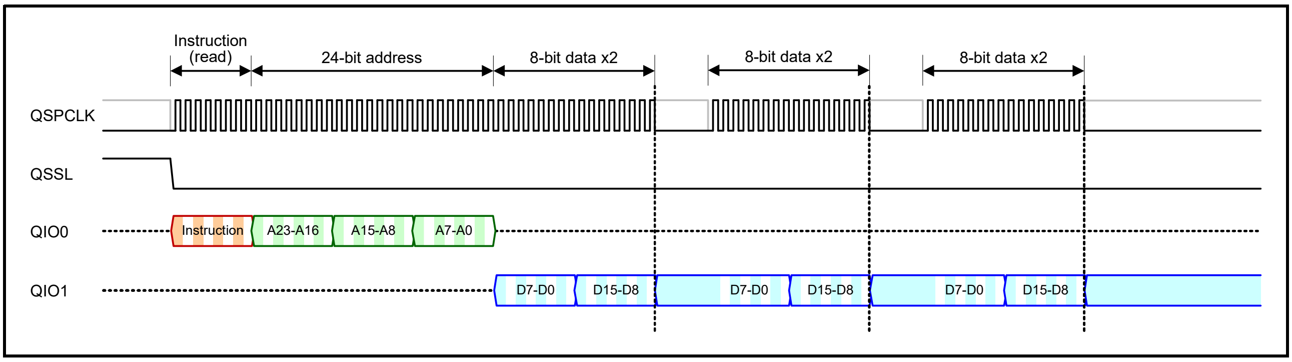


Figure 39.32 Successive data read operations using the SPI bus cycle extension

39.8 XIP Control

Some serial flash devices allow latencies to be reduced by skipping instruction code reception for flash reads. This instruction code skip function is selected in mode data received during the dummy cycle period of the previous serial bus cycle.

In the dummy cycle of the Fast Read instructions, the QSPI controls the XIP mode of the serial flash by using the serial data signal to send the mode data set in the SFMXD[7:0] bits in the SFMSDC register during the first 2 cycles, as shown in Figure 39.33.

The mode data to enable the XIP mode differs for each serial flash. Accordingly, set the appropriate mode data in the SFMXD[7:0] bits.

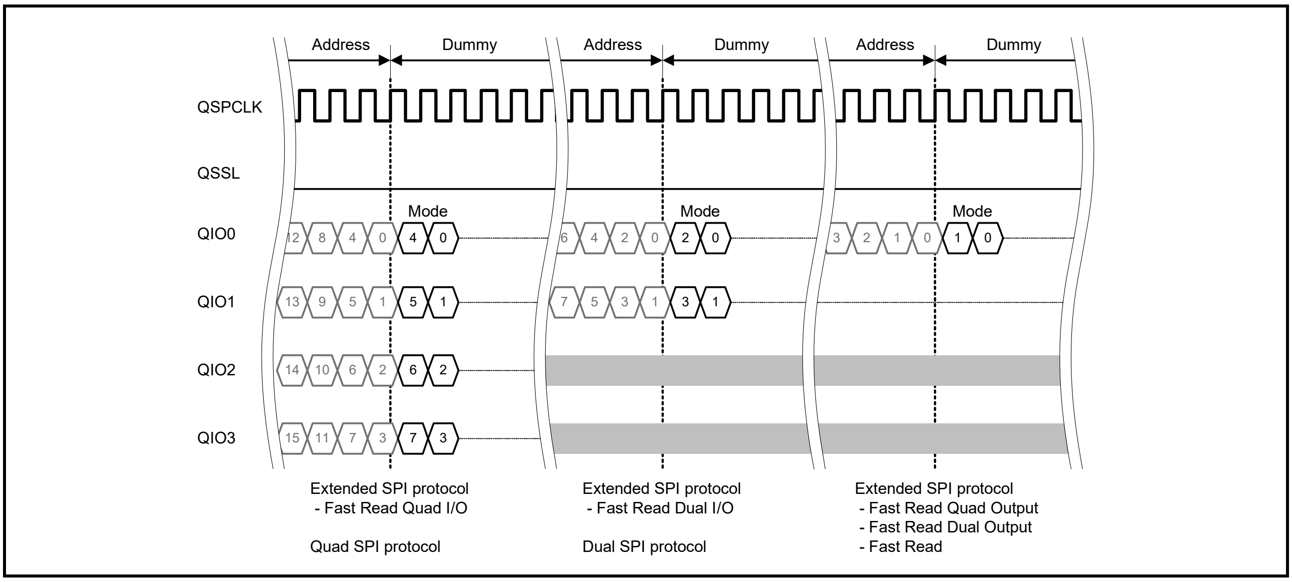


Figure 39.33 XIP mode control data

39.8.1 Selecting the XIP Mode

To select the XIP mode, specify the XIP mode configuration for the serial flash device in the SFMXD[7:0] bits in the SFMSDC register, and set the SFMXEN bit to 1. In the dummy cycle of the next Fast Read instruction, the mode data specified in the SFMXD[7:0] bits is transferred to the serial flash device. From that point, the XIP mode is enabled in both the serial flash controller and the serial flash device. To confirm completion of the XIP mode select procedure, read 1 from the SFMXST bit in the SFMSDC register.

Note: Set the SFMXD[7:0] bits in the SFMSDC register to the XIP mode setting data specified for the actual serial flash device. The XIP mode of the serial flash controller is only enabled in the SFMXEN bit, regardless of the SFMXD[7:0] setting in the SFMSDC register.

39.8.2 Releasing the XIP Mode

To release the XIP mode, specify the release configuration for the serial flash in the SFMXD[7:0] bits in the SFMSDC register, and clear the SFMXEN bit to 0. In the dummy cycle of the next Fast Read instruction, the mode data specified in the SFMXD[7:0] bits is transferred to the serial flash during the first two-cycle period. From that point, the XIP mode is disabled in both the QSPI and the serial flash device. To confirm completion of the XIP mode release procedure, read 0 from the SFMXST bit in the SFMSDC register

Note: Set the SFMXD[7:0] bits in the SFMSDC register to the XIP mode setting data specified for the actual serial flash device. The XIP mode of the serial flash controller is only disabled in the SFMXEN bit, regardless of the SFMXD[7:0] setting in the SFMSDC register.

39.9 QIO2 and QIO3 Pin States

The QIO2 and QIO3 pin states depend on the serial interface read mode specified in the SFMRM[2:0] bits in the SFMSMD register.

Table 39.9 QIO2 and QIO3 pin states

SFMSMD.SFMRM[2:0] bits	QIO2 pin state[1]	QIO3 pin state[2]	Remarks
111	Setting prohibited		
110			
101	Input or output as a serial data signal (standby level is Hi-Z)	Input or output as serial data signal (standby level is Hi-Z)	Fast Read Quad I/O
100			Fast Read Quad Output
011	Output SFMWPL bit variable of SFMPMD register (initial output variable is low level)	Output high level	Fast Read Dual I/O
010			Fast Read Dual Output
001			Fast Read
000			Read (Initial State)

Note 1. The serial flash can also use the QIO2 pin for the WP function.
Note 2. The serial flash can also use the QIO3 pin for the HOLD or RESET function.

39.10 Direct Communication Mode

39.10.1 About Direct Communication

The QSPI can read the serial flash contents by automatically converting a ROM read bus cycle to an SPI bus cycle. However, serial flash devices have many different functions in addition to memory data read, including ID information read, erase, programming, and status information read. There is no standardized instruction set for using these functions, and more functions are being added rapidly by different vendors to different devices. It is difficult to support these functions by hardware control.

The QSPI flexibly supports these serial flash devices by providing a means for the software to directly communicate with the serial flash, so that the software can create any SPI bus cycle required.

39.10.2 Using Direct Communication Mode

To communicate directly with serial flash, transition to direct communication mode by setting the DCOM bit in the SFMCMD register to 1. While direct communication mode is selected, ordinary flash read operation is disabled. For ordinary flash access after direct communication, terminate direct communication mode by setting the DCOM bit in the SFMCMD register to 0.

Note: If the QSPI is set to the XIP mode, you must terminate the XIP mode before starting direct communication mode.

39.10.3 Generating the SPI Bus Cycle during Direct Communication

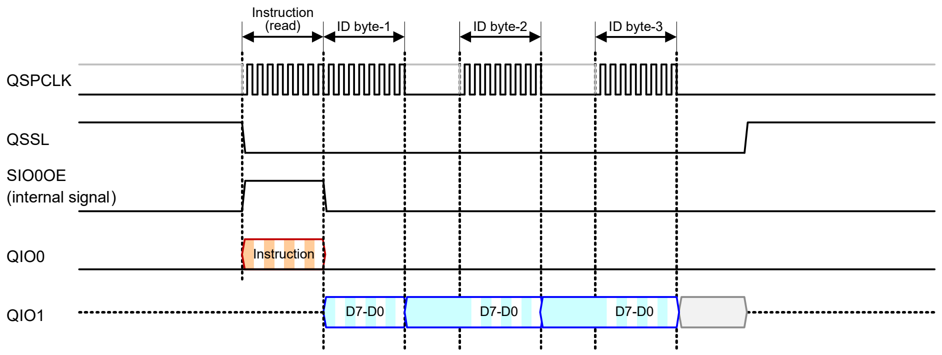
The SPI bus cycle in direct communication starts on the first access to the SFMCOM port and ends with a write to the SFMCMD register, after a series of I/O operations is performed through the SFMCOM port. At that point, a write to the SFMCOM port is converted to a one-byte transmission to the SPI bus, and a read from the SFMCOM port is converted to a one-byte reception from the SPI bus.

During the period from the first access to the SFMCOM port to the last write operation to the SFMCMD register, the serial flash select signal is held active to notify the serial flash that a series of SPI bus cycles is in progress.

Note: In direct communication mode, all writes to registers other than SFMCMD (including SFMSMD, SFMSSC, SFMSKC, SFMSST, SFMCST, SFMSIC, SFMSAC, SFMSDC, SFMSPC, and SFMPMD) are disabled. With this circuit configuration, writing to a register area other than the SFMCOM port terminates the SPI bus cycle. However, do not write to a register area other than SFMCMD as a way to terminate the SPI bus cycle. This operation is not guaranteed as a normal function.

The following is an example program for direct communication.

```
//### CAUTION! ### This code must be outside the serial flash that is going to be controlled.

// Define specific instruction codes of the target serial flash device.
#define  Instruction_FREAD  0x0B // Fast Read
#define  Instruction_RDSR   0x05 // Read Status register
#define  Instruction_RDID   0x9F // Read Identification
#define  Instruction_WREN   0x06 // Write Enable
#define  Instruction_CERA   0xC7 // Chip Erase

unsigned char  mfid, mtype, mcap, data, temp;

SFMCMD = 0x01; // Enable direct operation

// Get the device identification assigned by JEDEC.
SFMCOM = Instruction_RDID; // put “Read Identification” instruction (open SPI bus cycle)
mfid   = (unsigned char) SFMCOM; // get “Manufacturer Identification”
mtype  = (unsigned char) SFMCOM; // get “Memory Type”
mcap   = (unsigned char) SFMCOM; // get “Memory Capacity”
SFMCMD = 0x01h; // close SPI bus cycle

// Get one byte from the address 0x012345h.
SFMCOM = Instruction_FREAD; // put “Fast Read” instruction (open SPI bus cycle)
SFMCOM = 0x01; // put upper byte of the address 0x012345
SFMCOM = 0x23; // put middle byte of the target address 0x012345
SFMCOM = 0x45; // put lower byte of the target address 0x012345
temp   = (unsigned char) SFMCOM; // get one byte dummy code for FAST READ transaction
data   = (unsigned char) SFMCOM; // get the data
SFMCMD = 0x01; // close SPI bus cycle

// Erase All contents.
SFMCOM = Instruction_WREN; // put “Write Enable” instruction (open SPI bus cycle)
SFMCMD = 0x01; // close SPI bus cycle
SFMCOM = Instruction_CERA; // put “Chip Erase” instruction (open SPI bus cycle)
SFMCMD = 0x01; // close SPI bus cycle
SFMCOM = Instruction_RDSR; // put “Read Status Register” instruction (open SPI bus cycle)
while (SFMCOM & 0x01){}; // Polling “Write Progress Bit” until completion
SFMCMD = 0x01; // close SPI bus cycle

SFMCMD = 0x00; // Disable direct operation
```



```
(1)  SFMCMD = 01h;                    // Enable direct operation
                                      // Get the device identification assigned by JEDEC
(2)  SFMCOM = Instruction_RDID;       // Put “Read Identification” instruction (open the transaction)
(3)  mfid = (unsigned char) SFMCOM;   // Get ID byte-1 “Manufacturer Identification”
(4)  mtype = (unsigned char)          // Get ID byte-2 “Memory Type”
     SFMCOM;
(5)  mcap = (unsigned char) SFMCOM;   // Get ID byte-3 “Memory Capacity”
(6)  SFMCMD = 01h;                    // Close the transaction
(7)  SFMCMD = 00h;                    // Disable direct operation
```

Figure 39.34 Example of direct communication timing for ID read

Note: When Extended SPI protocol is used in direct communication mode, the standard Read or Fast Read instruction must be used to reference the contents of the serial flash. The QSPI does not support Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, or Fast Read Quad I/O transfers in this configuration. When these high-speed read operations are required, use ordinary flash access.

39.11 Operation

39.11.1 Procedure for Changing Settings in Multiple Control Registers

The settings of the QSPI control registers can be changed dynamically during system operation. However, when the settings of multiple control registers are changed sequentially, an SPI bus cycle might occur before all of the registers are updated. The register setting sequence must be carefully designed so that the SPI bus timing specification is satisfied at all stages of register setting changes.

```
//
// Making QSPCLK faster
//
SFMSMD = 0x0041; // SFMPAE: 0 SFMPFE: 1 SFMSE:00 SFMRM:01 (prefetch enable fast read)
SFMSSC = 0x04; // SFMSLD: 0 SFMSHD: 0 SFMSW:4 (minimum QSSL high width = 5 sck)
SFMSKC = 0x00; // SFMDTY: 0 SFMDV: 0 (1/2 mode)  ### switch clock speed last ###

//
// Making QSPCLK slower
//
SFMSKC = 0x06; // SFMDTY: 0 SFMDV:6 (1/8 mode)  ### switch clock speed first ###
SFMSSC = 0x01; // SFMSLD: 0 SFMSHD:0 SFMSW: 1 (minimum QSSL high width = 2 sck)
SFMSMD = 0x0040; // SFMPAE: 0 SFMPFE:1 SFMSE: 00 SFMRM:00 (prefetch enable, standard read)
```

39.12 Interrupts

When the EROMR bit in the SFMCST register sets to 1, the QSPI requests an interrupt. The EROMR bit sets to 1 when a ROM read access is detected in direct communication mode. Interrupt requests are retained until the EROMR bit is cleared by a 0 write. For details, see section 14, Interrupt Controller Unit (ICU).

39.13 Usage Notes

39.13.1 Settings for the Module-Stop Function

QSPI operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The QSPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

40. Cyclic Redundancy Check (CRC) Calculator

40.1 Overview

The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB- or MSB-first communication. Additionally, various CRC generation polynomials are available for your application. The snoop function allows monitoring of reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer.

Table 40.1 lists the CRC calculator specifications and Figure 40.1 shows a block diagram.

Table 40.1 CRC calculator specifications

Parameter	Specifications for 8-bit data	Specifications for 32-bit data
Data size	8-bit	32-bit
Data for CRC calculation[*1]	CRC code generated for data in 8n-bit units (where n is a whole number)	CRC code generated for data in 32n-bit units (where n is a whole number)
CRC processor unit	Operation executed on 8 bits in parallel	Operation executed on 32 bits in parallel
CRC generating polynomial	One of three generating polynomials selectable [8-bit CRC] • $X^8 + X^2 + X + 1$ (CRC-8) [16-bit CRC] • $X^{16} + X^{15} + X^2 + 1$ (CRC-16) • $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT)	One of two generating polynomials selectable [32-bit CRC] • $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) • $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C)
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB- or MSB-first communication	
Module-stop function	Module-stop state can be set to reduce power consumption	
CRC snoop	Monitor reads from and writes to a certain register address	—

Note 1. The circuit cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

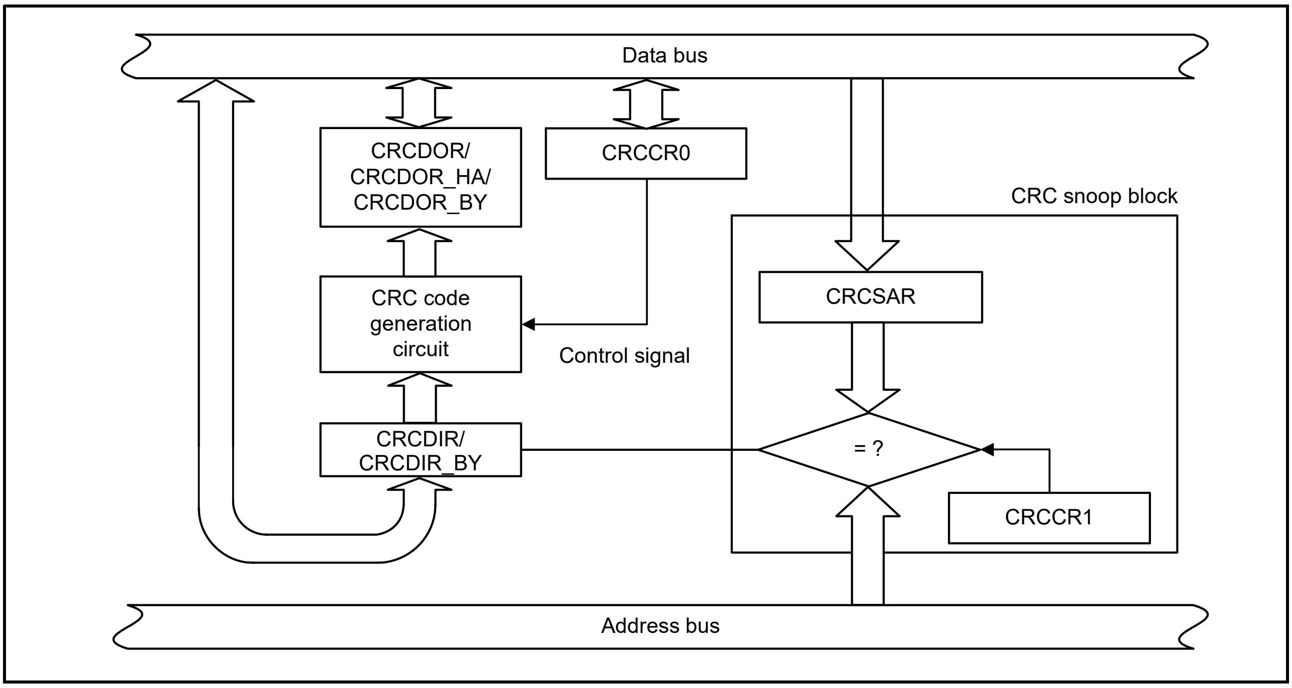


Figure 40.1 CRC calculator block diagram

40.2 Register Descriptions

40.2.1 CRC Control Register 0 (CRCCR0)

Address(es): CRC.CRCCR0 4007 4000h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	GPS[2:0]	CRC Generating Polynomial Switching	b2 b0 0 0 0: Do not calculate 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Other: Do not calculate.	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LMS	CRC Calculation Switching	0: Generate CRC for LSB-first communication 1: Generate CRC for MSB-first communication.	R/W
b7	DORCLR	CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear	1: Clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register. This bit is read as 0.	W*1

Note 1. This bit must always be set to 1 when writing to this register.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY)

Write 1 to the DORCLR bit to clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register to 0000_0000h. This bit is read as 0. Only 1 can be written.

LMS bit (CRC Calculation Switching)

The LMS bit selects the bit order of generated CRC code. Transmit the lower-order byte of the CRC code first for LSB-first communication and the higher-order byte first for MSB-first communication. For details on transmitting and receiving CRC code, see section 40.3, Operation.

GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

40.2.2 CRC Control Register 1 (CRCCR1)

Address(es): CRC.CRCCR1 4007 4001h

	b7	b6	b5	b4	b3	b2	b1	b0
	CRCSEN	CRCSWR	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CRCSWR	Snoop-On-Write/Read Switch	0: Snoop-on-read 1: Snoop-on-write.	R/W

Bit	Symbol	Bit name	Description	R/W
b7	CRCSEN	Snoop Enable	0: Disabled 1: Enabled.	R/W

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of the access in the address monitoring function. When the bit is set to 0 (initial value), the CRC snoop operation to read a specific register address is enabled. When the bit is set to 1, the CRC snoop operation to write to a specific register address is enabled.

CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, CRC snoop operation is enabled. When the bit is set to 0, CRC snoop operation is disabled.

40.2.3 CRC Data Input Register (CRCDIR/CRCDIR_BY)

Address(es): CRC.CRCDIR/CRCDIR_BY 4007 4004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCDIR is a 32-bit read/write register to write data to for CRC-32 or CRC-32C calculation. CRCDIR_BY is an 8-bit read/write register to write data to for CRC-8, CRC-16, or CRC-CCITT calculation.

40.2.4 CRC Data Output Register (CRCDOR/CRCDOR_HA/CRCDOR_BY)

Address(es): CRC.CRCDOR/CRCDOR_HA/CRCDOR_BY 4007 4008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCDOR is a 32-bit read/write register for CRC-32 or CRC-32C. CRCDOR_HA is a 16-bit read/write register for CRC-16 or CRC-CCITT. CRCDOR_BY is an 8-bit read/write register for CRC-8. Because its initial value is 0000_0000h, rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform the calculations using a value other than the initial value.

Data written to the CRCDIR/CRCDIR_BY register is CRC-calculated, and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following transferred data and the result is 0000_0000h, there is no CRC error.

When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in CRCDOR_BY.

When a 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$ or $X^{16} + X^{12} + X^5 + 1$ polynomial) is in use, the valid CRC code is obtained in CRCDOR_HA.

40.2.5 Snoop Address Register (CRCSAR)

Address(es): CRC.CRCSAR 4007 400Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CRCSA[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b13 to b0	CRCSA[13:0]	Register Snoop Address	These bits store the TDR or RDR address in the SCI module to snoop.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CRCSA[13:0] bits (Register Snoop Address)

The CRCSA[13:0] bits specify the lower 14 bits of the register address monitored by the CRC snoop operation. Only the following addresses can be used for the CRCSA[13:0] bits:

- 4007 0003h: SCI0.TDR, 4007 0005h: SCI0.RDR
- 4007 0023h: SCI1.TDR, 4007 0025h: SCI1.RDR
- 4007 0043h: SCI2.TDR, 4007 0045h: SCI2.RDR
- 4007 0063h: SCI3.TDR, 4007 0065h: SCI3.RDR
- 4007 0083h: SCI4.TDR, 4007 0085h: SCI4.RDR
- 4007 00A3h: SCI5.TDR, 4007 00A5h: SCI5.RDR
- 4007 00C3h: SCI6.TDR, 4007 00C5h: SCI6.RDR
- 4007 00E3h: SCI7.TDR, 4007 00E5h: SCI7.RDR
- 4007 0103h: SCI8.TDR, 4007 0105h: SCI8.RDR
- 4007 0123h: SCI9.TDR, 4007 0125h: SCI9.RDR
- 4007 000Fh: SCI0.FTDRL, 4007 0011h: SCI0.FRDRL
- 4007 002Fh: SCI1.FTDRL, 4007 0031h: SCI1.FRDRL
- 4007 004Fh: SCI2.FTDRL, 4007 0051h: SCI2.FRDRL
- 4007 006Fh: SCI3.FTDRL, 4007 0071h: SCI3.FRDRL
- 4007 008Fh: SCI4.FTDRL, 4007 0091h: SCI4.FRDRL
- 4007 00AFh: SCI5.FTDRL, 4007 00B1h: SCI5.FRDRL
- 4007 00CFh: SCI6.FTDRL, 4007 00D1h: SCI6.FRDRL
- 4007 00EFh: SCI7.FTDRL, 4007 00F1h: SCI7.FRDRL
- 4007 010Fh: SCI8.FTDRL, 4007 0111h: SCI8.FRDRL
- 4007 012Fh: SCI9.FTDRL, 4007 0131h: SCI9.FRDRL.

40.3 Operation

40.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB- or MSB-first transfers.

The following examples illustrate CRC code generation for input data (F0h) using the 16-bit CRC-CCITT-generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in CRCDOR_BY. When an 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

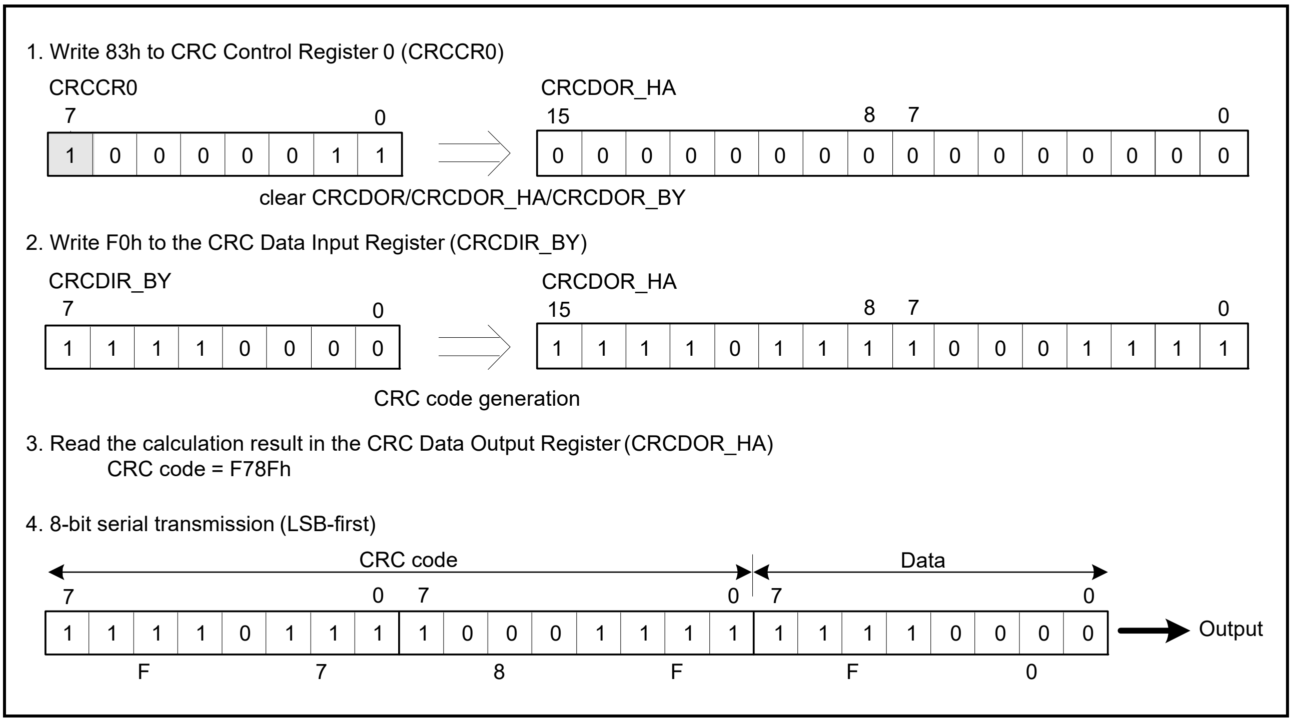


Figure 40.2 LSB-first data transmission

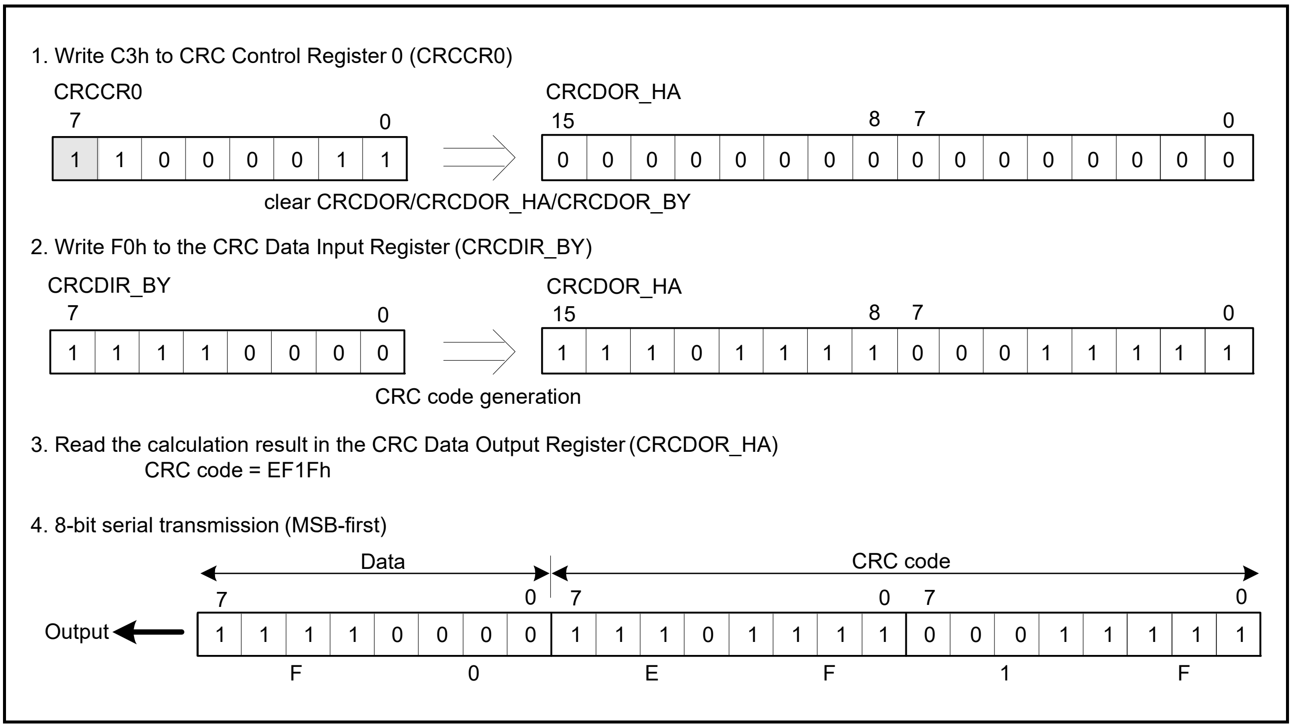


Figure 40.3 MSB-first data transmission

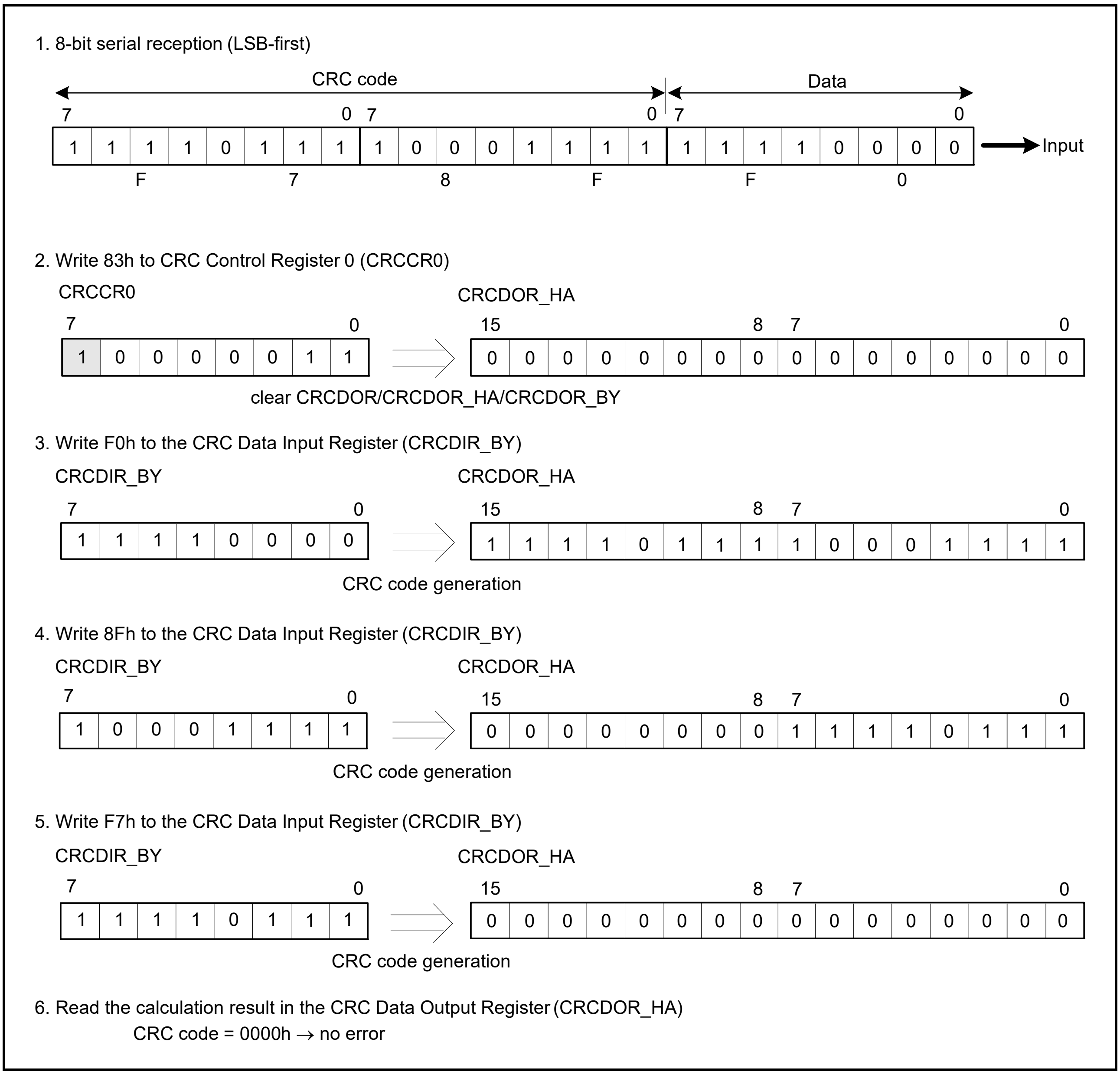


Figure 40.4 LSB-first data reception

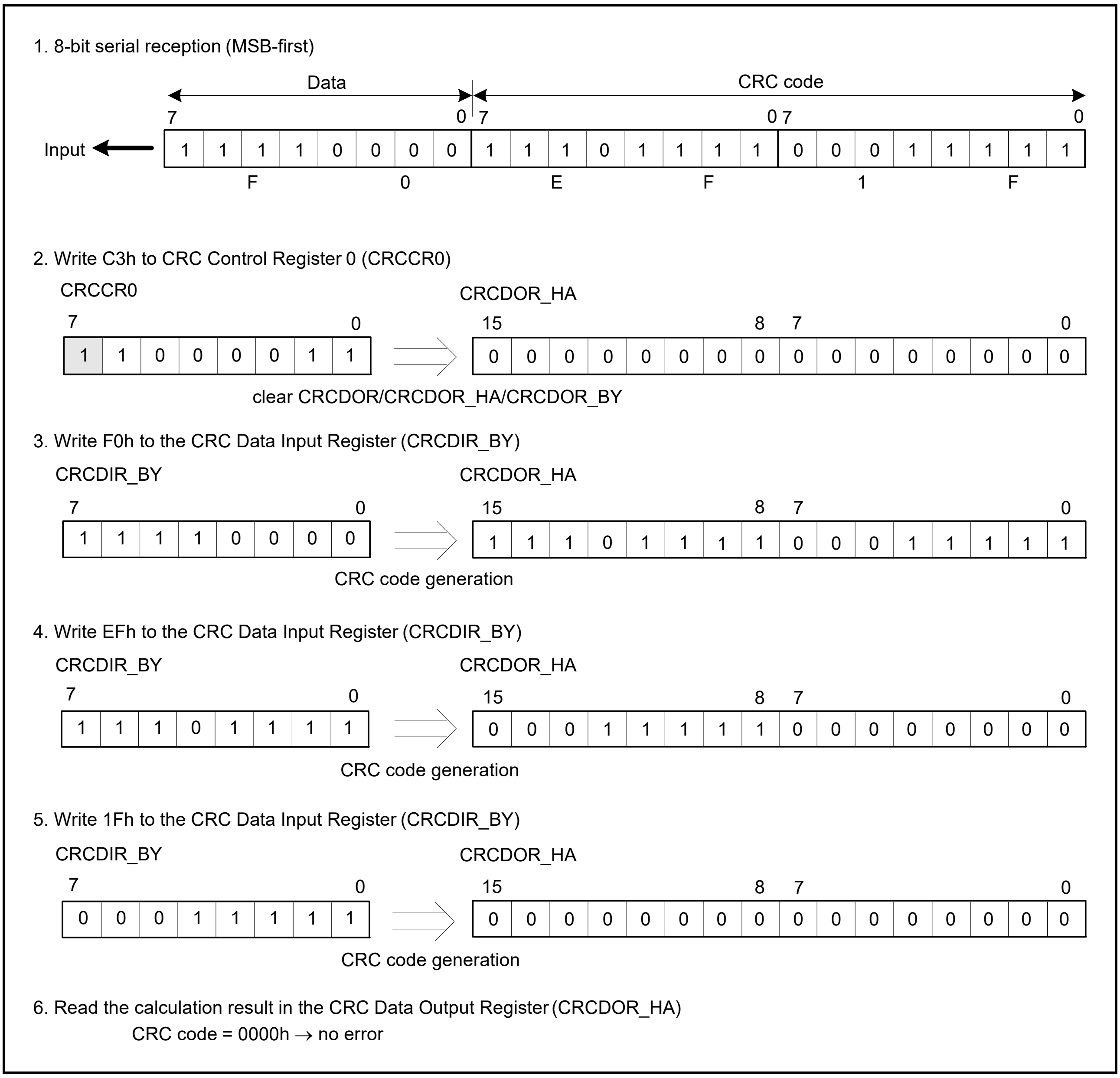


Figure 40.5 MSB-first data reception

40.3.2 CRC Snoop

The CRC snoop function monitors reads from and writes to a specified I/O register address and performs CRC calculation on the data read from and written to the register address automatically. Because the CRC snoop recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, writing data to the CRCDIR_BY register is not required. All I/O register addresses specified in section 40.2.5, Snoop Address Register (CRCSAR) are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the serial transmit buffer, and reads from the serial receive buffer.

To use this function, write a target I/O register address to the CRCSA13 to CRCSA0 bits in the CRCSAR register, and set the CRCSEN bit in the CRCCR1 register to 1. Then set the CRCSWR bit in the CRCCR1 register to 1 to enable snooping on writes to the target address, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target address.

When both the CRCSEN and CRCSWR bits are set to 1 and data is written to a target I/O register address in a bus master module (including the CPU, DMAC, and DTC), the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculation. Similarly, when the CRCSEN bit is set to 1 and the CRCSWR bit to 0, and data is read from a target I/O register address in a bus master module (including the CPU, DMAC, and DTC), the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculation.

CRC calculation is performed 1 byte at a time. When the target I/O register address is accessed in words (16 bits) or long words (32 bits), CRC code is generated on the lower byte (1 byte) of data.

40.4 Usage Notes

40.4.1 Settings for the Module-Stop Function

CRC calculator operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CRC calculator is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

40.4.2 Note on Transmission

The transmission sequence for the CRC code differs depending on whether transmission is LSB-first or MSB-first.

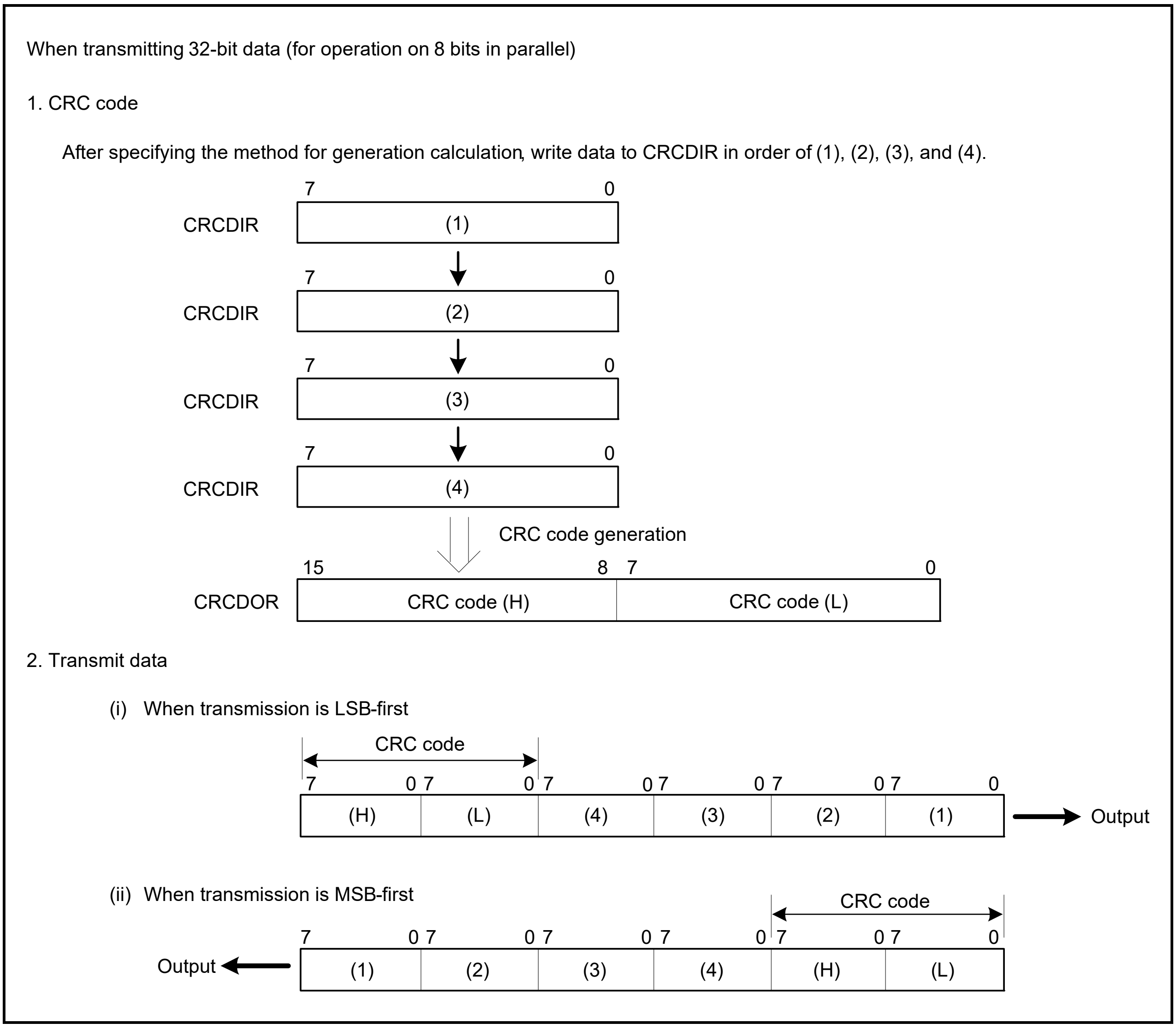


Figure 40.6 LSB-first and MSB-first data transmission

41. Serial Sound Interface Enhanced (SSIE)

41.1 Overview

The Serial Sound Interface Enhanced (SSIE) can transmit and receive audio data to and from various devices that support any of audio data formats, such as I^2S, monaural, and TDM.

41.2 Features

Table 41.1 Features of SSIE

Parameter		Description
Number of channels		Two channels, SSIE0 and SSIE1
Communication mode		• Master/slave • Transmission/reception(SSIE0 full-duplex communication) • Transmission/reception(SSIE1 half-duplex communication)
Communication format		• I^2S format • Monaural format • TDM format
Serial data		• MSB first • Data can be left-justified or right-justified. • Data delay (1 clock cycle) or no delay selectable for the period from SSILRCK/SSIFS to SSITXD0/SSIRXD0/SSIDATA1 • System word length: 8, 16, 24, 32, 48, 64, 128, or 256 bits • Data word length: 8, 16, 18, 20, 22, 24, or 32 bits • Padding polarity: Low or high
Bit clock (SSIBCK)	In master mode	• Two clock sources available (AUDIO_CLK/GPT output (GTIOC1A)) • Clock source division ratio: 1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, and 1/128. • Supply/stop is selectable while communication is halted.
	In master/slave mode	• Polarity (rising edge or falling edge) selectable
LR clock/frame synchronization (SSILRCK/SSIFS)	In master mode	• Polarity (low level or high level) selectable • Supply/stop is selectable while communication is halted.
Transmit data (SSITXD0/SSIDATA1) and receive data (SSIRXD0/SSIDATA1)	Transmission	• Muting method (transmission of transmit FIFO data or transmission of data fixed to 0) selectable
FIFO	Capacity	• Transmit FIFO/receive FIFO: 4 bytes × 32 stages
	Data alignment	• Data alignment method (left-justification or right-justification) selectable for the data transfer between FIFO and shift register
Interrupt	Interrupt output	• Communication error/idle mode • Receive data full • Transmit data empty
Low power consumption function		• Whether to supply the audio clock selectable in master mode
Module stop function		• Module stop state can be set to reduce power consumption.

The following table lists and defines the terms used for the communication formats SSIE can use:

Table 41.2 Definition of terms

Term	Definition
Start trigger	First edge of the signal on the SSILRCK/SSIFS pin when the signal is set to the value specified in LRCKP to enable communication
Frame boundary	Point where SSIE starts transferring the first data of a frame or the point where SSIE ends transferring the last data of the frame

Table 41.2 Definition of terms

Term	Definition
Frame word number	Number of sound channels per frame
System word length	Number of bits per channel
Data word length	Number of significant bits per channel
Control bits for communication formats	• SSICR register: FRM, DWL, SWL, LRCKP, SPDP, SDTA, PDTA, and DEL bits • SSIFCR register: BSW bit • SSIOFR register: OMOD bit • SSISCR register: TDES and RDFS bits

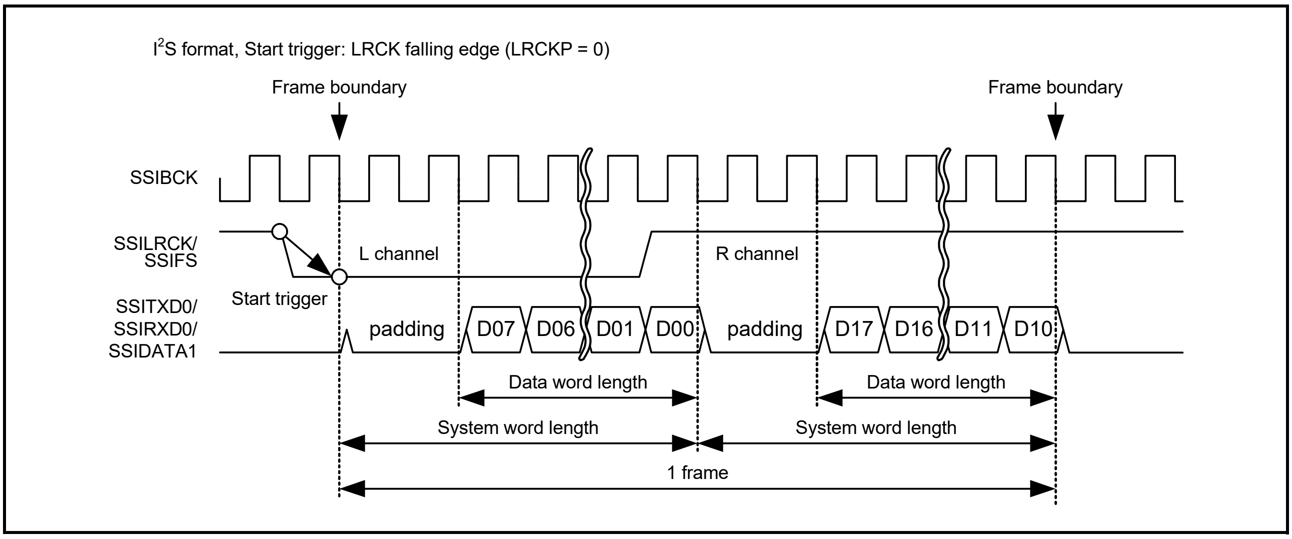
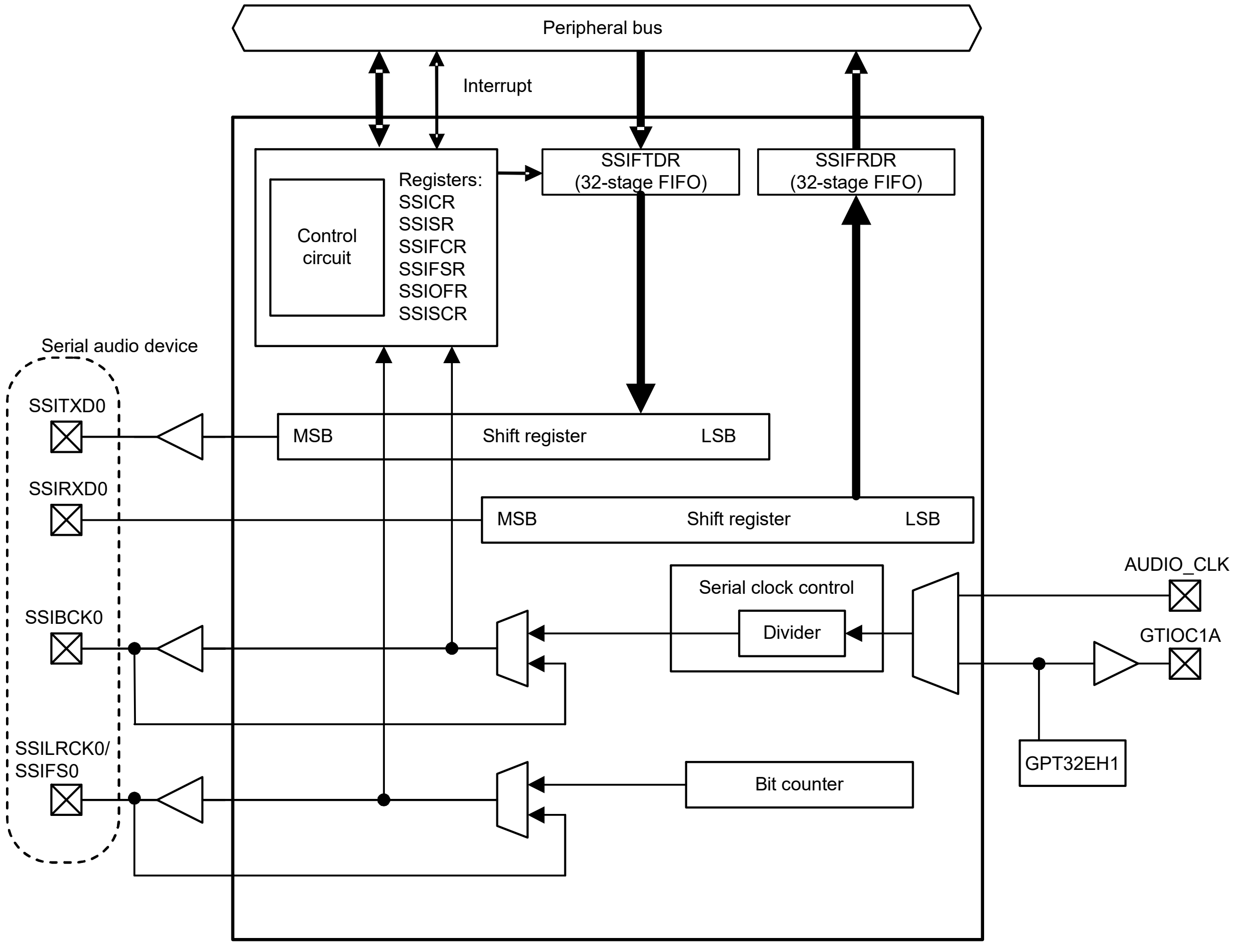


Figure 41.1 Definition of communication format

41.3 Block Diagram

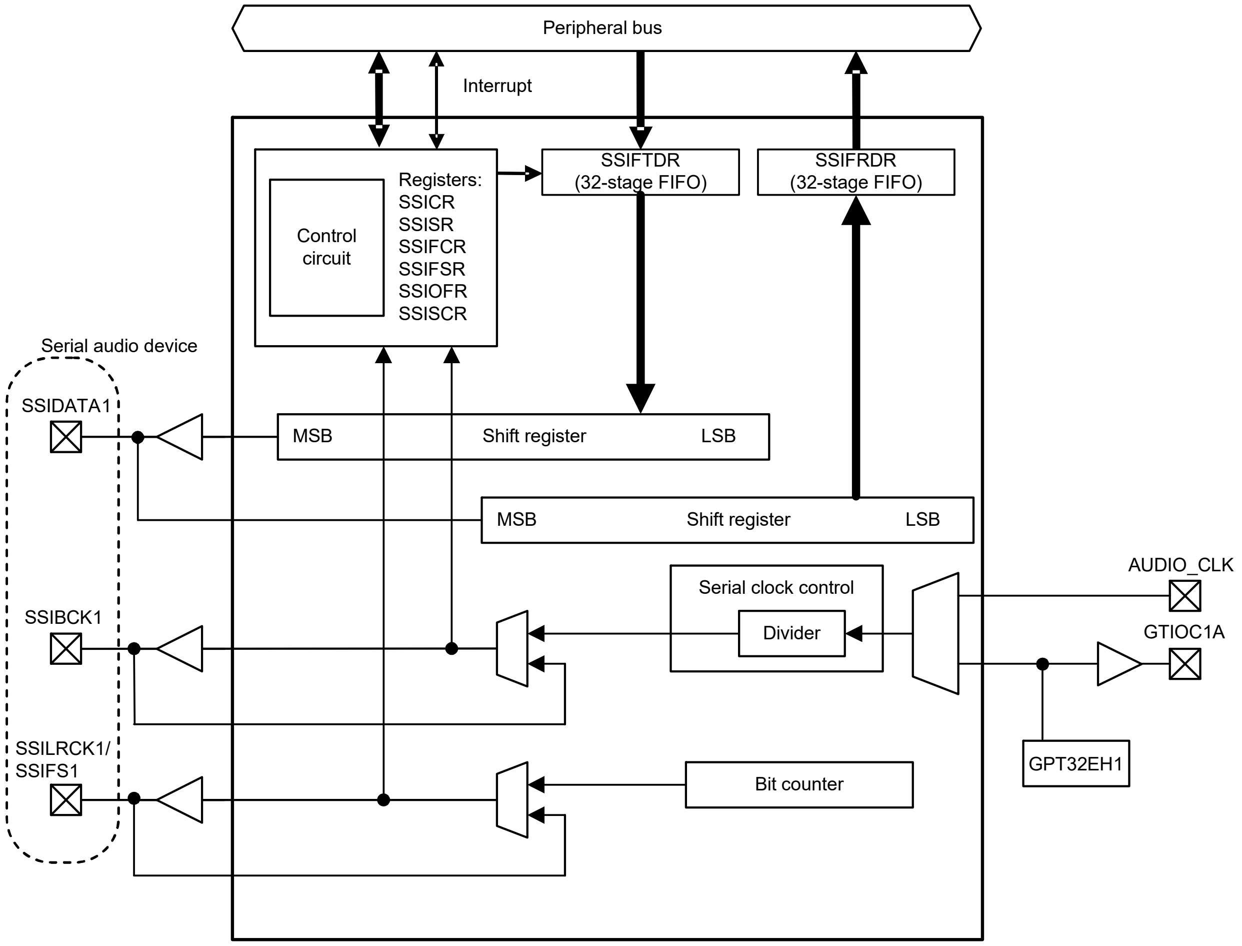
Figure 41.2 and Figure 41.3 show a block diagram of SSIE.



[Legend]

SSICR:	Control register
SSISR:	Status register
SSIFCR:	FIFO control register
SSIFSR:	FIFO status register
SSIOFR:	Audio format register
SSIFTDR:	Transmit FIFO data register
SSIFRDR:	Receive FIFO data register
SSISCR:	Status control register

Figure 41.2 SSIE block diagram (SSIE0)

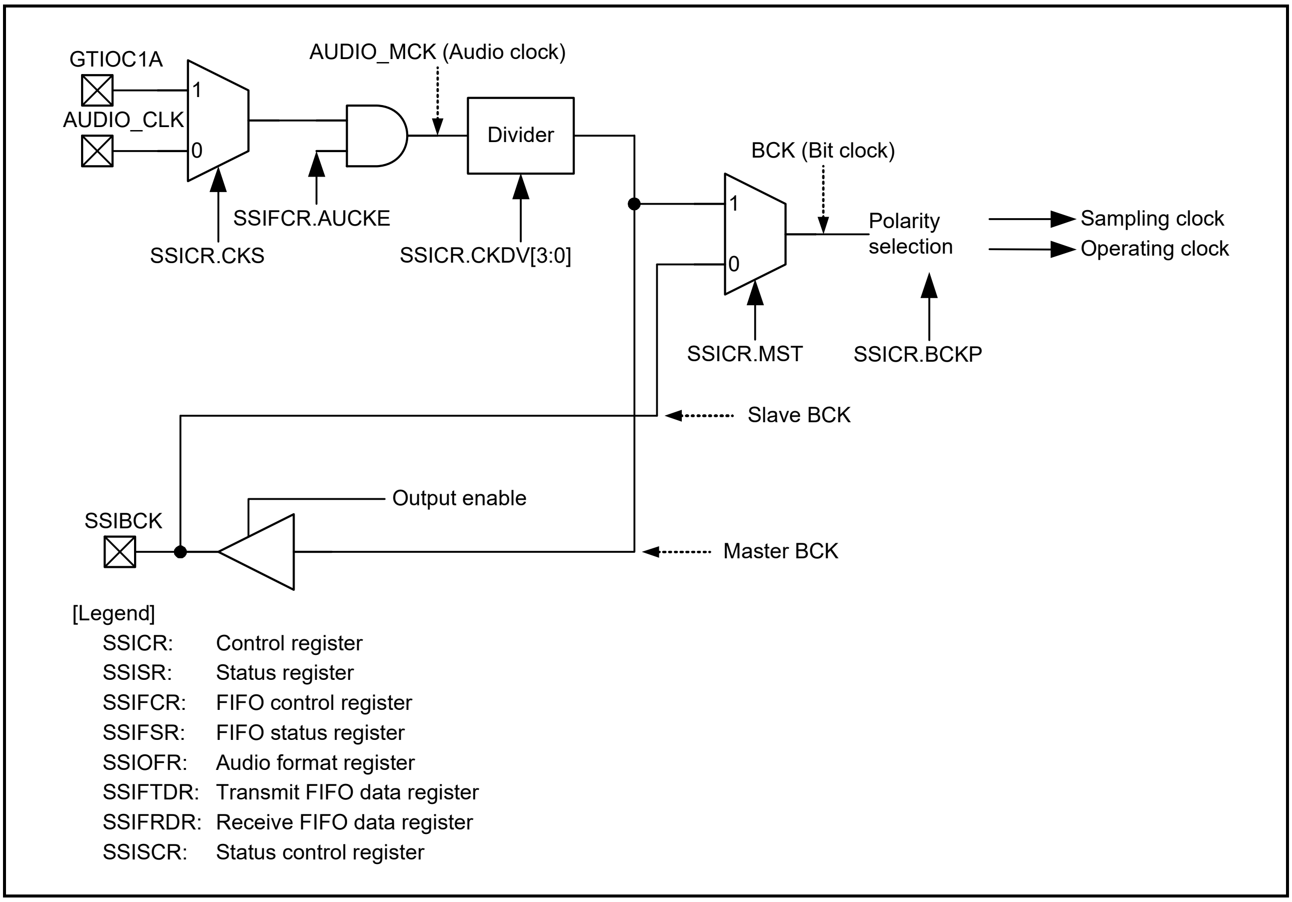


[Legend]

SSICR:	Control Register
SSISR:	Status Register
SSIFCR:	FIFO Control Register
SSIFSR:	FIFO Status Register
SSIOFR:	Audio Format register
SSIFTDR:	Transmit FIFO Data Register
SSIFRDR:	Receive FIFO Data Register
SSISCR:	Status Control register

Figure 41.3 SSIE block diagram (SSIE1)

Figure 41.4 shows the clock configuration of SSIE.



[Legend]

SSICR:	Control register
SSISR:	Status register
SSIFCR:	FIFO control register
SSIFSR:	FIFO status register
SSIOFR:	Audio format register
SSIFTDR:	Transmit FIFO data register
SSIFRDR:	Receive FIFO data register
SSISCR:	Status control register

Figure 41.4 SSIE clock configuration

41.4 Register Descriptions

41.4.1 Control Register (SSICR)

Address(es): SSIE0.SSICR 4004 E000h, SSIE1.SSICR 4004 E100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0 b1	REN TEN	Transmission and Reception Enable*2	00: Disables transmission and reception 01: Enables reception (starts reception) 10: Enables transmission (starts transmission) 11: Enables transmission and reception (starts transmission and reception).	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	MUEN	Mute Enable	0: Disables muting on the next frame boundary 1: Enables muting on the next frame boundary.	R/W

Bit	Symbol	Bit name	Description	R/W
b7 to b4	CKDV[3:0]	Selects Bit Clock Division Ratio*1	b7 b4 0 0 0 0: AUDIO_MCK 0 0 0 1: AUDIO_MCK/2 0 0 1 0: AUDIO_MCK/4 0 0 1 1: AUDIO_MCK/8 0 1 0 0: AUDIO_MCK/16 0 1 0 1: AUDIO_MCK/32 0 1 1 0: AUDIO_MCK/64 0 1 1 1: AUDIO_MCK/128 1 0 0 0: AUDIO_MCK/6 1 0 0 1: AUDIO_MCK/12 1 0 1 0: AUDIO_MCK/24 1 0 1 1: AUDIO_MCK/48 1 1 0 0: AUDIO_MCK/96 1 1 0 1: Setting prohibited 1 1 1 0: Setting prohibited 1 1 1 1: Setting prohibited.	R/W
b8	DEL	Selects Serial Data Delay*1	0: Delay of 1 cycle of SSIBCK between SSILRCK/SSIFS and SSITXD0/SSIRXD0/SSIDATA1 1: No delay between SSILRCK/SSIFS and SSITXD0/SSIRXD0/SSIDATA1 In the monaural format, this bit controls the waveform of SSILRCK/SSIFS. For details, see section 41.5.2, Monaural Format.	R/W
b9	PDTA	Selects Placement Data Alignment*1	0: Left-justifies placement data (SSIFTDR, SSIFRDR) 1: Right-justifies placement data (SSIFTDR, SSIFRDR).	R/W
b10	SDTA	Selects Serial Data Alignment *1	0: Transmits and receives serial data first and then padding bits 1: Transmit and receives padding bits first and then serial data.	R/W
b11	SPDP	Selects Serial Padding Polarity*1	0: Padding data is at a low level 1: Padding data is at a high level.	R/W
b12	LRCKP	Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal*1	0: The initial value is at a high level The start trigger for a frame is synchronized with a falling edge of SSILRCK/SSIFS 1: The initial value is at a low level The start trigger for a frame is synchronized with a rising edge of SSILRCK/SSIFS.	R/W
b13	BCKP	Selects Bit Clock Polarity*1	0: SSILRCK/SSIFS and SSITXD0/SSIRXD0/SSIDATA1 change at a falling edge (SSILRCK/SSIFS and SSIRXD0/SSIDATA1 are sampled at a rising edge of SSIBCK) 1: SSILRCK/SSIFS and SSITXD0/SSIRXD0/SSIDATA1 change at a rising edge (SSILRCK/SSIFS and SSIRXD0/SSIDATA1 are sampled at a falling edge of SSIBCK).	R/W
b14	MST	Master Enable*1	0: Slave-mode communication 1: Master-mode communication.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	SWL[2:0]	Selects System Word Length *1	b18 b16 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 24 bits 0 1 1: 32 bits 1 0 0: 48 bits 1 0 1: 64 bits 1 1 0: 128 bits 1 1 1: 256 bits.	R/W
b21 to b19	DWL[2:0]	Selects Data Word Length*1	b21 b19 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits 1 1 0: 32 bits 1 1 1: Setting prohibited.	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Bit name</th><th>Description</th><th>R/W</th></tr>
<tr><td>b23, b22</td><td>FRM[1:0]</td><td>Selects Frame Word Number *1</td><td>
<table>
<tr><td></td><td colspan="3">Communication format (SSIOFR.OMOD[1:0])</td></tr>
<tr><td>FRM[1:0]</td><td>I^2S (00b)</td><td>Monaural (10b)</td><td>TDM (01b)</td></tr>
<tr><td>00b</td><td>2</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>01b</td><td rowspan="3">Setting prohibited</td><td rowspan="3">Setting prohibited</td><td>4</td></tr>
<tr><td>10b</td><td>6</td></tr>
<tr><td>11B</td><td>8</td></tr>
</table>
</td><td>R/W</td></tr>
<tr><td>b24</td><td>—</td><td>Reserved</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>b25</td><td>IIEN</td><td>Idle Mode Interrupt Output Enable</td><td>0: Disables idle mode interrupt output
1: Enables idle mode interrupt output.</td><td>R/W</td></tr>
<tr><td>b26</td><td>ROIEN</td><td>Receive Overflow Interrupt Output Enable</td><td>0: Disables receive overflow interrupt output
1: Enables receive overflow interrupt output.</td><td>R/W</td></tr>
<tr><td>b27</td><td>RUIEN</td><td>Receive Underflow Interrupt Output Enable</td><td>0: Disables receive underflow interrupt output
1: Enables receive underflow interrupt output.</td><td>R/W</td></tr>
<tr><td>b28</td><td>TOIEN</td><td>Transmit Overflow Interrupt Output Enable</td><td>0: Disables transmit overflow interrupt output
1: Enables transmit overflow interrupt output.</td><td>R/W</td></tr>
<tr><td>b29</td><td>TUIEN</td><td>Transmit Underflow Interrupt Output Enable</td><td>0: Disables transmit underflow interrupt output
1: Enables transmit underflow interrupt output.</td><td>R/W</td></tr>
<tr><td>b30</td><td>CKS</td><td>Selects an Audio Clock for Master-mode Communication *1</td><td>0: Selects the AUDIO_CLK input
1: Selects the GTIOC1A (GPT output).</td><td>R/W</td></tr>
<tr><td>b31</td><td>—</td><td>Reserved</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
</table>

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

Note 2. If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

With this register, select an audio clock, control interrupt requests, select data formats, and set an operation mode.

TEN and REN bits (Transmission and Reception Enable)

These bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCK/SSIFS signal. For details, see section 41.8.2 to section 41.8.4. When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

If you want to stop SSIE before a frame boundary is reached, perform a software reset procedure.

MUEN bit (Mute Enable)

This bit sets/clears the mute function for the data output from the SSITXD0/SSIDATA1 pin. When this bit is set to 1 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to 0 at the next frame boundary. When this bit is set to 0 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to the data of transmit FIFO data register at the next frame boundary. Note that this bit controls data only. Status flags and interrupt signals are normally generated.

Changing the value of this bit must be performed only after setting the communication format to be used.

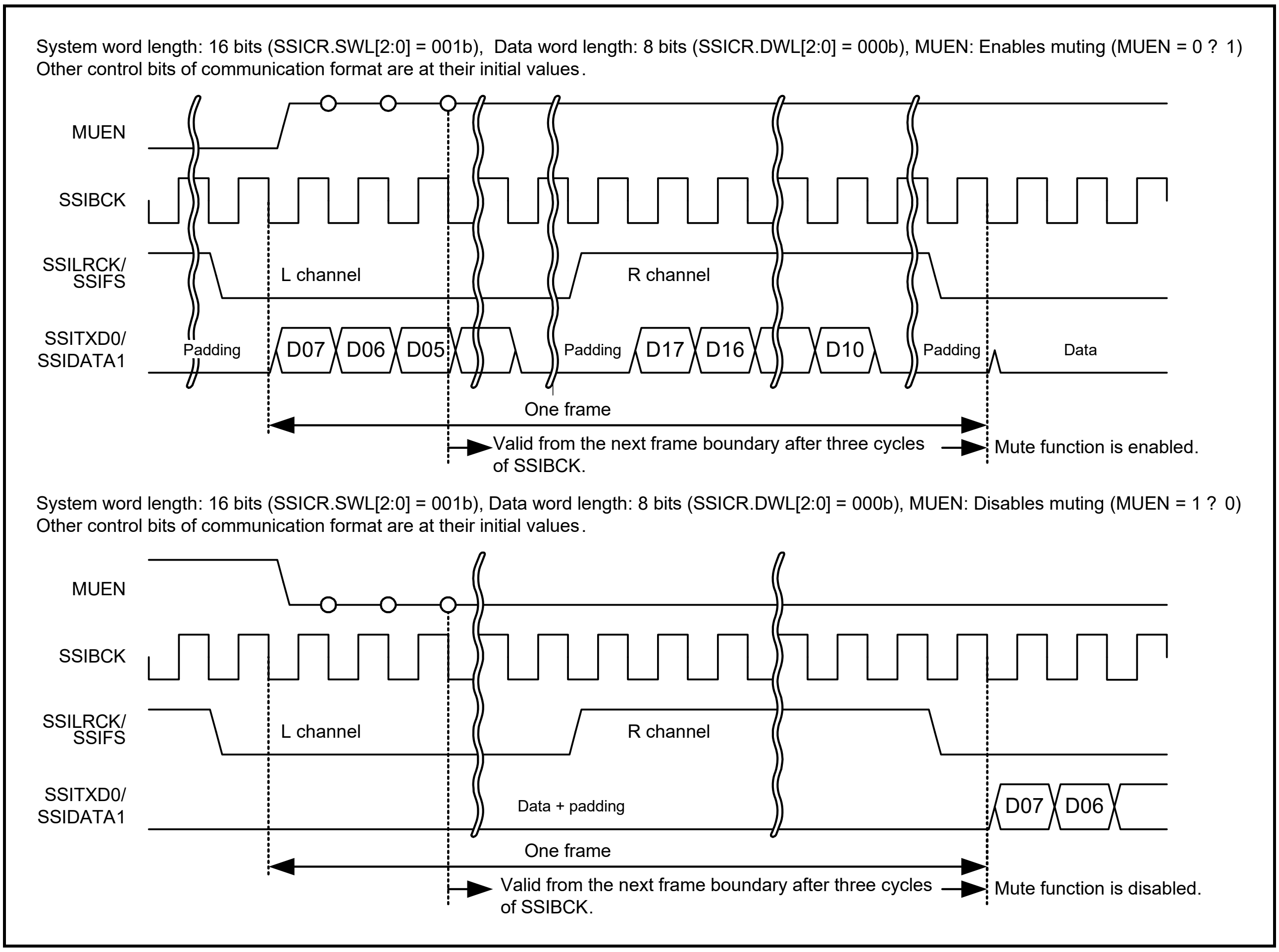


Figure 41.5 Transmit data with the mute function set

CKDV[3:0] bits (Selects Bit Clock Division Ratio)

These bits set the division ratio of the bit clock based on AUDIO_MCK in master-mode communication (MST = 1). In slave-mode communication (MST = 0), setting of these bits are invalid.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in SSIFCR.

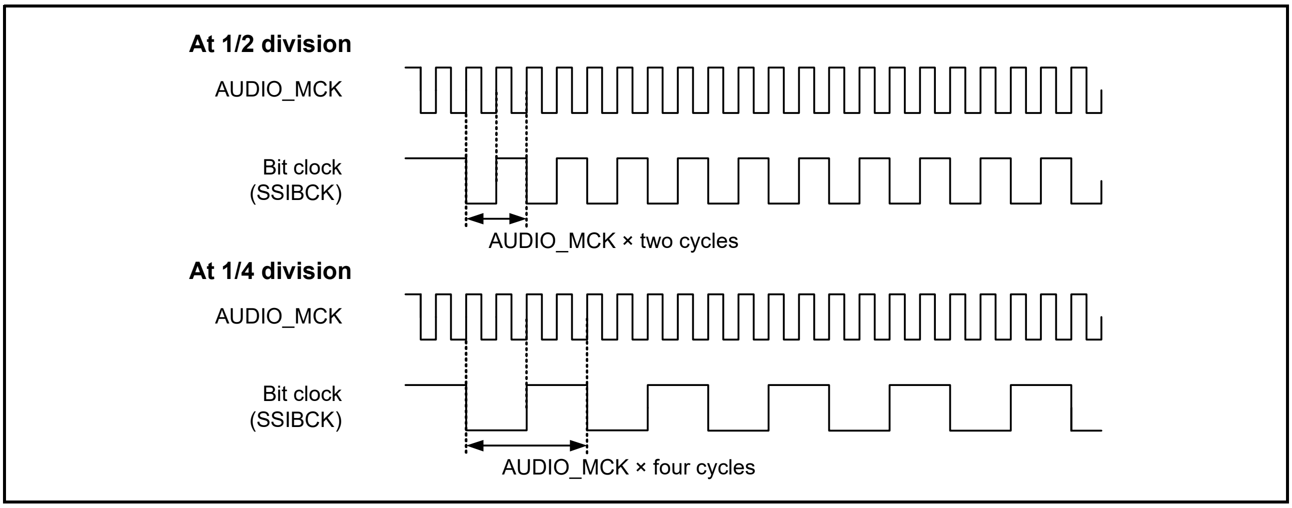


Figure 41.6 Sampling frequencies in master-mode communication

DEL bit (Selects Serial Data Delay)

This bit sets whether or not there will be a delay between SSILRCK/SSIFS and SSITXD0/SSIRXD0/SSIDATA1.

For the I^{2}S or TDM format, set the DEL bit to 0. When the monaural format is used, setting of this bit changes the high period width of SSILRCK/SSIFS. For details, see section 41.5.2, Monaural Format. When using a compatible communication format, specify a setting of this bit that enables communication.

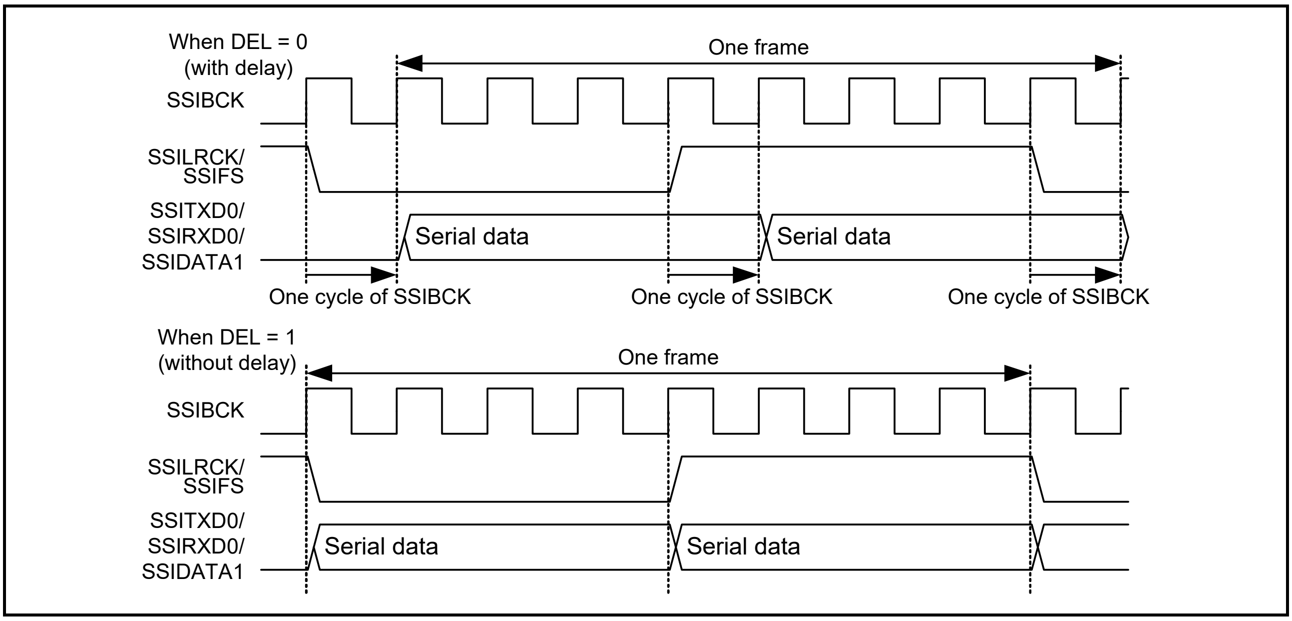
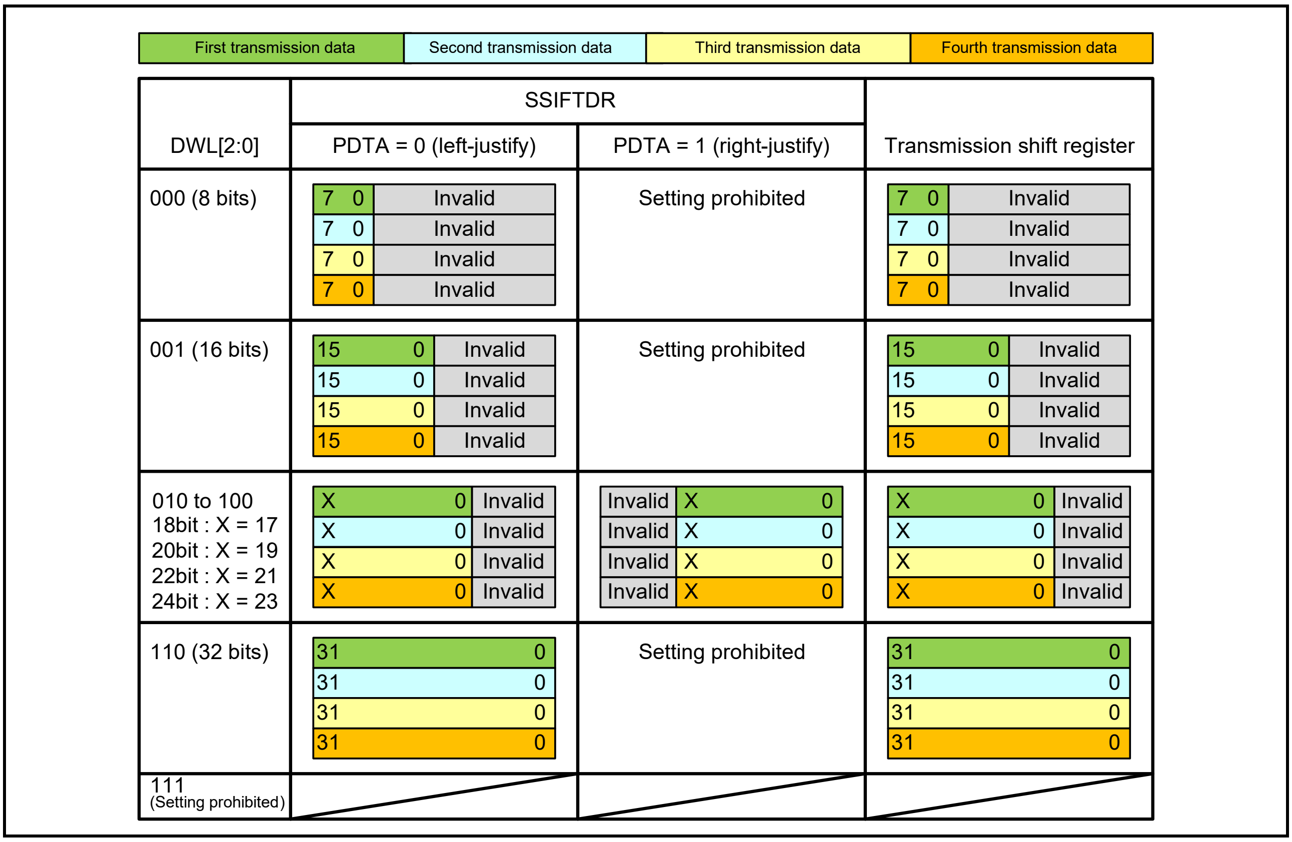


Figure 41.7 Setting of delay in serial data

PDTA bit (Selects Placement Data Alignment)

This bit sets how to align placement data. With the setting of data word length as 32 bits (SSICR.DWL[2:0] = 110b), this bit is invalid.

At transmission, see Figure 41.8.

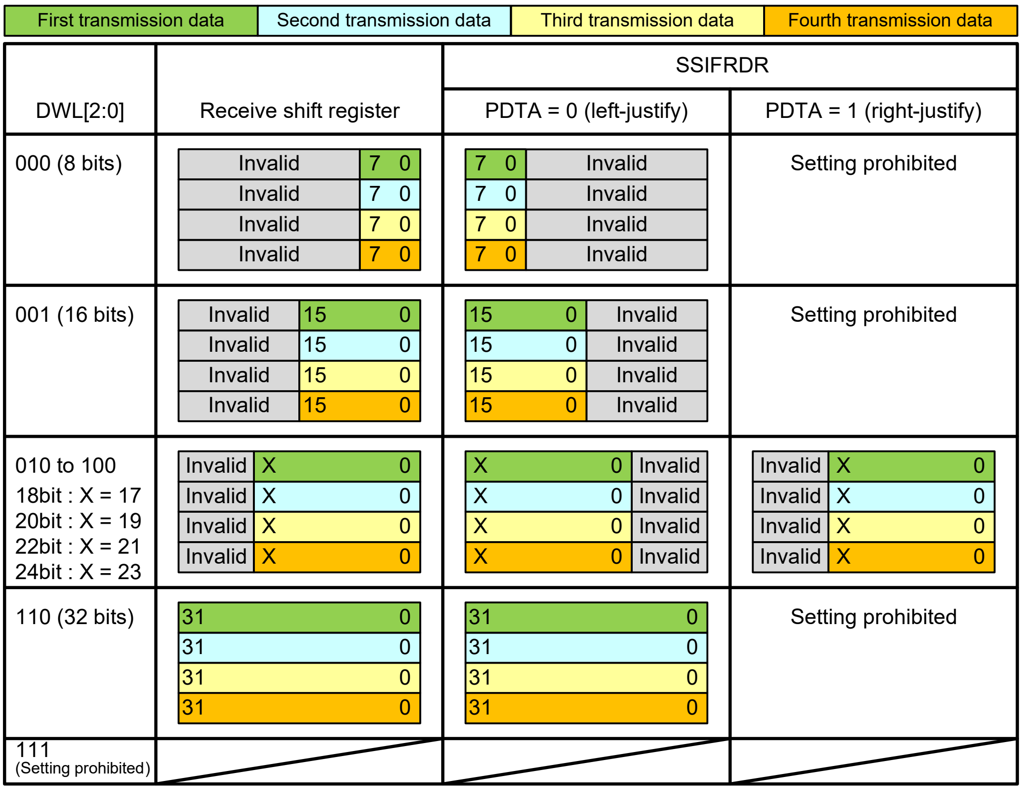


🟩 First transmission data 🟦 Second transmission data 🟨 Third transmission data 🟧 Fourth transmission data

DWL[2:0]	SSIFTDR PDTA = 0 (left-justify)	SSIFTDR PDTA = 1 (right-justify)	Transmission shift register
000 (8 bits)	7 0 Invalid 7 0 Invalid 7 0 Invalid 7 0 Invalid	Setting prohibited	7 0 Invalid 7 0 Invalid 7 0 Invalid 7 0 Invalid
001 (16 bits)	15 0 Invalid 15 0 Invalid 15 0 Invalid 15 0 Invalid	Setting prohibited	15 0 Invalid 15 0 Invalid 15 0 Invalid 15 0 Invalid
010 to 100 18bit : X = 17 20bit : X = 19 22bit : X = 21 24bit : X = 23	X 0 Invalid X 0 Invalid X 0 Invalid X 0 Invalid	Invalid X 0 Invalid X 0 Invalid X 0 Invalid X 0	X 0 Invalid X 0 Invalid X 0 Invalid X 0 Invalid
110 (32 bits)	31 0 31 0 31 0 31 0	Setting prohibited	31 0 31 0 31 0 31 0
111 (Setting prohibited)			

Figure 41.8 Alignment of placement data at transmission

At reception, see Figure 41.9.



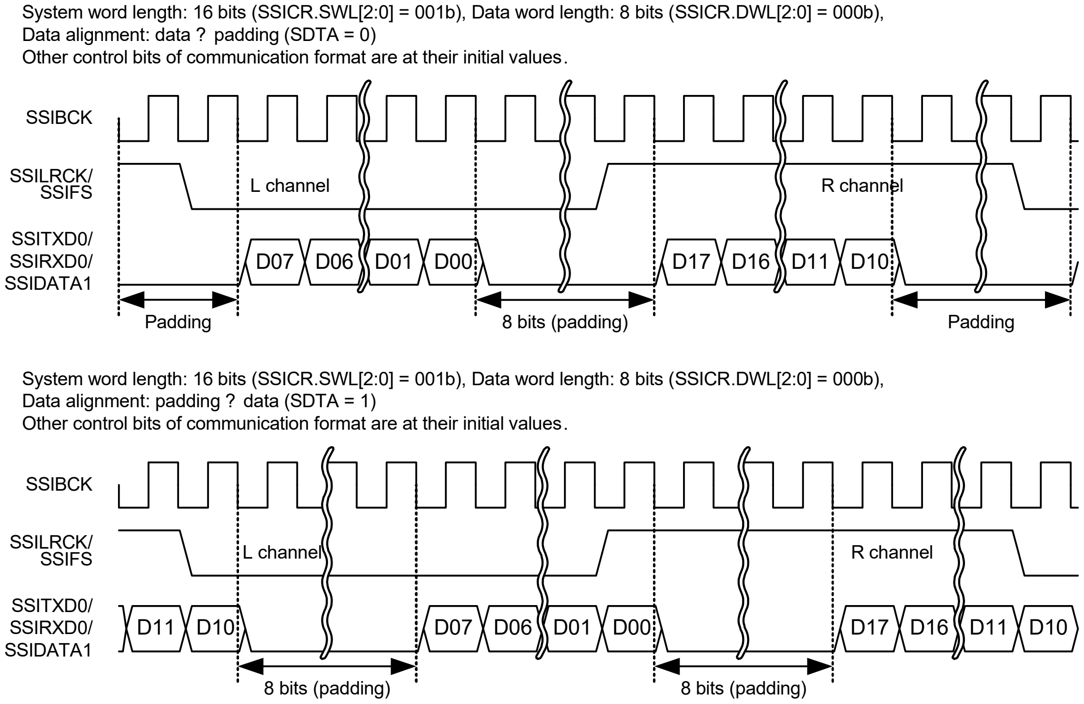
First transmission data	Second transmission data	Third transmission data	Fourth transmission data

DWL[2:0]	Receive shift register	SSIFRDR PDTA = 0 (left-justify)	SSIFRDR PDTA = 1 (right-justify)
000 (8 bits)	Invalid | 7 0 Invalid | 7 0 Invalid | 7 0 Invalid | 7 0	7 0 | Invalid 7 0 | Invalid 7 0 | Invalid 7 0 | Invalid	Setting prohibited
001 (16 bits)	Invalid | 15 0 Invalid | 15 0 Invalid | 15 0 Invalid | 15 0	15 0 | Invalid 15 0 | Invalid 15 0 | Invalid 15 0 | Invalid	Setting prohibited
010 to 100 18bit : X = 17 20bit : X = 19 22bit : X = 21 24bit : X = 23	Invalid | X 0 Invalid | X 0 Invalid | X 0 Invalid | X 0	X 0 | Invalid X 0 | Invalid X 0 | Invalid X 0 | Invalid	Invalid | X 0 Invalid | X 0 Invalid | X 0 Invalid | X 0
110 (32 bits)	31 0 31 0 31 0 31 0	31 0 31 0 31 0 31 0	Setting prohibited
111 (Setting prohibited)			

Figure 41.9 Alignment of placement data at reception

SDTA bit (Selects Serial Data Delay)

This bit sets how to align serial data and padding bits. For communication without padding bits, this bit is invalid.



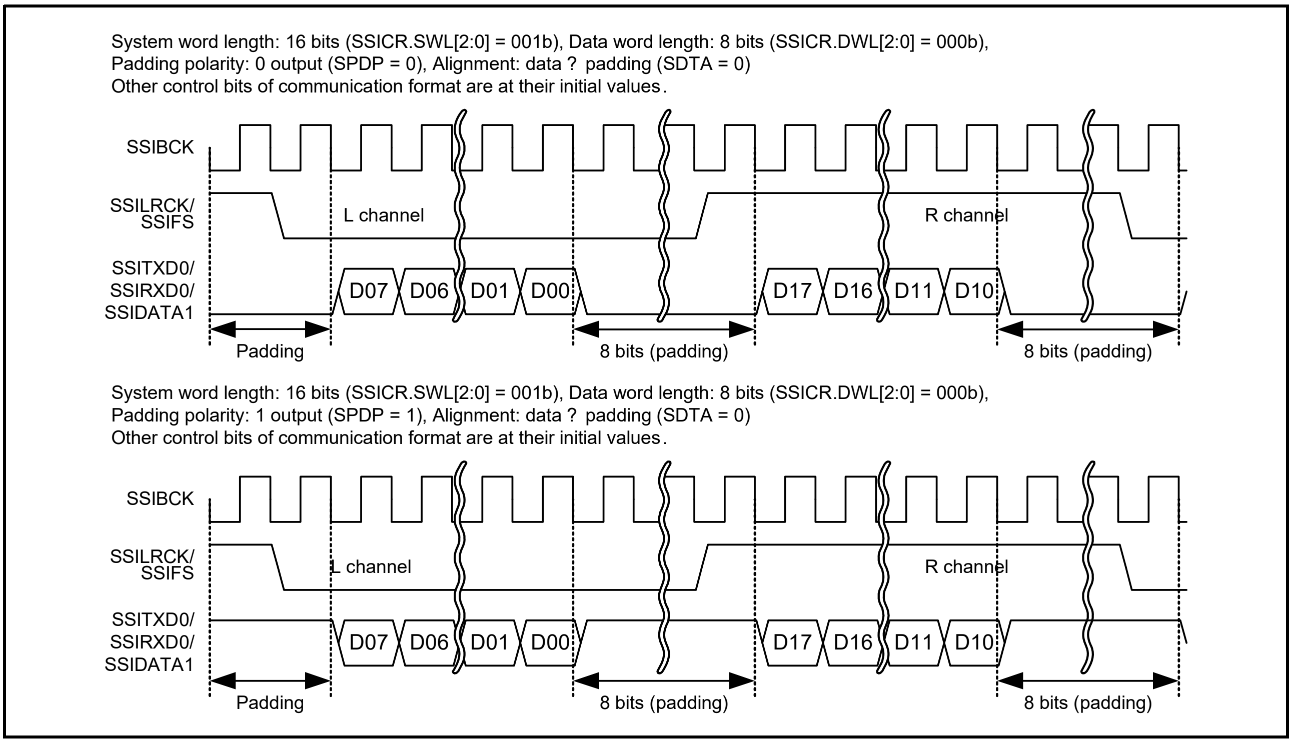
System word length: 16 bits (SSICR.SWL[2:0] = 001b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
Data alignment: data ? padding (SDTA = 0)
Other control bits of communication format are at their initial values.

System word length: 16 bits (SSICR.SWL[2:0] = 001b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
Data alignment: padding ? data (SDTA = 1)
Other control bits of communication format are at their initial values.

Figure 41.10 Alignment setting of serial data with padding bits

SPDP bit (Selects Serial Padding Polarity)

This bit sets polarity of padding bits.



System word length: 16 bits (SSICR.SWL[2:0] = 001b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
Padding polarity: 0 output (SPDP = 0), Alignment: data ? padding (SDTA = 0)
Other control bits of communication format are at their initial values.

System word length: 16 bits (SSICR.SWL[2:0] = 001b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
Padding polarity: 1 output (SPDP = 1), Alignment: data ? padding (SDTA = 0)
Other control bits of communication format are at their initial values.

Figure 41.11 Padding bit polarity

LRCKP bit (Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal)

This bit sets the initial value and polarity of SSILRCK/SSIFS. Set this bit according to the communication format to be used in SSIE. See Table 41.3 Initial output value and polarity of SSILRCK/SSIFS pin. For the slave-mode communication (MST = 0), only the start trigger is used.

Writing to these bits must be performed when the LR clock supply to the SSILRCK/SSIFS pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in SSIOFR.

Table 41.3 Initial output value and polarity of SSILRCK/SSIFS pin

Communication Format	Expected Initial State	Setting Value of LRCKP
I^{2}S	High	0
Monaural	Low	1
TDM	Low	1

Note: When the format to be used is compatible with the I^{2}S, monaural, or TDM format, specify settings to enable communication with the respective formats.

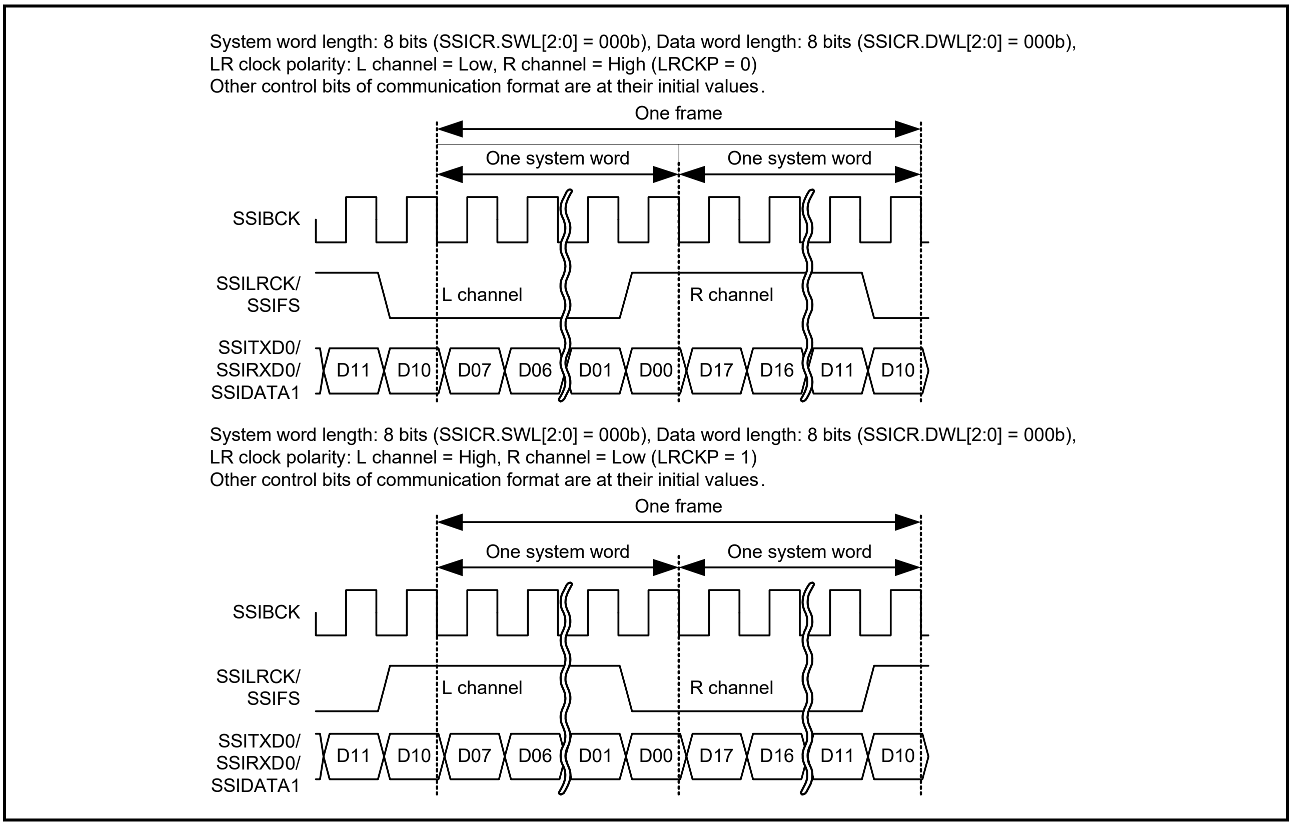


Figure 41.12 LR clock/frame synchronization polarity setting

BCKP bit (Selects Bit Clock Polarity)

This bit sets the bit clock polarity.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 41.4.3, FIFO Control Register (SSIFCR).

Table 41.4 Bit clock polarity

Communication	Master/Slave	Timing	BCKP = 0	BCKP = 1
Reception	Slave	At SSILRCK/SSIFS sampling	SSIBCK rising edge	SSIBCK falling edge
	Master/slave	At SSIRXD0/SSIDATA1 sampling	SSIBCK rising edge	SSIBCK falling edge
Transmission	Master	At change of SSILRCK/SSIFS output	SSIBCK falling edge	SSIBCK rising edge
	Master/slave	At change of SSITXD0/SSIDATA1 output	SSIBCK falling edge	SSIBCK rising edge

MST bit (Master Enable)

This bit sets master-/slave-mode communication.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in section 41.4.3, FIFO Control Register (SSIFCR).

SWL[2:0] bits (Selects System Word Length)

These bits set the number of bits in one system word. Padding bits are sent and received in relation with one data word set with DWL[2:0]. See Table 41.11 for details.

Writing to these bits must be performed when the LR clock supply to the SSILRCK/SSIFS pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in section 41.4.7, Audio Format Register (SSIOFR).

DWL[2:0] bits (Selects Data Word Length)

These bits set the number of bits in one data word. The data word length (number of bits per data word) must not exceed the system word length (number of bits per system word). For details, see Table 41.11.

FRM[1:0] bits (Selects Frame Word Number)

These bits set the frame word number in individual communication formats.

Writing to these bits must be performed when the LR clock supply to the SSILRCK/SSIFS pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in SSIOFR.

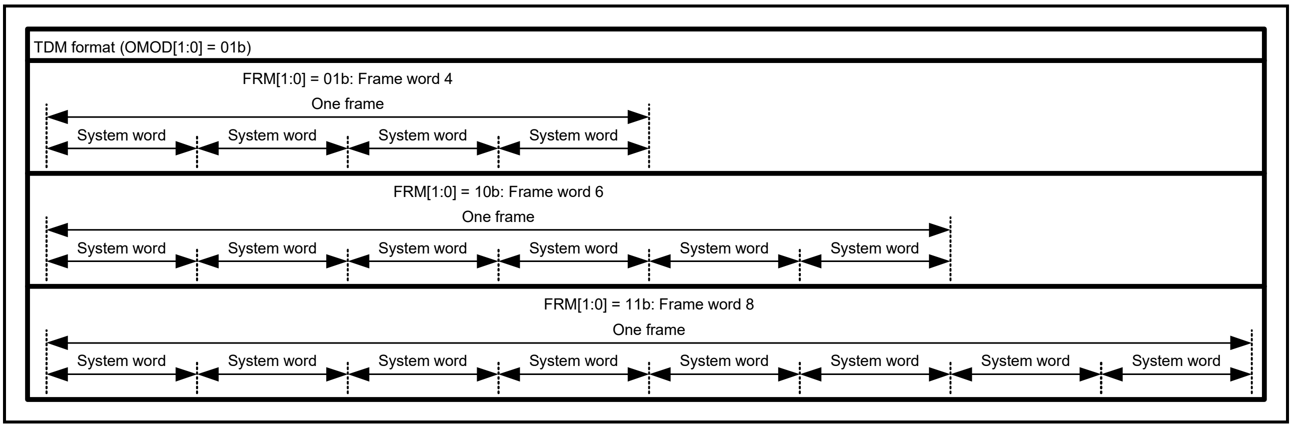


Figure 41.13 Frame word number

IIEN bit (Idle Mode Interrupt Output Enable)

This bit enables/disables output of idle mode interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.IIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.IIRQ = 1.

ROIEN bit (Receive Overflow Interrupt Output Enable)

This bit enables/disables output of receive overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.ROIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.ROIRQ = 1.

RUIEN bit (Receive Underflow Interrupt Output Enable)

This bit enables/disables output of receive underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.RUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.RUIRQ = 1.

TOIEN bit (Transmit Overflow Interrupt Output Enable)

This bit enables/disables output of transmit overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TOIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TOIRQ = 1.

TUIEN bit (Transmit Underflow Interrupt Output Enable)

This bit enables/disables output of transmit underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TUIRQ = 1.

CKS bit (Selects an Audio Clock for Master-mode Communication)

This bit sets the audio clock in master-mode communication (MST = 1). In slave-mode communication (MST = 0), setting of this bit is invalid.

Writing to this bit must be performed when the supply of AUDIO_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in SSIFCR.

41.4.2 Status Register (SSISR)

Address(es): SSIE0.SSISR 4004 E004h, SSIE1.SSISR 4004 E104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b24 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	IIRQ	Idle Mode Status Flag	0: In the communication state 1: In the idle state.	R
b26	ROIRQ	Receive Overflow Error Status Flag	0: No receive overflow error is generated 1: A receive overflow error is generated.	R/W
b27	RUIRQ	Receive Underflow Error Status Flag	0: No receive underflow error is generated 1: A receive underflow error is generated.	R/W
b28	TOIRQ	Transmit Overflow Error Status Flag	0: No transmit overflow error is generated 1: A transmit overflow error is generated.	R/W
b29	TUIRQ	Transmit Underflow Error Status flag	0: No transmit underflow error is generated 1: A transmit underflow error is generated.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is configured with status flags that indicate SSIE operational state.

IIRQ bit (Idle Mode Status Flag)

This is a status flag that indicates the idle state. It indicates whether SSIE is in the idle state or communication state.

For details, see Figure 41.14 and Figure 41.15.

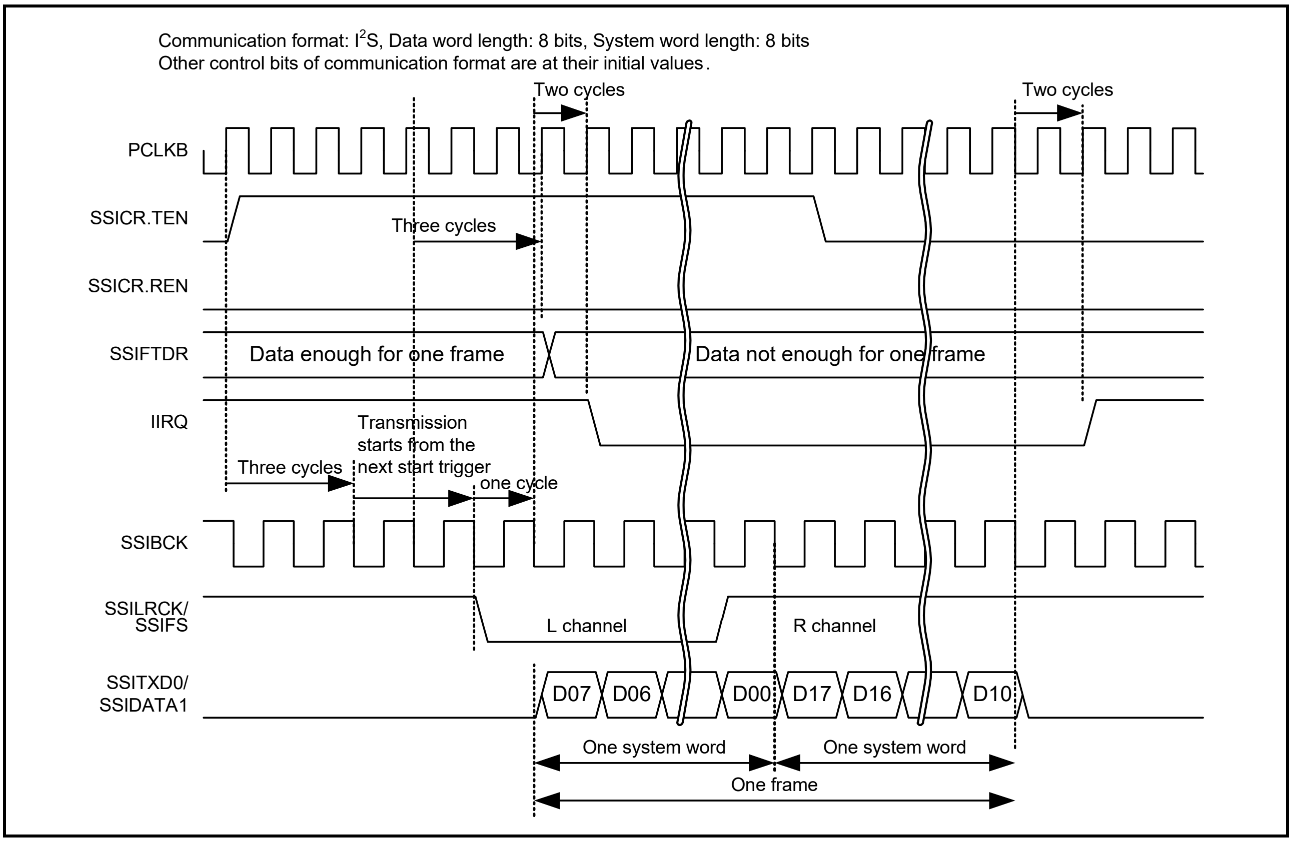


Figure 41.14 IIRQ setting timing (transmission)

- Transmitter (dedicated to transmission)

[Clearing condition]

While transmission was enabled (SSICR.TEN = 1 and SSICR.REN = 0), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger was generated by the SSILRCK/SSIFS signal.

[Clearing timing]

1 SSIBCK cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

2 PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.

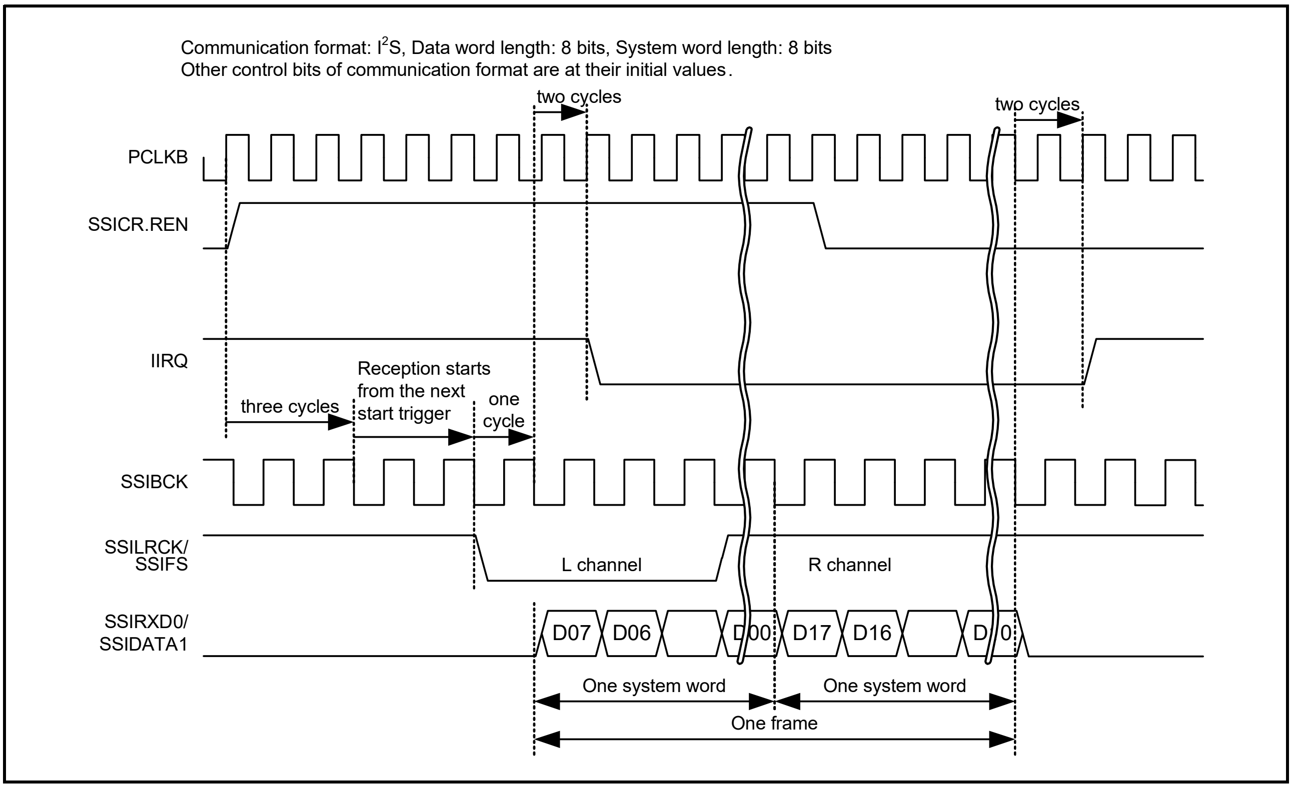


Figure 41.15 IIRQ setting timing (reception)

- Receiver (dedicated to reception)

[Clearing condition]

While reception was enabled (SSICR.TEN = 0 and SSICR.REN = 01, a start trigger was generated by the SSILRCK/SSIFS signal.

[Clearing timing]

1 SSIBCK cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), reception of one frame was complete.

[Setting timing]

2 PCLKB cycles after the end of reception (at a frame boundary) that is the setting condition.

- Transceiver (transmission and reception)

[Clearing condition]

While transmission and reception were enabled (SSICR.TEN = 1 and SSICR.REN = 1), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger is generated by the SSILRCK/SSIFS signal.

[Clearing timing]

1 SSIBCK cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

2 PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.

ROIRQ bit (Receive Overflow Error Status Flag)

This is a status flag that indicates a receive overflow error. This flags is set by automatic determination but it must be cleared by register access. This flag indicates that received data is supplied at a higher rate than requested. Data is not transferred from the receive shift register to SSIFRDR where a receive overflow error is generated. For the procedure to recover from the overflow error, see section 41.8.6, Error Handling. This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition:

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 41.19)
2. 1 PCLKB cycle after writing 1 to SSICR.REN.*3

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.REN.

Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

[Setting condition]

At completion of receiving new data while SSIFRDR is full.

[Setting timing]

3 cycles of PCLKB after reception is completed.

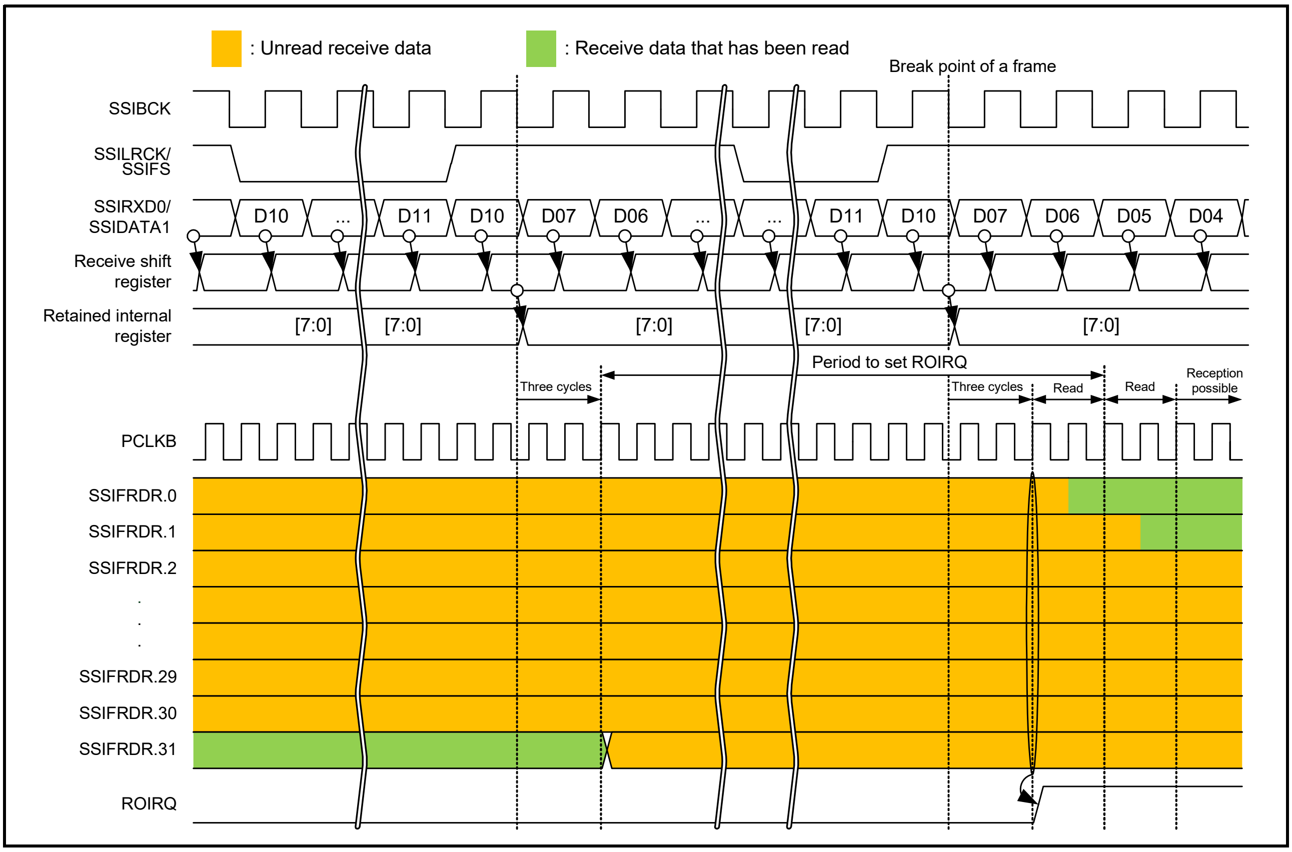


Figure 41.16 ROIRQ setting timing

RUIRQ bit (Receive Underflow Error Status Flag)

This is a status flag that indicates a receive underflow error. This flags is set by automatic determination but it must be cleared by register access. This flag indicates that SSIFRDR is read while it is empty. Data read from SSIFRDR where a receive underflow error is generated is invalid. See section 41.8.6, Error Handling for the error recovery procedure. This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST). Note, however, that this flag is not set even if the SSIFRDR register is read while the receive FIFO data register is reset (by setting SSIFCR.RFRST to 1).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 41.19)
2. 1 PCLKB cycle after writing 1 to SSICR.REN.*3

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.REN.

Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

[Setting condition]

Reading from SSIFRDR while it is empty.

[Setting timing]

At completion of reading from SSIFRDR. See Figure 41.17.

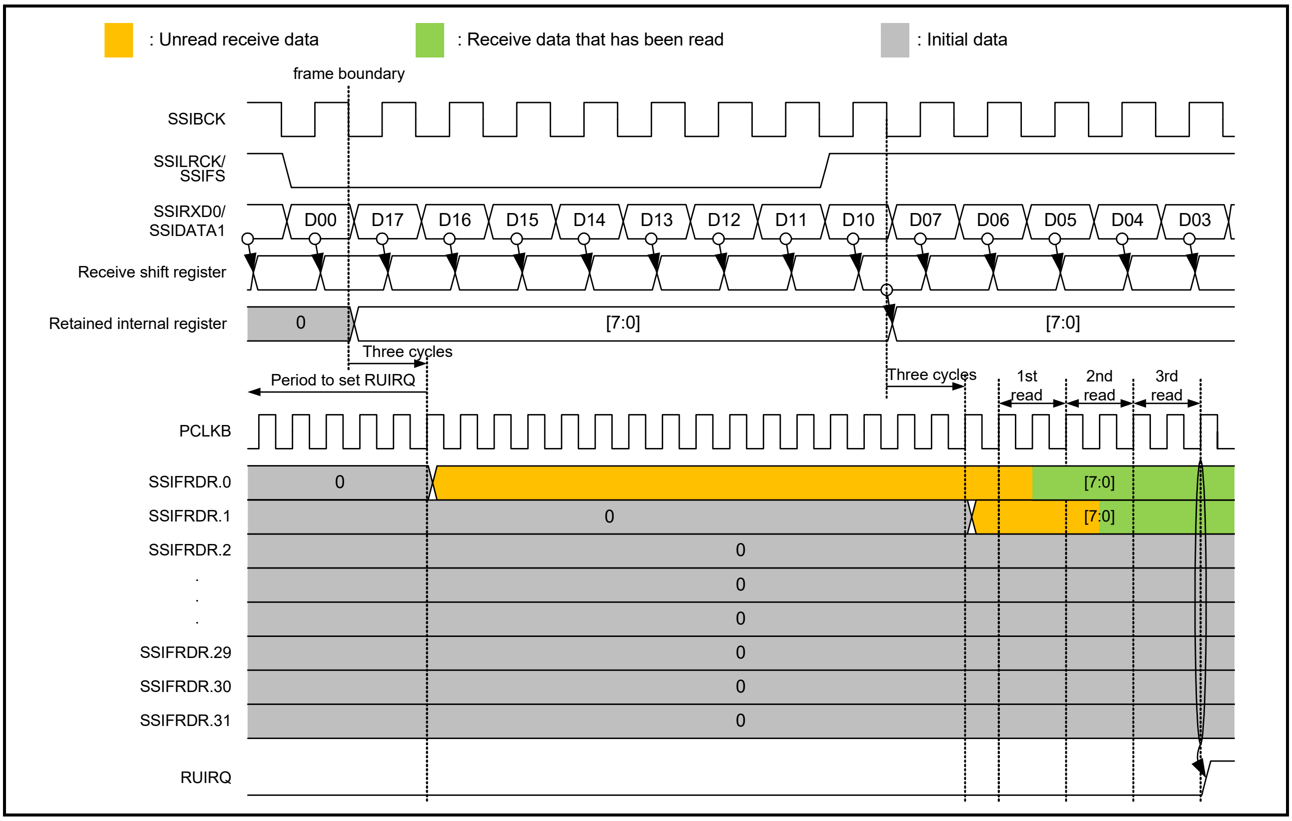


Figure 41.17 RUIRQ setting timing

TOIRQ bit (Transmit Overflow Error Status Flag)

This is a status flag that indicates a transmit overflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that an attempt has been made to write data to the SSIFTDR register when the register is full of data. The data writing that causes a transmit overflow is ignored. For the procedure to recover from the overflow error, see section 41.8.6, Error Handling. This flag is not cleared by a transmit FIFO data register reset (SSIFCR.TFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

(1) Writing 0 to this bit after reading 1 from this bit*2

(2) Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

(1) When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 41.19)

(2) 1 PCLKB cycle after writing 1 to SSICR.TEN.*3

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.TEN.

Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

[Setting condition]

An attempt was made to write data to the SSIFTDR register when the register is full of data.

[Setting timing]

At completion of writing to SSIFTDR. For details, see Figure 41.18.

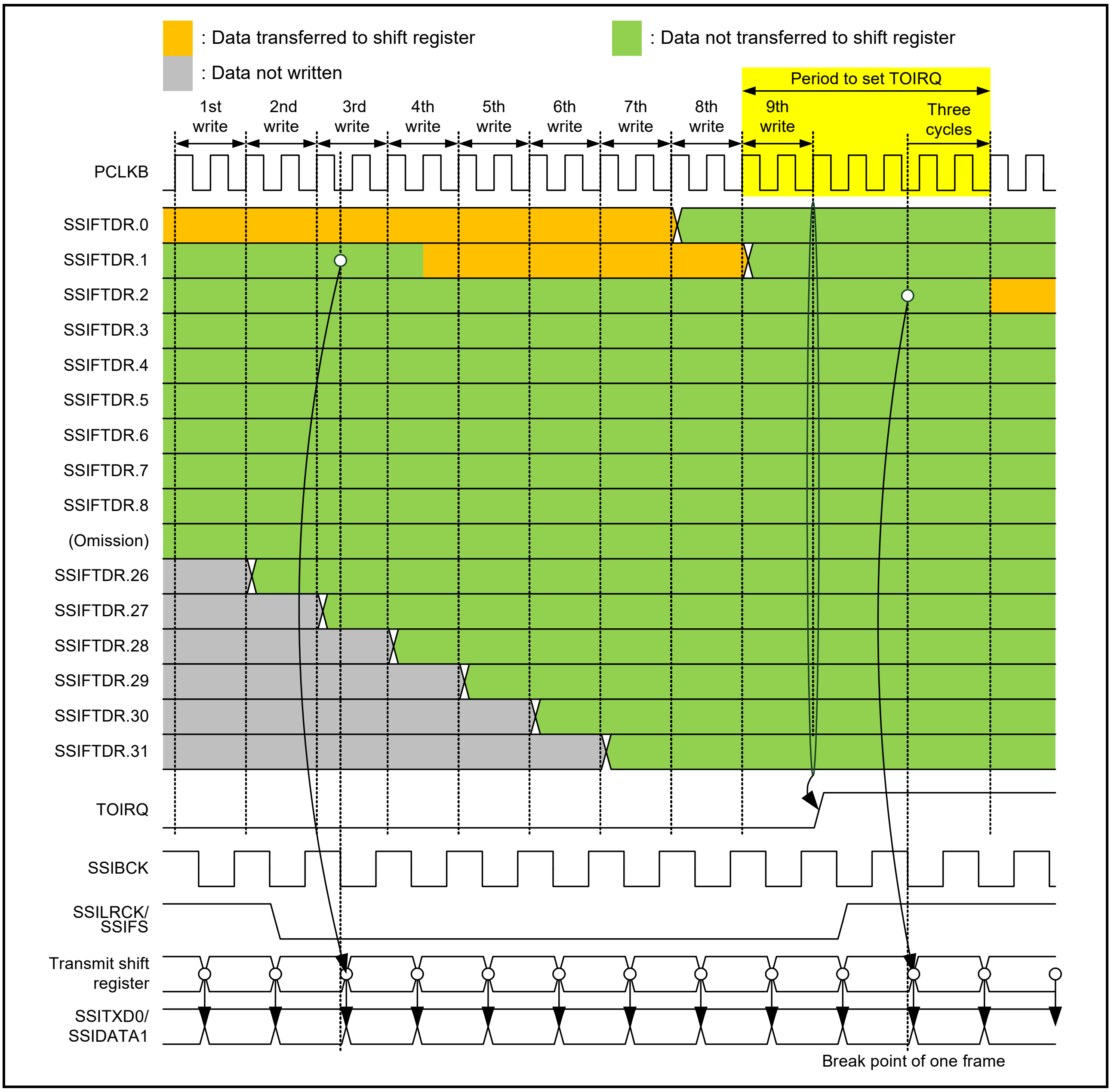


Figure 41.18 TOIRQ setting timing

TUIRQ bit (Transmit Underflow Error Status flag)

This is a status flag that indicates a transmit underflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that writing the serial data required for a frame to SSIFTDR did not catch up with transmission of the frame. Even if this flag is cleared after it has been set, the SSITXD0/SSIDATA1 output remains to be 0. To output the data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the communication stop procedure in Figure 41.56 and error-handling procedure in Figure 41.57. For the procedure to recover from an error, see section 41.8.6, Error Handling. This flag is not cleared by a reset of transmit FIFO data register (by the SSIFCR.TFRST signal).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit*2
2. Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit
2. 1 PCLKB cycle after writing 1 to SSICR.TEN.*3

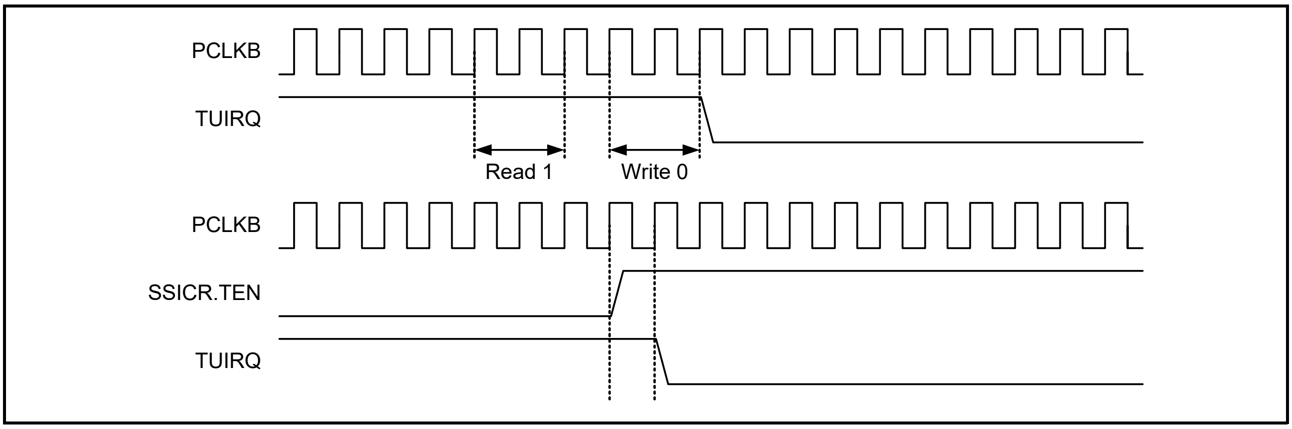


Figure 41.19 TUIRQ clearing timing

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- 1 PCLKB cycle passes after 1 has been written to SSICR.TEN.

Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

[Setting condition]

When communication continues over a frame boundary, the transmit data required for the next frame has not been written to SSIFTDR. For details, see Figure 41.20 and Figure 41.21.

[Setting timing]

3 PCLKB cycles after the frame boundary. For details, see Figure 41.20.

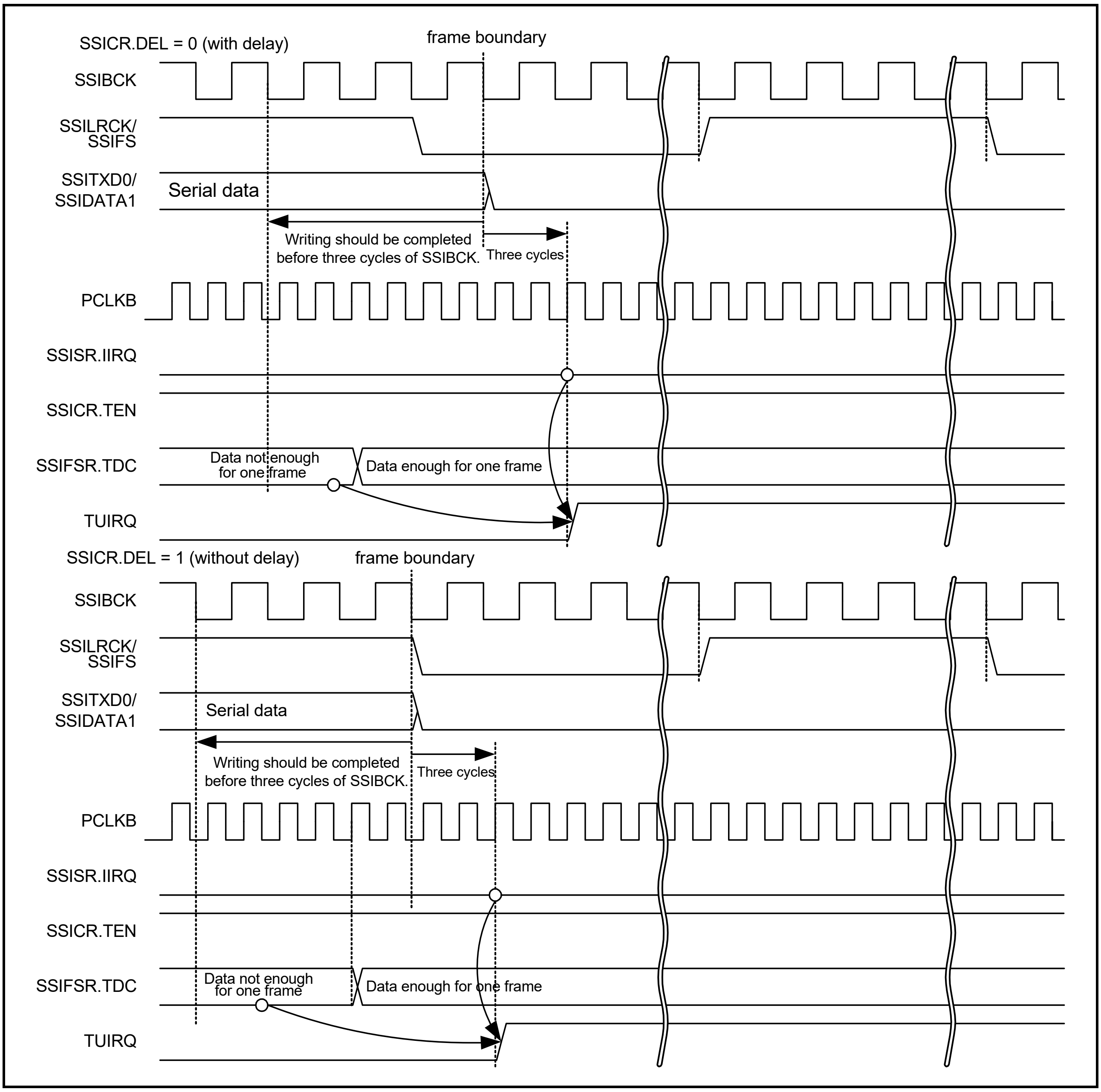


Figure 41.20 TUIRQ setting timing (when communication continues)

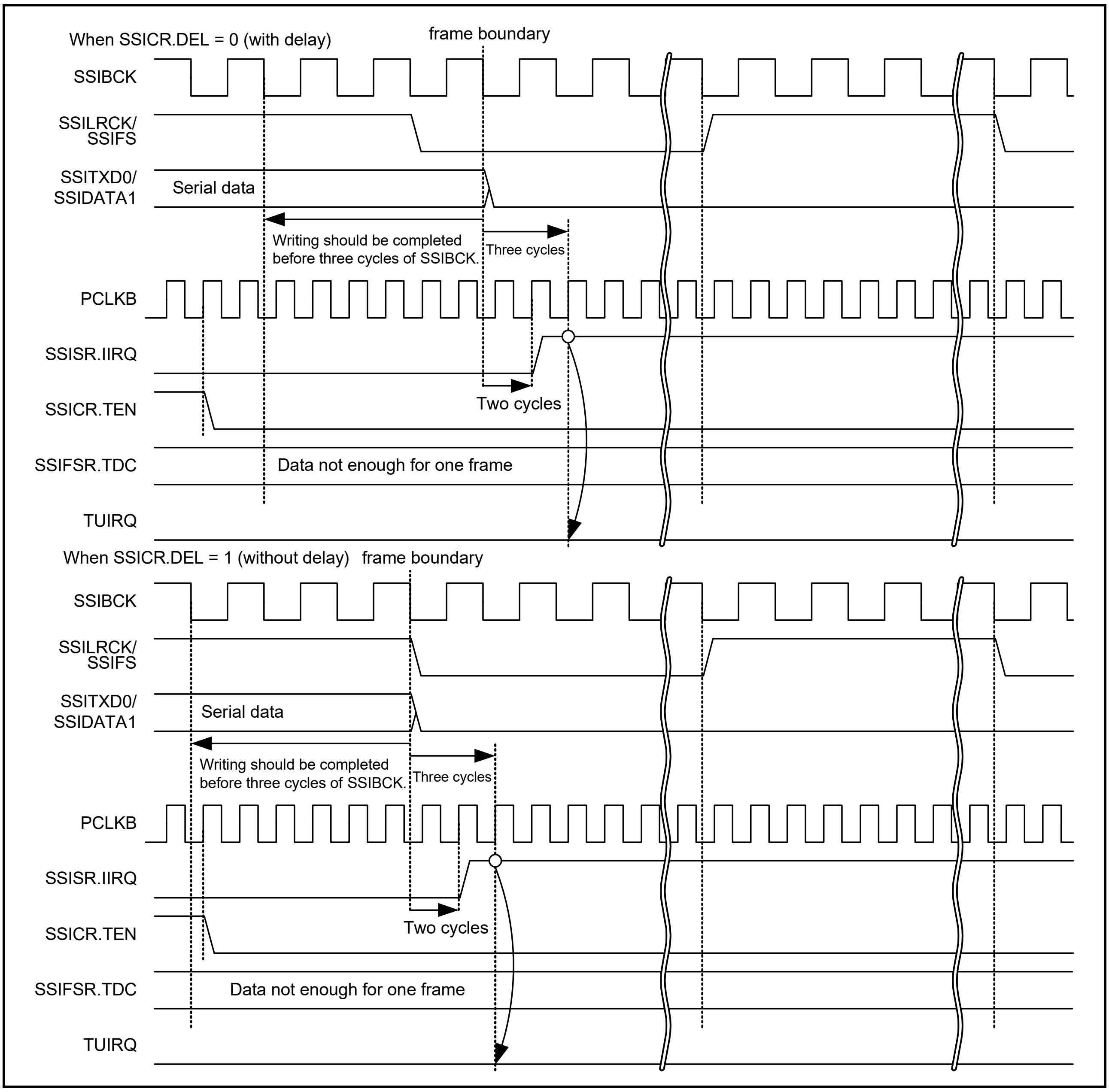


Figure 41.21 TUIRQ setting timing (when communication stops)

41.4.3 FIFO Control Register (SSIFCR)

Address(es): SSIE0.SSIFCR 4004 E010h, SSIE1.SSIFCR 4004 E110h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RFRST	Receive FIFO Data Register Reset*1	0: Clears a receive data FIFO reset condition 1: Sets a receive data FIFO reset condition.	R/W
b1	TFRST	Transmit FIFO Data Register Reset*1	0: Clears a transmit data FIFO reset condition 1: Sets a transmit data FIFO reset condition.	R/W
b2	RIE	Receive Data Full Interrupt Output Enable	0: Disables receive data full interrupts 1: Enables receive data full interrupts.	R/W
b3	TIE	Transmit Data Empty Interrupt Output Enable	0: Disables transmit data empty interrupts 1: Enables transmit data empty interrupts.	R/W
b10 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	BSW	Byte Swap Enable*1	0: Disables byte swap 1: Enables byte swap.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SSIRST	Software Reset	0: Clears a software reset condition 1: Sets a software reset condition.	R/W
b30 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	AUCKE	AUDIO_MCK Enable in Master-mode Communication*1	0: Disables supply of AUDIO_MCK 1: Enables supply of AUDIO_MCK.	R/W

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

This register sets a software reset, byte swap, and enable/disable of interrupt requests.

RFRST bit (Receive FIFO Data Register Reset)

This bit sets a software reset of the receive FIFO data register (SSIFRDR). Writing 1 to this bit initializes the internal state related to SSIFRDR. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 41.5. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

Table 41.5 Bits subject to software reset by the RFRST bit

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	00h	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	04h	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	10h	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	14h	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	18h	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	1ch	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	20h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	24h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

TFRST bit (Transmit FIFO Data Register Reset)

This bit sets a software reset of the transmit FIFO data register (SSIFTDR). Writing 1 to this bit initializes the internal state related to SSIFTDR. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 41.6. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

Table 41.6 Bits subject to software reset by the TFRST bit

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	00h	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	04h	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	10h	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	14h	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	18h	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	1ch	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	20h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	24h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

RIE bit (Receive Data Full Interrupt Output Enable)

This bit enables/disables output of receive data full interrupts. Use a receive data full interrupt as an interrupt to trigger data reading from the receive FIFO data register. Write 1 to this bit after specifying the setting condition for receive data full interrupt (by using the SSISCR.RDFS bit). Figure 41.22 shows the timing of generating the receive data full interrupt.

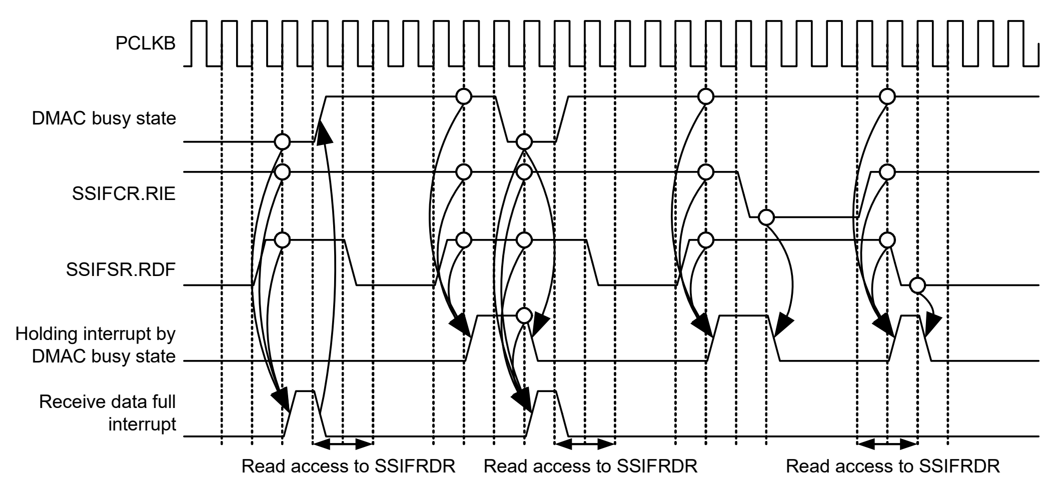


Figure 41.22 Timing of receive data full interrupt

TIE bit (Transmit Data Empty Interrupt Output Enable)

This bit enables/disables output of transmit data empty interrupts. Use a transmit data empty interrupt as an interrupt to trigger data writing to the transmit FIFO data register. Write 1 to this bit after specifying the setting condition for transmit data empty interrupt (by using the SSISCR.TDES bit). Figure 41.23 shows the timing of generating the transmit data empty interrupt.

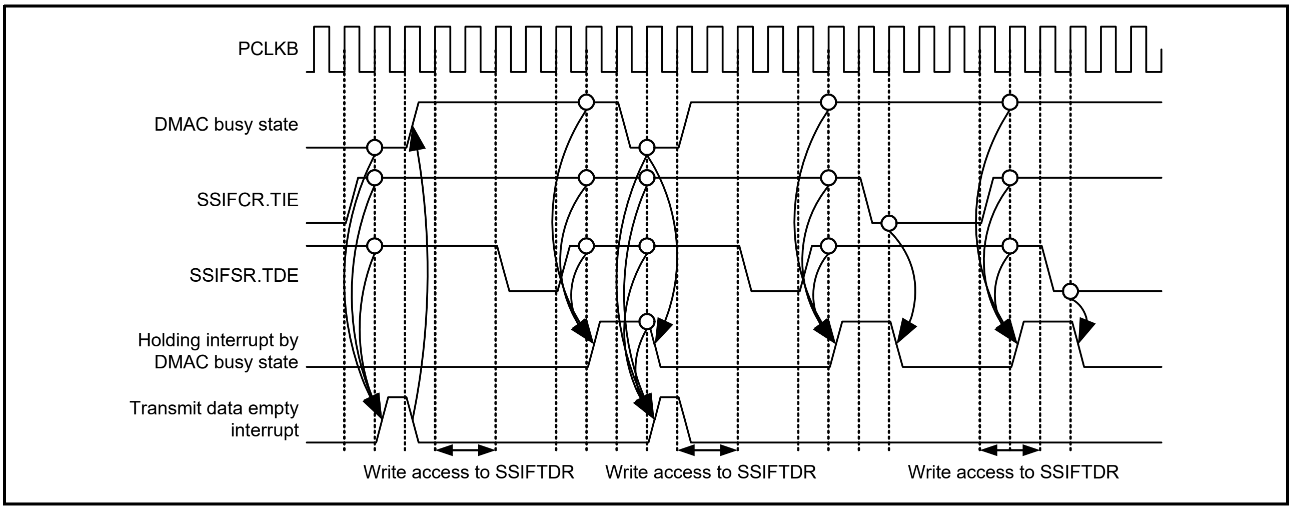


Figure 41.23 Timing of transmit data empty interrupt

BSW bit (Byte Swap Enable)

This bit enables/disables byte swap of register access for the transmit FIFO data register (SSIFTDR) and the receive FIFO data register (SSIFRDR). This bit is valid only with 16-bit access or 32-bit access to SSIFTDR and SSIFRDR. For details, see Figure 41.24.

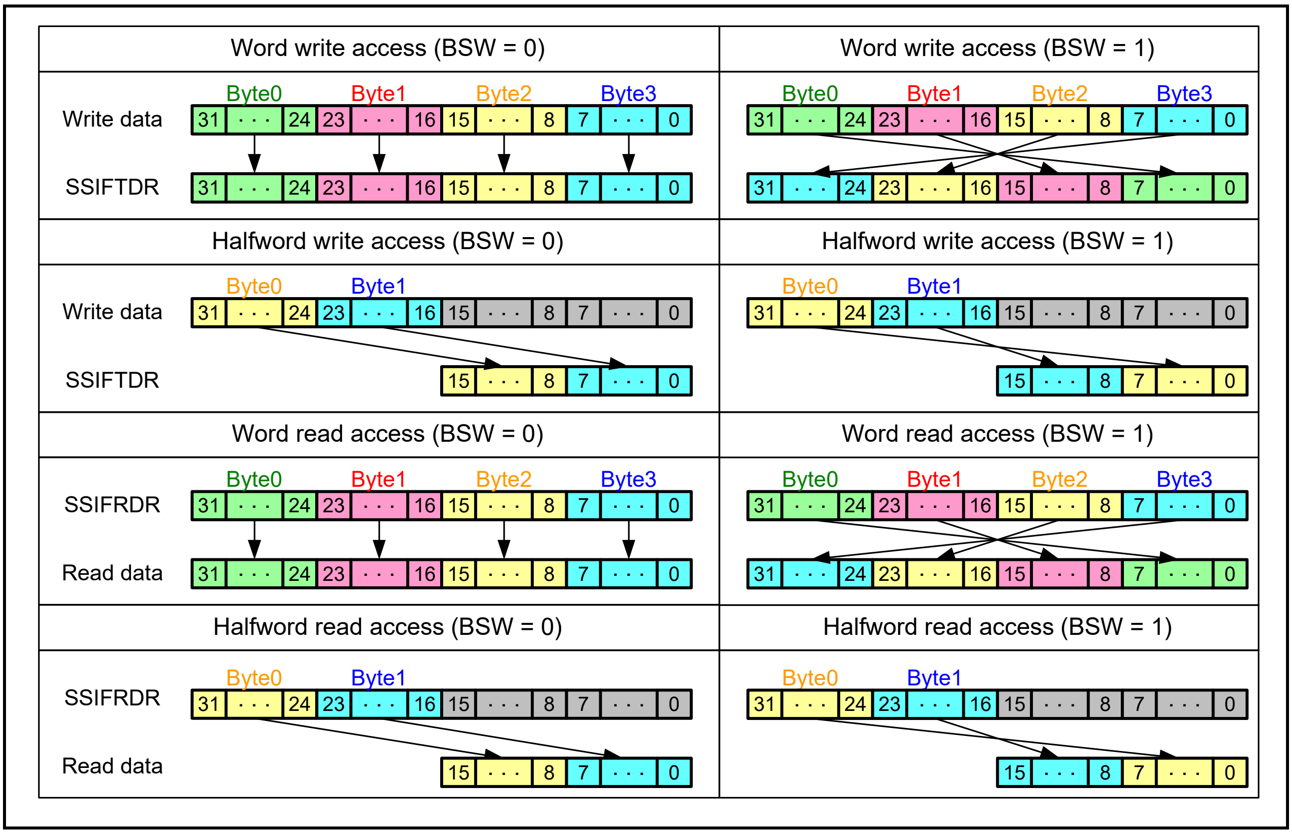


Figure 41.24 Operation example of byte swap

SSIRST bit (Software Reset)

This bit sets a software reset of SSIE. Writing 1 to this bit initializes the internal state of SSIE. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 41.7. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

To stop communication of SSIE immediately, after turning off the peripheral functions, write 1 to this bit. Initialization by a software reset is performed without any relation with the bit clock.

Table 41.7 Bits subject to software reset by the SSIRST bit

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	00h	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	04h	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	10h	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	14h	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	18h	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	1ch	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	20h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	24h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

AUCKE bit (AUDIO_MCK Enable in Master-mode Communication)

This bit enables/disables supply to AUDIO_MCK while in master-mode communication (MST = 1).

Changing the value of this bit must be performed only after specifying the settings related to AUDIO_MCK (by using the CKS, MST, BCKP, and CKDV bits in the SSICR register).

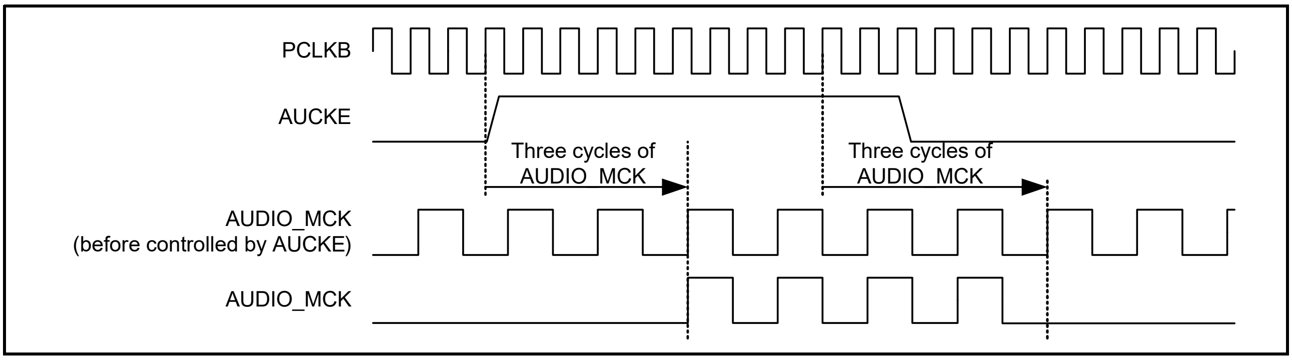


Figure 41.25 Stop/resume of AUDIO_MCK

Note: In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCK. To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in Figure 41.52 or wait for an idle state by taking the procedure to resume communication in Figure 41.58.
In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.ADCKE. If 0 is written to SSIFCR.ADCKE before SSIE becomes idle, take the procedure to start communication in Figure 41.52.

Figure 41.26 and Figure 41.27 show the timings of signal operation in the period from setting this bit to 1 to the output to the SSIBCK pin.

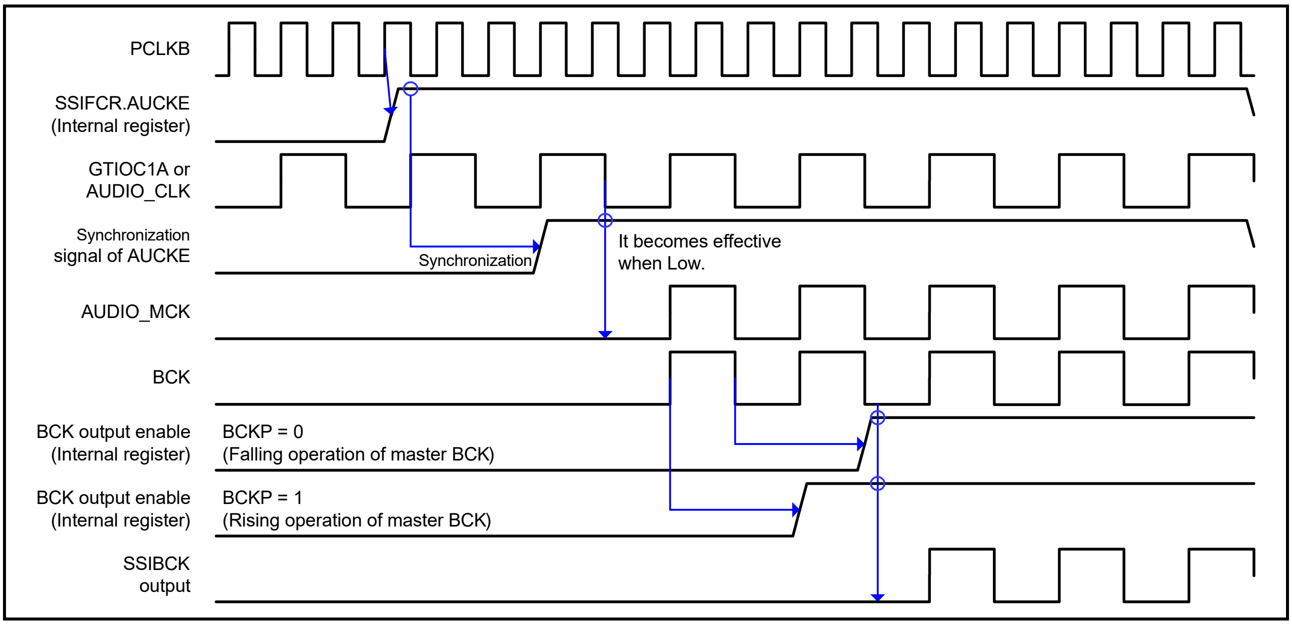


Figure 41.26 Timing diagram for the operation from system reset to start of master-mode communication

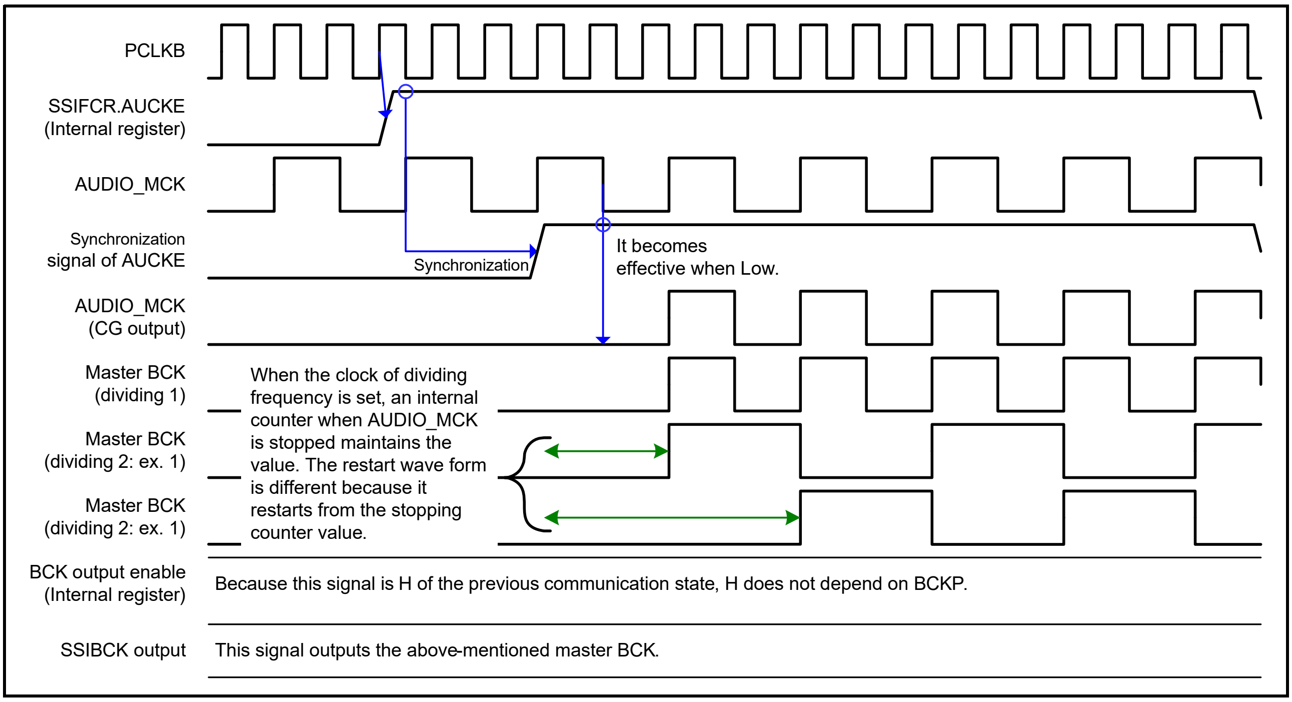


Figure 41.27 Timing diagram for the operation from stop of communication to start of master-mode communication

Note: If the supply of AUDIO_MCK stops, the value of the SSIBCK pin is held. Therefore, the SSIBCK signal might stop in the H (high level) state.

41.4.4 FIFO Status Register (SSIFSR)

Address(es): SSIE0.SSIFSR 4004 E014h, SSIE1.SSIFSR 4004 E114h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RDF	Receive Data Full Flag	0: The size of received data in SSIFRDR is not more than the value of SSISCR.RDFS 1: The size of received data in SSIFRDR is not less than the value of SSISCR.RDFS plus one.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	RDC[5:0]	Number of Receive FIFO Data Indication Flag	Number of receive FIFO data indication flag.	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TDE	Transmit Data Empty Flag	0: The free space of SSIFTDR is not more than the value of SSISCR.TDES 1: The free space of SSIFTDR is not less than the value of SSISCR.TDES plus one.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b29 to b24	TDC[5:0]	Number of Transmit FIFO Data Indication Flag	Number of transmit FIFO data indication flag.	R
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is configured with status flags that indicate the status of the transmit FIFO data register and the receive FIFO data register.

RDF bit (Receive Data Full Flag)

This bit indicates that the receive FIFO data register (SSIFRDR) has unread received data not less than the amount set with the SSISCR.RDFS bit plus one. This flag is set by automatic determination but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.

[Clearing condition]

Either of the following two:*1

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)*2
2. Last access (DTC/DMAC operation) to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 41.19)
2. After the PCLKB cycle in which the last access instruction is issued to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

Note 1. These bits are cleared by a software reset (SSIFCR.SSIRST = 1) and receive FIFO data register reset (SSIFCR.RFRST = 1). Reset conditions available for these bits are the software reset and receive FIFO data register reset as well as the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:
- A software reset is done (SSIFCR.SSIRST = 1).
- A receive FIFO data register reset is done (SSIFCR.RFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

SSIFRDR has data not less than the amount set with the SSISCR.RDFS bit plus one.

[Setting timing]

At completion of transfer from the shift register that results in SSIFRDR having data not less than the amount set with the SSISCR.RDFS bit plus one.

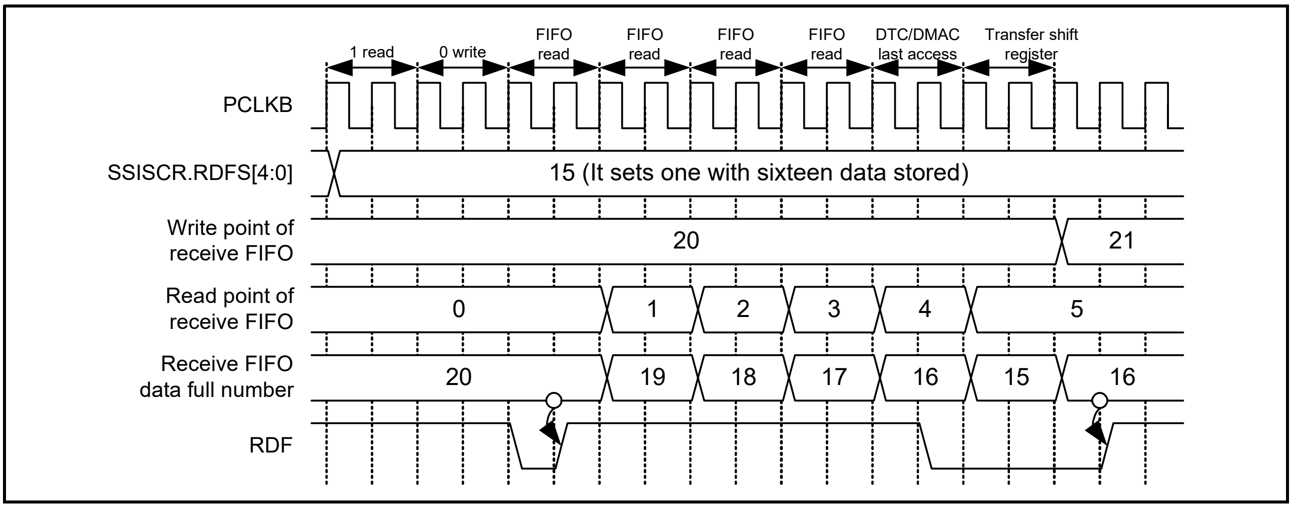


Figure 41.28 Timing diagram for setting and clearing RDF

RDC[5:0] bits (Number of Receive FIFO Data Indication Flag)

These bits indicate the number of valid data that are stored in the receive FIFO data register (SSIFRDR). With this flag as 0h, there is no received data. With 20h, the register is filled with received data and there is no free space.

TDE bit (Transmit Data Empty Flag)

This bit indicates that the transmit FIFO data register (SSIFTDR) has free space not less than the amount set with the SSIFCR.TTRG bit plus one. This flag is set by automatic determination but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.*1

[Clearing condition]

Either of the following two:

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)*2
2. Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

(1) When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 41.19)

(2) Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1) and transmit FIFO data register reset (SSIFCR.TFRST = 1). The software reset and transmit FIFO data register reset have priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:
- A software reset is done (SSIFCR.SSIRST = 1).
- A transmit FIFO data register reset is done (SSIFCR.TFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

SSIFTDR has free space not less than the amount set with the SSIFCR.TTRG bit plus one.

[Setting timing]

While operating on PCLKB, SSIFTDR is found to have free space not less than "size set in the SSISCR.TDES bits + 1."

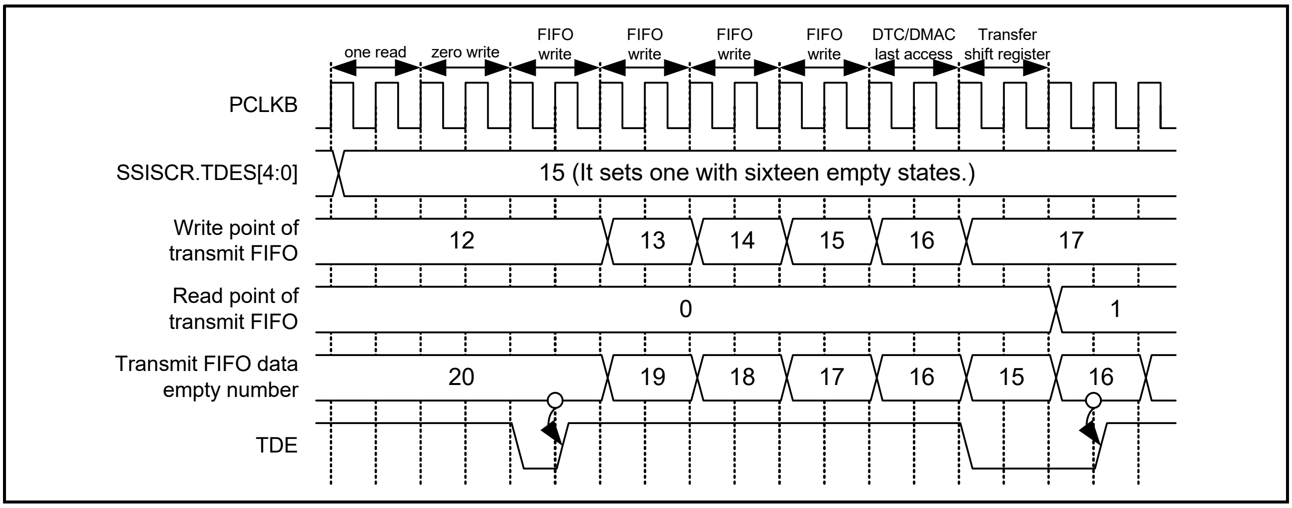


Figure 41.29 Timing diagram for setting and clearing TDE

TDC[5:0] bits (Number of Transmit FIFO Data Indication Flag)

These bits indicate the number of valid data that are stored in the transmit FIFO data register (SSIFTDR). With this flag as 0h, there is no data to be transmitted. With 20h, there is no space to write data.

41.4.5 Transmit FIFO Data Register (SSIFTDR)

Address(es): SSIE0.SSIFTDR 4004 E018h, SSIE1.SSIFTDR 4004 E118h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SSIFTDR [31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSIFTDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SSIFTDR[31:0]	Transmit FIFO Data	Transmit FIFO data.	W

This register stores data to be serially transmitted. 0 is returned when this register is read.

When you use this register for transmission, specify data writing to this register as the DTC/DMAC operation that is triggered by a transmit data empty interrupt. Determine the access size to this register according to the data word length to be communicated in Table 41.8.

Table 41.8 Register access restriction to FIFOs

	Access Size			
SSICR.DWL[2:0]	Data Word Length	Byte	Halfword	Word
000b	8	√	—	—
001b	16	—	√	—
010b	18	—	—	√
011b	20	—	—	√
100b	22	—	—	√
101b	24	—	—	√

Table 41.8 Register access restriction to FIFOs

	Access Size			
SSICR.DWL[2:0]	Data Word Length	Byte	Halfword	Word
110b	32	—	—	√
111b	Setting prohibited	—	—	—

Figure 41.30 shows register access to the transmit FIFO data register.

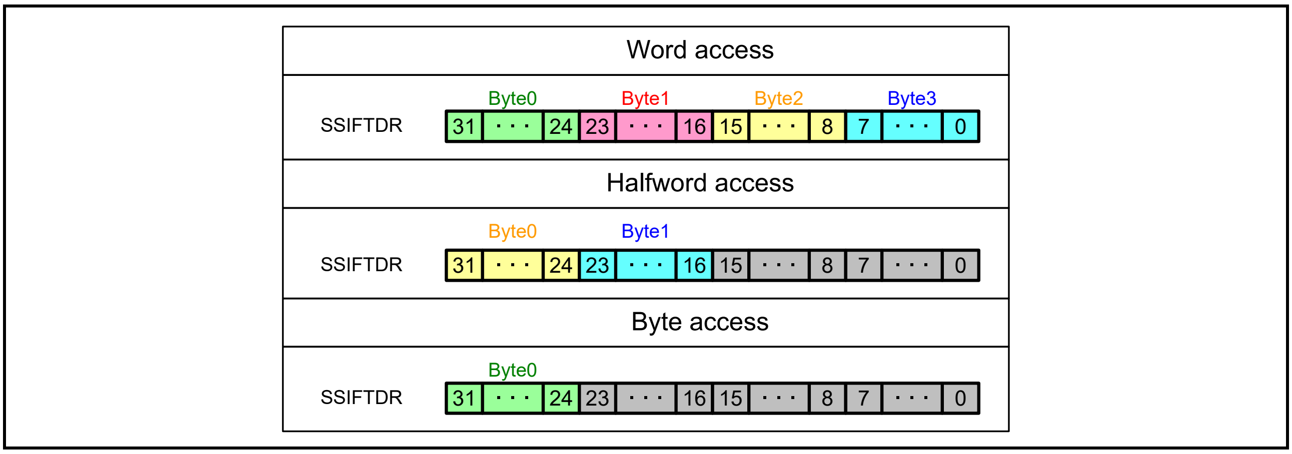


Figure 41.30 Example of register access to the transmit FIFO data register

Figure 41.31 shows the configurations and operation examples of the transmit FIFO data register and transmit shift register. The configurations are for storing data to FIFO and not related with communication.

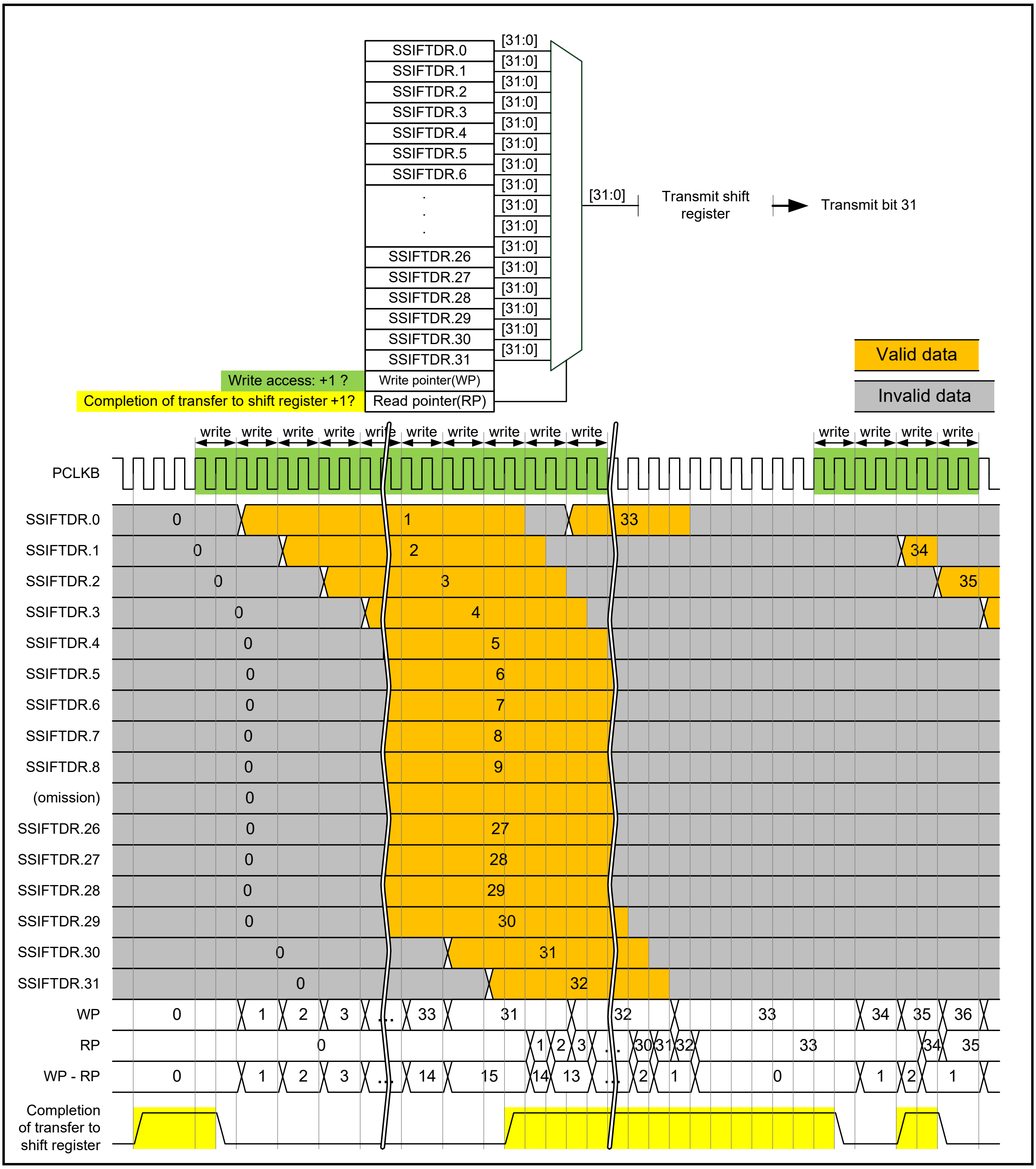


Figure 41.31 Configuration of the transmit FIFO data register and transmit shift register, and FIFO operation example

41.4.6 Receive FIFO Data Register (SSIFRDR)

Address(es): SSIE0.SSIFRDR 4004 E01Ch, SSIE1.SSIFRDR 4004 E11Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SSIFRDR [31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSIFRDR [15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	SSIFRDR[31:0]	Receive FIFO Data	Receive FIFO data.	R

When you use this register for reception, specify data reading from this register as the DTC/DMAC operation that is triggered by a receive data full interrupt. Determine the access size to this register according to the data word length to be communicated in Table 41.8.

Register access to the receive FIFO data register is same as for the transmit FIFO data register.

Figure 41.31 shows the configurations and operation examples of the receive FIFO data register and receive shift register.

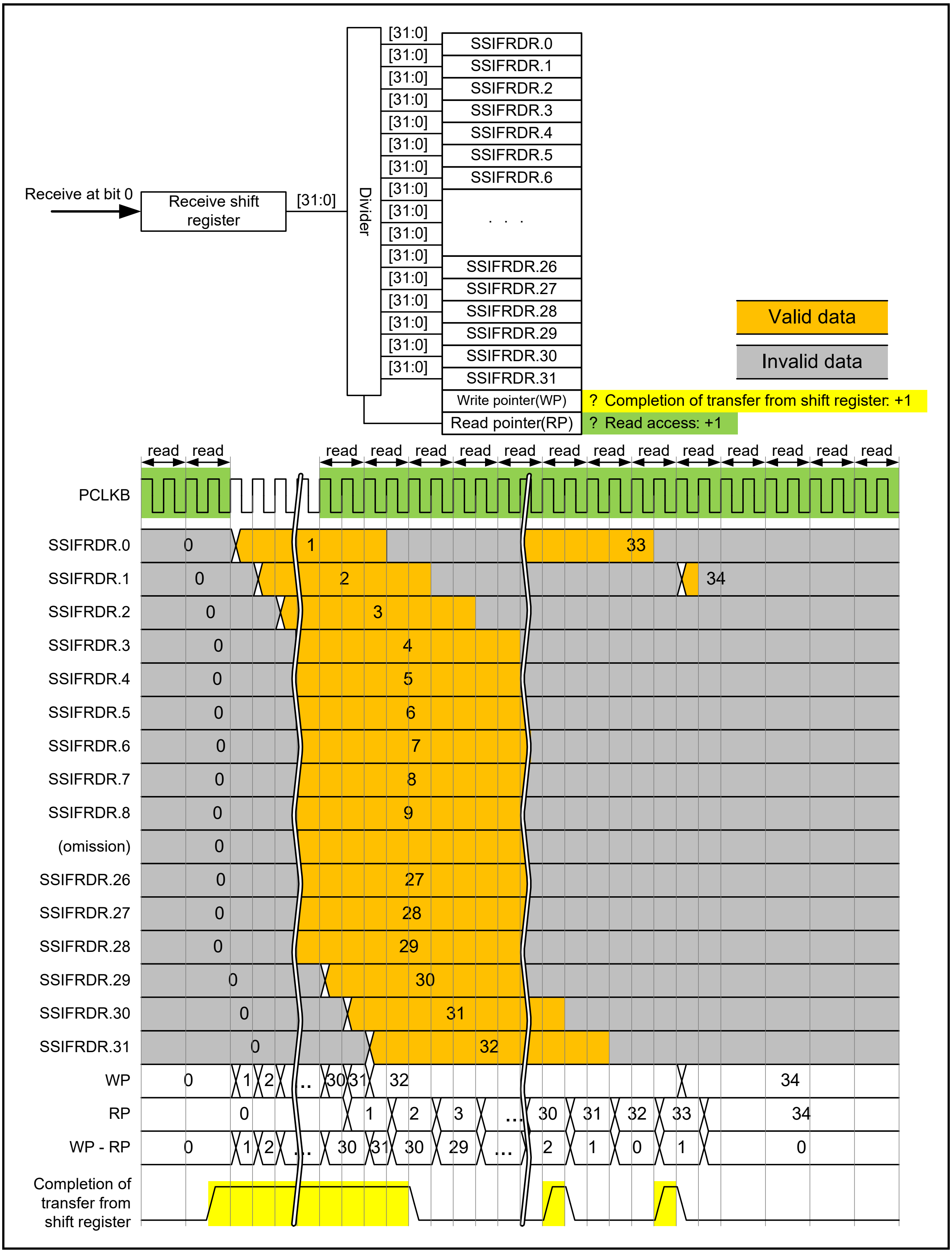


Figure 41.32 Configuration of the transmit FIFO data register and transmit shift register, and FIFO operation example

41.4.7 Audio Format Register (SSIOFR)

Address(es): SSIE0.SSIOFR 4004 E020h, SSIE1.SSIOFR 4004 E120h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	OMOD[1:0]	Audio Format Select*3, *4	00: I2S format 01: TDM format 10: Monaural format 11: Setting prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	LRCONT	Whether to Enable LRCK/FS Continuation*1, *2	0: Disables LRCK/FS continuation 1: Enables LRCK/FS continuation.	R/W
b9	BCKASTP	Whether to Enable Stopping BCK Output When SSIE is in Idle Status*1, *2	0: Always outputs BCK to the SSIBCK pin 1: Automatically controls output of BCK to the SSIBCK pin.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is valid only in master-mode communication (SSICR.MST = 1). The setting is invalid in slave-mode communication (SSICR.MST = 0).

Note 2. The BCKASTP and LRCONT bits must not be set to 1 together.

Note 3. While SSIE is communicating (SSISR.IIRQ = 0), writing to these bits is prohibited. If the value of these bits is changed by writing, subsequent operation is unpredictable.

Note 4. If the communication format of other-party device is compatible with a communication format of SSIE, specify and use the communication format that enables communication with the other-party device.

This register is used to set an audio format (which involves the settings of communication format, LR clock/frame synchronization continuation mode, and BCK output stop).

OMOD[1:0] bits (Audio Format Select)

These bits set an audio format. Writing to these bits must be performed when the LR clock supply to the SSILRCK/SSIFS pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in reference 41.4.7.

LRCONT bit (Whether to Enable LRCK/FS Continuation)

This bit enables or disables the output from SSILRCK/SSIFS pin when the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSISR.IIRQ = 1).

Even in the idle state, a signal can output from the SSILRCK/SSIFS pin when this bit is set to 1 (to enable LR clock/frame synchronization continuation) in master mode (SSICR.MST = 1).

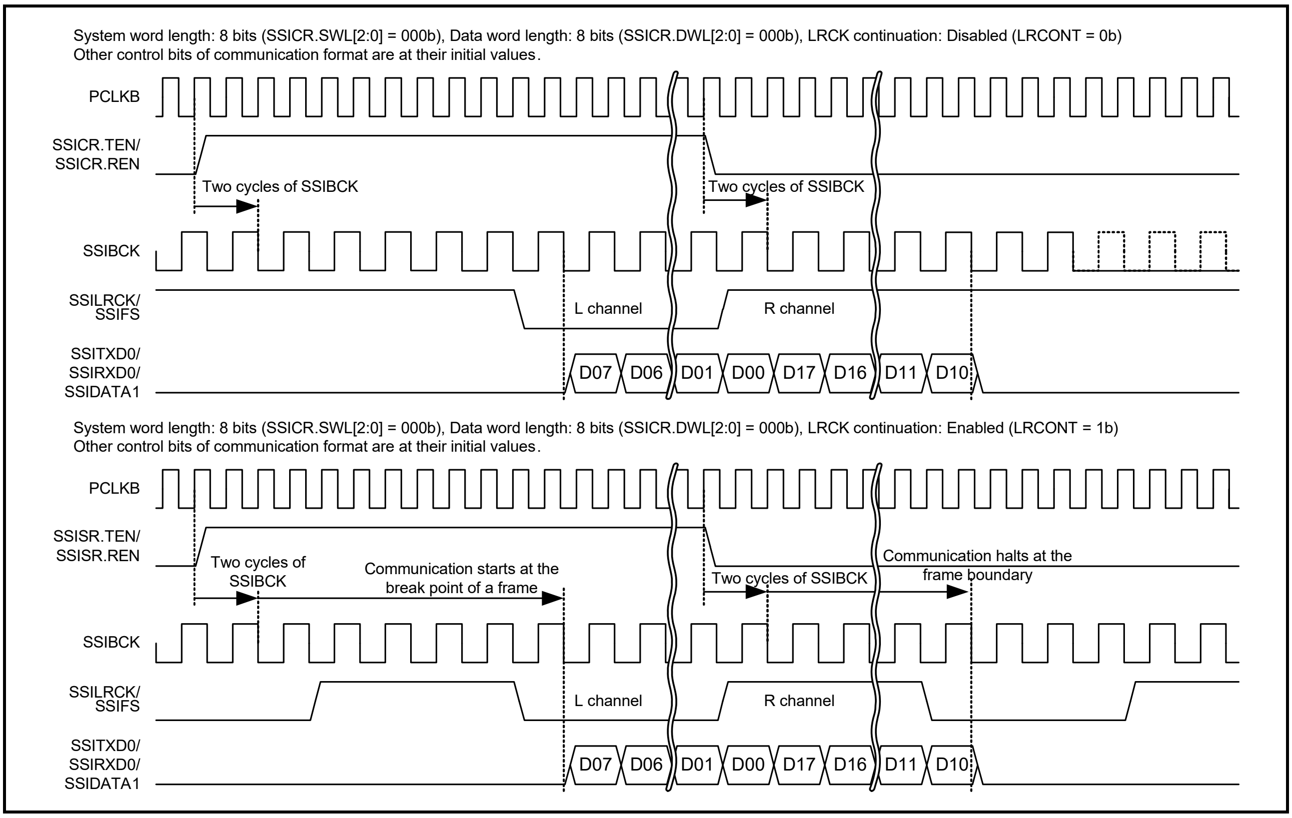


Figure 41.33 Example of LR clock/frame synchronization continuation operation

BCKASTP bit (Whether to Enable Stopping BCK Output When SSIE is in Idle Status)

This bit turns on or off the function to output BCK to the SSIBCK pin according to the communication shown in Figure 41.34 and Figure 41.35 in master-mode communication (SSICR.MST = 1).

Changing the value of this bit must be performed only after setting the communication format to be used.

This bit must be used in the following way:

Write 0 to the BCKASTP bit, and then start communication. During the communication, write 1 to the BCKASTP bit. By this operation, the bit clock output to the SSIBCK pin stops automatically when the communication stops. To resume the communication, set SSIE to the idle state (SSICR.IIRQ = 1), enable the supply of AUDIO_MCK (SSIFCR.AUCKE = 1), and then write 0 to the BCKASTP bit.

When the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSICR.IIRQ = 1):

Table 41.9 BCKASTP bit status and SSIBCK pin output

BCKASTP Bit	SSIBCK Pin Output Status
0	Output
1	Stopped

Note: The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCK pin before and during communication. In such a case, use the BCKASTP bit to stop the clock only after communication. For the timing of enabling the clock stop function, see Figure 41.34.

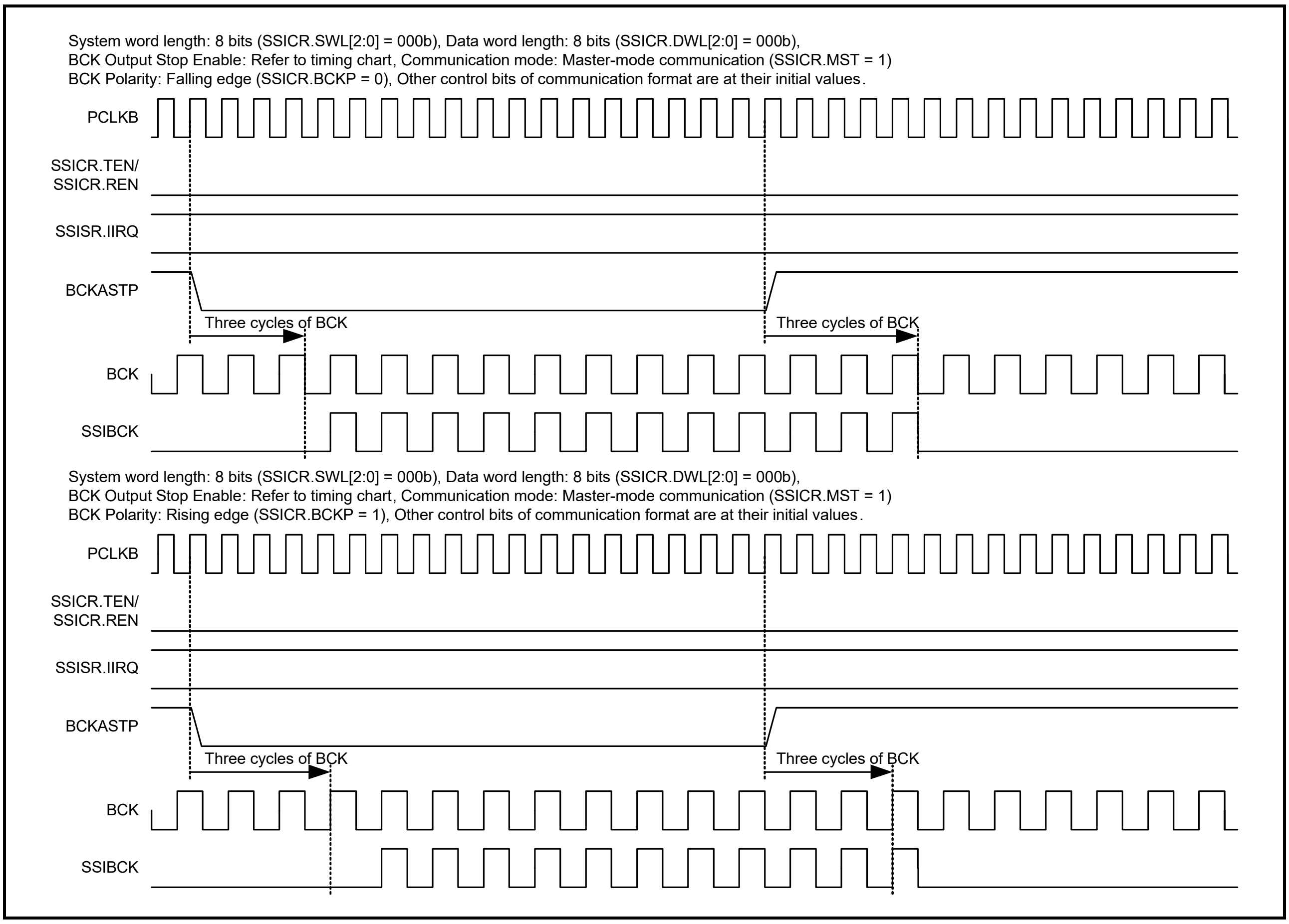


Figure 41.34 Example operation of the BCKASTP bit (idle state)

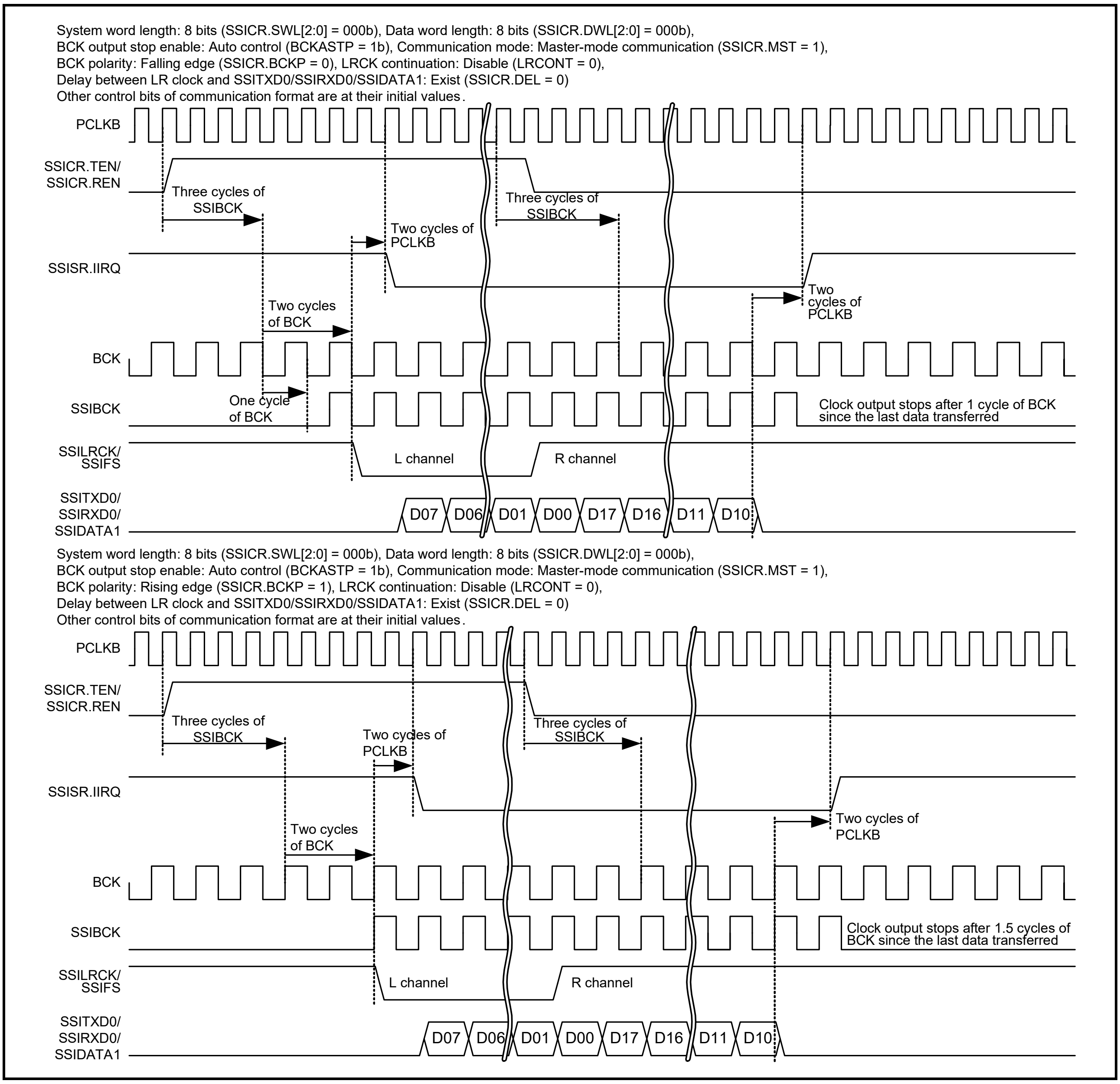
When the communication mode is master-mode communication (SSICR.MST = 1) and the BCK output stop function is enabled (BCKASTP = 1):

Details of the BCK output to the SSIBCK pin are as follows:

Output start timing: BCK is output in appropriate timing so that a valid edge is generated when the LR clock/frame synchronization signal shifts to a valid value.

Output stop timing: 1 to 1.5 clock cycles after a frame boundary.

For details about the timings, see the timing diagram in Figure 41.35.



System word length: 8 bits (SSICR.SWL[2:0] = 000b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
BCK output stop enable: Auto control (BCKASTP = 1b), Communication mode: Master-mode communication (SSICR.MST = 1),
BCK polarity: Falling edge (SSICR.BCKP = 0), LRCK continuation: Disable (LRCONT = 0),
Delay between LR clock and SSITXD0/SSIRXD0/SSIDATA1: Exist (SSICR.DEL = 0)
Other control bits of communication format are at their initial values.

System word length: 8 bits (SSICR.SWL[2:0] = 000b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
BCK output stop enable: Auto control (BCKASTP = 1b), Communication mode: Master-mode communication (SSICR.MST = 1),
BCK polarity: Rising edge (SSICR.BCKP = 1), LRCK continuation: Disable (LRCONT = 0),
Delay between LR clock and SSITXD0/SSIRXD0/SSIDATA1: Exist (SSICR.DEL = 0)
Other control bits of communication format are at their initial values.

Figure 41.35 Example operation of the BCKASTP bit (communication operation with BCKASTP = 1)

41.4.8 Status Control Register (SSISCR)

Address(es): SSIE0.SSISCR 4004 E024h, SSIE1.SSISCR 4004 E124h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	RDFS[4:0]	RDF Setting Condition Select*1	b4 b0 0 0 0 0 0: SSIFRDR has one stage or more data size 0 0 0 0 1: SSIFRDR has two stages or more data size (snip) 1 1 1 1 0: SSIFRDR has thirty-one stages or more data size 1 1 1 1 1: SSIFRDR has thirty-two stages or more data size.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	TDES[4:0]	TDE Setting Condition Select*1	b12 b8 0 0 0 0 0: SSIFTDR has one stage or more free space 0 0 0 0 1: SSIFTDR has two stages or more free space (snip) 1 1 1 1 0: SSIFTDR has thirty-one stages or more free space 1 1 1 1 1: SSIFTDR has thirty-two stages or more free space.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

RDFS[4:0] bits (RDF Setting Condition Select)

These bits set the setting condition of the receive data full flag (RDF).

TDES[4:0] bits (TDE Setting Condition Select)

These bits set the setting condition of the transmit data empty flag (TDE).

41.5 Communication Formats

SSIE supports three communication formats. Table 41.10 shows supported communication formats.

Table 41.10 Supported communication formats

Communication Format	SSIOFR.OMOD[1:0]
I^{2}S format	00
TDM format	01
Monaural format	10

The following describes the serial data structure shared by communication formats. A serial data structure is defined by the system word length (set in SSICR.SWL[2:0]) and the data word length (set in SSICR.DWL[2:0]). If the data word length is shorter than the system word length, padding bits are transferred in the serial data. For details, see Figure 41.36.

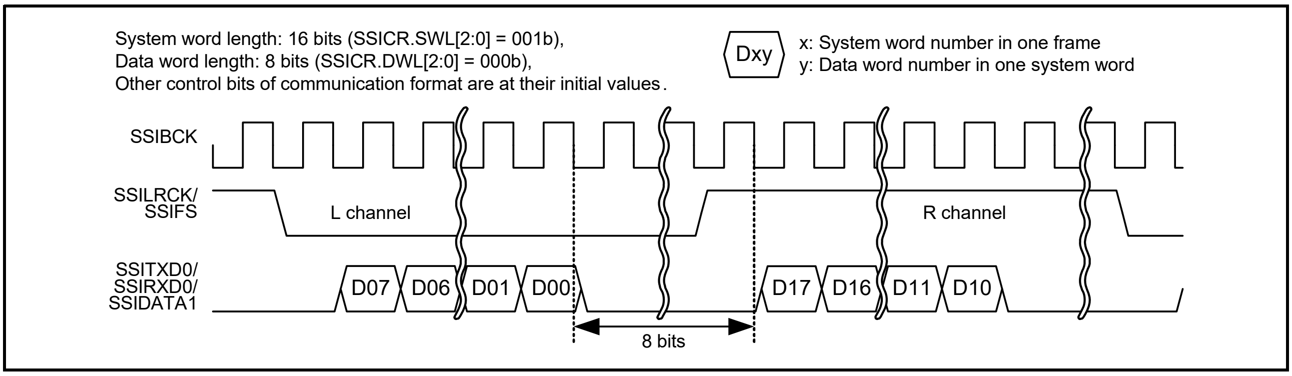


Figure 41.36 Example of padding bit transfer (I2S format: system word length > data word length)

Table 41.11 lists the number of padding bits to be transferred with each combination of system word length (SSICR.SWL[2:0]) and data word length (SSICR.DWL[2:0]). "-" indicates that the setting is prohibited.

Table 41.11 Number of padding bits

	SSICR.DWL[2:0]	**000b**	**001b**	**010b**	**011b**	**100b**	**101b**	**110b**	**111b**
SSICR.SWL[2:0]	**System Word Length**	**8**	**16**	**18**	**20**	**22**	**24**	**32**	**Setting prohibited**
000b	8	0	—	—	—	—	—	—	—
001b	16	8	0	—	—	—	—	—	—
010b	24	16	8	6	4	2	0	—	—
011b	32	24	16	14	12	10	8	0	—
100b	48	40	32	30	28	26	24	16	—
101b	64	56	48	46	44	42	40	32	—
110b	128	120	112	110	108	106	104	96	—
111b	256	248	240	238	236	234	232	224	—

41.5.1 I2S Format

The I2S format is a communication format used for connection with I2S-compatible serial devices. With this format setting (SSIOFR.OMOD[1:0] = 00b), one frame is configured with two system words, one for the channel L and the other for channel R. The SSILRCK/SSIFS signals are at a low level for the channel L and at a high level for the channel R. Set the polarity of the signals with the SSICR.LRCKP bit. Figure 41.37 shows the I2S format without padding. See Figure 41.36 for the format with padding.

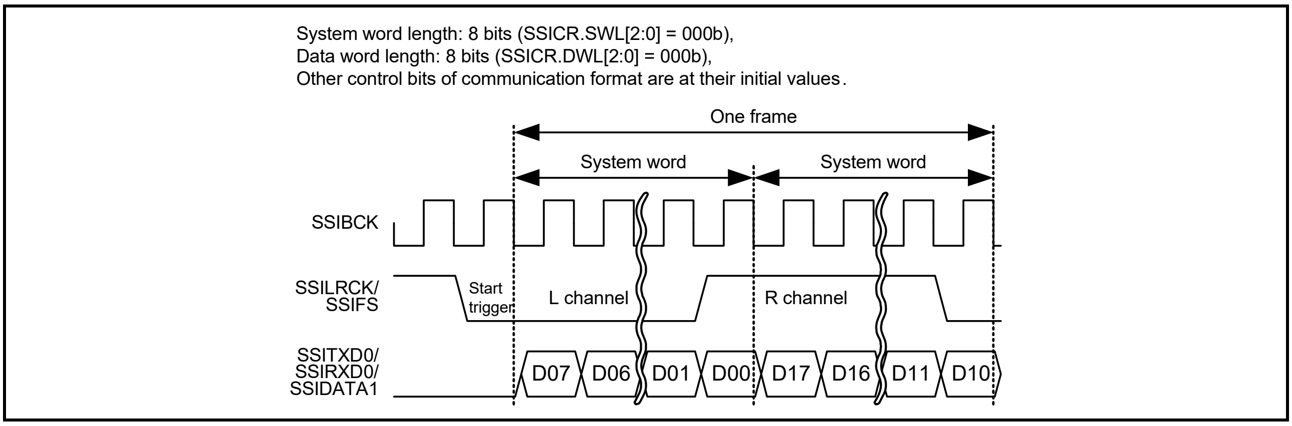


Figure 41.37 I2S format (without padding: system word length = data word length)

For the state of external pins when SSIE is in the idle state, see reference 41.7.1.

Note: SSIE has the SSILRCK/SSIFS pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCK/SSIFS pin only as a trigger to start communication.

41.5.2 Monaural Format

The monaural format is a communication format used for connection with monaural-compatible serial devices. When the monaural format is specified (SSIOFR.OMOD[1:0] = 10b) for use, one frame consists of one system word. Also, a rising edge of the SSILRCK/SSIFS signal indicates a communication start trigger. Figure 41.38 and Figure 41.39 respectively show the monaural formats without and with padding.

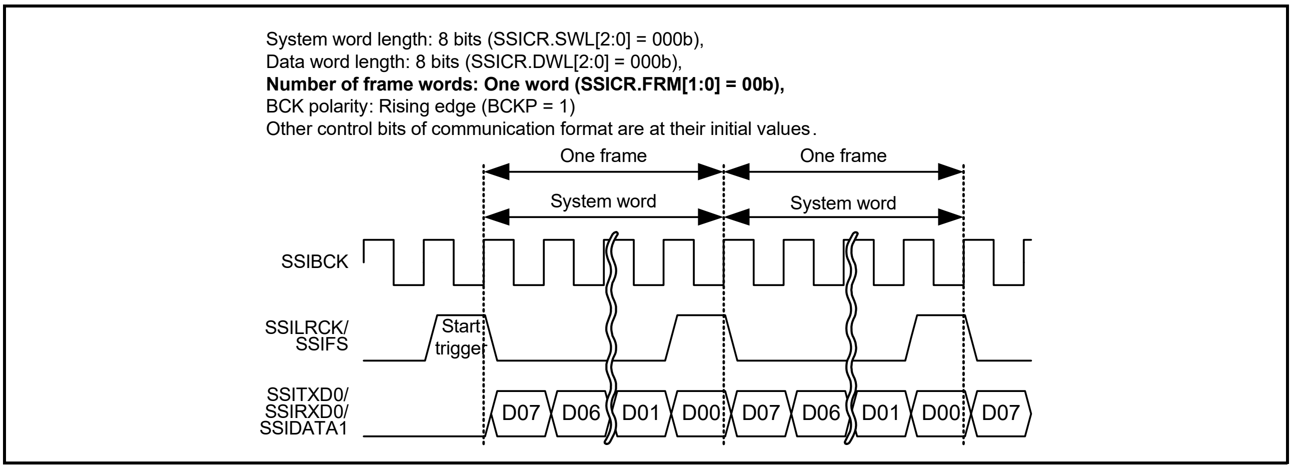


Figure 41.38 Short frame in monaural format (without padding: system word length = data word length)

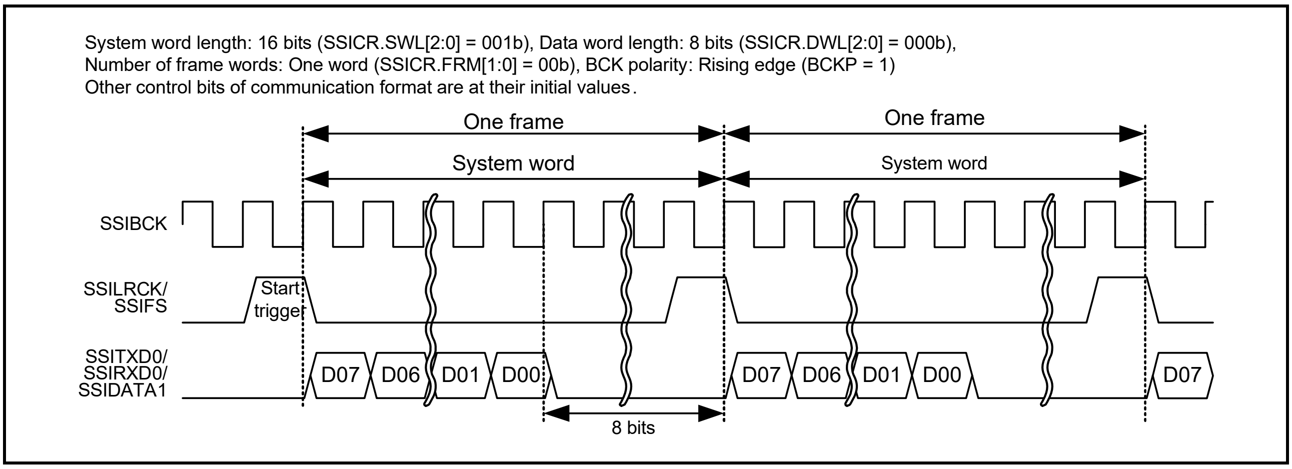


Figure 41.39 Short frame in monaural format (with padding: system word length > data word length)

The monaural formats supported by SSIE consist of short frames and long frames. See reference 41.5.2.1 and reference 41.5.2.2 for the difference between these two frames.

For the state of external pins state when SSIE is in the idle state, see reference 41.7.1.

Note: SSIE has the SSILRCK/SSIFS pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCK/SSIFS pin only as a trigger to start communication.

41.5.2.1 Short frame

When a short frame is used (SSICR.DEL = 0), the SSILRCK/SSIFS signal indicating the start of serial data is set to high level only for 1 cycle of SSIBCK. Data transfer starts at the falling edge of the signal.

41.5.2.2 Long frame

When a long frame is used (SSICR.DEL = 1), the SSILRCK/SSIFS signal indicating the start of serial data is set to high level only for 2 cycles of SSIBCK. See Figure 41.40. Data transfer starts at the rising edge of the signal.

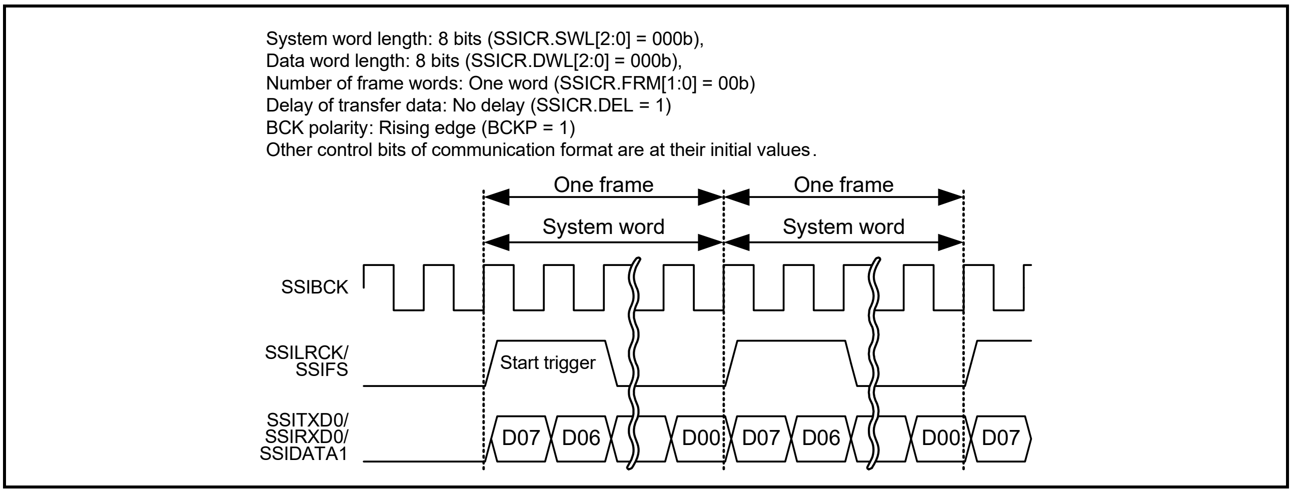

System word length: 8 bits (SSICR.SWL[2:0] = 000b),
Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
Number of frame words: One word (SSICR.FRM[1:0] = 00b)
Delay of transfer data: No delay (SSICR.DEL = 1)
BCK polarity: Rising edge (BCKP = 1)
Other control bits of communication format are at their initial values.

Figure 41.40 Long frame in monaural format (without padding)

41.5.3 TDM Format

The TDM format is a communication format used for connection with TDM-compatible multi-channel devices. With this format setting (SSIOFR.OMOD[1:0] = 01b), one frame is configured with four to eight system words set with the SSICR.FRM[1:0] bits. With this format, the SSILRCK/SSIFS signal is at a high level for the first one system word and at a low level for the rest. The pulse generated on the SSILRCK/SSIFS signal is defined as the SYNC pulse and its rising edge means a start of one frame. Figure 41.41 and Figure 41.42 respectively show the TDM formats without and with padding.

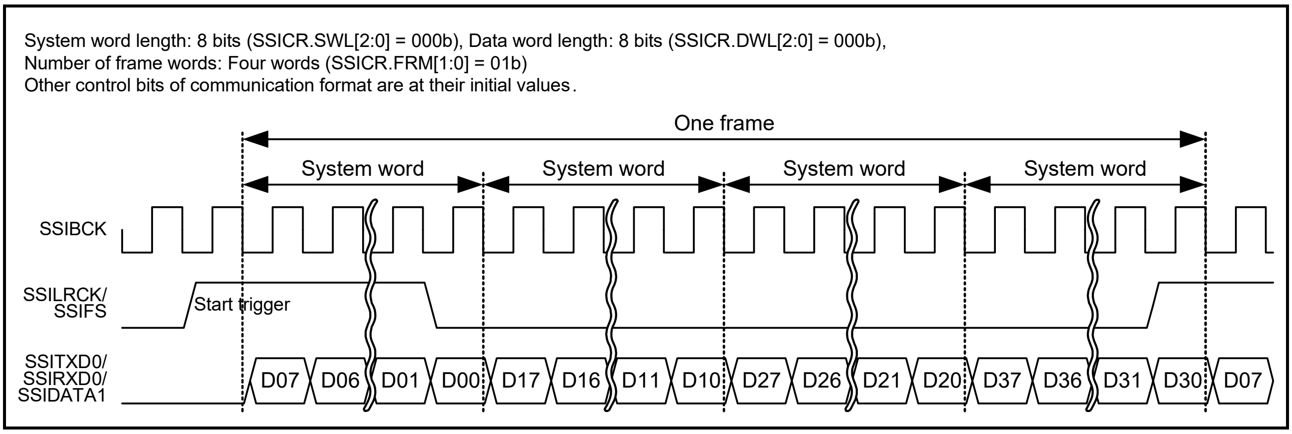

System word length: 8 bits (SSICR.SWL[2:0] = 000b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b),
Number of frame words: Four words (SSICR.FRM[1:0] = 01b)
Other control bits of communication format are at their initial values.

Figure 41.41 TDM format (without padding: system word length = data word length)

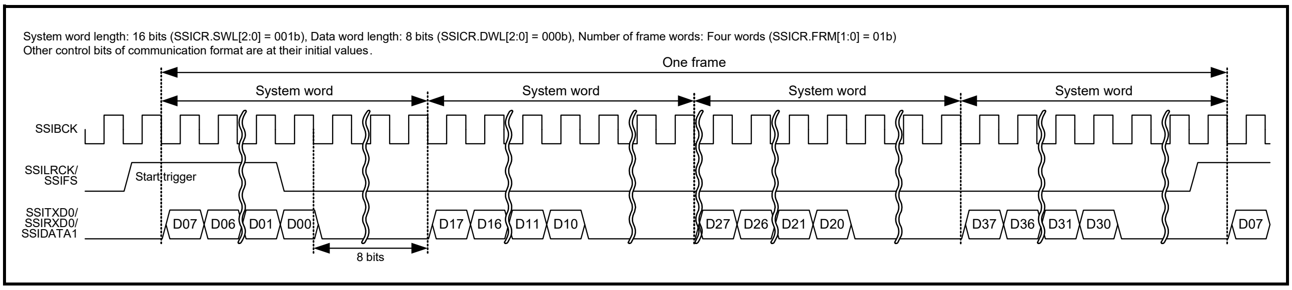

System word length: 16 bits (SSICR.SWL[2:0] = 001b), Data word length: 8 bits (SSICR.DWL[2:0] = 000b), Number of frame words: Four words (SSICR.FRM[1:0] = 01b)
Other control bits of communication format are at their initial values.

Figure 41.42 TDM format (with padding: system word length > data word length)

For the state of external pins when SSIE is in the idle state, see reference 41.7.1.

Note: SSIE has the SSILRCK/SSIFS pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCK/SSIFS pin only as a trigger to start communication.

41.6 Communication Modes

SSIE supports the following communication modes. Table 41.13 lists the control bits that are not available with each communication mode. See reference 41.6.1 to reference 41.6.5 for details of these communication modes.

Table 41.12 Communication modes

Communication Mode	SSICR.MST Bit	SSICR.REN Bit	SSICR.TEN Bit
Slave-mode transmission	0	0	1
Slave-mode reception	0	1	0
Slave-mode transmission and reception	0	1	1
Master-mode transmission	1	0	1
Master-mode reception	1	1	0
Master-mode transmission and reception	1	1	1

Table 41.13 Control bits that cannot be used in each communication mode

Communication Mode Control Bit	Slave-mode Reception	Slave-mode Transmission	Slave-mode Transmission and Reception	Master-mode Reception	Master-mode Transmission	Master-mode Transmission and Reception
SSICR.CKS	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.CKDV	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.MUEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.TEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.REN	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.AUCKEN	Invalid	Invalid	Invalid	Available	Available	Available
SSIFCR.TIE	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RIE	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.TFRST	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RFRST	Available	Invalid	Available	Available	Invalid	Available
SSIOFR.BCKASTP	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.LRCONT	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.OMOD	Available	Available	Available	Available	Available	Available
SSISCR.TDES	Invalid	Available	Available	Invalid	Available	Available
SSISCR.RDFS	Available	Invalid	Available	Available	Invalid	Available

"Invalid" means it has no effect on operation. Writing is possible.

41.6.1 Slave-mode Communication

SSIE operates in slave mode with SSICR.MST = 0. The SSIBCK and SSILRCK/SSIFS signals to be used for serial-data communication must be supplied from an external device. If these signals do not match the communication format set for SSIE, operation is not guaranteed.

41.6.2 Master-mode Communication

SSIE operates in master mode with SSICR.MST = 1. The SSIBCK and SSILRCK/SSIFS signals to be used for serial-data communication must be internally generated from the audio clock. These signals use the format according to the setting of SSIE. If the communication format the slave device uses does not match the communication format set for SSIE, the operation is unpredictable.

41.6.3 Transmission

SSIE transmits serial data to the other-party device when the SSICR.TEN bit is 1 and the SSICR.REN bit is 0. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

41.6.4 Reception

SSIE receives serial data from the other-party device when the SSICR.TEN bit is 0 and the SSICR.REN bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

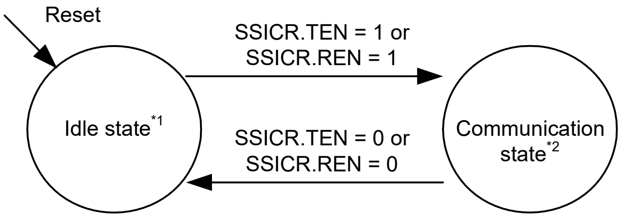
41.6.5 Transmission and Reception

SSIE transmits and receives serial data to and from the other-party device when the SSICR.TEN bit is 1 and the SSICR.REN bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

41.7 Operation

SSIE has the following two main operation states Figure 41.43 shows SSIE state transition.

- Idle state (SSISR.IIRQ = 1)
- Communication state (SSISR.IIRQ = 0).



Note 1. See reference 41.8.1 for details of the idle state.
Note 2. See reference 41.8.2 for details of the communication state.

Figure 41.43 SSIE state transition

41.7.1 Idle State

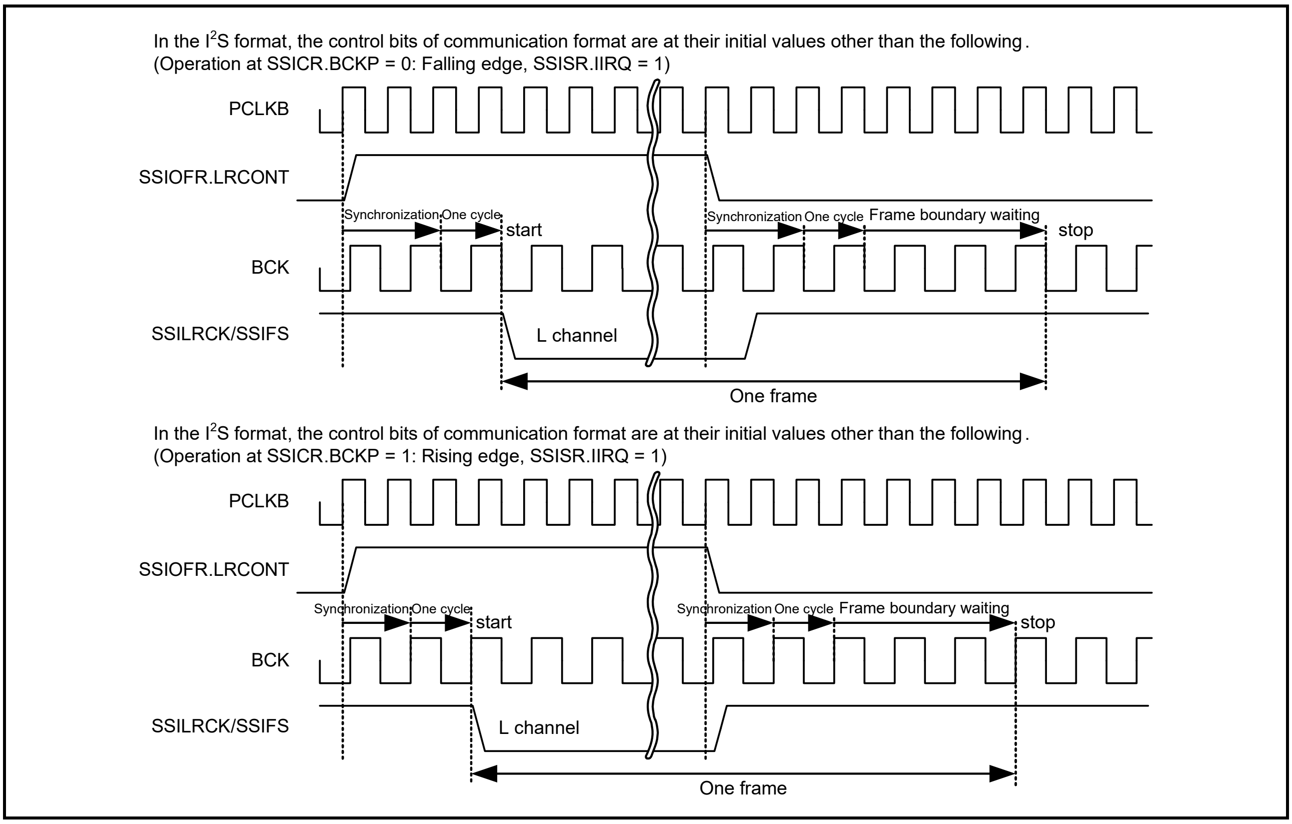
In this state, communication of SSIE is halted. If, however, the SSICR.MST bit is 1, output of the BCK and LR clock/frame synchronization signals to external pins can be controlled according to the settings of SSIOFR.BCKASTP and SSIOFR.LRCONT bits. This function is common to all formats. For details, see Table 41.14.

Table 41.14 Output from external pins in the idle state

			Output from Pins		
SSICR.MST	SSIOFR.BCKASTP	SSIOFR.LRCONT	SSIBCK	SSILRCK/SSIFS	SSITXD0/SSIDATA1
0	—	—	Stop	Stop	Stop
1	0	0	Supply	Stop	Stop
1	0	1	Supply	Supply	Stop
1	1	0	Stop	Stop	Stop

Table 41.14 Output from external pins in the idle state

			Output from Pins		
SSICR.MST	**SSIOFR.BCKASTP**	**SSIOFR.LRCONT**	**SSIBCK**	**SSILRCK/SSIFS**	**SSITXD0/SSIDATA1**
1	1	1	Stop	Supply	Stop



In the I²S format, the control bits of communication format are at their initial values other than the following.
(Operation at SSICR.BCKP = 0: Falling edge, SSISR.IIRQ = 1)

In the I²S format, the control bits of communication format are at their initial values other than the following.
(Operation at SSICR.BCKP = 1: Rising edge, SSISR.IIRQ = 1)

Figure 41.44 Example of disabling LR clock/frame synchronization continuation by SSIOFR.LRCONT

Note: To stop the output to the SSILRCK/SSIFS pin with SSIOFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected.

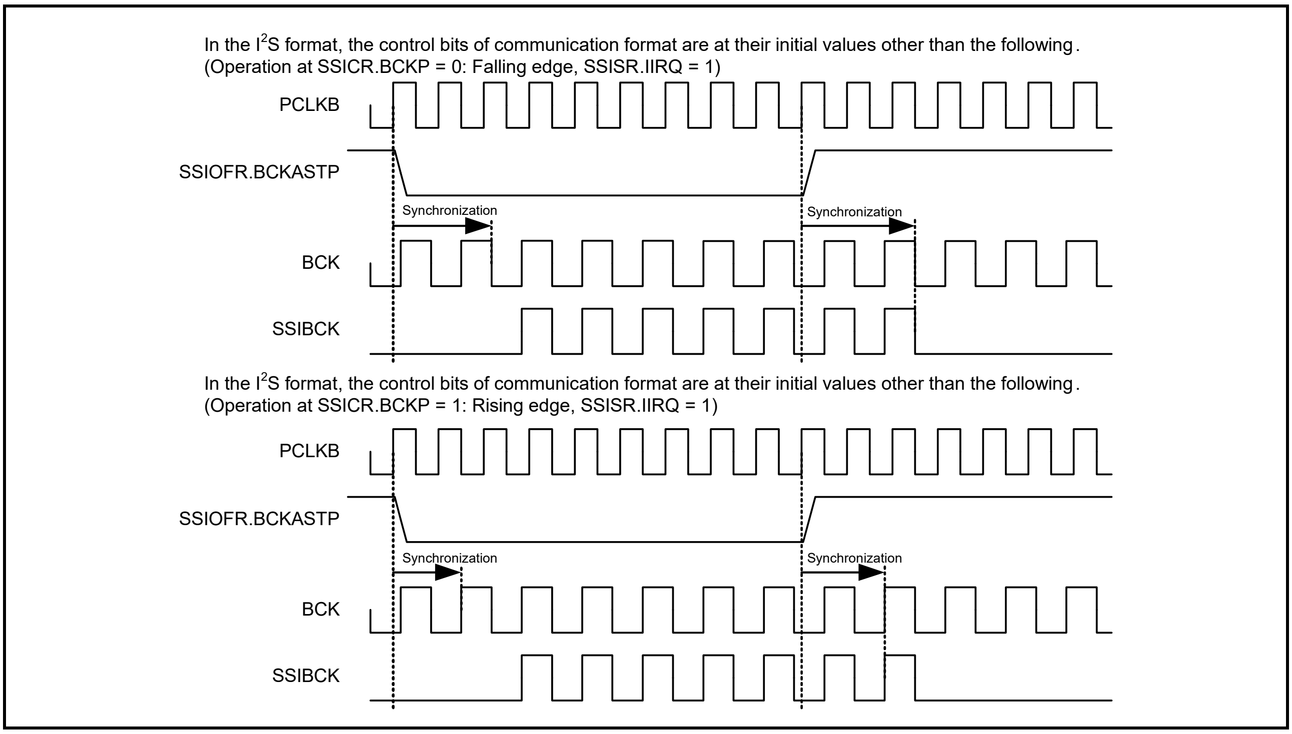


Figure 41.45 Example of stopping SSIBCK with SSIOFR.BCKASTP

Note: To stop the output to the SSIBCK pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected.

41.7.2 Communication States

In this state, SSIE is during communication. Figure 41.46 shows transitions of communication states and Table 41.15 lists the conditions for transition. If the transition condition is not satisfied, the state does not transit.

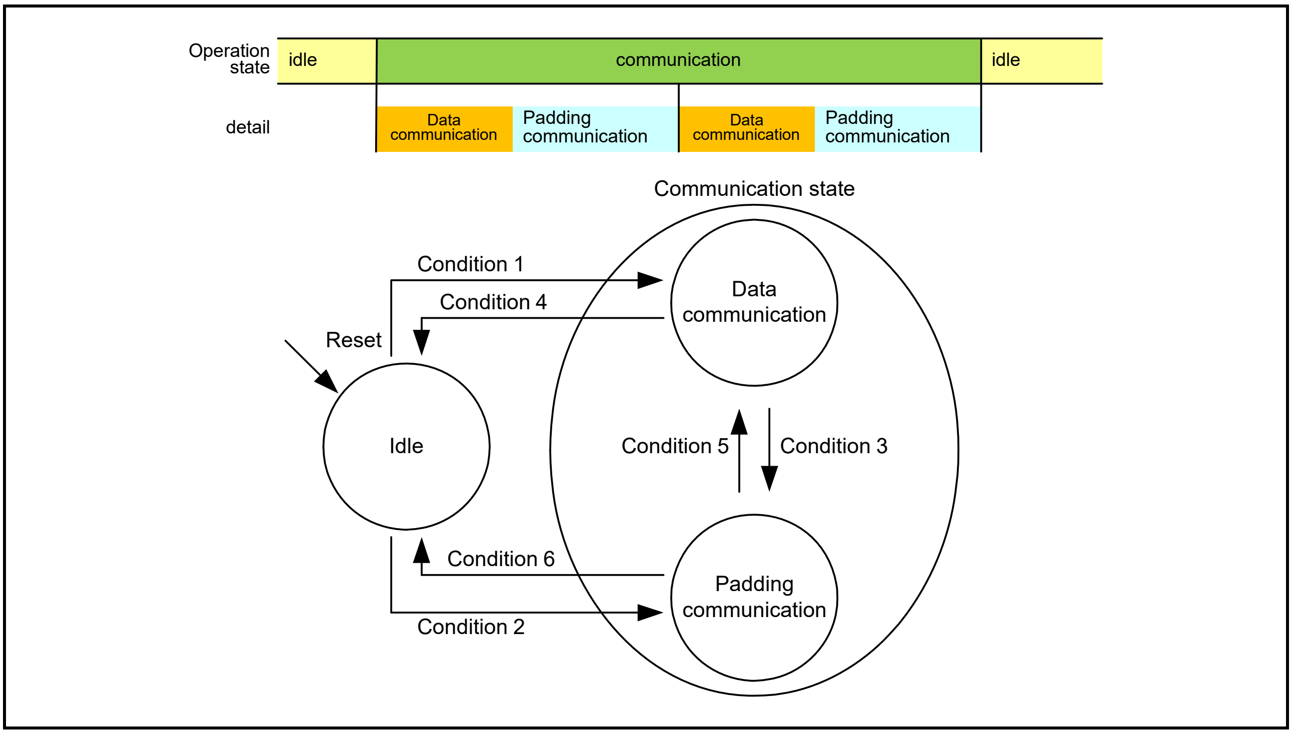


Figure 41.46 **Communication state transition**

Table 41.15 **Condition for communication state transition**

Condition Number	Condition for Transition
1	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 0 or in the setting without padding bits.
2	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 1 and in the setting with padding bits.
3	The following three conditions are all met: • SSICR.TEN = 1 or SSICR.REN = 1 • In the setting with padding bits • The last bit of the data words has been transferred.
4	Both the following two conditions are met: • SSICR.SDTA = 1 or without padding bits • While SSICR.TEN = 0 and SSICR.REN = 0, the last bit of the data words in a frame has been transferred.
5	Transfer of the last padding bit is completed while SSICR.TEN = 1 or SSICR.REN = 1
6	Both the following two conditions are met: • SSICR.SDTA = 0 and with padding bits • While SSICR.TEN = 0 and SSICR.REN = 0, the last padding bit has been transferred.

See Table 41.11 for the setting with/without padding bits.

41.7.2.1 Data communication state

In this state, SSIE is during communication. Data of the data word length set with the SSICR.DWL[2:0] bits is transmitted, received, or transmitted and received.

- State Transition in the Setting without Padding Bits

During communication (SSISR.IIRQ = 0), SSIE is during data communication for all the time. By disabling transmission and reception (SSICR.TEN = 0, SSICR.REN = 0), SSIE transits to the idle state. For details, see Figure 41.47 and Figure 41.48.

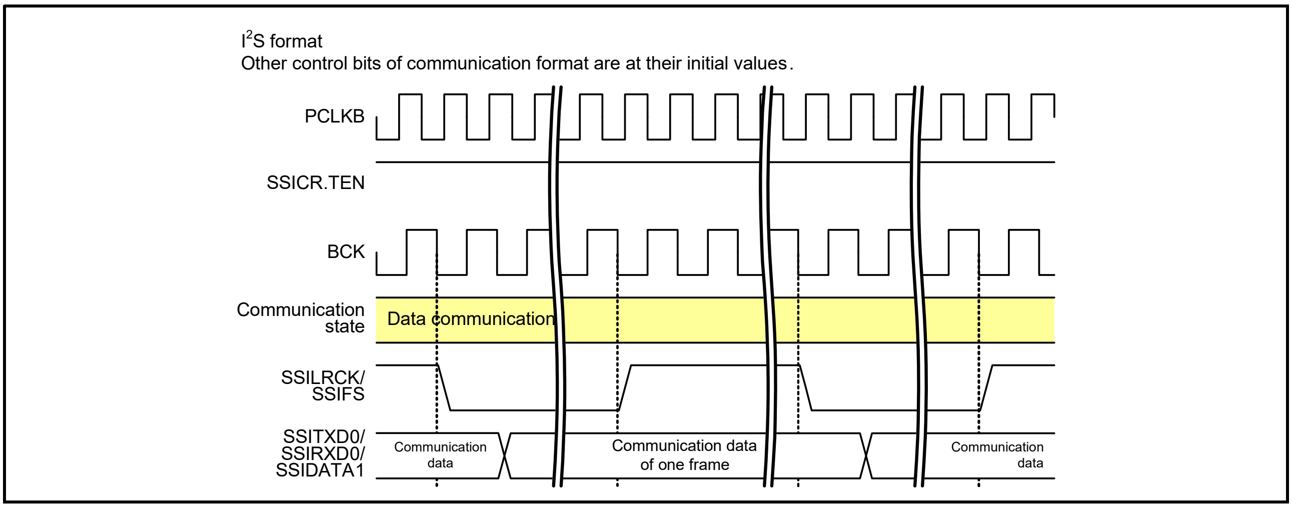


Figure 41.47 Continuation of the data communication

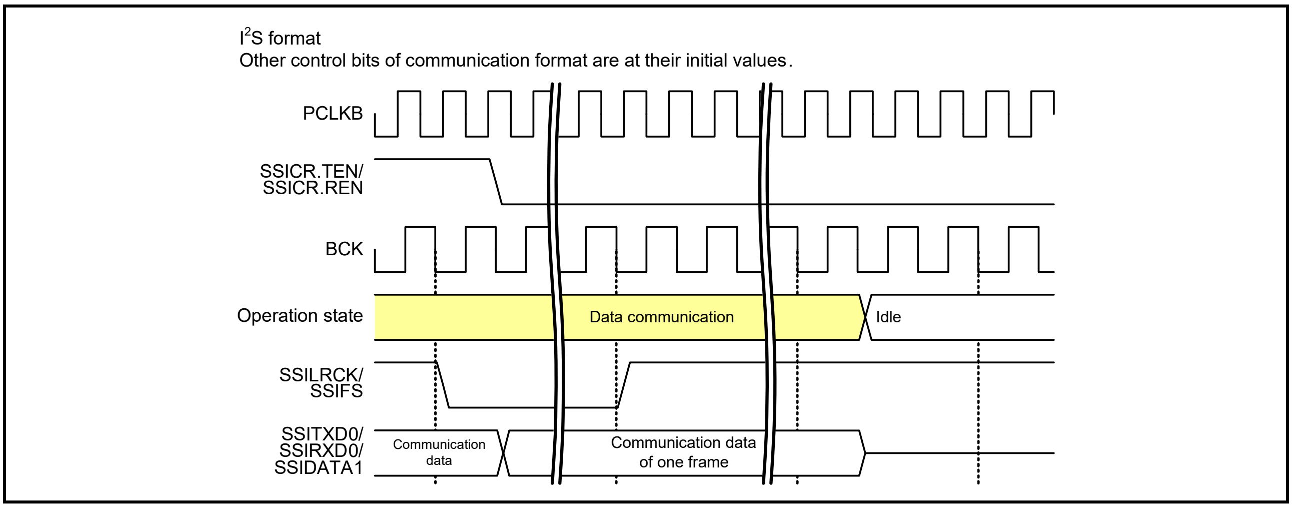


Figure 41.48 Halt from the data communication (without padding bits)

- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last bit of a data word during communication (SSISR.IIRQ = 0), SSIE transitions from the data communication state to the padding communication state in Figure 41.49. Except in the status with SSICR.SDTA = 1 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the data communication state to the idle state when it stops communication in Figure 41.51.

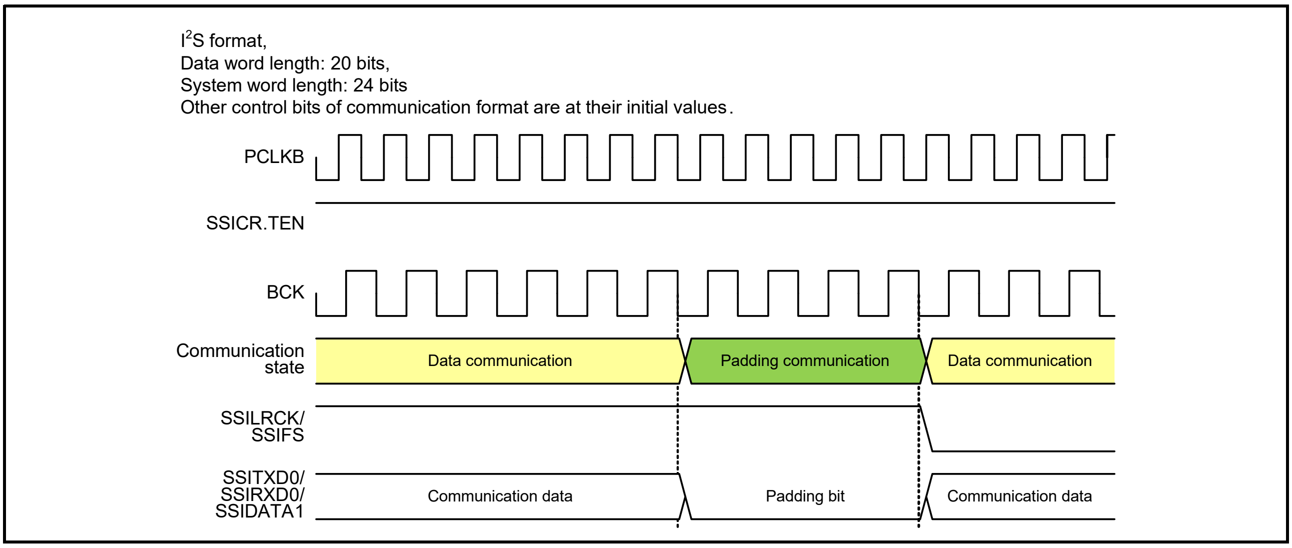


Figure 41.49 Transition from data communication to padding communication

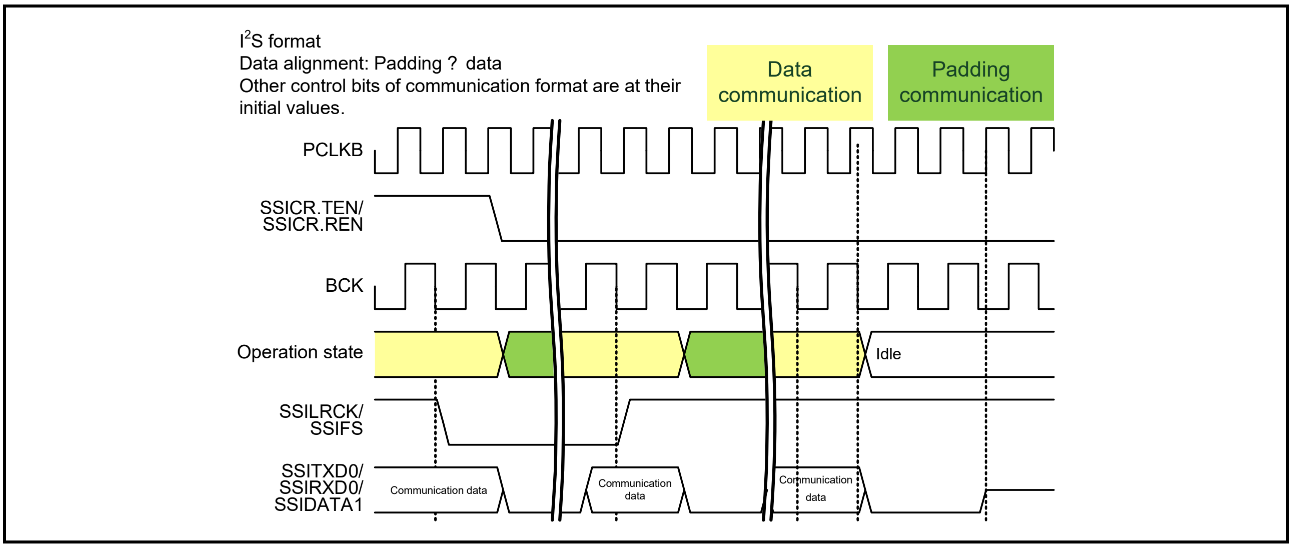


Figure 41.50 Halt from data communication (with padding bits)

41.7.2.2 Padding communication

In this state, SSIE is during communication. The padding bits set with the SSICR.SWL[2:0] bits and SSICR.DWL[2:0] bits are transmitted, received, or transmitted and received.

- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last padding bit during communication (SSISR.IIRQ = 0), SSIE transitions to the data communication state in Figure 41.49. If SSIE is in the status with SSICR.SDTA = 0 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the padding communication state to the idle state when it stops communication in Figure 41.51.

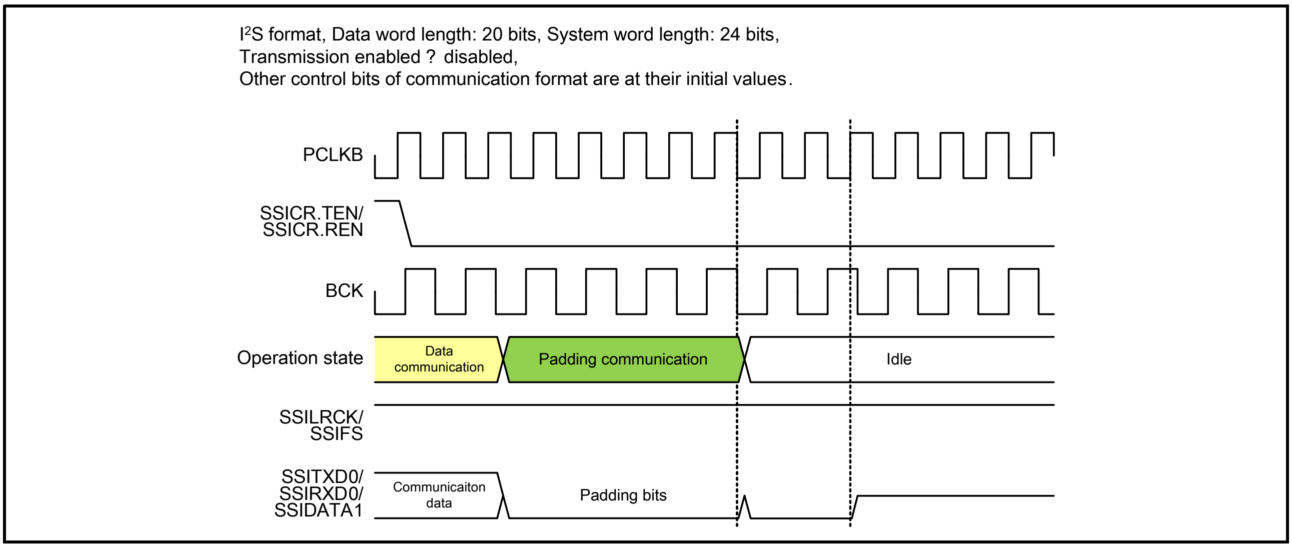


Figure 41.51 Halt from the padding communication

41.8 Communication Operation

Figure 41.52 shows the communication flow of SSIE.

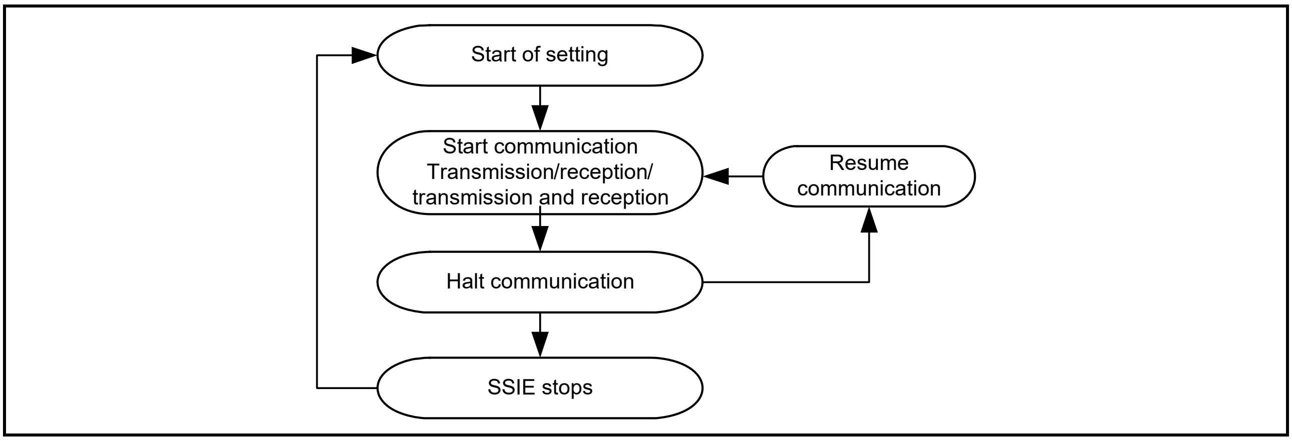
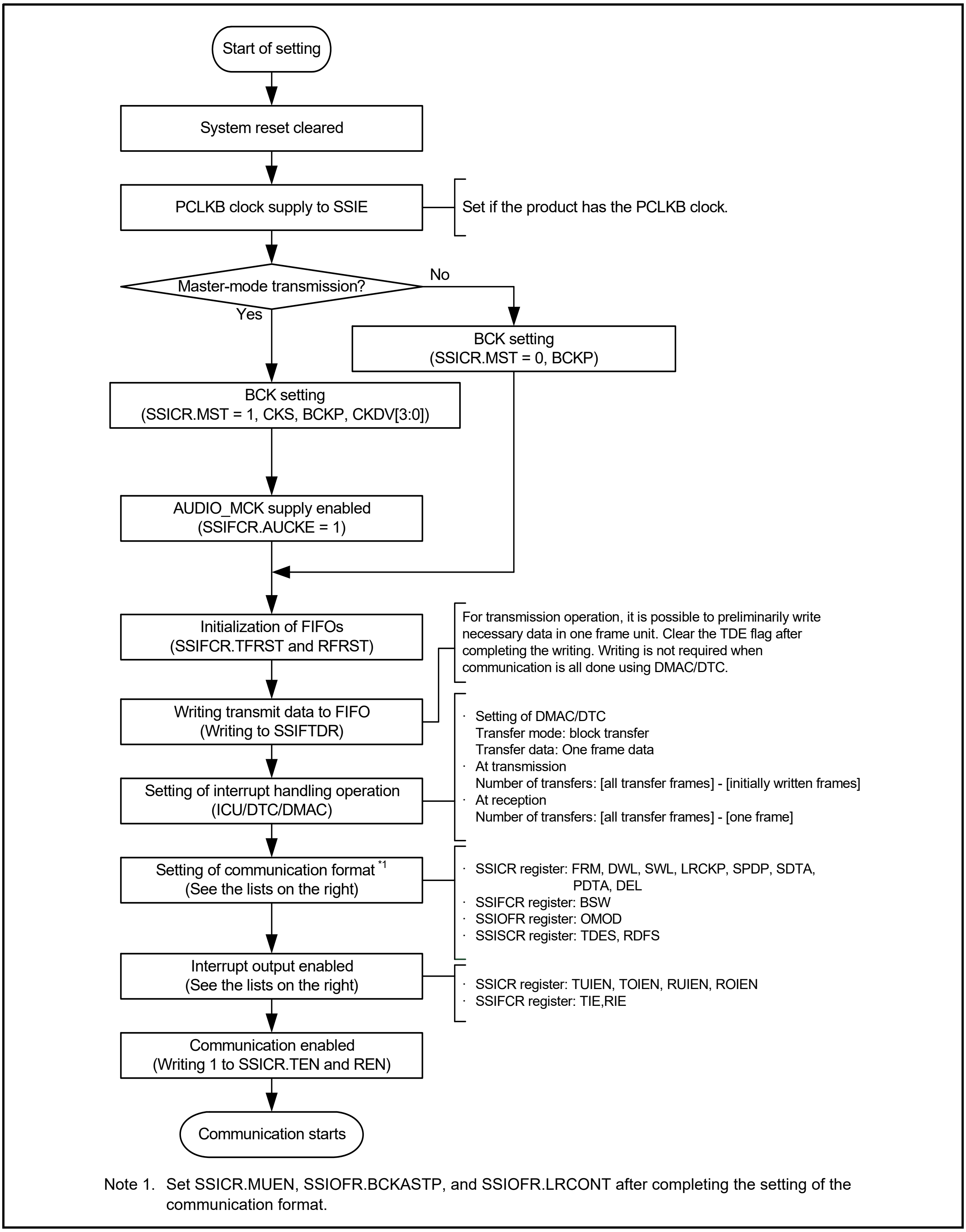


Figure 41.52 SSIE communication operation

The procedure of each operation is described from reference 41.8.1 to reference 41.8.7.

41.8.1 Start Communication

This section describes how to start communication of SSIE. Figure 41.53 shows the procedure to start communication. Be sure to follow the procedure. See reference 41.8.2 for transmission operation and reference 41.8.3 for reception operation.



Note 1. Set SSICR.MUEN, SSIOFR.BCKASTP, and SSIOFR.LRCONT after completing the setting of the communication format.

Figure 41.53 Procedure to start communication (CPU operation procedure)

SSIE can perform continuous communication based on interrupts by the DTC/DMAC. For transmission, write 1 to SSIFCR.TIE, SSICR.TUIEN, and SSICR.TOIEN. For reception, write 1 to SSIFCR.RIE, SSICR.RUIEN, and SSICR.ROIEN.

41.8.2 Transmission

The transmission procedure in Figure 41.54 must be followed throughout a transmission operation.

After transmission is enabled (SSICR.TEN = 1 and SSICR.REN = 0), SSIE starts transmission when a start trigger is generated by SSILRCK/SSIFS with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE outputs a transmit data empty interrupt to the DTC/DMAC according to the TDE setting condition (SSISCR.TDES) and the status of transmit data empty interrupt enable (SSIFCR.TIE) bit specified in the communication start procedure. This interrupt requests writing to the transmit FIFO data register (SSIFTDR). In the communication start procedure, specify writing to the transmit FIFO data register (SSIFTDR) as the DTC/DMAC operation in response to the transmit data empty interrupt. With this setting, SSIE can continuously transmit data not through the CPU. The transmit data empty interrupt is generated when the free space size of transmit FIFO data register reaches the value set in SSISCR.TDES. The number of times of writing must be specified in accordance with the free space size of the transmit FIFO data register indicated by the transmit data empty interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.

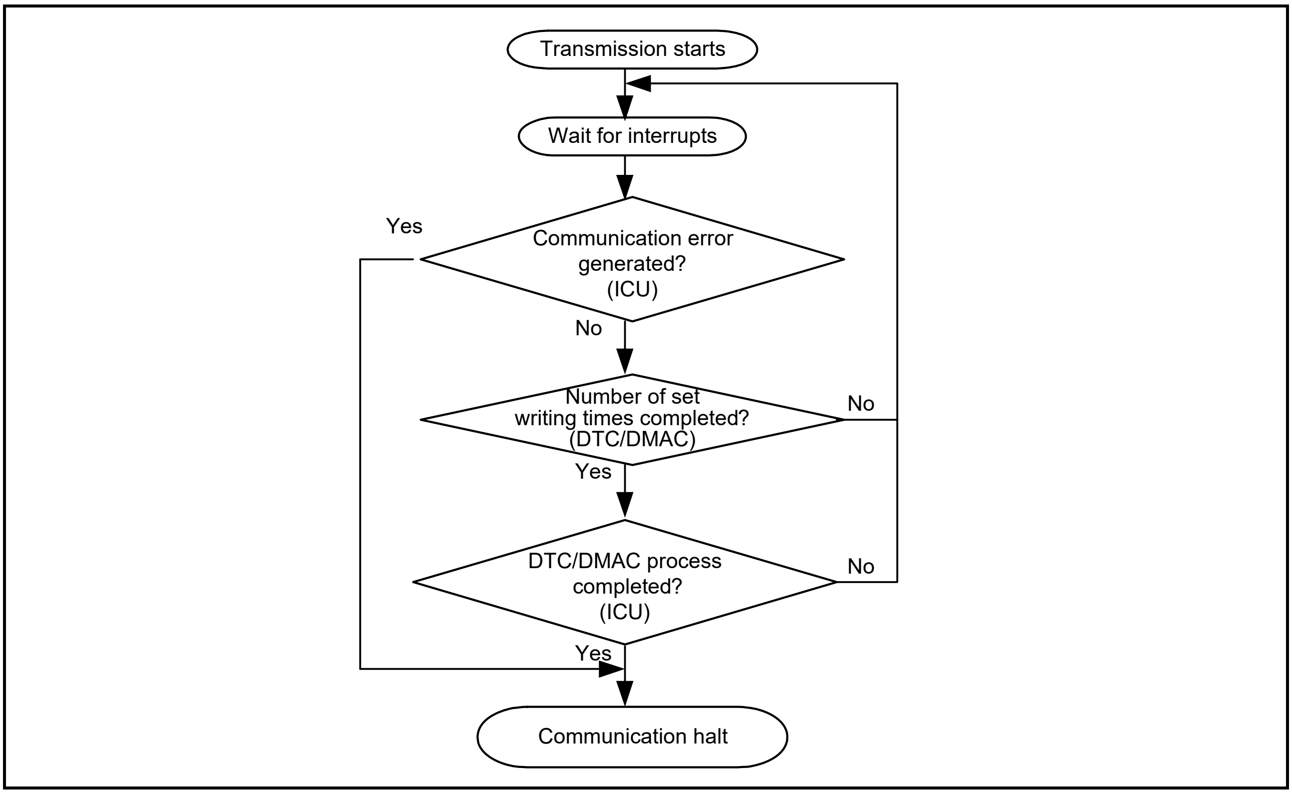


Figure 41.54 Transmission procedure

Note: The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

41.8.3 Reception

The reception procedure in Figure 41.55 must be followed throughout a reception operation.

After reception is enabled (SSICR.TEN = 0 and SSICR.REN = 1), SSIE starts reception when a start trigger is generated by SSILRCK/SSIFS. SSIE outputs a receive data full interrupt to the DTC/DMAC according to the RDF setting condition (SSISCR.RDFS) and the status of receive data full interrupt enable (SSIFCR.RIE) bit specified in the communication start procedure. This interrupt requests data reading from the receive FIFO data register (SSIFRDR). In the communication start procedure, specify reading from the receive FIFO data register (SSIFRDR) as the DTC/DMAC operation in response to the receive data full interrupt. With this setting, SSIE can continuously read data not through the CPU. The receive data full interrupt is generated when data as much as the capacity of receive FIFO data register has been stored. The number of times of reading must be specified in accordance with the data size of the receive FIFO data register indicated by the receive data full interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.

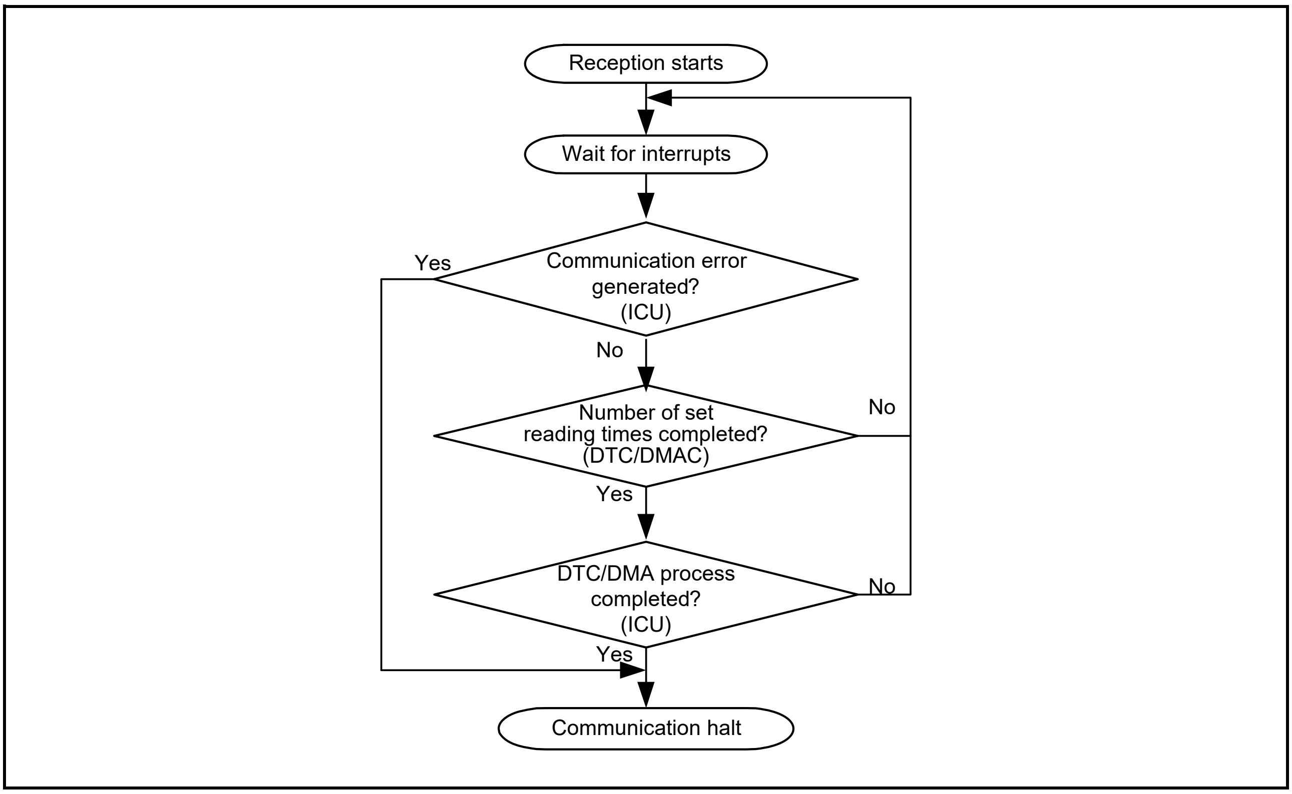


Figure 41.55 Reception procedure

Note: The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

41.8.4 Transmission and Reception

After transmission and reception are enabled (SSICR.TEN = 1 and SSICR.REN = 1), SSIE starts transmission and reception when a start trigger is generated by SSILRCK/SSIFS with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE can continuously transmit and receive data by performing the procedures described in reference 41.8.2 and reference 41.8.3, respectively. For how to stop transmission and reception, see reference 41.8.5.

41.8.5 Halt Communication

This section describes how to halt communication of SSIE. Figure 41.56 shows the procedure to halt communication. Be sure to follow the procedure.

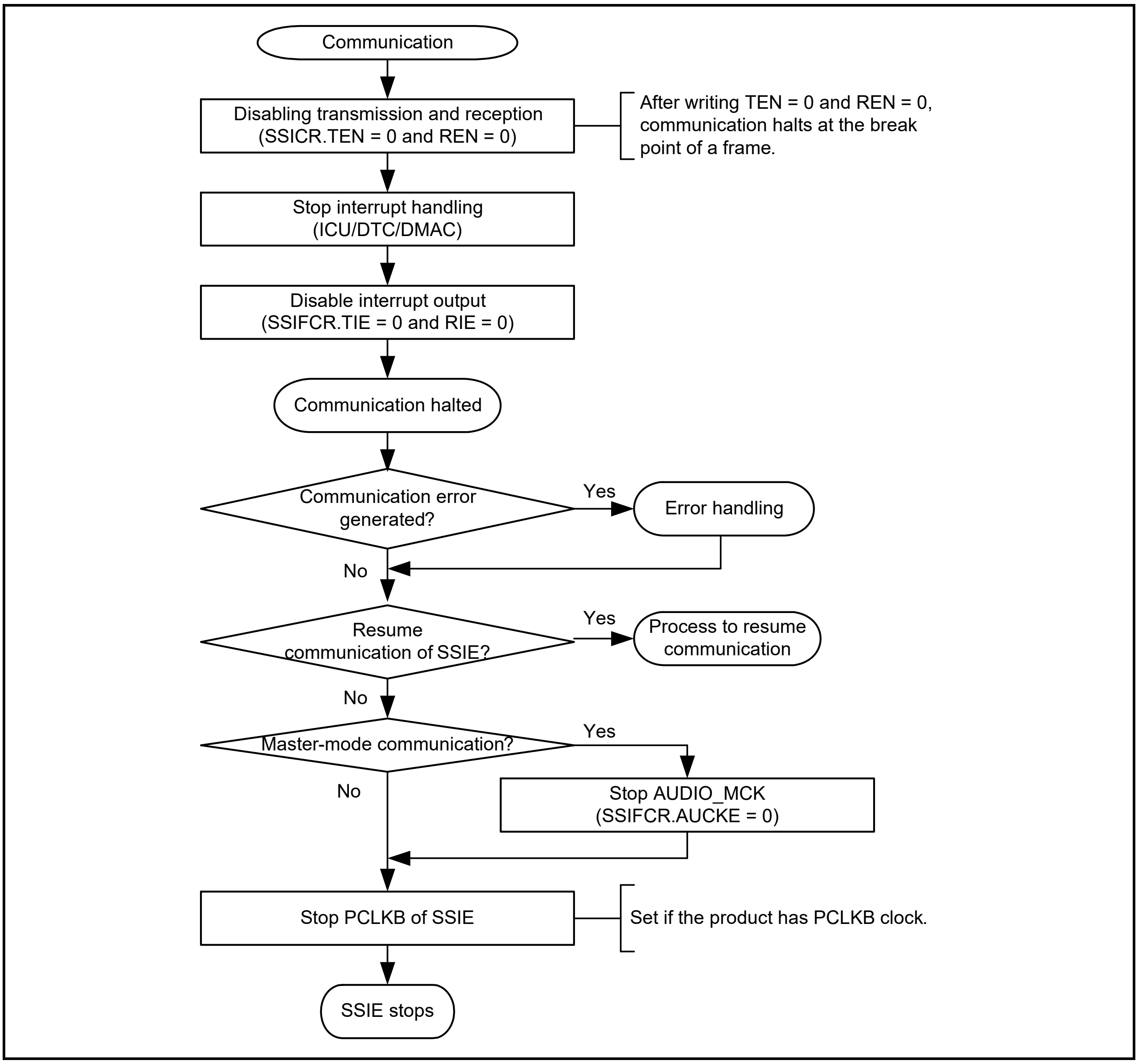


Figure 41.56 Procedure to halt communication (CPU operation procedure)

To halt the communication of SSIE, supply of the following clocks are required until the SSISR.IIRQ bit indicates an idle state.

- Input clock from the SSIBCK pin when SSICR.MST = 0
- AUDIO_MCK when SSICR.MST = 1

To resume communication of SSIE in the previous setting, see reference 41.8.7.

Note: When communication of SSIE is halted according to the procedure to halt communication in Figure 41.56, resume communication according to the procedure to resume communication in Figure 41.58.

41.8.6 Error Handling

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error

- Receive overflow error.

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

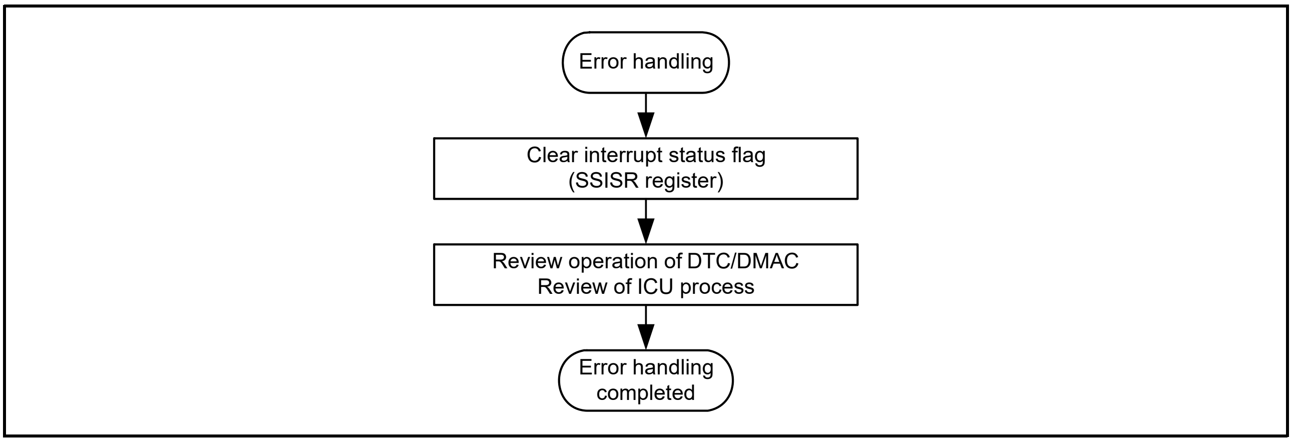


Figure 41.57 Error-handling procedure

Four error operations are described as follows. When the interrupt output enable bit of the SSICR register is enabled and error flags are set, an error interrupt is generated. See descriptions of flags in reference 41.4.2 for the setting conditions of error flags.

(1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57. After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

(2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the transmit FIFO data register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57. When you resume communication, deal with the invalid serial data appropriately.

(3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

(4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

41.8.7 Resume Communication

When you resume the communication using SSIE, follow the communication resume procedure in Figure 41.58. The communication resume procedure is designed on the assumption that you resume the communication stopped by the communication stop procedure without changing any settings. If you want to change clock and slave/master settings, use and follow the communication start procedure in Figure 41.53. For details about the transmission operation and reception operation after starting communication, see reference 41.8.2 and reference 41.8.3, respectively.

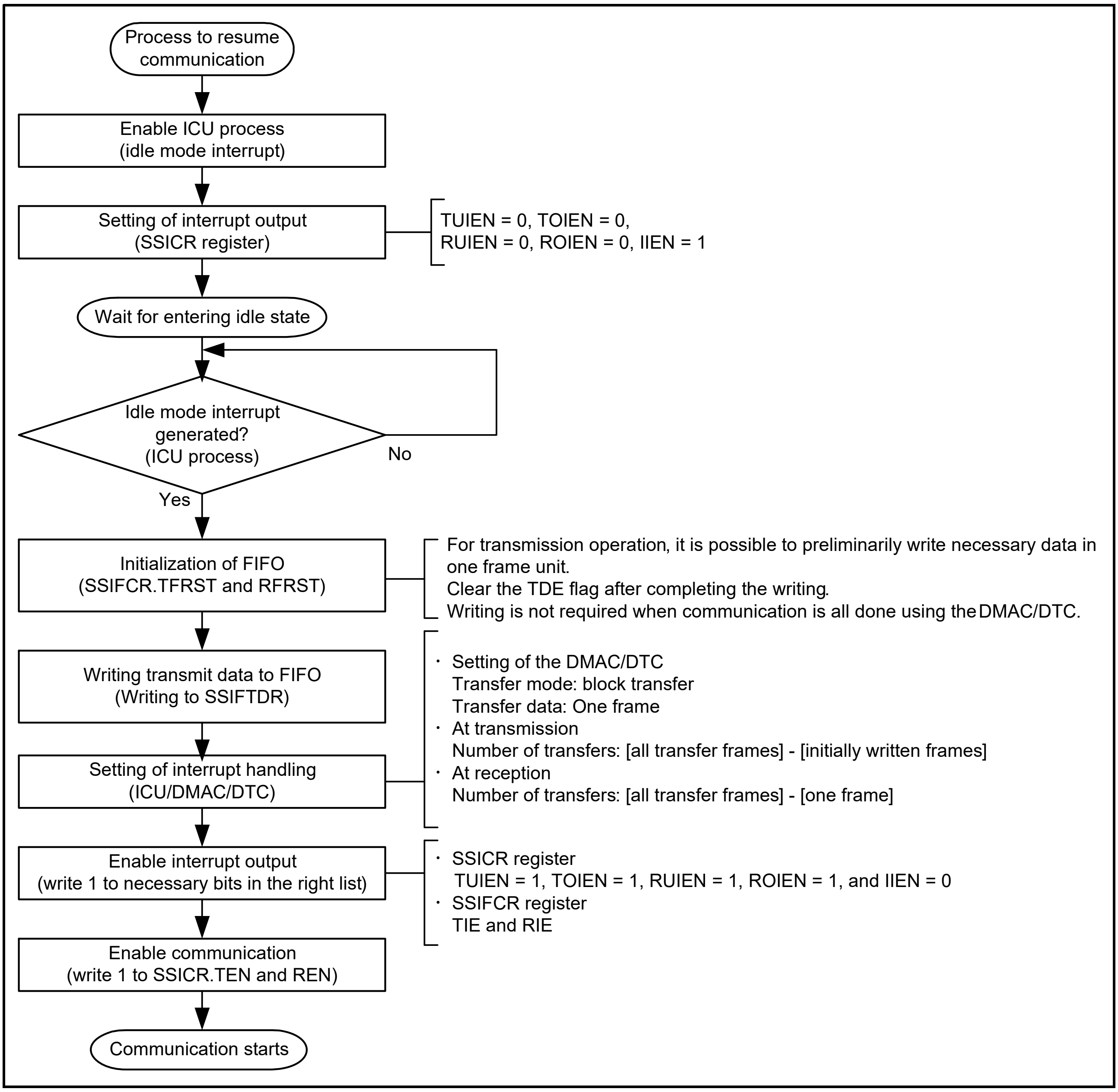


Figure 41.58 Procedure to resume communication (CPU operation procedure)

41.9 Interrupts

Table 41.16 lists the interrupt sources. Set enable/disable of interrupt output of each source with the TUIEN, TOIEN, RUIEN, ROIEN, and IIEN bits in the SSICR register and the TIE and RIE bits in the SSIFCR register.

Table 41.16 SSIE interrupt sources

Channel	Interrupt source	Description	Interrupt flag	DMAC/DTC activation
SSIE0	SSIE0_SSIF	• Transmit underflow interrupt • Transmit overflow interrupt • Receive underflow interrupt • Receive overflow interrupt • Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE0_SSIRXI	Receive data full interrupt	SSIFSR.RDF	Possible
	SSIE0_SSITXI	Transmit data empty interrupt	SSIFSR.TDE	Possible
SSIE1	SSIE1_SSIF	• Transmit underflow interrupt • Transmit overflow interrupt • Receive underflow interrupt • Receive overflow interrupt • Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE1_SSIRT	• Receive data full interrupt • Transmit data empty interrupt	SSIFSR.RDF/ SSIFSR.TDE	Possible

41.9.1 SSIEn_SSIF Interrupt

This interrupt source combines five interrupts. Enable output of necessary interrupts before using SSIE. The five interrupts are operated by using the flags assigned to individual interrupts and interrupt output enable bits. To clear an interrupt, set the interrupt enable to 0 or clear the interrupt flag to 0.

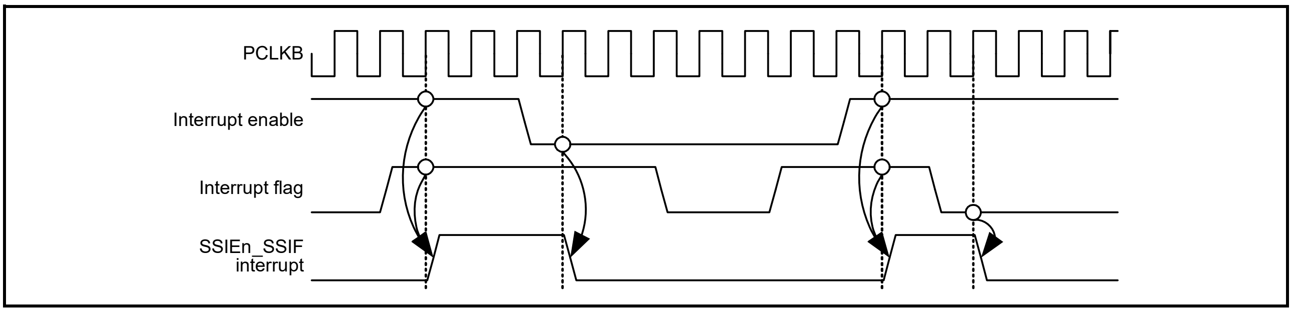


Figure 41.59 Timing Diagram of the common interrupt source, SSIEn_SSIF

- Transmit underflow interrupt

As the transmit underflow interrupt, SSISR.TUIRQ is output while SSICR.TUIEN = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TUIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

- Transmit overflow interrupt

As the transmit overflow interrupt, SSISR.TOIRQ is output while SSICR.TOIRQ = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TOIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

- Receive underflow interrupt

As the receive underflow interrupt, SSISR.RUIRQ is output while SSICR.RUIRQ = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.RUIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

- Receive overflow interrupt

As the receive overflow interrupt, SSISR.ROIRQ is output while SSICR.ROIRQ = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.ROIRQ = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

- Idle mode interrupt

As the idle mode interrupt, SSISR.IIRQ is output while SSICR.IIEN = 1. This interrupt is used to make sure that communication has stopped fully.

41.9.2 SSIE0_SSITXI Interrupt [Full-duplex communication]

The transmit data empty interrupt is a pulse interrupt that is output when the following condition is met:

- SSIFCR.TIE = 1 and SSIFSR.TDE = 1

  SSIE operation: When the value of SSIFSR.TDE changes from 0 to 1 while the value of SSIFCR.TIE is 1

  CPU instruction: When the value of SSIFCR.TIE changes from 0 to 1 while the value of SSIFSR.TDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see Figure 41.60.

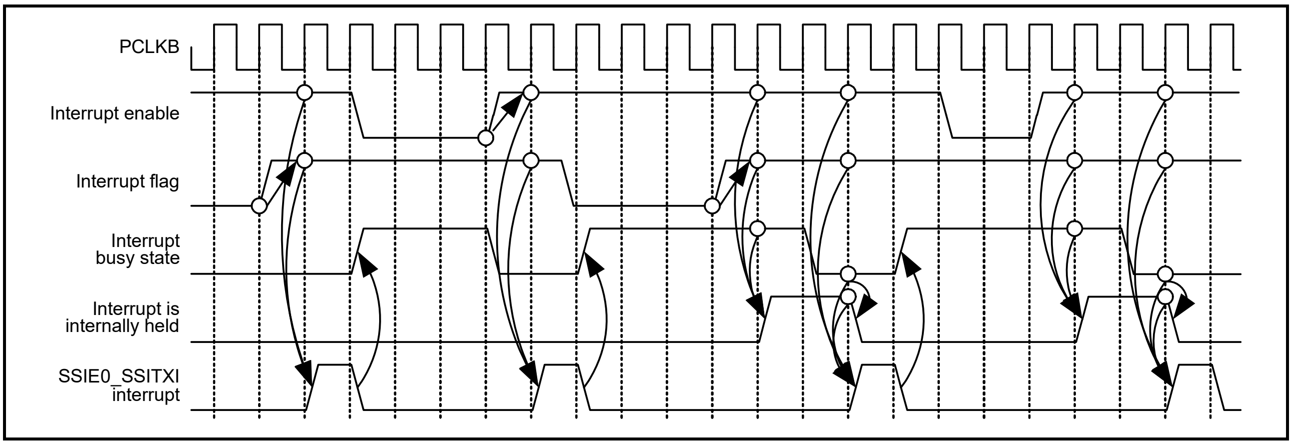


Figure 41.60 SSIE0_SSITXI interrupt timing diagram

41.9.3 SSIE0_SSIRXI Interrupt [Full-duplex communication]

The receive data full interrupt is a pulse interrupt that is output when the following condition is met:

- SSIFCR.RIE = 1 and SSIFSR.RDF = 1.

  SSIE operation: When the value of SSIFSR.RDF changes from 0 to 1 while the value of SSIFCR.RIE is 1

  CPU instruction: When the value of SSIFCR.RIE changes from 0 to 1 while the value of SSIFSR.RDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. The behavior of this interrupt is the same as the behavior shown in Figure 41.60.

41.9.4 SSIE1_SSIRT Interrupt [Half-duplex communication]

This interrupt is output by two sources, transmit data empty interrupt and receive data full interrupt. When this interrupt is generated, read the interrupt flag and specify the interrupt source.

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see Figure 41.61.

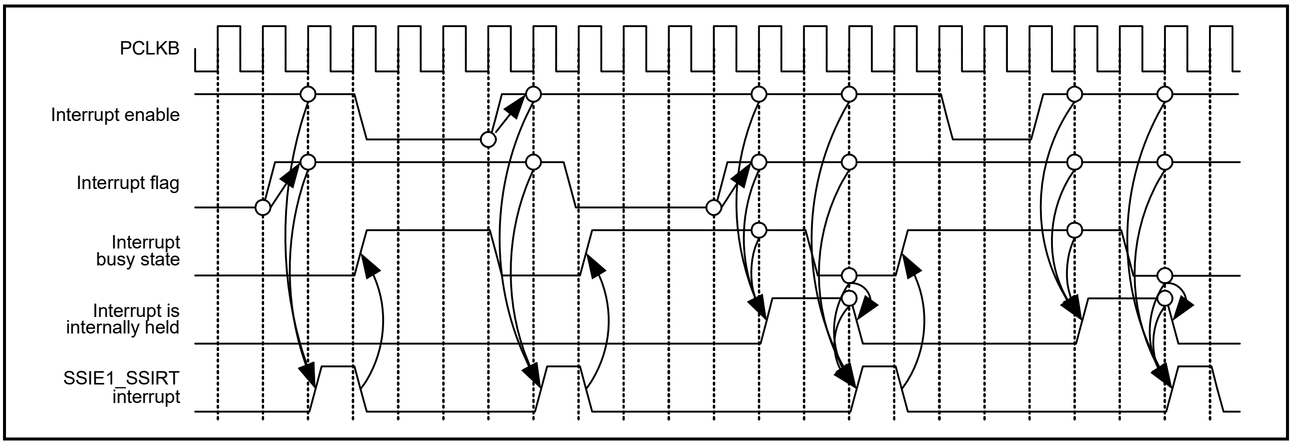


Figure 41.61 SSIE1_SSIRT interrupt timing diagram

41.10 Software Resets

SSIE has three software reset bits to reset its states.

- SSIE software reset (SSIFCR.SSIRST)
- Transmit FIFO data register reset (SSIFCR.TFRST)
- Receive FIFO data register reset (SSIFCR.RFRST).

This section describes the procedures for the three types of software resets.

41.10.1 Software Reset Procedure

(1) SSIE Software Reset

For the SSIE software reset bit (SSIFCR.SSIRST), follow the procedure shown in Figure 41.62. After a reset, the same setting is applied when it is resumed. To change the settings of clocks and slave/master mode, follow the procedure to start communication in Figure 41.53. See reference 41.8.2 and reference 41.8.3 respectively for transmission and reception after communication is resumed.

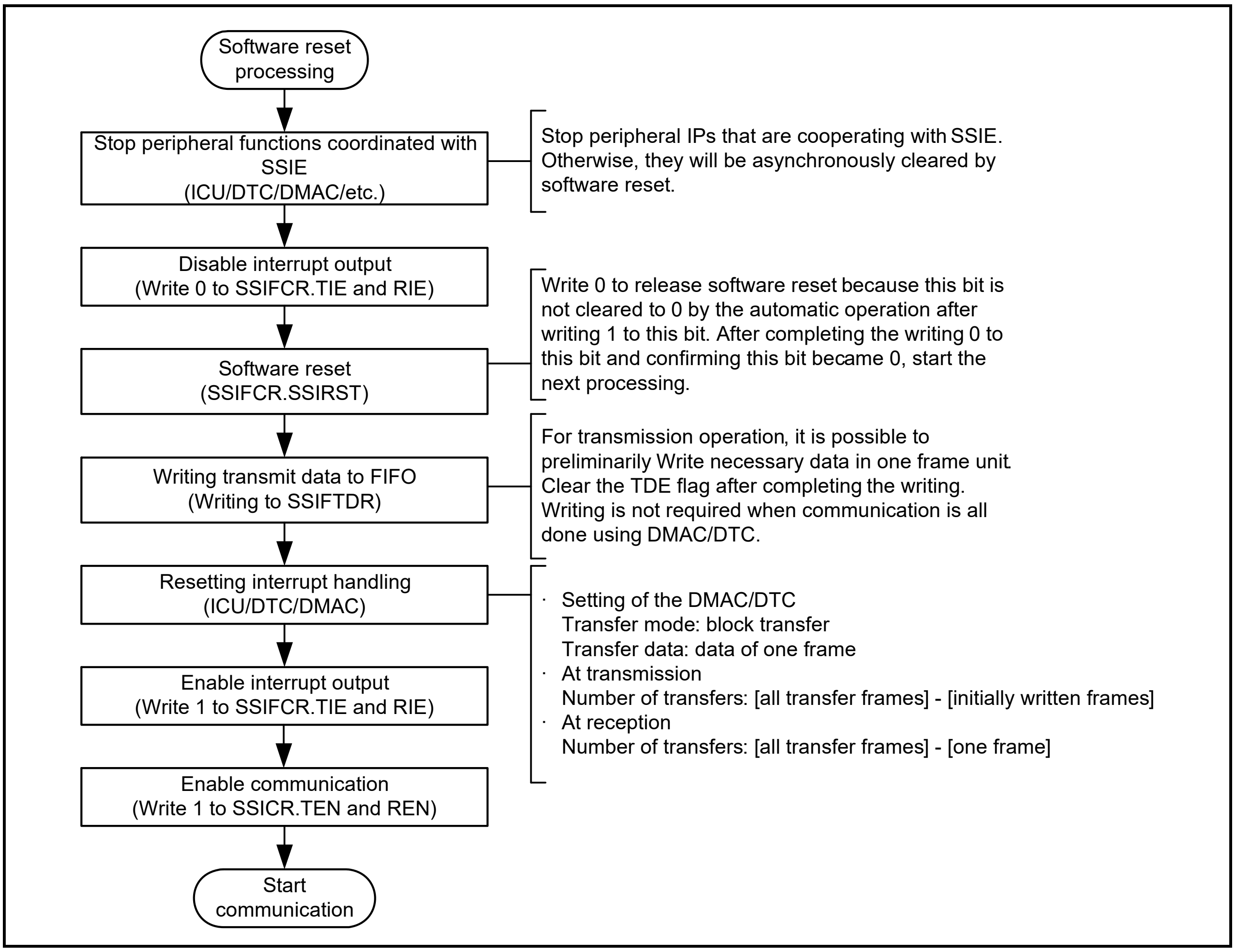


Figure 41.62 Software reset procedure (CPU operation procedure)

(2) Transmit FIFO data register reset

To perform a transmit FIFO data register reset, follow instructions in the procedure to start communication in Figure 41.53 and procedure to resume communication in Figure 41.58.

(3) Receive FIFO data register reset

To perform a receive FIFO data register reset, follow instructions in the procedure to start communication in Figure 41.53 and procedure to resume communication in Figure 41.58.

41.11 Notes

41.11.1 Notes for Slave-mode Communication

41.11.1.1 ADCKE control

In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCK. To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in Figure 41.53 or wait for an idle state by taking the procedure to resume communication in Figure 41.58.

41.11.1.2 SSILRCK/SSIFS pin

SSIE has the SSILRCK/SSIFS pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCK/SSIFS pin only as a trigger to start communication.

41.11.2 Notes for Master-mode Communication

41.11.2.1 ADCKE control

In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.ADCKE.

41.11.2.2 LRCONT control

To stop the output to the SSILRCK/SSIFS pin with SSIOFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected. For details, see Figure 41.44.

41.11.2.3 BCKASTP control

To stop the output to the SSIBCK pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected. For details, see Figure 41.45.

The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCK pin before and during communication.

41.11.3 Notes for Communication Flow

41.11.3.1 When an error interrupt is generated

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error
- Receive overflow error.

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

(1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57. After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

(2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the Transmit FIFO Data Register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the Transmit FIFO Data Register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57. When you resume communication, deal with the invalid serial data appropriately.

(3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

(4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in Figure 41.56 and error-handling procedure in Figure 41.57.

41.11.3.2 Transmit data empty interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

41.11.3.3 Receive data full interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

41.11.3.4 Switching transfer modes

1. For state transition from transmission, reception, and transmission and reception, disable transmission and reception (SSICR.TEN = 0, SSICR.REN = 0).
2. Confirm it is in the idle state (SSISR.IIRQ = 1).
3. In the idle state, set the SSICR.TEN bit and the SSICR.REN bit again and resume transfer.

41.11.3.5 Resume communication after halting SSIE

When communication of SSIE is halted according to the procedure to halt communication in Figure 41.56, resume communication according to the procedure to resume communication in Figure 41.58.

41.11.4 Write Access Restriction

41.11.4.1 SSICR register

If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

(1) TEN Bit and REN Bit

These bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCK/SSIFS signal. For details, see reference 41.8.2, reference 41.8.3, and reference 41.8.4. When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

41.11.4.2 SSISR register

(1) Clearing TUIRQ and TOIRQ

After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

(2) Clearing RUIRQ and ROIRQ

After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

41.11.4.3 Communication state

Writing to the bits with orange-shaded area in Table 41.17 is prohibited. If written, the operation performed immediately after writing is not guaranteed.

Table 41.17 Bits protected from writing during communication

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	00h	+0	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
SSISR	04h	+0	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
SSIFCR	10h	+0	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
		+2	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
SSIFSR	14h	+0	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]						—	—	—	—	—	—	—	RDF
SSIFTDR	18h	+0	FTDR[31:16]															
		+2	FTDR[15:0]															
SSIFRDR	1ch	+0	FRDR[31:16]															
		+2	FRDR[15:0]															
SSIOFR	20h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	24h	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

42. Sampling Rate Converter (SRC)

42.1 Overview

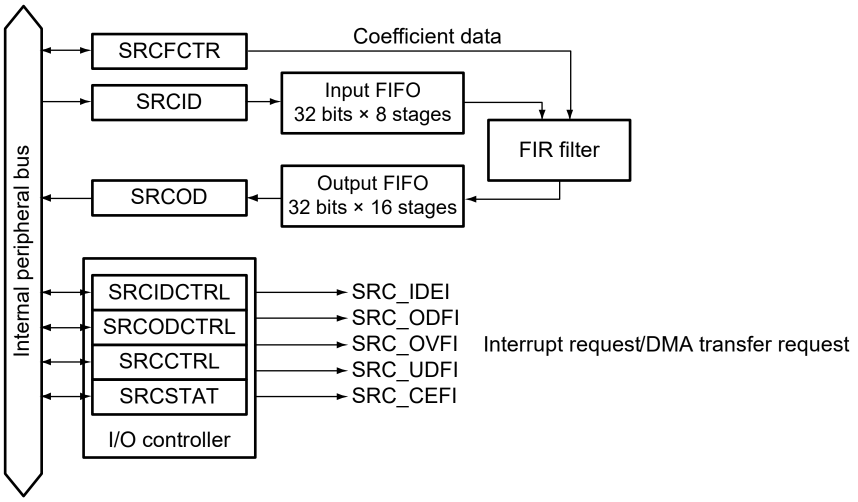
The Sampling Rate Converter (SRC) is used to convert the sampling rate of data produced by various audio decoders, including WMA, MP3, and AAC. Both 16-bit stereo and monaural data are supported. The sampling rate of the input signal can be one of the following (in kHz): 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, or 48 kHz. The sampling rate of the output signal can be one of the following (in kHz): 8, 16, 32, 44.1, or 48 kHz. Independent FIFOs are provided for input and output. In a typical application, a DMA controller can be used to transfer PCM audio data from the SRAM (for example) to the SRC. Sample-converted audio data from the SRC can then be transferred using the DMA Controller to the SSIE interface, from where it can be transmitted to an external audio codec.

Table 42.1 shows the SRC specifications and Figure 42.1 shows a block diagram.

Table 42.1 SRC specifications

Parameter		Specifications
Data size		16 bits (stereo/monaural)
Sampling rates	Input	Selectable to 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, or 48 kHz
	Output	Selectable to 8*1, 16*1, 32, 44.1, or 48 kHz
Processing capacity		Maximum of 7.7 μs for one sample output interval (PCLKB = 60 MHz, 462 clocks)
SNR		80 db or higher
Interrupt sources		Five Input FIFO empty, output FIFO full, output FIFO overflow, output FIFO underflow, and conversion end
DMA transfer sources		Two Input FIFO empty and output FIFO full
Module-stop function		Module-stop state can be set to reduce power consumption

Note 1. Only when input of 44.1 kHz is selected.



SRCFCTR: Filter Coefficient Table
SRCID: Input Data Register
SRCOD: Output Data Register
SRCIDCTRL: Input Data Control Register
SRCODCTRL: Output Data Control Register
SRCCTRL: Control Register
SRCSTAT: Status Register

Figure 42.1 SRC block diagram

42.2 Register Descriptions

42.2.1 Input Data Register (SRCID)

Address(es): SRC.SRCID 4004 DFF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SRCID register is a 32-bit write-only register used to input the data before sampling rate conversion. All the bits are read as 0. The data input to SRCID is stored in the 8-stage input FIFO. When the number of data units in the input FIFO is 8, writing to SRCID has no effect.

For stereo data, bits [31:16] are for Lch data, and bits [15:0] are for Rch data. For monaural data, data in bits [31:16] is valid, and data in bits 15 to 0 is invalid.

The data subject to sampling rate conversion is aligned differently depending on the IED setting in SRCIDCTRL. Table 42.2 shows the correspondence between the IED setting and data alignment.

Table 42.2 Data alignment before sampling rate conversion

IED	Lch[15:8]	Lch[7:0]	Rch[15:8]	Rch[7:0]
0	SRCID[31:24]	SRCID[23:16]	SRCID[15:8]	SRCID[7:0]
1	SRCID[23:16]	SRCID[31:24]	SRCID[7:0]	SRCID[15:8]

42.2.2 Output Data Register (SRCOD)

Address(es): SRC.SRCOD 4004 DFF4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SRCOD register is a 32-bit read-only register used to output the data after sampling rate conversion. The data in the 16-stage output FIFO is read through SRCOD. When the output FIFO is empty after the start of conversion, the value previously read is read again.

The data in SRCOD is aligned differently depending on the OCH and OED settings in SRCODCTRL. Table 42.3 shows the correspondence between the OCH and OED settings and data alignment in SRCOD.

Table 42.3 Data alignment in SRCOD (1 of 2)

OCH	OED	SRCOD[31:24]	SRCOD[23:16]	SRCOD[15:8]	SRCOD[7:0]
0	0	Lch[15:8]	Lch[7:0]	Rch[15:8]*1	Rch[7:0]*1
	1	Lch[7:0]	Lch[15:8]	Rch[7:0]*1	Rch[15:8]*1

Table 42.3 Data alignment in SRCOD (2 of 2)

OCH	OED	SRCOD[31:24]	SRCOD[23:16]	SRCOD[15:8]	SRCOD[7:0]
1*2	0	Rch[15:8]	Rch[7:0]	Lch[15:8]	Lch[7:0]
	1	Rch[7:0]	Rch[15:8]	Lch[7:0]	Lch[15:8]

Note 1. When processing monaural data, the data in these bits is invalid. Discard the invalid data after reading from SRCOD in 32-bit units.

Note 2. When processing monaural data, the data in these bits is invalid.

42.2.3 Input Data Control Register (SRCIDCTRL)

Address(es): SRC.SRCIDCTRL 4004 DFF8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	IED	IEN	—	—	—	—	—	—	IFTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	IFTRG[1:0]	Input FIFO Data Triggering Number	b1 b0 0 0: 0 0 1: 2 1 0: 4 1 1: 6.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	IEN	Input FIFO Empty Interrupt Enable	0: Disable input FIFO empty interrupts 1: Enable input FIFO empty interrupts.	R/W
b9	IED	Input Data Endian*1	0: Little endian 1: Big endian.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only rewrite this bit while the SRCCTRL.SRCEN bit is 0.

The SRCIDCTRL register is a 16-bit read/write register that specifies the endian format of input data, enables or disables the interrupt requests, and specifies the triggering number of data units.

IFTRG[1:0] bits (Input FIFO Data Triggering Number)

The IFTRG[1:0] bits specify the data unit count at which the IINT flag in the Status Register (SRCSTAT) sets to 1. When the number of data units stored in the input FIFO becomes equal to or less than the specified triggering number, the IINT flag sets to 1.

IEN bit (Input FIFO Empty Interrupt Enable)

The IEN bit enables or disables issuing of the input FIFO empty interrupt request when the number of data units in the input FIFO becomes equal to or less than the triggering number specified in the IFTRG[1:0] bits, resulting in the IINT flag in the Status Register (SRCSTAT) setting to 1.

IED bit (Input Data Endian)

The IED bit specifies the endian format of the input data.

42.2.4 Output Data Control Register (SRCODCTRL)

Address(es): SRC.SRCODCTRL 4004 DFFAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OCH	OED	OEN	—	—	—	—	—	—	OFTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	OFTRG[1:0]	Output FIFO Data Trigger Number	b1 b0 0 0: 1 0 1: 4 1 0: 8 1 1: 12.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OEN	Output FIFO Full Interrupt Enable	0: Disable output FIFO full interrupts 1: Enable output FIFO full interrupts.	R/W
b9	OED	Output Data Endian	0: Little endian 1: Big endian.	R/W
b10	OCH	Output Data Channel Exchange *1	0: Do not exchange channels (use same order as data input) 1: Exchange channels (use opposite order from data input).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only rewrite this bit while the SRCCTRL.SRCEN bit is 0.

The SRCODCTRL register is a 16-bit read/write register that specifies whether to exchange the channels for the output data, specifies the endian format of output data, enables or disables the interrupt requests, and specifies the triggering number of data units.

OFTRG[1:0] bits (Output FIFO Data Trigger Number)

The OFTRG[1:0] bits specify the data unit count at which the OINT flag in the Status Register (SRCSTAT) sets to 1. When the number of data units in the output FIFO becomes equal to or greater than the specified triggering number, the OINT flag sets to 1.

OEN bit (Output FIFO Full Interrupt Enable)

The OEN bit enables or disables issuing of the output FIFO full interrupt request when the number of data units in the output FIFO becomes equal to or greater than the number specified in the OFTRG[1:0] bits, resulting in the OINT flag in the Status Register (SRCSTAT) setting to 1.

OED bit (Output Data Endian)

The OED bit specifies the endian format of the output data.

OCH bit (Output Data Channel Exchange)

The OCH bit specifies whether to exchange the channels for the Output Data Register (SRCOD). Do not set this bit to 1 when processing monaural data.

42.2.5 Control Register (SRCCTRL)

Address(es): SRC.SRCCTRL 4004 DFFCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FICRAE	—	CEEN	SRCEN	UDEN	OVEN	FL	CL	IFS[3:0]				—	OFS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	OFS[2:0]	Output Sampling Rate	b2 b0 0 0 0: 44.1 kHz 0 0 1: 48.0 kHz 0 1 0: 32.0 kHz 0 1 1: Setting prohibited 1 0 0: 8.0 kHz*1 1 0 1: 16.0 kHz.*1 Other settings are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b4	IFS[3:0]	Input Sampling Rate	b7 b4 0 0 0 0: 8.0 kHz 0 0 0 1: 11.025 kHz 0 0 1 0: 12.0 kHz 0 0 1 1: Setting prohibited 0 1 0 0: 16.0 kHz 0 1 0 1: 22.05 kHz 0 1 1 0: 24.0 kHz 0 1 1 1: Setting prohibited 1 0 0 0: 32.0 kHz 1 0 0 1: 44.1 kHz 1 0 1 0: 48.0 kHz. Other settings are prohibited.	R/W
b8	CL	Internal Work Memory Clear	Writing 1 to this bit clears the input FIFO, output FIFO, input buffer memory, intermediate memory, and accumulator.	R/W
b9	FL	Internal Work Memory Flush	Writing 1 to this bit starts conversion of the sampling rate for all data in the input FIFO, input buffer memory, and intermediate memory (flush processing).	R/W
b10	OVEN	Output FIFO Overflow Interrupt Enable	0: Disable output FIFO overflow interrupts 1: Enable output FIFO overflow interrupts.	R/W
b11	UDEN	Output FIFO Underflow Interrupt Enable	0: Disable output FIFO underflow interrupts 1: Enable output FIFO underflow interrupts.	R/W
b12	SRCEN	Module Enable	0: Disable SRC module operation 1: Enable SRC module operation.*2	R/W
b13	CEEN	Conversion End Interrupt Enable	0: Disable conversion end interrupts 1: Enable conversion end interrupts.	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	FICRAE	Filter Coefficient Table Access Enable	0: Disable reads to and writes from filter coefficient table RAM 1: Enable reads to and writes from filter coefficient table RAM.	R/W

Note 1. Only valid when the IFS[3:0] bits are 1001b.
Note 2. When SRCEN = 1, do not change the settings of the following bits:
IED bit in SRCIDCTRL, OED and OCH bits in SRCODCTRL, OFS[2:0], IFS[3:0], and FICRAE bits in SRCCTRL.

The SRCCTRL register is a 16-bit read/write register that enables or disables access to the Filter Coefficient Table, module operations, and interrupt requests, and specifies flush processing, clear processing of the internal work memory, and the input and output sampling rates.

OFS[2:0] bits (Output Sampling Rate)

The OFS[2:0] bits specify the output sampling rate.

IFS[3:0] bits (Input Sampling Rate)

The IFS[3:0] bits specify the input sampling rate.

CL bit (Internal Work Memory Clear)

Writing 1 to the CL bit clears the input FIFO, output FIFO, input buffer memory, intermediate buffer memory, and accumulator, and then the CL bit clears to 0. This bit is read as 0. Even if SRCEN = 0, writing 1 to this bit clears the processing.

FL bit (Internal Work Memory Flush)

Writing 1 to the FL bit initiates flush processing by starting conversion of the sampling rate of all the data in the input FIFO, input buffer memory, and intermediate memory. This bit is read as 0. When SRCEN = 0, writing 1 to this bit does not trigger flush processing.

In addition, when 1 is written to the FL bit while the number of data units in the input buffer memory is less than the values shown in Table 42.6, valid output data cannot be received. The internal work memory is cleared without triggering the flush processing.

OVEN bit (Output FIFO Overflow Interrupt Enable)

The OVEN bit enables or disables issuing of the output FIFO overflow interrupt request when the OVF flag in the Status Register (SRCSTAT) is set to 1.

When OVEN = 1: Conversion processing stops until the OVF flag is cleared by the CPU accessing SRCSTAT when the output FIFO overflow interrupt is generated. Writing of conversion results to the output FIFO also stops.

When OVEN = 0: The OVF flag automatically clears when the output FIFO has space, and conversion processing can be continued.

UDEN bit (Output FIFO Underflow Interrupt Enable)

The UDEN bit enables or disables issuing of the output FIFO underflow interrupt request when the output FIFO is read and the UDF flag in the Status Register (SRCSTAT) sets to 1 while the number of data units in the output FIFO is zero.

SRCEN bit (Module Enable)

The SRCEN bit enables or disables SRC operation. Writing 1 to these bits while SRCEN = 0 clears the internal work memory.

CEEN bit (Conversion End Interrupt Enable)

The CEEN bit enables or disables issuing of a conversion end interrupt request when the CEF flag in the Status Register (SRCSTAT) sets to 1 after flush processing is complete and all the output data is read.

FICRAE bit (Filter Coefficient Table Access Enable)

The FICRAE bit enables or disables access to the filter coefficient table RAM. After flush processing is complete, the number of output data units obtained as a result of conversion can be calculated using the following formulas:

$$\frac{\text{Number of output data units} - 1}{\text{Output sampling rate}} = \frac{\text{Number of input data units} \times n - 1}{\text{Input sampling rate} \times n}$$

$$\text{Number of output data units} = \left[(\text{Number of input data units} \times n - 1) \times \frac{\text{Output sampling rate}}{\text{Input sampling rate} \times n} \right] + 1$$

The value of n can be obtained from Table 42.4. The number of input data units must be equal to or greater than the values in Table 42.5.

Table 42.4 Sampling rate settings and value of n

OFS[2:0] setting (output sampling rate [kHz])	IFS[3:0] setting (input sampling rate [kHz])								
	0000b (8.0)	0001b (11.025)	0010b (12.0)	0100b (16.0)	0101b (22.05)	0110b (24.0)	1000b (32.0)	1001b (44.1)	1010b (48.0)
000b (44.1)	6	4	4	3	2	2	3	—	1
001b (48.0)	6	4	4	3	2	2	3	1	—
010b (32.0)	4	8	4	2	4	2	—	2	1
100b (8.0)	—	—	—	—	—	—	—	1	—
101b (16.0)	—	—	—	—	—	—	—	1	—

Conversion processing does not start, and so output data is not obtained, until the specified number of data units are input. The minimum number of input data units necessary for obtaining the first output data depends on the IFS and OFS bit settings. Table 42.5 shows the relation between the settings in the IFS and OFS bits and the number of initial input data required. Table 42.6 shows the relation between the settings in the IFS and OFS bits and the number of initial input data required for processing.

Table 42.5 Relation between sampling rate settings and number of initial input data units required

OFS[2:0] setting (output sampling rate [kHz])	IFS[3:0] setting (input sampling rate [kHz])								
	0000b (8.0)	0001b (11.025)	0010b (12.0)	0100b (16.0)	0101b (22.05)	0110b (24.0)	1000b (32.0)	1001b (44.1)	1010b (48.0)
000b (44.1)	38	40	40	43	48	48	43	—	63
001b (48.0)	38	40	40	43	48	48	43	32	—
010b (32.0)	40	37	40	48	40	48	—	48	63
100b (8.0)	—	—	—	—	—	—	—	63	—
101b (16.0)	—	—	—	—	—	—	—	63	—

Table 42.6 Relation between sampling rate settings and number of input data units required for flush processing

OFS[2:0] setting (output sampling rate [kHz])	IFS[3:0] setting (input sampling rate [kHz])								
	0000b (8.0)	0001b (11.025)	0010b (12.0)	0100b (16.0)	0101b (22.05)	0110b (24.0)	1000b (32.0)	1001b (44.1)	1010b (48.0)
000b (44.1)	27	24	24	22	16	16	22	—	1
001b (48.0)	27	24	24	22	16	16	22	32	—
010b (32.0)	24	29	24	16	24	16	—	16	1
100b (8.0)	—	—	—	—	—	—	—	1	—
101b (16.0)	—	—	—	—	—	—	—	1	—

42.2.6 Status Register (SRCSTAT)

Address(es): SRC.SRCSTAT 4004 DFFEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OFDN[4:0]					IFDN[3:0]				—	CEF	FLF	UDF	OVF	IINT	OINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	OINT	Output FIFO Full Interrupt Request Flag	0: Number of data units in output FIFO has not become equal to or greater than specified triggering number 1: Number of data units in output FIFO has become equal to or greater than specified triggering number.	R/(W) *1
b1	IINT	Input FIFO Empty Interrupt Request Flag	0: Number of data units in input FIFO has not become equal to or smaller than specified triggering number 1: Number of data units in input FIFO has become equal to or smaller than specified triggering number.	R/(W) *1
b2	OVF	Output FIFO Overflow Interrupt Request Flag	0: No output FIFO overflow occurred 1: Output FIFO overflow occurred.	R/(W) *1
b3	UDF	Output FIFO Underflow Interrupt Request Flag	0: No output FIFO underflow occurred 1: Output FIFO underflow occurred.	R/(W) *1
b4	FLF	Flush Processing Status Flag	0: Flush processing complete 1: Flush processing in progress.	R
b5	CEF	Conversion End Flag	0: Not all output data read 1: All output data read.	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b7	IFDN[3:0]	Input FIFO Data Count	Indicates the number of data units in the input FIFO.	R
b15 to b11	OFDN[4:0]	Output FIFO Data Count	Indicates the number of data units in the output FIFO.	R

Note 1. Only 0 can be written after having read as 1.

The SRCSTAT register is a 16-bit read/write register that indicates the number of data units in the input and output FIFOs, whether the various interrupt sources were generated, and the flush processing status.

OINT flag (Output FIFO Full Interrupt Request Flag)

The OINT flag indicates that the number of data units in the output FIFO has become equal to or greater than the triggering number specified in the OFTRG[1:0] bits in the Output Data Control Register (SRCODCTRL).

[Setting condition]

- When the number of data units in the output FIFO becomes equal to or greater than the specified triggering number.

[Clearing conditions]

- Writing 0 to the OINT flag after reading it as 1
- When the last DMA transfer is executed
- Writing 1 to the SRCCTRL.CL bit
- Writing 1 to the SRCCTRL.SRCEN bit while it is 0.

IINT flag (Input FIFO Empty Interrupt Request Flag)

The IINT flag indicates that the number of data units in the input FIFO has become equal to or smaller than the triggering number specified in the IFTRG[1:0] bits in the Input Data Control Register (SRCIDCTRL).

[Setting conditions]

- When the number of data units in the input FIFO becomes equal to or smaller than the specified triggering number

- Writing 1 to the SRCCTRL.CL bit
- Writing 1 to the SRCCTRL.SRCEN bit while it is 0.

[Clearing conditions]

- Writing 0 to the IINT flag after reading is as 1
- When the last DMA transfer is executed.

OVF flag (Output FIFO Overflow Interrupt Request Flag)

The OVF flag indicates that the sampling rate conversion for the next data completes when the output FIFO is full. The conversion stops until the OVF flag is cleared.

[Setting condition]

- When the sampling rate conversion for the next data completes when the output FIFO is full.

[Clearing conditions]

- Writing 0 to the OVF flag after reading it as 1 while the SRCCTRL.OVEN bit is 1
- When the number of data units in the output FIFO decreases after reading SRCOD while the SRCCTRL.OVEN bit is 0
- Writing 1 to the SRCCTRL.CL bit
- Writing 1 to the SRCCTRL.SRCEN bit while it is 0.

UDF flag (Output FIFO Underflow Interrupt Request Flag)

The UDF flag indicates that the output FIFO is read when the number of data units in the output FIFO is zero.

[Setting condition]

- When the output FIFO is read while the number of data units in the output FIFO is zero.

[Clearing conditions]

- Writing 0 to the UDF flag after reading it as 1
- Writing 1 to the SRCCTRL.CL bit
- Writing 1 to the SRCCTRL.SRCEN bit while it is 0.

FLF flag (Flush Processing Status Flag)

The FLF flag indicates whether flush processing is in progress or not.

[Setting condition]

- Writing 1 to the SRCCTRL.FL bit
  (When flush processing is not in progress, however, FLF does not set to 1.)

[Clearing conditions]

- When flush processing completes
- Writing 1 to the SRCCTRL.CL bit
- Writing 1 to the SRCCTRL.SRCEN bit while it is 0.

CEF flag (Conversion End Flag)

The CEF flag indicates that all the output data is read after flush processing completes.

[Setting condition]

- When the number of data units in the output FIFO is zero on completion of flush processing.

[Clearing conditions]

- Writing 0 to the CEF flag after reading it as 1.

- Writing 1 to the SRCCTRL.CL bit
- Writing 1 to the SRCCTRL.SRCEN bit while it is 0.

IFDN[3:0] bits (Input FIFO Data Count)

The IFDN[3:0] bits indicate the number of data units in the input FIFO.

OFDN[4:0] bits (Output FIFO Data Count)

The OFDN[4:0] bits indicate the number of data units in the output FIFO.

42.2.7 Filter Coefficient Table n (SRCFCTRn) (n = 0 to 5551)

Address(es): SRCRAM.SRCFCTR0 to 5551 4004 8000h to 4004 D6BFh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	SRCFCOE[21:16]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SRCFCOE[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

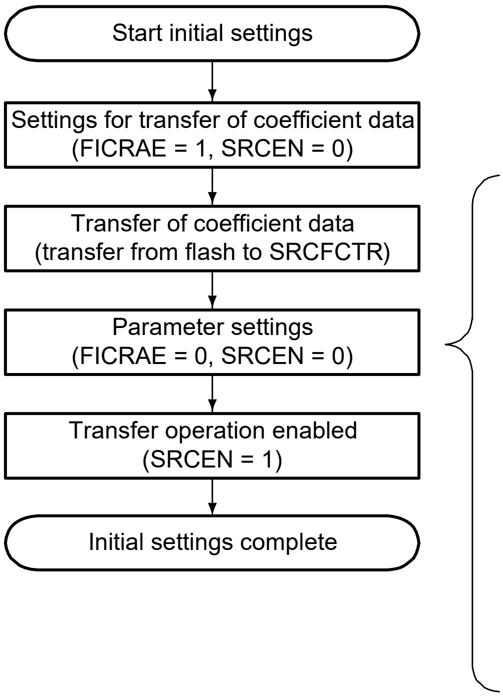
Bit	Symbol	Bit name	Description	R/W
b21 to b0	SRCFCOE[21:0]	Filter Coefficient Table	Stores the filter coefficient value.	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SRCFCTR0 to SRCFCTR5551 are 32-bit read/write SRAM modules that store the filter coefficients to be used for sampling rate conversion. This SRAM can be read from and written to through the peripheral bus only when the FICRAE bit is 1 and the SRCEN bit is 0 in SRCCTRL. Bits 31 to 22 are reserved and are read as 0, and their write values should be 0. Bits 21 to 0 are used for storage of the filter coefficient values, whose initial values are undefined.

42.3 Operation

42.3.1 Initial Settings

Figure 42.2 shows an example flow for the initial settings. After the module-stop state is released, the filter coefficient data stored in the flash and other areas must be transferred to the Filter Coefficient Table (SRCFCTR) before SRC conversion starts. When a filter coefficient value is already stored in the Filter Coefficient Table, skip this transfer and set the required parameters to start the conversion.



Register	Bit	Parameters to be set
SRCCTRL	FICRAE	Filter Coefficient Table access enable
	SRCEN	Module enable

Register	Bit	Parameters to be set
SRCCTRL	CEEN	Enabling/disabling of the SRC_CEFI interrupt
	UDEN	Enabling/disabling of the SRC_UDFI interrupt
	OVEN	Enabling/disabling of the SRC_OVFI interrupt
	IFS[3:0]	Input sampling rate
	OFS[2:0]	Output sampling rate
SRCIDCTRL	IED	Input data endianness
	IEN	Enabling/disabling of the SRC_IDEI interrupt
	IFTRG[1:0]	Input FIFO triggering number
SRCODCTRL	OCH	Exchanging of output data channels
	OED	Output data endianness
	OEN	Enabling/disabling of the SRC_ODFI interrupt
	OFTRG[1:0]	Output FIFO triggering number

Figure 42.2 Example flow for initial settings

42.3.2 Data Input

Figure 42.3 shows an example flow for data input.

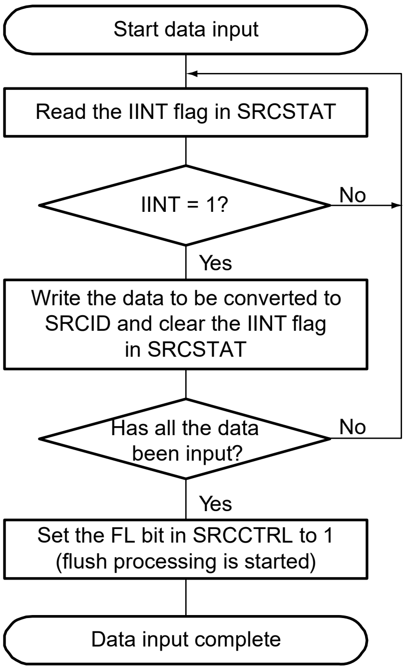


Figure 42.3 Data input flow

(1) When issuing interrupts to the CPU

1. Set the IEN bit in SRCIDCTRL to 1.
2. When the IINT flag in SRCSTAT sets to 1, the IDEI interrupt request is issued. In the interrupt processing routine, read the IINT flag and confirm that it is 1, write data to SRCID, and write 0 to the IINT flag. Then return from the interrupt processing routine.
3. Repeat step 2 until all the data is input, and write 1 to the FL bit in SRCCTRL.

(2) When using interrupts to activate the DMAC

1. Assign the SRC_IDEI interrupt of the SRC to one channel of the DMAC.
2. Set the IEN bit in SRCIDCTRL to 1.
3. When the IINT flag in SRCSTAT sets to 1, the SRC_IDEI interrupt request is issued, activating the DMAC. When data is written to the SRCID register using DMA transfer, and when the number of data units in the input FIFO exceeds the triggering number specified in the IFTRG[1:0] bits in SRCIDCTRL, the IINT flag in SRCSTAT clears to 0.
4. Repeat step 3 until all the data is input, and write 1 to the FL bit in SRCCTRL.

(3) When using SSIE interface interrupts to activate the DMAC to transfer input data from the SSIE interface

1. Assign the SSIE interface to one channel of the DMAC as a DMA transfer request source. Set SSIFRDR of the SSIE interface as a transfer source and SRCID of the SRC as a transfer destination, and set the SSIE interface to enable reception operation.
2. When the RDF bit in SSIFSR sets to 1, the SSIE interface issues an interrupt request, activating the DMAC. The DMAC then reads data from SSIFRDR and writes the data to SRCID.
3. Repeat step 2 until all the data is input, and write 1 to the FL bit in SRCCTRL.

Note: The input FIFO has eight stages. The number of data units that can be transferred (the empty space in the FIFO) when an SRC_IDEI interrupt request is issued depends on the settings in the IFTRG[1:0] bits in SRCIDCTRL. Because the input FIFO is not equipped with a function to prevent or detect overflow, the transferred data is destroyed when overflow occurs. To prevent this, take the settings in the IFTRG[1:0] bits in SRCCTRL into consideration when setting the number of data units to be continuously transferred by the DMA.

42.3.3 Data Output

Figure 42.4 shows an example flow for data output.

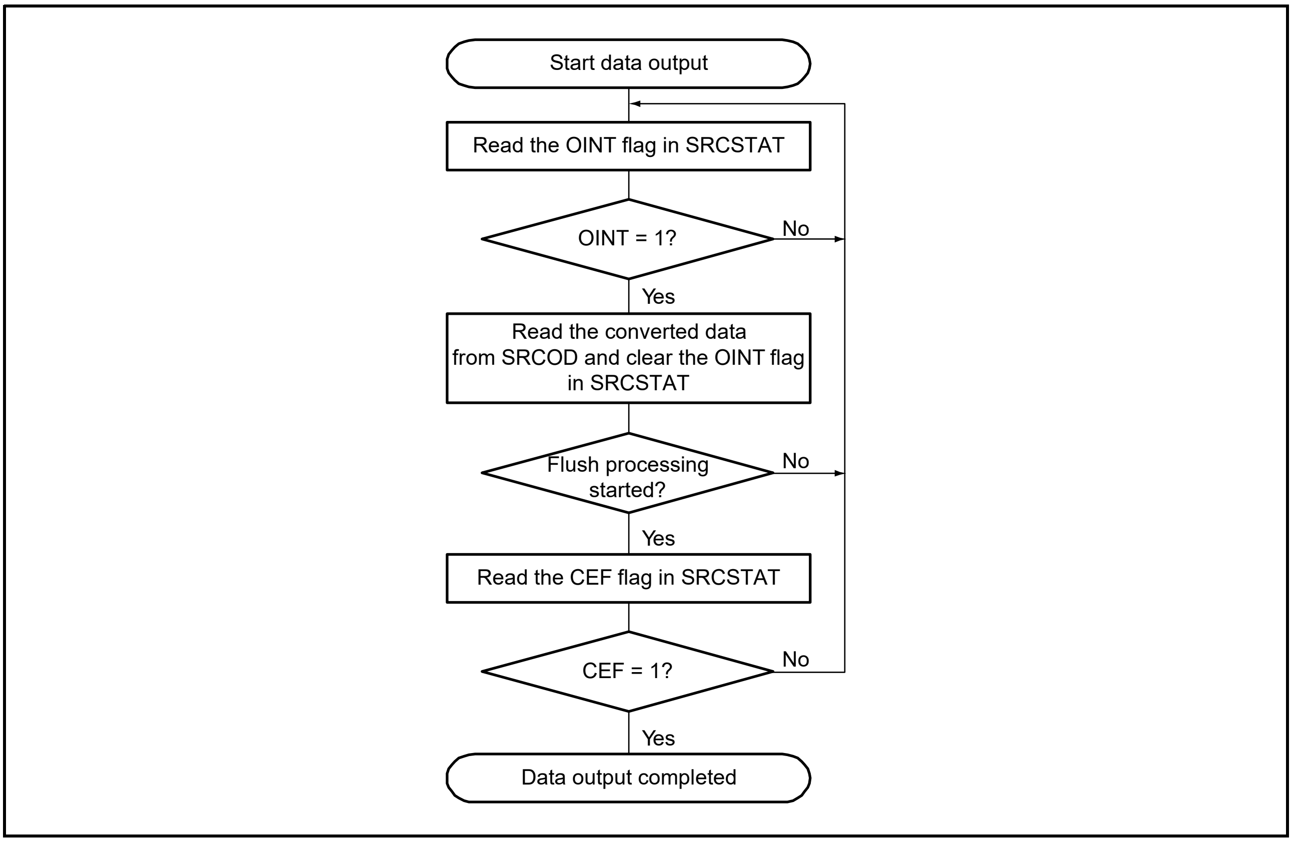


Figure 42.4 Data output flow

(1) When issuing interrupts to the CPU

1. Set the OEN bit in SRCODCTRL to 1.
2. When the OINT flag in SRCSTAT sets to 1, the SRC_ODFI interrupt request is issued. In the interrupt processing routine, read the OINT flag and confirm that it is 1, read data from SRCOD, and write 0 to the OINT flag. Then return from the interrupt processing routine.
3. After flush processing starts, repeat step 2 until the CEF flag in SRCSTAT is read as 1.

(2) When using interrupts to activate the DMAC

1. Assign the SRC_ODFI interrupt of the SRC to one channel of the DMAC.
2. Set the OEN bit in SRCODCTRL to 1.
3. When the OINT flag in SRCSTAT sets to 1, the SRC_ODFI interrupt request is issued, activating the DMAC. When data is read from SRCOD using DMA transfer, and when the number of data units in the output data FIFO becomes equal to or less than the triggering number specified in the OFTRG[1:0] bits, the OINT flag in SRCSTAT clears to 0.
4. After flush processing starts, repeat step 3 until the FLF flag in SRCSTAT is read as 0.

(3) When using SSIE interface interrupts to activate the DMAC to transfer output data to the SSIE interface

1. Set the OVEN bit in SRCCTRL to 0 to disable SRC_OVFI interrupt request generation.
2. Assign the SSIE interface to one channel of the DMAC as a DMA transfer request source. Set SRCID of the SRC as a transfer source and SSIFTDR of the SSIE interface as a transfer destination, and set the SSIE interface to enable transmission operation.
3. When the TDE bit in SSIFSR sets to 1, the SSIE interface issues an interrupt request, activating the DMAC. The DMAC then reads data from SRCOD and writes the data to SSIFTDR.

4. After flush processing starts, repeat step 3 until the CEF flag in SRCSTAT is read as 1.

Note 1. The output FIFO has 16 stages. The conversion stops when no data is read and an overflow occurs in the output FIFO. Even in an overflow state, data can be read from the output FIFO, but the procedure to restart conversion might be required depending on the settings. (For details, see the OVEN bit in SRCCTRL.)

Note 2. When the number of data units in the output FIFO is zero, incorrect data is read. To prevent this, take the settings of the OFTRG[1:0] bits into consideration when setting the number of data units to be continuously transferred by the DMAC.

42.4 Interrupts

The SRC interrupt sources include:

- Input FIFO empty (SRC_IDEI)
- Output FIFO full (SRC_ODFI)
- Output FIFO overflow (SRC_OVFI)
- Output FIFO underflow (SRC_UDFI)
- Conversion end (SRC_CEFI).

Table 42.7 lists the interrupt request types and generation conditions.

Table 42.7 Interrupt requests and generation conditions

Interrupt request	Abbreviation	Interrupt condition	DMAC activation
Input FIFO empty	SRC_IDEI	IINT = 1, IEN = 1, and SRCEN = 1	Possible
Output FIFO full	SRC_ODFI	OINT = 1, OEN = 1, and SRCEN = 1	Possible
Output FIFO overflow	SRC_OVFI	OVF = 1, OVEN = 1, and SRCEN = 1	Not possible
Output FIFO underflow	SRC_UDFI	UDF = 1, UDEN = 1, and SRCEN = 1	Not possible
Conversion end	SRC_CEFI	CEF = 1, CEEN = 1, and SRCEN = 1	Not possible

When an interrupt condition is satisfied, the CPU executes the interrupt exception handling routine. Clear the interrupt source flags during this routine.

The SRC_IDEI and SRC_ODFI interrupts can activate the DMAC. If the DMAC is activated, the interrupts from the SRC are not sent to the CPU.

Do not clear the IINT and OINT flags through a write by the CPU (writing 0 after reading 1) during the DMA transfer.

42.5 Usage Notes

42.5.1 Notes on Accessing Registers

The following writes to SRCCTRL require 3 cycles of the peripheral clock (PCLKB) for the values to be updated in SRCSTAT:

- Writes of 1 to the FL bit in SRCCTRL, for the FLF flag in SRCSTAT to set
- Writes of 1 to the CL bit in SRCCTRL, for each bit in SRCSTAT to initialize
- Writes of 1 to the SRCEN bit in SRCCTRL while the SRCEN bit is 0, for each bit in SRCSTAT to initialized.

However, because the CPU executes any subsequent instruction without waiting for the completion of writes to a register, the updated state of SRCSTAT cannot be correctly read by an instruction immediately after the write instruction to SRCCTRL. To check the updated state of SRCSTAT, perform a dummy read of SRCCTRL or SRCSTAT after the instruction used to write to SRCCTRL.

42.5.2 Notes on Flush Processing

When 1 is written to the FL bit in the SRC Control Register (SRCCTRL), the SRC continues conversion processing by adding 0-data to the input data endpoint. Because of this, only execute flush processing when the audio data endpoint is input and there is no subsequent data.

To perform conversion again after flush processing, clear the internal work memory in either of the following ways.

- Write 1 to the CL bit in SRCCTRL
- Write 0 and then 1 to the SRCEN bit in SRCCTRL.

42.5.3 Notes on DMAC or DTC Transfer

When the DMAC or DTC is used for data transfer to the I/O data registers (SRCID and SRCOD), do not clear the IINT and OINT flags in the Status Register (SRCSTAT) by the CPU (writing 0 after reading 1) during transfer by the DMAC or DTC.

42.5.4 Notes on SRC Operation

Do not access the Filter Coefficient Table while the SRC is operating (SRCCTRL.SRCEN = 1).

42.5.5 Settings for the Module-Stop Function

SRC operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The SRC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

43. SD/MMC Host Interface (SDHI)

43.1 Overview

The Secure Digital Host Interface (SDHI) and MultiMediaCard (MMC) Interface provide the functionality required to connect a variety of external memory cards with the MCU. The SDHI supports both 1-bit and 4-bit buses for connecting different memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA).

The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84-B451) device access. This interface also provides backward compatibility and supports for high-speed SDR transfer modes.

Table 43.1 lists the SD/MMC Host Interface specifications and Figure 43.1 shows a block diagram.

Table 43.1 SD/MMC Host Interface specifications

Interface	Parameter	Specifications
SD	SD bus interface	• Compatible with SD memory card and SDIO card • Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode • Compatible with SD, SDHC, and SDXC formats
SD and MMC shared	SDHI clock frequency	The SDHI clock is generated by dividing PCLKA by 2^n (n = 1 to 9).
	Error check functions	CRC7 (command/response), CRC16 (transfer data)
	Interrupt sources	Card access interrupt (SDHI_MMCn_ACCS), SDIO access interrupt (SDHI_MMCn_SDIO), Card detection interrupt (SDHI_MMCn_CARD) (n = 0 to 1)
	DMA transfer sources	DMAC and DTC triggerable by the SBFAI interrupt SD buffer is read and write accessible using the DMAC
	Other functions	• Card detect function • Write protect support
MMC	MMC bus interface	Transfer bus mode selectable from 1-bit, 4-bit, or 8-bit
	Transfer modes	Backward compatible mode or high-speed SDR mode selectable
	Other functions	e.MMC device access supported

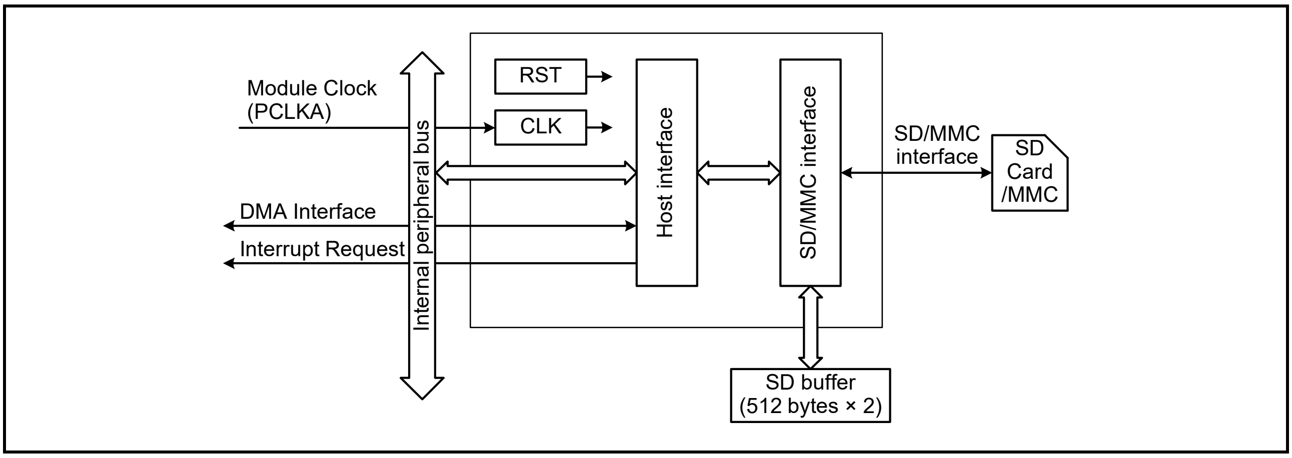


Figure 43.1 SD/MMC Host Interface block diagram

Table 43.2 SDHI I/O pins

Channel	Pin name	I/O	Description
Ch 0	SD0CLK	Output	SDHI clock
	SD0CMD	I/O	Command output, response input
	SD0DAT0	I/O	Data 0 (DAT0)
	SD0DAT1	I/O	Data 1 (DAT1), SDIO interrupt
	SD0DAT2	I/O	Data 2 (DAT2), SDIO Read wait
	SD0DAT3	I/O	Data 3 (DAT3), SD Card detect
	SD0DAT4	I/O	MMC Data 4 (DAT4)
	SD0DAT5	I/O	MMC Data 5 (DAT5)
	SD0DAT6	I/O	MMC Data 6 (DAT6)
	SD0DAT7	I/O	MMC Data 7 (DAT7)
	SD0CD	Input	SD card detection
	SD0WP	Input	SD card write protection
Ch 1	SD1CLK	Output	SDHI clock
	SD1CMD	I/O	Command output, response input
	SD1DAT0	I/O	Data 0 (DAT0)
	SD1DAT1	I/O	Data 1 (DAT1), SDIO interrupt
	SD1DAT2	I/O	Data 2 (DAT2), SDIO Read wait
	SD1DAT3	I/O	Data 3 (DAT3), SD Card detect
	SD1DAT4	I/O	MMC Data 4 (DAT4)
	SD1DAT5	I/O	MMC Data 5 (DAT5)
	SD1DAT6	I/O	MMC Data 6 (DAT6)
	SD1DAT7	I/O	MMC Data 7 (DAT7)
	SD1CD	Input	SD card detection
	SD1WP	Input	SD card write protection

43.2 Register Descriptions

43.2.1 Command Type Register (SD_CMD)

Address(es): SDHI0.SD_CMD 4006 2000h, SDHI1.SD_CMD 4006 2400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CMDIDX[5:0]	Command Index Field Value Select	These bits configure the command index field value. The examples shown include the bit values for the ACMD[1:0] bits. b7 b0 0 0 0 0 0 1 1 0: CMD6 0 0 0 1 0 0 1 0: CMD18 0 1 0 0 1 1 0 1: ACMD13	R/W

Bit	Symbol	Bit name	Description	R/W
b7, b6	ACMD[1:0]	Command Type Select	b7 b6 0 0: CMD 0 1: ACMD. Other settings are prohibited.	R/W
b10 to b8	RSPTP[2:0]	Response Type Select*1	b10 b8 0 0 0: Normal mode Depending on the command, the response type and transfer method are selected in the ACMD[1:0] and CMDIDX[5:0] bits. At this time, the values for b15 to b11 in this register are invalid. 0 1 1: Extended mode and no response 1 0 0: Extended mode and R1, R5, R6, or R7 response 1 0 1: Extended mode and R1b response 1 1 0: Extended mode and R2 response 1 1 1: Extended mode and R3 or R4 response. Other settings are prohibited.	R/W
b11	CMDTP	Data Transfer Select*2	0: Do not include data transfer (bc, bcr, or ac) in command 1: Include data transfer (adtc) in command.	R/W
b12	CMDRW	Data Transfer Direction Select*3	0: Write (SD/MMC Host Interface → SD card/MMC) 1: Read (SD/MMC Host Interface ← SD card/MMC).	R/W
b13	TRSTP	Block Transfer Select*3	0: Single block transfer 1: Multiple blocks transfer.	R/W
b15, b14	CMD12AT[1:0]	CMD12 Automatic Issue Select*4	b15 b14 0 0: Automatically issue CMD12 during multiblock transfer 0 1: Do not automatically issue CMD12 during multiblock transfer. Other settings are prohibited.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Some commands cannot be used in normal mode.
Note 2. The CMDTP bit is only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are only valid when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SD_CMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. See Table 43.8 and Table 43.9 for setting examples. Do not write to the SD_CMD register when the SD_INFO2.CBSY flag is 1.

43.2.2 SD Command Argument Register (SD_ARG)

Address(es): SDHI0.SD_ARG 4006 2008h, SDHI1.SD_ARG 4006 2408h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	These bits specify command format[39:8] (argument).	R/W

The SD_ARG register is used for setting the argument field value. Set the SD_ARG register before setting the SD_CMD register. The argument field value of the automatically issued CMD12 is 0000_0000h regardless of the SD_ARG register value.

43.2.3 SD Command Argument Register 1 (SD_ARG1)

Address(es): SDHI0.SD_ARG1 4006 200Ch, SDHI1.SD_ARG1 4006 240Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	These bits specify command format[39:24] (argument).	R/W
b31 to b16	—	Reserved	These bits are read as 0.	R

The SD_ARG1 register is used for setting the argument field value. Set the SD_ARG1 register before setting the SD_CMD register. The argument field value of the automatically issued CMD12 is 0000_0000h regardless of the SD_ARG1 register value.

43.2.4 Data Stop Register (SD_STOP)

Address(es): SDHI0.SD_STOP 4006 2010h, SDHI1.SD_STOP 4006 2410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SEC	—	—	—	—	—	—	—	STP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	STP	Transfer Stop	Data transfer stops when this bit is set to 1.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SEC	Block Count Register Value Select *1	0: Disable SD_SECCNT register value 1: Enable SD_SECCNT register value.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The SD_STOP register stops data transfer. During a multiblock transfer sequence, the SD_SECCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SD_STOP register.

STP bit (Transfer Stop)

When the STP bit is set to 1 during multiple block transfer, CMD12 is issued to halt the transfer through the SDHI. However, if a command sequence is halted because of a communications error or timeout, CMD12 is not issued. Although continued buffer access is possible even after STP is set to 1, the buffer access error bit (ILR or ILW) in SD_INFO2 is set accordingly.

When STP is set to 1 during transfer for single block write, the access end flag sets when SD_BUF becomes empty, and CMD12 is not issued. If SD_BUF does contain data, the access end flag sets on completion of reception of the busy state without CMD12 being issued.

When STP is set to 1 during transfer for single block read, the access end flag sets immediately after the STP bit is set, and CMD12 is not issued.

When STP is set to 1 during reception of the busy state after an R1b response, the access end flag sets on completion of reception of the busy state without CMD12 being issued.

When STP is set to 1 after a command sequence is completed, CMD12 is not issued and the access end flag does not set.

Set STP to 1 after the response end flag sets.

Set STP to 0 after the access end flag sets.

SEC bit (Block Count Register Value Select)

When SD_CMD is set in the following section to start the command sequence while the SEC bit is set to 1, CMD12 is automatically issued to stop multiblock transfer with the number of blocks set in SD_SECCNT.

CMD18 or CMD25 in normal mode (SD_CMD[10:8] = 000)

SD_CMD[15:13] = 001 in extended mode (CMD12 is automatically issued, multiple block transfer)

When the command sequence is halted because of a communications error or timeout, CMD12 is not automatically issued.

43.2.5 Block Count Register (SD_SECCNT)

Address(es): SDHI0.SD_SECCNT 4006 2014h, SDHI1.SD_SECCNT 4006 2414h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When performing a multiblock transfer, SD_SECCNT is a read/write register used to set the number of blocks to be transferred. For example, when the register value is 0000_0001h, 1 block is transferred. When the register value is 0000_FFFFh, 65,535 blocks are transferred and when the register value is FFFF_FFFFh, 4,294,967,295 blocks are transferred. Do not set this register to 0000_0000h. Do not rewrite the SD_SECCNT register when the SD_INFO2.CBSY flag is 1.

43.2.6 SD Card Response Register 10 (SD_RSP10), SD Card Response Register 32 (SD_RSP32), SD Card Response Register 54 (SD_RSP54)

Address(es): SDHI0.SD_RSP10 4006 2018h, SDHI1.SD_RSP10 4006 2418h, SDHI0.SD_RSP32 4006 2020h, SDHI1.SD_RSP32 4006 2420h SDHI0.SD_RSP54 4006 2028h, SDHI1.SD_RSP54 4006 2428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

43.2.7 SD Card Response Register 1 (SD_RSP1), SD Card Response Register 3 (SD_RSP3), SD Card Response Register 5 (SD_RSP5)

Address(es): SDHI0.SD_RSP1 4006 201Ch, SDHI1.SD_RSP1 4006 241Ch, SDHI0.SD_RSP3 4006 2024h, SDHI1.SD_RSP3 4006 2424h, SDHI0.SD_RSP5 4006 202Ch, SDHI1.SD_RSP5 4006 242Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	These bits store the response from the SD card/MMC.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

43.2.8 SD Card Response Register 76 (SD_RSP76)

Address(es): SDHI0.SD_RSP76 4006 2030h, SDHI1.SD_RSP76 4006 2430h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	—	These bits store the response from the SD card/MMC.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

43.2.9 SD Card Response Register 7 (SD_RSP7)

Address(es): SDHI0.SD_RSP7 4006 2034h, SDHI1.SD_RSP7 4006 2434h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	—	These bits store the response from the SD card/MMC.	R

Bit	Symbol	Bit name	Description	R/W
b31 to b8	—	Reserved	These bits are read as 0.	R

SD_RSP10, SD_RSP32, SD_RSP54, SD_RSP1, SD_RSP3, SD_RSP5, SD_RSP76, and SD_RSP7 are read-only registers that store the response from the SD card/MMC. Depending on the type of response from the SD card/MMC, the SD/MMC Host Interface divides and stores the response among the four registers.

Table 43.3 lists the correspondence between the response type and its storage destination.

Table 43.3 Correspondence between response type and storage destination

Response type	SD_RSP10 register	SD_RSP32 register	SD_RSP54 register	SD_RSP1 register	SD_RSP3 register	SD_RSP5 register	SD_RSP76 register	SD_RSP7 register
R1	[39:8]	—	[39:8]*1	—	—	—	—	—
R1b	[39:8]	—	[39:8]*1	—	—	—	—	—
R2	[39:8]	[71:40]	[103:72]	—	—	—	[127:104]	—
R3	[39:8]	—	—	—	—	—	—	—
R4	[39:8]	—	—	—	—	—	—	—
R5	[39:8]	—	—	—	—	—	—	—
R6	[39:8]	—	—	—	—	—	—	—
R7	[39:8]	—	—	—	—	—	—	—

Note 1. The responses for CMD18 and CMD25 are stored in registers SD_RSP10 and SD_RSP54. Therefore, even if the SD_RSP10 register is overwritten with the response for the automatically issued CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SD_RSP54 register.

43.2.10 SD Card Interrupt Flag Register 1 (SD_INFO1)

Address(es): SDHI0.SD_INFO1 4006 2038h, SDHI1.SD_INFO1 4006 2438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM	SDWPMON	—	SDCDMON	SDCDIN	SDCDRM	ACEND	—	RSPEND
Value after reset:	0	0	0	0	0	x	0	0	x	0	x	0	0	0*1	0	0*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RSPEND	Response End Detection Flag	0: Response end not detected 1: Response end detected.	R(/W)*2
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	ACEND	Access End Detection Flag	0: Access end not detected 1: Access end detected.	R(/W)*2
b3	SDCDRM	SDnCD Removal Flag	0: SD card/MMC removal not detected by the SDnCD pin 1: SD card/MMC removal detected by the SDnCD pin.	R(/W)*2
b4	SDCDIN	SDnCD Insertion Flag	0: SD card/MMC insertion not detected by the SDnCD pin 1: SD card/MMC insertion detected by the SDnCD pin.	R(/W)*2
b5	SDCDMON	SDnCD Pin Monitor Flag	0: SDnCD pin level is high*3 1: SDnCD pin level is low.*3	R
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	SDWPMON	SDnWP Pin Monitor Flag	0: SDnWP pin level is high 1: SDnWP pin level is low.	R

Bit	Symbol	Bit name	Description	R/W
b8	SDD3RM	SDnDAT3 Removal Flag	0: SD card/MMC removal not detected by the SDnDAT3 pin 1: SD card/MMC removal detected by the SDnDAT3 pin.	R(/W)*2
b9	SDD3IN	SDnDAT3 Insertion Flag	0: SD card/MMC insertion not detected by the SDnDAT3 pin 1: SD card/MMC insertion detected by the SDnDAT3 pin.	R(/W)*2
b10	SDD3MON	SDnDAT3 Pin Monitor Flag	0: SDnDAT3 pin level is low 1: SDnDAT3 pin level is high.	R
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value is initialized by a reset and also on reset triggered by the SOFT_RST.SDRST flag.
Note 2. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 3. The flag changes when the pin level continues for the period set in the SD_OPTION.CTOP[3:0] bits or longer.

The SD_INFO1 register indicates the detection of a response end or access end for a command sequence. The SD_INFO1 register also indicates the detection SD card/MMC insertion/removal and the write protection status.

During a multiblock transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag sets to 1, but the RSPEND flag remains set to 0.

If the command sequence is stopped because of a communication error or timeout, the ACEND flag or RSPEND flag sets to 1.

After a reset is canceled, the SDD3MON bit, SDD3IN flag, and SDD3RM flag values are changed in accordance with the status of the SDnDAT3 (n = 0, 1) pin, and their values are changed when data is being transferred in wide bus mode. These 3 bits are used only for SD card. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

RSPEND flag (Response End Detection Flag)

The RSPEND flag indicates that a response end was detected.

[Setting conditions]

- When reception of the response is completed
- When transmission of a command without response is completed
- When reception of the busy state after R1b response is completed
- When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed for transfer of multiple block read
- When reception of the response to CMD52 that was issued by setting the C52PUB bit to 1 is completed for transfer of multiple block write
- This bit is set when a command sequence is halted because of a communications error or timeout.

[Clearing conditions]

- When 0 is written to RSPEND
- When a command without data is issued.

Note: When a command is issued in absence of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

ACEND flag (Access End Detection Flag)

The ACEND flag indicates that an access end was detected.

[Setting conditions]

- When read access to the buffer is completed for transfer of single block read
- When read access to the buffer for the last block of data is completed for transfer of multiple block read
- When read access to the buffer and reception of the response to CMD12 are completed for transfer of multiple block read with automatic issuing of CMD12
- When reception of the busy state after reception of the CRC status is completed for transfer of single block write

- When reception of the busy state after reception of the CRC status of the last block of data is completed for transfer of multiple block write
- When reception of the response busy state for CMD12 is completed for transfer of multiple block write with automatic issuing of CMD12
- When reception of the response to CMD12 that was issued by setting the STP bit to1 is completed for transfer of multiple block read
- When reception of the response busy state for CMD12 that was issued by setting the STP bit to1 is completed for transfer of multiple block write
- When reception of the response to CMD52 that was issued by setting the IOABT bit to 1 is completed for transfer of multiple block read
- When reception of the response to CMD52 that was issued by setting the IOABT bit to1 is completed for transfer of multiple block write
- This bit is set when a command sequence is halted because of a communications error or timeout.

[Clearing conditions]

- When 0 is written to ACEND
- When the access end bit is set to 1.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDCDRM flag (SDnCD Removal Flag)

The SDCDRM flag indicates that SDnCD was removed.

[Setting condition]

- After a change in SDnCD from 0 to 1, Mcycle elapsed with SDnCD held at 1.

[Clearing conditions]

- When 0 is written to SDCDRM.

Note: Mcycle is set in bits [3:0] in SD_OPTION.

SDCDIN flag (SDnCD Insertion Flag)

The SDCDIN flag indicates that SDnCD was inserted.

[Setting condition]

- After a change in SDnCD from 1 to 0, Mcycle elapsed with SDnCD held at 0.

[Clearing conditions]

- When 0 is written to SDCDIN.

Note: Mcycle is set in bits [3:0] in SD_OPTION.

SDD3RM flag (SDnDAT3 Removal Flag)

The SDD3RM flag indicates that SDnDAT3 was removed.

[Setting condition]

- After a change in SDnDAT3 from 1 to 0, two cycles of PCLKA elapsed with SDnDAT3 held at 0.

[Clearing condition]

- When 0 is written to SDD3RM.

SDD3IN flag (SDnDAT3 Insertion Flag)

The SDD3IN flag indicates that SDnDAT3 was inserted.

[Setting condition]

- After a change in SDnDAT3 from 0 to 1, two cycles of PCLKA elapsed with SDnDAT3 held at 1.

[Clearing condition]

- When 0 is written to SDD3IN.

43.2.11 SD Card Interrupt Flag Register 2 (SD_INFO2)

Address(es): SDHI0.SD_INFO2 4006 203Ch, SDHI1.SD_INFO2 4006 243Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ILA	CBSY	SD_CLK_CTRLEN	—	—	—	BWE	BRE	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	0*1	0*1	1*1	0	0	0	0*1	0*1	x	0*1	0*1	0*1	0*1	0*1	0*1	0*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	CMDE	Command Error Detection Flag	0: Command error not detected 1: Command error detected.	R/W*1
b1	CRCE	CRC Error Detection Flag	0: CRC error not detected 1: CRC error detected.	R/W*1
b2	ENDE	End Bit Error Detection Flag	0: End bit error not detected 1: End bit error detected.	R/W*1
b3	DTO	Data Timeout Detection Flag	0: Data timeout not detected 1: Data timeout detected.	R/W*1
b4	ILW	SD_BUF0 Illegal Write Access Detection Flag	0: Illegal write access to the SD_BUF0 register not detected 1: Illegal write access to the SD_BUF0 register detected.	R/W*1
b5	ILR	SD_BUF0 Illegal Read Access Detection Flag	0: Illegal read access to the SD_BUF0 register not detected 1: Illegal read access to the SD_BUF0 register detected.	R/W*1
b6	RSPTO	Response Timeout Detection Flag	0: Response timeout not detected 1: Response timeout detected.	R/W*1
b7	SDD0MON	SDHI_D0 Pin Status Flag	0: SDnDAT0 pin is low 1: SDnDAT0 pin is high.	R
b8	BRE	SD_BUF0 Read Enable Flag	0: Disable read access to the SD_BUF0 register 1: Enable read access to the SD_BUF0 register.	R/W*1
b9	BWE	SD_BUF0 Write Enable Flag	0: Disable write access to the SD_BUF0 register 1: Enable write access to the SD_BUF0 register.	R/W*1
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	SD_CLK_CTRLEN	SD_CLK_CTRL Write Enable Flag	0: SD/MMC bus (CMD and DAT lines) is busy, so write access to the SD_CLK_CTRL.CLKEN and CLKSEL[7:0] bits is disabled 1: SD/MMC bus (CMD and DAT lines) is not busy, so write access to the SD_CLK_CTRL.CLKEN and CLKSEL[7:0] bits is enabled.	R
b14	CBSY	Command Sequence Status Flag	0: Command sequence complete 1: Command sequence in progress (busy).	R
b15	ILA	Illegal Access Error Detection Flag	0: Illegal access error not detected 1: Illegal access error detected.	R/W*1
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SD_INFO2 register indicates the status of the SD buffer and the status of the SD card/MMC. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

CMDE flag (Command Error Detection Flag)

The CMDE flag indicates that a command error was detected. The command sequence stops when a command error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section section 43.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 43.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- The command index of the transmitted command differs from the command index of the received response.
- The command index of a command issued within a command sequence differs from the command index of the received response.

[Clearing condition]

- When 0 is written to CMDE.

CRCE flag (CRC Error Detection Flag)

The CRCE flag indicates that a CRC error was detected. The command sequence stops when a CRC error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 43.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 43.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the CRC status.
- When a CRC error occurs in the read data.
- When a CRC error occurs in the response.
- A CRC error in the response to a command issued within a command sequence.

[Clearing condition]

- When 0 is written to CRCE.

ENDE flag (End Bit Error Detection Flag)

The ENDE flag indicates that an end bit error was detected. The command sequence is stopped when an end bit error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 43.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 43.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the response length (and the end bit is not detected).
- When an error occurs in the read data length (and the end bit is not detected among the valid bits).
- When an error occurs in the CRC status length (and the end bit is not detected).
- An error in the length of a response to a command issued within a command sequence, for example when the end bit is not detected.

[Clearing condition]

- When 0 is written to ENDE.

DTO flag (Data Timeout Detection Flag)

The DTO flag indicates that a data timeout was detected. The command sequence stops when a data timeout occurs.

[Setting conditions]

- After R1b response, the busy state (SDnDAT0 = 0) continues for longer than Ncycle.
- After CRC status, the busy state (SDnDAT0 = 0) continues for longer than Ncycle.
- After write data, the CRC status is not received though Ncycle has elapsed.
- After read command, read data is not received though a time longer than Ncycle has elapsed.
- After CMD12 is issued within a command sequence, the busy state (SDnDAT0 = 0) for longer than Ncycle continues.
- After the reception of read data, read data for the next block are not received though a time longer than Ncycle has elapsed.
- After release of the read wait state, read data for the next block are not received though a time longer than Ncycle has elapsed.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to DTO.

ILW flag (SD_BUF0 Illegal Write Access Detection Flag)

The ILW flag indicates that an SD_BUF0 illegal write access was detected.

[Setting conditions]

- When data is written to SD_BUF0 while it is not in the data read/write command state.
- When data is written to SD_BUF0 while SD_BUF is full.
- When data is written to SD_BUF0 while an error occurs in the CRC status or CRC status length.
- When data is written to SD_BUF0 while a busy state after the CRC status continues for longer than Ncycle.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to ILW.

ILR flag (SD_BUF0 Illegal Read Access Detection Flag)

The ILR flag indicates that an SD_BUF0 illegal read access was detected.

[Setting conditions]

- When SD_BUF is empty while SD_BUF0 is read.
- When data with a CRC error or END error is read from SD_BUF0.

[Clearing condition]

- When 0 is written to ILR.

RSPTO flag (Response Timeout Detection Flag)

The RSPTO flag indicates that a response timeout was detected. The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 43.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 43.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting condition]

- When a response is not received though a time longer than 640 cycles of SD/MMC clock has elapsed (including a response to a command issued within a command sequence).

[Clearing condition]

- When 0 is written to RSPTO.

SDD0MON flag (SDHI_D0 Pin Status Flag)

The SDD0MON flag indicates the status of the SDHI_D0 pin. If the data timeout (DTO) is set but the response timeout (RSPTO) is not set after the Erase command is issued, the end of the Erase sequence (SDD0MON = 1) is confirmed by polling DAT0.

If a communication error or timeout occurs during a write sequence, the DAT0 bit might retain the value 0.

While the SD/MMC clock is stopped, the DAT0 bit retains the value before the clock is stopped.

BRE flag (SD_BUF0 Read Enable Flag)

The BRE flag indicates that SD_BUF0 is enabled for reading.

[Setting conditions]

- When data set in SD_SIZE is stored in SD_BUF0 at single block transfer.
- When data set in SD_SIZE is stored in either bank 1 or bank 2 of SD_BUF0 at multiple block transfer.

[Clearing conditions]

- When 0 is written to BRE
- Reading of a block of data from SD_BUF0 by DMA transfer

When data is read from SD_BUF0 by the CPU, clear BRE then read the amount of data specified in SD_SIZE.

Even if a CRC error or an END error occurs while block data is read, data is stored in SD_BUF0 and BRE is set.

BWE flag (SD_BUF0 Write Enable Flag)

The BWE flag indicates that SD_BUF0 is enabled for writing.

[Setting conditions]

- When SD_BUF0 is empty at single block transfer.
- When either bank 1 or bank 2 of SD_BUF0 is empty at multiple block transfer.

[Clearing conditions]

- When 0 is written to BWE.
- Writing of a block of data to SD_BUF0 by DMA transfer.

When data is written to SD_BUF0 by the CPU, clear BWE and then write the amount of data specified in SD_SIZE.

SD_CLK_CTRLEN flag (SD_CLK_CTRL Write Enable Flag)

When a command sequence is started by writing to SD_CMD, the CBSY bit is set to 1 and, at the same time, the SD_CLK_CTRLEN bit is set to 0. The SD_CLK_CTRLEN bit is set to 1 after 8 cycles of SDCLK have elapsed after the CBSY bit clears to 0 on completion of the command sequence.

ILA flag (Illegal Access Error Detection Flag)

The ILA flag indicates that an illegal access error was detected.

[Setting conditions]

- Writing of data to SD_CMD within a command sequence (CBSY = 1).
- When SD_CMD[11] = 1 (command with data transfer) and SD_CMD[7:0] = 0000 1100b (CMD12) are set in SD_CMD.

[Clearing condition]

- When 0 is written to ILA.

43.2.12 SD INFO1 Interrupt Mask Register (SD_INFO1_MASK)

Address(es): SDHI0.SD_INFO1_MASK 4006 2040h, SDHI1.SD_INFO1_MASK 4006 2440h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SDD3INM	SDD3RMM	—	—	—	SDCDINM	SDCDRMM	ACENDM	—	RSPENDM
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1

Bit	Symbol	Bit name	Description	R/W
b0	RSPENDM	Response End Interrupt Request Mask	0: Do not mask response end interrupt request 1: Mask response end interrupt request.	R/W
b1	—	Reserved	This bit is read as 0 and cannot be modified.	R
b2	ACENDM	Access End Interrupt Request Mask	0: Do not mask access end interrupt request 1: Mask access end interrupt request.	R/W
b3	SDCDRMM	SDnCD Removal Interrupt Request Mask	0: Do not mask SD card/MMC removal interrupt request by the SDnCD pin 1: Mask SD card/MMC removal interrupt request by the SDnCD pin.	R/W
b4	SDCDINM	SDnCD Insertion Interrupt Request Mask	0: Do not mask SD card/MMC insertion interrupt request by the SDnCD pin 1: Mask SD card/MMC insertion interrupt request by the SDnCD pin.	R/W
b7 to b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b8	SDD3RMM	SDnDAT3 Removal Interrupt Request Mask	0: Do not mask SD card/MMC removal interrupt request by the SDnDAT3 pin 1: Mask SD card/MMC removal interrupt request by the SDnDAT3 pin.	R/W
b9	SDD3INM	SDnDAT3 Insertion Interrupt Request Mask	0: Do not mask SD card/MMC insertion interrupt request by the SDnDAT3 pin 1: Mask SD card/MMC insertion interrupt request by the SDnDAT3 pin.	R/W
b31 to b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

The SD_INFO1_MASK register enables or disables interrupt requests from the status flags in the SD_INFO1 register. See Table 43.5, Interrupt sources, for details on the relationship between the status flags and the requested interrupt source.

43.2.13 SD INFO2 Interrupt Mask Register (SD_INFO2_MASK)

Address(es): SDHI0.SD_INFO2_MASK 4006 2044h, SDHI1.SD_INFO2_MASK 4006 2444h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ILAM	—	—	—	—	—	BWEM	BREM	—	RSPTOM	ILRM	ILWM	DTOM	ENDEM	CRCEM	CMDEM
Value after reset:	1	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	CMDEM	Command Error Interrupt Request Mask	0: Do not mask command error interrupt request 1: Mask command error interrupt request.	R/W
b1	CRCEM	CRC Error Interrupt Request Mask	0: Do not mask CRC error interrupt request 1: Mask CRC error interrupt request.	R/W
b2	ENDEM	End Bit Error Interrupt Request Mask	0: Do not mask end bit detection error interrupt request 1: Mask end bit detection error interrupt request.	R/W
b3	DTOM	Data Timeout Interrupt Request Mask	0: Do not mask data timeout interrupt request 1: Mask data timeout interrupt request.	R/W
b4	ILWM	SD_BUF0 Register Illegal Write Interrupt Request Mask	0: Do not mask illegal write detection interrupt request for the SD_BUF0 register 1: Mask illegal write detection interrupt request for the SD_BUF0 register.	R/W
b5	ILRM	SD_BUF0 Register Illegal Read Interrupt Request Mask	0: Do not mask illegal read detection interrupt request for the SD_BUF0 register 1: Mask illegal read detection interrupt request for the SD_BUF0 register.	R/W
b6	RSPTOM	Response Timeout Interrupt Request Mask	0: Do not mask response timeout interrupt request 1: Mask response timeout interrupt request.	R/W
b7	—	Reserved	This bit is 0 when read and cannot be modified.	R
b8	BREM	BRE Interrupt Request Mask	0: Do not mask read enable interrupt request for the SD buffer 1: Mask read enable interrupt request for the SD buffer.	R/W
b9	BWEM	BWE Interrupt Request Mask	0: Do not mask write enable interrupt request for the SD_BUF0 register 1: Mask write enable interrupt request for the SD_BUF0 register.	R/W
b10	—	Reserved	This bit is read as 0.	R
b11	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b14 to b12	—	Reserved	These bits are read as 0.	R
b15	ILAM	Illegal Access Error Interrupt Request Mask	0: Do not mask illegal access error interrupt request 1: Mask illegal access error interrupt request.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0.
When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_INFO2_MASK register enables or disables interrupt requests from the status flags in the SD_INFO2 register. See Table 43.5 for details on the relationship between the status flags and the requested interrupt source.

43.2.14 SD Clock Control Register (SD_CLK_CTRL)

Address(es): SDHI0.SD_CLK_CTRL 4006 2048h, SDHI1.SD_CLK_CTRL 4006 2448h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CLKCTRLEN	CLKEN	CLKSEL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CLKSEL[7:0]	SDHI Clock Frequency Select*1	b7 b0 0 0 0 0 0 0 0 0: PCLKA/2 0 0 0 0 0 0 0 1: PCLKA/4 0 0 0 0 0 0 1 0: PCLKA/8 0 0 0 0 0 1 0 0: PCLKA/16 0 0 0 0 1 0 0 0: PCLKA/32 0 0 0 1 0 0 0 0: PCLKA/64 0 0 1 0 0 0 0 0: PCLKA/128 0 1 0 0 0 0 0 0: PCLKA/256 1 0 0 0 0 0 0 0: PCLKA/512. Other settings are prohibited.	R/W
b8	CLKEN	SD/MMC Clock Output Control*1	0: Disable SD/MMC clock output (fix SDnCLK signal low) 1: Enable SD/MMC clock output.	R/W
b9	CLKCTRLEN	SD/MMC Clock Output Automatic Control Select	0: Disable automatic control of SD/MMC clock output 1: Enable automatic control of SD/MMC clock output.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

The SDCLKCTRL register controls the SD/MMC clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SD_CMD register to start a command sequence. Do not write to the SDCLKCTRL register when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

CLKCTRLEN bit (SD/MMC Clock Output Automatic Control Select)

The CLKCTRLEN bit enables or disables the automatic control function for SD/MMC clock output, which causes the SD/MMC clock to output only within a command sequence.

The timing with which SD/MMC clock output starts and stops is as follows:

- SD/MMC clock output starts after writing to SD_CMD
- SD/MMC clock output stops when 8 cycles of SD/MMC clock have elapsed after the end of the command sequence.

In addition, SD/MMC clock is fixed to 0 while CLKEN of SD_CLK_CTRL is 0, regardless of the value of this bit.

43.2.15 Transfer Data Length Register (SD_SIZE)

Address(es): SDHI0.SD_SIZE 4006 204Ch, SDHI1.SD_SIZE 4006 244Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LEN[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	LEN[9:0]	Transfer Data Size Setting	These bits specify the transfer data size.*1	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.

The SD_SIZE register sets the transfer data size.

LEN[9:0] bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set in the LEN[9:0] bits from 1 byte to 512 bytes. When CMD12 is automatically issued during a multiblock transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multiblock transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32-, 64-, 128-, or 256-byte multiblock read transfer can only be performed during an SDIO multiblock transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

43.2.16 SD Card Access Control Option Register (SD_OPTION)

Address(es): SDHI0.SD_OPTION 4006 2050h, SDHI1.SD_OPTION 4006 2450h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WIDTH	—	WIDTH8	—	—	—	—	TOUTMASK	TOP[3:0]				CTOP[3:0]			
Value after reset:	0*2	1	0*2	0	0	0	0	0*2	1*2	1*2	1*2	0*2	1*2	1*2	1*2	0*2

Bit	Symbol	Bit name	Description		R/W
b3 to b0	CTOP[3:0]	Card Detection Time Counter *1	b3 b0 0 0 0 0: PCLKA × 2^{10} 0 0 0 1: PCLKA × 2^{11} 0 0 1 0: PCLKA × 2^{12} 0 0 1 1: PCLKA × 2^{13} 0 1 0 0: PCLKA × 2^{14} 0 1 0 1: PCLKA × 2^{15} 0 1 1 0: PCLKA × 2^{16} 0 1 1 1: PCLKA × 2^{17}.	b3 b0 1 0 0 0: PCLKA × 2^{18} 1 0 0 1: PCLKA × 2^{19} 1 0 1 0: PCLKA × 2^{20} 1 0 1 1: PCLKA × 2^{21} 1 1 0 0: PCLKA × 2^{22} 1 1 0 1: PCLKA × 2^{23} 1 1 1 0: PCLKA × 2^{24} 1 1 1 1: Setting prohibited.	R/W

Bit	Symbol	Bit name	Description	R/W
b7 to b4	TOP[3:0]	Timeout Counter*1	b7 b4 0 0 0 0: SDHI clock × 2^{13} 0 0 0 1: SDHI clock × 2^{14} 0 0 1 0: SDHI clock × 2^{15} 0 0 1 1: SDHI clock × 2^{16} 0 1 0 0: SDHI clock × 2^{17} 0 1 0 1: SDHI clock × 2^{18} 0 1 1 0: SDHI clock × 2^{19} 0 1 1 1: SDHI clock × 2^{20}. b7 b4 1 0 0 0: SDHI clock × 2^{21} 1 0 0 1: SDHI clock × 2^{22} 1 0 1 0: SDHI clock × 2^{23} 1 0 1 1: SDHI clock × 2^{24} 1 1 0 0: SDHI clock × 2^{25} 1 1 0 1: SDHI clock × 2^{26} 1 1 1 0: SDHI clock × 2^{27} 1 1 1 1: Setting prohibited.	R/W
b8	TOUTMASK	Timeout Mask	0: Activate timeout 1: Inactivate timeout (do not set RSPTO and DTO bits of SD_INFO2 or E6 to E0 bits of SDERRSTS2). When timeout occurs because of an inactivated timeout, execute a software reset to terminate the command sequence.	R/W
b12 to b9	—	Reserved	These bits are read as 0.	R
b13	WIDTH8*2	Bus Width	See bit 15 WIDTH bit.	R/W
b14	—	Reserved	This bit is read as 1.	R
b15	WIDTH	Bus Width*2	b15 b13 0 1: 8-bit width 0 0: 4-bit width 1 0: 1-bit width 1 1: 1-bit width. For 1-byte write transfers, set 4-bit or 1-bit width. Do not set 8-bit width.	R/W
b31 to b16	—	Reserved	These bits are read as 0.	R

Note 1. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.
Note 2. The initial value is applied at a reset and when the SOFT_RST.SDRST flag is 0.

The SD bus width and timeout counter are set in the SD_OPTION register.

43.2.17 SD Error Status Register 1 (SD_ERR_STS1)

Address(es): SDHI0.SD_ERR_STS1 4006 2058h, SDHI1.SD_ERR_STS1 4006 2458h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0*3	1*3	0*3	0*3	0*3	0*3	0*3	0	0	0*3	0*3	0*3	0*3	0*3	0*3

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	CMDE0	Command Error Flag 0	0: No error exists in command index field value of a command*1 response 1: Error exists in command index field value of a command*1 response.	R
b1	CMDE1	Command Error Flag 1	0: No error exists in command index field value of a command*2 response 1: Error exists in command index field value of a command*2 response (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the CMDE0 flag).	R
b2	RSPLENE0	Response Length Error Flag 0	0: No error exists in command*1 response length 1: Error exists in command*1 response length.	R

Bit	Symbol	Bit name	Description	R/W
b3	RSPLENE1	Response Length Error Flag 1	0: No error exists in command*2 response length 1: Error exists in command*2 response length (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPLENE0 flag).	R
b4	RDLENE	Read Data Length Error Flag	0: No read data length error occurred 1: Read data length error occurred.	R
b5	CRCLENE	CRC Status Token Length Error Flag	0: No CRC status token length error occurred 1: CRC status token length error occurred.	R
b7, b6	—	Reserved	These bits are read as 0.	R
b8	RSPCRCE0	Response CRC Error Flag 0	0: No CRC error detected in command*1 response 1: CRC error detected in command*1 response.	R
b9	RSPCRCE1	Response CRC Error Flag 1	0: No CRC error detected in command*2 response (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPCRCE0 flag) 1: CRC error detected in command*2 response.	R
b10	RDCRCE	Read Data CRC Error Flag	0: No CRC error detected in read data 1: CRC error detected in read data.	R
b11	CRCTKE	CRC Status Token Error Flag	0: No error detected in CRC status token 1: Error detected in CRC status token.	R
b14 to b12	CRCTK[2:0]	CRC Status Token	These bits store the CRC status token value (normal value is 010b).	R
b15	—	Reserved	This bit is read as 0	R
b31 to b16	—	Reserved	These bits are read as undefined	R

Note 1. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 2. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 3. The initial value is applied at a reset and when the SOFT_RST.SDRST flag is 0.

The SD_ERR_STS1 register indicates the CRC status token, CRC error, end bit error, and command error.

43.2.18 SD Error Status Register 2 (SD_ERR_STS2)

Address(es): SDHI0.SD_ERR_STS2 4006 205Ch, SDHI1.SD_ERR_STS2 4006 245Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0	0	0*4	0*4	0*4	0*4	0*4	0*4	0*4

Bit	Symbol	Bit name	Description	R/W
b0	RSPTO0	Response Timeout Flag 0	0: After command*1 was issued, response was received in less than 640 cycles of the SD/MMC clock 1: After command*1 was issued, response was not received in 640 or more cycles of the SD/MMC clock.	R
b1	RSPTO1	Response Timeout Flag 1	0: After command*2 was issued, response was received in less than 640 cycles of the SD/MMC clock 1: After command*2 was issued, response was not received after 640 or more cycles of the SD/MMC clock (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the RSPTO0 flag).	R

Bit	Symbol	Bit name	Description	R/W
b2	BSYTO0	Busy Timeout Flag 0	0: After R1b response was received, SD/MMC was released from the busy state during the specified period*3 1: After R1b response was received, SD/MMC was in the busy state after the specified period*3 elapsed.	R
b3	BSYTO1	Busy Timeout Flag 1	0: After CMD12 was automatically issued, SD/MMC was released from the busy state during the specified period*3 1: After CMD12 was automatically issued, SD/MMC was in the busy state after the specified period*3 elapsed (with SD_CMD.CMDIDX[5:0] setting, an error that occurs with CMD12 issue is indicated in the BSYTO0 flag).	R
b4	RDTO	Read Data Timeout Flag	When a read command is issued, this flag sets to 1 when read data is not received after the specified period*3 elapses. When read data is received, this flag sets to 1 when the next block of read data is not received after the specified period*3 elapses. When the SD/MMC exits the read wait state, this flag sets to 1 when the next block of read data is not received after the specified period*3 elapses.	R
b5	CRCTO	CRC Status Token Timeout Flag	0: After CRC data was written to the SD card/MMC, a CRC status token was received during the specified period*3 1: After CRC data was written to the SD card/MMC, a CRC status token was not received after the specified period*3 elapsed.	R
b6	CRCBSYTO	CRC Status Token Busy Timeout Flag	0: After a CRC status token was received, the SD/MMC was released from the busy state during the specified period*3 1: After a CRC status token was received, the SD/MMC was in the busy state after the specified period*3 elapsed.	R
b31 to b7	—	Reserved	These bits are read as 0.	R

Note 1. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 2. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.
Note 3. Set the SD_OPTION.TOP[3:0] bits to select the number of *n* cycles.
Note 4. The initial value is applied at a reset and when the SOFT_RST.SDRST flag is 0.

The SD_ERR_STS2 register indicates the timeout status.

43.2.19 SD Buffer Register (SD_BUF0)

Address(es): SDHI0.SD_BUF0 4006 2060h, SDHI1.SD_BUF0 4006 2460h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

When writing to the SD card, the write data is written to this register. When reading from the SD card, the read data is read from this register. This register is internally connected to two 512-byte buffers.

If both buffers are not empty when executing multiple block read, the SD card/MMC clock is stopped to suspend receiving data. When one of the buffers is empty, the SD card/MMC clock is supplied to resume receiving data.

43.2.20 SDIO Mode Control Register (SDIO_MODE)

Address(es): SDHI0.SDIO_MODE 4006 2068h, SDHI1.SDIO_MODE 4006 2468h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	INTEN	SDIO Interrupt Acceptance Enable*1	0: Disable SDIO interrupt acceptance 1: Enable SDIO interrupt acceptance.	R/W
b1	—	Reserved	This bit is read as 0.	R
b2	RWREQ	Read Wait Request	0: Allow SD/MMC to exit read wait state 1: Request for SD/MMC to enter read wait state.	R/W
b7 to b3	—	Reserved	These bits are read as 0.	R
b8	IOABT	SDIO Abort	If this bit is set to 1 during multiblock transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted.	R/W
b9	C52PUB	SDIO None Abort	If this bit is set to 1 during multiblock transfer triggered by CMD53, CMD52 is issued after the transfer process is complete, and the command sequence is completed.	R/W
b31 to b10	—	Reserved	These bits are read as 0.	R

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The SDIO_MODE register controls reception of the SDIO interrupt, CMD52 issuance during multiblock transfer, and read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

RWREQ bit (Read Wait Request)

When RWREQ is set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks.

[Read wait state releasing]

- The read wait state is released, when RWREQ is cleared to 0 in the read wait state.
- When IOABT is set to 1 in the read wait state, RWREQ is automatically cleared to 0 after CMD52 is issued, and then the read wait state is released.
- When C52PUB and RWREQ are set to 1 simultaneously in the CMD53 (multiple block) read sequence, the read wait state is not automatically released. Therefore, after the CMD52 response is received, clear RWREQ. You must set RWREQ and C52PUB simultaneously.

When RWREQ is set to 1 while the last block in the CMD53 (multiple block) read sequence is transferred, the read wait state is not entered and RWREQ is automatically cleared to 0 by setting access end. Set RWREQ to 1 after the response end flag sets.

IOABT bit (SDIO Abort)

When the IOABT bit is set to 1 in a CMD53 (multiple block) sequence, the CMD53 sequence is halted and CMD52 is issued. However, if a command sequence is halted because of a communication error or timeout, CMD52 is not issued. Although continued buffer access is possible even after IOABT is set to 1, the buffer access error bit (ILR or ILW) in SD_INFO2 is set accordingly. Set SD_ARG before setting IOABT to 1.

When IOABT is set to 1 during transfer for a single block write, the access end flag sets when SD_BUF0 becomes empty, and CMD52 is not issued. If SD_BUF0 contains data, the access end flag sets on completion of reception of the busy state without CMD52 being issued.

When IOABT is set to 1 during transfer for single block read, the access end flag sets immediately after IOABT is set, and CMD52 is not issued.

When IOABT is set to 1 during reception of the busy state after an R1b response, the access end flag sets on completion of reception of the busy state without CMD52 being issued.

When IOABT is set to 1 after a command sequence is completed, CMD52 is not issued and the access end flag does not set.

Set IOABT to 1 after the response end flag sets.

Set IOABT to 0 after the access end flag sets.

C52PUB bit (SDIO None Abort)

When the C52PUB bit is set to 1 in the CMD53 (multiple block) write sequence, CMD52 is automatically issued between blocks if SD_BUF0 becomes empty. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag sets to 1.

When C52PUB and RWREQ are set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks and CMD52 is automatically issued. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag sets to 1.

If C52PUB is set to 1 in the CMD53 (multiple block) read sequence, you must set RWREQ to 1 in addition to C52PUB.

Set SD_ARG before setting C52PUB to 1.

Set C52PUB to 1 after the response end flag sets.

43.2.21 SDIO Interrupt Flag Register (SDIO_INFO1)

Address(es): SDHI0.SDIO_INFO1 4006 206Ch, SDHI1.SDIO_INFO1 4006 246Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXWT	EXPUB52	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IOIRQ	SDIO Interrupt Status Flag	0: No SDIO interrupt detected 1: SDIO interrupt detected.	R(/W)*1
b2, b1	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b13 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	EXPUB52	EXPUB52 Status Flag	Indicates the status of the EXPUB52.	R(/W)*1
b15	EXWT	EXWT Status Flag	Indicates the status of the EXWT.	R(/W)*1
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the bit.

The SDIO_INFO1 register indicates the status of the SDIO card access. Set flags to be cleared to 0. Set flags that are not being cleared to 1.

IOIRQ flag (SDIO Interrupt Status Flag)

The IOIRQ flag indicates that an SDIO interrupt occurred.

[Setting condition]

- When SDIO interrupt from an SDIO card is received while INTEN in SDIO_MODE is set to 1.

[Clearing condition]

- When 0 is written to IOIRQ.*1

Note 1. Before clearing this bit, access the SDIO card to negate the SDIO interrupt signal from the SDIO card. If the interrupt signal is not negated, this bit can be set again.

EXPUB52 flag (EXPUB52 Status Flag)

The EXPUB52 flag indicates the EXPUB52 status.

[Setting conditions]

- While the last block in the CMD53 (multiple block) sequence is transferred, C52PUB in SDIO_MODE is set to 1.
- While C52PUB is set to 1 in the CMD53 (multiple block) write sequence, the last block is transferred.

[Clearing condition]

- When 0 is written to EXPUB52.

EXWT flag (EXWT Status Flag)

The EXWT flag indicates the EXWT status.

[Setting condition]

- While the last block in the CMD53 (multiple block) read sequence is transferred, RWREQ in SDIO_MODE is set to 1.

[Clearing condition]

- When 0 is written to EXWT.

43.2.22 SDIO INFO1 Interrupt Mask Register (SDIO_INFO1_MASK)

Address(es): SDHI0.SDIO_INFO1_MASK 4006 2070h, SDHI1.SDIO_INFO1_MASK 4006 2470h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXWTM	EXPUB52M	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQM
Value after reset:	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	IOIRQM	IOIRQ Interrupt Mask Control	0: Do not mask IOIRQ interrupts 1: Mask IOIRQ interrupts.	R/W
b2, b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b13 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	EXPUB52M	EXPUB52 Interrupt Request Mask Control	0: Do not mask EXPUB52 interrupt requests 1: Mask EXPUB52 interrupt requests.	R/W

Bit	Symbol	Bit name	Description	R/W
b15	EXWTM	EXWT Interrupt Request Mask Control	0: Do not mask EXWT interrupt requests 1: Mask EXWT interrupt requests.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SDIO_INFO1_MASK register enables or disables interrupt requests from the status flags in the SDIO_INFO1 register. See Table 43.5, Interrupt sources for details on the relationship between the status flags and the requested interrupt source.

43.2.23 DMA Mode Enable Register (SD_DMAEN)

Address(es): SDHI0.SD_DMAEN 4006 21B0h, SDHI1.SD_DMAEN 4006 25B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAEN	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0.	R
b1	DMAEN	DMA Transfer Enable[*1, *2]	0: Disable use of DMA transfer to access SD_BUF0 register 1: Enable use of DMA transfer to access SD_BUF0 register.	R/W
b3, b2	—	Reserved	These bits are read as 0.	R
b4	—	Reserved	This bit is read as 1.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	—	Reserved	These bits are read as 0.	R
b9, b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11, b10	—	Reserved	These bits are read as 0.	R
b12	—	Reserved	This bit is read as 1.	R
b31 to b13	—	Reserved	These bits are read as 0.	R

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY bit is 1.

Note 2. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0. When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_DMAEN register enables or disables DMA transfers.

DMAEN bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SD_CMD register.

43.2.24 Software Reset Register (SOFT_RST)

Address(es): SDHI0.SOFT_RST 4006 21C0h, SDHI1.SOFT_RST 4006 25C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	SDRST	Software Reset Control	0: Reset SD/MMC Host Interface software 1: Cancel reset of SD/MMC Host Interface software.	R/W
b2, b1	—	Reserved	These bits are read as 1.	R
b31 to b3	—	Reserved	These bits are read as 0.	R

Table 43.4 lists the bits and flags initialized by SD/MMC Host Interface software reset.

Table 43.4 Bits and flags initialized by SD/MMC Host Interface software reset

Register	Bit/flag
SD_STOP	SEC
SD_INFO1	RSPEND, ACEND
SD_INFO2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, SDD0MON, BRE, BWE, SD_CLK_CTRLEN, ILA
SD_CLK_CTRL	CLKEN
SD_OPTION	CTOP[3:0], TOP[3:0], WIDTH Bits b8 and b13 in the SD_OPTION register are also initialized by the SDHI software reset.
SD_ERR_STS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SD_ERR_STS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIO_INFO1	IOIRQ, EXPUB52, EXWT

43.2.25 SD Interface Mode Setting Register (SDIF_MODE)

Address(es): SDHI0.SDIF_MODE 4006 21CCh, SDHI1.SDIF_MODE 4006 25CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NOCHKCR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b8	NOCHKCR	CRC Check Mask	CRC check mask bit for MMC test commands. Set when CRC16 or CRC status value check is not executed. 0: Enable CRC check 1: Disable CRC Check (ignore CRC16 valued when reading and ignore CRC status value when writing).	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NOCHKCR bit (CRC Check Mask)

The NOCHKCR bit is used for MMC test commands. This bit is set when CRC16 or CRC status value check is not executed.

43.2.26 Swap Control Register (EXT_SWAP)

Address(es): SDHI0.EXT_SWAP 4006 21E0h, SDHI1.EXT_SWAP 4006 25E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	BRSWP	BWSWP	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0 and cannot be modified.	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	This bit is read as 0 and cannot be modified.	R
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	—	Reserved	This bit is read as 0 and cannot be modified.	R
b6	BWSWP	SD_BUF0 Swap Write*1	0: Normal write operation 1: Swap the byte endian order before writing to SD_BUF0 register.	R/W
b7	BRSWP	SD_BUF0 Swap Read*1	0: Normal read operation 1: Swap the byte endian order before reading SD_BUF0 register.	R/W
b10 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14, b13	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b16	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The EXT_SWAP register selects whether or not the byte endian order is swapped when accessing the SD_BUF0 register. See section 43.3.1 for details on the differences in accessing the SD_BUF0 register based on the EXT_SWAP register value.

43.3 Operation

43.3.1 SD/MMC Interface

When data is read from the SD card/MMC, the process is as follows:

1. The SD/MMC Host Interface receives data from the SD card/MMC through the SDnDAT signal (see Figure 43.2 and Figure 43.3).

2. The received data is stored in SD_BUF of the MMC Host Interface (see Figure 43.4).
3. The data stored in SD_BUF is read from SD_BUF0 (see Figure 43.5).

When data is written to the SD card/MMC, the specified procedure is reversed.

When accessing SD_BUF0, pay attention to the transfer order in SDnDAT and the store order in SD_BUF. If required, you can change the byte endian of the data read from or written to SD_BUF0 using the SDSWAP register. See Figure 43.6.

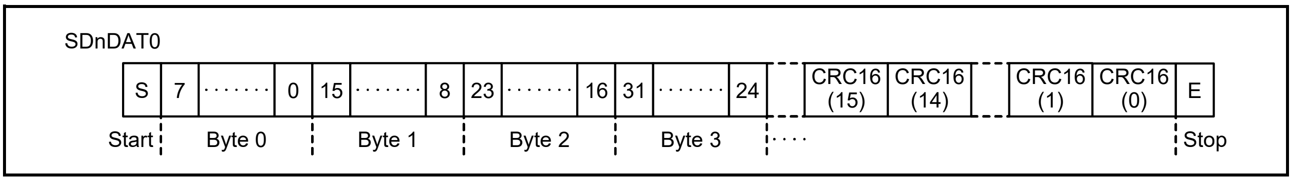


Figure 43.2 SDnDAT in 1-bit width mode

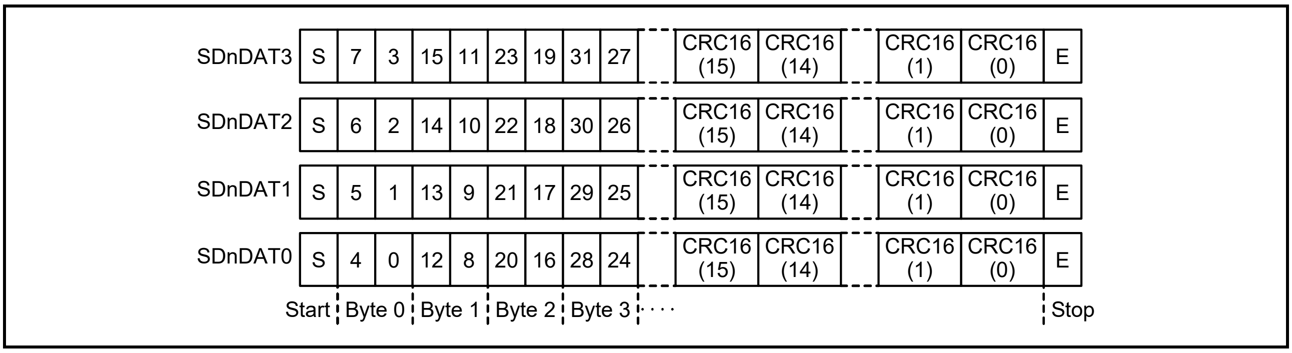


Figure 43.3 SDnDAT in 4-bit width mode

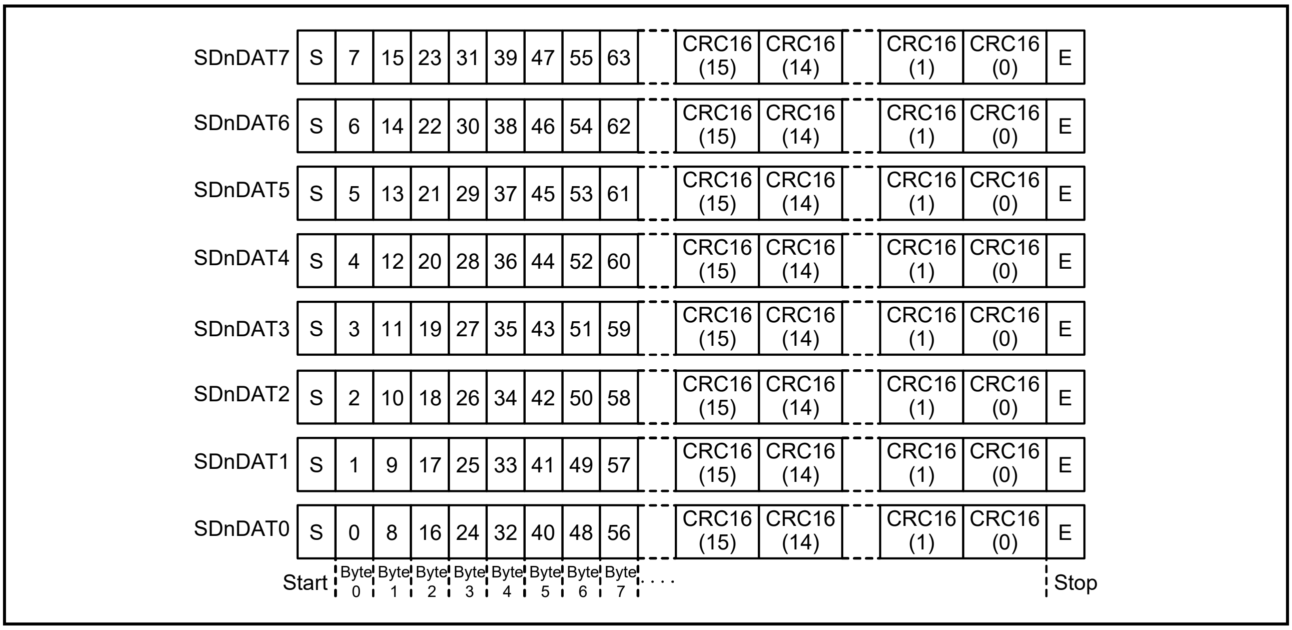


Figure 43.4 SDnDAT in 8-bit width mode

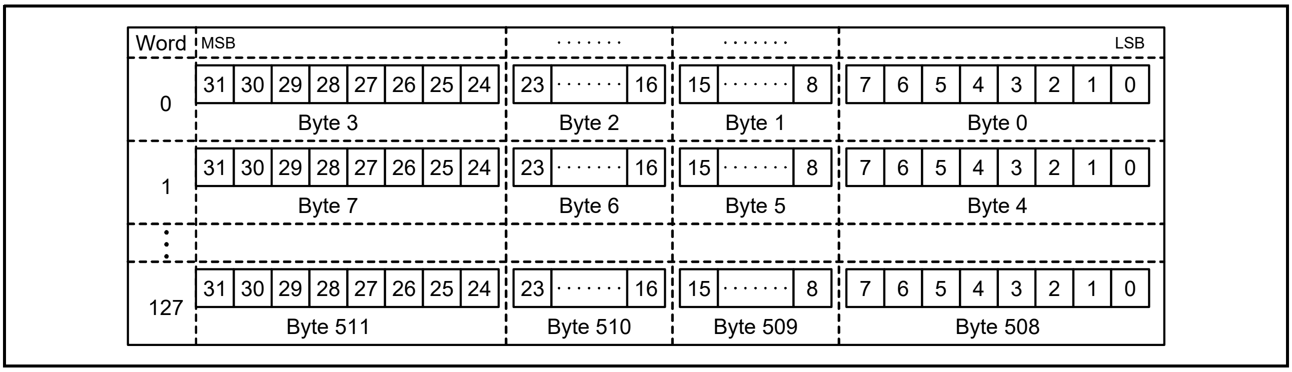


Figure 43.5 SD_BUF store data

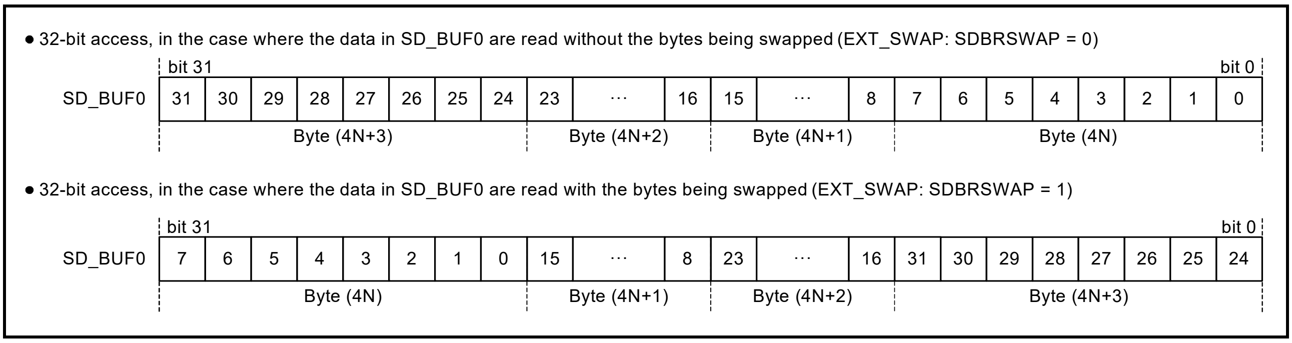


Figure 43.6 Read from SD_BUF0

43.3.2 Card Detect/Write Protect

43.3.2.1 Card detect

The SD/MMC Host Interface has two types of card detect functions.

(1) Card detect with SDnCD (n = 0, 1)

Figure 43.7 shows the timing for card detect using SDnCD. SDnCD is connected to the card socket and pulled up on the host device. The resistance of the pull-up resistor is determined by the specification of the SD/MMC host device.

(2) Card insertion

SDnCD is pulled down when a card is inserted. At this point, if SDnCD is pulled down for the Mcycle period (set in SD_OPTION), SDCDIN in SD_INFO1 is set to 1. It is cleared by writing 0.

(3) Card removal

SDnCD is pulled up when a card is removed. At this point, if SDnCD is pulled up for the Mcycle period (set in SD_OPTION), SDCDRM in SD_INFO1 is set to 1. It is cleared by writing 0.

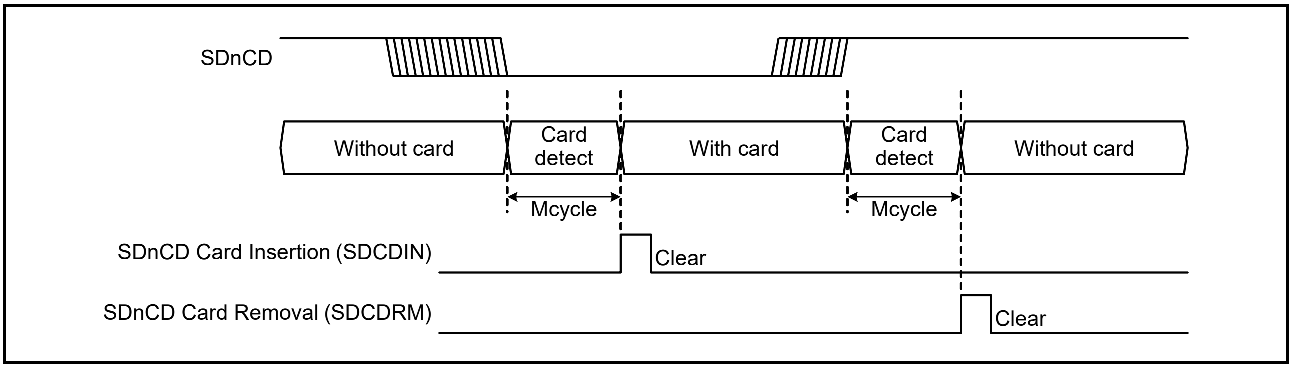


Figure 43.7 Example of card detect with SDnCD

(4) SD card detect with SDnDAT3 (n = 0, 1)

Figure 43.8 shows the timing when the SD card is detected with SDnDAT3. In addition, SDnDAT3 is pulled down by the host device, and the resistance value for pulling down is determined by the specification of the SD host device.

(5) Card insertion

When an SD card is inserted, SDnDAT3 is pulled up and SDD3IN in SD_INFO1 is set to 1. It is cleared by writing 0.

(6) Card removal

When an SD card is removed, SDnDAT3 is pulled down and SDD3RM in SD_INFO1 is set to 1. It is cleared by writing 0.

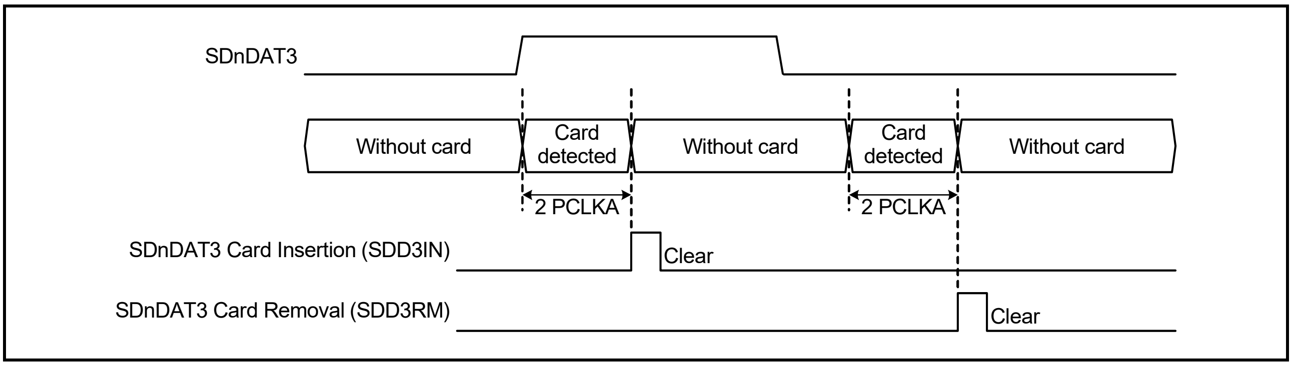


Figure 43.8 SD card detect with SDnDAT3

43.3.2.2 Write protect

The SD/MMC Host Interface has two types of write protect functions.

(1) Write protect with SDnWP (n = 0, 1)

SDnWP is connected to the card socket and pulled up or pulled down by the card insertion. The selection of pulling up or pulling down and the resistance value is determined by the specification of the SD host device. When the SDnWP state is reflected to SDWPMON in SD_INFO1, the write protect state is set after the SD card is inserted.

(2) Write protect with command

The internal write protection of the card and the lock/unlock operation of the card are realized by the command.

43.3.3 Interrupt Request and DMA Transfer Request

43.3.3.1 Interrupts

Table 43.5 lists the SDHI interrupt sources. The SDHI requests an interrupt when:

- The status flags in registers SD_INFO1, SD_INFO2, and SDIO_INFO1 set to 1

- The associated bits in the SD_INFO1_MASK, SD_INFO2_MASK, and SDIO_INFO1_MASK registers are 0.

When clearing the status flags in registers SD_INFO1, SD_INFO2, and SDIO_INFO1, write 0 to the status flags to be cleared and write 1 to the status flags that are not being cleared.

Table 43.5 Interrupt sources

	Status flag register		Interrupt mask register		Interrupt name	
Interrupt sources	Register symbol	Bit symbol	Register symbol	Bit symbol	Ch 0	Ch 1
Card Access Interrupt (CACI)	SD_INFO1	ACEND	SD_INFO1_MASK	ACENDM	SDHI_MMC0_ACCS	SDHI_MMC1_ACCS
		RSPEND		RSPENDM		
	SD_INFO2	ILA	SD_INFO2_MASK	ILAM		
		BWE		BWEM		
		BRE		BREM		
		RSPTO		RSPTOM		
		ILR		ILRM		
		ILW		ILWM		
		DTO		DTOM		
		ENDE		ENDEM		
		CRCE		CRCEM		
		CMDE		CMDEM		
SDIO Access Interrupt (SDACI)	SDIO_INFO1	EXWT	SDIO_INFO1_MASK	EXWTM	SDHI_MMC0_SDIO	SDHI_MMC1_SDIO
		EXPUB52		EXPUB52M		
		IOIRQ		IOIRQM		
Card Detect Interrupt (CDETI)	SD_INFO1	SDD3IN	SD_INFO1_MASK	SDD3INM	SDHI_MMC0_CARD	SDHI_MMC1_CARD
		SDD3RM		SDD3RMM		
		SDCDIN		SDCDINM		
		SDCDRM		SDCDRMM		

43.3.3.2 DMA transfer requests (SDHI_MMCn_ODMSDBREQ, n = 0 to 1)

The SD/MMC Host Interface has two types of DMA transfer requests.

(1) SD_BUF write DMA transfer request

- When the BWE bit in SD_INFO2 is set to 1 while the DMAEN bit in SD_DMAEN is set to 1, the SD_BUF write DMA transfer request is asserted.
- The SD_BUF write DMA transfer request is negated when the last data in one block (based on the transfer data size set in SD_SIZE) is transferred. The SD_BUF write DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. However, if a communications error or timeout occurs at the DMA transfer, the SD_BUF write DMA transfer request is not negated.
- The BWE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request for writing to SD_BUF by DMA transfer.
- The number of DMA transfers must be n x one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF write DMA transfer request is negated.
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Because the BWE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit, or to a communications error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer is not issued while the BWE bit is set.

(2) SD_BUF read DMA transfer request

- When the BRE bit in SD_INFO2 is set to 1 while the DMAEN bit in the SD_DMAEN register is set to 1, the SD_BUF read DMA transfer request is asserted.
- The SD_BUF read DMA transfer request is negated when the last data in one block (based on the transfer data size set in SD_SIZE) is transferred. The SD_BUF read DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. However, if a communications error or timeout occurs at the DMA transfer, the SD_BUF read DMA transfer request is not negated.
- The BRE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request to write to SD_BUF by DMA transfer.
- The number of DMA transfers must be n x one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF read DMA transfer request is negated.
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Because the BRE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit or in response to a communications error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer is not issued while the BRE bit is set.

43.3.4 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the associated status flag in the SD_INFO2 register sets to 1. Also, depending on the source of the error, the associated flag in the SD_ERR_STS1 or SD_ERR_STS2 register sets to 1.

The status flags in registers SD_ERR_STS1 and SD_ERR_STS2 clear to 0 by writing to the SD_CMD register, or by setting the SOFT_RST.SDRST bit to 0.

Table 43.6 Communication errors

Communication error	Interrupt flag register		Error status register		This occurs when...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
End bit error	SD_INFO2	ENDE	SD_ERR_STS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error*1
				RSPLENE0	The response length is in error*2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response*1
				RSPCRCE0	There is a CRC error in the response*2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match*1
				CMDE0	The command index field value for the transmitted command and received response do not match*2

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Table 43.7 Timeouts

Timeout	Interrupt flag register		Error status register		This occurs when...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Response timeout	SD_INFO2	RSPTO	SD_ERR_STS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse*1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse*2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set*3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set*3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set*3 elapses
					After the read data is received, the next block read data is not received even after at least the period set*3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set*3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set*3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set*3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 3. The period is set in the SD_OPTION.TOP[3:0] bits.

43.3.5 Command without Data Transfer (SD/MMC)

Figure 43.9 and Figure 43.10 show example flows.

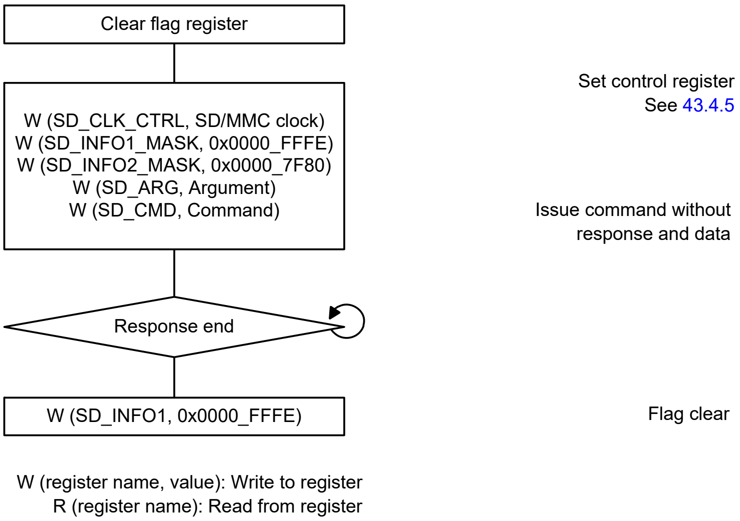


Figure 43.9 Example flow of command without response and data

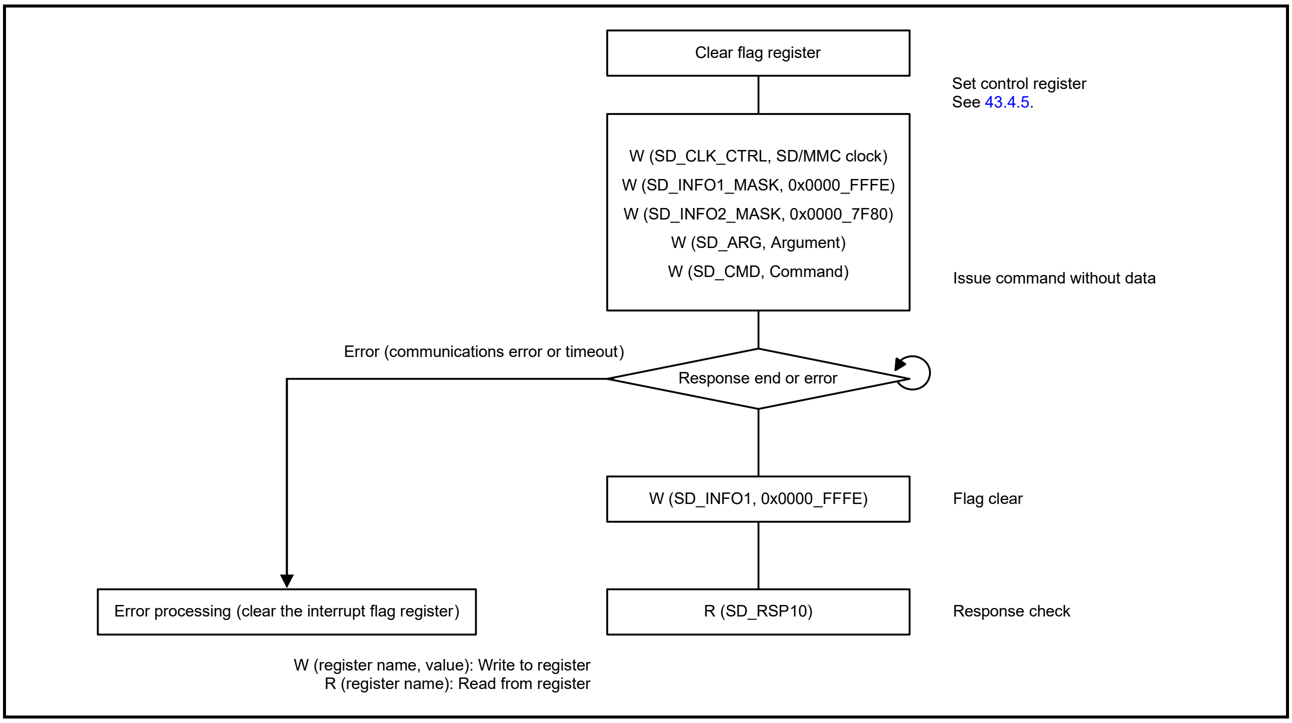


Figure 43.10 Example flow of command without data

43.3.5.1 Operation for command without data transfer

The following legend is used for description of register read/write.

W (register name, value): Write to register

R (register name): Read from register

The operation is described in the following section.

(1) Command without response and data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock and interrupt masking. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to SD_CMD.
Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When transmission of a command is completed, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

(2) Command without data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock and interrupt masking. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to the SD_CMD.
Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When a response is received, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

e. Read a response from SD_RSP10. Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

43.3.6 Single Block Read (SD/MMC)

Figure 43.11 shows an example flow of a single block read operation.

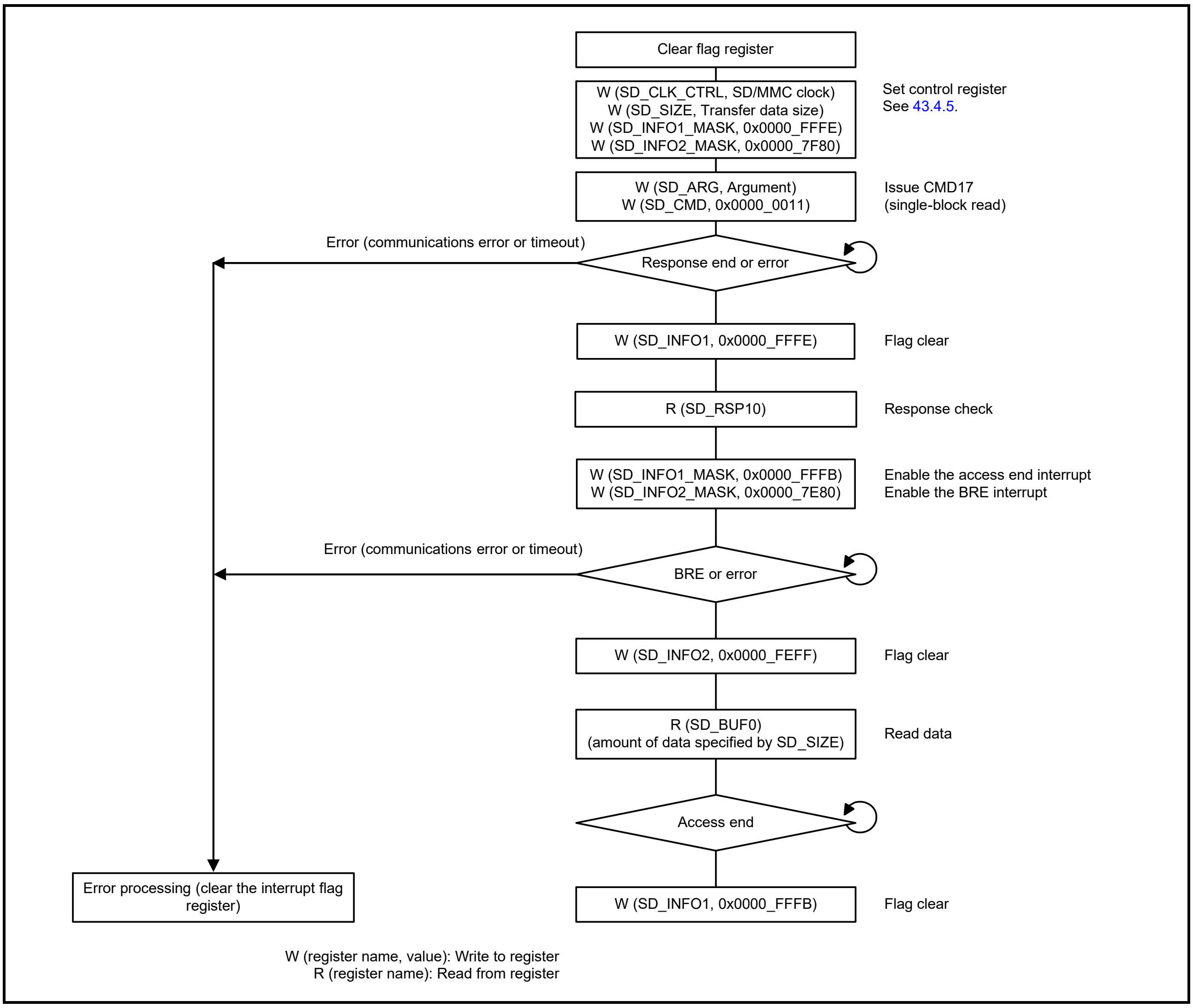


Figure 43.11 Example flow of single block read operation

43.3.6.1 Single block read operation

The operation of the single block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD17)
Set CMD17 argument in SD_ARG and write 0x0000_0011 to SD_CMD. CMD17 is issued and the single block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO1 is set, halting the command sequence also leads to the generation of an interrupt.

e. Data receive from SD card/MMC and data read
Write 0x0000_FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_7E80

to SD_INFO2_MASK to enable the BRE interrupt. When the data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified in SD_SIZE from SD_BUF0.
A communication error or timeout might be generated if data is being received while reading of SD_BUF0 is in progress.

f. Operation complete
When the data read from SD_BUF0 is completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to end the single block read operation.
Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

43.3.7 Single Block Write (SD/MMC)

Figure 43.12 shows an example flow of a single block write operation.

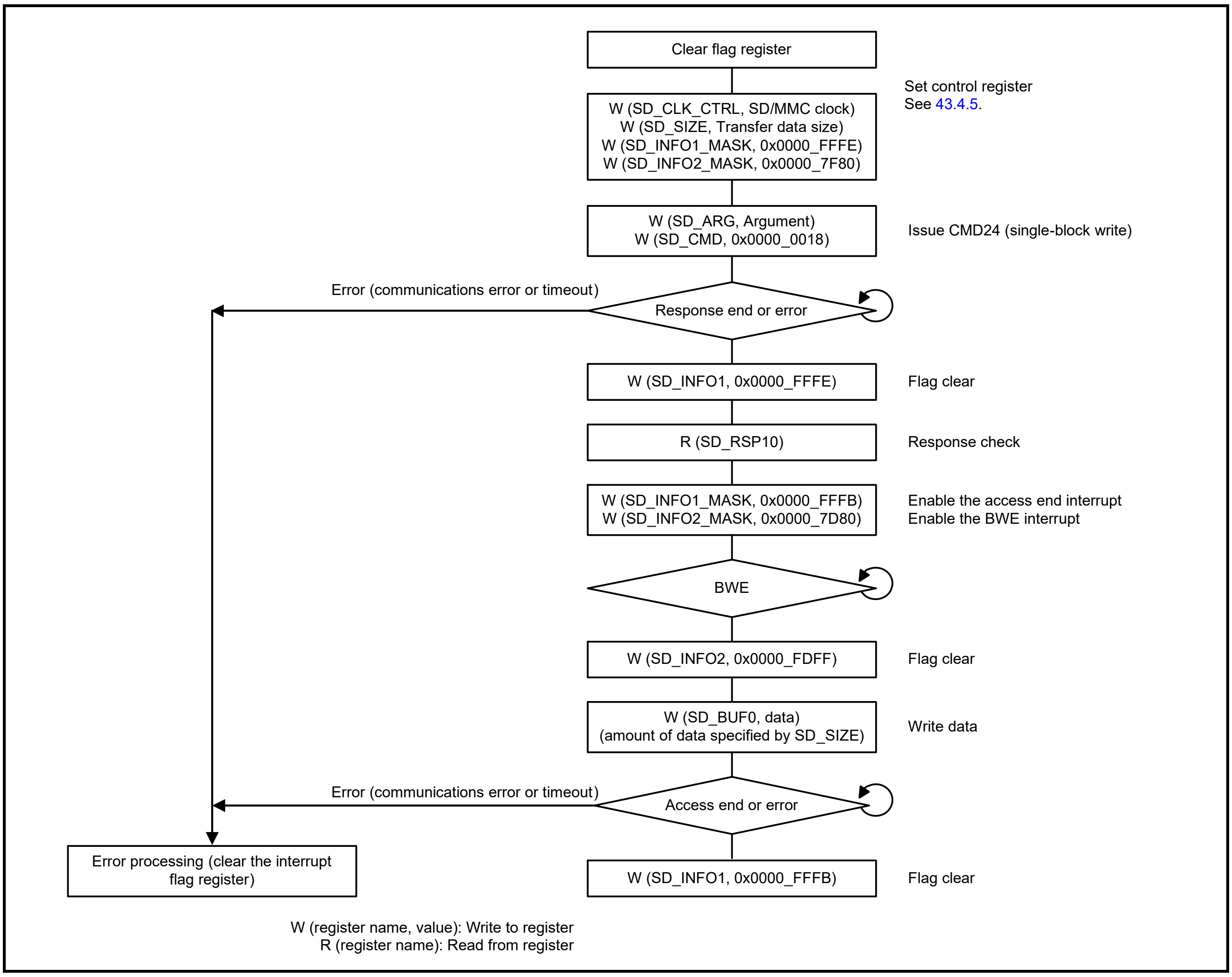


Figure 43.12 Example of single block write operation

43.3.7.1 Single block write operation

The operation of the single block write is described as follows:

a. Flag register clear
   First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
   Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD24)
   Set CMD24 argument in SD_ARG and write 0x0000_0018 to SD_CMD. CMD24 is issued and the single block write operation is started.

d. Response check
   On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO is set, halting the command sequence also leads to the generation of an interrupt.

e. Data write and data transmit to SD card/MMC
   Write 0x0000_FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card/MMC.
   However, a communications error or timeout might be generated if data is being transmitted after writing to SD_BUF0.

f. Operation complete
   When the CRC status and busy state are received from the SD card/MMC, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to end the single block write operation.
   In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

43.3.8 Multiple Block Read (SD/MMC)

Figure 43.13 shows an example flow of a multiple block read operation.

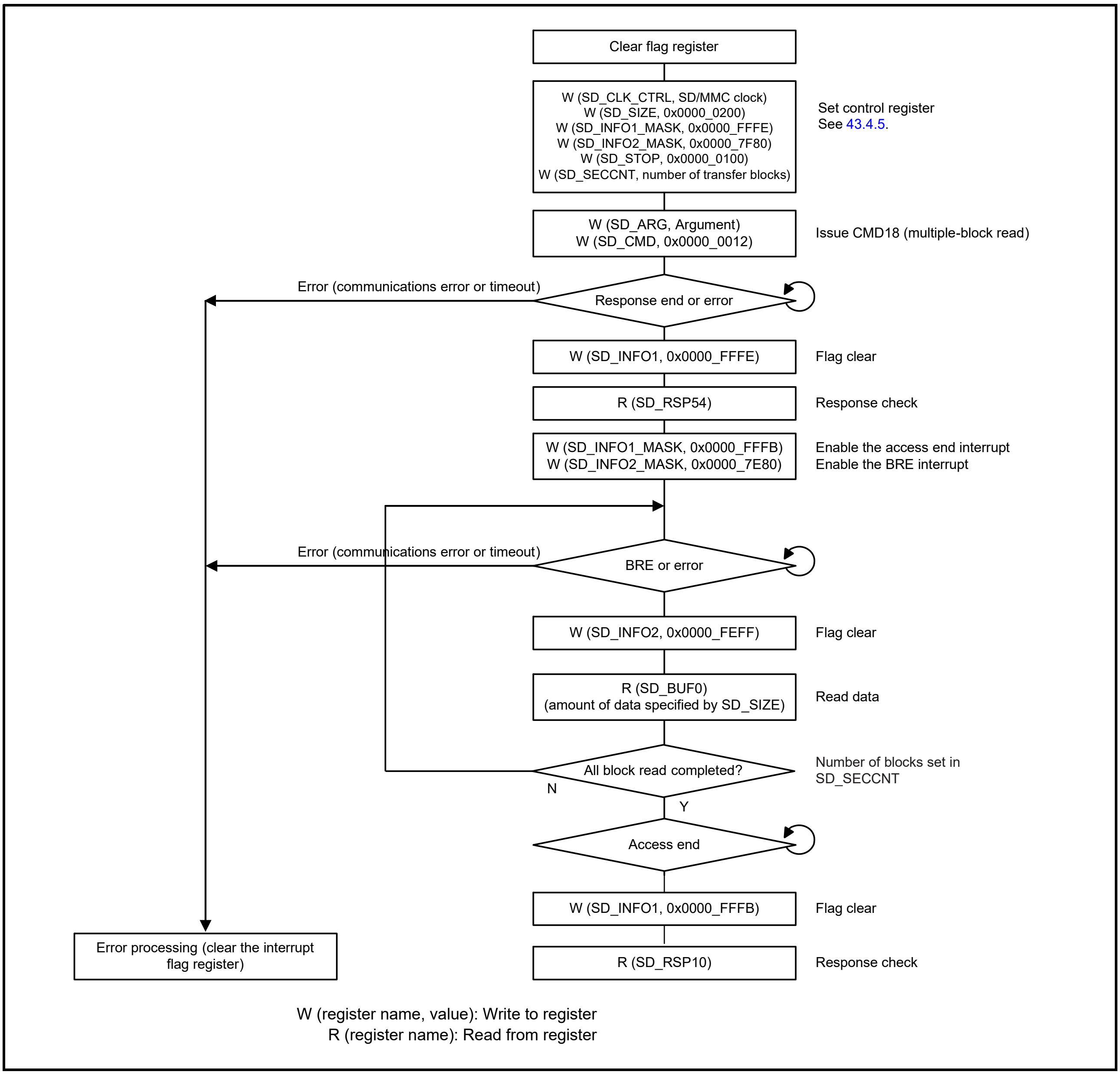


Figure 43.13 Example of multiple block read operation

43.3.8.1 Multiple block read operation

The operation of the multiple block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set SEC in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD18)
Set CMD18 argument in SD_ARG and write 0x0000_0012 to SD_CMD. CMD18 is issued and the multiple block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear

RSPEND to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data receive from SD card/MMC and data read
Write 0x0000_FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_7E80 to SD_INFO2_MASK to enable the BRE interrupt. When one-block data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified in SD_SIZE from SD_BUF0. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while reading of SD_BUF0 is in progress. CMD12 is automatically issued to stop multiblock transfer with the number of blocks that is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0x0000_0000.

f. Operation complete
When all-block data read and the CMD12 response received are completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to read the response. This is the end of multiple block read operation. In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

43.3.9 Multiple Block Write (SD/MMC Using Internal Timer)

Figure 43.14 shows an example flow of a multiple block write using internal timer.

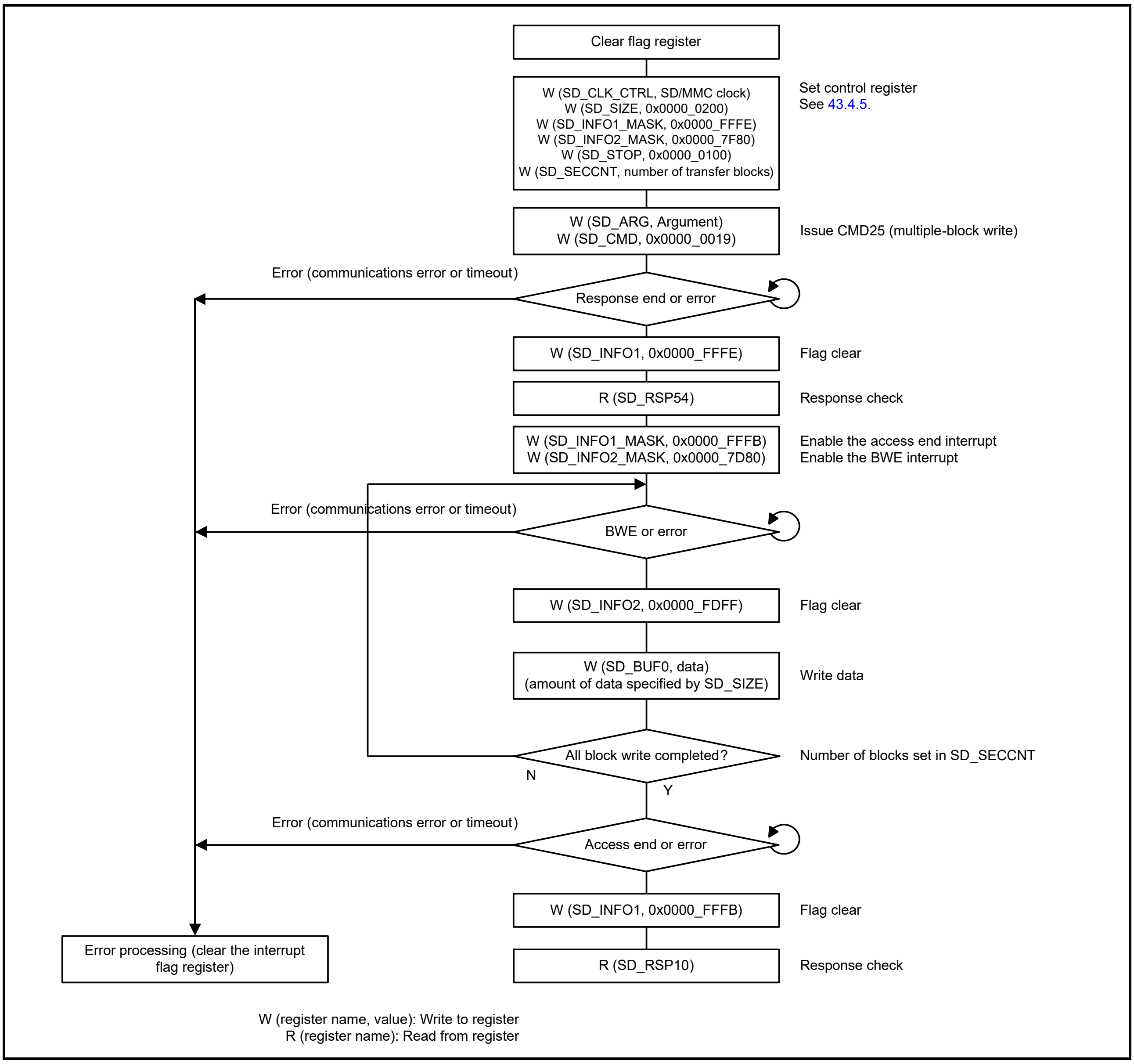


Figure 43.14 Example of multiple block write operation using internal timer

43.3.9.1 Multiple block write operation using internal timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD25)
Set CMD25 argument in SD_ARG and write 0x0000_0019 to SD_CMD. CMD25 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error,

the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to SD card/MMC
Write 0x0000_FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0x0000_7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card/MMC. The CRC status and busy state are received from the SD card/MMC. This repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while writing to SD_BUF0 is in progress. CMD12 is automatically issued to stop multiblock transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0x0000_0000.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communications error or timeout occurs.

43.3.10 Multiple Block Write (MMC using external timer)

Figure 43.15 shows an example flow of a multiple block write using an external timer.

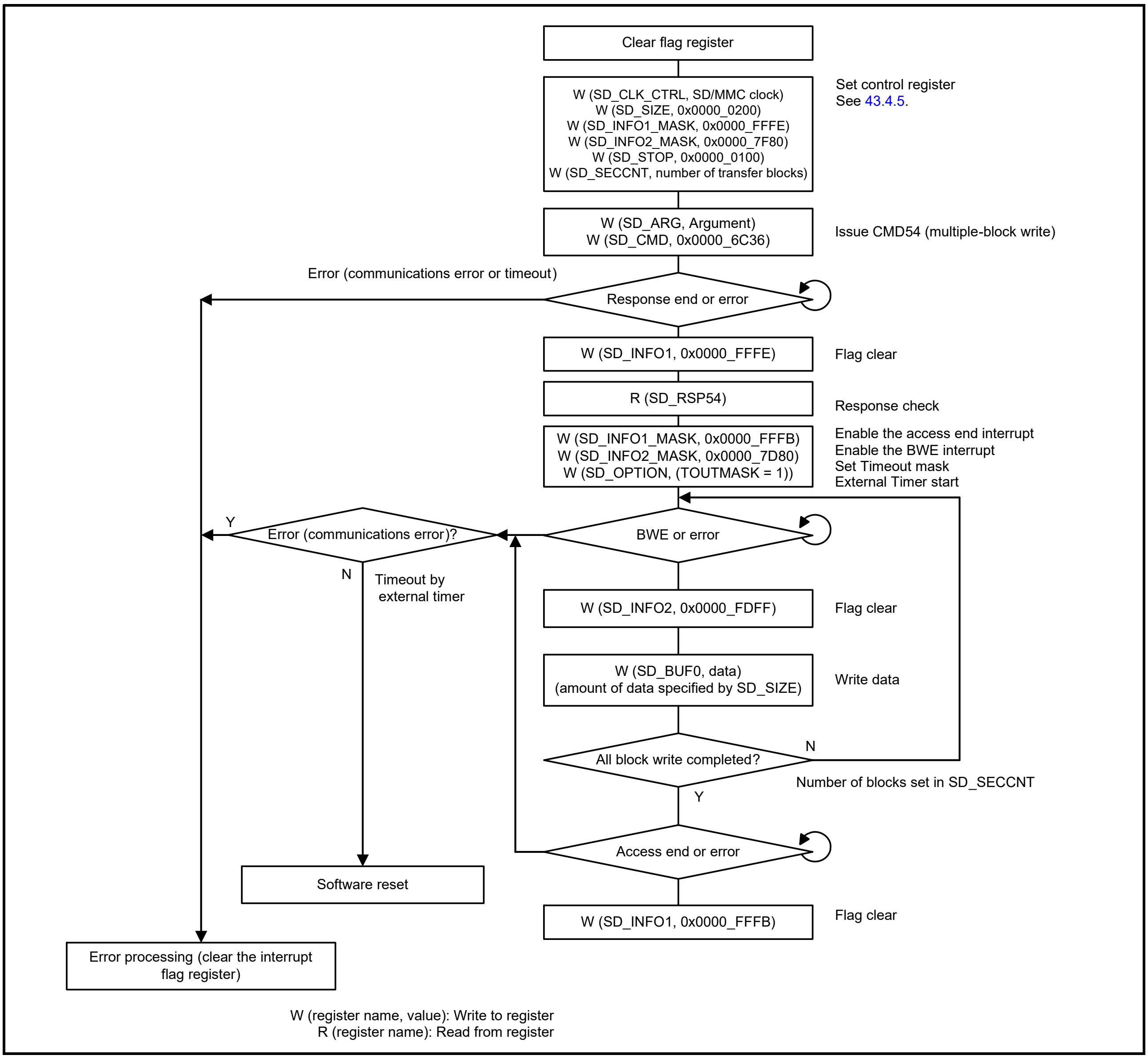


Figure 43.15 **Example of multiple block write operation using external timer**

43.3.10.1 Multiple block write operation using external timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD54)
Set CMD54 Argument in SD_ARG and write 0x0000_6C36 to SD_CMD. CMD54 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes

CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by when the ACEND bit (access end) bit in SD_INFO1 sets to 1 on completion of response reception. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to MMC
Write 0x0000_FFFB to SD_INFO1_MASK to enable the access end interrupt, write 0x0000_7D80 to SD_INFO2_MASK to enable the BWE interrupt and set 1 to TOUTMASK of SD_OPTION to inactivate timeout. In addition, start external timer. When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified in SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the MMC. The CRC status and busy state are received from the MMC. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data is being received while writing to SD_BUF0 is in progress.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communications error or timeout occurs when receiving response. Execute software reset if a timeout by external timer occurs when transmitting data.

43.3.11 IO_RW_DIRECT Command (SD: CMD52)

Figure 43.16 shows an example flow of an IO_DIRECT command (CMD52) operation.

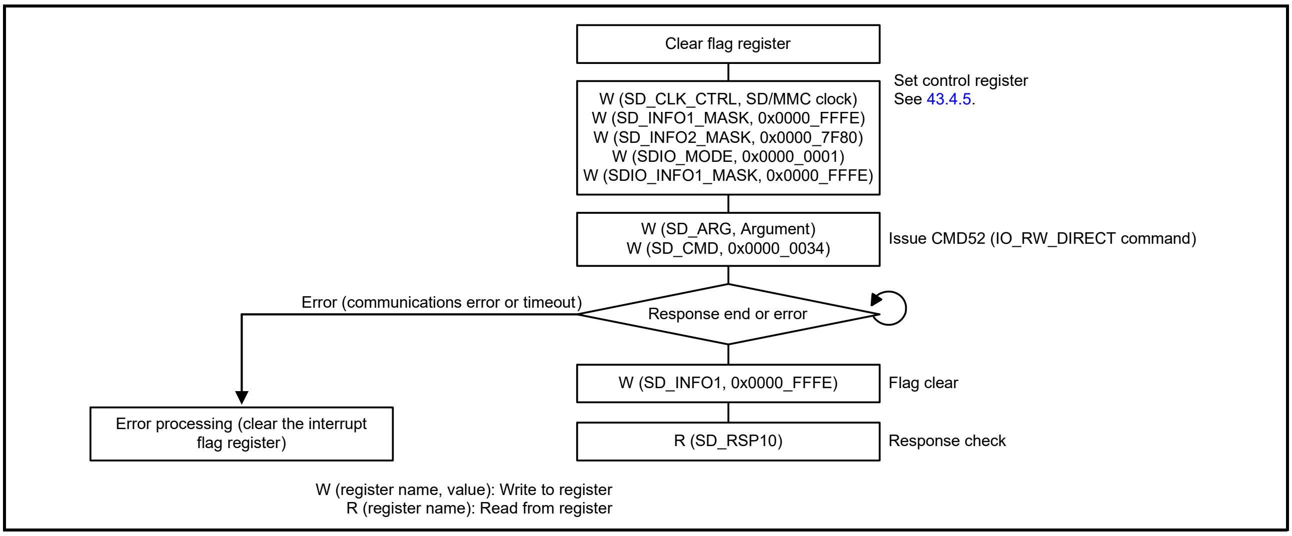


Figure 43.16 Example of IO_RW_DIRECT command (CMD52) operation

43.3.12 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read)

Figure 43.17 shows an example flow for a CMD53 multiple block read operation.

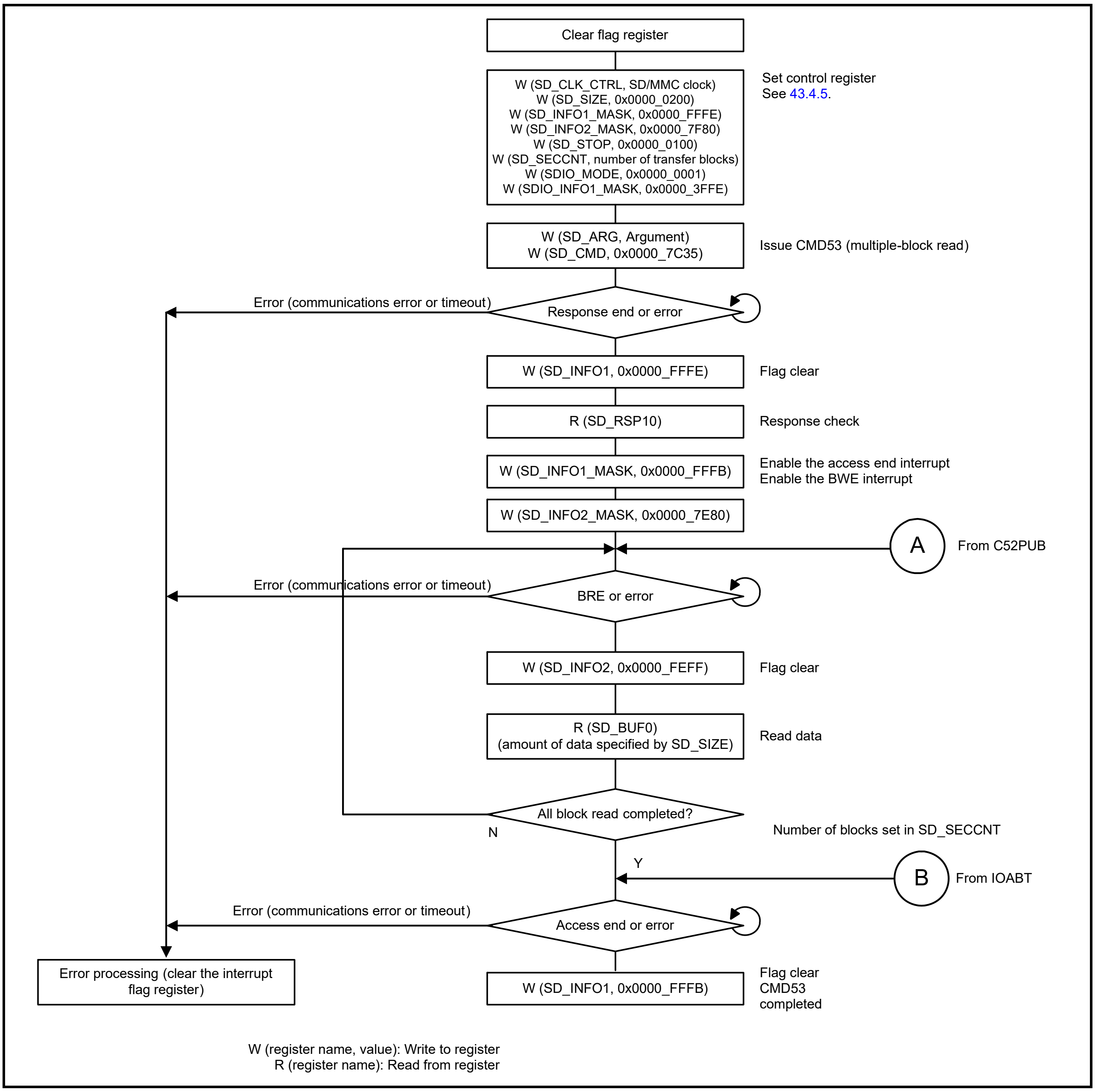


Figure 43.17 Example of IO_RW_EXTENDED command (CMD53) for multiple block read operation

Figure 43.18 shows an example flow when CMD52 (SDIO abort) is issued during a CMD53 multiple block read.

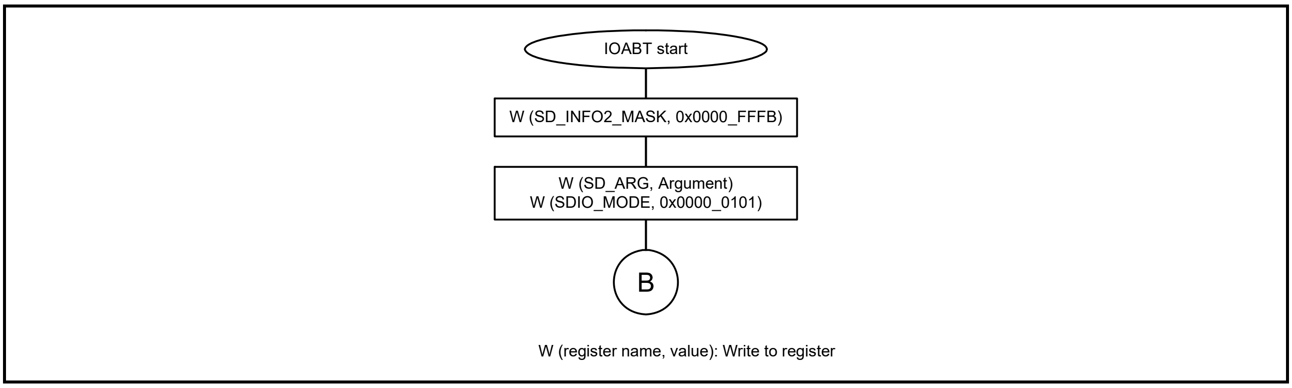


Figure 43.18 Flow when CMD52 (SDIO abort) is issued during a CMD53 multiple block read

Figure 43.19 shows an example flow when CMD52 (SDIO none abort) is issued at a CMD53 multiple block read while the SDHI is in the read wait state.

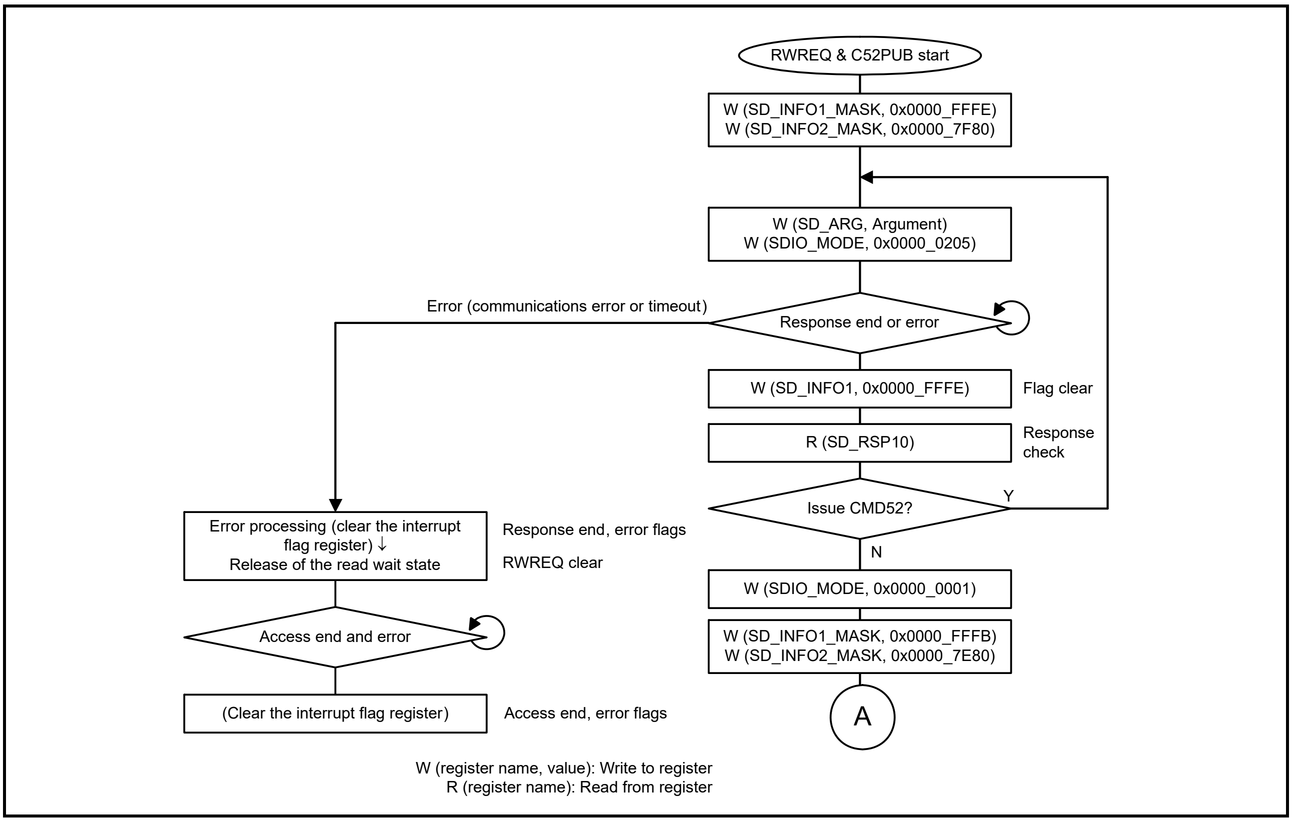


Figure 43.19 Flow when CMD52 (SDIO no abort) is issued during a CMD53 multiple block read while the SD Host Interface is in read wait state

43.3.13 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write)

Figure 43.20 shows an example flow for a CMD53 multiple block write.

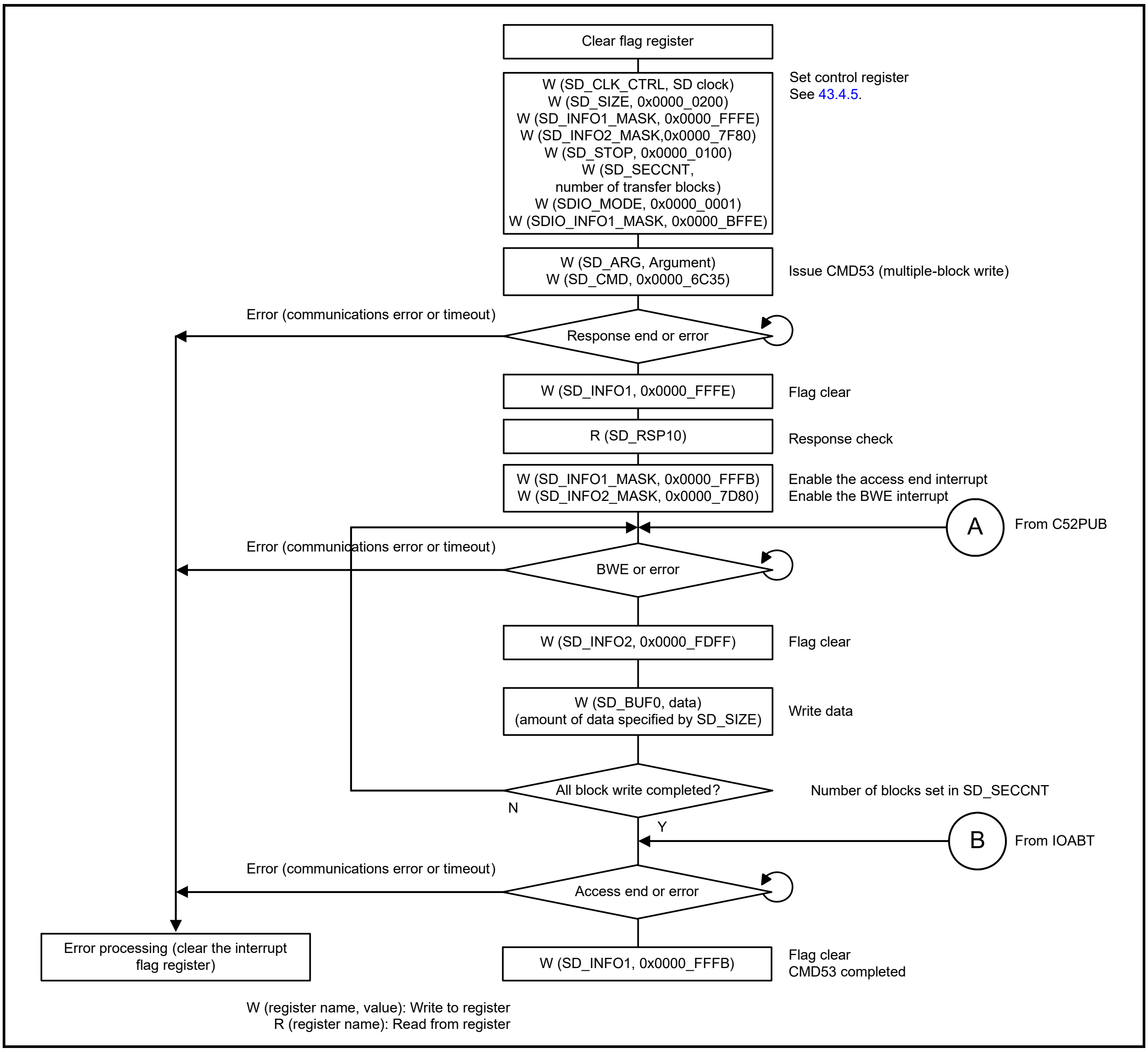


Figure 43.20 Example of IO_RW_EXTENDED command during a CMD53 multiple block write operation

Figure 43.21 shows an example flow when CMD52 (SDIO abort) is issued at CMD53 multiple block write.

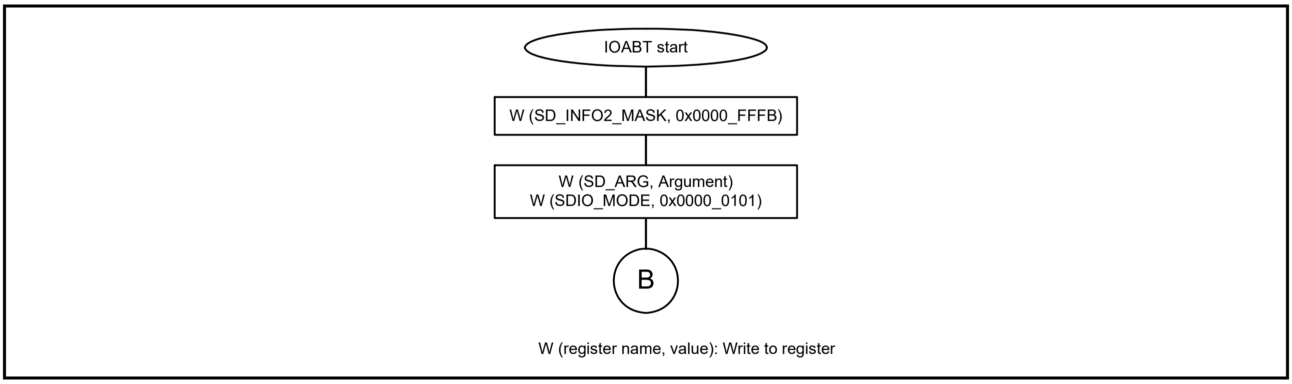


Figure 43.21 Flow when CMD52 (SDIO Abort) is issued during a CMD53 multiple block write

Figure 43.22 shows an example flow when CMD52 (SDIO none abort) is issued at CMD53 multiple block write.

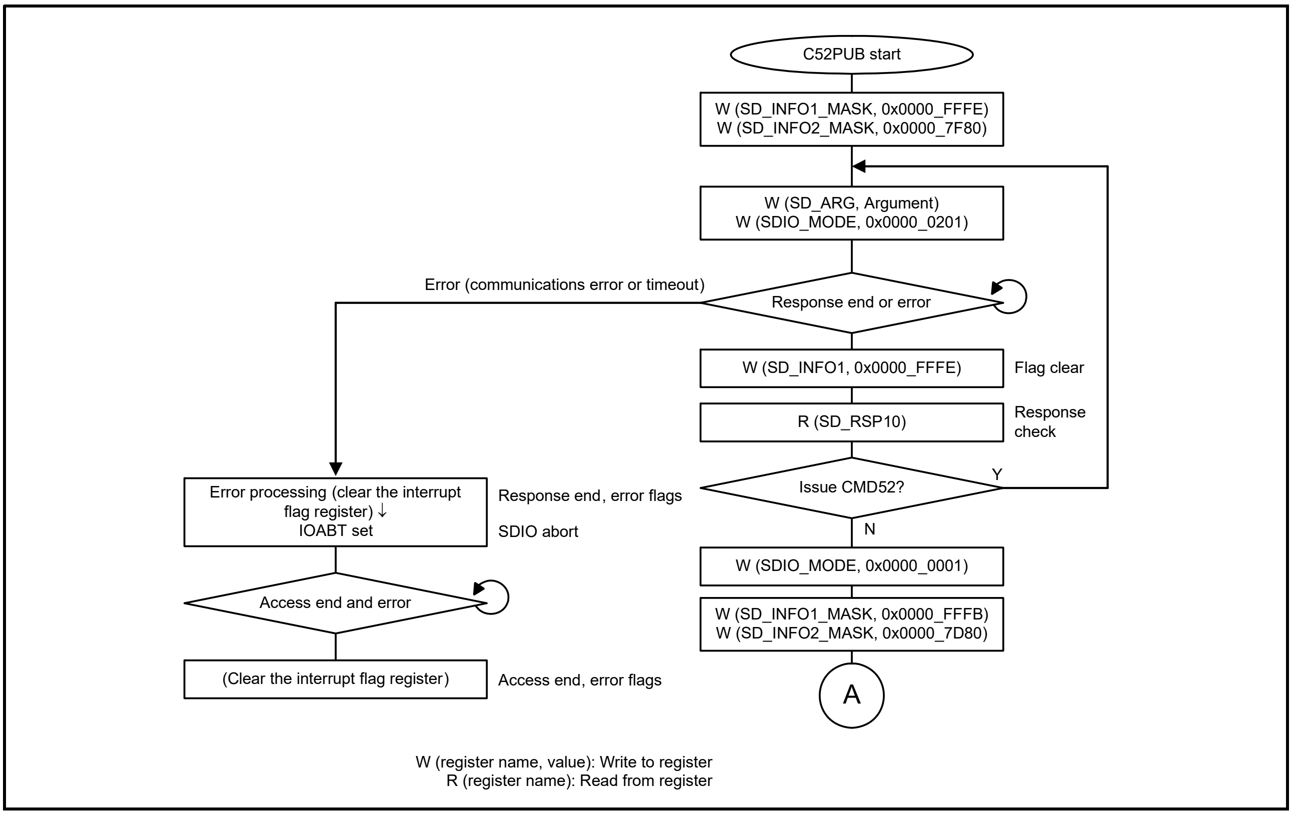


Figure 43.22 Flow when CMD52 (SDIO no abort) is issued during a CMD53 multiple block write

43.3.14 DMA Transfer (SD/MMC)

43.3.14.1 SD_BUF DMA transfer

Figure 43.23 shows an example flow for SD_BUF DMA read when CMD18 multiple block read is issued.

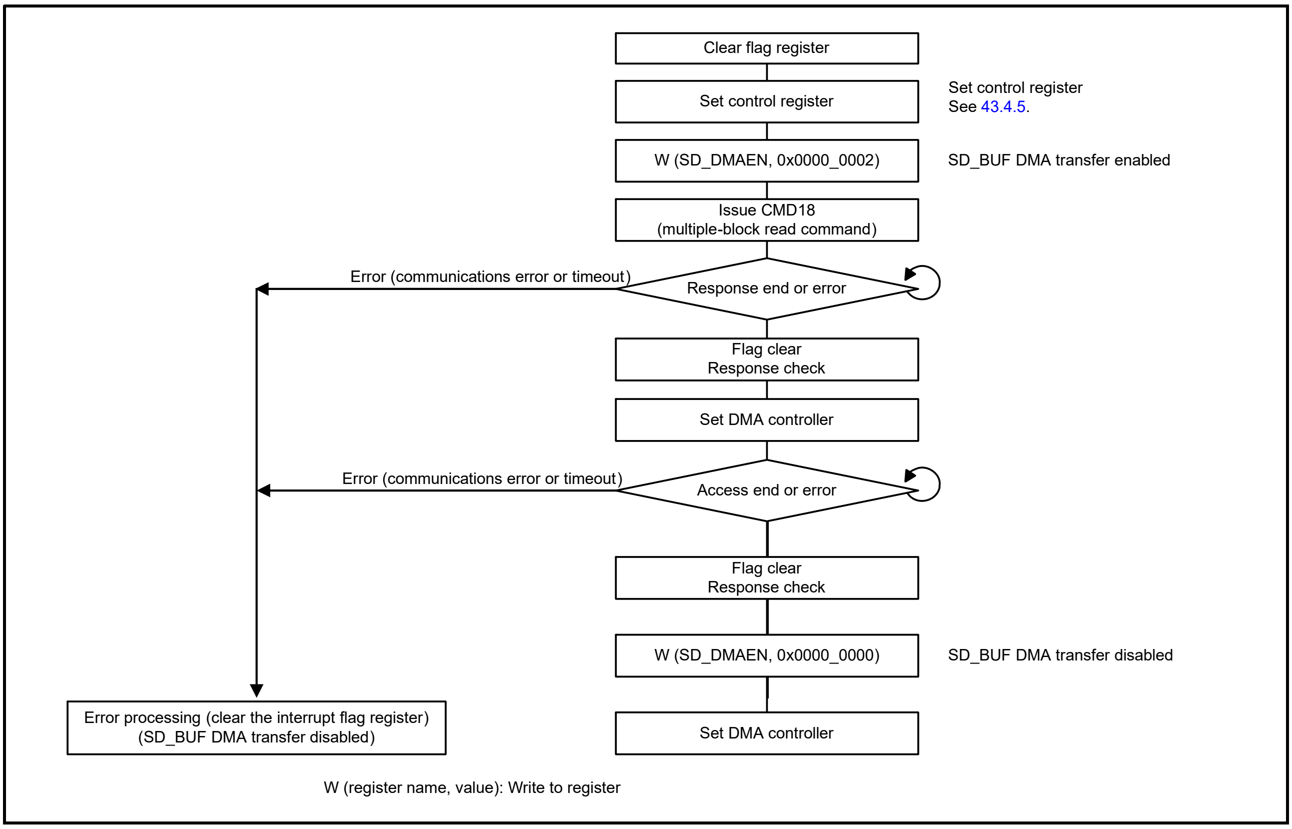


Figure 43.23 Example of SD_BUF_DMA read operation

Figure 43.24 shows an example flow for SD_BUF DMA write when CMD25 multiple block write is issued.

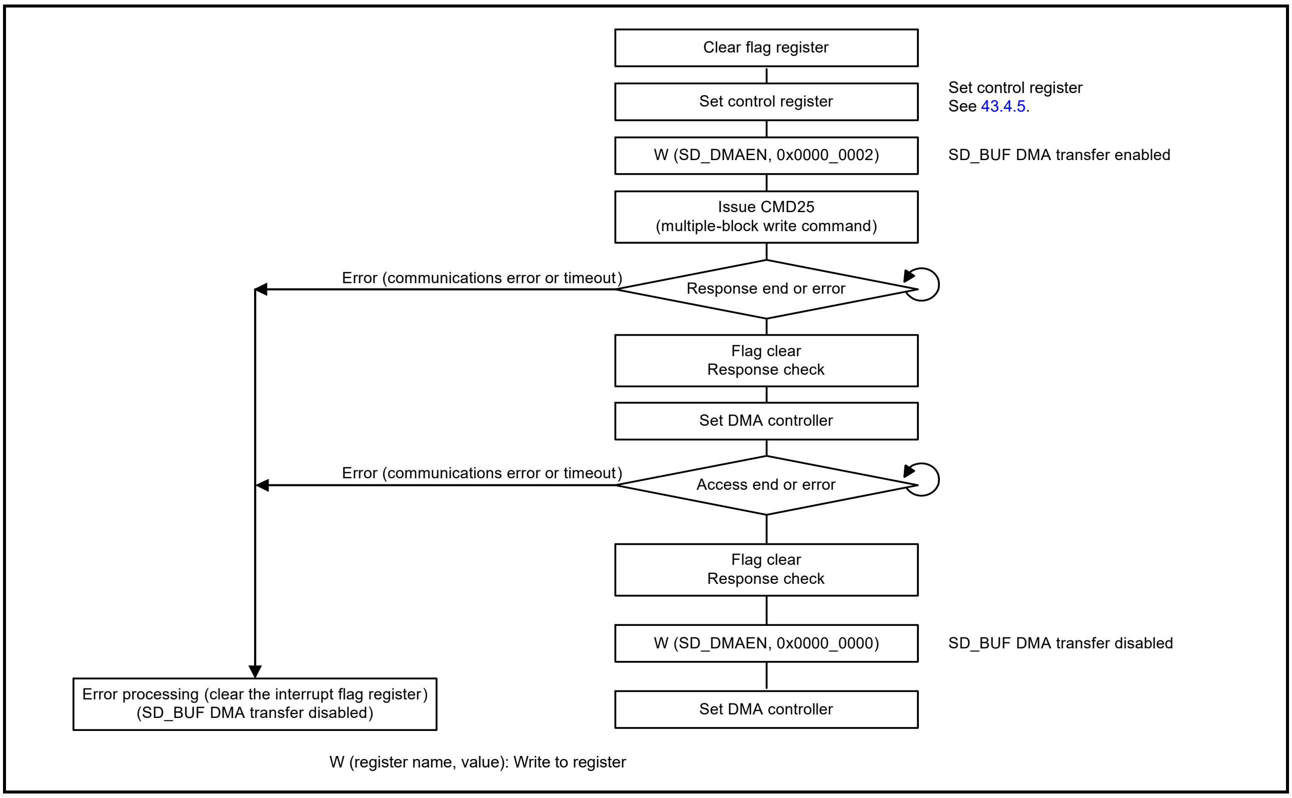


Figure 43.24 Example of SD_BUF_DMA write operation

43.3.15 Example of SD_CMD Register Setting

Table 43.8 and Table 43.9 list the SD_CMD register setting.

Table 43.8 Example SD_CMD register settings for SD (1 of 2)

Type	Commamd	Example SD_CMD register setting	Remark
CMD	CMD0	0000_0000h	
	CMD2	0000_0002h	
	CMD3	0000_0003h	
	CMD4	0000_0004h	
	CMD5	0000_0705h or 0000_0005h	
	CMD6	0000_1C06h or 0000_0006h	
	CMD7	0000_0007h	When the card is placed in the deselected state, the response timeout flag sets because there is no response.
	CMD8	0000_0408h or 0000_0008h	
	CMD9	0000_0009h	
	CMD10	0000_000Ah	
	CMD11	0000_040Bh or 0000_000Bh	
	CMD12	0000_000Ch	
	CMD13	0000_000Dh	
	CMD15	0000_000Fh	
	CMD16	0000_0010h	
	CMD17	0000_0011h	
	CMD18	0000_0012h	With automatic CMD12
	CMD20	0000_0514h or 0000_0014h	
	CMD24	0000_0018h	
	CMD25	0000_0019h	With automatic CMD12
	CMD27	0000_001Bh	
	CMD28	0000_001Ch	
	CMD29	0000_001Dh	
	CMD30	0000_001Eh	
	CMD32	0000_0020h	
	CMD33	0000_0021h	
	CMD38	0000_0026h	
	CMD42	0000_002Ah	
	CMD52	0000_0434h or 0000_0034h	
	CMD53	0000_1C35h	Single read
		0000_0C35h	Single write
		0000_7C35h	Multiple read
		0000_6C35h	Multiple write
		0000_0035h	The value on the left can be set for both single and multiple operations. However, the CF39 bit in SD_ARG must be set as follows. Read: 0 Write: 1
	CMD55	0000_0037h	
	CMD56	0000_0038h	

Table 43.8 Example SD_CMD register settings for SD (2 of 2)

Type	Commamd	Example SD_CMD register setting	Remark
ACMD	ACMD6	0000_0046h	
	ACMD13	0000_004Dh	
	ACMD22	0000_0056h	
	ACMD23	0000_0057h	
	ACMD41	0000_0069h	
	ACMD42	0000_006Ah	
	ACMD51	0000_0073h	

Table 43.9 Example SD_CMD register settings for MMC (1 of 2)

Type	Command	Example SD_CMD register setting	Remark
CMD	CMD0	0000_0000h	
	CMD1	0000_0701h	
	CMD2	0000_0002h	
	CMD3	0000_0003h	
	CMD4	0000_0004h	
	CMD5	0000_0505h	
	CMD6	0000_0506h	(with response busy)
		0000_0406h	(without response busy)
	CMD7	0000_0007h	When the card is placed in the deselected state, the response timeout flag sets because there is no response.
	CMD8	0000_1C08h	
	CMD9	0000_0009h	
	CMD10	0000_000Ah	
	CMD12	0000_000Ch	
	CMD13	0000_000Dh	
	CMD14	0000_1C0Eh	Required setting: SD_IFMODE = 0000_0100h (CRC check is invalid)
	CMD15	0000_000Fh	
	CMD16	0000_0010h	
	CMD17	0000_0011h	
	CMD18	0000_7C12h	Pre-defined
	CMD19	0000_0C13h	Required setting: SD_IFMODE = 0000_0100h (CRC check is invalid)
	CMD21	0000_1C15h	DDR mode is inhibited
	CMD23	0000_0017h	
	CMD24	0000_0018h	
	CMD25	0000_6C19h	Pre-defined
	CMD26	0000_0C1Ah	
	CMD27	0000_001Bh	
	CMD28	0000_001Ch	
	CMD29	0000_001Dh	
	CMD30	0000_001Eh	
	CMD31	0000_1C1Fh	

Table 43.9 Example SD_CMD register settings for MMC (2 of 2)

Type	Command	Example SD_CMD register setting	Remark
CMD	CMD35	0000_0423h	-
	CMD36	0000_0424h	-
	CMD38	0000_0026h	-
	CMD39	0000_0427h	-
	CMD40	0000_0428h	-
	CMD42	0000_002Ah	-
	CMD49	0000_0C31h	-
	CMD53	0000_7C35h	-
	CMD54	0000_6C36h	-
	CMD55	0000_0037h	-
	CMD56	0000_0038h	-

43.4 Usage Notes

43.4.1 SD_BUF Illegal Write Access (SD/MMC)

When writing data to SD_BUF0 after the single block write or multi block write command is issued, the data of the size specified in SD_SIZE must be written.

If the data exceeds the size specified in SD_SIZE is written, the ERR4 bit in SD_INFO2 is set to 1. In addition, the data written to SD_BUF0 might not be transmitted and the SD_CLK_CTRLEN bit in SD_INFO2 is held at the value of 0. If this occurs, clearing the SDRST bit in SOFT_RST to 0 and then restoring its value to 1 clears the SD_CLK_CTRLEN bit to 1.

However, this does not apply to the single byte or three bytes when the SD_SIZE setting is odd, or to the fraction of bytes when the SD_SIZE setting is even (the 2 bytes that are not in a 4-byte unit), because the portion of dummy data writing is regarded as excess data and ignored.

43.4.2 Block Number Constraint for Multiple Block Read (SD)]

When performing a multiple block read of one or two blocks, depending on the timing with which the SD card response register is read, the response value might not be read properly. To prevent this, do one of the following:

1. When receiving one or two blocks of data, use single block reading.
2. Read the response to CMD18 from SD_RSP54.

43.4.2.1 Mechanism of incorrect reading

Figure 43.25 shows the processing flows of the SDHI (hardware) operation and software operation when a multiple block read is performed on two blocks. As shown in the incorrect operation in Figure 43.25, when an interrupt is generated on reception of the CMD18 response and the timing with which the SD card response register (SD_RSP10) is read by the interrupt is delayed, the data during the CMD12 response reception or the CMD12 response might be read. This problem does not occur for multiple block reads of three or more blocks, because CMD12 is not issued until the block of data is read. The problem also does not occur for multiple block writes, because the CMD25 response is read before the block of data is sent.

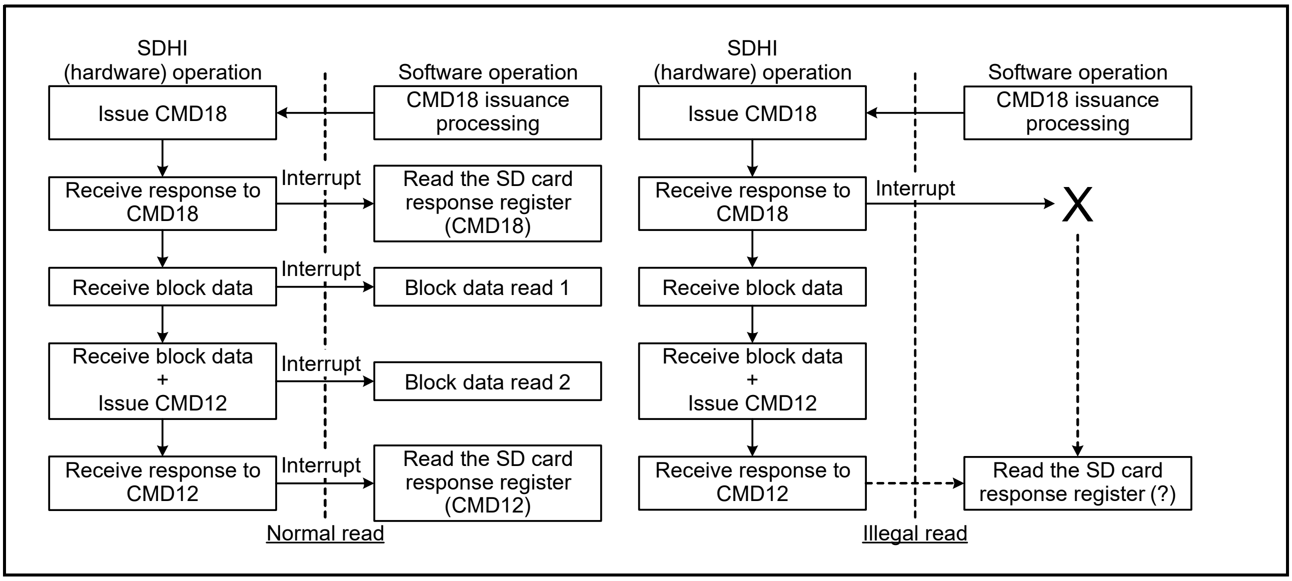


Figure 43.25 Multiple block read operation flow chart (two blocks)

43.4.3 Automatic Control of SD/MMC Clock Output (SD/MMC)

In the SD Card/MMC standard, 74 cycles of SD/MMC clock must be output before initialization of the card. For this reason, use automatic control of SD/MMC clock output after 74 cycles of SD/MMC clock are output. In addition, if automatic control of SD/MMC clock output was in use, SD/MMC clock output is stopped on completion of the sequence for a communications error or timeout. When state transitions within the SD card/MMC are necessary after completion of the sequence, release automatic control of SD/MMC clock output and restart supply of the SD/MMC clock to the SD card/MMC.

43.4.4 Control of the C52PUB Setting for Multiple Block Write (SD)

If the C52PUB bit in SDIO_MODE is set to 1 during a sequence of multiple block write because of CMD53, CMD52 is not issued until SD_BUF becomes empty. For this reason, set the C52PUB bit after suspending writing to SD_BUF by using one of the following procedures, as appropriate:

(a) When DMA transfer is not in use

1. Before setting the C52PUB bit, suspend writing to SD_BUF by making the setting in SD_INFO2 to disable BWE interrupts.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 because the issuing of CMD52 is completed, restart writing to SD_BUF by making the setting in SD_INFO2 to enable BWE interrupts.

(b) When DMA transfer is in use

1. Every time DMA transfer of the value set in SD_SIZE × n blocks (where n = 1, 2,....) proceeds, suspend writing to SD_BUF by DMA transfer before the C52PUB bit is set.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 because the issuing of CMD52 is completed, restart writing to SD_BUF by DMA transfer.

43.4.5 Notes on SD_CLK_CTRL Register Settings (SD/MMC)

When the SD_CLK_CTRLEN bit in SD_INFO2 is 0, SD_CLK_CTRL cannot be written to. Before writing to SD_CLK_CTRL, you must check that the SD_CLK_CTRLEN bit in SD_INFO2 is 1.

43.4.6 Specification Limitations

1. The Suspend/Resume operation of the SDIO is not supported.
2. The SPI bus is not supported. (SD/MMC)
3. The shared bus and 8-bit SD bus of the embedded SDIO are not supported.
4. Stream transfer of MMC is not supported.
5. High Priority Interrupt (HPI) of MMC is not supported.
6. Boot Operation/Alternative Boot Operation of MMC is not supported.
7. Open-ended multiple block transfer of MMC is not supported.

43.4.7 STP Bit Setting during Multiple Block Read (SD/MMC)

During execution of multiple block read with automatic CMD12 execution by setting the SEC bit in SD_STOP to 1, even if the STP bit in SD_STOP is set to 1 to forcibly stop the execution, the command sequence might not stop depending on the timing of setting the STP bit.

To avoid this, when setting the STP bit in SD_STOP to 1 during multiple block transfer, clear the SEC bit in SD_STOP to 0 at the same time. (Even when the SD_CLK_CTRLEN bit in SD_INFO2 is 0, change the SEC bit from 1 to 0.)

When the command sequence does not stop because the SEC bit was not cleared to 0, the command sequence can be stopped by clearing the SDRST bit in SOFT_RST to 0.

When forcibly terminating the CMD53 multiple block transfer through the IOABT bit in SDIO_MODE, you must leave the SEC bit in SD_STOP as 1.

43.4.8 Register Setting Notes

1. All registers in section 43.2, Register Descriptions are accessed in 32-bit access-only.
2. When setting registers, set them after the I/O Port Register setting.

44. Parallel Data Capture Unit (PDC)

44.1 Overview

The MCU provides one Parallel Data Capture Unit (PDC). The PDC communicates with external I/O devices, including image sensors, and transfers parallel data such as an image output from the external I/O device through the DTC or DMAC to the on-chip SRAM and external address spaces (the CS and SDRAM areas). Table 44.1 lists the PDC specifications, Figure 44.1 shows a block diagram, and Table 44.2 lists the I/O pins.

Table 44.1 PDC specifications

Parameter	Specifications
Capture range	Any amount of parallel data within the following ranges in the vertical and horizontal directions: • Vertical direction: 1 to 4095 lines • Horizontal direction: 4 to 4095 bytes
Parallel transfer clock (PIXCLK)	Operating frequency: 1 to 27 MHz*1
Interrupt sources	• Receive data ready • Frame end • Overrun • Underrun • Error in the setting for the number of lines • Error in setting for the number of bytes per line
Startup of DTC or DMAC	Frame end and receive data ready interrupts can start DTC or DMAC
Parallel transfer clock output (PCKO)	• Operating frequency: 1 to 30 MHz*2 • Clock source: Peripheral module clock B (PCLKB) • Frequency division ratio: Selectable from 2, 4, 6, 8, 10, 12, 14, and 16.
Other functions	• PDC reset function • Selectable active polarity for VSYNC and HSYNC signals • Monitoring of VSYNC and HSYNC signals • Endian order selectable
Module-stop function	Module-stop state can be set to reduce power consumption
Internal bus interface	Internal peripheral bus 5

Note 1. Set the frequency of the parallel data transfer clock (PIXCLK) to a value less than that of 0.6 × PCLKB (peripheral module clock).

Note 2. The operating frequency is 30 MHz when peripheral module clock B (PCLKB) is 60 MHz and the frequency division ratio is 2.

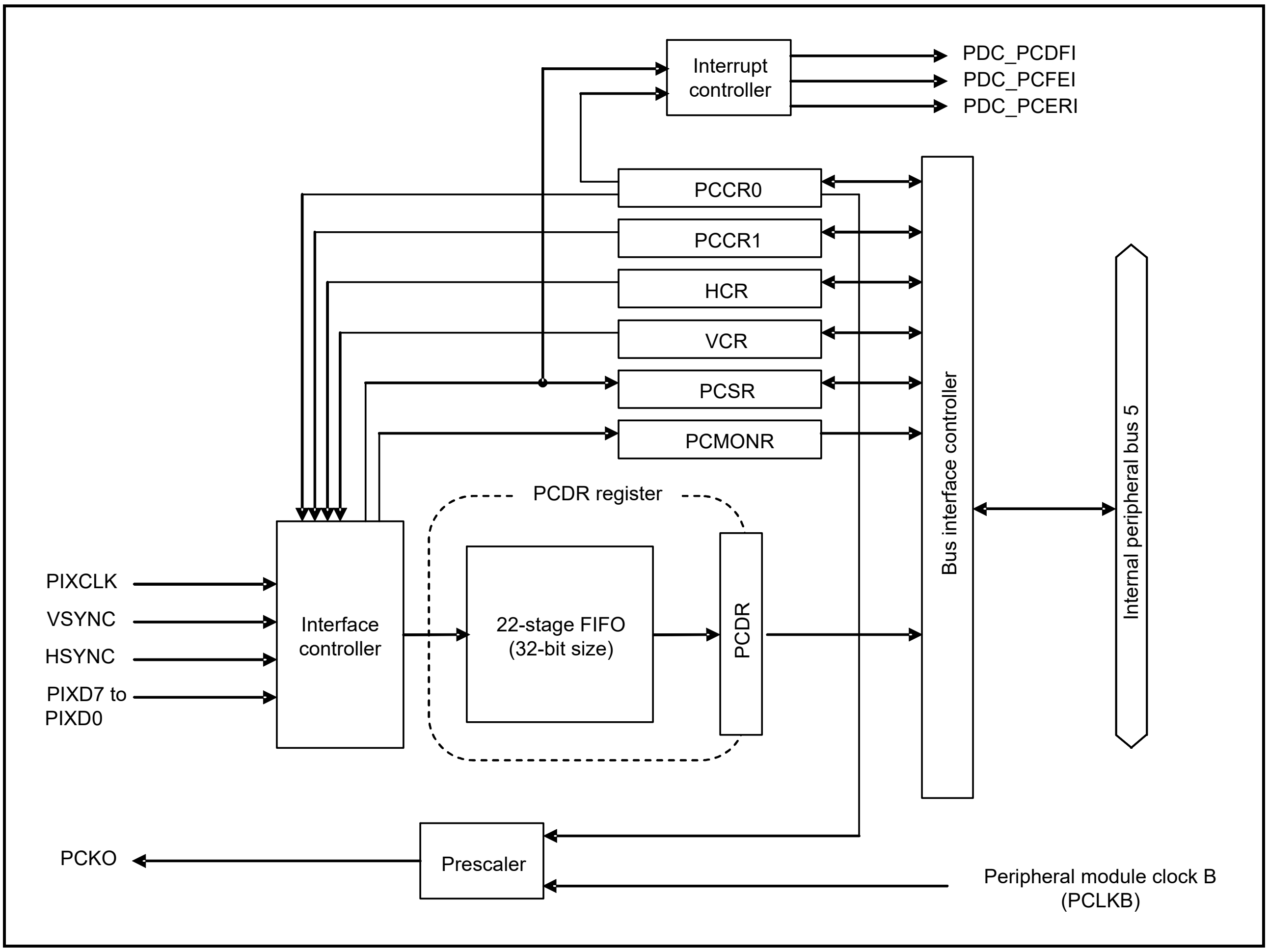


Figure 44.1 PDC block diagram

Table 44.2 PDC I/O pins

Pin name	I/O	Description
PIXCLK	Input	Parallel transfer clock
VSYNC	Input	Vertical synchronization signal
HSYNC	Input	Horizontal synchronization signal
PIXD7 to PIXD0	Input	8-bit data
PCKO	Output	Output for the parallel transfer clock

44.2 Register Descriptions

44.2.1 PDC Control Register 0 (PCCR0)

Address(es): PDC.PCCR0 4009 4000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	EDS	PCKDIV[2:0]			PCKOE	HERIE	VERIE	UDRIE	OVIE	FEIE	DFIE	PRST	HPS	VPS	PCKE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PCKE	PIXCLK Input Enable	0: Disable PIXCLK input 1: Enable PIXCLK input.	R/W
b1	VPS	VSYNC Signal Polarity Select	0: Set VSYNC signal to active high 1: Set VSYNC signal to active low.	R/W
b2	HPS	HSYNC Signal Polarity Select	0: Set HSYNC signal to active high 1: Set HSYNC signal to active low.	R/W
b3	PRST	PDC Reset	0: Do not apply PDC reset 1: Reset PDC.	R/(W) *1
b4	DFIE	Receive Data Ready Interrupt Enable	0: Disable receive data ready interrupt requests 1: Enable receive data ready interrupt requests.	R/W
b5	FEIE	Frame End Interrupt Enable	0: Disable frame end interrupt requests 1: Enable frame end interrupt requests.	R/W
b6	OVIE	Overrun Interrupt Enable	0: Disable overrun interrupt requests 1: Enable overrun interrupt requests.	R/W
b7	UDRIE	Underrun Interrupt Enable	0: Disable underrun interrupt requests 1: Enable underrun interrupt requests.	R/W
b8	VERIE	Vertical Line Number Setting Error Interrupt Enable	0: Disable vertical line number setting error interrupt requests 1: Enable vertical line number setting error interrupt requests.	R/W
b9	HERIE	Horizontal Byte Number Setting Error Interrupt Enable	0: Disable horizontal byte number setting error interrupt requests 1: Enable horizontal byte number setting error interrupt requests.	R/W
b10	PCKOE	PCKO Output Enable	0: Disable PCKO output (fix to high level) 1: Enable PCKO output.	R/W
b13 to b11	PCKDIV[2:0]	PCKO Frequency Division Ratio Select	b13 b11 0 0 0: PCLKB/2 0 0 1: PCLKB/4 0 1 0: PCLKB/6 0 1 1: PCLKB/8 1 0 0: PCLKB/10 1 0 1: PCLKB/12 1 1 0: PCLKB/14 1 1 1: PCLKB/16.	R/W
b14	EDS	Endian Select	0: Little endian 1: Big endian.	R/W
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit.

Only set the PCCR0 register while the PCE bit in the PCCR1 register is 0.

PCKE bit (PIXCLK Input Enable)

The PCKE bit enables or disables input through the PIXCLK pin. Set this bit to 1 before enabling reception. After enabling input through the PIXCLK pin, use the PRST bit to initialize the PDC.

Disable reception operations before setting this bit to 0.

VPS bit (VSYNC Signal Polarity Select)

The VPS bit selects the active polarity of the VSYNC signal.

HPS bit (HSYNC Signal Polarity Select)

The HPS bit selects the active polarity of the HSYNC signal.

PRST bit (PDC Reset)

The PRST bit initializes the internal status of the PDC and the PDC registers targeted by reset. See section 44.3.11, Reset State, for the target registers. Set the PCKE bit to 1 before resetting the PDC.

When 1 is written to the PRST bit, initialization starts in synchronization with the PIXCLK. After initialization completes, the PRST bit clears to 0. After a PDC reset, ensure that the PIXCLK pin has an input signal. Also, after 1 is written to the PRST bit, do not proceed to the next step until verifying that the bit has returned to 0.

For consecutive PDC resets, wait for at least 1 PIXCLK cycle after verifying that the PRST bit has returned to 0.

DFIE bit (Receive Data Ready Interrupt Enable)

The DFIE bit enables or disables the generation of receive data ready interrupt requests.

FEIE bit (Frame End Interrupt Enable)

The FEIE bit enables or disables the generation of frame end interrupt requests.

OVIE bit (Overrun Interrupt Enable)

The OVIE bit enables or disables the generation of overrun interrupt requests.

UDRIE bit (Underrun Interrupt Enable)

The UDRIE bit enables or disables the generation of underrun interrupt requests.

VERIE bit (Vertical Line Number Setting Error Interrupt Enable)

The VERIE bit enables or disables the generation of vertical line number setting error interrupt requests.

HERIE bit (Horizontal Byte Number Setting Error Interrupt Enable)

The HERIE bit enables or disables the generation of horizontal byte number setting error interrupt requests.

PCKOE bit (PCKO Output Enable)

The PCKOE bit enables or disables an output from PCKO. When the PCKOE bit is cleared to 0 during low output of PCKO, it might cause high output on clearing, resulting in corruption of the duty cycle.

PCKDIV[2:0] bits (PCKO Frequency Division Ratio Select)

The PCKDIV[2:0] bits select the frequency division ratio of PCKO. The PCKO output is a clock signal derived by dividing the PCLKB clock signal by a value from 2 to 16, based on the setting in the PCKDIV[2:0] bits. The PCKO operating frequency, the resulting PCLKB division, must fall within the range from 1 to 30 MHz.

EDS bit (Endian Select)

The EDS bit selects the endian order for the captured data.

44.2.2 PDC Control Register 1 (PCCR1)

Address(es): PDC.PCCR1 4009 4004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PCE	PDC Operation Enable	0: Disable reception operations 1: Enable reception operations.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PCE bit (PDC Operation Enable)

The PCE bit enables or disables reception operations. When the PCE bit is set to 1 during assertion of the VSYNC signal, the PDC starts reception operations from the next valid edge of the VSYNC signal.

Only clear the PCE bit to 0 while reception or continued reception operations are stopped, including for the frame end interrupt. For more on continued reception, see section 44.3.6, Continued Reception Operations at Frame End.

44.2.3 PDC Status Register (PCSR)

Address(es): PDC.PCSR 4009 4008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	HERF	VERF	UDRF	OVRF	FEF	FEMPF	FBSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	FBSY	Frame Busy Flag	0: Reception operations are stopped 1: Reception operations are ongoing.	R
b1	FEMPF	FIFO Empty Flag	0: FIFO is not empty 1: FIFO is empty.	R
b2	FEF	Frame End Flag	0: No frame end occurred 1: Frame end occurred.	R/(W) *1
b3	OVRF	Overrun Flag	0: No FIFO overrun occurred 1: FIFO overrun occurred.	R/(W) *1
b4	UDRF	Underrun Flag	0: No underrun occurred 1: Underrun occurred.	R/(W) *1
b5	VERF	Vertical Line Number Setting Error Flag	0: No vertical line number setting error occurred 1: Vertical line number setting error occurred.	R/(W) *1
b6	HERF	Horizontal Byte Number Setting Error Flag	0: No horizontal byte number setting error occurred 1: Horizontal byte number setting error occurred.	R/(W) *1

Bit	Symbol	Bit name	Description	R/W
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to these flags, to clear them after they are read as 1.

FBSY flag (Frame Busy Flag)

The FBSY flag indicates the state of PDC operations.

[Setting condition]

- On detection of the valid edge of the VSYNC signal after the enabling of reception operations.

[Clearing conditions]

- On reception of one frame of data in accordance with the settings in the VCR and HCR registers*1
- When an overrun, underrun, vertical line number setting error, or horizontal byte number setting error occurs
- When the PCCR1.PCE bit is 0.

Note 1. This flag is 0 during continued reception operations.

FEMPF flag (FIFO Empty Flag)

The FEMPF flag indicates the state of the FIFO when a vertical line number setting error or a horizontal byte number setting error occurs. It clears to 0 following an overrun and undefined following an underrun.

[Setting conditions]

- On reading of the PCDR register while the FIFO is empty
- On detection of a valid edge of the VSYNC signal
- On PDC reset.

[Clearing condition]

- On storage of the data captured in the FIFO.

FEF flag (Frame End Flag)

The FEF flag indicates the end of a frame.

[Setting condition]

- Reception of one frame of data in accordance with the settings in the VCR and HCR registers.*1

[Clearing conditions]

- On PDC reset
- When 0 is written to the flag after it is read as 1.

Note 1. For continued reception operations, this flag sets to 1 on their completion.

OVRF flag (Overrun Flag)

The OVRF flag indicates the occurrence of an overrun.

[Setting condition]

- When receive data arrives while the FIFO is full.

[Clearing conditions]

- On PDC reset
- When 0 is written to the flag after it is read as 1.

UDRF flag (Underrun Flag)

The UDRF flag indicates the occurrence of an underrun.

[Setting condition]

- On reading of the PCDR register while the FIFO is empty.

[Clearing conditions]

- On PDC reset
- When 0 is written to the flag after it is read as 1.

VERF flag (Vertical Line Number Setting Error Flag)

The VERF flag indicates an error in the setting for the number of lines.

[Setting condition]

- When the VSYNC signal is negated because fewer lines were captured than the value in the VCR register.

[Clearing conditions]

- On PDC reset
- When 0 is written to the flag after it is read as 1.

HERF flag (Horizontal Byte Number Setting Error Flag)

The HERF flag indicates an error in the number of bytes in a line.

[Setting condition]

- When the HSYNC signal is negated because fewer bytes in a line were captured than the value in the HCR register.

[Clearing conditions]

- On PDC reset
- When 0 is written to the flag after it is read as 1.

44.2.4 PDC Pin Monitor Register (PCMONR)

Address(es): PDC.PCMONR 4009 400Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HSYNC	VSYNC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VSYNC	VSYNC Signal Status Flag	0: VSYNC signal level is low 1: VSYNC signal level is high.	R
b1	HSYNC	HSYNC Signal Status Flag	0: HSYNC signal level is low 1: HSYNC signal level is high.	R
b31 to b2	—	Reserved	These bits are read as 0.	R

VSYNC flag (VSYNC Signal Status Flag)

The VSYNC flag indicates the state of the VSYNC signal.

HSYNC flag (HSYNC Signal Status Flag)

The HSYNC flag indicates the state of the HSYNC signal.

44.2.5 PDC Receive Data Register (PCDR)

Address(es): PDC.PCDR 4009 4010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The PDC includes a 32-bit-wide, 22-stage FIFO for the storage of captured data. The FIFO is mapped to the 4-byte PCDR register and four bytes of data are read from the PCDR register at a time. The receive data ready flag sets for every 32 bytes of received data, and this also results in a receive data ready interrupt if the DFIE bit in the PCCR0 register is set to 1. When a receive data ready interrupt is generated, read the PCDR register eight times. Figure 44.2 shows a schematic view of the PCDR register.

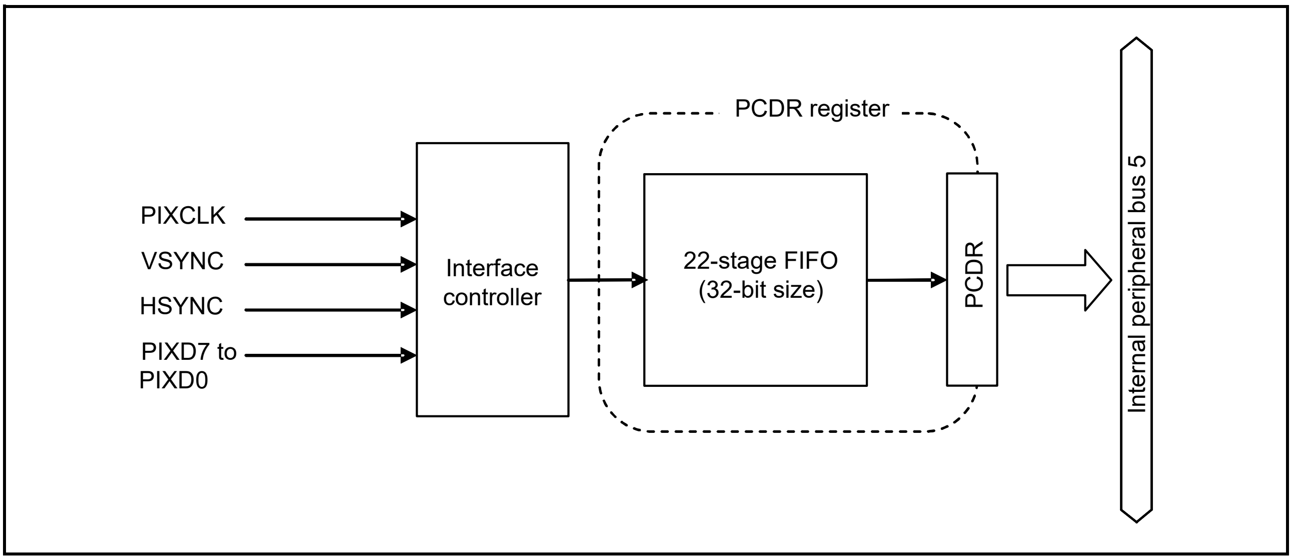


Figure 44.2 Schematic view of PCDR register

For the format of the captured data, either big or little endian can be selected in the EDS bit of the PCCR0 register. Figure 44.3 shows the data arrangements for the endian formats.

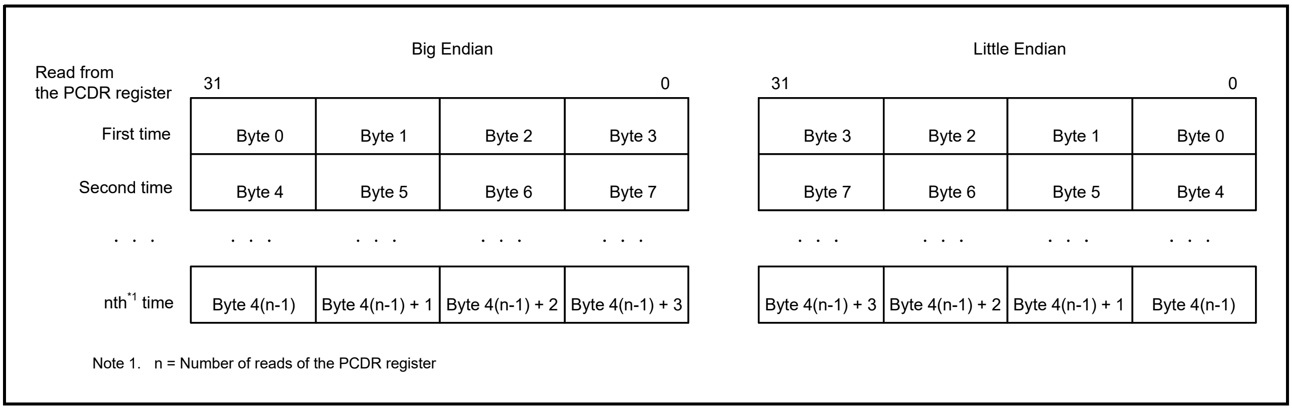


Figure 44.3 Endian formats

44.2.6 Vertical Capture Register (VCR)

Address(es): PDC.VCR 4009 4014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	VSZ[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	VST[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	VST[11:0]	Vertical Capture Start Line Position	These bits specify the number of the line where capture is to start.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	VSZ[11:0]	Vertical Capture Size	These bits specify the number of lines to be captured.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For the relationship between the VCR register setting and the capture range, see section 44.3.3, VCR and HCR Register Settings and the Capture Range. Only set the VCR register while the PCE bit in the PCCR1 register is 0.

VST[11:0] bits (Vertical Capture Start Line Position)

The VST[11:0] bits specify the number of the line where capture is to start. To set the first line, set these bits to 000h; to set the 4095th line, set them to FFEh. The VST[11:0] setting must be within the range from 000h to FFEh and, in combination with the VSZ[11:0] setting, satisfy the following relationship:

Setting range of the VST[11:0] bits: $1 \leq \text{VST[11:0]} + \text{VSZ[11:0]} \leq \text{FFFh}$.

VSZ[11:0] bits (Vertical Capture Size)

The VSZ[11:0] bits specify the number of lines to be captured. To set one line, set these bits to 001h; to set 4095 lines, set them to FFFh. The VSZ[11:0] setting must be within the range from 001h to FFFh and, in combination with the VST[11:0] setting, satisfy the following relationship:

Setting range of the VSZ[11:0] bits: $1 \leq \text{VST[11:0]} + \text{VSZ[11:0]} \leq \text{FFFh}$.

44.2.7 Horizontal Capture Register (HCR)

Address(es): PDC.HCR 4009 4018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	HSZ[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HST[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	HST[11:0]	Horizontal Capture Start Byte Position	These bits specify the horizontal position in bytes where capture is to start.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b16	HSZ[11:0]	Horizontal Capture Size	These bits specify the number of bytes to be captured horizontally.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For the relationship between the HCR register setting and the capture range, see section 44.3.3, VCR and HCR Register Settings and the Capture Range. Only set the HCR register while the PCE bit in the PCCR1 register is 0.

HST[11:0] bits (Horizontal Capture Start Byte Position)

The HST[11:0] bits specify the horizontal position in bytes where capture is to start. To set the first byte, set these bits to 000h; to set the 4092th byte, set them to FFBh. The HST[11:0] setting must be within the range from 000h to FFBh and, in combination with the HSZ[11:0] setting, satisfy the following relationship:

Setting range of the HST[11:0] bits: $1 \leq \text{HST}[11:0] + \text{HSZ}[11:0] \leq \text{FFFh}$.

HSZ[11:0] bits (Horizontal Capture Size)

The HSZ[11:0] bits specify the number of bytes to be captured per line. To set four bytes, set these bits to 004h; to set 4095 bytes, set them to FFFh. The HSZ[11:0] setting must be within the range from 004h to FFFh and, in combination with the HST[11:0] setting, satisfy the following relationship:

Setting range of the HSZ[11:0] bits: $1 \leq \text{HST}[11:0] + \text{HSZ}[11:0] \leq \text{FFFh}$.

44.3 Operation

44.3.1 Transfer Formats

The PDC supports the four transfer formats shown in Figure 44.4 to Figure 44.7. The format is determined by the settings in the VPS and HPS bits in the PCCR0 register.

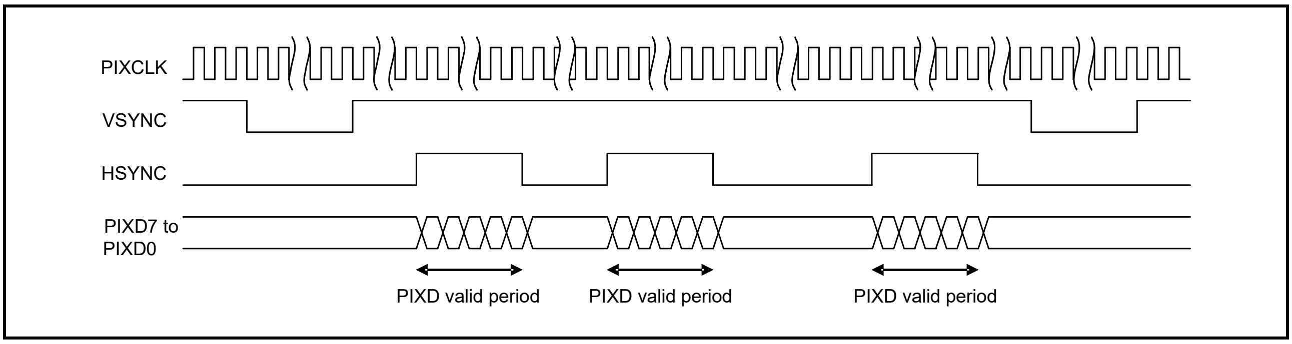


Figure 44.4 PDC transfer format when VPS = 0 and HPS = 0

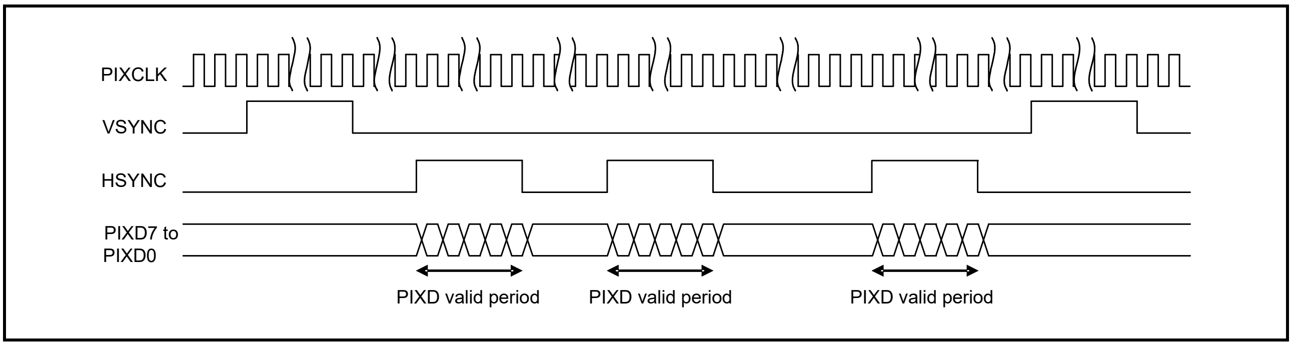


Figure 44.5 PDC transfer format when VPS = 1 and HPS = 0

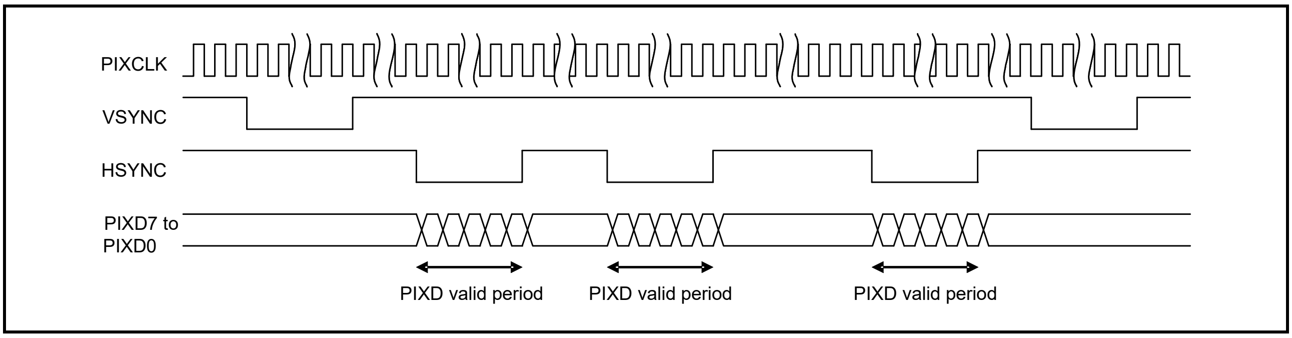


Figure 44.6 PDC transfer format when VPS = 0 and HPS = 1

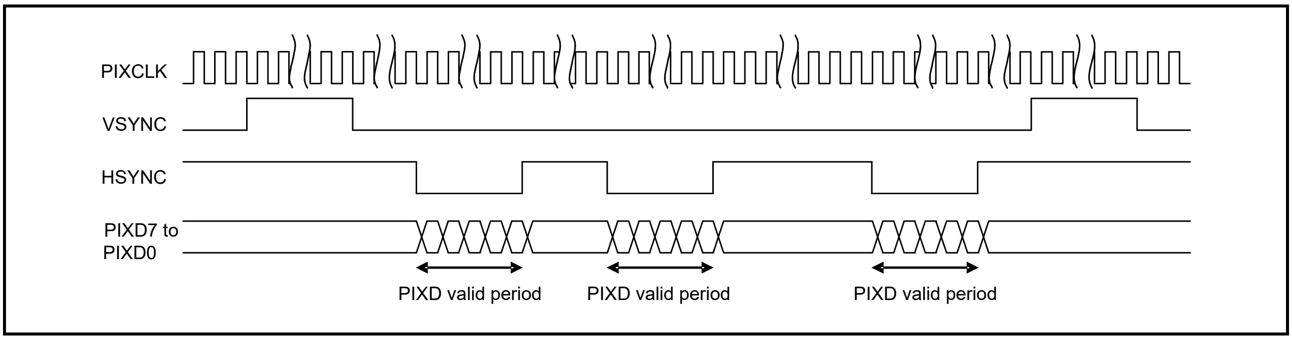


Figure 44.7 PDC transfer format when VPS = 1 and HPS = 1

44.3.2 Transfer Timing

Figure 44.8 and Table 44.3 show the timing of transfers by the PDC.

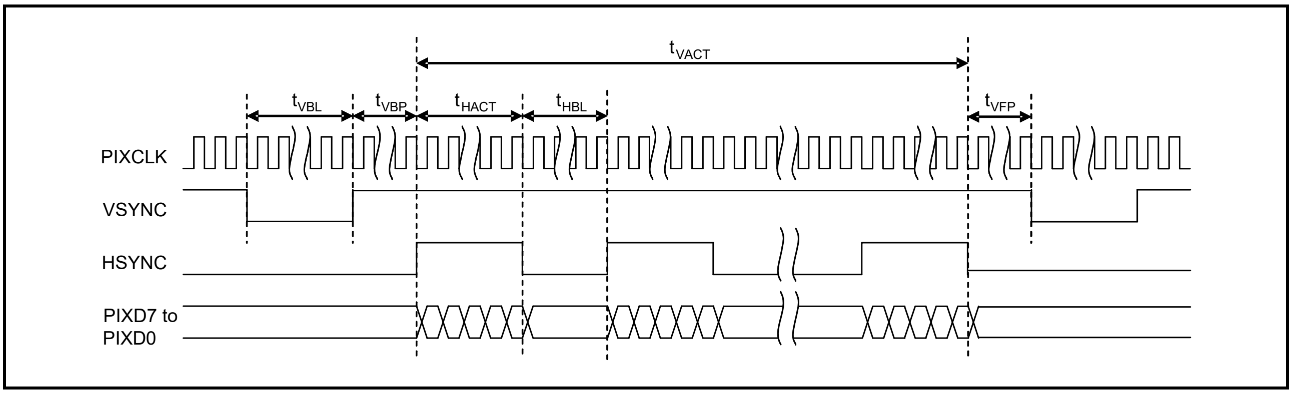


Figure 44.8 PDC transfer timing when VPS = 0 and HPS = 0

Table 44.3 PDC transfer timing when VPS = 0 and HPS = 0

Parameter	Symbol	Min[*1]	Max	Unit
Vertical blanking period	t_{VBL}	128	-	PIXCLK
Vertical backporch	t_{VBP}	10	-	PIXCLK
Horizontal valid period	t_{HACT}	4	4095	PIXCLK
Horizontal blanking period	t_{HBL}	128	-	PIXCLK
Vertical frontporch	t_{VFP}	10	-	PIXCLK
Vertical valid period	t_{VACT}	1	4095	Line

Note 1. The minimum values are the lowest the PDC is capable of achieving. Operation at these values cannot guarantee the avoidance of overruns, vertical line number setting errors, or horizontal byte number setting errors.

44.3.3 VCR and HCR Register Settings and the Capture Range

Figure 44.9 and Figure 44.10 show the relationship between the settings in the VCR and HCR registers and the capture range.

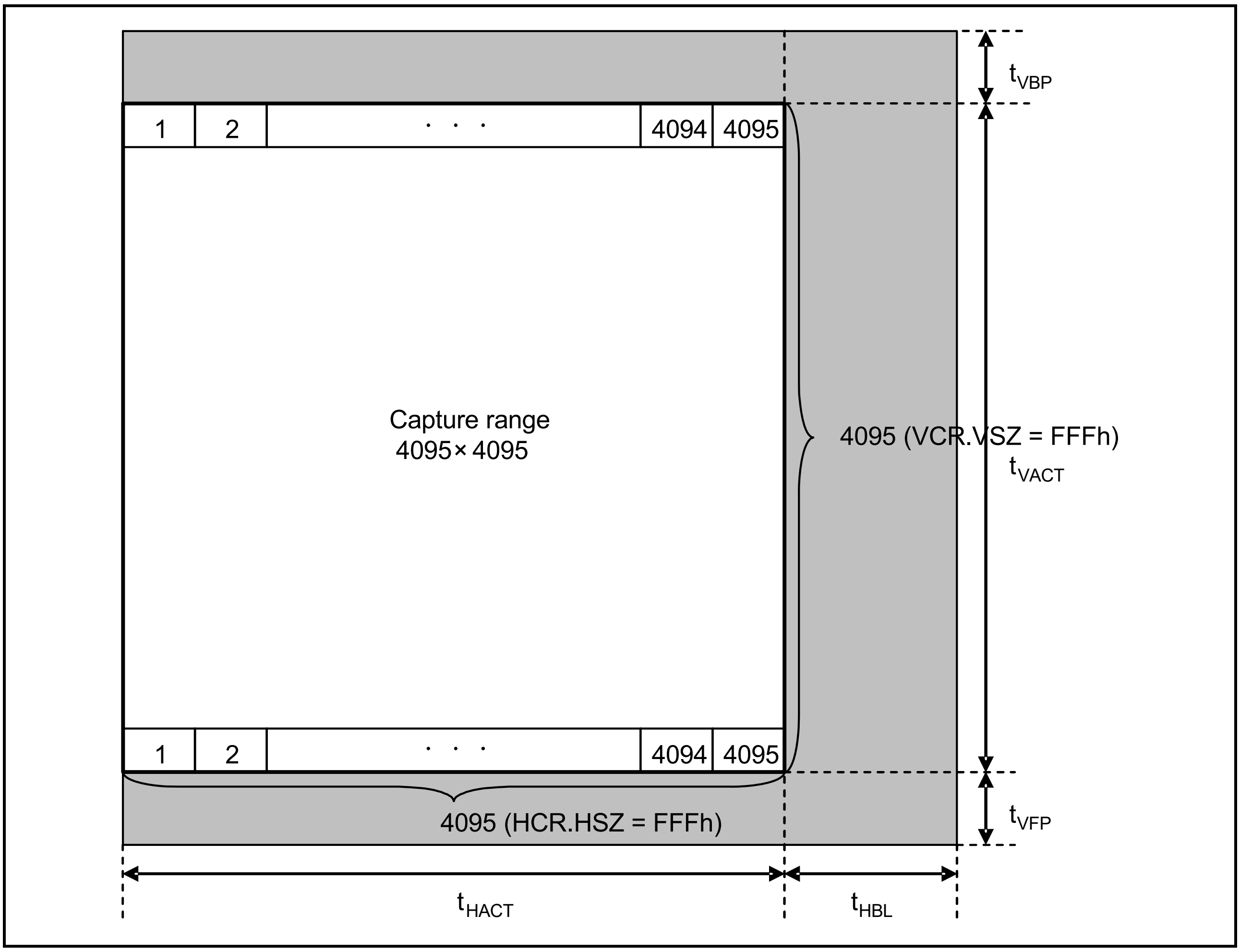


Figure 44.9 Settings in the VCR and HCR registers and the capture range when VCR = 0FFF 0000h and HCR = 0FFF 0000h

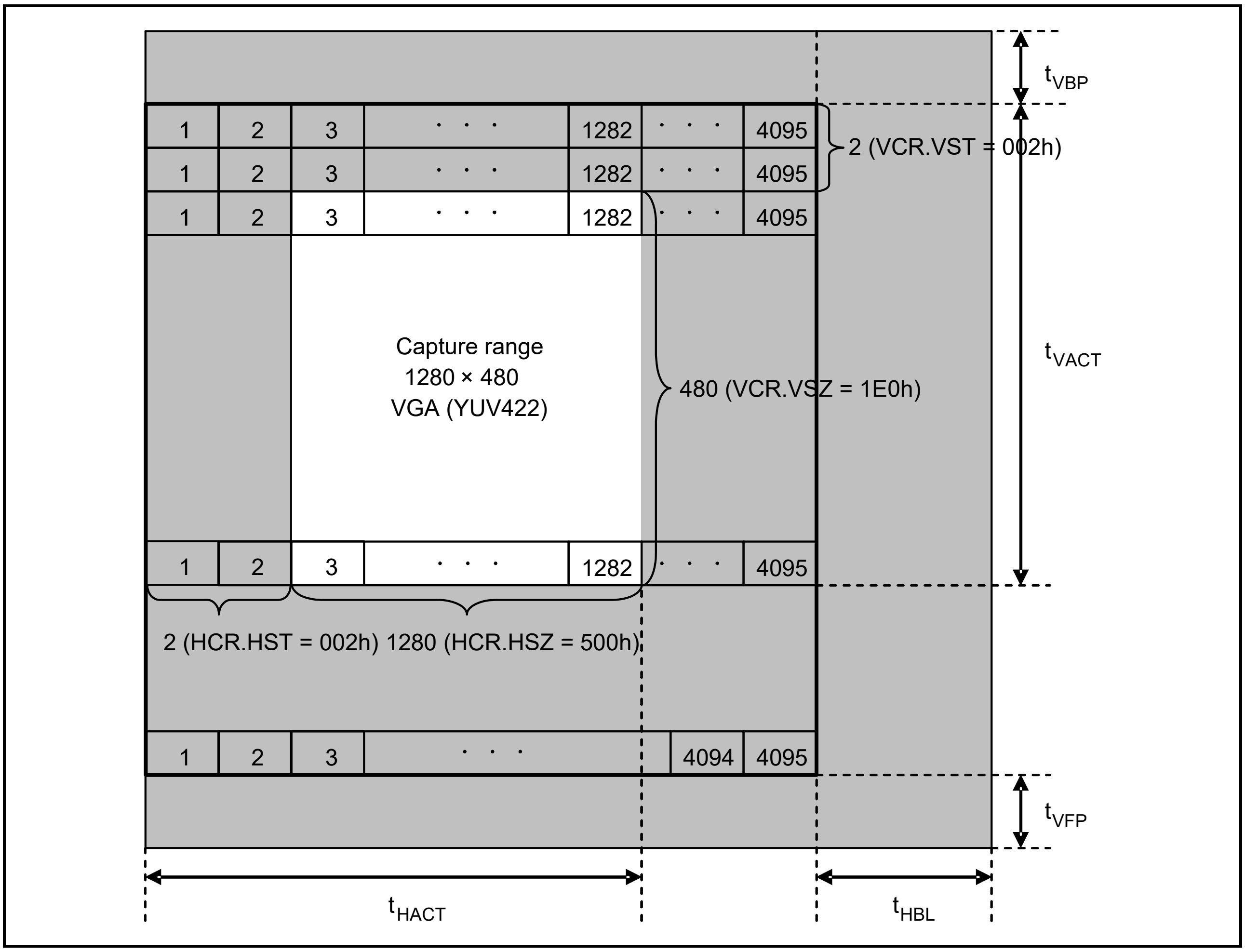


Figure 44.10 Settings in the VCR and HCR registers and the capture range when VCR = 01E0 0002h and HCR = 0500 0002h

44.3.4 Reception Operation

Figure 44.11 shows an example of reception operations when receive data ready interrupts (to start the DTC or DMAC) and frame end interrupts are in use.

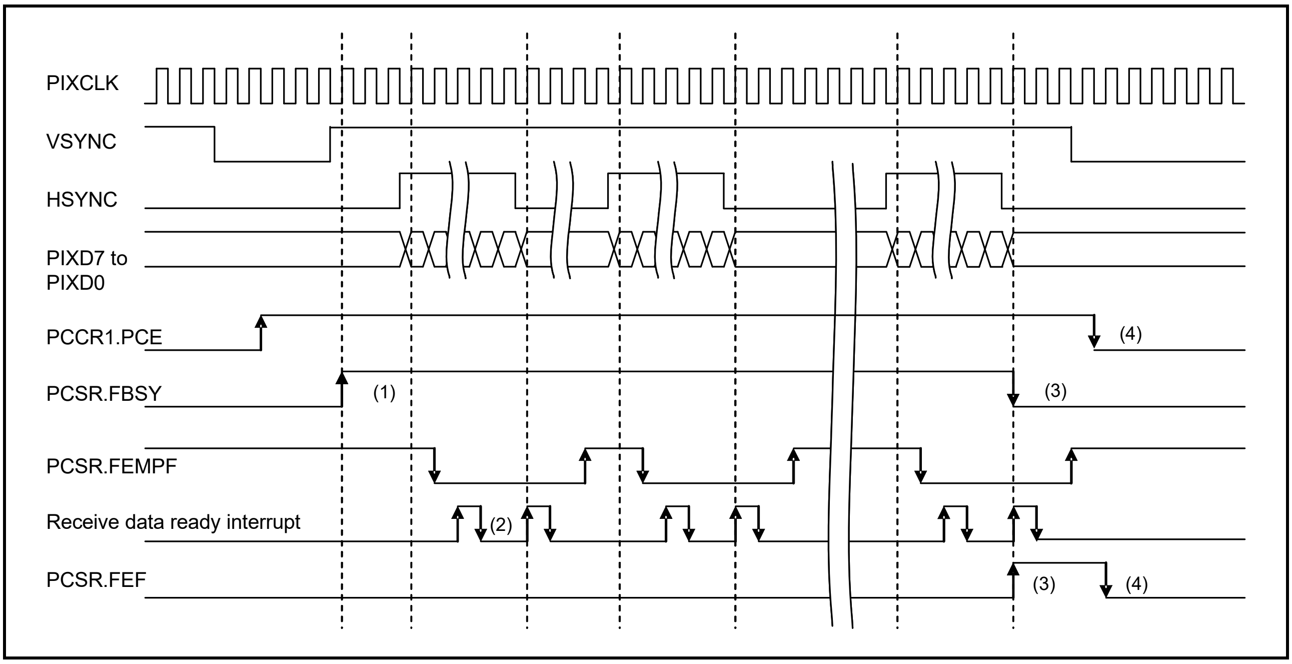


Figure 44.11 Example of reception operations

This section describes the actual operations at the times indicated by (1), (2), (3), and (4) in Figure 44.11.

When a valid edge of the VSYNC signal is detected after the PCE bit in the PCCR1 register is set to 1, the FEMPF flag in the PCSR register sets to 1 and the FIFO is initialized. Concurrently, the FBSY flag in the PCSR register sets to 1 and reception operations start.

When data within the capture range set in the VCR and HCR registers is received, the data is stored in the FIFO. The PDC generates a receive data ready interrupt every time it receives 32 bytes of data, and the interrupt starts transfer of the captured data by the DTC or DMAC to the on-chip SRAM or an external address space. The FIFO is likely to overrun if reading the PCDR register takes more time than reception of the data. Check the OVRF flag in the PCSR register to verify an overrun.

After reception of the last byte of data is complete, the FBSY flag in the PCSR register clears to 0 and the FEF flag in the PCSR register sets to 1 so that a receive data ready interrupt and frame end interrupt are generated.

The FEMPF flag in the PCSR register is polled by the frame end interrupt, after which the program must verify the completion of data transfer by the DTC or DMAC. After the PCE bit in the PCCR1 register is cleared to 0, the FEF flag in the PCSR register also clears to 0, and the reception of one frame of data is complete.

If the FEF flag in the PCSR register sets to 1 before the PCE bit in the PCCR1 register is set to 1, valid edges of the VSYNC signal are not detected and reception operations are not started. Clear the FEF flag in the PCSR register to 0 to start data reception operations.

44.3.5 Operation during Horizontal Blanking Period

If the horizontal blanking period begins but the number of received data bytes has not reached 32 bytes since the previous receive data ready, the count of the received data bytes is retained and carried over to the next valid period. Figure 44.12 shows an example of operation during the horizontal blanking period.

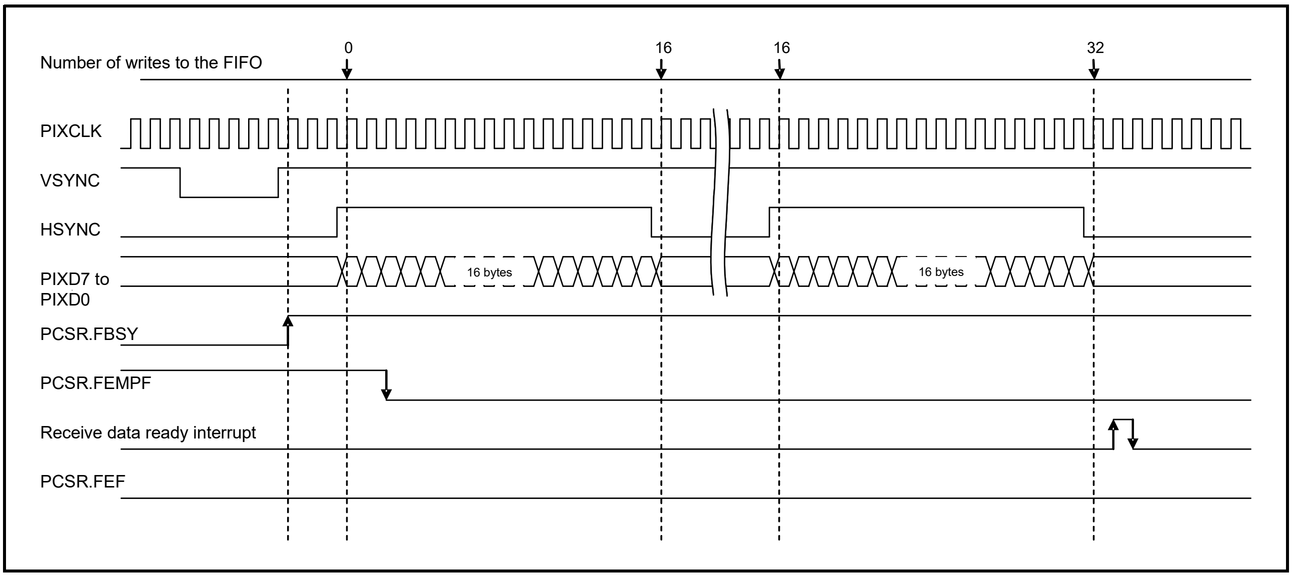


Figure 44.12 Example operation during horizontal blanking period

44.3.6 Continued Reception Operations at Frame End

When the last of the data is received but fewer than 32 bytes of data have been received since the previous receive data ready, the PDC continues to receive data until the number reaches 32, an operation called *continued reception*. When continued reception ends, the PDC generates a receive data ready interrupt and a frame end interrupt. Always input PIXCLK during continued reception. If the data stored in the FIFO is read during this operation, the values are undefined. Figure 44.13 shows an example operation at frame end.

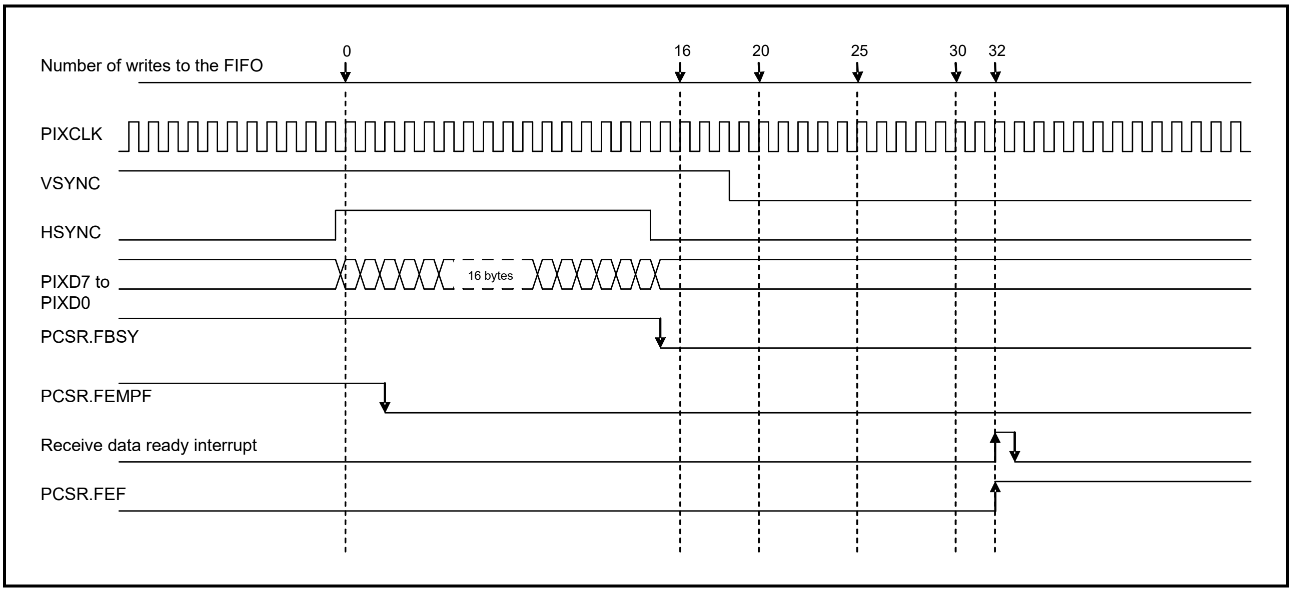


Figure 44.13 Example of continued reception operations at frame end

44.3.7 Error Detection

The PDC provides error detection capabilities, enabling the software to respond to errors during reception operations. Table 44.4 summarizes the conditions for each type of error and the interrupt flags set in response.

Table 44.4 Error detection

Error factor	Conditions for error detection	Interrupt flag	Operation example
Overrun	Receive data arrives while the FIFO is full.*1	PCSR.OVRF	Figure 44.14
Underrun	PCDR register is read while the FIFO is empty.	PCSR.UDRF	Figure 44.15
Vertical line number error	VSYNC signal is negated when the number of captured lines is less than the value set in the VCR register.	PCSR.VERF	Figure 44.16
Horizontal byte number error	HSYNC signal is negated when the number of bytes captured in a line is less than the value set in the HCR register.	PCSR.HERF	Figure 44.17

Note 1. This includes data reception during continued reception operations.

When an error is detected, the PDC sets the associated interrupt flag to 1 to stop reception operations. While the interrupt flag is 1, the PDC does not detect valid edges of the VSYNC signal and does not start reception operations. Clear all error source interrupt flags to 0 to start reception operations.

When an error occurs, data stored in the FIFO is disabled.

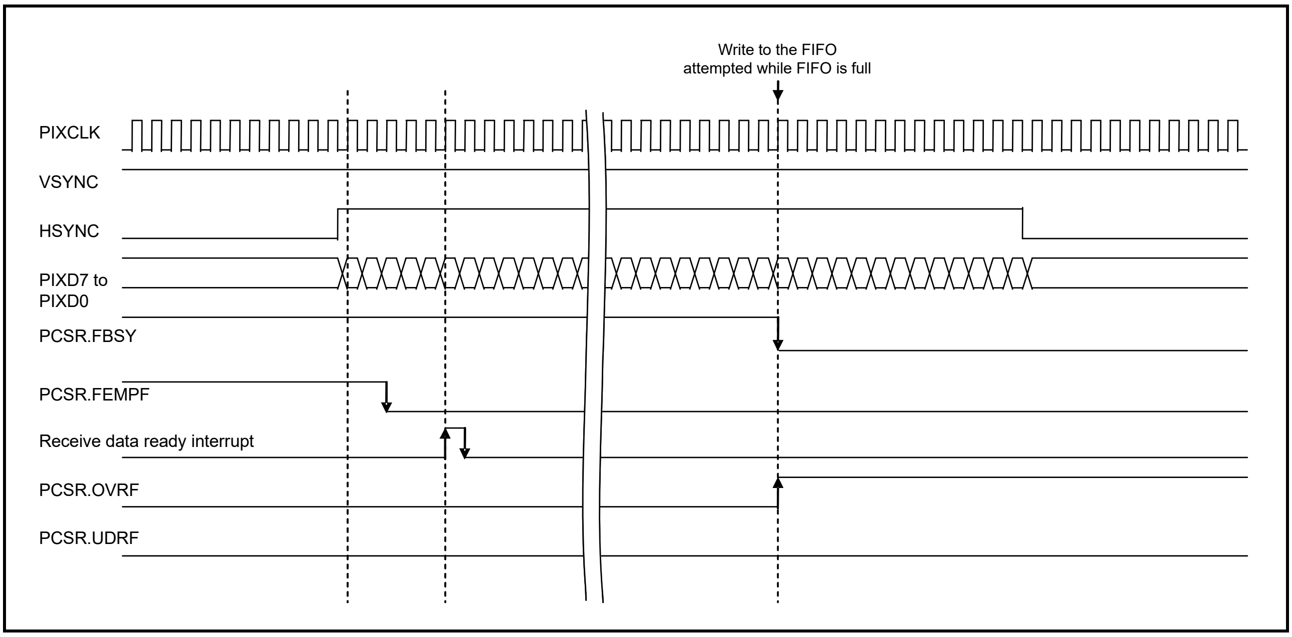


Figure 44.14 Operation when an overrun is detected

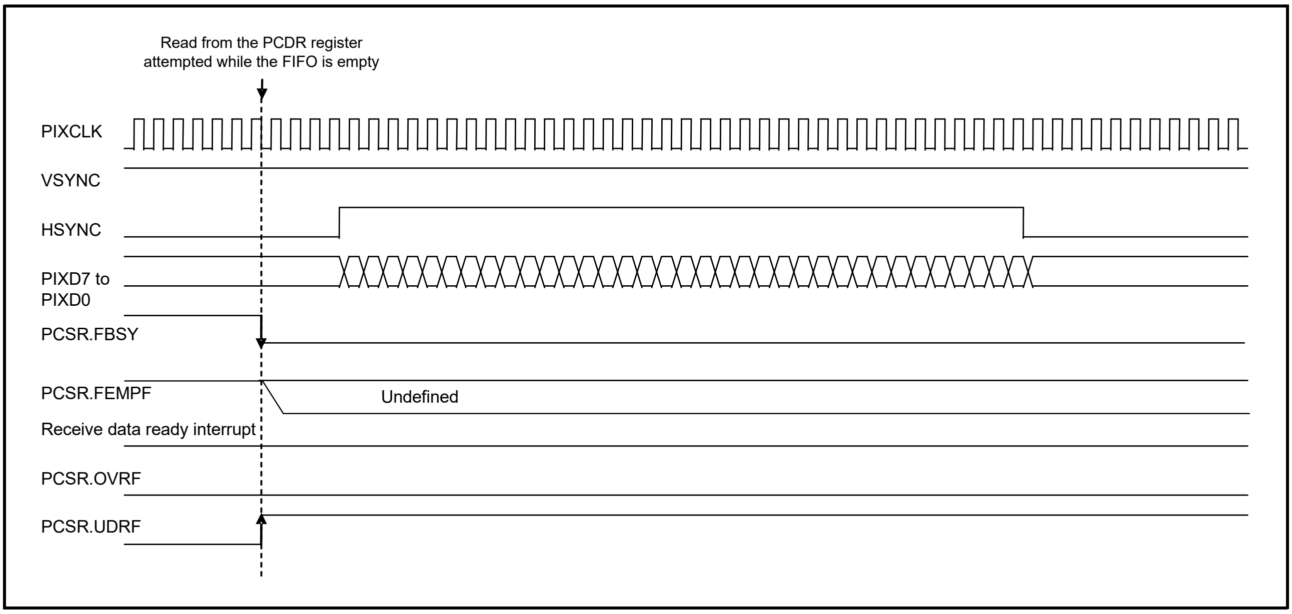


Figure 44.15 Operation when an underrun is detected

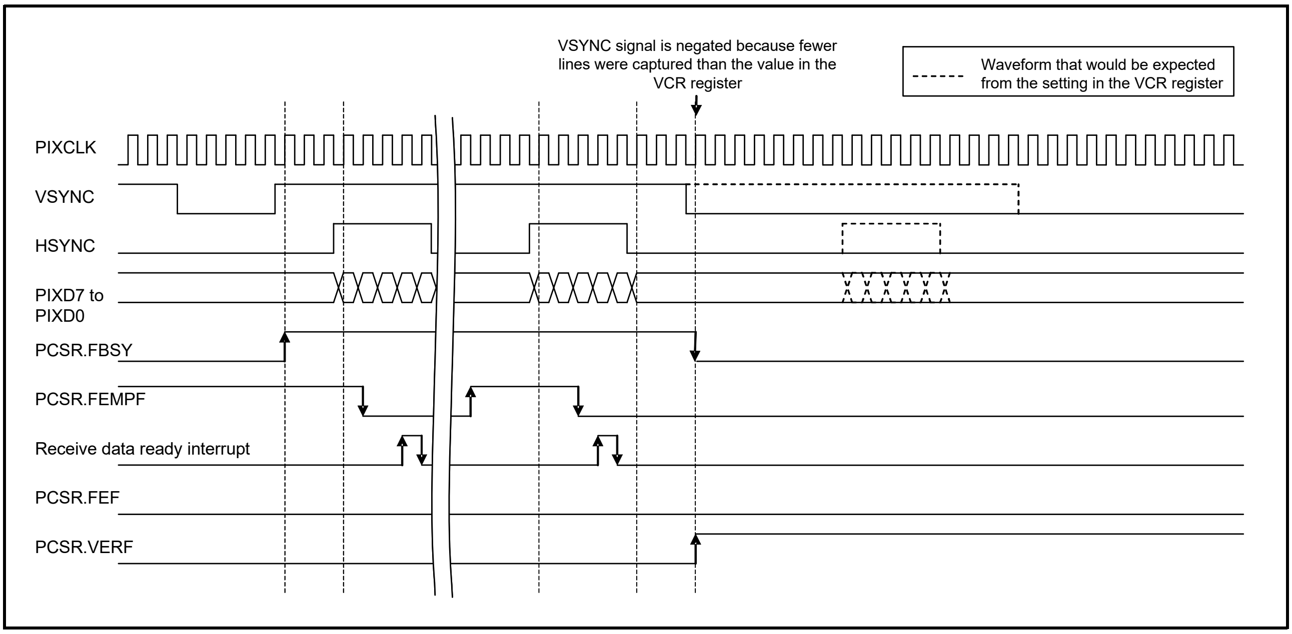


Figure 44.16 Operation when a vertical line number setting error is detected

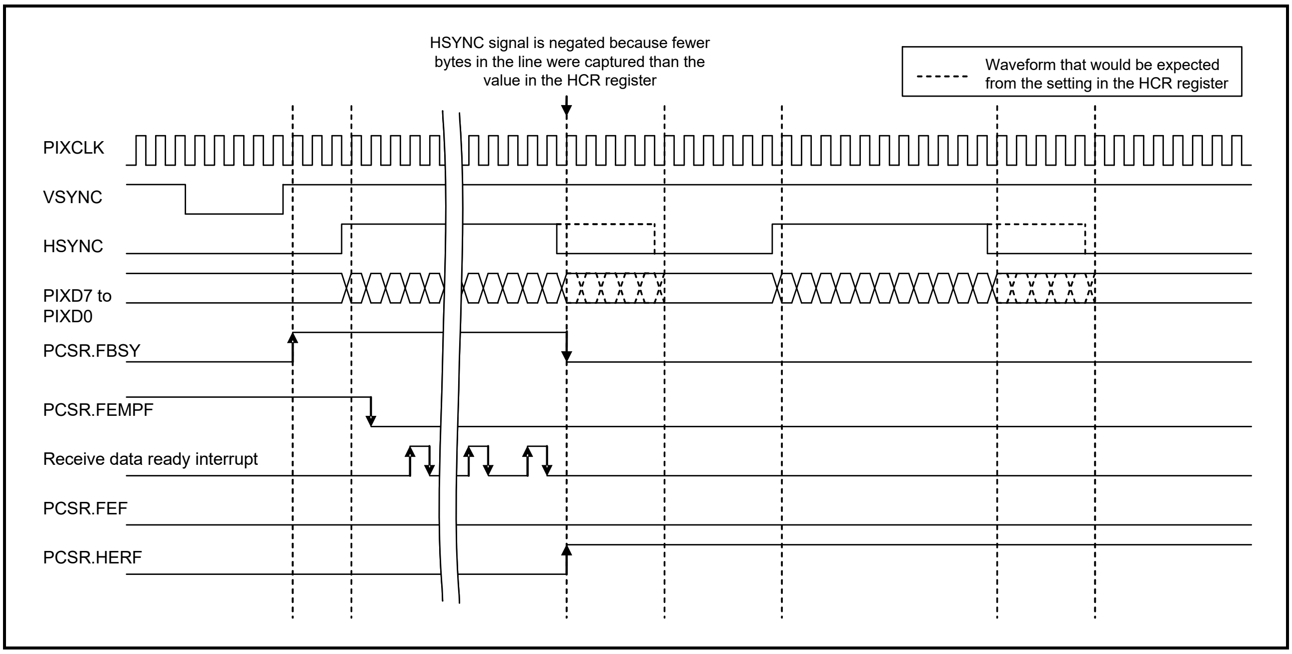


Figure 44.17 Operation when a horizontal byte number setting error is detected

44.3.8 Initial Settings

Figure 44.18 shows an example flow for initial settings. For a description of how to set up the input and output ports and the Interrupt Controller Unit (ICU), see the descriptions given in the sections on the relevant blocks.

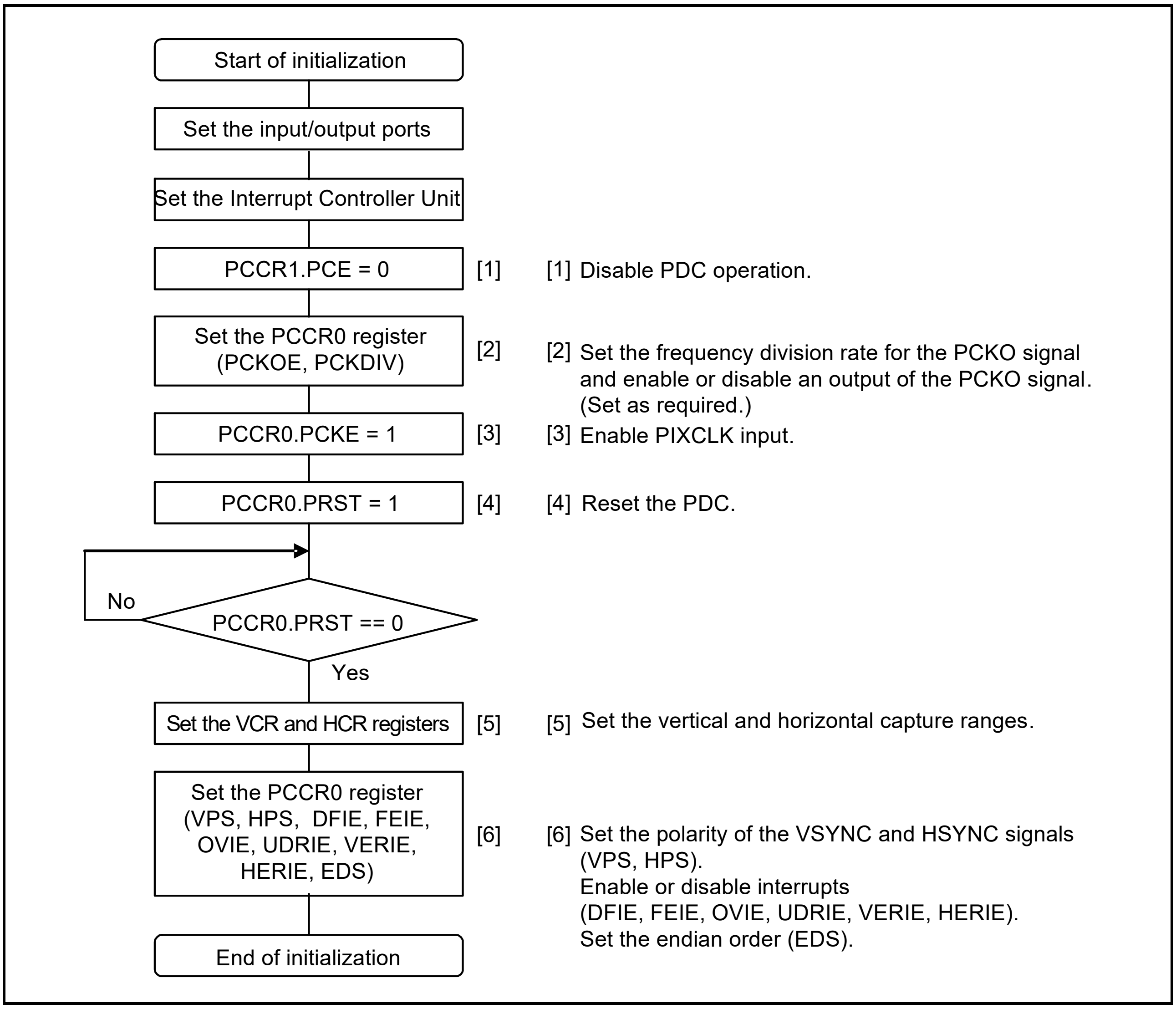
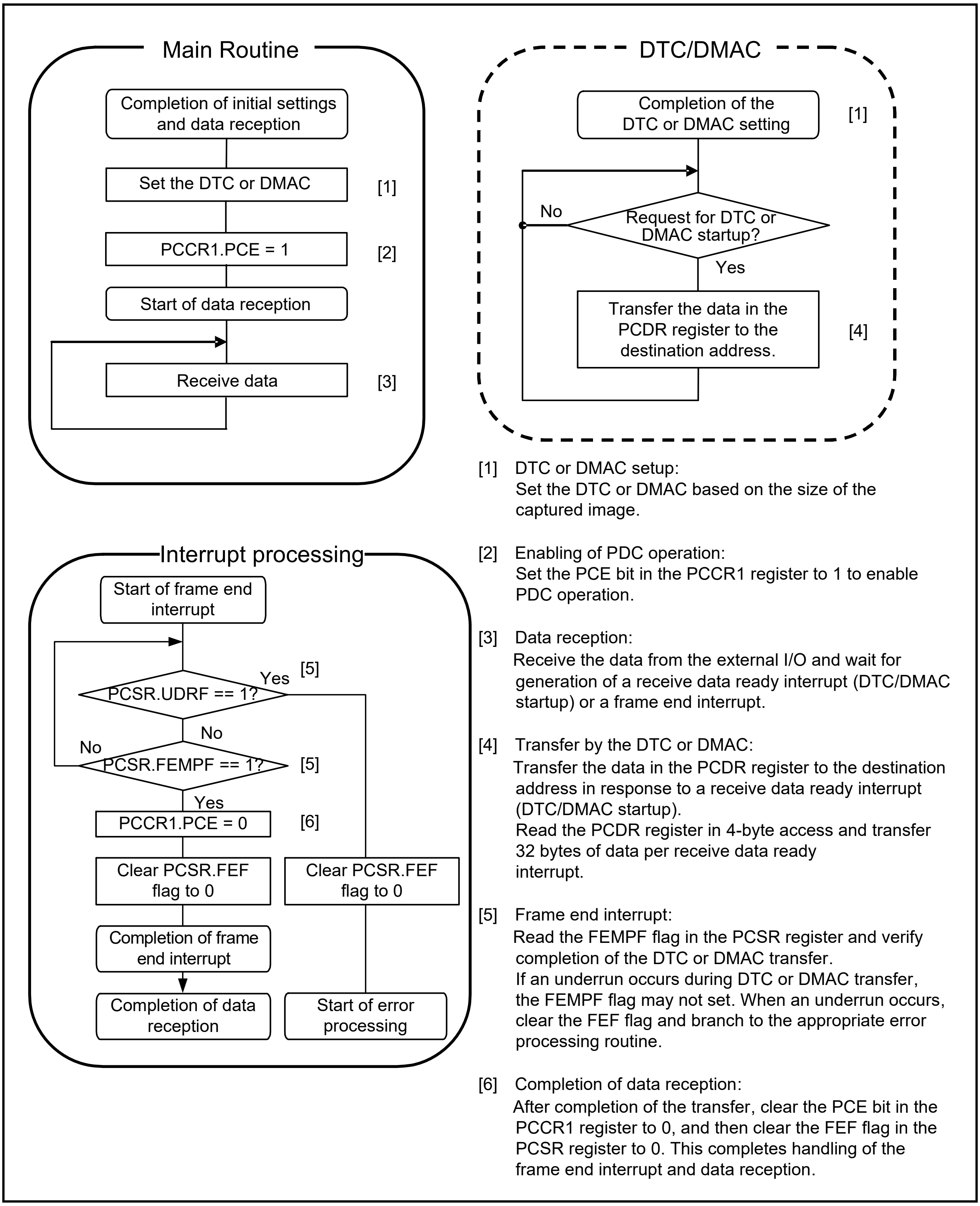


Figure 44.18 Example flow for initial PDC settings

44.3.9 Operation Flows

Figure 44.19 shows an example operation flow when the receive data ready interrupt (to start the DTC or DMAC) and frame end interrupt are in use. For a description of how to set up the DTC or DMAC, see section 17, DMA Controller (DMAC), and section 18, Data Transfer Controller (DTC).



[1] DTC or DMAC setup:
Set the DTC or DMAC based on the size of the captured image.

[2] Enabling of PDC operation:
Set the PCE bit in the PCCR1 register to 1 to enable PDC operation.

[3] Data reception:
Receive the data from the external I/O and wait for generation of a receive data ready interrupt (DTC/DMAC startup) or a frame end interrupt.

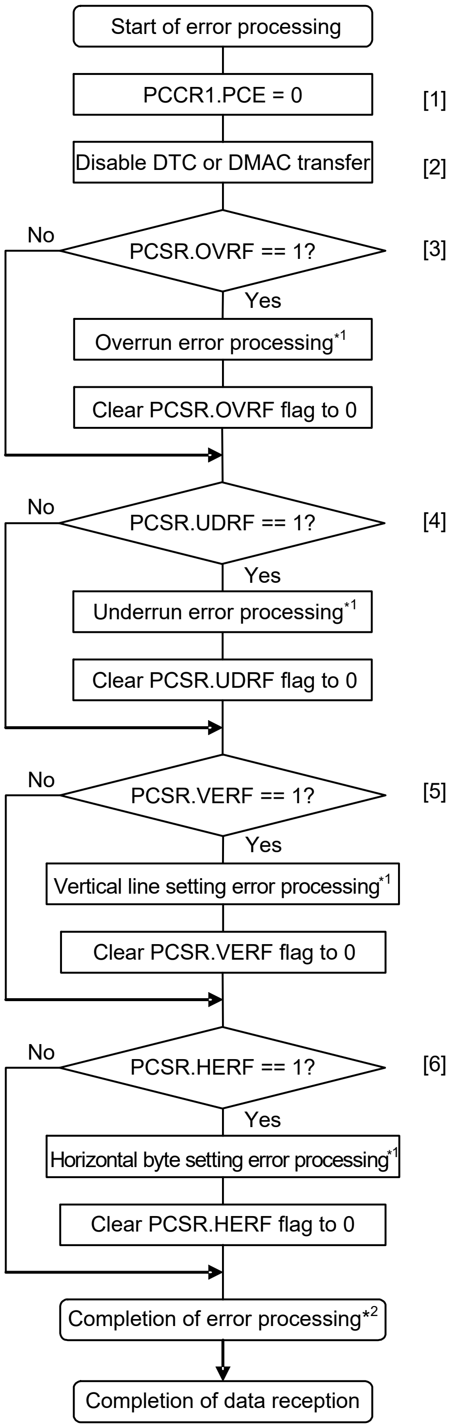
[4] Transfer by the DTC or DMAC:
Transfer the data in the PCDR register to the destination address in response to a receive data ready interrupt (DTC/DMAC startup).
Read the PCDR register in 4-byte access and transfer 32 bytes of data per receive data ready interrupt.

[5] Frame end interrupt:
Read the FEMPF flag in the PCSR register and verify completion of the DTC or DMAC transfer.
If an underrun occurs during DTC or DMAC transfer, the FEMPF flag may not set. When an underrun occurs, clear the FEF flag and branch to the appropriate error processing routine.

[6] Completion of data reception:
After completion of the transfer, clear the PCE bit in the PCCR1 register to 0, and then clear the FEF flag in the PCSR register to 0. This completes handling of the frame end interrupt and data reception.

Figure 44.19 Example operation flow

Figure 44.20 shows an example of the operation flow when responding to an error interrupt.



[1] Disabling of PDC operation:
Clear the PCE bit in the PCCR1 register to 0 to disable PDC operation.

[2] Disabling of the DTC or DMAC transfer:
Disable the transfer of the DTC or DMAC.

[3] Overrun flag check:
Read the PCSR register to check the state of the OVRF flag. If the flag is set, overrun error handling proceeds, and the flag is then cleared.

[4] Underrun flag check:
Read the PCSR register to check the state of The UDRF flag. If the flag is set, underrun error handling proceeds, and the flag is then cleared.

[5] Check of the vertical line setting error flag:
Read the PCSR register to check the state of the VERF flag. If the flag is set, vertical line setting error handling proceeds, and the flag is then cleared.

[6] Check of the horizontal byte setting error flag:
Read the PCSR register to check the status of the HERF flag. If the flag is set, horizontal byte setting error handling proceeds, and the flag is then cleared.

Note 1. Processing for each type of error is defined by the user, so the user needs to prepare error processing for each.

Note 2. If the CPU, DTC, or DMAC reads the PCDR register after an error interrupt occurrs in response to an interrupt source other than an underrun, an underrun may occur.
If an underrun occurs on completion of error processing, run the error processing again and clear the UDRF flag.
Since underruns still require detection after error processing, do not disable the underrun interrupt before the error processing.

Figure 44.20 Example error processing flow

44.3.10 Interrupt Sources

The PDC interrupt sources include:

- Receive data ready
- Frame end
- Overrun

- Underrun
- Vertical line number setting error
- Horizontal byte number setting error.

The PDC can start the DTC or DMAC in response to a receive data ready interrupt request for the transfer of data.

Table 44.5 summarizes the PDC interrupt sources. When an interrupt condition listed in Table 44.5 is satisfied, the associated interrupt is generated. For receive data ready interrupts, the program can clear the interrupt source flag by reading the PCDR register. For the frame end interrupts, clear the FEF flag in the PCSR register. For an overrun, underrun, vertical line number setting error, or a horizontal byte number setting error, the program must check the flags to identify the error source flag, because their interrupt vectors are allocated to the same address by PDC_PCERI. After identifying the source, the program must clear the associated error interrupt source flag (OVRF, UDRF, VERF, or HERF) in the PCSR register.

When the DTC or DMAC module is to handle data transfer, first select the module. After enabling the module for transfers, set up the PDC. For information on setting up the DTC and DMAC, see section 17, DMA Controller (DMAC), and section 18, Data Transfer Controller (DTC).

On completion of output, the request flag clears automatically. An interrupt request signal retained internally can also be cleared by setting the associated interrupt enable bit (the DFIE bit in the PCCR0 register) to 0.

Table 44.5 PDC interrupt sources

Interrupt source	Abbreviation	Interrupt conditions	DTC/DMAC activation
Receive Data Ready	PDC_PCDFI	Receive data ready occurs while the DFIE bit in the PCCR0 register is 1.	Possible
Frame End	PDC_PCFEI	Frame end occurs while the FEIE bit in the PCCR0 register is 1.	Impossible
Errors	PDC_PCERI	• An overrun occurs while the OVIE bit in the PCCR0 register is 1 • An underrun occurs while the UDRIE bit in the PCCR0 register is 1 • A vertical line number setting error occurs while the VERIE bit in the PCCR0 register is 1 • A horizontal byte number setting error occurs while the HERIE bit in the PCCR0 register is 1.	Impossible

44.3.11 Reset State

The PDC has two types of resets: a PDC reset (writing 1 to PCCR0.PRST bit) and other resets.

Other resets include:

- RES pin reset
- Power-on reset
- Voltage monitor reset 0
- Voltage monitor reset 1
- Voltage monitor reset 2
- Deep Software Standby reset
- Independent watchdog timer reset
- Watchdog timer reset
- Software reset
- SRAM parity error reset
- SRAM ECC error reset
- Illegal instruction reset
- Oscillation stop detection reset
- Bus master MPU error reset

- Bus slave MPU error reset
- Stack pointer error reset
- Watchdog timer reset in reset sequence.

Table 44.6 shows the register states following the two types of resets.

Table 44.6 Register states on reset

PDC register	PDC reset	Other resets
PCCR0	Retained	Reset
PCCR1	Retained	Reset
PCSR	Reset	Reset
PCMONR	Retained	Reset
PCDR	Retained	Reset
VCR	Retained	Reset
HCR	Retained	Reset

44.4 Usage Notes

44.4.1 Settings for the Module-Stop Function

PDC operation can be disabled or enabled using the MSTPC2 bit in Module Stop Control Register C (MSTPCRC). The PDC is initially stopped (MSTPC2 =1) after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

44.4.2 Constraints on the Low-Power Function

When reducing PDC power consumption by using the low-power function, set the PCE bit in the PCCR1 register to 0 to disable reception operations, and set the PCKE bit in the PCCR0 register to 0 to disable input through the PIXCLK pin. Use the low-power function after these settings are complete.

If the PCKOE bit in the PCCR0 register is set to 1, set it to 0 to stop output of the PCKO signal, in addition to disabling input through the PIXCLK pin in PCKE. Use the low-power function after these settings are complete.

44.4.3 Constraints on Error Interrupts

When an error interrupt occurs, the DTC or DMAC might still be transmitting parallel data, depending on their operation state. Because of this, the error interrupt processing routine must prohibit data transmission by the DTC or DMAC immediately after prohibiting PDC operation (PCCR1.PCE = 0).

44.4.4 Constraints on Using the DTC

When the DTC is used with the receive data ready interrupt, set the DISEL bit in the MRB register to 0 and the SZ bit in the MRA register to 10b.

The maximum number of blocks the DTC can transfer in block transfer mode is 65,536. If 32 bytes are transferred per block transfer, this represents a total of up to 2,097,152 bytes. If more data is to be transferred, set up the DTC again during the horizontal blanking period. For details, see section 18, Data Transfer Controller (DTC).

44.4.5 Constraints on Using the DMAC

When the DMAC is used with the receive data ready interrupt, set the SZ bit in the DMTMD register to 10b, and configure the DESL[8:0] bits in the DELSRn register (n = 0 to 7) appropriately.

The maximum number of blocks the DMAC can transfer in block transfer mode is 65,536. If 32 bytes are transferred per block transfer, this represents a total of up to 2,097,152 bytes. If more data is to be transferred, set up the DMAC again during the horizontal blanking period. For details, see section 14, Interrupt Controller Unit (ICU) and section 17, DMA Controller (DMAC).

45. Boundary Scan

45.1 Overview

The boundary scan function provides a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std. 1149.1, and IEEE Standard Test Access Port and Boundary Scan Architecture. Table 45.1 lists the boundary scan specifications, Figure 45.1 shows a block diagram, and Table 45.2 lists the I/O pins.

Table 45.1 Boundary scan specifications

Parameter	Specifications
Execution condition	Boundary scan must be executed when the RES pin is driven low.
Test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode

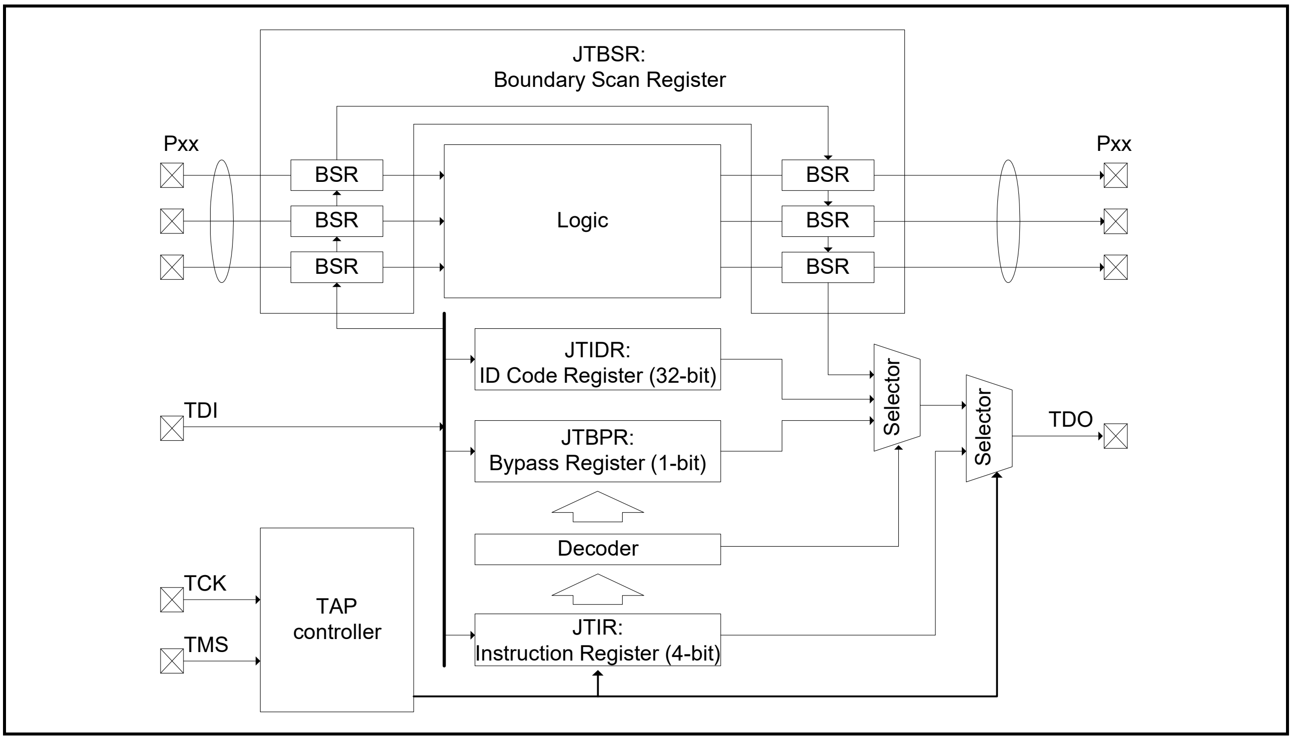


Figure 45.1 Boundary scan function block diagram

Table 45.2 Boundary scan I/O pins

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. The input clock duty cycle is 50% when the boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin

Note: The MCU does not support the TRST pin for the JTAG interface.

45.2 Register Descriptions

Table 45.3 lists the boundary scan registers.

Table 45.3 Boundary scan registers

Register name	Symbol	Value after reset
Instruction Register	JTIR	Eh
ID Code Register	JTIDR	0832 9447h
Bypass Register	JTBPR	Undefined
Boundary Scan Register	JTBSR	Undefined

Usage notes for the boundary scan registers:

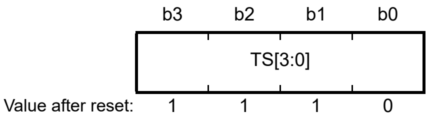
- Instructions can be input to the Instruction Register (JTIR) through the TDI pin by serial transfer.
- The Bypass Register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
- The Boundary Scan Register (JTBSR), which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data is being shifted in.

Table 45.4 shows the availability of serial transfer for the registers.

Table 45.4 Serial transfer for registers

Register name	Serial input	Serial output
Instruction Register (JTIR)	Available	Available
ID Code Register (JTIDR)	Available	Available
Bypass Register (JTBPR)	Available	Available
Boundary Scan Register (JTBSR)	Available	Available

45.2.1 Instruction Register (JTIR)



Bit	Symbol	Bit name	Description	R/W
b3 to b0	TS[3:0]	Test Bit Set	The command configuration for these bits is shown in Table 45.5.	—

Table 45.5 Command configuration

TS3	TS2	TS1	TS0	Instruction
0	0	0	0	EXTEST
0	0	0	1	SAMPLE/PRELOAD
0	0	1	1	IDCODE (Renesas code)
0	1	0	1	CLAMP
0	1	1	0	HIGHZ
1	1	1	1	BYPASS
Other settings				Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin. The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

45.2.2 ID Code Register (JTIDR)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DID[31:0]															
Value after reset:	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DID[31:0]															
Value after reset:	1	0	0	1	0	1	0	0	0	1	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b31 to b0	DID[31:0]	Device ID	These bits store the fixed value that indicates the device IDCODE.	—

JTIDR data is output from the TDO pin when the IDCODE instruction is executed. After a reset release, the IDCODE of JTIDR changes into the Arm® debug code. See the *ARM® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480F).

45.2.3 Bypass Register (JTBPR)

JTBPR is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode. The JTBPR register cannot be read from or written to by the CPU.

45.2.4 Boundary Scan Register (JTBSR)

JTBSR is a shift register for controlling the external input and output pins of the MCU, and is distributed across the pads. To apply the JTBSR register in boundary-scan testing, issue the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions. The BSDL file describes the associations between the JTBSR bits and the pins of the MCU. The value after reset is undefined.

45.3 Operation

During a reset, the JTAG ports, TCK, TMS, TDI, and TDO, are assigned as default pin functions. The TCK, TMS, and TDI pins are pulled up by the pull-up resistors. Boundary scan testing can be executed after the setup time elapses when POR is negated and RES is driven low.

45.3.1 TAP Controller

Figure 45.2 shows the state transition diagram of the TAP controller. All transitions are controlled by the TMS signal.

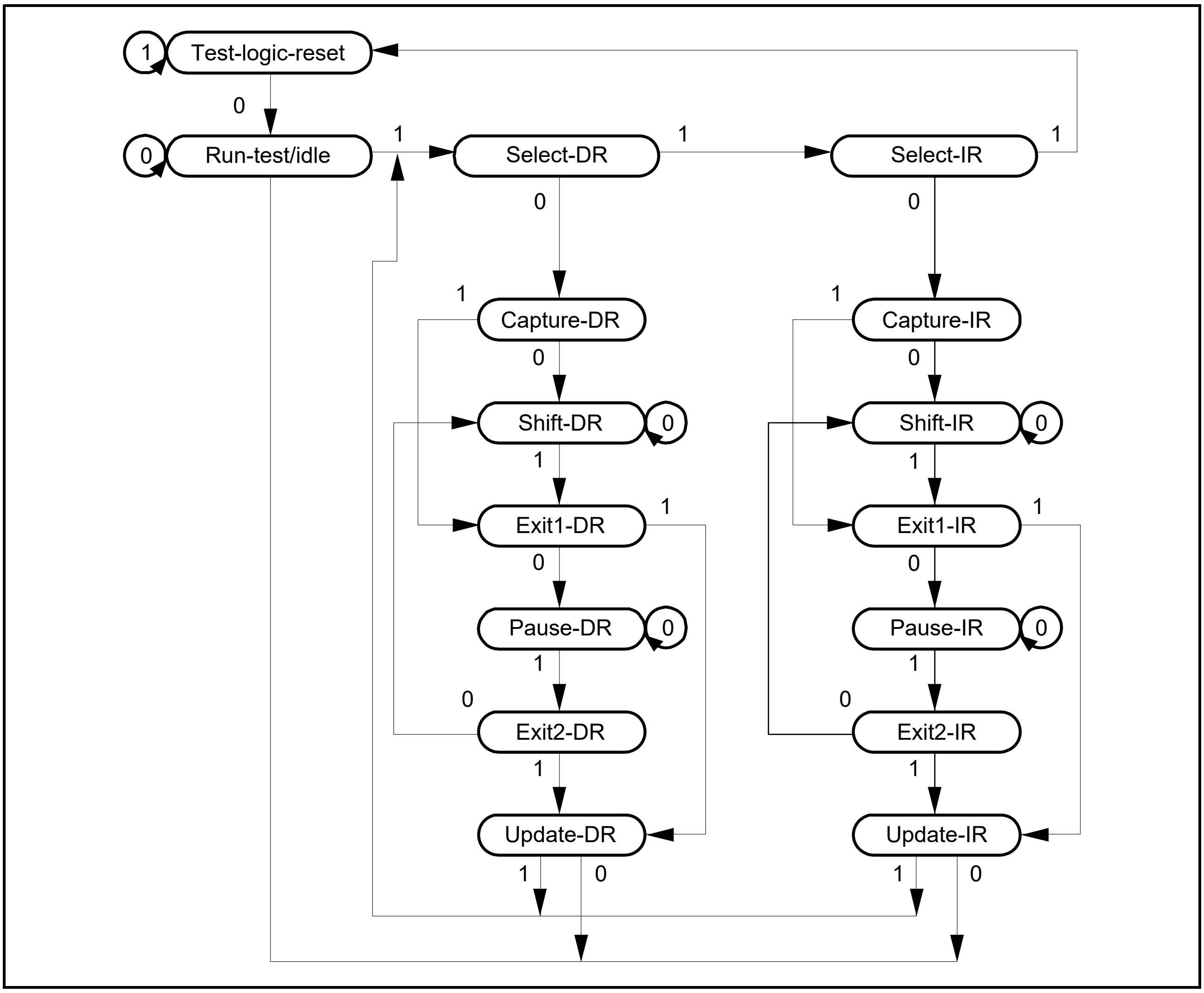


Figure 45.2 State transition diagram of TAP controller

45.3.2 Commands

(1) BYPASS

The BYPASS instruction drives the Bypass Register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The Bypass Register (JTBPR) is connected between the TDI and TDO pins. Bypass operation is initiated from the Shift-DR operation. The TDO is low in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST

The EXTEST instruction is used to test external circuits when the MCU is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified in the SAMPLE/PRELOAD instruction) from the Boundary Scan Register (JTBSR) to the print circuit board, and input pins are used to input the test result.

(3) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input data from the internal circuits of the MCU to the Boundary Scan Register (JTBSR), output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the MCU and output signals are also directly output to the external circuits. The MCU system circuit is not affected by this instruction.

In SAMPLE operation, the Boundary Scan Register (JTBSR) latches a snapshot of the data transferred from the input pins to the internal circuit or data transferred from the internal circuit to the output pins. The latched data is read from the scan path. The JTBSR register latches the data snapshot on the rising edge of the TCK pin in the Capture-DR state. The data snapshot is only transferred from the internal circuit to the output pins during a reset.

In PRELOAD operation, the initial value is written from the scan path to the parallel output latch of the Boundary Scan Register (JTBSR) prior to the EXTEST instruction execution. If EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE

When the IDCODE instruction is selected, the ID Code Register (JTIDR) value is output to the TDO pin in the Shift-DR state of the TAP controller. In this case, the JTIDR register value is output LSB-first. During this instruction execution, the test circuit does not affect the system circuit.

(5) CLAMP

When the CLAMP instruction is selected, output pins output the Boundary Scan Register (JTBSR) value that was specified in the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the JTBSR register is maintained regardless of the TAP controller state.

The Bypass Register (JTBPR) is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

(6) HIGHZ

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of Boundary Scan Register (JTBSR) is maintained regardless of the state of the TAP controller.

The Bypass Register (JTBPR) is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

45.4 Usage Notes

The boundary scan function is subject to the following constraints:

- The boundary scan must be executed when the RES pin is driven low
- Serial data input/output is in LSB order, as shown in Figure 45.3.

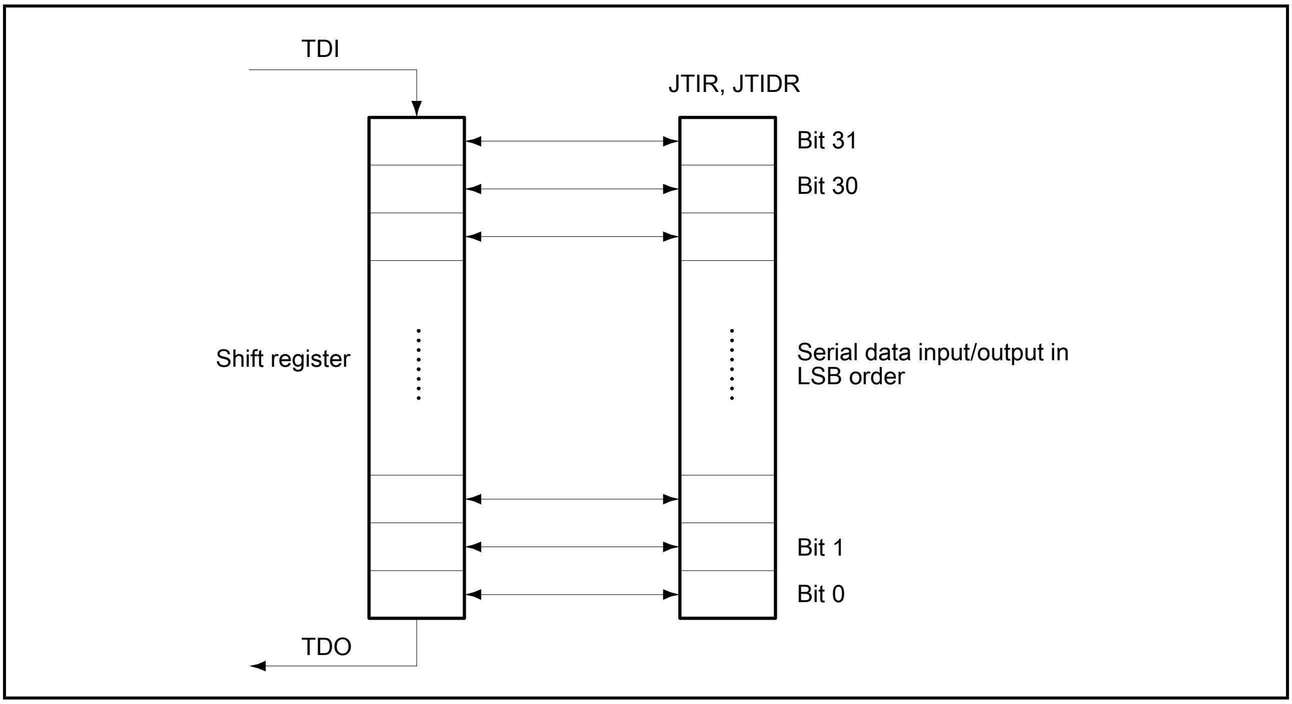


Figure 45.3 Serial data input/output

The following pins cannot be boundary-scanned:

- Power supply pins (VCC, VCL, VCL0, VSS, VBATT, AVCC0, AVSS0, VCC_USB, VSS_USB, AVCC_USBHS, AVSS_USBHS, PVSS_USBHS, VCC_USBHS VSS1_USBHS, and VSS2_USBHS) cannot be boundary-scanned
- Analog reference pins (VREFH0, VREFL0, VREFH, VREFL, USBHS_RREF) cannot be boundary-scanned
- Clock pins (EXTAL, XTAL, XCIN, and XCOUT) cannot be boundary-scanned
- Reset signal (RES) cannot be boundary-scanned
- USB-dedicated pins (USB_DP, USB_DM, USBHS_DP, USBHS_DM) cannot be boundary-scanned
- The boundary-scan pins (TCK, TMS, TDI, and TDO) cannot be boundary-scanned.

46. Secure Cryptographic Engine (SCE7)

The MCU incorporates a Secure Cryptographic Engine (SCE7) module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the Renesas Synergy Software Package (SSP) Crypto library, the SCE7 can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).

The SCE7 module can only be used with the SSP Crypto library. For details, see the Crypto Framework and the SCE Crypto Driver sections in the *Renesas Synergy™ Software Package (SSP) User's Manual*.

46.1 Overview

Table 46.1 shows the SCE7 specifications and Figure 46.1 shows the SCE7 block diagram.

Table 46.1 SCE7 specifications (1 of 2)

Parameter	Description
Access control	Access management circuit • In case of irregular access to the SCE7 due to a falsified program or runaway execution of a program, this circuit blocks all subsequent access and stops the output of data from the SCE7.
Encryption engine	Advanced Encryption Standard (AES): Compliant with NIST FIPS PUB 197 algorithm • Key sizes: 128, 192, or 256 bits • Block size: 128 bits • Chaining modes - ECB, CBC, CTR: Compliant with NIST SP 800-38A - GCM: Compliant with NIST SP 800-38D - XTS: Compliant with NIST SP 800-38E - GCTR. • Throughput for 128-bit data - 11 PCLKB cycles for 128-bit key - 15 PCLKB cycles for 256-bit key*1. AES-GCM • AES-GCM is realized by combining AES-GCTR and GHASH. Triple Data Encryption Standard (3DES): • 192-bit key length • Operates on a fixed 8-byte block of data • Used in legacy Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocols • Throughput for 64-bit data - 16 PCLKB cycles for 56-bit key. Alleged RC4 (ARC4) • 2048-bit key length • Throughput for 128-bit data - 16 PCLKB cycles for 2048-bit key. Key management • Wrapped keys are only valid within the SCE7.
Generation of random numbers	128-bit true random number generator

Table 46.1 SCE7 specifications (2 of 2)

Parameter	Description

Signature generation and verification	RSA
	• Support for 1024-bit and 2048-bit key sizes
	• Signature generation, signature verification, public-key encryption, private-key decryption.
	
	DSA
	• Support for DSA key sizes:
	  • (1024-bit, 160-bit)
	  • (2048-bit, 224-bit)
	  • (2048-bit, 256-bit).
	• Signature generation, signature verification.
	
	ECC
	• Support for curve P-192, P-224, P-256, and P-384
	• Signature generation, signature verification
	• Scalar multiplication.

Message digest computation	HASH • SHA1, SHA224, SHA256, and MD5.
Unique ID	• A unique ID to the MCU, is accessible from the access management circuit through the dedicated bus • Combining the unique ID with the key generation information prevents the illicit copying of data to another MCU.
Privileged mode	• The privileged mode signal is connected to the access management circuit and is used to limit control of the SCE7 module to privileged mode only.
Low power consumption	Setting of the module stop state is possible.

Note 1. This does not include the overhead for calling functions of the SSP Crypto library.

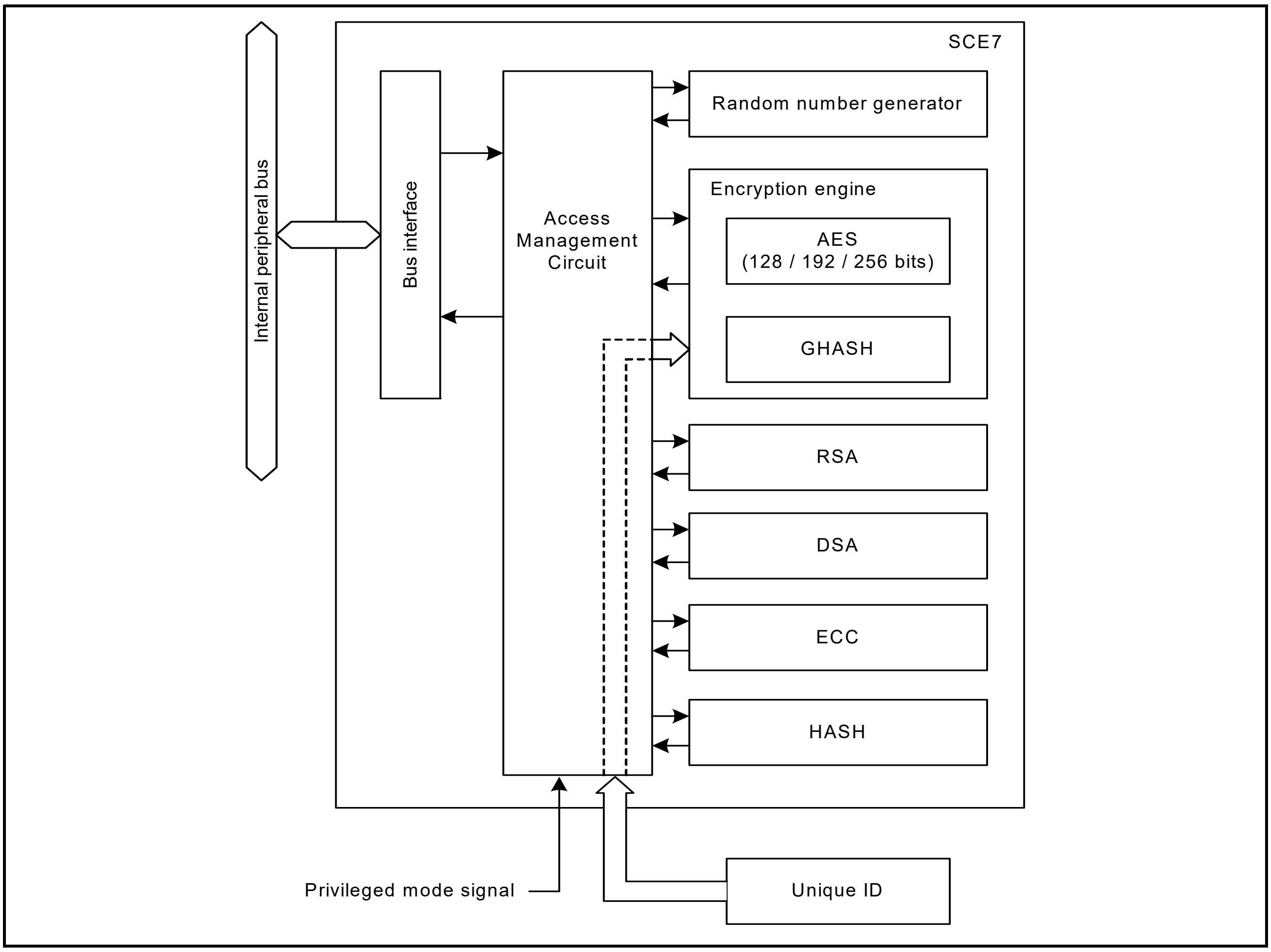


Figure 46.1 SCE7 block diagram

46.2 Operation

46.2.1 Encryption Engine

The encryption engine performs the following operation in hardware, see Figure 46.2:

- Plaintext to ciphertext encryption
- Ciphertext to plaintext decryption.

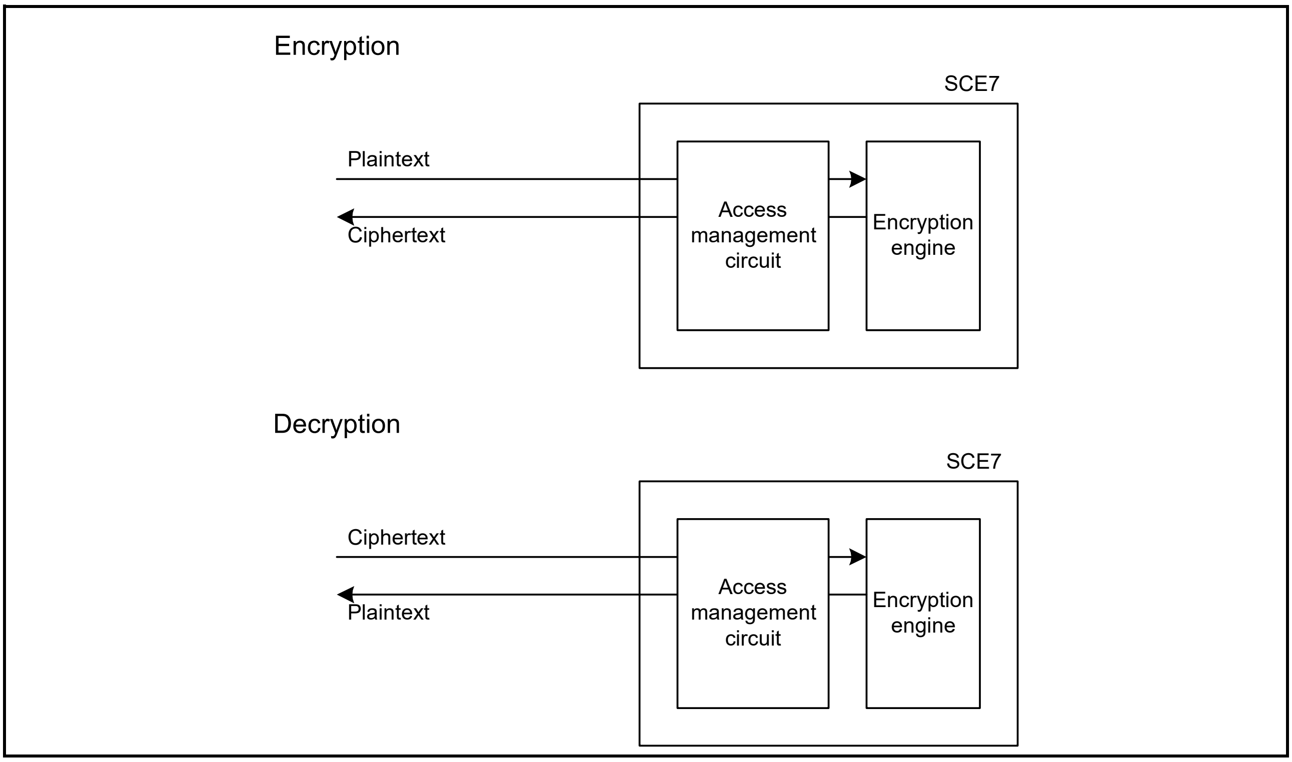


Figure 46.2 Encryption and decryption processes by encryption engine

46.2.2 Encryption and Decryption

To encrypt or decrypt data:

1. Input the data to encrypt or decrypt in the SCE7. The SCE7 converts the plaintext data to ciphertext or ciphertext data to plaintext.
2. Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output. Figure 46.3 shows the encryption engine timing.

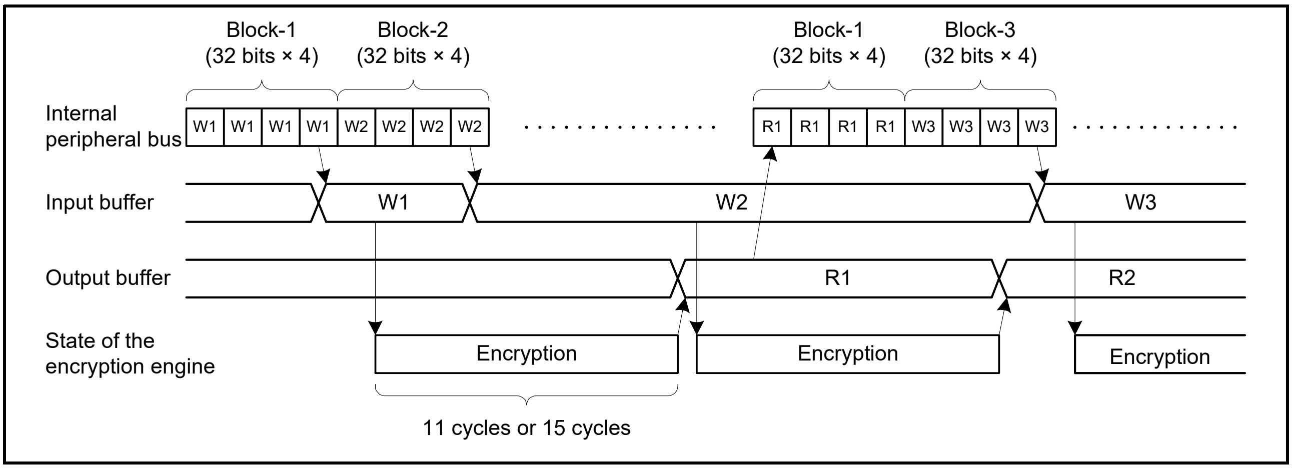


Figure 46.3 Encryption and decryption timing (AES)

46.3 Usage Notes

46.3.1 Software Standby Mode

When Software Standby mode is entered while the encryption engine is in processing, proper processing cannot be resumed after exiting Software Standby mode. Software Standby mode should therefore be entered while the encryption engine is not running.

46.3.2 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable SCE7 operation. The SCE7 module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

47. 12-Bit A/D Converter (ADC12)

47.1 Overview

The MCU provides two 12-bit successive approximation A/D converter (ADC12) units. In unit 0, up to 13 analog input channels are selectable. In unit 1, up to 11 analog input channels, temperature sensor output, and internal reference voltage are selectable for conversion. The A/D conversion accuracy is selectable from 12-, 10-, and 8-bit conversion, making it possible to optimize the trade-off between speed and resolution in generating a digital value.

ADC12 features include:

- 13 channels (unit 0), 11 channels (unit 1)
- PCLKB = 60 MHz (maximum)
- PCLKC = 60 MHz (maximum)
- Analog channels: AN000 to AN007, AN016 to AN020 (unit 0), AN100 to AN103, AN105 to AN107, AN116 to AN119 (unit 1)
- Resolution: 12-bit, 10-bit, 8-bit
- Dedicated sample-and-hold circuit embedded
- Programmable Gain Amplifier embedded.

The ADC12 supports the following operating modes:

- Single scan mode for converting the analog inputs of arbitrarily selected channels in ascending order of channel number
- Continuous scan mode for sequentially converting analog inputs of arbitrarily selected channels continuously in ascending order of channel number
- Group scan mode for arbitrarily dividing analog inputs of channels into two groups (A and B) and converting the analog input of the selected channel for each group in ascending order of channel number.

In group scan mode, you can start Group A and Group B A/D conversion at different times by individually selecting their scan start conditions. In addition, when a priority control operation for Group A is set, the ADC12 accepts Group A scan starting during Group B A/D conversion, suspending Group B conversion. This allows you to assign higher priority to A/D conversion start for Group A.

In double-trigger mode, the analog input of an arbitrarily selected channel is converted in single scan mode or group scan mode (Group A), and data converted by the first and second A/D conversion start triggers are stored in different registers, providing duplexing of A/D-converted data.

Self-diagnosis is performed once at the beginning of each scan, and one of the three voltage values generated in the ADC12 is A/D-converted.

The temperature sensor output and the internal reference voltage are selectable at the same time as the analog input of the channel. First A/D conversion is performed for the analog input of the channel, next the temperature sensor output, and then the internal reference voltage.

The ADC12 provides a compare function (Window A and Window B). This compare function specifies the upper reference value for Window A and lower reference value for Window B, and outputs an interrupt request when the A/D-converted value of the selected channel meets the comparison conditions.

Table 47.1 lists the ADC12 specifications and Table 47.2 list the functions. Figure 47.1 shows a block diagram of ADC12 unit 0 and Figure 47.2 shows a block diagram of ADC12 unit 1. Table 47.3 lists the I/O pins.

Table 47.1 ADC12 specifications (1 of 3)

Parameter	Specifications
Number of units	Two units, 0 and 1
Input channels	• Unit 0: Up to 13 channels • Unit 1: Up to 11 channels

Table 47.1 ADC12 specifications (2 of 3)

Parameter	Specifications
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits, selectable to 12-bit, 10-bit, or 8-bit conversion
Conversion time	0.4 µs/channel, when A/D conversion clock PCLKC (ADCLK) is operating at 60 MHz (See Table 60.40 and Table 60.41 about the condition)
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock PCLKC (ADCLK)*1 can be set with the following division ratios: PCLKB to PCLKC (ADCLK) frequency ratios = 1:1, 2:1, 4:1, 8:1, 1:2, 1:4
Data registers	• 24 registers for analog input (13 for unit 0, 11 for unit 1), one for A/D-converted data duplication in double-trigger mode in each unit, and 2 for A/D-converted data duplication in extended operation in double-trigger mode in each unit • One register for temperature sensor output • One register for internal reference voltage • One register for self-diagnosis • Storing of A/D conversion results in A/D data registers • 8-, 10-, and 12-bit accuracy output for A/D conversion results • A/D-converted value addition mode, in which the sum of all A/D conversion results are stored in the in the A/D data registers as the conversion accuracy bit count + 2 bits.*4 • Double-trigger mode (selectable in single scan and group scan modes): The first unit of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second unit is stored in the duplexing register. • Extended operation in double-trigger mode (available for specific triggers): A/D-converted analog-input data on one selected channel is stored in the duplexing register provided for the associated trigger
Operating modes	• Single scan mode: - A/D conversion is performed only once on the analog inputs of arbitrarily selected channels, on the temperature sensor output, and on the internal reference voltage. • Continuous scan mode: - A/D conversion is performed repeatedly on the analog inputs of arbitrarily selected channels, on the temperature sensor output, and on the internal reference voltage. • Group scan mode: - Analog inputs of arbitrarily selected channels, the temperature sensor output, and the internal reference voltage are divided into Group A and Group B, and A/D conversion of the analog input selected on a group basis is performed only once. - The scan start conditions can be independently selected for Group A and Group B, allowing A/D conversion of Group A and Group B to be started independently. • Group scan mode (when Group A is given priority): - If a Group A trigger is input during A/D conversion on Group B, the A/D conversion on Group B stops and A/D conversion is processed on Group A. - Restart (rescan) of Group B conversion after completion of Group A conversion can be set.
Conditions for A/D conversion start	• Software trigger • Synchronous triggers from the Event Link Controller (ELC). • Asynchronous triggering by the external trigger pins, ADTRG0 (unit 0) and ADTRG1 (unit 1)
Functions	• Dedicated sample-and-hold function with optional constant sampling and 3 channels in units 0 and 1 • Variable sampling state count • Self-diagnosis of ADC12 • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (discharge and precharge functions) • Double-trigger mode (duplication of A/D conversion data) • Switching function for 8-, 10-, and 12-bit conversion*2 • Automatic clear function for A/D data registers • Digital comparison of values in the comparison and data registers, and between values in the data registers
Programmable gain amplifier	• Amplification of analog input signals to enable A/D conversion, with 3 channels in units 0 and 1 • Compatible with single-ended input and differential input

Table 47.1 ADC12 specifications (3 of 3)

Parameter	Specifications
Interrupt sources and ELC events	• ADC12i_ADI: A/D scan end interrupt • ADC12i_GBADI: A/D scan end interrupt for Group B • ADC12i_CMPAI: Window A compare match • ADC12i_CMPBI: Window B compare match • ADC12i_WCMPM: compare match • ADC12i_WCMPUM: compare mismatch
ELC interface	Scan can be started by a trigger from the ELC
Bus interface	Bus clock synchronized with peripheral clock (PCLKB), maximum frequency = 60 MHz
Reference voltage	• Unit 0: VREFH0 is the high potential reference voltage. VREFL0 is the low potential reference voltage. • Unit 1: VREFH is the high potential reference voltage. VREFL is the low potential reference voltage.
Module-stop function	Module-stop state can be set to reduce power consumption*3

i = 0 for unit 0, and i = 1 for unit 1.

Note 1. Peripheral module clock PCLKB is specified in the SCKDIVCR.PCKB[2:0] bits, and A/D conversion clock ADCLK in the SCKDIVCR.PCKC[2:0] bits in units 0 and 1.

Note 2. Changing the A/D conversion accuracy also changes the A/D conversion time. For details, see section 47.3.6, Analog Input Sampling and Scan Conversion Time.

Note 3. For details, see section 11, Low Power Modes.

Note 4. The number of extended bits for addition varies with the A/D conversion accuracy and the number of addition times. A 2-bit extension is up to 4 times conversion (3 times addition) when the A/D conversion accuracy is 8, 10, or 12 bits. A 4-bit extension is 16 times conversion (15 times addition) when the A/D conversion accuracy is 12 bits.

Table 47.2 ADC12 functions

Parameter			Unit 0 (ADC120)	Unit 1 (ADC121)
Analog input channel			AN000 to AN007, AN016 to AN020 Internal reference voltage Temperature sensor output	AN100 to AN103, AN105 to AN107, AN116 to AN119 Internal reference voltage Temperature sensor output
Conditions for A/D conversion start	Software	Software trigger	Enabled	Enabled
	External trigger	Trigger input pin	ADTRG0	ADTRG1
	Synchronous trigger (trigger from ELC)	ELC trigger	ELC_AD00, ELC_AD01	ELC_AD10, ELC_AD11
Channel-dedicated sample-and-hold function	Target channel		AN000 to AN002	AN100 to AN102
Programmable gain amplifier	Target channel		AN000 to AN002	AN100 to AN102
	Differential input pin		PGAVSS000	PGAVSS100
Interrupt			ADC120_ADI ADC120_GBADI ADC120_CMPAI ADC120_CMPBI	ADC121_ADI ADC121_GBADI ADC121_CMPAI ADC121_CMPBI
Output to ELC			ADC120_ADI ADC120_WCMPM ADC120_WCMPUM	ADC121_ADI ADC121_WCMPM ADC121_WCMPUM
Module-stop function settings*1, *2			MSTPCRD.MSTPD16 bit	MSTPCRD.MSTPD15 bit

Note 1. For details, see section 11, Low Power Modes.

Note 2. Wait for 1 µs or longer to start A/D conversion after release from the module-stop state.

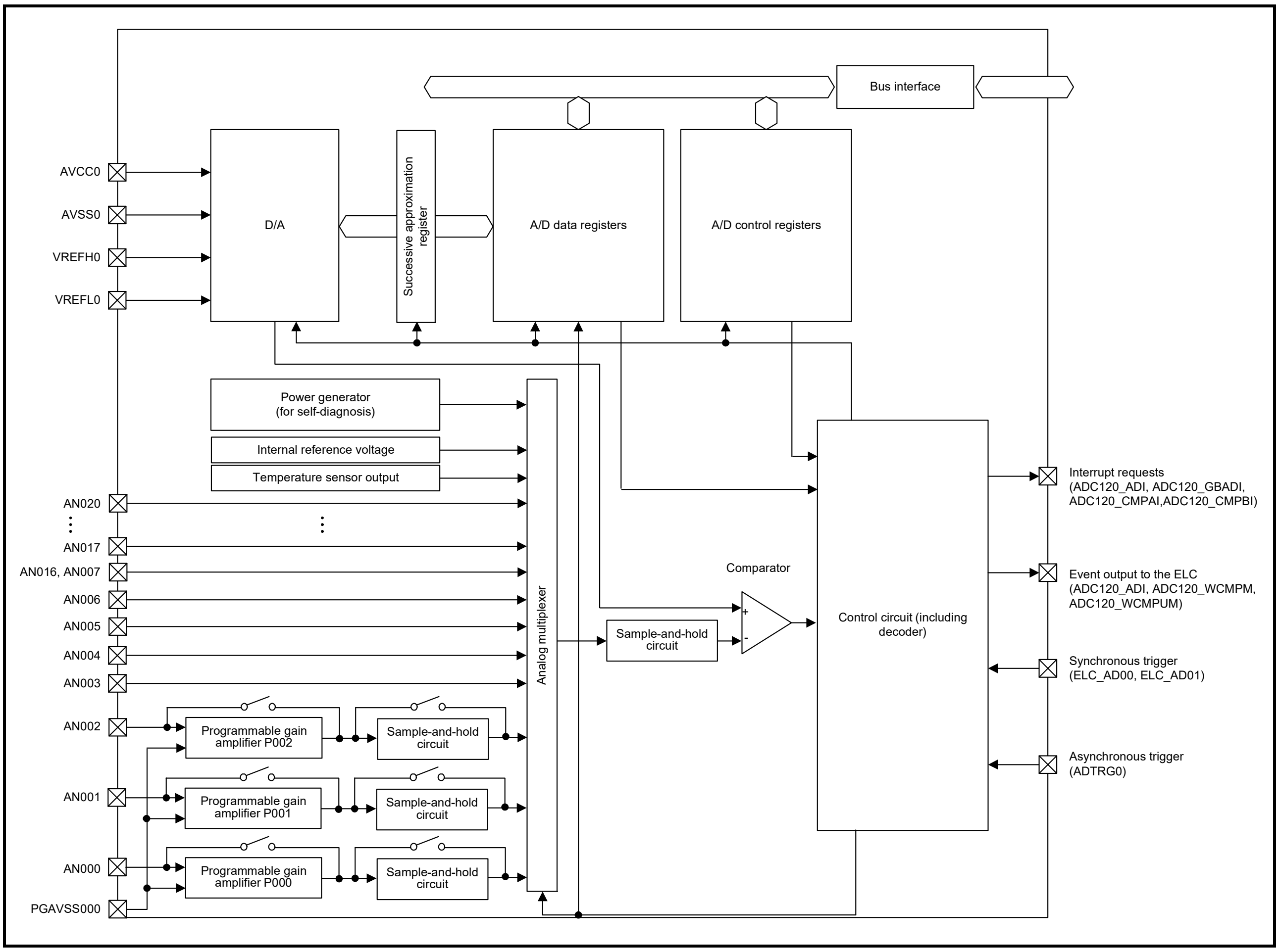


Figure 47.1 ADC12 unit 0 block diagram

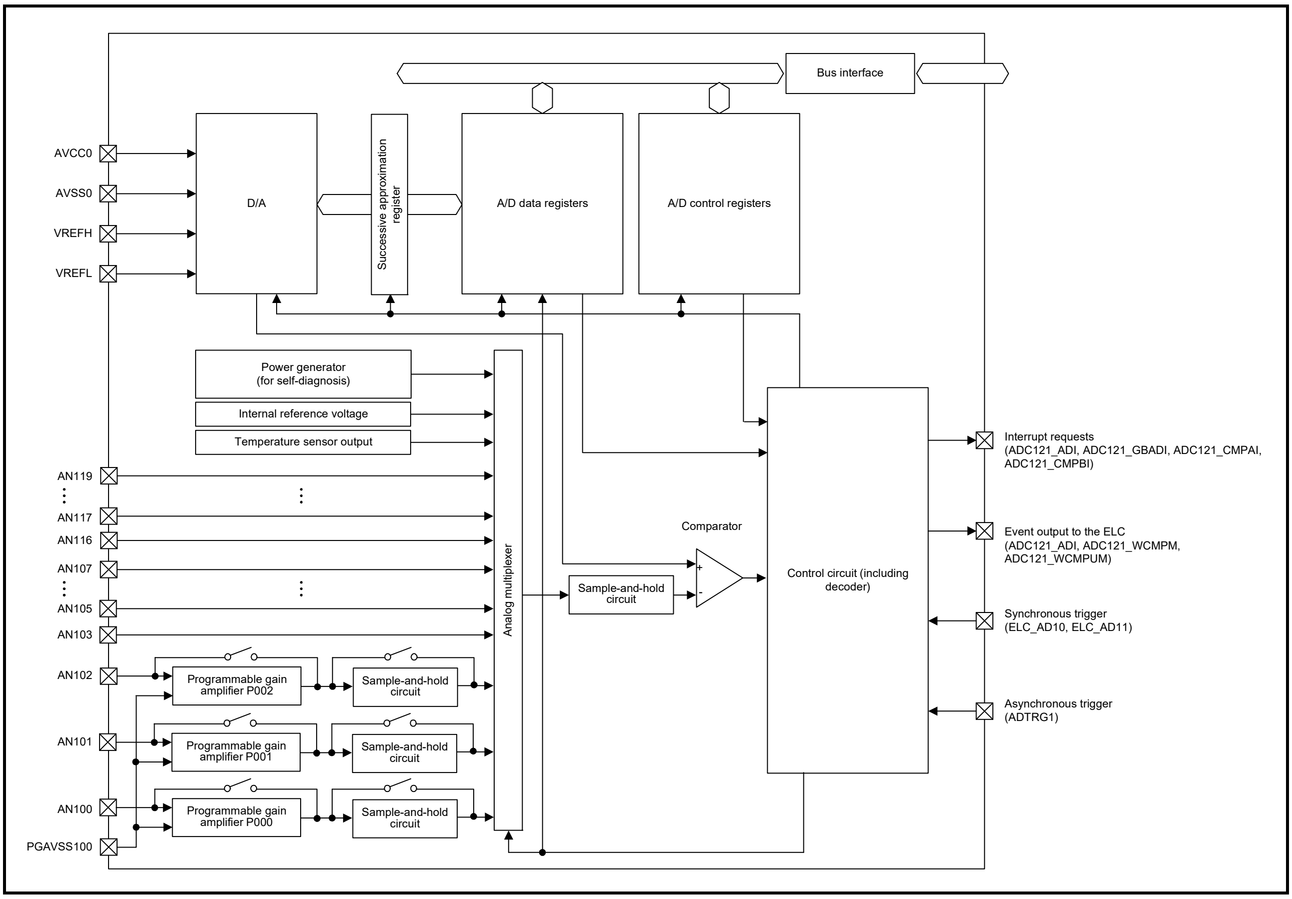


Figure 47.2 ADC12 unit 1 block diagram

Table 47.3 ADC12 I/O pins

Unit	Pin name	I/O	Function
Unit 0	AVCC0	Input	Analog block power supply pin
	AVSS0	Input	Analog block power supply ground pin
	VREFH0	Input	Reference power supply pin
	VREFL0	Input	Reference power supply ground pin
	AN000 to AN007, AN016 to AN020	Input	Analog input pins 0 to 7 and 16 to 20
	ADTRG0	Input	External trigger input pin for starting A/D conversion, active low
	PGAVSS000	Input	Differential input pin
Unit 1	AVCC0	Input	Analog block power supply pin
	AVSS0	Input	Analog block power supply ground pin
	VREFH	Input	Reference power supply pin for ADC12 unit 1 and DAC
	VRELF	Input	Reference power supply ground pin for ADC12 unit 1 and DAC
	AN100 to AN103, AN105 to AN107, AN116 to AN119	Input	Analog input pins 0 to 3, 5 to 7, and 16 to 19
	ADTRG1	Input	External trigger input pin for starting A/D conversion, active low
	PGAVSS100	Input	Differential input pin

47.2 Register Descriptions

47.2.1 A/D Data Registers y (ADDRy), A/D Data Duplexing Register (ADDBLDR), A/D Data Duplexing Register A (ADDBLDRA), A/D Data Duplexing Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)

The data registers include:

- ADDRy registers (y = 0 to 7, 16 to 20 in unit 0 and y = 0 to 3, 5 to 7, 16 to 19 in unit 1): 16-bit read-only registers for storing the A/D conversion results
- ADDBLDR register: 16-bit read-only register for storing the A/D conversion results in response to the second trigger in double-trigger mode
- ADDBLDRA and ADDBLDRB registers: 16-bit read-only registers for storing the A/D conversion results in response to the respective triggers during extended operation in double-trigger mode
- ADTSDR register: 16-bit read-only register for storing the A/D conversion result of the temperature sensor output
- ADOCDR register: 16-bit read-only register for storing the A/D result of the internal reference voltage.

The following conditions determine the formats for data in these registers:

- The setting in the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right setting)
- The setting in the A/D Conversion Accuracy Specify bits (ADCER.ADPRC[1:0]) (8-, 10-, or 12-bit setting).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

The data formats for each condition are as follows:

Settings for flush-right data with 12-bit accuracy

Address(es): ADC120.ADDR0 4005 C020h to ADC120.ADDR7 4005 C02Eh,
ADC120.ADDR16 4005 C040h to ADC120.ADDR20 4005 C048h,
ADC120.ADDBLDR 4005 C018h, ADC120.ADDBLDRA 4005 C084h, ADC120.ADDBLDRB 4005 C086h,
ADC120.ADTSDR 4005 C01Ah, ADC120.ADOCDR 4005 C01Ch,
ADC121.ADDR0 4005 C220h to ADC121.ADDR3 4005 C226h,
ADC121.ADDR5 4005 C22Ah to ADC121.ADDR7 4005 C22Eh,
ADC121.ADDR16 4005 C240h to ADC121.ADDR19 4005 C246h,
ADC121.ADDBLDR 4005 C218h, ADC121.ADDBLDRA 4005 C284h, ADC121.ADDBLDRB 4005 C286h,
ADC121.ADTSDR 4005 C21Ah, ADC121.ADOCDR 4005 C21Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value.	R
b15 to b12	—	Reserved	These bits are read as 0.	R

Settings for flush-right data with 10-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	AD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value.	R
b15 to b10	—	Reserved	These bits are read as 0.	R

Settings for flush-right data with 8-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	AD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value.	R
b15 to b8	—	Reserved	These bits are read as 0.	R

Settings for flush-left data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0.	R
b15 to b4	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value.	R

Settings for flush-left data with 10-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[9:0]										—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0.	R
b15 to b6	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value.	R

Settings for flush-left data with 8-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0.	R
b15 to b8	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value.	R

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in A/D-converted value addition mode. When A/D-converted value average mode is selected, this register indicates the mean of the A/D-converted values on the specified channel. The value is stored in the A/D data register based on the setting in the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 8-, 10-, or 12-bit accuracy (ADPRC bit setting), 1, 2, 3 or 4 times can be selected for A/D-converted value addition. 16 times can also be selected for addition mode, but only with 12-bit accuracy selected. In addition mode, this register indicates the value that is obtained by adding the A/D-converted values on a specific channel. The conversion results sum is retained in the A/D data register as a 2-bit-extended value of the conversion accuracy specified. The value is stored in the A/D data register based on the setting in the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

When converting 1, 2, 3, or 4 times in addition mode with 8-, 10-, or 12-bit accuracy specified, the conversion result is stored in the A/D data register as a 2-bit-extended value of the specified accuracy.

When converting 16 times in addition mode with 12-bit accuracy specified the conversion result is stored in the A/D data register as a 4-bit-extended value of the specified accuracy.

The data formats for each condition are as follows:

Settings for flush-right data with 12-bit accuracy in A/D-converted value addition mode

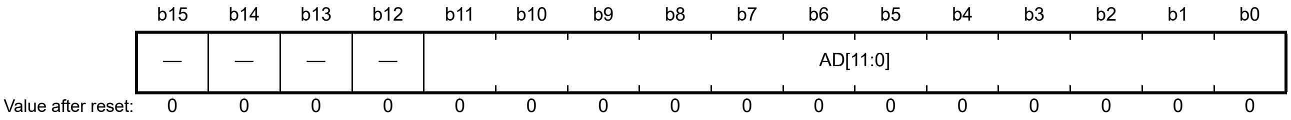
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[15:0]*1															
	—	—	AD[13:0]*2													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	AD[15:0]*1	Added Value 15 to 0	16 -bit sum of A/D conversion results.	R

Bit	Symbol	Bit name	Description	R/W
b13 to b0	AD[13:0]*2	Added Value 13 to 0	14-bit sum of A/D conversion results.	R
b15, b14	—	Reserved	These bits are read as 0.	R

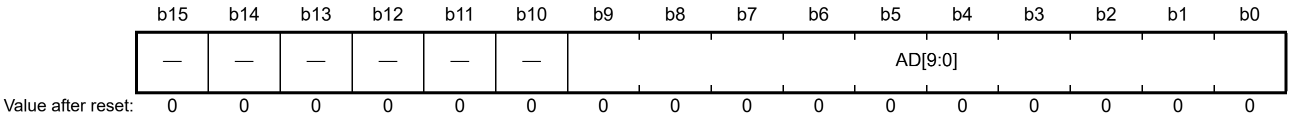
Note 1. Used when 16 conversion times is specified in A/D-converted value addition mode.
Note 2. Used when 1, 2, 3, or 4 conversion times is specified in A/D-converted value addition mode.

Settings for flush-right data with 10-bit accuracy in A/D-converted value addition mode



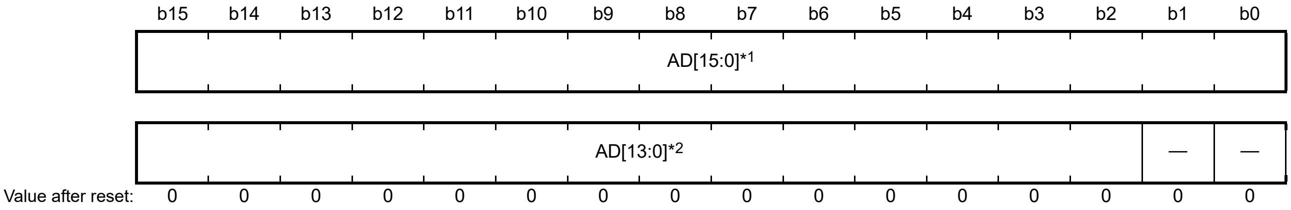
Bit	Symbol	Bit name	Description	R/W
b11 to b0	AD[11:0]	Added Value 11 to 0	12-bit sum of A/D conversion results.	R
b15 to b12	—	Reserved	These bits are read as 0.	R

Settings for flush-right data with 8-bit accuracy in A/D-converted value addition mode



Bit	Symbol	Bit name	Description	R/W
b9 to b0	AD[9:0]	Added Value 9 to 0	10-bit sum of A/D conversion results	R
b15 to b10	—	Reserved	These bits are read as 0.	R

Settings for flush-left data with 12-bit accuracy in A/D-converted value addition mode



Bit	Symbol	Bit name	Description	R/W
b15 to b0	AD[15:0]*1	Added Value 15 to 0	16 -bit sum of A/D conversion results.	R

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0.	R
b15 to b2	AD[13:0]*2	Added Value 13 o 0	14-bit sum of A/D conversion results.	R

Note 1. Used when 16 conversion times is selected in A/D-converted value addition mode.
Note 2. Used when 1, 2, 3, or 4 conversion times is selected in A/D-converted value addition mode.

Settings for flush-left data with 10-bit accuracy in A/D-converted value addition mode

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0.	R
b15 to b4	AD[11:0]	Added Value 11 to 0	12-bit sum of A/D conversion results.	R

Settings for flush-left data with 8-bit accuracy in A/D-converted value addition mode

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[9:0]										—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0.	R
b15 to b6	AD[9:0]	Added Value 9 to 0	10-bit sum of A/D conversion results.	R

47.2.2 A/D Self-Diagnosis Data Register (ADRD)

ADRD is a 16-bit read-only register that holds the A/D conversion results based on the self-diagnosis of the ADC12. In addition to the AD[11:0] bits indicating the A/D-converted value, it includes the self-diagnosis status bit (DIAGST).

The following conditions determine the formats for data in this registers:

- The setting in the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right setting)
- The setting in the A/D Conversion Accuracy Specify bits (ADCER.ADPRC[1:0]) (8-, 10-, or 12-bit setting).

The A/D-converted value addition and average modes cannot be applied to the A/D self-diagnosis function. For details on self-diagnosis, see section 47.2.11, A/D Control Extended Register (ADCER).

This section describes the data formats for each condition.

Settings for flush-right data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value.	R
b13, b12	—	Reserved	These bits are read as 0.	R
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis not executed after power-on 0 1: Self-diagnosis was executed using the 0 V voltage 1 0: Self-diagnosis was executed using the reference power supply*1 voltage x 1/2 1 1: Self-diagnosis was executed using the reference power supply*1 voltage. For details on self-diagnosis, see section 47.2.11, A/D Control Extended Register (ADCER).	R

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

Settings for flush-right data with 10-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		—	—	—	—	AD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value.	R
b13 to b10	—	Reserved	These bits are read as 0.	R
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis not executed after power-on 0 1: Self-diagnosis was executed using the 0 V voltage 1 0: Self-diagnosis was executed using the reference power supply*1 voltage x 1/2 1 1: Self-diagnosis was executed using the reference power supply*1 voltage For details on self-diagnosis, see section 47.2.11, A/D Control Extended Register (ADCER).	R

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

Settings for flush-right data with 8-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		—	—	—	—	—	—	AD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value	R
b13 to b8	—	Reserved	These bits are read as 0.	R
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis not executed after power-on 0 1: Self-diagnosis was executed using the 0 V voltage 1 0: Self-diagnosis was executed using the reference power supply*1 voltage x 1/2 1 1: Self-diagnosis was executed using the reference power supply*1 voltage. For details on self-diagnosis, see section 47.2.11, A/D Control Extended Register (ADCER).	R

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

Settings for flush-left data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	DIAGST[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis not executed after power-on 0 1: Self-diagnosis using the voltage of 0 V was executed 1 0: Self-diagnosis using the voltage of reference power supply*1 × 1/2 was executed 1 1: Self-diagnosis using the voltage of reference power supply*1 was executed. For details on self-diagnosis, see section 47.2.11, A/D Control Extended Register (ADCER).	R
b3, b2	—	Reserved	These bits are read as 0.	R
b15 to b4	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

Settings for flush-left data with 10-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[9:0]										—	—	—	—	DIAGST[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis not executed after power-on 0 1: Self-diagnosis was executed using the 0 V voltage 1 0: Self-diagnosis was executed using the reference power supply*1 × 1/2 voltage 1 1: Self-diagnosis was executed using the reference power supply*1 voltage. For details on self-diagnosis, see section 47.2.11, A/D Control Extended Register (ADCER).	R
b5 to b2	—	Reserved	These bits are read as 0.	R
b15 to b6	AD[9:0]	Converted Value 9 to 0	10-bit A/D-converted value.	R

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

Settings for flush-left data with 8-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[7:0]								—	—	—	—	—	—	DIAGST[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis not executed after power-on 0 1: Self-diagnosis was executed using the 0 V voltage 1 0: Self-diagnosis was executed using the reference power supply*1 × 1/2 voltage 1 1: Self-diagnosis was executed using the reference power supply*1 voltage. For details on self-diagnosis, see section 47.2.11, A/D Control Extended Register (ADCER).	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b15 to b8	AD[7:0]	Converted Value 7 to 0	8-bit A/D-converted value	R

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

47.2.3 A/D Control Register (ADCSR)

Address(es): ADC120.ADCSR 4005 C000h, ADC121.ADCSR 4005 C200h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS[1:0]		—			TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double-triggered operation. The setting is only valid in double-trigger mode.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt and ELC Event Enable	0: Disable ADC12i_GBADI interrupt generation on Group B scan completion 1: Enable ADC12i_GBADI interrupt generation on Group B scan completion. Group B scan only works in group scan mode.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselect double-trigger mode 1: Select double-trigger mode.	R/W
b8	EXTRG	Trigger Select*1	0: Start A/D conversion by a synchronous trigger (ELC) 1: Start A/D conversion by the asynchronous trigger (ADTRGi).	R/W
b9	TRGE	Trigger Start Enable	0: Disable A/D conversion to be started by the synchronous or asynchronous trigger 1: Enable A/D conversion to be started by the synchronous or asynchronous trigger.	R/W
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited.	R/W
b15	ADST	A/D Conversion Start	0: Stop A/D conversion process 1: Start A/D conversion process.	R/W

i = 0 for unit 0, and i = 1 for unit 1.

Note 1. To start A/D conversion using an external pin (asynchronous trigger):
After a high-level signal is input to the external pin (ADTRG0 in unit 0; ADTRG1 in unit 1), write 1 to both the TRGE and EXTRG bits in ADCSR and drive the external pin signals low. With these settings, the scan conversion process starts on detection of the falling edge of ADTRG0 in unit 0 and ADTRG1 in unit 1. For this configuration, the pulse width of the low-level input must be at least 1.5 PCLKB clock cycles.

DBLANS[4:0] bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double-trigger mode. The A/D conversion results from the specified analog input channel are stored in A/D Data Register y when conversion is started by the first trigger, and in the A/D Data Duplexing Register when started by the second trigger. Table 47.4 shows the channel selection settings for double-triggered operation.

A/D-converted value addition/average mode can be set with double-trigger mode for the channel selected in the DBLANS[4:0] bits by using the ADADS0 and ADADS1 registers. In double-trigger mode, the channels selected in the ADANSA0 and ADANSA1 registers are invalid, and the channel selected in the DBLANS[4:0] bits is A/D-converted instead.

When double-trigger mode is used in group scan mode, double-trigger control is only applied to Group A and not to Group B. This means that multi-channel analog input, temperature sensor output, and internal reference voltage can be selected for Group B even in double-trigger mode.

Only set the DBLANS[4:0] bits while the ADST bit is 0. Do not set them at the same time that you write 1 to the ADST bit.

To enter A/D-converted value addition/average mode when in double-trigger mode, set the channel selected in the DBLANS[4:0] bits in the ADANSA0 and ADANSA1 registers.

Table 47.4 Relationship between DBLANS bit settings and double-trigger enabled channels

Unit 0		Unit 1	
DBLANS[4:0]	**Duplication channel**	**DBLANS[4:0]**	**Duplication channel**
00000	AN000	00000	AN100
00001	AN001	00001	AN101
00010	AN002	00010	AN102
00011	AN003	00011	AN103
00100	AN004	00100	—
00101	AN005	00101	AN105
00110	AN006	00110	AN106
00111	AN007	00111	AN107
Unit 0		**Unit 1**	
DBLANS[4:0]	**Duplication channel**	**DBLANS[4:0]**	**Duplication channel**
10000	AN016	10000	AN116
10001	AN017	10001	AN117
10010	AN018	10010	AN118
10011	AN019	10011	AN119
10100	AN020	10100	—

Note: A/D-converted data from the self-diagnosis function, temperature sensor output, and internal reference voltage cannot be used in double-trigger mode.
Settings other than those listed in Table 47.4 are prohibited.

GBADIE bit (Group B Scan End Interrupt and ELC Event Enable)

The GBADIE bit enables or disables Group B scan end interrupt (ADC12i_GBADI (i = 0, 1)) in group scan mode.

DBLE bit (Double Trigger Mode Select)

The DBLE bit selects or deselects double-trigger mode. Double-trigger mode can only be operated by the synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits.

Double-trigger operation is as follows:

1. The ADC12i_ADI (i = 0, 1) interrupt is not output on completion of the first conversion but on completion of the second conversion.
2. The A/D conversion results from the duplication channel (selected in DBLANS[4:0]) started by the first trigger are stored in A/D Data Register y and those started by the second trigger are stored in the A/D Data Duplexing Register.

When DBLE is set, selecting double-trigger mode, the channels specified in the ADANSA0 and ADANSA1 registers are invalid. Double-trigger mode is deselected by setting DBLE to 0. Setting DBLE to 1 again enables the same double-trigger operation described in 1. and 2. for the first time scanning with the first trigger.

Do not select double-trigger mode in continuous scan mode. Additionally, do not select double-trigger mode for conversion of the temperature sensor output or internal reference voltage except for Group B scan in group scan mode. Software triggering cannot be set in double-trigger mode. Always clear the ADST bit to 0 before setting the DBLE bit. In other words, do not set the DBLE bit at that same time as writing 1 to the ADST bit.

EXTRG bit (Trigger Select)

The EXTRG bit selects the synchronous or asynchronous trigger as the trigger for starting A/D conversion.

TRGE bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous and asynchronous triggers. In group scan mode, set this bit to 1.

ADCS[1:0] bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs, up to a maximum of 13 channels in unit 0 and 11 channels in unit 1, and selected in the ADANSA0 and ADANSA1 registers in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, the scan conversion stops. When the temperature sensor output or internal reference voltage is selected, A/D conversion of the designated analog input channels is followed by A/D conversion of the temperature sensor output and the internal reference voltage, in that order.

In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs selected in the ADANSA0 and ADANSA1 registers in ascending order of channel number, and when 1 cycle of A/D conversion completes for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion stops even if scanning is in progress. When the temperature sensor output or internal reference voltage is selected, A/D conversion of the designated analog input channels is followed by A/D conversion of the temperature sensor output and the internal reference voltage, in that order.

In group scan mode, scanning is started by the synchronous trigger (ELC) selected in the TRSA[5:0] bits in ADSTRGR. A/D conversion is performed on the Group A analog inputs, up to the maximum channels selected in the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops. On the same trigger, A/D conversion is also performed on the Group B analog inputs, up to the maximum channels selected in the ADANSB0 and ADANSB1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops. If the conversion processes in Group A and Group B occur at the same time, those conversions cannot be controlled separately. In this case, set the Group A Priority Control Setting bit (ADGSPCR.PGS) in the A/D Group Scan Priority Control Register (ADGSPCR) to 1 to give priority to Group A conversion. When the temperature sensor output or internal reference voltage is selected, A/D conversion of the designated analog input channels is followed by A/D conversion of the temperature sensor output and the internal reference voltage, in that order.

In group scan mode, select different channels and triggers for Group A and Group B. Clear the ADST bit to 0 before setting the ADCS[1:0] bits. In other words, do not set the ADCS[1:0] bits at the same time as writing 1 to the ADST bit.

Table 47.5 Selectable targets for A/D conversion depending on scan and double-trigger mode settings

Scan mode setting	Double-trigger mode setting	Targets for A/D conversion				
		Self-diagnosis	Analog input (including Group A)	Analog input (Group B)	Temperature sensor output	Internal reference voltage
Single scan	DBLE = 0	✓	✓	-	✓	✓
	DBLE = 1	-	✓ (1 ch only)	-	-	-
Continuous scan	DBLE = 0	✓	✓	-	✓	✓
	DBLE = 1	-	-	-	-	-
Group scan	DBLE = 0	✓	✓	✓	✓	✓
	DBLE = 1	-	✓ (1 ch only)	✓	✓	✓

✓: Selectable. -: Not selectable.

ADST bit (A/D Conversion Start)

The ADST bit starts or stops the A/D conversion process. Before setting the ADST bit to 1, set the A/D conversion clock, conversion mode, and analog input for the conversion target.

[Setting conditions]

- 1 is written by software
- The synchronous trigger (ELC) selected in the ADSTRGR.TRSA[5:0] bits is detected when ADCSR.EXTRG is 0 and ADCSR.TRGE is 1
- The synchronous trigger (ELC) selected in the ADSTRGR.TRSB[5:0] bits is detected when ADCSR.TRGE is set to 1 in group scan mode
- The asynchronous trigger is detected when the ADCSR.TRGE and ADCSR.EXTRG bits are set to 1 and the ADSTRGR.TRSA[5:0] bits are set to 000000b
- When Group A priority control operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1, and each time A/D conversion of Group B starts.

[Clearing conditions]

- 0 is written by software
- The A/D conversion of all the selected channels, the temperature sensor output or the internal reference voltage completes in single scan mode
- Group A scan completes in group scan mode
- Group B scan completes in group scan mode
- When Group A priority control operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1, and each time a scanning of Group B completes.

Note: When Group A priority control operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.
When Group A priority control operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 0. When forcing A/D conversion to terminate, follow the procedure for clearing the ADST bit.

Note: If the single scan continuous function is used (ADGSPCR.GBRP = 1) when the group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADST bit is retained to 1.

47.2.4 A/D Channel Select Register A0 (ADANSA0)

Address(es): ADC120.ADANSA0 4005 C004h, ADC121.ADANSA0 4005 C204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSA07	ANSA06	ANSA05	ANSA04	ANSA03	ANSA02	ANSA01	ANSA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	ANSA07 to ANSA00	A/D Conversion Channels Select	0: Do not select associated input channel 1: Select associated input channel.	R/W
b15 to b8	-	Reserved.	These bits are read as 0. The write value should be 0.	R/W

ANSAn bits (n = 00 to 07) (A/D Conversion Channels Select)

The ADANSA0.ANSAn bits select or deselect the analog input channels for A/D conversion for AN000 to AN007 (unit 0) and AN100 to AN103 and AN105 to AN107 (unit 1). The channels and the number of channels can be set arbitrarily. In unit 0, the ANSA00 bit is associated with AN000 and the ANSA07 bit with AN007. In unit 1, the ANSA00 bit is associated with AN100 and the ANSA07 bit with AN107.

In double-trigger mode, the channel selected in the ADANSA0 register is invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in Group A instead.

In group scan mode, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

Only set the ADANSA0 register while the ADCSR.ADST bit is 0.

47.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): ADC120.ADANSA1 4005 C006h, ADC121.ADANSA1 4005 C206h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ANSA20	ANSA19	ANSA18	ANSA17	ANSA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	ANSA20 to ANSA16	A/D Conversion Channels Select	0: Do not select associated input channel 1: Select associated input channel	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSAn bits (n = 16 to 20) (A/D Conversion Channels Select)

The ADANSA1.ANSAn bits select or deselect the analog input channels for A/D conversion for AN016 to AN020 (unit 0) and AN116 to AN119 (unit 1). The channels and the number of channels can be set arbitrarily. In unit 0, the ANSA16 bit is associated with AN016 and the ANSA20 bit with AN020. In unit 1, the ANSA16 bit is associated with AN116 and the ANSA19 bit with AN119.

In double-trigger mode, the ANSA1[15:0] bits are invalid, and the channel selected in the ADCSR.DBLANS[15:0] bits is selected in Group A instead.

In group scan mode, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

Only set the ADANSA1 register while the ADCSR.ADST bit is 0.

47.2.6 A/D Channel Select Register B0 (ADANSB0)

Address(es): ADC120.ADANSB0 4005 C014h, ADC121.ADANSB0 4005 C214h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSB07	ANSB06	ANSB05	ANSB04	ANSB03	ANSB02	ANSB01	ANSB00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	ANSB07 to ANSB00	A/D Conversion Channels Select	0: Do not select associated input channel 1: Select associated input channel.	R/W
b15 to b8	-	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSBn bits (n = 00 to 07) (A/D Conversion Channels Select)

The ADANSB0.ANSBn bits select the analog input channels for A/D conversion for AN000 to AN007 (unit 0) and AN100 to AN103 and AN105 to AN107 (unit 1) in Group B in group scan mode. The ADANSB0 register is only used for group scan mode, not for any other modes. Exclude the channels specified in Group A (the channels associated with Group A, selected in the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double-trigger mode), both the selected channels and the number of channels to be set.

In unit 0, the ANSB00 bit is associated with AN000 and the ANSB07 bit with AN007. In unit 1, the ANSB00 bit is associated with AN100 and the ANSB07 bit with AN107.

Only set the ADANSB0 register while the ADCSR.ADST bit is 0.

47.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): ADC120.ADANSB1 4005 C016h, ADC121.ADANSB1 4005 C216h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ANSB20	ANSB19	ANSB18	ANSB17	ANSB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	ANSB20 to ANSB16	A/D Conversion Channels Select	0: Do not select associated input channel 1: Select associated input channel.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSBn bits (n = 16 to 20) (A/D Conversion Channels Select)

The ADANSB1.ANSBn bits select the analog input channels for A/D conversion for AN016 to AN020 (unit 0) and AN116 to AN119 (unit 1) in Group B in group scan mode. The ADANSB1 register is only used for group scan mode, not for any other modes. Exclude the channels specified in Group A (the channels associated with Group A, selected in the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double-trigger mode), both the selected channels and the number of channels to be set.

In unit 0, the ANSB16 bit is associated with AN016 and the ANSB20 bit with AN020. In unit 1, the ANSB16 bit is associated with AN116 and the ANSB19 bit with AN119.

Only set the ADANSB1 register bits while the ADST bit is 0.

47.2.8 A/D-Converted Value Addition/Average Channel Select Register 0 (ADADS0)

Address(es): ADC120.ADADS0 4005 C008h, ADC121.ADADS0 4005 C208h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ADS07	ADS06	ADS05	ADS04	ADS03	ADS02	ADS01	ADS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	ADS07 to ADS00	A/D-Converted Value Addition/Average Channel Select	0: Do not select associated input channel 1: Select associated input channel	R/W
b15 to b8	-	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (n = 00 to 07) (A/D-Converted Value Addition/Average Channel Select)

When the ADSn bit with the same number as the A/D-converted channel selected in the ANSAn bits (n = 00 to 07) in ADANSA0 or the ADCSR.DBLANS[4:0] bits and the ANSBn bits (n = 00 to 07) in ADANSB0 is set to 1, A/D conversion of the analog input of the selected channels is performed successively 1 to 16 times, as specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. For A/D-converted channels for which addition or average mode is not selected, a normal one-time conversion is executed, and the conversion result is stored in the A/D data register.

In unit 0, the ADS00 bit is associated with AN000 and the ADS07 bit with AN007. In unit 1, the ADS00 bit is associated with AN100 and the ADS07 bit with AN107.

Only set the ADADS0 register bits while the ADCSR.ADST bit is 0.

47.2.9 A/D-Converted Value Addition/Average Channel Select Register 1 (ADADS1)

Address(es): ADC120.ADADS1 4005 C00Ah, ADC121.ADADS1 4005 C20Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	ADS20	ADS19	ADS18	ADS17	ADS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	ADS20 to ADS16	A/D-Converted Value Addition/Average Channel Select	0: Do not select associated input channel 1: Select associated input channel.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (n = 16 to 20) (A/D-Converted Value Addition/Average Channel Select)

When the ADSn bit with the same number as the A/D-converted channel selected in the ANSAn bits (n = 16 to 20) in ADANSA1 or ADCSR.DBLANS[4:0] bits and ANSBn bits (n = 16 to 20) in ADANSB1 is set to 1, A/D conversion of the analog input of the selected channels is performed successively 1 to 16 times, as specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. For A/D-converted channels for which addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored in the A/D data register.

In unit 0, the ADS16 bit is associated with AN016 and the ADS20 bit with AN020. In unit 1, the ADS16 bit is associated with AN116 and the ADS19 bit with AN119.

Only set the ADADS1 register while the ADCSR.ADST bit is 0.

Figure 47.3 shows a scanning operation sequence in which both the ADADS0.ADS02 and ADS06 bits are set to 1.

In this example, addition mode is selected (ADADS.AVEE = 0), the time conversion is set to 4 (ADADC.ADC[1:0] = 11b), and the AN000 to AN006 channels are selected (ADANSA0.ANSA0[15:0] = 007Fh) in continuous scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with AN000. The AN002 conversion is performed successively 4 times, and the added (integrated) value is returned to A/D data register ADDR2. Next, the AN003 conversion process starts. The AN006 conversion is performed successively 4 times and the added (integrated) value is returned to A/D data register ADDR6. After conversion of AN006, the conversion operation is once again performed in the same sequence from AN000.

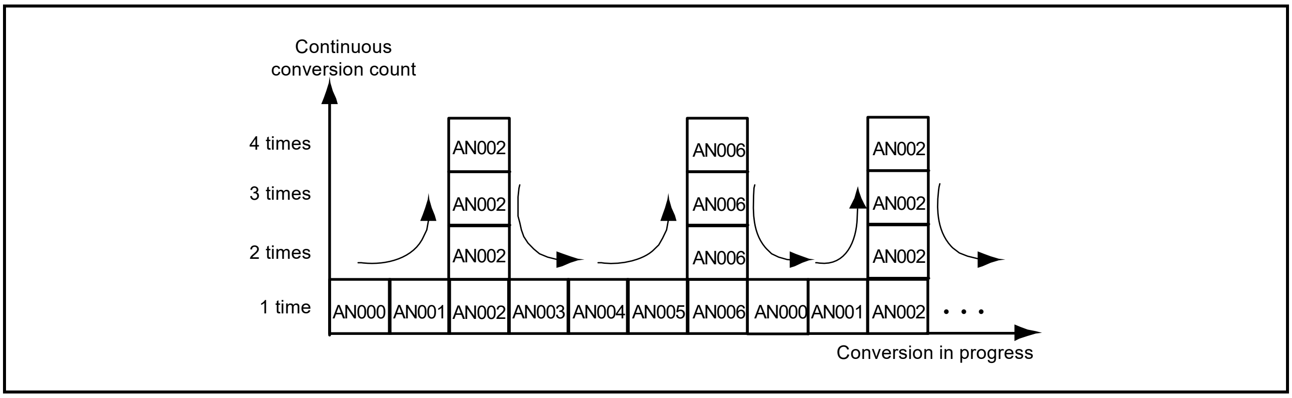


Figure 47.3 Scan conversion sequence with ADADC.ADC[2:0] = 011b, ADADS0.ADS02 = 1, and ADS06 = 1

47.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): ADC120.ADADC 4005 C00Ch, ADC121.ADADC 4005 C20Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	ADC[2:0]	Count Select	b2 b0 0 0 0: 1-time conversion (no addition; same as normal conversion) 0 0 1: 2 time conversion (one addition) 0 1 0: 3-time conversion (two additions) 0 1 1: 4 time conversion (three additions) 1 0 1: 16-time conversion (15 additions). Other settings are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Disable average mode*1 1: Enable average mode.*2	R/W

Note 1. When average mode is deselected by setting the ADADC.AVEE bit to 0, set the addition count to 1, 2, 3, 4 or 16-time conversion. 16-time conversion can only be used with 12-bit accuracy selected.

Note 2. When average mode is selected by setting the ADADC.AVEE bit to 1, set the addition count to 1-, 2-, or 4-time conversion. Do not set the addition count to 3- or 16-time conversion (ADC[2:0] = 010b or 101b).

ADC[2:0] bits (Count Select)

The ADC[2:0] bits set the count for all channels for which A/D conversion and addition/average mode are selected, including the channels selected in double-trigger mode in the ADCSR.DBLANS[4:0] bits. The count also applies to A/D conversion of temperature sensor output and internal reference voltage.

When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the count to 3-time conversion

(ADADC.ADC[2:0] = 010b). Additionally, the combination of 16-time conversion (ADADC.ADC[2:0] = 101b) with a conversion accuracy setting of 8 or 10 bits (ADCER.ADPRC[1:0] = 10b or 01b) is a prohibited setting, as described in section 47.2.2.

Only set the ADC[2:0] bits while the ADCSR.ADST bit is 0. When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[2:0] bits to any value other than 000b. When the conversion accuracy is 8 or 10 bits (ADCER.ADPRC[1:0] = 10b or 01b), do not set the ADC[2:0] bits to 101b.

AVEE bit (Average Mode Enable)

The AVEE bit selects addition or average mode for all channels for which A/D conversion and addition/average mode are selected, including the channels selected in double-trigger mode in the ADCSR.DBLANS[4:0] bits, temperature sensor output, and internal reference voltage.

When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to 3-time conversion (ADADC.ADC[2:0] = 010b).

Only set the AVEE bits while the ADCSR.ADST bit is 0.

47.2.11 A/D Control Extended Register (ADCER)

Address(es): ADC120.ADCER 4005 C00Eh, ADC121.ADCER 4005 C20Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	ADPRC[1:0]		—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2, b1	ADPRC[1:0]	A/D Conversion Accuracy Specify	b2 b1 0 0: 12-bit accuracy 0 1: 10-bit accuracy 1 0: 8-bit accuracy 1 1: Setting prohibited.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	A/D Data Register Automatic Clearing Enable	0: Disable automatic clearing 1: Enable automatic clearing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited when self-diagnosis is enabled 0 1: 0 V 1 0: Reference power supply voltage*1 x 1/2 1 1: Reference power supply voltage.*1	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Select rotation mode for self-diagnosis voltage 1: Select fixed mode for self-diagnosis voltage.	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disable ADC12 self-diagnosis 1: Enable ADC12 self-diagnosis.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Select flush-right for the A/D data register format 1: Select flush-left for the A/D data register format.	R/W

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

ADPRC[1:0] bits (A/D Conversion Accuracy Specify)

The ADPRC[1:0] bits set the A/D conversion accuracy to 8-, 10-, or 12-bit accuracy. Changing the A/D conversion accuracy also changes the bit width of valid data stored in the result register and the A/D conversion time. For details, see section 47.3.6, Analog Input Sampling and Scan Conversion Time.

Only set the ADPRC[1:0] bits while the ADCSR.ADST bit is 0.

ACE bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0s) of ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR after any of these registers is read by the CPU, DTC, or DMAC. Automatic clearing of the A/D data registers enables detection of failures that do not update the A/D data registers.

DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis fixed voltage mode. For details, see the ADCER.DIAGLD bit description.

Do not execute self-diagnosis by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

DIAGLD bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or fixed voltage is used in self-diagnosis. Setting this bit to 0 allows conversion of the voltages in rotation mode where 0 V, the reference power supply × 1/2, and the reference power supply are converted, in that order. After reset, when the self-diagnosis voltage rotation mode is selected, self-diagnosis is executed from 0 V. The fixed voltage specified in the ADCER.DIAGVAL[1:0] bits is converted when self-diagnosis fixed voltage mode is selected. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion completes. When scan conversion restarts, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Only set the DIAGLD bit while the ADCSR.ADST bit is 0.

DIAGM bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis. Self-diagnosis is used to detect a failure of the ADC12. In self-diagnosis mode, one of the internally generated voltage values (0, the reference power supply × 1/2, or the reference power supply) is converted. When conversion completes, information on the converted voltage and the conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). ADRD can be read by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When the double-trigger mode is set (ADCSR.DBLE = 1), always deselect self-diagnosis (DIAGM = 0). When self-diagnosis is selected in group scan mode, self-diagnosis is executed separately on Group A and Group B.

Only set the DIAGM bit while the ADCSR.ADST bit is 0.

ADRFMT bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADCMPDR0/1, ADWINLLB, ADWINULB, or ADRD.

Only set the ADRFMT bit the ADCSR.ADST bit is 0.

47.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): ADC120.ADSTRGR 4005 C010h, ADC121.ADSTRGR 4005 C210h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B	These bits specify the A/D conversion start trigger for Group B in group scan mode.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b13 to b8	TRSA[5:0]	A/D Conversion Start Trigger Select	These bits specify the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for Group A is selected.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TRSB[5:0] bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in Group B. The TRSB[5:0] bits must only be set in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for Group B, setting a software trigger or an asynchronous trigger is prohibited. In group scan mode, set the TRSB[5:0] bits to a value other than 000000b and set the ADCSR.TRGE bit to 1.

When Group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows Group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by the trigger might have no effect.

When the trigger from a module operated at 120 MHz (GPT) is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see section 47.3.6, Analog Input Sampling and Scan Conversion Time.

Table 47.6 lists the A/D conversion startup sources selected in the TRSB[5:0] bits.

TRSA[5:0] bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in Group A is selected. When scanning is executed in group scan mode or double-trigger mode, setting a software trigger or an asynchronous trigger is prohibited.

When using a synchronous trigger (ELC) as the A/D conversion start source, set the TRGE bit in ADCSR to 1 and the EXTRG bit in ADCSR to 0.

When using the asynchronous trigger (ADTRGn), set the TRGE bit in ADCSR to 1 and the EXTRG bit in ADCSR to 1.

The software trigger (ADCSR.ADST) is enabled regardless of the settings in the ADCSR.TRGE bit, the ADCSR.EXTRG bit, or the TRSA[5:0] bits. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by the trigger might have no effect.

When the trigger from a module operated at 120 MHz (GPT) is selected as an A/D conversion start trigger, a delay period for synchronization processing occurs. For details, see section 47.3.6, Analog Input Sampling and Scan Conversion Time.

Table 47.7 lists the A/D conversion start sources selected in the TRSA[5:0] bits.

Table 47.6 Selection of A/D conversion start sources in the TRSB[5:0] bits

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselected state		1	1	1	1	1	1
ELC_ADC00 (unit 0), ELC_ADC10 (unit 1)	ELC	0	0	1	0	0	1
ELC_ADC01 (unit 0), ELC_ADC11 (unit 1)	ELC	0	0	1	0	1	0
ELC_ADC00/ELC_ADC01 (unit 0) ELC_ADC10/ELC_ADC11 (unit 1)	ELC	0	0	1	0	1	1

Table 47.7 Selection of A/D activation sources in the TRSA[5:0] bits

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselected state		1	1	1	1	1	1
ADTRGn	Input pin for the trigger	0	0	0	0	0	0
ELC_ADC00 (unit 0), ELC_ADC10 (unit 1)	ELC	0	0	1	0	0	1
ELC_ADC01 (unit 0), ELC_ADC11 (unit 1)	ELC	0	0	1	0	1	0
ELC_ADC00/ELC_ADC01 (unit 0) ELC_ADC10/ELC_ADC11 (unit 1)	ELC	0	0	1	0	1	1

47.2.13 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): ADC120.ADEXICR 4005 C012h, ADC121.ADEXICR 4005 C212h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	OCSB	TSSB	OCSA	TSSA	—	—	—	—	—	—	OCSAD	TSSAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select	0: Do not select addition/average mode for temperature sensor output 1: Select addition/average mode for temperature sensor output.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Do not select addition/average mode for internal reference voltage 1: Select addition/average mode for internal reference voltage.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage.	R/W
b10	TSSB	Temperature Sensor Output A/D Conversion Select for Group B	Selection for Group B in group scan mode. 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output.	R/W
b11	OCSB	Internal Reference Voltage A/D Conversion Select for Group B	Selection for Group B in group scan mode. 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TSSAD bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. The maximum addition count differs depending on the conversion accuracy (see section 47.2.2, A/D Self-Diagnosis Data Register (ADRD)). When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D Temperature sensor Data Register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADTSDR.

Only set the TSSAD bit while the ADCSR.ADST bit is 0.

OCSAD bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. The maximum addition count differs depending on the conversion accuracy (see 47.2.2 A/D Self-Diagnosis Data Register (ADRD)). When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D Internal Reference Voltage Data Register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADOCDR.

Only set the OCSAD bit while the ADCSR.ADST bit is 0.

TSSA bit (Temperature Sensor Output A/D Conversion Select)

The TSSA bit selects A/D conversion of the temperature sensor output for Group A in single scan mode, continuous scan mode, or group scan mode. When A/D conversion of the temperature sensor output is selected and performed, set the ADCSR.DBLE bit to 0.

Only set the TSSA bit while the ADCSR.ADST bit is 0.

OCSA bit (Internal Reference Voltage A/D Conversion Select)

The OCSA bit selects A/D conversion of the internal reference voltage for Group A in single scan mode, continuous scan mode, or group scan mode. When A/D conversion of the internal reference voltage is selected and performed, set the ADCSR.DBLE bit to 0.

Only set the OCSA bit while the ADCSR.ADST bit is 0. In addition, wait for 400 ns or more after the OCSA bit is set to 1 before starting A/D conversion.

TSSB bit (Temperature Sensor Output A/D Conversion Select for Group B)

The TSSB bit selects A/D conversion of the temperature sensor output for Group B in group scan mode.

Only set the TSSB bit while the ADCSR.ADST bit is 0. Do not set the TSSB bit to 1 while the TSSA bit is 1.

OCSB bit (Internal Reference Voltage A/D Conversion Select for Group B)

The OCSB bit selects A/D conversion of the internal reference voltage for Group B in group scan mode.

Only set the OCSB bit while the ADCSR.ADST bit is 0. Do not set the OCSB bit to 1 while the OCSA bit is 1.

Moreover, start the A/D conversion after waiting for 400 ns or more after the OCSB bit is set to 1.

47.2.14 A/D Sampling State Register n (ADSSTRn) (n = 00 to 07, L, T, O)

Address(es): ADC120.ADSSTR00 4005 C0E0h to ADC120.ADSSTR07 4005 C0E7h,
ADC120.ADSSTRL 4005 C0DDh, ADC120.ADSSTRT 4005 C0DEh, ADC120.ADSSTRO 4005 C0DFh,
ADC121.ADSSTR00 4005 C2E0h to ADC121.ADSSTR03 4005 C2E3h,
ADC121.ADSSTR05 4005 C2E5h to ADC121.ADSSTR07 4005 C2E7h,
ADC121.ADSSTRL 4005 C2DDh, ADC121.ADSSTRT 4005 C2DEh, ADC121.ADSSTRO 4005 C2DFh

	b7	b6	b5	b4	b3	b2	b1	b0
	SST[7:0]							
Value after reset:	0	0	0	0	1	0	1	1

Bit	Symbol	Bit name	Description	R/W
b7 to b0	SST[7:0]	Sampling Time Setting	These bits set the sampling time in the range from 5 to 255 states.	R/W

The ADSSTRn register sets the sampling time for analog input. If one state is 1 ADCLK (A/D conversion clock) cycle and the ADCLK clock is 60 MHz, one state is 16.7 ns. The initial value is 11 states. The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow.

Only set the SST[7:0] bits while the ADCSR.ADST bit is 0.

The lower limit of the sampling time setting depends on the frequency ratio, as follows:

- If the frequency ratio of PCLKB to PCLKC(ADCLK) = 1:1, 2:1, 4:1, or 8:1, the sampling time must be set to a value of more than 5 states

- If the frequency ratio of PCLKB to PCLKC(ADCLK) = 1:2 or 1:4, the sampling time must be set to a value of more than 6 states.

Table 47.8 shows the relationship between the A/D Sampling State Register and the associated channels.

For details, see section 47.3.6, Analog Input Sampling and Scan Conversion Time.

Table 47.8 Relationship between A/D sampling state register and associated channels

	Associated channels	
Bit name	**Unit 0**	**Unit 1**
ADSSTR00.SST[7:0] bits*[1]	AN000	AN100
ADSSTR01.SST[7:0] bits	AN001	AN101
ADSSTR02.SST[7:0] bits	AN002	AN102
ADSSTR03.SST[7:0] bits	AN003	AN103
ADSSTR04.SST[7:0] bits	AN004	-
ADSSTR05.SST[7:0] bits	AN005	AN105
ADSSTR06.SST[7:0] bits	AN006	AN106
ADSSTR07.SST[7:0] bits	AN007	AN107
ADSSTRL.SST[7:0] bits	AN016 to AN020	AN116 to AN119
ADSSTRT.SST[7:0] bits	Temperature sensor output	Temperature sensor output
ADSSTRO.SST[7:0] bits	Internal reference voltage	Internal reference voltage

Note 1. When the self-diagnosis function is selected, the sampling time set in the ADSSTR00.SST[7:0] bits is applied.

47.2.15 A/D Sample and Hold Circuit Control Register (ADSHCR)

Address(es): ADC120.ADSHCR 4005 C066h, ADC121.ADSHCR 4005 C266h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	SSTSH[7:0]	Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting	Sampling time (4 to 255 states).	R/W
b10 to b8	SHANS[2:0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select	Select whether to use or bypass channel-dedicated sample-and-hold circuits for AN000 to AN002 (unit 0) and AN100 to AN102 (unit 1). 0: Bypass the circuits 1: Use the circuits.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SSTSH[7:0] bits (Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting)

The SSTSH[7:0] bits set the sampling time for the channel-dedicated sample-and-hold circuits. If one state is 1 ADCLK (A/D conversion clock) cycle and the ADCLK clock is 60 MHz, one state is 16.7 ns. The initial value is 24 states. The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow.

Only set the SSTSH[7:0] bits while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and 255 or less.

SHANS[2:0] bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

The SHANS[2:0] bits select whether to use or bypass the channel-dedicated sample-and-hold circuits for AN000 to AN002 (unit 0) and AN100 to AN102 (unit 1). In unit 0, the SHANS[0] bit is associated with AN000, the SHANS[1] bit with AN001, and the SHANS[2] bit with AN002. In unit 1, the SHANS[0] bit is associated with AN100, the SHANS[1] bit with AN101, and the SHANS[2] bit with AN102.

If any channel from among AN000 to AN002 (unit 0) or AN100 to AD102 (unit 1) is selected for Group B while operation is in group scan mode under Group A priority control, use this setting to bypass the dedicated sample-and-hold circuit of the channel.

Only set the SHANS[2:0] bits while the ADCSR.ADST bit is 0 and the ADSHMSR.SHMD bit is 0.

47.2.16 A/D Sample and Hold Operation Mode Selection Register (ADSHMSR)

Address(es): ADC120.ADSHMSR 4005 C07Ch, ADC121.ADSHMSR 4005 C27Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SHMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SHMD	Sampling Operation Selection	0: Disable continuous sampling function 1: Enable continuous sampling function.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SHMD bit (Sampling Operation Selection)

Setting SHMD to 1 enables the constant sampling function of the channel-dedicated sample-and-hold selected in the ADSHCR.SHANS[2:0] bits. Only set the SHMD bit while the ADCSR.ADST bit is 0.

When the sampling function is enabled, the sample-and-hold circuit operates sampling while the ADC12 is not operating, and operates holding while the ADC12 is operating.

Note: The ADCSR.ADST bit must become 1 after 400 ns or more elapses after the SHMD bit is set to 1 (when the permissible signal source impedance is 1 kΩ). The sampling period of the sample-and-hold circuit must be 400 ns or more (when the permissible signal source impedance is 1 kΩ).

47.2.17 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): ADC120.ADDISCR 4005 C07Ah, ADC121.ADDISCR 4005 C27Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PCHG	ADNDIS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	ADNDIS[3:0]	Disconnection Detection Assist Setting	b3 - b0 0000: The disconnection detection assist function is disabled 0001: Setting prohibited Others: The number of states for the discharge or precharge period.	R/W
b4	PCHG	Precharge/discharge select	0: Discharge 1: Precharge.	R/W

Bit	Symbol	Bit name	Description	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADDISCR register selects either precharge or discharge, and the period of precharge or discharge for the A/D disconnection detection assist function. Only set the ADDISCR register when the ADCSR.ADST bit is 0.

If any of the following functions are used, the disconnection detection assist function must be disabled:

- The temperature sensor
- The internal reference voltage
- A/D self-diagnosis
- The programmable gain amplifier without bypass enabled.

ADNDIS[3:0] bits (Disconnection Detection Assist Setting)

The ADNDIS[3:0] bits specify the period of precharge or discharge. When ADNDIS[3:0] = 0000b, the disconnection detection assist function is disabled. Setting the ADNDIS[3:0] bits to 0001b is prohibited. Except when ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge or discharge. When the ADNDIS[3:0] bits are set to any values other than 0000b or 0001b, the disconnection detection assistance function is enabled.

PCHG bit (Precharge/discharge select)

Setting the PCHG bit to 1 selects precharge and setting the PCHG bit to 0 selects discharge.

47.2.18 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): ADC120.ADGSPCR 4005 C080h, ADC121.ADGSPCR 4005 C280h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GBRP	—	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PGS	Group A Priority Control Setting[*1]	0: Operate without Group A priority control 1: Operate with Group A priority control.	R/W
b1	GBRSCN	Group B Restart Setting	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Do not restart Group B scanning after it is stopped by Group A priority control 1: Restart Group B scanning after it is stopped by Group A priority control.	R/W
b14 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	GBRP	Group B Single Scan Continuous Start[*2]	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Do not continuously activate single scan for Group B 1: Continuously activate single scan for Group B.	R/W

Note 1. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting PGS to 1. Operation is not guaranteed if these bits are set to any other value.

Note 2. When the GBRP bit is set to 1, single scan is performed continuously for Group B regardless of the setting in the GBRSCN bit.

PGS bit (Group A Priority Control Setting)

Set the PGS bit to 1 to give priority to operation on Group A. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting the PGS bit to 1. Operation is not guaranteed if the bits are set to any other value.

When the PGS bit is set to 0, a clear operation must be performed by software as described in section 47.6.2, Constraints on Stopping A/D Conversion. When the PGS bit is set to 1, use the settings described in section 47.3.4.3, Operation with group A priority control.

GBRSCN bit (Group B Restart Setting)

The GBRSCN bit controls the restarting of scan operation on Group B when operation on Group A is given priority. If a scan operation on Group B is stopped by a Group A trigger input with the GBRSCN bit set to 1, the scan operation is restarted on completion of the Group A conversion. Also, if a Group B trigger is input during A/D conversion on Group A, the scan operation on Group B is restarted on completion of Group A conversion.

When the GBRSCN bit is set to 0, triggers input during A/D conversion are ignored. Only set the GBRSCN bit while the ADCSR.ADST bit is 0.

The setting in the GBRSCN bit is valid when the PGS bit is 1.

GBRP bit (Group B Single Scan Continuous Start)

Set the GBRP bit to perform a single scan operation continuously on Group B. Setting the GBRP bit to 1 starts a single scan on Group B. On completion of the scan, another single scan on Group B starts automatically. If a Group B conversion stops because of an operation on Group A, the Group A operation takes priority, and single scan on Group B restarts automatically on completion of the Group A conversion.

Disable Group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting in the GBRSCN bit. Only set the GBRP bit while the ADCSR.ADST is 0.

The setting in the GBRP bit is valid when the PGS bit is 1.

47.2.19 A/D Compare Function Control Register (ADCMPCR)

Address(es): ADC120.ADCMPCR 4005 C090h, ADC121.ADCMPCR 4005 C290h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPAIE	WCMPE	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CMPAB[1:0]	Window A/B Composite Conditions Setting	b1 b0 0 0: Output ADC12i_WCMPM when Window A OR Window B comparison conditions are met; otherwise, output ADC12i_WCMPUM 0 1: Output ADC12i_WCMPM when Window A EXOR Window B comparison conditions are met; otherwise, output ADC12i_WCMPUM 1 0: Output ADC12i_WCMPM when Window A AND Window B comparison conditions are met; otherwise, output ADC12i_WCMPUM 1 1: Setting prohibited. These bits are valid when both Window A and Window B are enabled (CMPAE = 1 and CMPBE = 1).	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	CMPBE	Compare Window B Operation Enable	0: Disable compare Window B operation Disable ADC12i_WCMPM and ADC12i_WCMPUM outputs. 1: Enable compare Window B operation.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	CMPAE	Compare Window A Operation Enable	0: Disable compare Window A operation Disable ADC12i_WCMPM and ADC12i_WCMPUM outputs. 1: Enable compare Window A operation.	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b13	CMPBIE	Compare B Interrupt Enable	0: Disable ADC12i_CMPBI interrupt when comparison conditions (Window B) are met 1: Enable ADC12i_CMPBI interrupt when comparison conditions (Window B) are met.	R/W
b14	WCMPE	Window Function Setting	0: Disable window function Window A and Window B operate as a comparator to compare the single value on the lower side with the A/D conversion result. 1: Enable window function. Window A and Window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result.	R/W
b15	CMPAIE	Compare A Interrupt Enable	0: Disable ADC12i_CMPAI interrupt when comparison conditions (Window A) are met 1: Enable ADC12i_CMPAI interrupt when comparison conditions (Window A) are met.	R/W

Note: i = 0: unit 0, i = 1: unit 1.

CMPAB[1:0] bits (Window A/B Composite Conditions Setting)

The CMPAB[1:0] bits are valid when both Window A and Window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits specify the compare function match/mismatch event output conditions and monitoring conditions of ADWINMON.MONCONB.

Only set the CMPAB[1:0] bits while the ADCSR.ADST bit is 0.

CMPBE bit (Compare Window B Operation Enable)

The CMPBE bit enables or disables the compare Window B operation. Set the CMPBE bit while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, and B1 (ADANSA0, ADANSA1, ADANSB0, and ADANSB1)
- OCSB, TSSB, OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- CMPCHB[5:0] bits in the Window B Channel Select Register (ADCMPBNSR).

CMPAE bit (Compare Window A Operation Enable)

The CMPAE bit enables or disables the compare Window A operation. Set the CMPAE bit while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, and B1 (ADANSA0, ADANSA1, ADANSB0, and ADANSB1)
- OCSB, TSSB, OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- Window A Channel Select Registers 0 and 1 (ADCMPANSR0 and ADCMPANSR1)
- Window A Extended Input Select Register (ADCMPANSER)

CMPBIE bit (Compare B Interrupt Enable)

The CMPBIE bit enables or disables the ADC12i_CMPBI (i = 0, 1) interrupt output when the comparison conditions (Window B) are met.

WCMPE bit (Window Function Setting)

The WCMPE bit enables or disables the window function. Set the WCMPE bit while the ADCSR.ADST bit is 0.

CMPAIE bit (Compare A Interrupt Enable)

The CMPAIE bit enables or disables the ADC12i_CMPAI (i = 0, 1) interrupt output when the comparison conditions (Window A) are met.

47.2.20 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0)

Address(es): ADC120.ADCMPANSR0 4005 C094h, ADC121.ADCMPANSR0 4005 C294h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPCHA07	CMPCHA06	CMPCHA05	CMPCHA04	CMPCHA03	CMPCHA02	CMPCHA01	CMPCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CMPCHA07 to CMPCHA00	Compare Window A Channel Select	0: Disable compare function for associated input channel 1: Enable compare function for associated input channel In unit 0, bit [7] (CMPCHA07) is associated with to AN007 and bit 0 (CMPCHA00) with AN000. In unit 1, bit [7] (CMPCHA07) is associated with AN107 and bit 0 (CMPCHA00) with AN100.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (n = 00 to 07) (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bit with the same number as the A/D conversion channel selected in the ADANSA0.ANSAn bits (n = 00 to 07) and the ADANSB0.ANSBn bits (n = 00 to 07).

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

47.2.21 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1)

Address(es): ADC120.ADCMPANSR1 4005 C096h, ADC121.ADCMPANSR1 4005 C296h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CMPCHA20	CMPCHA19	CMPCHA18	CMPCHA17	CMPCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	CMPCHA20 to CMPCHA16	Compare Window A Channel Select	0: Disable compare function for associated input channel 1: Enable compare function for associated input channel. In unit 0, bit 4 (CMPCHA20) is associated with AN020 and bit 0 (CMPCHA16) with AN016. In unit 1, bit [3] (CMPCHA19) is associated with AN119 and bit 0 (CMPCHA16) with AN116.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (n = 16 to 20) (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bit with the same number as the A/D conversion channel selected in the ADANSA1.ANSAn bits (n = 16 to 20) and the ADANSB1.ANSBn bits (n = 16 to 20).

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

47.2.22 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): ADC120.ADCMPANSER 4005 C092h, ADC121.ADCMPANSER 4005 C292h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPOCA	CMPTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPTSA	Temperature Sensor Output Compare Select	0: Exclude the temperature sensor output from the compare Window A target range 1: Include the temperature sensor output in the compare Window A target range.	R/W
b1	CMPOCA	Internal Reference Voltage Compare Select	0: Exclude the internal reference voltage from the compare Window A target range. 1: Include the internal reference voltage in the compare Window A target range.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPTSA bit (Temperature Sensor Output Compare Select)

The compare Window A function is enabled by setting the CMPTSA bit to 1 while the ADEXICR.TSSA bit or the ADEXICR.TSSB bit is 1. Set the CMPTSA bit while the ADCSR.ADST bit is 0.

CMPOCA bit (Internal Reference Voltage Compare Select)

The compare Window A function is enabled by setting the CMPOCA bit to 1 while the ADEXICR.OCSA bit or the ADEXICR.OCSB bit is 1. Set the CMPOCA bit while the ADCSR.ADST bit is 0.

47.2.23 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

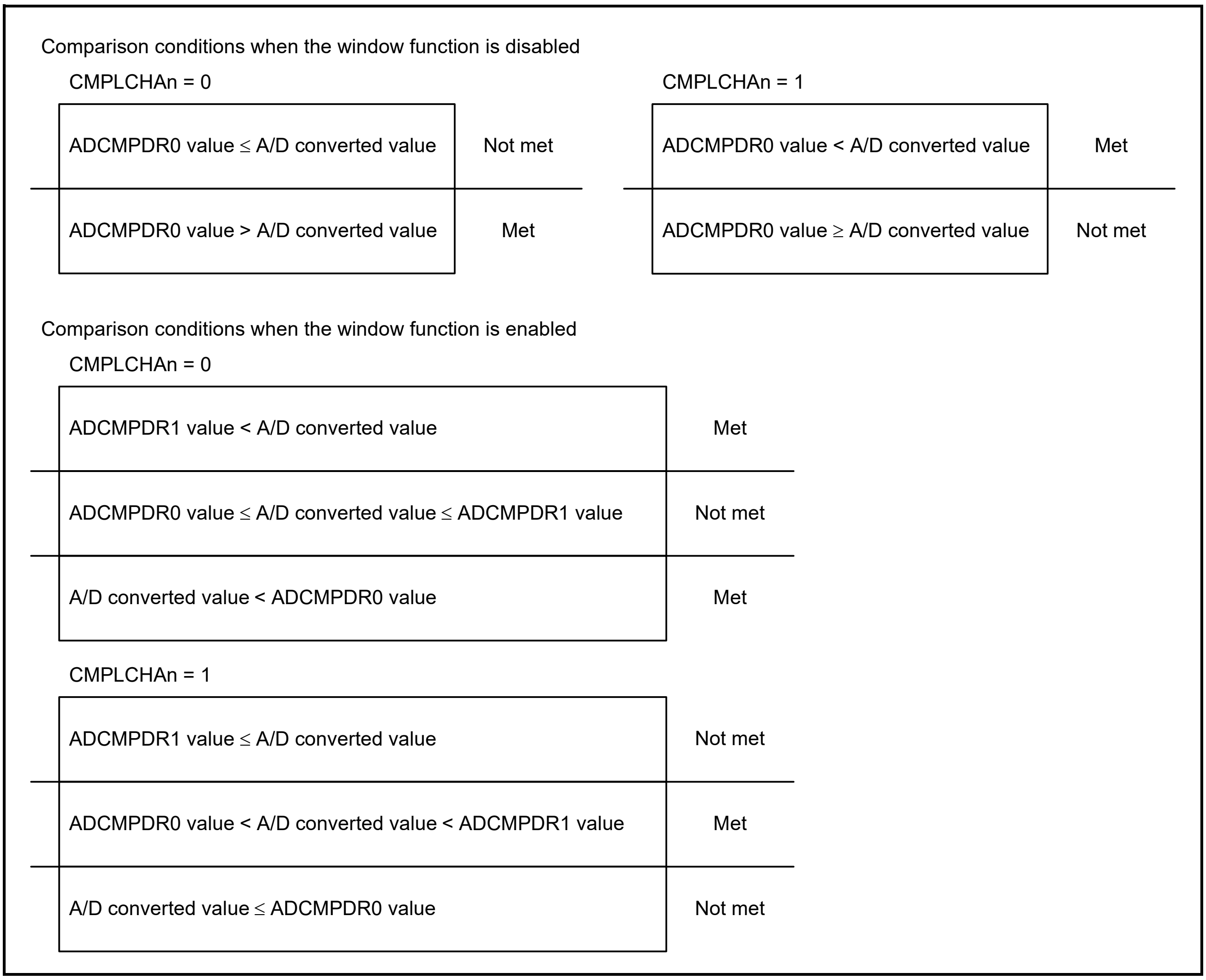
Address(es): ADC120.ADCMPLR0 4005 C098h, ADC121.ADCMPLR0 4005 C298h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPLCHA07	CMPLCHA06	CMPLCHA05	CMPLCHA04	CMPLCHA03	CMPLCHA02	CMPLCHA01	CMPLCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CMPLCHA07 to CMPLCHA00	Compare Window A Comparison Condition Select	These bits set comparison conditions for channels to which Window A comparison conditions are applied, selected from AN000 to AN007 (unit 0) and AN100 to AN103, AN105 to AN107 (unit 1). Comparison conditions are shown in Figure 47.4. • When window function is disabled (ADCMPCR.WCMPE = 0) 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When window function is enabled (ADCMPCR.WCMPE = 1) 0: A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: ADCMPDR0 value < A/D-converted value < ADCMPDR1 value.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (n = 00 to 07) (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for channels to which Window A comparison conditions are applied, selected from AN000 to AN007 (unit 0) and AN100 to AN103, AN105 to AN107 (unit 1). These bits can be set for each analog input to be compared. In unit 0, CMPLCHA00 is associated with AN000 and CMPLCHA07 with AN007. In unit 1, CMPLCHA00 is associated with AN100 and CMPLCHA07 with AN107. When the comparison result of each analog input meets the set condition, the ADCMPSR0.CMPSTCHAn flag sets to 1 and a compare interrupt (ADC12i_CMPAI (i = 0, 1)) is generated.



Comparison conditions when the window function is disabled

CMPLCHAn = 0

ADCMPDR0 value ≤ A/D converted value	Not met
ADCMPDR0 value > A/D converted value	Met

CMPLCHAn = 1

ADCMPDR0 value < A/D converted value	Met
ADCMPDR0 value ≥ A/D converted value	Not met

Comparison conditions when the window function is enabled

CMPLCHAn = 0

ADCMPDR1 value < A/D converted value	Met
ADCMPDR0 value ≤ A/D converted value ≤ ADCMPDR1 value	Not met
A/D converted value < ADCMPDR0 value	Met

CMPLCHAn = 1

ADCMPDR1 value ≤ A/D converted value	Not met
ADCMPDR0 value < A/D converted value < ADCMPDR1 value	Met
A/D converted value ≤ ADCMPDR0 value	Not met

Figure 47.4 Explanation of comparison conditions for compare function Window A

47.2.24 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1)

Address(es): ADC120.ADCMPLR1 4005 C09Ah, ADC121.ADCMPLR1 4005 C29Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CMPLCHA20	CMPLCHA19	CMPLCHA18	CMPLCHA17	CMPLCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	CMPLCHA20 to CMPLCHA16	Compare Window A Comparison Condition Select	These bits set comparison conditions for channels to which Window A comparison conditions are applied, selected from AN016 to AN020 (unit 0) and AN116 to AN119 (unit 1). Comparison conditions are shown in Figure 47.4. • When window function is disabled (ADCMPCR.WCMPE = 0) 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When window function is enabled (ADCMPCR.WCMPE = 1) 0: A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: ADCMPDR0 value < A/D-converted value < ADCMPDR1 value.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (n = 16 to 20) (Compare Window A Comparison Condition Select)

The CMPLCHAn bits specify the comparison conditions for channels to which Window A comparison conditions are applied, selected from AN016 to AN020 (unit 0) and AN116 to AN119 (unit 1). These bits can be set for each analog input to be compared. In unit 0, CMPLCHA16 is associated with AN016 and CMPLCHA20 with AN020. In unit 1, CMPLCHA16 is associated with AN116 and CMPLCHA19 with AN119. When the comparison result of each analog input meets the set condition, the ADCMPSR1.CMPSTCHAn flag sets to 1 and a compare interrupt (ADC12i_CMPAI (i = 0, 1)) is generated.

47.2.25 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): ADC120.ADCMPLER 4005 C093h, ADC121.ADCMPLER 4005 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPLOCA	CMPLTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select	Comparison conditions are shown in Figure 47.4. • When window function is disabled (ADCMPCR.WCMPE = 0) 0: ADCMPDR0 value > A/D-converted value 1: ADCMPDR0 value < A/D-converted value. • When window function is enabled (ADCMPCR.WCMPE = 1) 0: A/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: ADCMPDR0 value < A/D-converted value < ADCMPDR1 value.	R/W.

Bit	Symbol	Bit name	Description	R/W
b1	CMPLOCA	Compare Window A Internal Reference Voltage Comparison Condition Select	Comparison conditions are shown in Figure 47.4. • When window function is disabled (ADCMPCR.WCMPE = 0) 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value. • When window function is enabled (ADCMPCR.WCMPE = 1) 0: A/D-converted value < ADCMPDR0 register value, or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

CMPLTSA bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

The CMPLTSA bit specifies comparison conditions when the temperature sensor output is the target for the Window A comparison condition. When the temperature sensor output comparison result meets the set condition, the ADCMPSER.CMPSTTSA flag sets to 1 and a compare interrupt (ADC12i_CMPAI (i = 0, 1)) is generated.

CMPLOCA bit (Compare Window A Internal Reference Voltage Comparison Condition Select)

The CMPLOCA bit specifies comparison conditions when the internal reference voltage is the target for the Window A comparison condition. When the internal reference voltage comparison result meets the set condition, the ADCMPSER.CMPSTOCA flag sets to 1 and a compare interrupt (ADC12i_CMPAI) is generated.

47.2.26 A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0), A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1), A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB), A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB)

Address(es): ADC120.ADCMPDR0 4005 C09Ch, ADC120.ADCMPDR1 4005 C09Eh,
ADC120.ADWINLLB 4005 C0A8h, ADC120.ADWINULB 4005 C0AAh,
ADC121.ADCMPDR0 4005 C29Ch, ADC121.ADCMPDR1 4005 C29Eh,
ADC121.ADWINLLB 4005 C2A8h, ADC121.ADWINULB 4005 C2AAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	Reference value	R/W

The ADCMPDRy (y = 0,1) register specifies the reference data when the compare Window A function is used. ADCMPDR0 sets the lower reference for Window A, and ADCMPDR1 sets the upper reference for Window A.

ADWINULB and ADWINLLB specify the reference data when the compare Window B function is used. ADWINLLB sets the lower reference for Window B, and ADWINULB sets the upper reference for Window B. The ADCMPDRy, ADWINULB, and ADWINLLB are read/write registers.

ADCMPDRy, ADWINULB, and ADWINLLB can be written to even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion.

Set these registers so that the upper reference is not less than the lower reference (ADCMPDR1 ≥ ADCMPDR0 and ADWINULBB ≥ ADWINLLB). ADCMPDR1 and ADWINULB are not used when the window function is disabled.

The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after a rewrite), and the lower reference (before a rewrite) with the A/D conversion result. (See Figure 47.5.) If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the associated compare window operation enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) is 0.

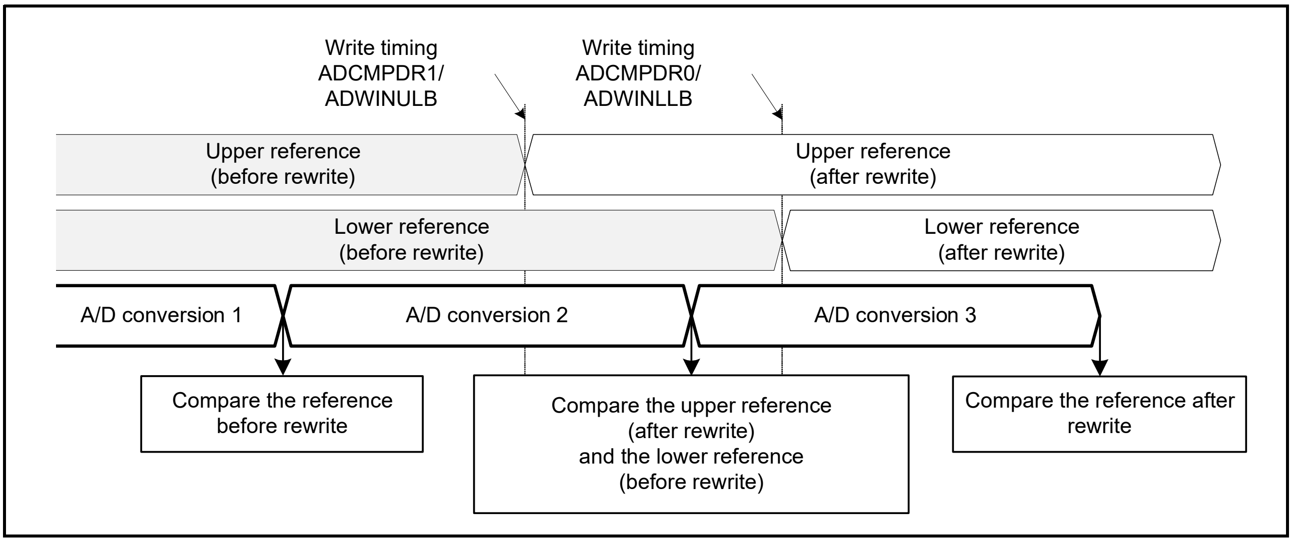


Figure 47.5 Comparison between upper and lower references before and after a rewrite

The ADCMPDRy, ADWINLLB, and ADWINULB registers use different formats depending on the following conditions:

- The value in the A/D Data Register Format Select bit (flush-right or flush-left)
- The value in the A/D Conversion Accuracy Specify bit (12-bit, 10-bit, or 8-bit)
- The value in the A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are as follows:

(1) When A/D-converted value addition mode is not selected

- Flush-right data with 12-bit accuracy: Lower 12 bits ([11:0]) are valid
- Flush-right data with 10-bit accuracy: Lower 10 bits ([9:0]) are valid
- Flush-right data with 8-bit accuracy: Lower 8 bits ([7:0]) are valid
- Flush-left data with 12-bit accuracy: Upper 12 bits ([15:4]) are valid
- Flush-left data with 10-bit accuracy: Upper 10 bits ([15:6]) are valid
- Flush-left data with 8-bit accuracy: Upper 8 bits ([15:8]) are valid.

(2) When A/D-converted value addition mode is selected

- Flush-right data with 12-bit accuracy: Lower 14 bits ([13:0]) are valid
- Flush-right data with 10-bit accuracy: Lower 12 bits ([11:0]) are valid
- Flush-right data with 8-bit accuracy: Lower 10 bits ([9:0]) are valid
- Flush-left data with 12-bit accuracy: Upper 14 bits ([15:2]) are valid
- Flush-left data with 10-bit accuracy: Upper 12 bits ([15:4]) are valid
- Flush-left data with 8-bit accuracy: Upper 10 bits ([15:6]) are valid.

47.2.27 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0)

Address(es): ADC120.ADCMPSR0 4005 C0A0h, ADC121.ADCMPSR0 4005 C2A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPSTCHA07	CMPSTCHA06	CMPSTCHA05	CMPSTCHA04	CMPSTCHA03	CMPSTCHA02	CMPSTCHA01	CMPSTCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CMPSTCHA07 to CMPSTCHA00	Compare Window A Flag	When Window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result of channels to which Window A comparison conditions are applied, selected from AN000 to AN007 (unit 0) and AN100 to AN103, AN105 to AN107 (unit 1). 0: Comparison conditions are not met 1: Comparison conditions are met.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn flags (n = 00 to 07) (Compare Window A Flag)

The CMPSTCHAn flags indicate the comparison results for channels to which Window A comparison conditions are applied, selected from AN000 to AN007 (unit 0) and AN100 to AN103, AN105 to AN107 (unit 1). When a comparison condition set in ADCMPLR0.CMPLCHAn is met at the end of A/D conversion, the associated CMPSTCHAn flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI (i = 0, 1)) is generated when this flag sets to 1. In unit 0, CMPSTCHA00 is associated with AN000 and CMPSTCHA07 with AN007. In unit 1, CMPSTCHA00 is associated with AN100 and CMPSTCHA07 with AN107.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR0.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

47.2.28 A/D Compare Function Window A Channel Status Register1 (ADCMPSR1)

Address(es): ADC120.ADCMPSR1 4005 C0A2h, ADC121.ADCMPSR1 4005 C2A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	CMPSTCHA20	CMPSTCHA19	CMPSTCHA18	CMPSTCHA17	CMPSTCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	CMPSTCHA20 to CMPSTCHA16	Compare Window A Flag	When Window A operation is enabled (ADCMPCR.CMPAE = 1), these bits indicate the comparison result of channels to which Window A comparison conditions are applied, selected from AN016 to AN020 (unit 0) and AN116 to AN119 (unit 1). 0: Comparison conditions are not met 1: Comparison conditions are met.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn flags (n = 16 to 20) (Compare Window A Flag)

The CMPSTCHAn flags indicate the comparison results for channels to which Window A comparison conditions are applied, selected from AN016 to AN020 (unit 0) and AN116 to AN119 (unit 1). When the comparison condition set in

ADCMPLR1.CMPLCHAn is met at the end of A/D conversion, the associated CMPSTCHAn flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI (i = 0, 1)) is generated when this flag sets to 1. In unit 0, CMPSTCHA16 is associated with AN016 and CMPSTCHA20 with AN020. In unit 1, CMPSTCHA16 is associated with AN116 and CMPSTCHA19 with AN119.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR1.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

47.2.29 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): ADC120.ADCMPSER 4005 C0A4h, ADC121.ADCMPSER 4005 C2A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPSTOCA	CMPSTTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag	When Window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the temperature sensor output comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b1	CMPSTOCA	Compare Window A Internal Reference Voltage Compare Flag	When Window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the internal reference voltage comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTTSA flag (Compare Window A Temperature Sensor Output Compare Flag)

The CMPSTTSA flag indicates the temperature sensor output comparison result. When the comparison condition set in ADCMPLER.CMPLTSA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTTSA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

CMPSTOCA flag (Compare Window A Internal Reference Voltage Compare Flag)

The CMPSTOCA flag indicates the internal reference voltage comparison result. When the comparison condition set in ADCMPLER.CMPLOCA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i_CMPAI) is generated when this flag sets to 1.

Writing 1 to the CMPSTOCA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLOCA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

47.2.30 A/D Compare Function Window B Channel Select Register (ADCMPBNSR)

Address(es): ADC120.ADCMPBNSR 4005 C0A6h, ADC121.ADCMPBNSR 4005 C2A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CMPCHB[5:0]	Compare Window B Channel Select	These bits select channels to be compared with the compare Window B conditions. The maximum channel is AN020 in unit 0 and AN119 in unit 1. b5 b0 Unit 0 Unit 1 0 0 0 0 0 0: AN000 AN100 0 0 0 0 0 1: AN001 AN101 0 0 0 0 1 0: AN002 AN102 0 0 0 0 1 1: AN003 AN103 0 0 0 1 0 0: AN004 — 0 0 0 1 0 1: AN005 AN105 0 0 0 1 1 0: AN006 AN106 0 0 0 1 1 1: AN007 AN107 0 1 0 0 0 0: AN016 AN116 : : 0 1 0 0 1 1: AN019 AN119 0 1 0 1 0 0: AN020 — 1 0 0 0 0 0: Temperature sensor 1 0 0 0 0 1: Internal reference voltage 1 1 1 1 1 1: Do no select. Other settings are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Compare Window B Comparison Condition Setting	This bit sets comparison conditions for channels for Window B. The comparison conditions are shown in Figure 47.6. • When window function is disabled (ADCMPCR.WCMPE = 0) 0: CMPLLB value > A/D-converted value 1: CMPLLB value < A/D-converted value. • When window function is enabled (ADCMPCR.WCMPE = 1) 0: A/D-converted value < CMPLLB value, or CMPULB value < A/D-converted value 1: CMPLLB value < A/D-converted value < CMPULB value.	R/W

CMPCHB[5:0] bits (Compare Window B Channel Select)

The CMPCHB[5:0] bits specify the channels to be compared with the compare Window B conditions from AN000 to AN007 and AN016 to AN020 (unit 0), AN100 to AN103, AN105 to AN107, and AN116 to AN119 (unit 1), the temperature sensor, and the internal reference voltage. The compare Window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected in the following bits:

Unit 0:

- ADANSA0.ANSA00 to ANSA07 bits
- ADANSA1.ANSA16 to ANSA20 bits
- ADANSB0.ANSB00 to ANSB07 bits
- ADANSB1.ANSB16 to ANSB20 bits.

Unit 1:

- ADANSA0.ANSA00 to ANSA03 bits
- ADANSA0.ANSA05 to ANSA07 bits
- ADANSA1.ANSA16 to ANSA19 bits
- ADANSB0.ANSB00 to ANSB03 bits
- ADANSB0.ANSB05 to ANSB07 bits
- ADANSB1.ANSB16 to ANSB19 bits.

Set the CMPCHB[5:0] bits while the ADCSR.ADST bit is 0.

CMPLB bit (Compare Window B Comparison Condition Setting)

The CMPLB bit specifies the comparison conditions for channels for Window B. When the comparison result of an analog input meets the set condition, the associated ADCMPBSR.CMPSTB flag sets to 1 and a compare interrupt request (ADC12i_CMPBI) (i = 0, 1)) is generated.

Compare conditions when the window function is disabled

CMPLB = 0

ADWINLLB value ≤ A/D converted value	Not met
ADWINLLB value > A/D converted value	Met

CMPLB = 1

ADWINLLB value < A/D converted value	Met
ADWINLLB value ≥ A/D converted value	Not met

Compare conditions when the window function is enabled

CMPLB = 0

A/D converted value > ADWINULB value	Met
ADWINLLB value ≤ A/D converted value ≤ ADWINULB value	Not met
A/D converted value < ADWINLLB value	Met

CMPLB = 1

A/D converted value ≥ ADWINULB value	Not met
ADWINLLB value <A/D converted value < ADWINULB value	Met
A/D converted value ≤ ADWINLLB value	Not met

Figure 47.6 Explanation of compare conditions for compare function Window B

47.2.31 A/D Compare Function Window B Status Register (ADCMPBSR)

Address(es): ADC120.ADCMPBSR 4005 C0ACh, ADC121.ADCMPBSR 4005 C2ACh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPSTB	Compare Window B Flag	When Window B operation is enabled (ADCMPCR.CMPBE = 1), this bit indicates the comparison result of channels to which Window B comparison conditions are applied, selected from AN000 to AN007 and AN016 to AN020 (unit 0), AN100 to AN103, AN105 to AN107, and AN116 to AN119 (unit 1), temperature sensor output, and internal reference voltage. 0: Comparison conditions are not met 1: Comparison conditions are met.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTB flag (Compare Window B Flag)

The CMPSTB flag indicates the comparison result of channels to which Window B comparison conditions are applied, selected from AN000 to AN007 and AN016 to AN020 (unit 0), AN100 to AN103, AN105 to AN107, and AN116 to AN119 (unit 1), the temperature sensor, and the internal reference voltage. When the comparison condition set in ADCMPBNSR.CMPLB is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPBIE bit is 1, a compare interrupt request (ADC12i_CMPBI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTB flag is invalid.

[Setting condition]

- The condition set in ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after reading 1.

47.2.32 A/D Compare Function Window A/B Status Monitor Register (ADWINMON)

Address(es): ADC120.ADWINMON 4005 C08Ch, ADC121.ADWINMON 4005 C28Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MONCOMB	Combination Result Monitor	This bit indicates the combination result. This bit is valid when both Window A and Window B operations are enabled. 0: Window A/B composite conditions are not met 1: Window A/B composite conditions are met.	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	MONCMPA	Comparison Result Monitor A	0: Window A comparison conditions are not met 1: Window A comparison conditions are met.	R
b5	MONCMPB	Comparison Result Monitor B	0: Window B comparison conditions are not met 1: Window B comparison conditions are met.	R
b7, b6	—	Reserved	These bits are read as 0.	R

MONCOMB bit (Combination Result Monitor)

The read-only MONCOMB bit indicates the combined result of comparison condition results A and B based on the combination condition set in the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set in the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set in the ADCMPCR.CMPAB[1:0] bits
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA bit (Comparison Result Monitor A)

The read-only MONCMPA bit is read as 1 when the A/D-converted value of the Window A target channel meets the condition set in ADCMPLR0/ADCMPLR1 and ADCMPLER. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPLR0.CMPLCHAn when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in ADCMPLR0.CMPLCHAn when ADCMPCR.CMPAE = 1
- ADCMPCR.CMPAE = 0 (automatically cleared when the ADCMPCR.CMPAE value changes from 1 to 0).

MONCMPB bit (Comparison Result Monitor B)

The read-only MONCMPB bit is read as 1 when the A/D-converted value of the Window B target channel meets the condition set in the ADCMPBNSR.CMPLB bit. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1
- ADCMPCR.CMPBE = 0 (automatically cleared when the ADCMPCR.CMPBE value changes from 1 to 0).

47.2.33 A/D Programmable Gain Amplifier Control Register (ADPGACR)

Address(es): ADC120.ADPGACR 4005 C1A0h, ADC121.ADPGACR 4005 C3A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P002GEN	P002ENAMP	P002SEL1	P002SEL0	P001GEN	P001ENAMP	P001SEL1	P001SEL0	P000GEN	P000ENAMP	P000SEL1	P000SEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	P000SEL0	PGA P000 Amplifier Bypass Enable	0: Do not output the signal in a path bypassing the PGA amplifier 1: Output the signal in a path bypassing the PGA amplifier.	R/W
b1	P000SEL1	PGA P000 Amplifier Transit Enable	0: Do not output the signal in a path through the PGA amplifier 1: Output the signal in a path through the PGA amplifier.	R/W
b2	P000ENAMP	PGA P000 Amplifier Enable	0: Do not use the PGA amplifier 1: Use the PGA amplifier.	R/W
b3	P000GEN	PGA P000 Gain Setting Enable	0: Disable gain setting 1: Enable gain setting.	R/W

Bit	Symbol	Bit name	Description	R/W
b4	P001SEL0	PGA P001 Amplifier Bypass Enable	0: Do not output the signal in a path bypassing the PGA amplifier 1: Output the signal in a path bypassing the PGA amplifier.	R/W
b5	P001SEL1	PGA P001 Amplifier Transit Enable	0: Do not output the signal in a path through the PGA amplifier 1: Output the signal in a path through the PGA amplifier.	R/W
b6	P001ENAMP	PGA P001 Amplifier Enable	0: Do not use the PGA amplifier 1: Use the PGA amplifier.	R/W
b7	P001GEN	PGA P001 Gain Setting Enable	0: Disable gain setting 1: Enable gain setting.	R/W
b8	P002SEL0	PGA P002 Amplifier Bypass Enable	0: Do not output the signal in a path bypassing the PGA amplifier 1: Output the signal in a path bypassing the PGA amplifier.	R/W
b9	P002SEL1	PGA P002 Amplifier Transit Enable	0: Do not output the signal in a path through the PGA amplifier 1: Output the signal in a path through the PGA amplifier.	R/W
b10	P002ENAMP	PGA P002 Amplifier Enable	0: Do not use the PGA amplifier 1: Use the PGA amplifier.	R/W
b11	P002GEN	PGA P002 Gain Setting Enable	0: Disable gain setting 1: Enable gain setting.	R/W
b12	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Note: See section 47.3.12, Programmable Gain Amplifiers, for details on setting these bits.

PnSEL0 bit (PGA Pn Amplifier Bypass Enable) (n = 000 to 002)

The PnSEL0 bit selects whether to output the signal in a path that bypasses the amplifier in the PGA for each programmable gain amplifier Pn.

PnSEL1 bit (PGA Pn Amplifier Transit Enable) (n = 000 to 002)

The PnSEL1 bit selects whether to output the signal in a path through the amplifier in the PGA for each programmable gain amplifier Pn.

PnENAMP bit (PGA Pn Amplifier Enable) (n = 000 to 002)

The PnENAMP bit selects whether to use the amplifier in the PGA for each programmable gain amplifier Pn.

PnGEN bit (PGA Pn Input Resistance Side Gain Selection Signal Enable) (n = 000 to 002)

The PnGEN bit enables or disables the gain setting for each programmable gain amplifier Pn.

47.2.34 A/D Programmable Gain Amplifier Gain Setting Register 0 (ADPGAGS0)

Address(es): ADC120.ADPGAGS0 4005 C1A2h, ADC121.ADPGAGS0 4005 C3A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P002GAIN[3:0]				P001GAIN[3:0]				P000GAIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	P000GAIN[3:0]	PGA P000 Gain Setting	• When differential input is disabled (ADPGADCR0.PnDEN = 0) 0 0 0 0: × 2.000 0 0 0 1: × 2.500 0 0 1 0: × 2.667 0 0 1 1: × 2.857 0 1 0 0: × 3.077 0 1 0 1: × 3.333 0 1 1 0: × 3.636 0 1 1 1: × 4.000 1 0 0 0: × 4.444 1 0 0 1: × 5.000 1 0 1 0: × 5.714 1 0 1 1: × 6.667 1 1 0 0: × 8.000 1 1 0 1: × 10.000 1 1 1 0: × 13.333. • When differential input is enabled (ADPGADCR0.PnDEN = 1)*1 0 0 0 1: × 1.500 0 1 0 1: × 2.333 1 0 0 1: × 4.000 1 0 1 1: × 5.667. Other settings are prohibited.	R/W
b7 to b4	P001GAIN[3:0]	PGA P001 Gain Setting		R/W
b11 to b8	P002GAIN[3:0]	PGA P002 Gain Setting		R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. For details on setting these bits, see section 47.3.12, Programmable Gain Amplifiers.

PnGAIN[3:0] bits (PGA Pn Gain Setting) (n = 000 to 002)

The PnGAIN[3:0] bits specify the gain of each PGA amplifier Pn. For differential inputs (ADPGADCR0.PnDEN = 1 and ADPGACR.PnGEN = 1), these bits set the gain in combination with ADPGADCR0.PnDG[1:0].

47.2.35 A/D Programmable Gain Amplifier Differential Input Control Register (ADPGADCR0)

Address(es): ADC120.ADPGADCR0 4005 C1B0h, ADC121.ADPGADCR0 4005 C3B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P002D EN	—	P002DG[1:0]		P001D EN	—	P001DG[1:0]		P000D EN	—	P000DG[1:0]	
Value after reset:	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	P000DG[1:0]	P000 Differential Input Gain Setting	When these bits are used, set {P000DEN, P000GEN} to 11b. b1 b0 0 0: × 1.5 0 1: × 2.333 1 0: × 4.0 1 1: × 5.667.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	P000DEN	P000 Differential Input Enable	0: Disable differential input 1: Enable differential input.	R/W
b5, b4	P001DG[1:0]	P001 Differential Input Gain Setting	When these bits are used, set {P001DEN, P001GEN} to 11b. b5 b4 0 0: × 1.5 0 1: × 2.333 1 0: × 4.0 1 1: × 5.667.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	P001DEN	P001 Differential Input Enable	0: Disable differential input 1: Enable differential input.	R/W
b9, b8	P002DG[1:0]	P002 Differential Input Gain Setting	When these bits are used, set {P002DEN, P002GEN} to 11b. b9 b8 0 0: × 1.5 0 1: × 2.333 1 0: × 4.0 1 1: × 5.667	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	P002DEN	P002 Differential Input Enable	0: Disable differential input 1: Enable differential input.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: For details on setting these bits, see section 47.3.12, Programmable Gain Amplifiers.

PnDG[1:0] bits (Pn Differential Input Gain Setting) (n = 000 to 002)

The PnDG[1:0] bits specify the gain of each PGA amplifier Pn when differential inputs are used. These bits are only valid when the PnDEN bit = 1 and the PnGEN bit = 1.

To use the PGA for differential inputs, set the ADPGADCR0.PnDG[1:0] bits in conjunction with the ADPGAGS0.PnGAIN[3:0] bits.

Example: To set the gain to × 1.5 using P000 for differential input, set:
ADPGAGS0.P000GAIN[3:0] = 0001b
ADPGADCR0.P000DG[1:0] = 00b

PnDEN bit (Pn Differential Input Enable) (n = 000 to 002)

The PnDEN bit enables or disables differential inputs for each PGA amplifier Pn.

47.3 Operation

47.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

Scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software. In group scan mode, the selected channels in Group A and the selected channels in Group B are scanned once after scan starts in response to the respective synchronous trigger (ELC).

In single scan mode and continuous scan mode, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for the ANn channels in Group A selected in the ADANSA0 and ADANSA1 registers first, and then for the ANn channels in Group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.

When self-diagnosis is selected, it is executed once at the beginning of each scan, and one of the three voltages generated internally in the ADC12 is converted.

The temperature sensor output and internal reference voltage can be selected at the same time as the analog input of the channels, and A/D conversion is performed on the analog input of channels, temperature sensor output, and internal reference voltage, in that order.

Double-trigger mode can be used with single scan mode or group scan mode. With double-trigger mode enabled (ADCSR.DBLE = 1), A/D conversion data of a channel selected in the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger (ELC) selected in the TRSA[5:0] bits in ADSTRGR. Only Group A in group scan mode can use the double-trigger mode.

The extended operation of double-trigger mode means the A/D conversion operation is generated from the synchronous trigger combination. This trigger combination is selected in ADSTRGR.TRSA[5:0] in double-trigger mode. ELC_AD00 and ELC_AD01 are associated with unit 0. ELC_AD10 and ELC_AD11 are associated with unit 1.

In the extended operation of double-trigger mode, in addition to normal double-trigger operation, A/D conversion data from the ELC_AD00 (unit 0) and ELC_AD10 (unit 1) triggers is stored in A/D Data Duplexing Register A (ADDBLDRA), and A/D conversion data from the ELC_AD01 (unit 0) and ELC_AD11 (unit 1) triggers is stored in A/D Data Duplexing Register B (ADDBLDRB). In the extended operation of double-trigger mode, when a combination of triggers occurs at the same time, the data duplexing register settings for the specified triggers do not work, and A/D conversion data is stored in A/D Data Duplexing Register B (ADDBLDRB).

When one synchronous trigger is input during the A/D conversion started by another synchronous trigger, the subsequent trigger is input when the other A/D conversion is canceled.

When any of the AN000 to AN002 (unit 0), and AN100 to AN102 (unit 1) channels are set as a channel-dedicated sample-and-hold circuit in the SHANS[2:0] bits in ADSHCR, the target analog input specified is sampled and held before the first A/D conversion of each scan.

47.3.2 Single Scan Mode

47.3.2.1 Basic operation without channel-dedicated sample-and-hold circuits

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).

4. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state.

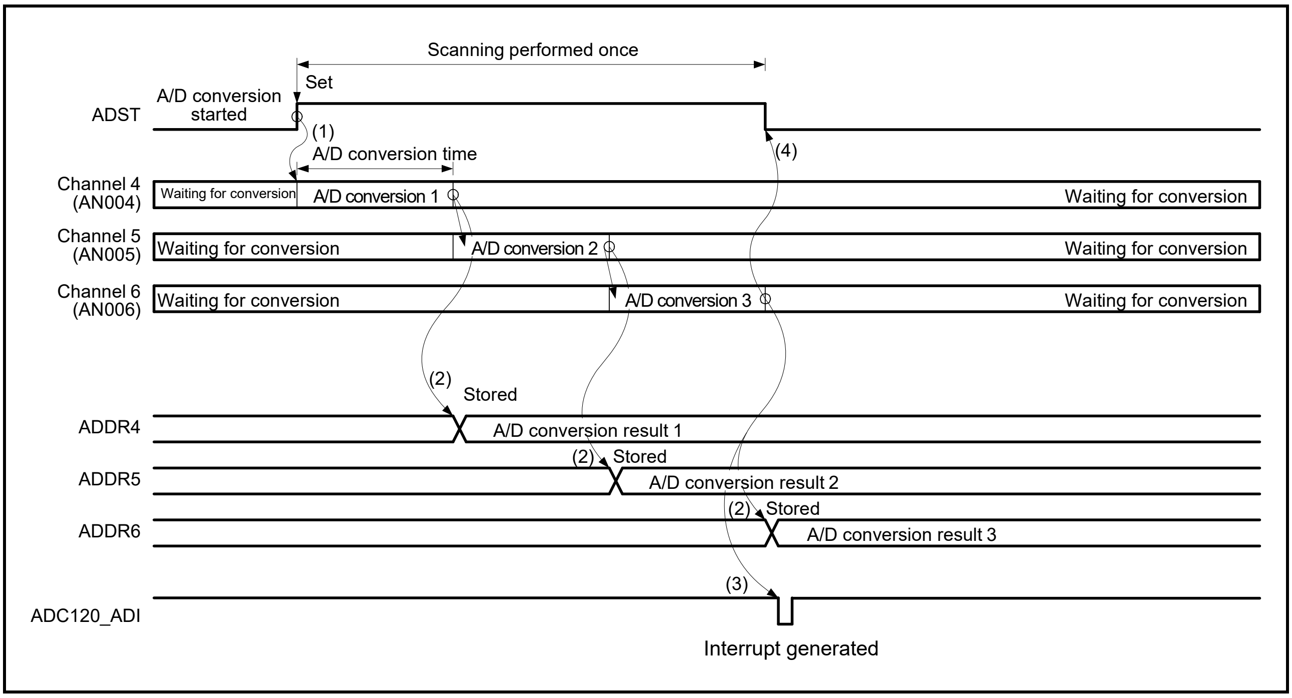


Figure 47.7 Example basic operation in single scan mode when AN004 to AN006 are selected

47.3.2.2 Basic operation with channel-dedicated sample-and-hold circuits and continuous sampling disabled

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed once on the analog input of all the specified channels. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[2:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling of all channels whose dedicated sample-and-hold circuit is to be used starts when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
5. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state.

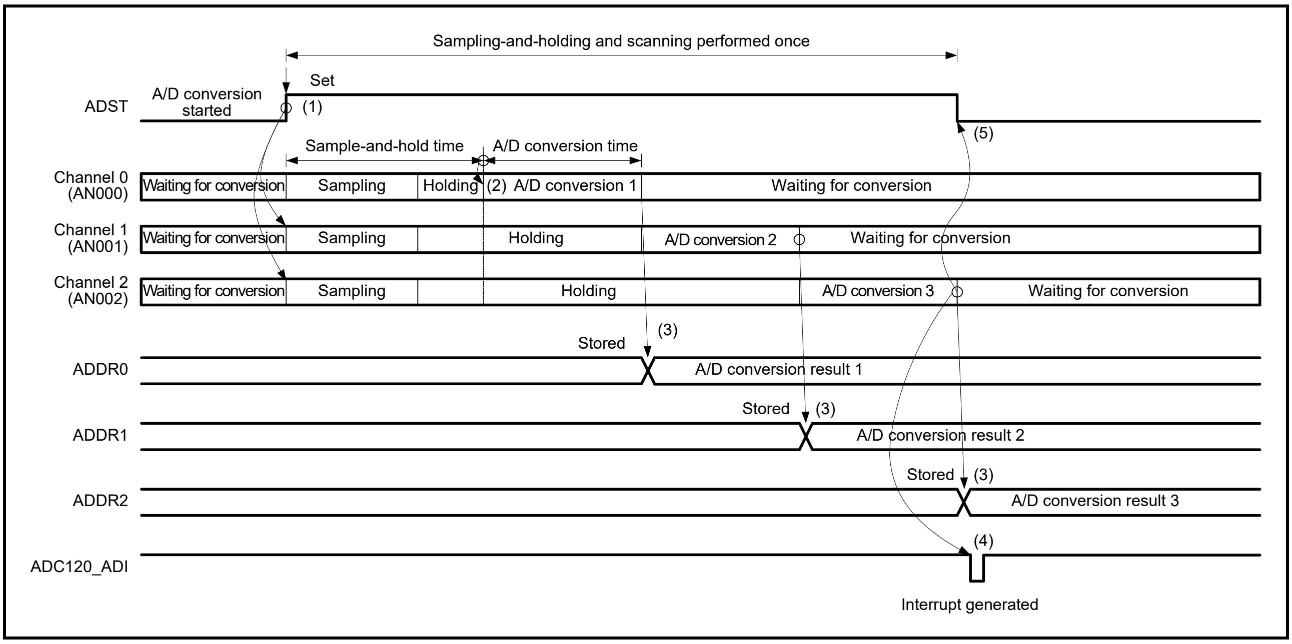


Figure 47.8 **Example operation in single scan mode when the channel-dedicated sample-and-hold circuits are used and AN000 to AN002 are selected**

47.3.2.3 Basic operation with channel-dedicated sample-and-hold circuits and continuous sampling enabled

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:0] bits specify the channels for which the channel-dedicated sample-and-hold circuits are to be used.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (ELC), or input of an asynchronous trigger.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
6. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).
7. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.

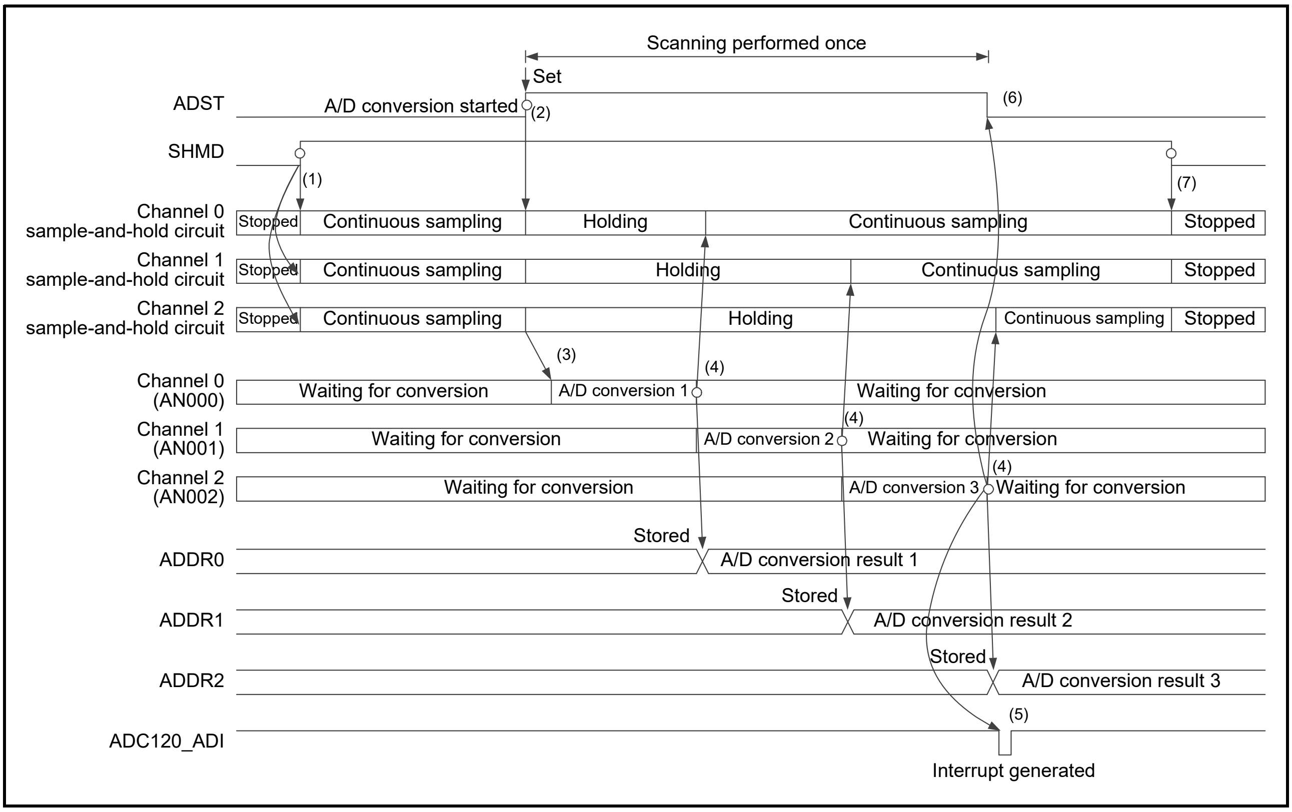


Figure 47.9 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000 to AN002 are selected, and continuous sampling is enabled

47.3.2.4 Channel selection and self-diagnosis without channel-dedicated sample-and-hold circuits

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0 (unit 0) or VREFH (unit 1) (× 0, × 1/2, or × 1) supplied to the ADC12, and then A/D conversion is performed once on the analog input of the specified channels.

The operation is as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input.
2. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
5. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state.

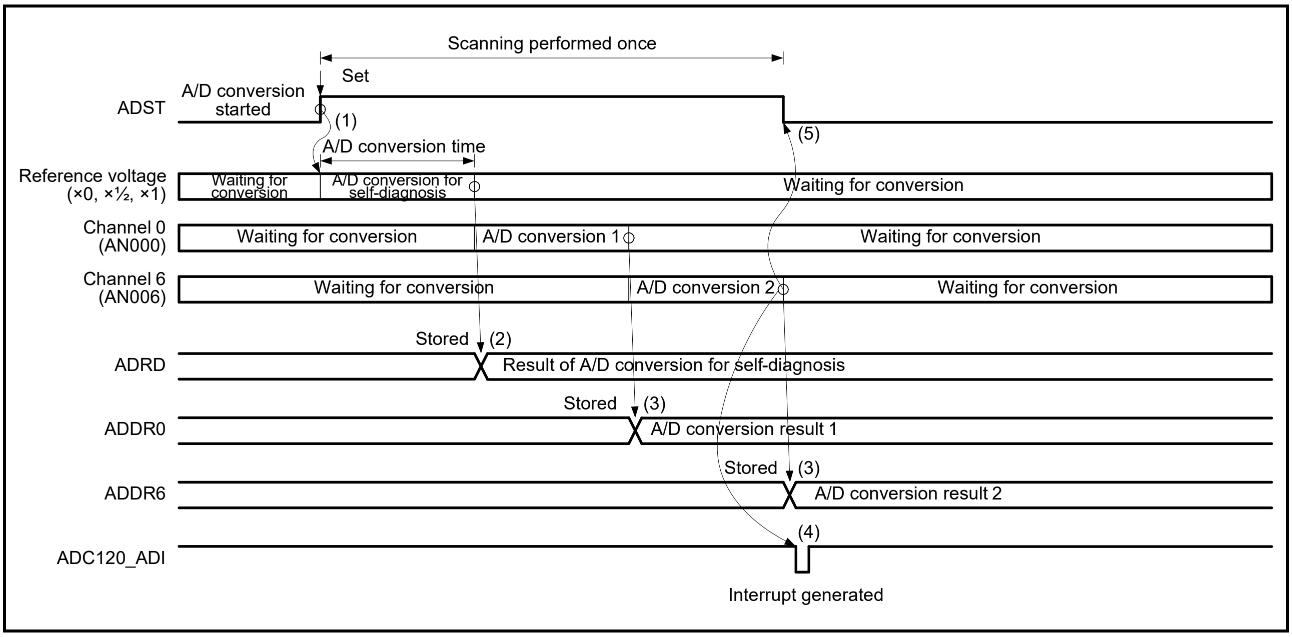


Figure 47.10 Example basic operation in single scan mode when AN000 and AN006 are selected with self-diagnosis

47.3.2.5 Channel selection and self-diagnosis with channel-dedicated sample-and-hold circuits and continuous sampling disabled

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, the sample-and-hold operation is performed first, and then A/D conversion is performed once for the reference voltage VREFH0 (unit 0) or VREFH (unit 1) (×0, ×1/2, or ×1) supplied to the ADC12. After that, A/D conversion is performed only once on the analog input of the selected channels.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
6. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state.

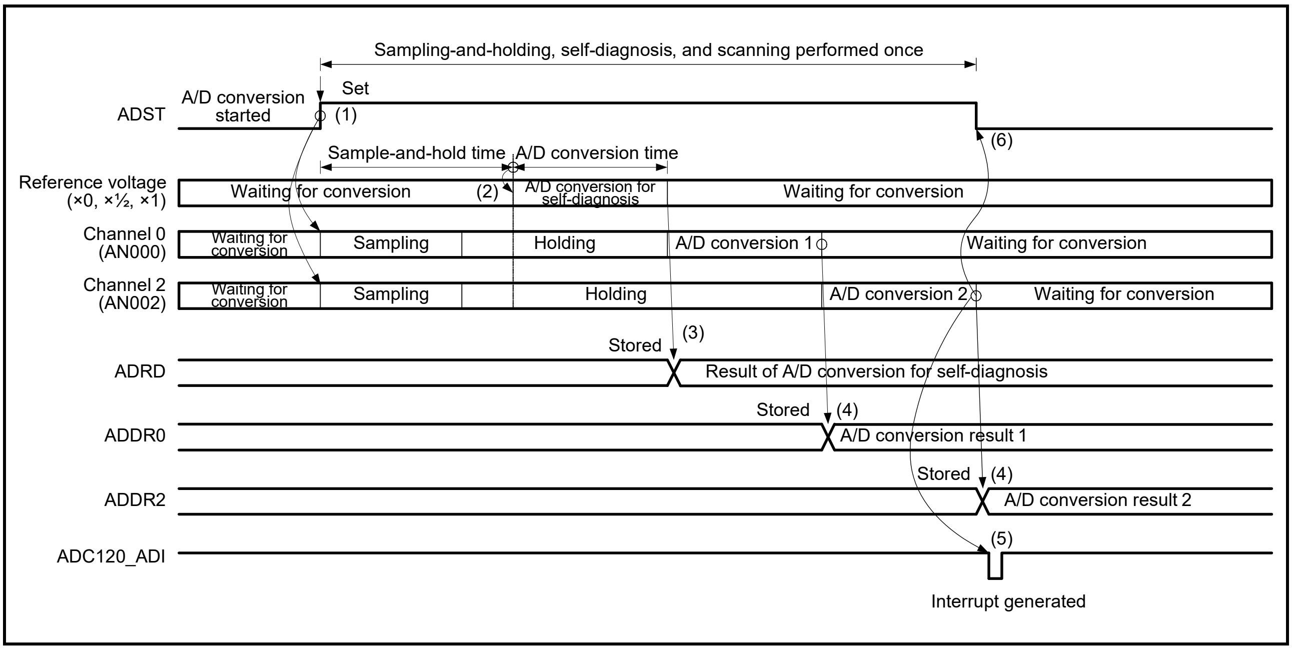


Figure 47.11 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000 and AN002 are selected with self-diagnosis, and continuous sampling is disabled

47.3.2.6 Channel selection and self-diagnosis with channel-dedicated sample-and-hold circuits and continuous sampling enabled

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, and this is followed by A/D conversion of the reference voltage VREFH0 (unit 0) or VREFH (unit 1) supplied to the ADC12. After that, A/D conversion is performed only once on the analog input of the selected channels.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) elapses after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion for self-diagnosis starts.
4. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
6. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).
8. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.

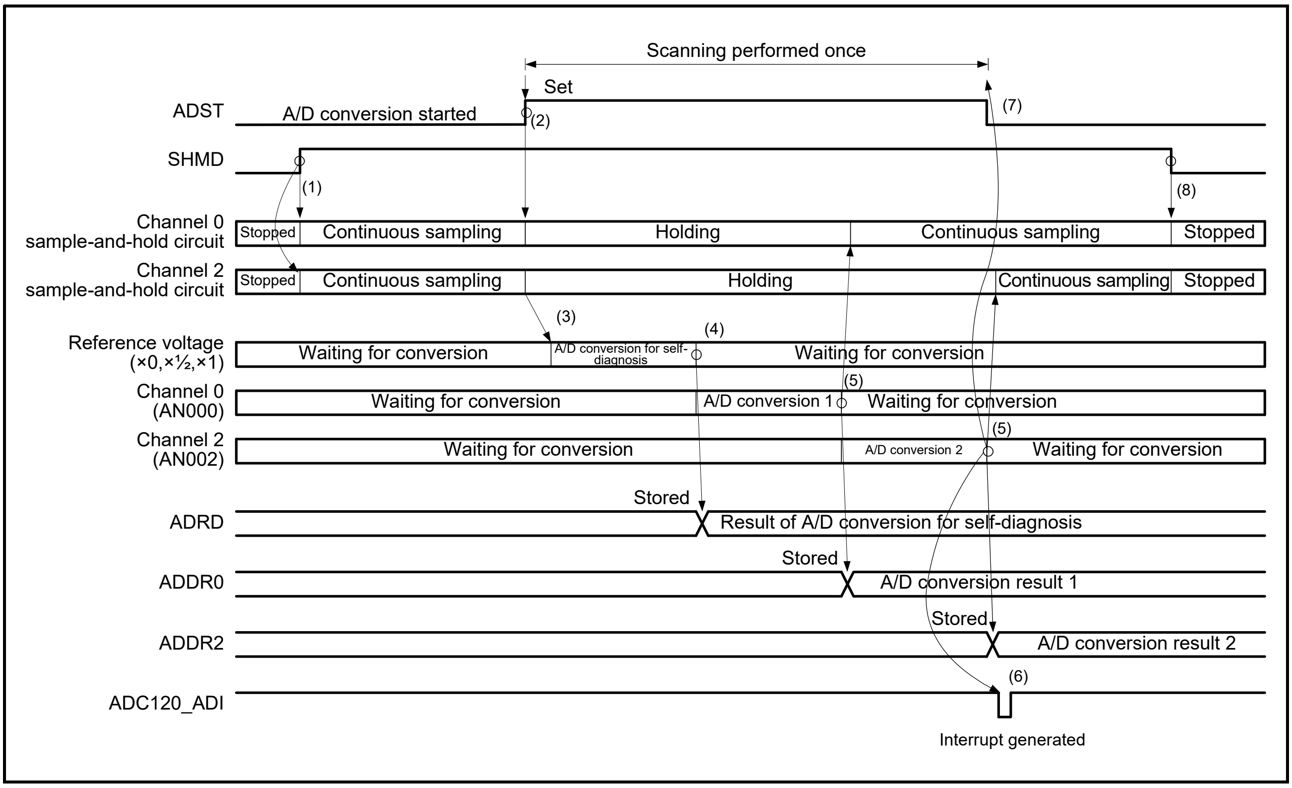


Figure 47.12 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000 to AN002 are selected with self-diagnosis, and continuous sampling is enabled

47.3.2.7 A/D conversion of temperature sensor output or internal reference voltage

When the channels and temperature sensor output or internal reference voltage are selected at the same time, A/D conversion is first performed on the analog input of the selected channels, and then A/D conversion is performed once on the temperature sensor output or internal reference voltage. When both temperature sensor output and internal reference voltage are selected, A/D conversion of the temperature sensor output and internal reference voltage is performed, in that order. With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

The operation is as follows:

1. When a software trigger, synchronous trigger (ELC), or asynchronous trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. On completion of A/D conversion on the channels, the result is stored in the associated A/D Data Register y (ADDRy), and then A/D conversion of the temperature sensor output starts.
3. On completion of A/D conversion of the temperature sensor output, the result is stored in the associated A/D Temperature Sensor Data Register (ADTSDR), and then A/D conversion of the internal reference voltage starts.
4. On completion of A/D conversion of the internal reference voltage, the result is stored in the associated A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
5. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 on completion of A/D conversion. Then the ADC12 enters a wait state.

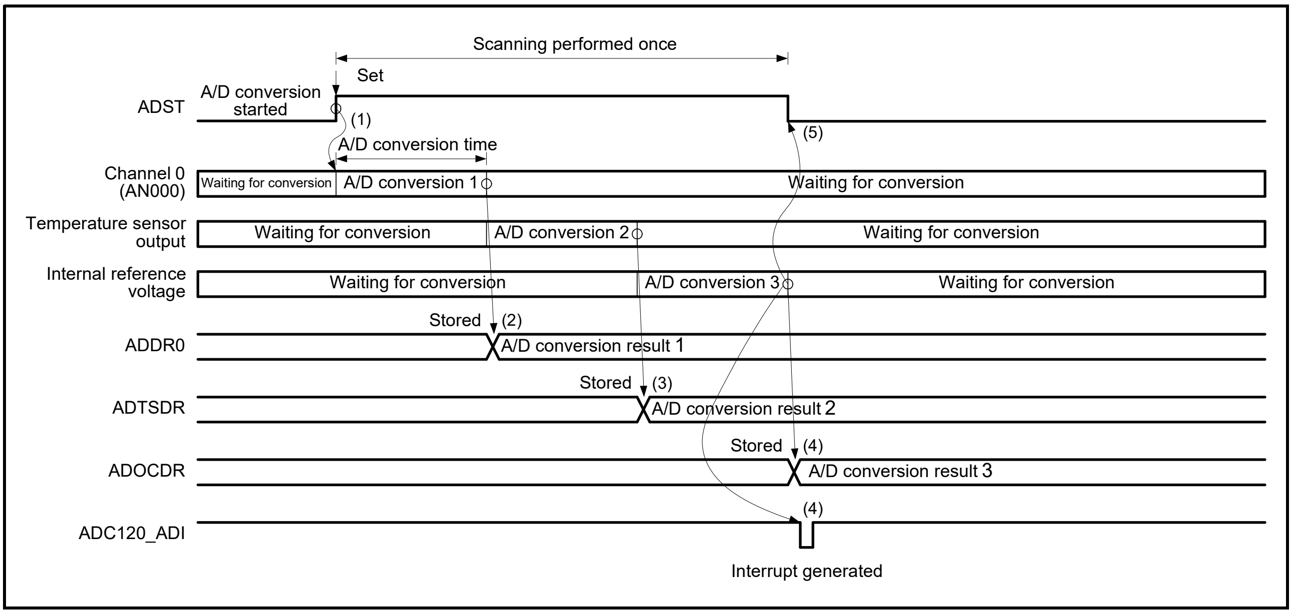


Figure 47.13 Example basic operation in single scan mode when AN000 and temperature sensor output or internal reference voltage are selected

47.3.2.8 A/D conversion in double-trigger mode

When double-trigger mode is selected in single scan mode, two rounds of single scan operation started by a synchronous trigger (ELC) are performed in sequence.

Deselect self-diagnosis, and set the addition/average mode select bits for both temperature sensor output (ADEXICR.TSSA) and internal reference voltage (ADEXICR.OCSA) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE to 1. When the ADCSR.DBLE is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In double-trigger mode, select a synchronous trigger (ELC) using the ADSTRGR.TRSA[5:0] bits, and in ADCSR, set the EXTRG bit to 0 and the TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (ELC), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
3. The ADST bit is automatically cleared to 0 and the ADC12 enters a wait state. An ADC12i_ADI (i = 0, 1) interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the result is stored in the A/D Data Duplexing Register (ADDBLDR), which is exclusively used in double-trigger mode.
6. An ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
7. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion completes. Then the ADC12 enters a wait state.

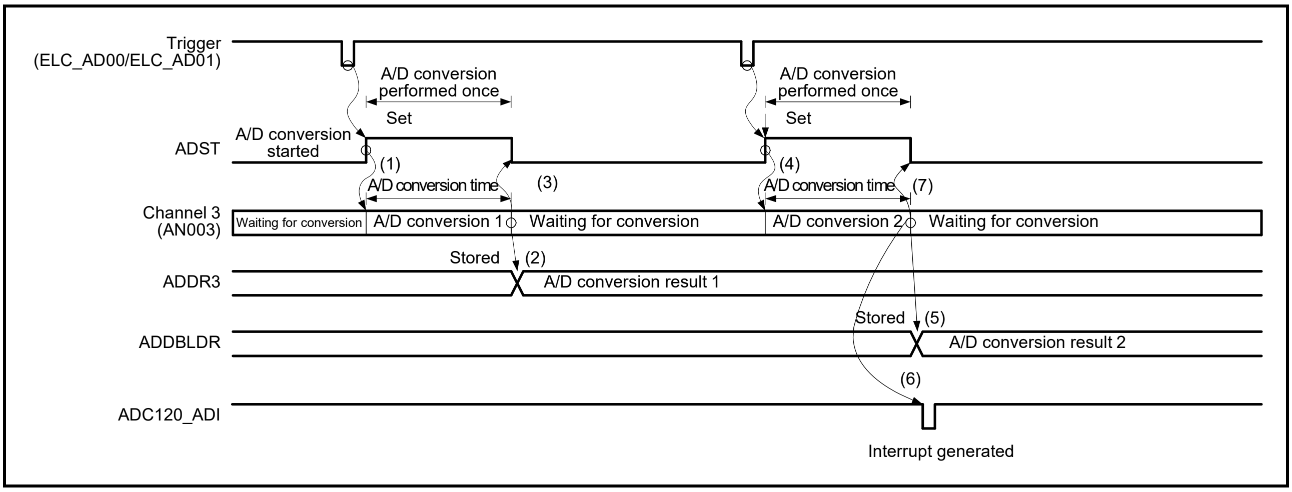


Figure 47.14 **Example operation in single scan mode when double-trigger mode is selected and AN003 is duplicated**

47.3.2.9 Extended operations when double-trigger mode is selected

When double-trigger mode is selected in single scan mode, and a synchronous trigger (ELC_AD00/ELC_AD01 (unit 0), ELC_AD10/ELC_AD11 (unit 1)) is selected as the trigger for the start of A/D conversion, two rounds of single scan operation are performed.

Deselect self-diagnosis, and set the addition/average mode select bits for both temperature sensor output (ADEXICR.TSSA and ADEXICR.TSSB) and internal reference voltage (ADEXICR.OCSA and ADEXICR.OCSB) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In extended double-trigger mode, select a synchronous trigger (ELC_AD00/ELC_AD01 (unit 0), ELC_AD10/ELC_AD11 (unit 1)) by setting the ADSTRGR.TRSA[5:0] bits to 0Bh, and in ADCSR, set the EXTRG bit to 0 and the TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (ELC_AD00/ELC_AD01 (unit 0), ELC_AD10/ELC_AD11 (unit 1)), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. When A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the trigger of ELC_ADi0 or ELC_ADi1 is input respectively (i=0, 1).
3. The ADCSR.ADST bit is automatically cleared to 0 and the ADC12 enters a wait state. An ADC12i_ADI (i = 0, 1) interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input (ELC_AD00/ELC_AD01 (unit 0), ELC_AD10/ELC_AD11 (unit 1)), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the result is stored in the A/D Data Duplexing Register (ADDBLDR) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the trigger of ELC_ADi0 or ELC_ADi1 is input respectively (i=0, 1).
6. An ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically cleared to 0 when A/D conversion completes. Then the ADC12 enters a wait state.

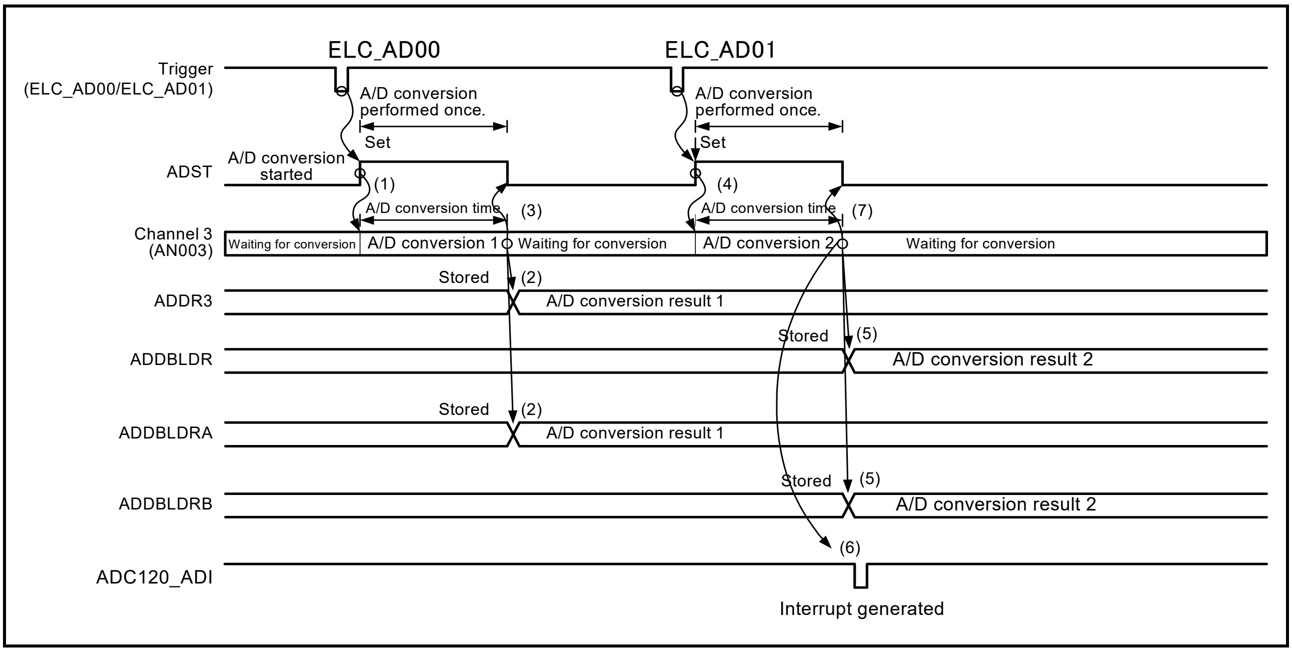


Figure 47.15 **Example extended operation in double-trigger mode (1) when duplication is selected for AN003 and ELC_AD00/ELC_AD01 is selected**

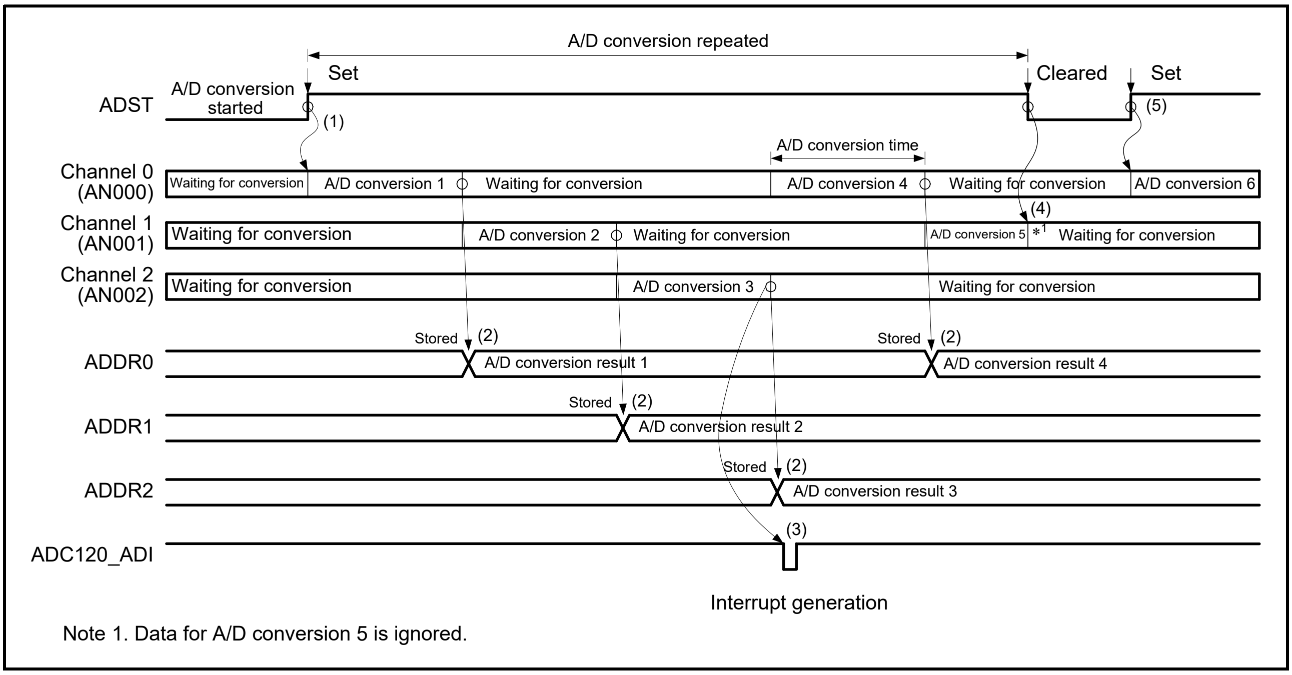
47.3.3 Continuous Scan Mode

47.3.3.1 Basic operation without channel-dedicated sample-and-hold circuits

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
3. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0,1) interrupt request is generated (no register setting). The ADC12 sequentially starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. The ADCSR.ADST bit is not automatically cleared, and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
5. When the ADST bit is then set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.



Note 1. Data for A/D conversion 5 is ignored.

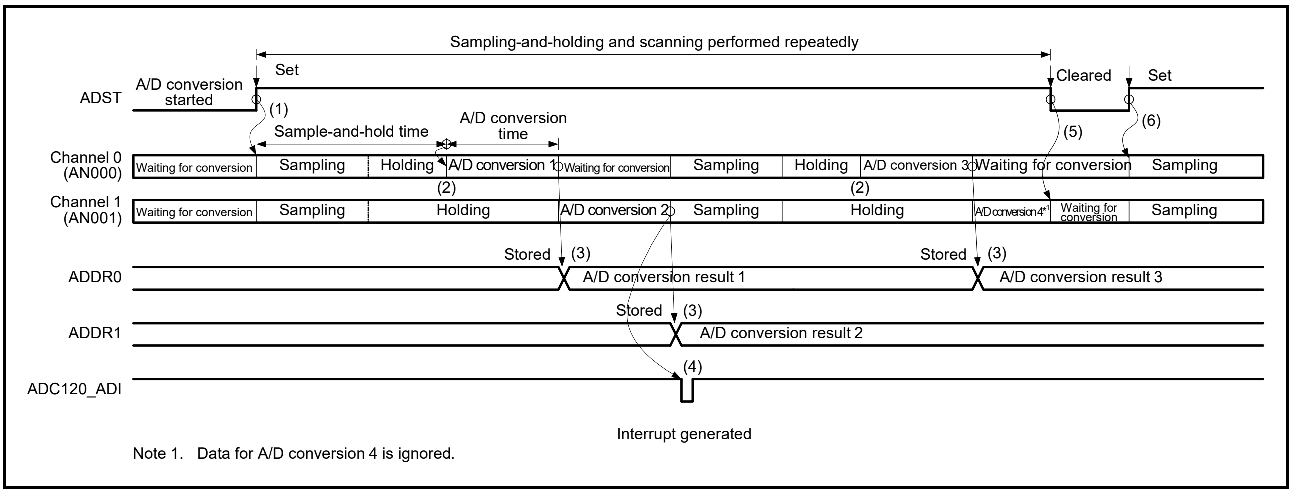
Figure 47.16 **Example basic operation in continuous scan mode when AN000 to AN002 are selected**

47.3.3.2 Basic operation with channel-dedicated sample-and-hold circuits and continuous sampling disabled

When the channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and-hold operation is first performed, and then A/D conversion is repeated on the analog input of all the specified channels. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[2:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
5. The ADST bit is not automatically cleared, and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all channels whose dedicated sample-and-hold circuit is to be used.



Note 1. Data for A/D conversion 4 is ignored.

Figure 47.17 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected

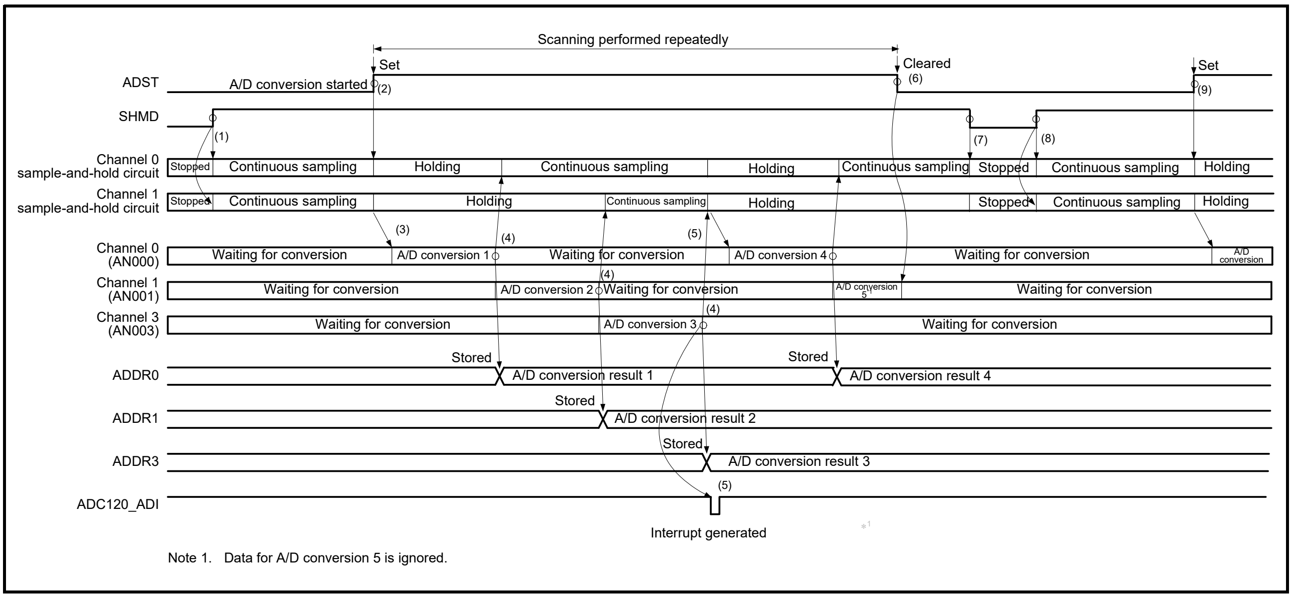
47.3.3.3 Basic operation with channel-dedicated sample-and-hold circuits and continuous sampling enabled

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, after which the analog inputs on all selected channels are A/D-converted as described in this section. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected in the ADSHCR.SHANS[2:0] bits.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) elapses after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting). Also, analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.
6. The ADCSR.ADST bit is not automatically cleared, and steps 3 to 5 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
7. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.
8. When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
9. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.

Note: If continuous scanning is performed when only the channels with the sample-and-hold circuits are selected, time for continuous sampling cannot be secured in the second and subsequent continuous scans. When continuous sampling by the channel-dedicated sample-and-hold circuits is enabled for continuous scanning, select one or more channels among AN003 to AN007 and AN016 to AN020, temperature sensor output, and internal reference voltage for unit 0, and AN103, AN105 to AN107 and AN116 to AN119, temperature sensor output, and internal reference voltage for unit 1, and set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).



Note 1. Data for A/D conversion 5 is ignored.

Figure 47.18 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001, and AN003 are selected, and continuous sampling is enabled

47.3.3.4 Channel selection and self-diagnosis without channel-dedicated sample-and-hold circuits

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0 (unit 0) or VREFH (unit 1) (×0, ×1/2, or ×1) supplied to the ADC12, and then A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input.
2. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting). At the same time, the ADC12 starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. The ADST bit is not automatically cleared, and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state.

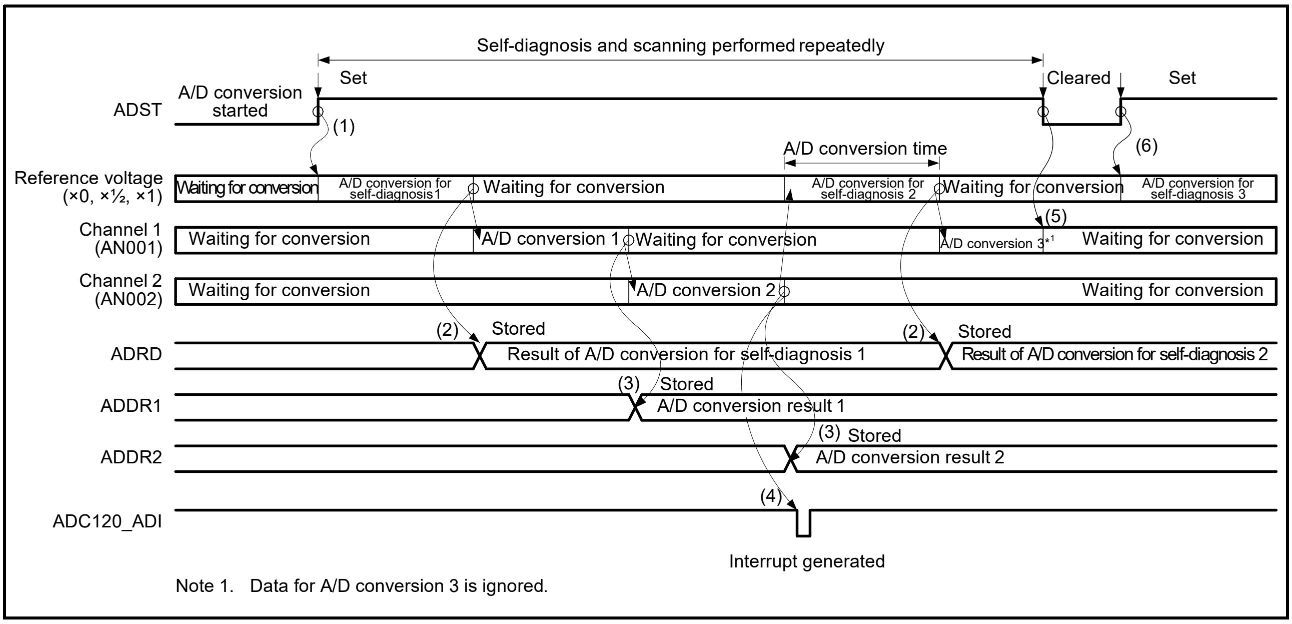


Figure 47.19 Example basic operation in continuous scan mode when AN001 and AN002 are selected with self-diagnosis

47.3.3.5 Channel selection and self-diagnosis with channel-dedicated sample-and-hold circuits and continuous sampling disabled

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and- hold operation is first performed, and then A/D conversion is performed for the reference voltage VREFH0 (unit 0) or VREFH (unit 1) (×0, ×1/2, or ×1) supplied to the ADC12, and then A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
6. The ADST bit is not automatically cleared, and steps 2 to 5 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
7. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all channels whose dedicated sample-and-hold circuit is to be used.

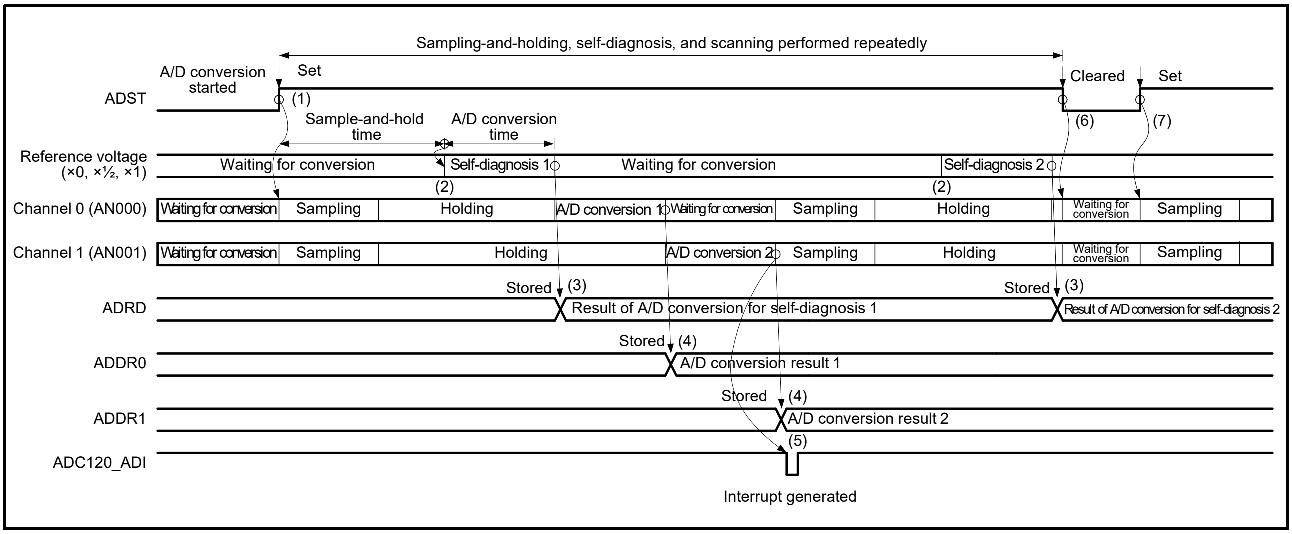


Figure 47.20 **Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected with self-diagnosis**

47.3.3.6 Channel selection and self-diagnosis with channel-dedicated sample-and-hold circuits and continuous sampling enabled

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operation is first performed, and this is followed by A/D conversion of the reference voltage VREFH0 (unit 0) or VREFH (unit 1) (×0, ×1/2, or ×1) supplied to the ADC12. After that, A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) elapses after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion for self-diagnosis starts.
4. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
6. When A/D conversion of all the selected channels completes, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting). Also, analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.
7. The ADCSR.ADST bit is not automatically cleared, and steps 3 to 6 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
8. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.
9. When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[2:0] bits start continuous sampling.
10. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.

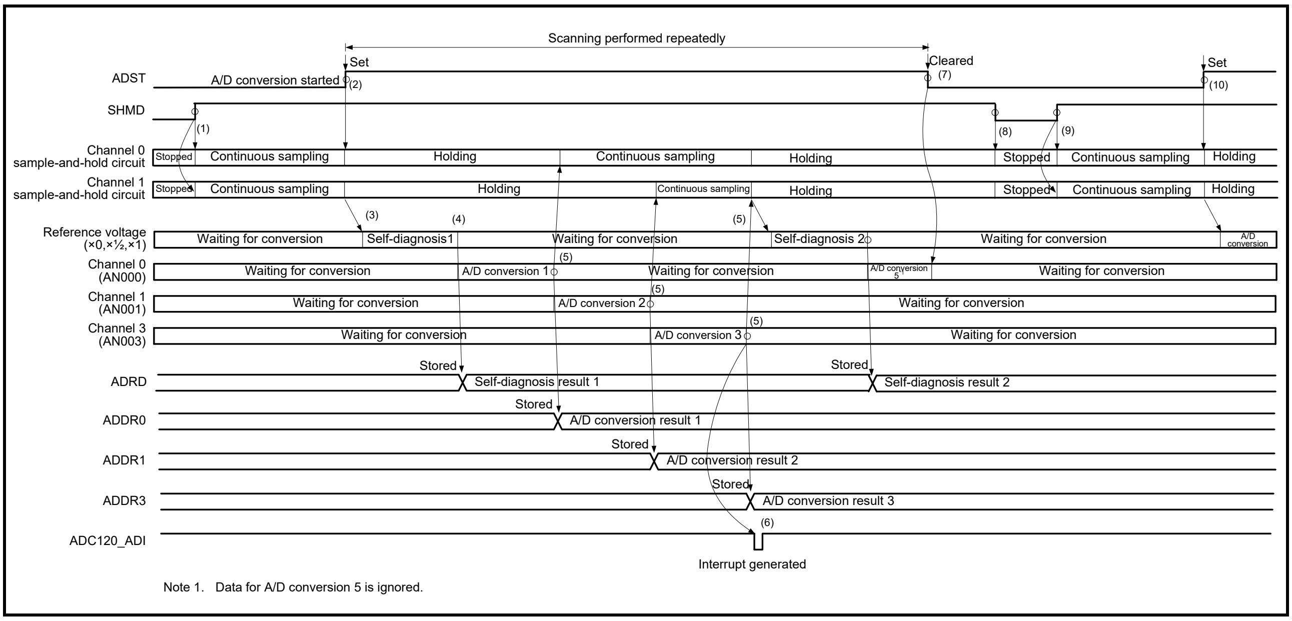


Figure 47.21 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001, and AN003 are selected with self-diagnosis, and continuous sampling is enabled

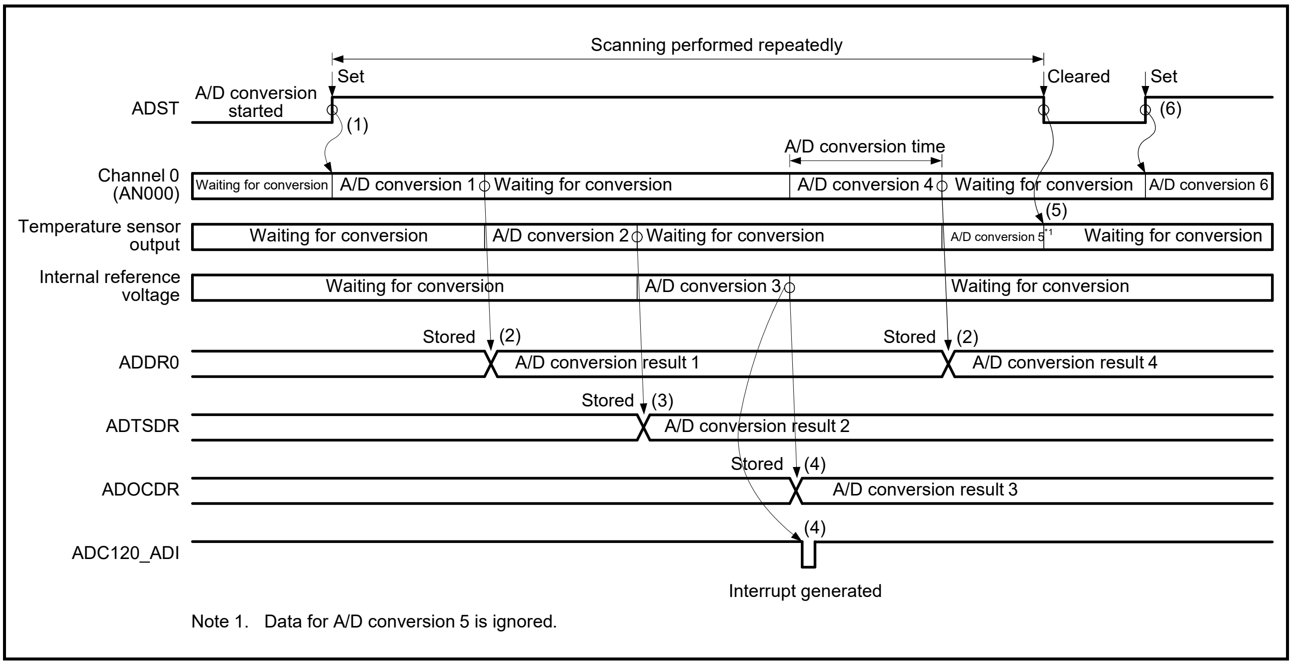
47.3.3.7 A/D conversion of temperature sensor output or internal reference voltage

When the channels and temperature sensor output or internal reference voltage are selected at the same time, A/D conversion is first performed on the analog input of the selected channels, and then A/D conversion of the temperature sensor output or internal reference voltage is repeated. When both temperature sensor output and internal reference voltage are selected, A/D conversion of the temperature sensor output and internal reference voltage is performed, in that order.

With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

The operation is as follows:

1. When a software trigger, synchronous trigger (ELC), or asynchronous trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. On completion of A/D conversion on the channels, the result is stored in the associated A/D Data Register y (ADDRy), and then A/D conversion of temperature sensor output starts.
3. On completion of A/D conversion of the temperature sensor output, the result is stored in the associated A/D Temperature Sensor Data Register (ADTSDR), and then A/D conversion of internal reference voltage starts.
4. On completion of A/D conversion of the internal reference voltage, the result is stored in the associated A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC12i_ADI interrupt request is generated. In addition, the ADC12 continuously starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the lowest number n.
5. The ADCSR.ADST bit is not cleared automatically, and steps 2 to 4 are repeated as long as this bit remains set to 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the lowest number n.



Note 1. Data for A/D conversion 5 is ignored.

Figure 47.22 Example basic operation in continuous scan mode when AN000 and temperature sensor output or internal reference voltage are selected

47.3.4 Group Scan Mode

47.3.4.1 Basic operation

In group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in Group A and Group B after scanning is started by a synchronous trigger (ELC). The scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers can be selected in the ADSTRGR.TRSA[5:0] bits for Group A and in the ADSTRGR.TRSB[5:0] bits for Group B. Use different triggers for Group A and Group B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger.

The Group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers and the ADEXICR.TSSA and OCSA bits, while the Group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers and the ADEXICR.TSSB and OCSB bits. Group A and Group B cannot use the same channels.

When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for Group A and Group B.

The following describes operation in group scan mode using a synchronous trigger from the ELC. In this example, the ELC_AD00 and ELC_AD01 (unit 0), and ELC_AD10 and ELC_AD11 (unit 1) triggers from the ELC are used to start conversion of Group A and Group B, respectively. Also, ELC_AD00 and ELC_AD01 (unit 0), and ELC_AD10 and ELC_AD11 (unit 1) are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of Group A is started by ELC_AD00 (unit 0) or ELC_AD10 (unit 1).
2. When Group A scanning completes, an ADC12i_ADI (i = 0, 1) interrupt is generated (no register setting).
3. Scanning of Group B is started by ELC_AD01 (unit 0) or ELC_AD11 (unit 1).
4. When Group B scanning completes, an ADC12i_GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (ADC12i_GBADI interrupt when scanning completion is enabled).

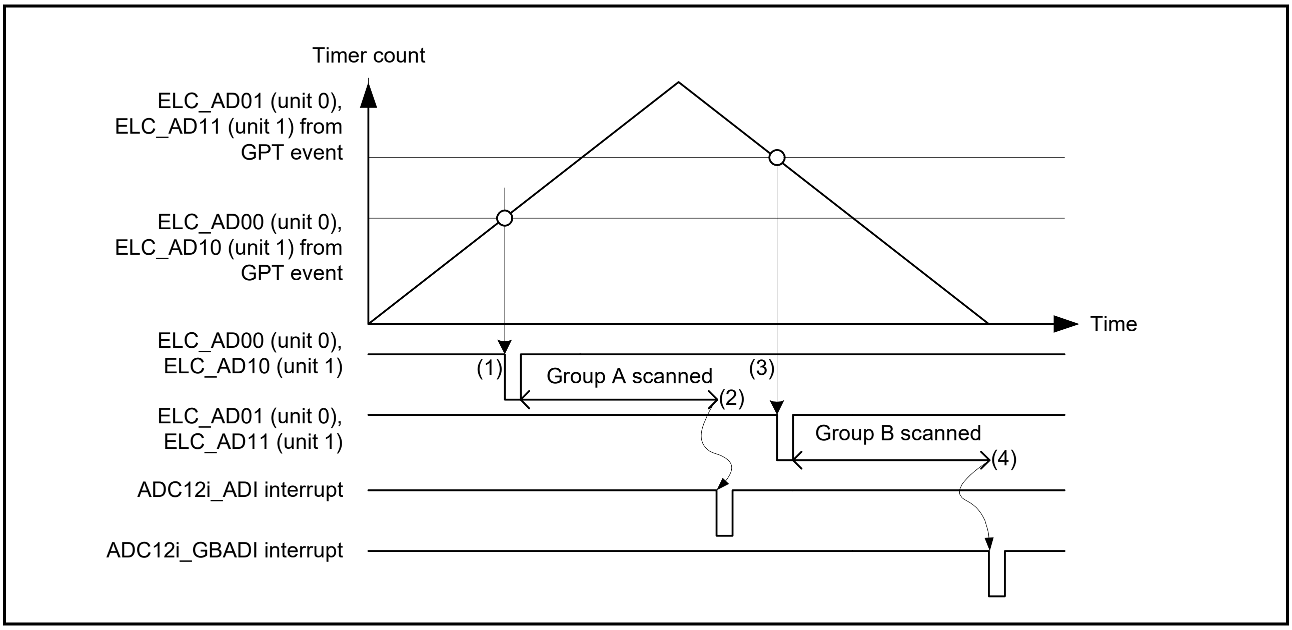


Figure 47.23 Example basic operation in group scan mode when synchronous triggers from the ELC are used

47.3.4.2 A/D conversion in double-trigger mode

When double-trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger (ELC) are performed as a sequence for Group A. For Group B, single scan operation started by a synchronous trigger (ELC) is performed once.

In group scan mode, the synchronous triggers can be selected in the TRSA[5:0] bits in ADSTRGR for Group A and in the TRSB[5:0] bits in ADSTRGR for Group B. Use different triggers for Group A and Group B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger or an asynchronous trigger (ADTRGn).

When ELC_AD00/ELC_AD01 (unit 0), ELC_AD10/ELC_AD11 (unit 1) is selected as the Group A synchronous triggers by setting the ADSTRGR.TRSA[5:0] bits to 0Bh, operation proceeds in extended double-trigger mode.

The channels to be A/D-converted are selected in the DBLANS[4:0] bits in the ADCSR register for Group A and in the ADANSB0 and ADANSB1 registers for Group B. The same channels cannot be selected for both groups.

When double-trigger mode is selected in group scan mode, set the A/D conversion select bits for both the temperature sensor output (ADEXICR.TSSA) and the internal reference voltage (ADEXICR.OCSA) to 0 (deselected). Self-diagnosis cannot be selected when double-trigger mode is selected in group scan mode.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the DBLE bit in ADCSR to 1.

The following describes operation in group scan mode with double-trigger mode using a synchronous trigger from the ELC. In this example, the ELC_AD00 and ELC_AD01 (unit 0), ELC_AD10 and ELC_AD11 (unit 1) triggers from the ELC are used to start conversion of Group A and Group B, respectively. Also, ELC_AD00 and ELC_AD01 (unit 0), ELC_AD10 and ELC_AD11 (unit 1) are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of Group B is started by the ELC_AD00 (unit 0) or ELC_AD10 (unit 1) trigger from the ELC.
2. When Group B scanning completes, an ADC12i_GBADI (i = 0, 1) interrupt is generated if the GBADIE bit in ADCSR is 1 (ADC12i_GBADI interrupt when scanning completion is enabled).
3. The first scan of Group A is started by the first ELC_AD01 (unit 0) or ELC_AD11 (unit 1) trigger.
4. When the first scan of Group A completes, the conversion result is stored in the associated A/D Data Register y (ADDRy); an ADC12i_ADI interrupt request is not generated, regardless of the ADIE bit setting in ADCSR.
5. The second scan of Group A is started by the second ELC_AD01 (unit 0) or ELC_AD11 (unit 1) trigger.

6. When the second scan of Group A completes, the conversion result is stored in ADDBLDR. An ADC12i_ADI interrupt is generated.

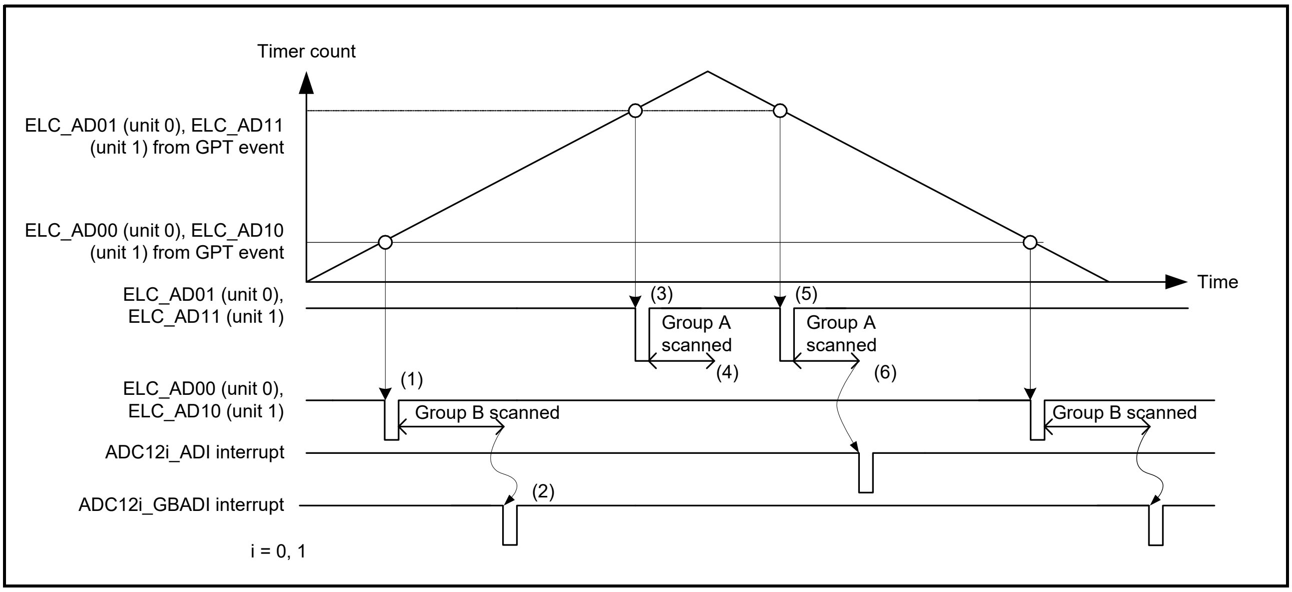


Figure 47.24 **Example basic operation in group scan mode with double-trigger mode when synchronous triggers from the ELC are used**

47.3.4.3 Operation with group A priority control

Setting the ADGSPCR.PGS bit to 1 in group scan mode makes operation proceed under Group A priority control. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure shown in Figure 47.25. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

In basic group scan mode, while A/D conversion is underway for group A or group B, input of the trigger for A/D conversion for the other group is ignored. Under Group A priority control, if a Group A trigger is input during A/D conversion for Group B, A/D conversion for Group B is discontinued and A/D conversion for Group A proceeds. If the setting in the ADGSPCR.GBRSCN bit is 0, the ADC12 enters wait state on completion of the A/D conversion for Group A. If the setting in the ADGSPCR.GBRSCN bit is 1, the ADC12 automatically restarts scanning for Group B from the head of the group after completion of the A/D conversion for Group A. Table 47.9 summarizes operations in response to the input of a trigger during A/D conversion with the settings in the ADGSPCR.GBRSCN bit.

Scan operations in Group A or Group B are the same in single scan mode. Additionally, single scanning continues to proceed if the ADGSPCR.GBRP bit is set to 1 during scanning operations for Group B.

For the trigger settings in group scan mode, select a synchronous trigger for Group A using the ADSTRGR.TRSA[5:0] bits, and select a synchronous trigger for Group B, different from that of Group A, using the ADSTRGR.TRSB[5:0] bits. Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting the ADGSPCR.GBRP bit to 1.

Additionally, as targets for A/D conversion, select channels for Group A using the ADANSA0 and ADANSA1 registers and the ADEXICR.TSSA and OCSA bits, and for Group B, select channels different from those for Group A using the ADANSB0 and ADANSB1 registers and the ADEXICR.TSSB and OCSB bits.

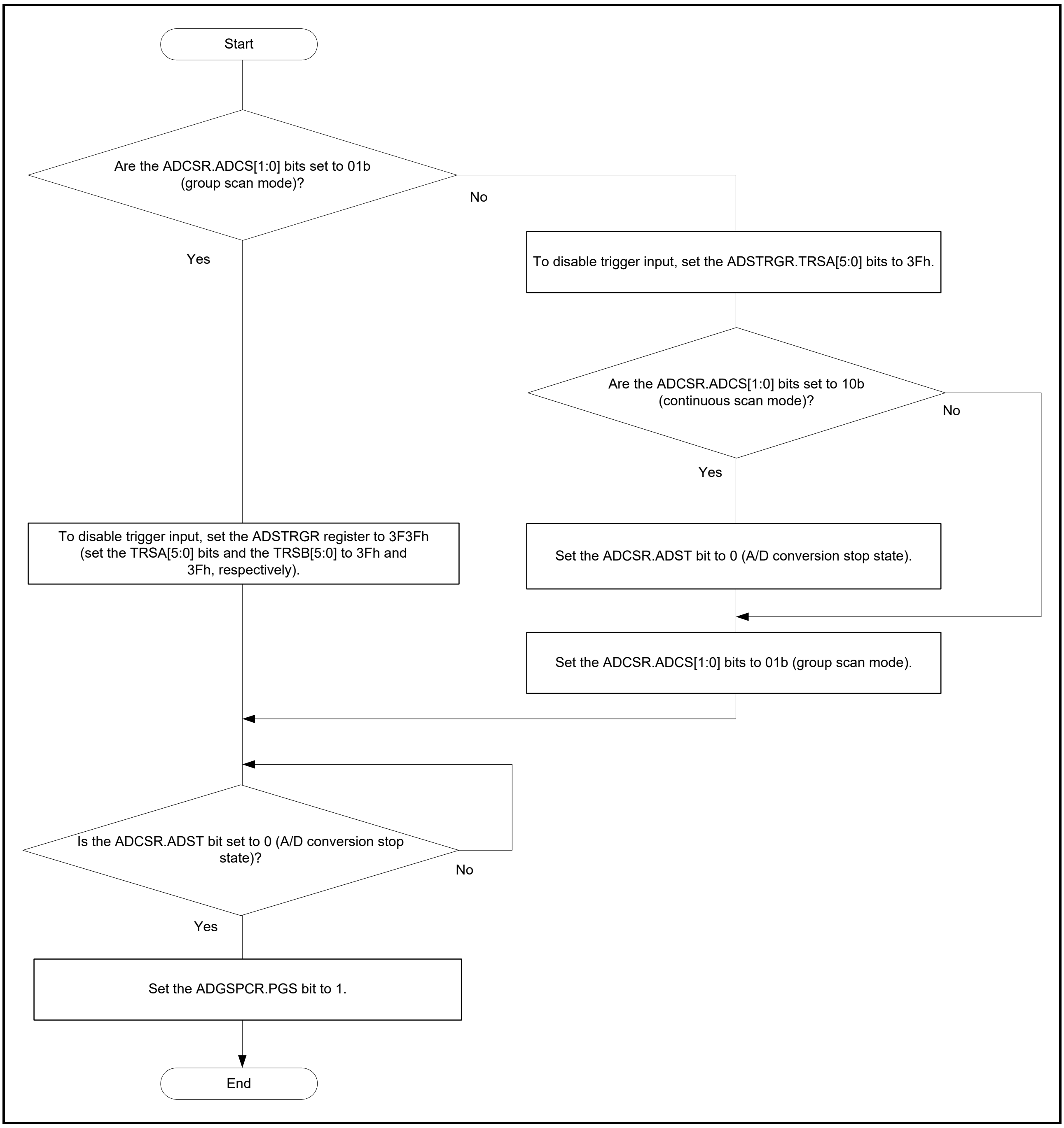


Figure 47.25 Flow for ADGSPCR.PGS bit setting

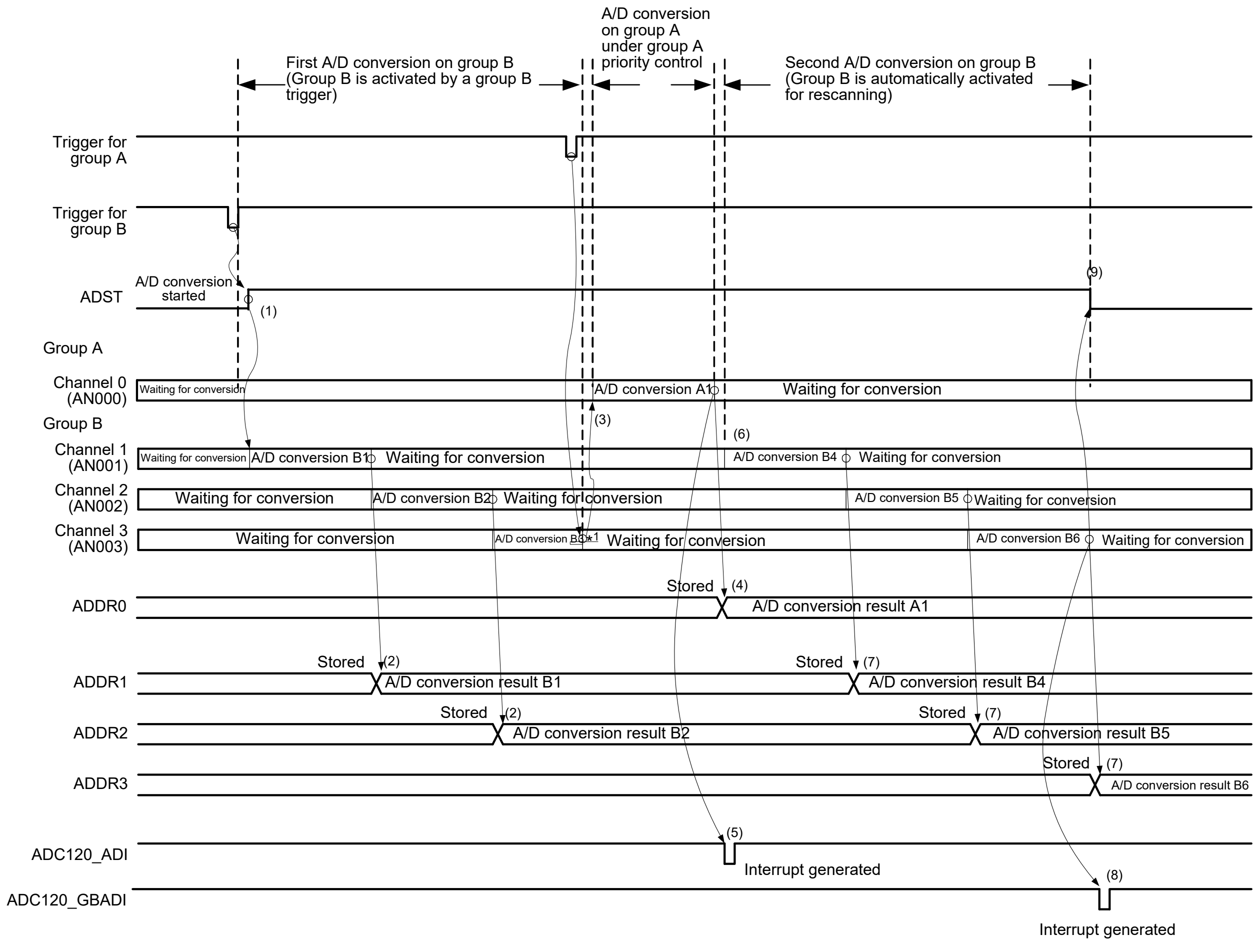
Table 47.9 Control of A/D conversion operations based on the ADGSPCR.GBRSCN bit settings

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for Group A is in progress	Input of Group A trigger	Trigger input is invalid	Trigger input is invalid
	Input of Group B trigger	Trigger input is invalid	Group B is converted after Group A conversion completes
When A/D conversion for Group B is in progress	Input of Group A trigger	Group B conversion stops and Group A conversion starts	• Group B conversion stops and Group A conversion starts • Group B Conversion starts after Group A conversion completes
	Input of Group B trigger	Trigger input is invalid	Trigger input is invalid

The following describes the operations in group scan mode under Group A priority control (ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 0 is selected for Group A and channels 1 to 3 are selected for Group B.

The operation is as follows:

1. When input of a trigger for Group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion, the result is stored in the associated A/D Data Register y (ADDRy).
3. When a group A trigger is input while A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1, A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n. If A/D conversion are not complete when conversion of group B is interrupted, A/D conversion result is not stored in the A/D Data Register (ADDRy).
4. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
5. An ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
6. A/D conversion for the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers restarts in order from the channel with the smallest number n with the ADCSR.ADST bit remains 1.
7. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
8. An ADC12i_GBADI interrupt request is generated if the setting in the ADCSR.GBADIE bit is 1 (ADC12i_GBADI interrupt when Group B scanning completion is enabled).
9. The ADCSR.ADST bit is automatically cleared and ADC12 enters the wait state when A/D conversion completes.



Note 1. Converted data from A/D conversion B3 is ignored.

Figure 47.26 Example operation with Group A priority control (1), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

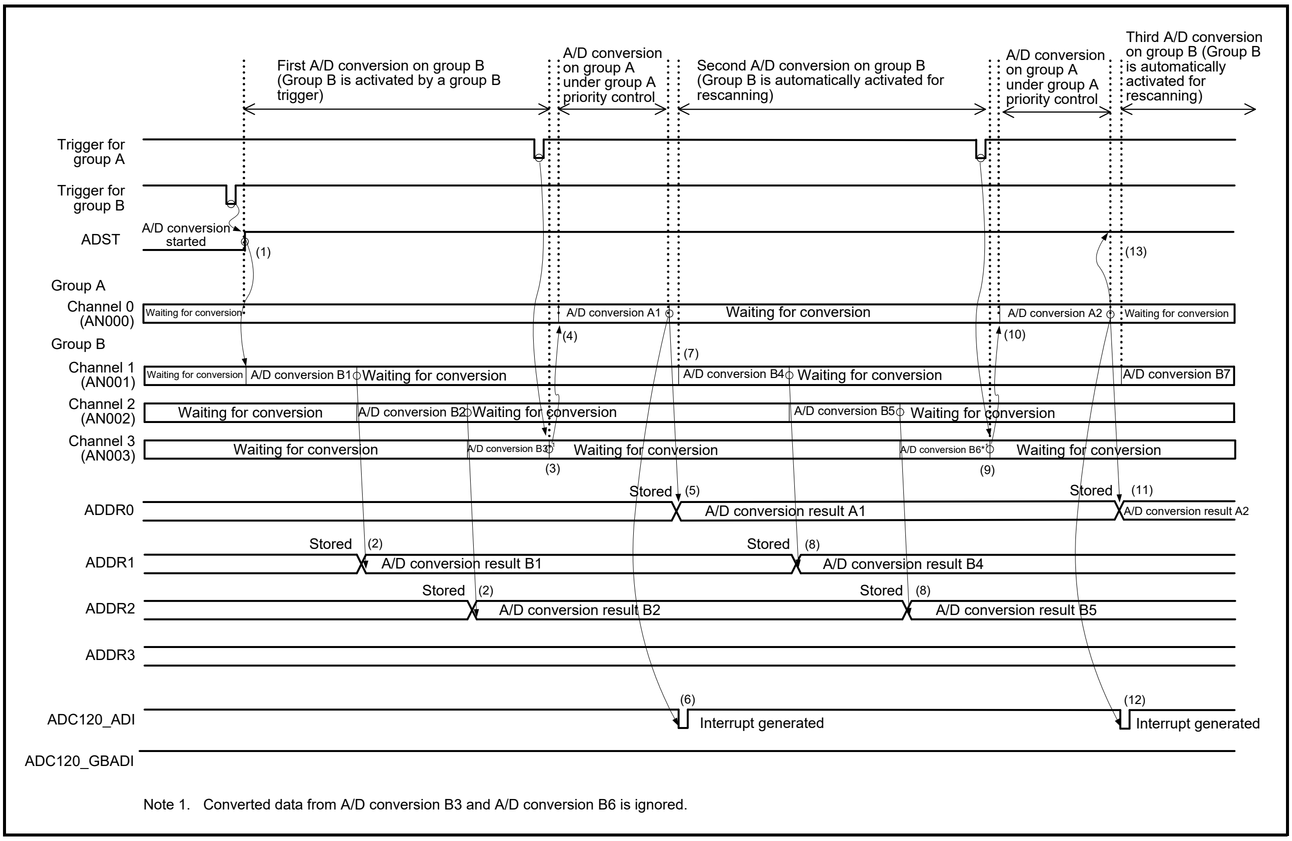
This section provides an example operation when a Group A trigger is input again during rescanning operation on Group B. In this example, channel 0 is selected for Group A and channels 1 to 3 are selected for Group B when operation on Group A is given priority (ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0).

The operation is as follows:

1. When a Group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels in Group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
3. When a group A trigger is input while A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. If A/D conversion are not complete when the conversion of group B is interrupted, A/D conversion result is not stored in the A/D Data Register (ADDRy).
4. A/D conversion for the ANn Group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
5. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
6. An ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
7. If the ADGSPCR.GBRSCN bit is 1 when the A/D conversion of group A completes, the ADCSR.ADST bit remains

1 and group B is rescanned. A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the smallest number n.

8. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
9. If a group A trigger is input during A/D conversion on group B for rescanning, the ADCSR.ADST bit remains 1 and the ongoing A/D conversion on group B is discontinued.
10. The ADCSR.ADST bit is set to 1 automatically, and A/D conversion for the ANn Group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
11. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
12. An ADC12i_ADI interrupt request is generated (no register setting).
13. If the ADGSPCR.GBRSCN bit is 1 when A/D conversion of group A are complete, the ADCSR.ADST bit remains 1 and group B is rescanned. A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the smallest number n.
14. If a Group A trigger is input during A/D conversion on Group B for rescanning, steps 9 to 13 are repeated. If a Group A trigger is not input, the ADCSR.ADST bit is cleared automatically on completion of A/D conversion on Group B and ADC12 enters a wait state.



Note 1. Converted data from A/D conversion B3 and A/D conversion B6 is ignored.

Figure 47.27 Example operation with Group A priority control (2), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

This section provides an example of a rescanning operation in which a Group B trigger is input during A/D conversion on Group A. In this example, channels 1 to 3 are selected for Group A and channel 0 is selected for Group B when operation on Group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

The operation is as follows:

1. When input of a trigger for Group A sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest

number n.

2. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
3. If a Group B trigger is input during A/D conversion on Group A, A/D conversion on Group B can be performed after the A/D conversion on Group A completes. (However, if Group A triggers are input continuously, the scan operation on Group B is canceled by Group A and is not performed.)
4. On completion of the A/D conversion on Group A, an ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
5. On completion of group A conversion, the ADCSR.ADST bit remains 1 and group B is rescanned. Next, A/D conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
6. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
7. On completion of the rescanning operation on Group B, an ADC12i_GBADI interrupt request is generated if the setting in the ADCSR.GBADIE bit is 1 (ADC12i_GBADI interrupt when scanning completion is enabled).
8. The ADCSR.ADST bit is automatically cleared and ADC12 enters the wait state when A/D conversion completes.

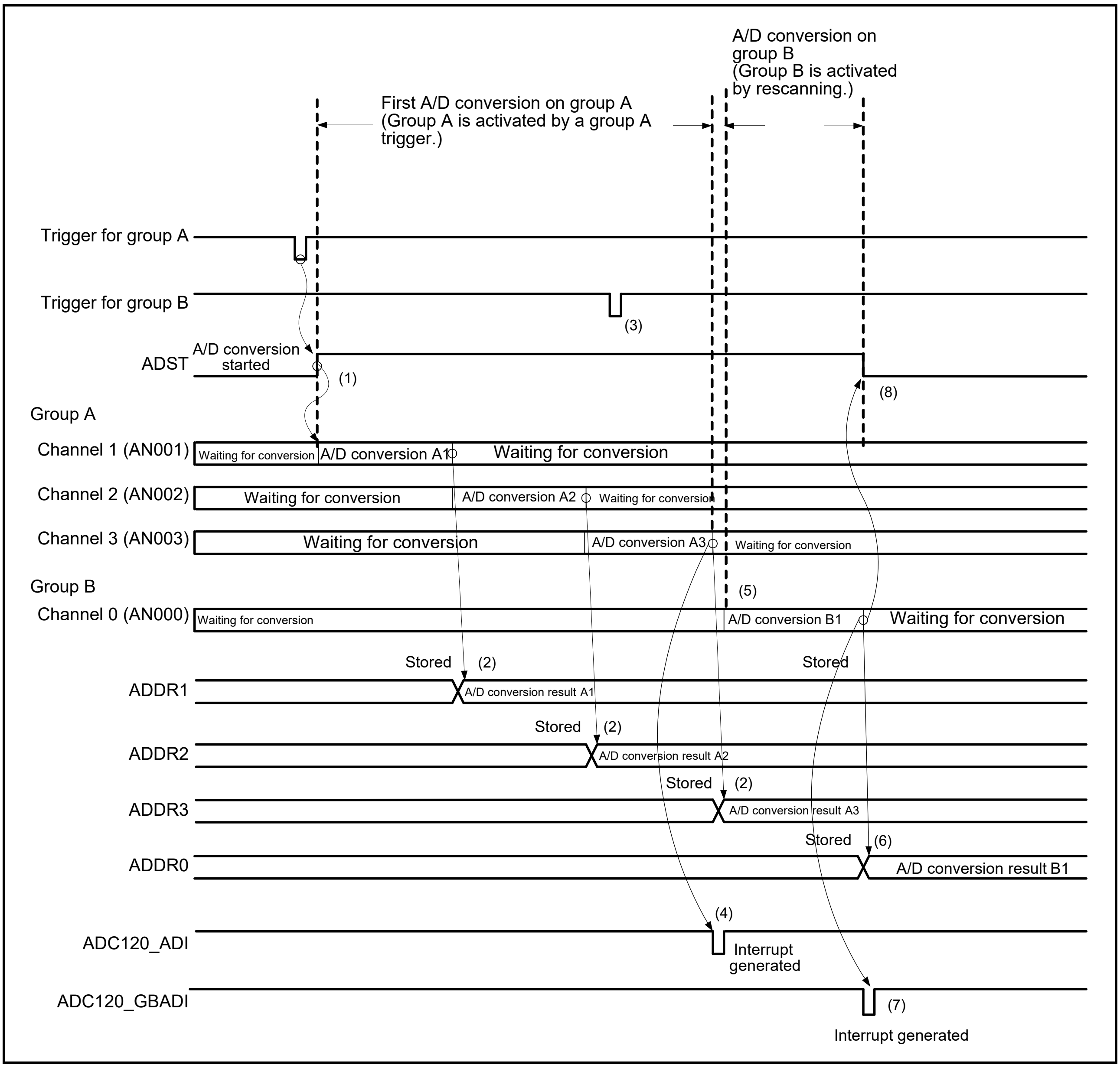


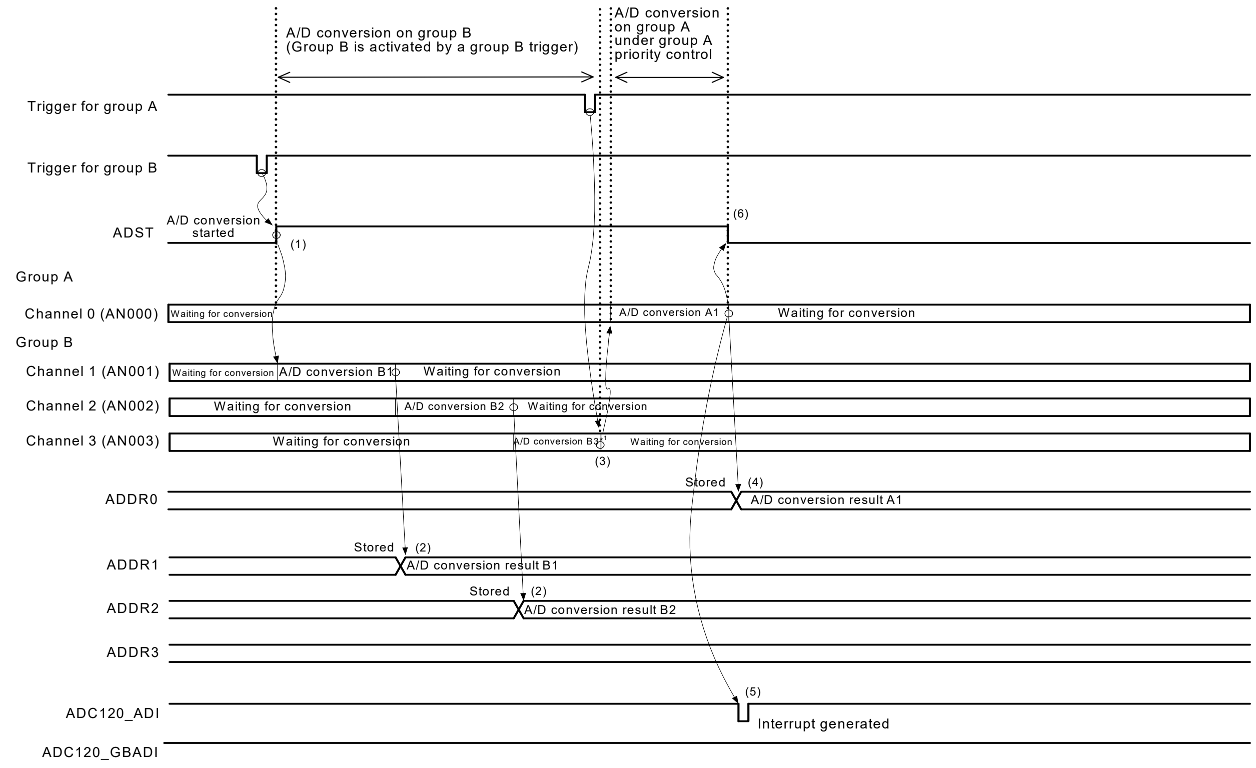
Figure 47.28 Example operation with Group A priority control (3), when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

This section provides an example of operation under Group A priority control in which channel 0 is selected for Group A and channels 1 to 3 are selected for Group B (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0).

The operation is as follows:

1. When input of a trigger for Group B sets the ADCSR.ADST bit to 1 (A/D conversion start), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
3. If a group A trigger is input while A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. Next, A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
4. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).

5. An ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
6. The ADCSR.ADST bit is automatically cleared and ADC12 enters the wait state when A/D conversion completes.



Note 1. Converted data of A/D conversion B3 is ignored.

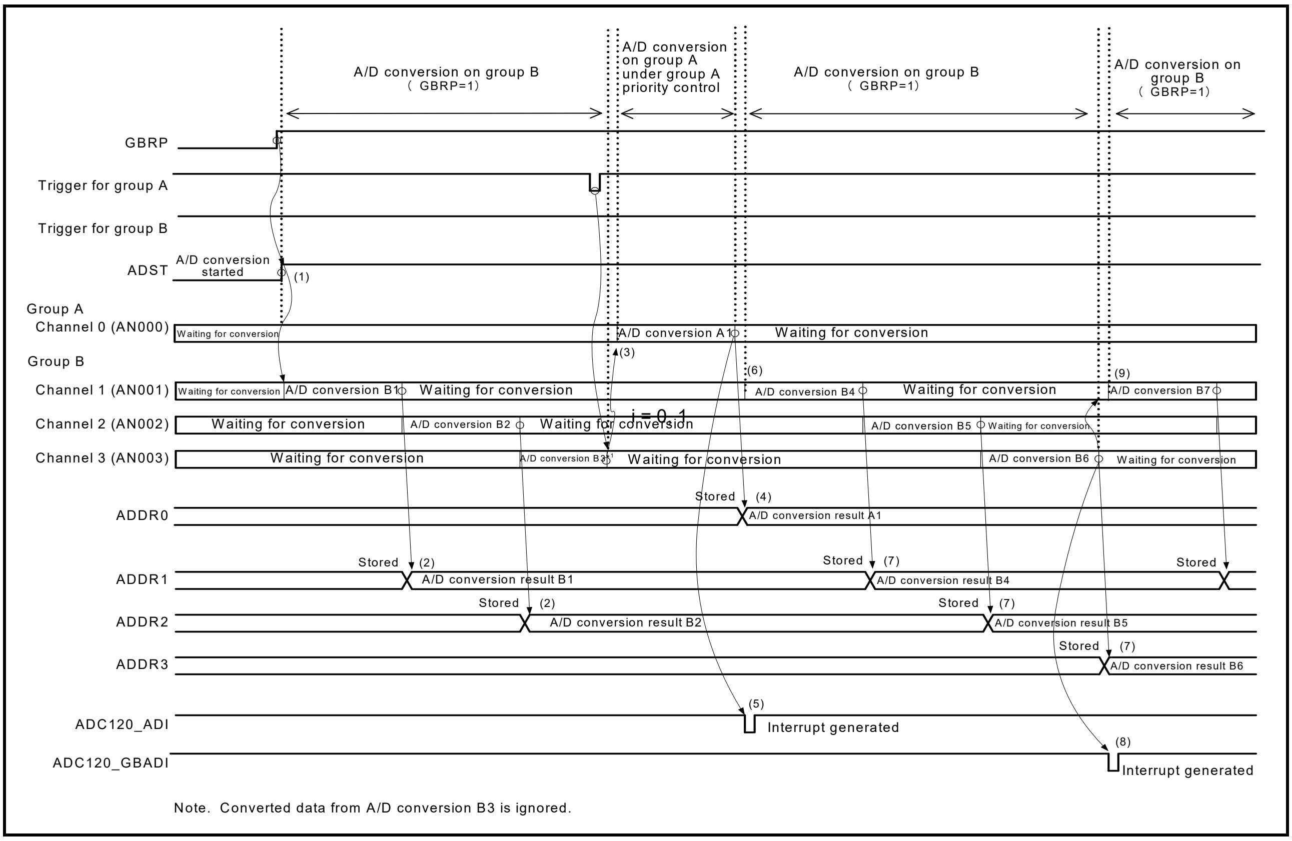
Figure 47.29 Example operation with Group A priority control (4), when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0

This section provides an example of operation under Group A priority control in which channel 0 is selected for Group A and channels 1 to 3 are selected for Group B (ADGSPCR.GBRP = 1).

The operation is as follows:

1. The ADCSR.ADST bit is set to 1 (A/D conversion start) when ADGSPCR.GBRP is set to 1, and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
3. If a group A trigger is input while A/D conversion for group B is in progress, and A/D conversion for group B is discontinued with the ADCSR.ADST bit remains 1. Next, A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
4. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
5. An ADC12i_ADI (i = 0, 1) interrupt request is generated (no register setting).
6. A/D conversion for the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers restarts in order from the channel with the smallest number n and with the ADCSR.ADST bit remains 1.
7. On completion of A/D conversion on a single channel, the result is stored in the associated A/D Data Register y (ADDRy).
8. An ADC12i_GBADI interrupt request is generated if the setting in the ADCSR.GBADIE bit is 1.

9. A/D conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n. Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. Follow the procedure for clearing the ADCSR.ADST bit operation by software, shown in Figure 47.40, to force A/D conversion to stop while ADGSPCR.GBRP = 1.



Note. Converted data from A/D conversion B3 is ignored.

Figure 47.30 Example operation with Group A priority control (5), when ADGSPCR.GBRP = 1

47.3.5 Compare Function for Windows A and B

47.3.5.1 Compare function windows A and B

The compare function compares a reference value with the A/D conversion result. The reference value can be set for Window A and Window B independently. When the compare function is in use, the self-diagnosis function and double-trigger mode cannot be used. The main differences between Window A and Window B are their different interrupt output signals and the constraint on Window B of only one selectable channel.

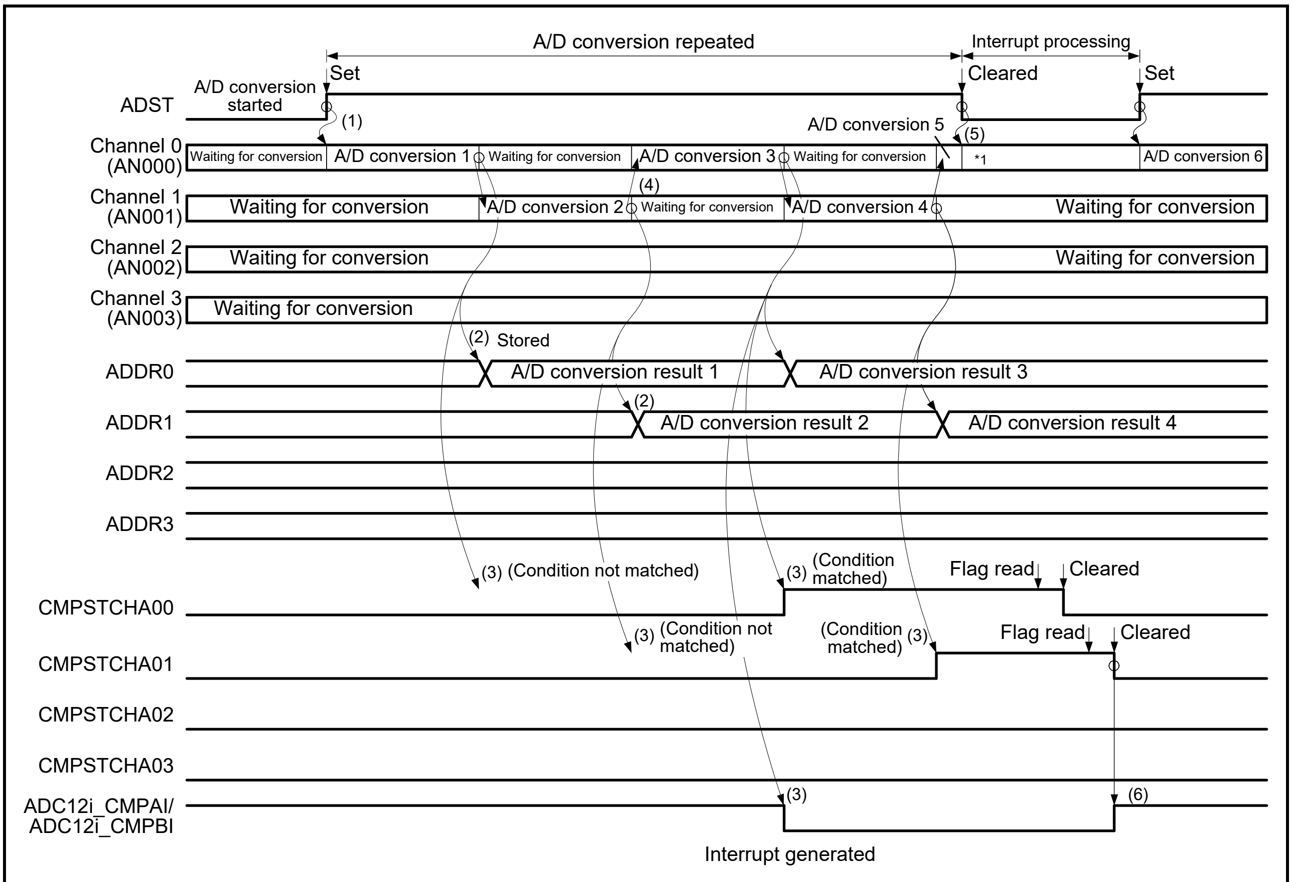
This section provides an example operation that combines continuous scan mode and the compare function.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger (ELC) or an asynchronous trigger, A/D conversion starts for the selected channels, temperature sensor, and internal reference voltage.
2. On completion of A/D conversion, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE = 1, if bits in the ADCMPANSRy register or the ADCMPANSER register are set for Window A, the A/D conversion result is compared with the set ADCMPDR0/1 register value. When ADCMPCR.CMPBE = 1, if bits in the ADCMPBNSR register are set for Window B, the A/D conversion result is compared with the ADWINULB/ADWINLLB register setting.
3. As a result of the comparison, when Window A meets the condition set in ADCMPLR0/1 or ADCMPLER, the Compare Window A Flag (ADCMPSR0.CMPSTCHA0n, ADCMPSR1.CMPSTCHA1n, ADCMPSER.CMPSTTSA, or ADCMPSER.CMPSTOCA) sets 1. At this time, if the ADCMPCR.CMPAIE bit is 1,

an ADC12i_CMPAI (i = 0, 1) interrupt request is generated. In the same way, when Window B meets the condition set in ADCMPBNSR.CMPLB, the Compare Window B Flag (ADCMPBSR.CMPSTB) sets to 1. At this time, if the ADCMPCR.CMPBIE bit is 1, an ADC12i_CMPBI interrupt request is generated.

4. On completion of all selected A/D conversions and comparisons, scan restarts.
5. After the ADC12i_CMPAI and ADC12i_CMPBI interrupts are accepted, the ADCSR.ADST bit is set to 0 (A/D conversion stop) and processing is performed for channels for which the compare flag is set to 1.
6. When all compare flags of Window A are cleared, the ADC12i_CMPAI interrupt request is canceled. In the same way, when all compare flags of Window B are cleared, the ADC12i_CMPBI interrupt request is reset. To perform comparison again, restart the A/D conversion.



Note 1. Data on conversion is ignored.
i = 0, 1

Figure 47.31 Example of compare function operation, when AN000 to AN003 are compared

47.3.5.2 Event output of compare function

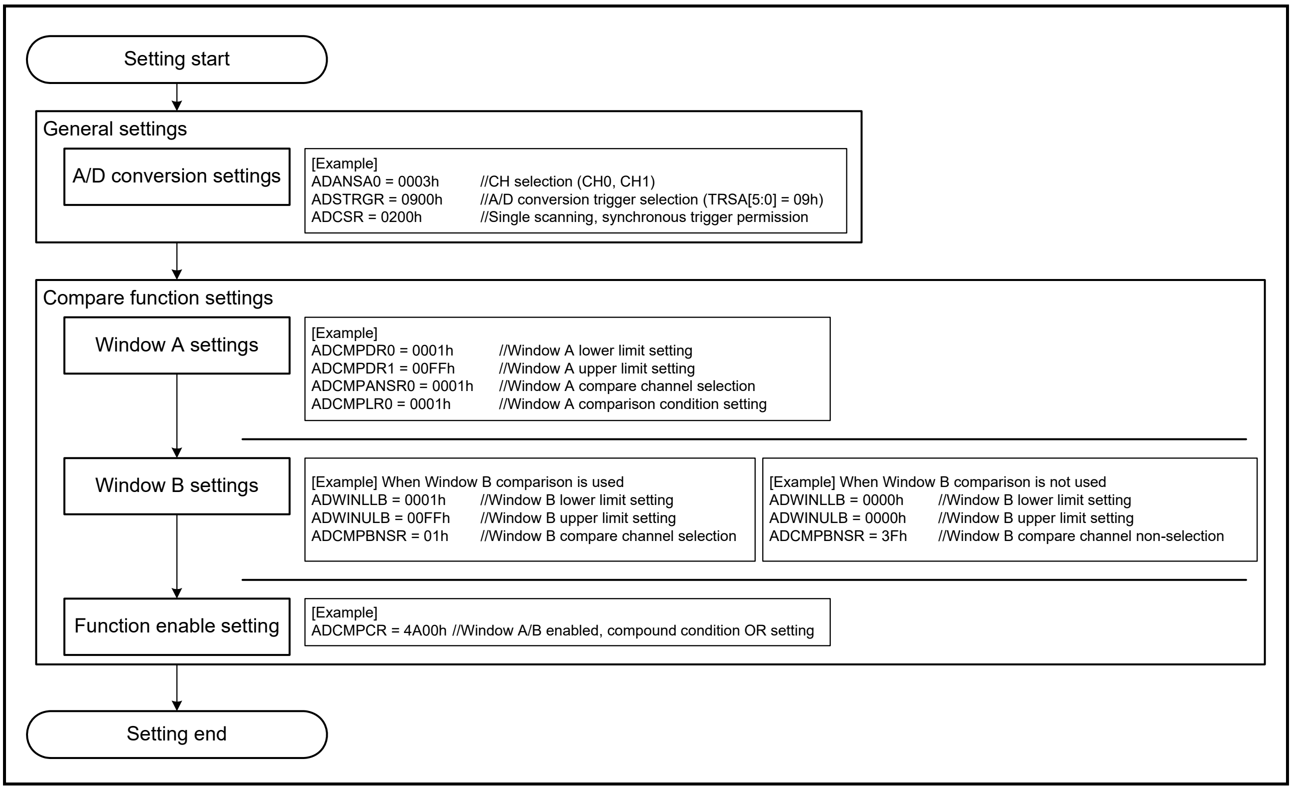
The event output of compare function specifies the upper reference voltage value for Window A and the lower reference voltage value for Window B, compares the A/D-converted value of the selected channel with the upper and lower reference voltage values, and then outputs the ADC12i_WCMPM and ADC12i_WCMPUM (i = 0, 1) events based on the event conditions (A OR B, A AND B, A XOR B) and comparison results of Window A and Window B.

If more than one channel is selected for Window A, and even one channel in Window A meets the comparison condition, the comparison result of Window A becomes met. When using this function, perform A/D conversion in single scan mode. Any channels from AN000 to AN007 and AN016 to AN020 (unit 0), and AN100 to AN103, AN105 to AN107, and AN116 to AN119 (unit 1), internal reference voltage, and temperature sensor output are selectable for Window A.

This section provides the setting procedure and example when using event output of compare function.

To set up event output for the compare function:

1. Confirm that the value in the ADCSR.ADCS bits is 00b (single scan mode).
2. Select the channel for Window A in ADCMPANSR0/1 and ADCMPANSER. Set window comparison conditions in the ADCMPLR0/1 and ADCMPLER registers. Set the upper and lower reference values in the ADCMPDR0 and 1 registers.
3. Select the channel and comparison conditions for Window B in the ADCMPBNSR register, and set the upper and lower reference values in the ADWINULB and ADWINLLB registers.
4. Set composite conditions for Window A/B, Window A/B operation enable, and interrupt output enable in ADCMPCR.



```
[Example]
ADANSA0 = 0003h        //CH selection (CH0, CH1)
ADSTRGR = 0900h        //A/D conversion trigger selection (TRSA[5:0] = 09h)
ADCSR = 0200h          //Single scanning, synchronous trigger permission
```

```
[Example]
ADCMPDR0 = 0001h       //Window A lower limit setting
ADCMPDR1 = 00FFh       //Window A upper limit setting
ADCMPANSR0 = 0001h     //Window A compare channel selection
ADCMPLR0 = 0001h       //Window A comparison condition setting
```

```
[Example] When Window B comparison is used
ADWINLLB = 0001h       //Window B lower limit setting
ADWINULB = 00FFh       //Window B upper limit setting
ADCMPBNSR = 01h        //Window B compare channel selection
```

```
[Example] When Window B comparison is not used
ADWINLLB = 0000h       //Window B lower limit setting
ADWINULB = 0000h       //Window B upper limit setting
ADCMPBNSR = 3Fh        //Window B compare channel non-selection
```

```
[Example]
ADCMPCR = 4A00h //Window A/B enabled, compound condition OR setting
```

Figure 47.32 Setting example when using the event output of the compare function

Notes on the event output usage when using only the Window A for compare function:

- Enable both Window A and Window B (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)
- Set the compound condition of Window A and Window B to “OR condition” (ADCMPCR.CMPAB[1:0] = 00b)
- Set the compared channel of Window B to “Do not select” (ADCMPBNSR.CMPCHB[5:0] = 111111b)
- Set the compare condition of Window B to “0 < results < 0 always means mismatch”. (ADCMPCR.WCMPE = 1, ADWINLLB.CMPLLB[15:0] = ADWINULB.CMPULB[15:0] = 0000h, and ADCMPBNSR.CMPLB = 1)

Figure 47.33 shows the event output operation example of compare function.

A scan end event (ADC12i_ADI) is output with the same timing of one time single scan completion. A match or mismatch event (ADC12i_WCMPM or ADC12i_WCMPUM) is output delayed 1 PCLKB depending on ADCMPCR.CMPAB[1:0] settings.

Note: The match and mismatch events are exclusive, so both events are never output simultaneously.

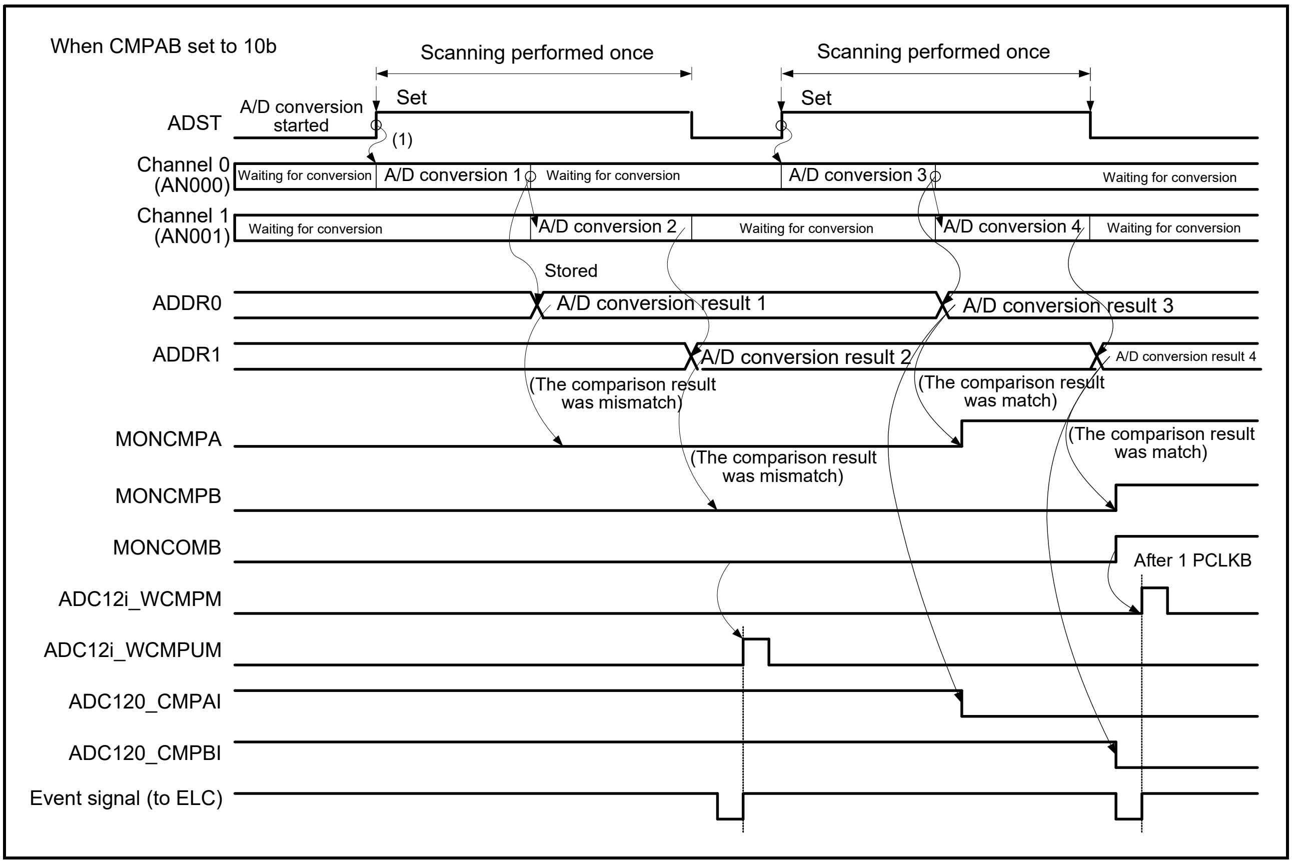


Figure 47.33 Example operation of the compare function event output, when AN000 to AN001 are compared

Note 1. Event output of compare function outputs match/mismatch from the comparison results of Window A and Window B, based on the ADCMPCR.CMPAB[1:0] settings.

Note 2. The comparison result of Window A is the logical addition of the comparison results of the comparison target channels of Window A. The comparison results of Window A and Window B are updated by each A/D conversion, and are kept even when single scan ends. Set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0 to clear the comparison results to 0.

47.3.5.3 Constraints on the compare function

The following constraints apply for the compare function:

- The compare function cannot be used together with the self-diagnosis function or double-trigger mode. (The compare function is not available for ADRD, ADDBLDR, ADDBLDRA, and ADDBLDRB.)
- Specify single scan mode when using match/mismatch event outputs.
- When the temperature sensor output or internal reference voltage is selected for Window A, Window B operations are disabled.
- When the temperature sensor output or internal reference voltage is selected for Window B, Window A operations are disabled.
- Setting the same channel for Window A and Window B is prohibited.
- Set the reference voltage values so that the high-potential reference voltage value is equal to or larger than the low-potential reference voltage value.

47.3.6 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated by a software trigger, a synchronous trigger (ELC), or an asynchronous trigger (ADTRGn). After the start-of-scanning-delay (t_D) elapses, processing by the channel-dedicated sample-and-hold circuits, processing for disconnection detection assistance, and processing of conversion for self-diagnosis all proceed, followed by processing for A/D conversion.

Figure 47.34 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (ELC). Figure 47.35 shows the scan conversion timing, in which scan conversion is activated by an asynchronous trigger, ADTRGn. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay (t_D), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH}),[*1] disconnection detection assistance processing time (t_{DIS}),[*2] self-diagnosis A/D conversion processing time (t_{DIAG}),[*3] A/D conversion processing time (t_{CONV} and t_{DSD}), channel-dedicated sample-and-hold circuit end time (t_{SHED}),[*4] and end-of-scanning-delay (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of the input sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the ADC12. If there is not sufficient sampling time because of the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTR register.

The time for conversion by successive approximation (t_{SAM}) is 13 ADCLK states with 12-bit accuracy selected, 11 ADCLK states with 10-bit accuracy selected, and 9 ADCLK states with 8-bit accuracy selected.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)^{*5} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}. The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed at $t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)^{*5} + t_{SHED}$.

Note 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SPLSH} = 0$.

Note 2. When disconnection detection assistance is not selected, $t_{DIS} = 0$.

Note 3. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.

Note 4. When no channel-dedicated sample-and-hold circuits are used, $t_{SHED} = 0$, assuming continuous scan mode is active. In single scan mode and group scan mode, t_{SHED} is included in the end-of-scanning-delay (t_{ED}).

Note 5. When input sampling times (t_{SPL}) of all selected channels are the same, this element equals $t_{CONV} \times n$. If each channel has a different sampling time, this element equals that of t_{SPL} and t_{SAM} set to each selected channel.

Table 47.10 Conversion times during scanning (in numbers of cycles of ADCLK and PCLKB)

Parameter			Symbol		Type/conditions			Unit
					Synchronous trigger*4	Asynchronous trigger	Software trigger	
Scan start processing time*1, *2	A/D conversion on Group A under Group A priority control.	Group B is to be stopped. (Group A is activated after Group B is stopped by of an A/D conversion source from Group A.)	t_D		3 PCLKB + 6 ADCLK 5 PCLKB + 3 ADCLK *5	-	-	Cycles
		Group B is not to be stopped. (Activation by an A/D conversion source from Group A.)			2 PCLKB + 4 ADCLK	-	-	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			2 PCLKB + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	All other				2 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time*1	Sampling time		t_{SPLSH}	t_{SH}	Without continuous sampling: setting in ADSHCR.SSTSH[7:0] (initial value = 18h × ADCLK) With continuous sampling: 0			
	Wait time between sampling and A/D Conversion			t_W	12			
Disconnection detection assistance processing time			t_{DIS}		Setting in ADNDIS[3:0] (initial value = 0h) × ADCLK			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	Setting in ADSSTR00 (initial value = 0Bh) × ADCLK*3	-	-	
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	15 ADCLK	-	-	
		10-bit conversion accuracy			13 ADCLK	-	-	
		8-bit conversion accuracy			11 ADCLK	-	-	
	Wait time between self-diagnosis conversion end and analog channel sampling start			t_{DED}	2 ADCLK			
	Wait time between last channel conversion end and self-diagnosis sampling start in continuous scan mode		t_{DSD}		2 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	Setting in ADSSTRn (n = 00 to 07, L, T, O) (initial value = 0Bh) × ADCLK + 0.5 ADCLK			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	13 ADCLK			
		10-bit conversion accuracy			11 ADCLK			
		8-bit conversion accuracy			9 ADCLK			
Channel-dedicated sample-and-hold end processing time			t_{SHED}		2 ADCLK			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK 2 PCLKB + 3 ADCLK*5			

Note 1. See Figure 47.34 and Figure 47.35 for an illustration of times t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.
Note 3. The sampling time setting should satisfy the electrical characteristics.
Note 4. This does not include the time consumed in the path from timer output to trigger input.
Note 5. If ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2 or 1:4), the scan end processing time changes.

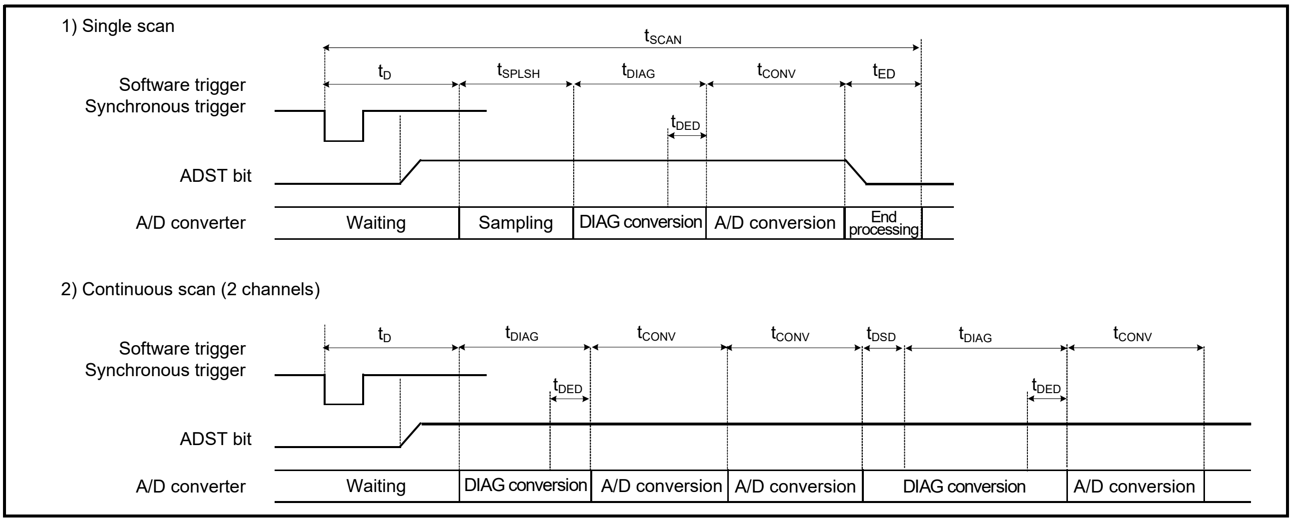


Figure 47.34 Scan conversion timing when activated by software or a synchronous trigger input (ELC)

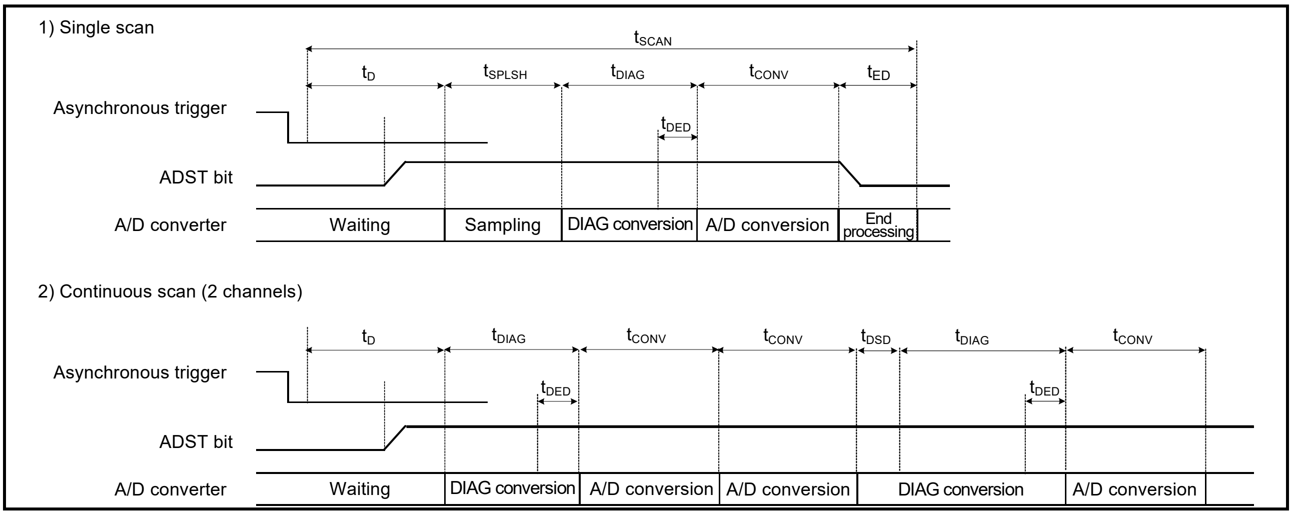


Figure 47.35 Scan conversion timing when activated by an asynchronous trigger input (ADTRG0)

47.3.7 Usage Example of A/D Data Register Automatic Clearing Function

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, temperature sensor output, or internal reference voltage is selected.

Setting the ACE bit in ADCER to 1 automatically clears the data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, and ADOCDR) to 0000h when the data registers are read by the CPU, DTC, or DMAC. This function enables detection of update failures by the data registers. In the following examples, the function to automatically clear the ADDRy register is enabled and disabled.

If the ACE bit in ADCER is 0 (automatic clearing disabled) and for some reason the A/D conversion result (0222h) is not written to the ADDRy register, the ADDRy value retains the old data (0111h). In addition, if this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, the old data (0111h) can be saved in the general-purpose register. When checking whether there is an update failure, it is necessary to frequently save the old data in SRAM or in a general-purpose register.

If the ACE bit in ADCER is 1 (automatic clearing enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMAC, ADDRy is automatically cleared to 0000h. Next, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general-purpose register using an A/D scan end interrupt at this point, 0000h is saved in the general-purpose register. Occurrence of an ADDRy update failure can be determined by checking that the read data value is 0000h.

47.3.8 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 1, 2, 3, 4, or 16*1 consecutive times, and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted 2 or 4 consecutive times, and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. However, this function cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition or average mode can be specified for A/D conversion of the channel select analog input, temperature sensor output, or internal reference voltage.

Note 1. The addition count can be set to 16 only when 12-bit accuracy is selected.

47.3.9 Disconnection Detection Assist Function

This converter incorporates a function that fixes the charge for sampling capacitance to the specified state (VREFH0 or VREFL0 for unit 0, VREFH or VREFL for unit 1) before the start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.

When using the disconnection detection assist function for the channel-dedicated sample-and-hold circuit, set ADSHMSR.SHMD bit to 0 (select disable continuous sampling function).

If any of the following functions are used, the disconnection detection assist function must be disabled:

- The temperature sensor
- The internal reference voltage
- A/D self-diagnosis
- The programmable gain amplifier without bypass enabled.

Figure 47.36 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 47.37 shows an example of disconnection detection when precharge is selected. Figure 47.38 shows an example of disconnection detection when discharge is selected.

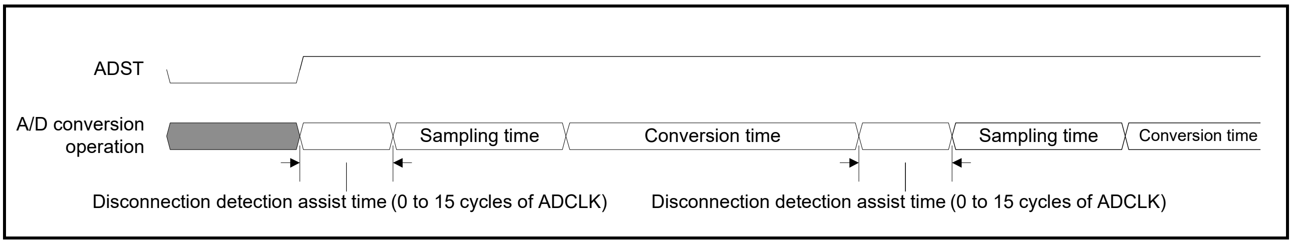
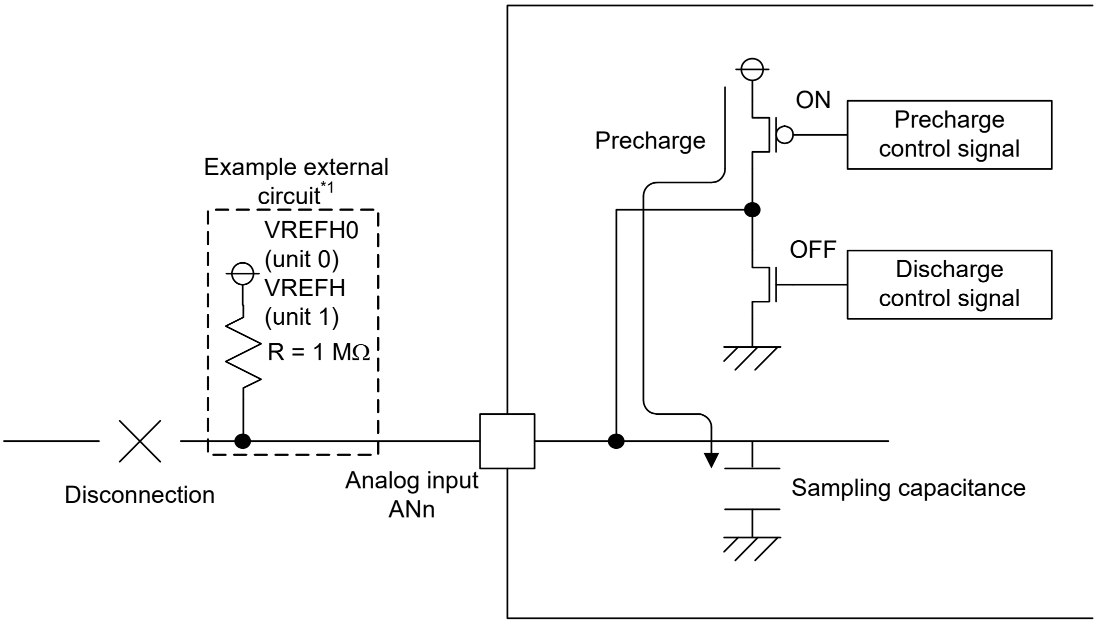
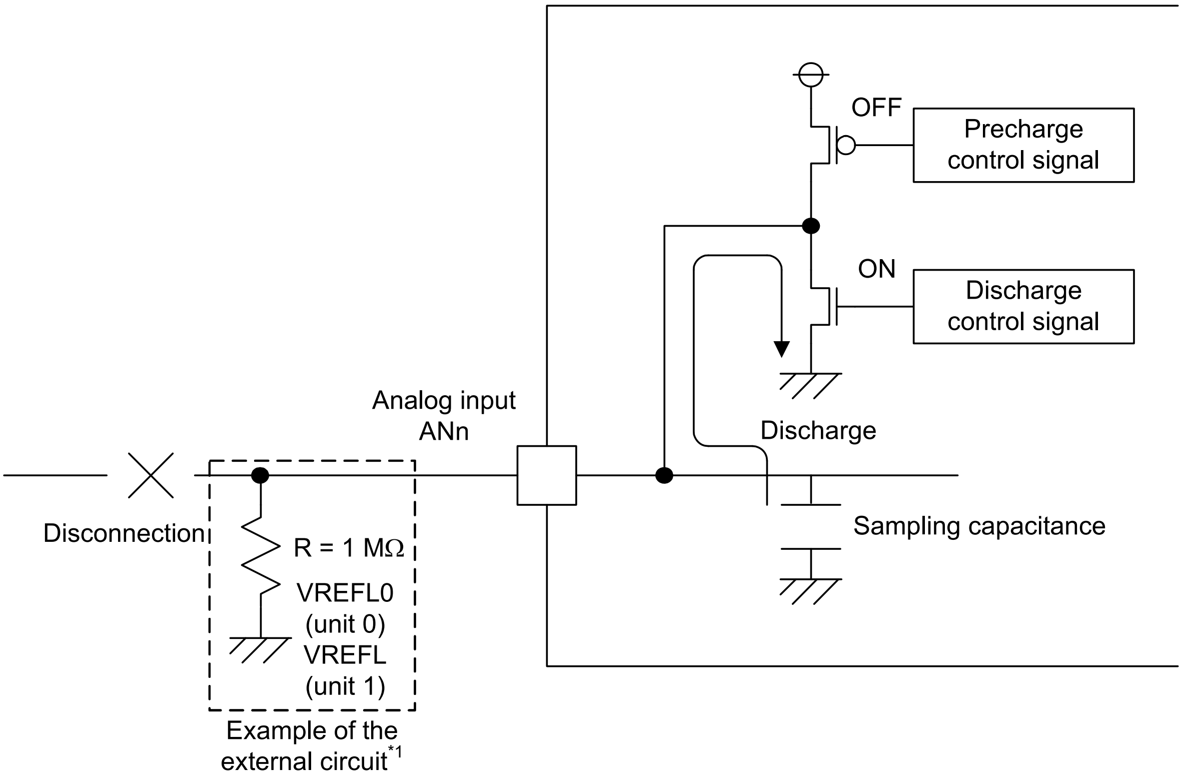


Figure 47.36 A/D conversion operation when the disconnection detection assist function is used



Note 1. The converted result should be only used after full evaluation, because the data result when disconnection occurs varies depending on the external circuit.

Figure 47.37 Example of disconnection detection when precharge is selected



Note 1. The converted result should only be used after full evaluation, because the data result when disconnection occurs varies depending on the external circuit.

Figure 47.38 Example of disconnection detection when discharge is selected

47.3.10 Starting A/D Conversion with an Asynchronous Trigger

A/D conversion can be started by the input of an asynchronous trigger. To start the ADC12 by an asynchronous trigger, first set the pin function in the PmnPFS register, next set the A/D Conversion Start Trigger Select bits (ADSTRGR.TRSA[5:0]) to 000000b, and then input a high-level signal to the asynchronous trigger (ADTRGn pin). Finally, set both the ADCSR.TRGE and ADCSR.EXTRG bits to 1. Figure 47.39 shows timing of the asynchronous trigger input.

An asynchronous trigger cannot be selected in the A/D conversion start trigger select bits (ADSTRGR.TRSB[5:0]) for Group B used in group scan mode. For details on setting the pin function, see section 20, I/O Ports.

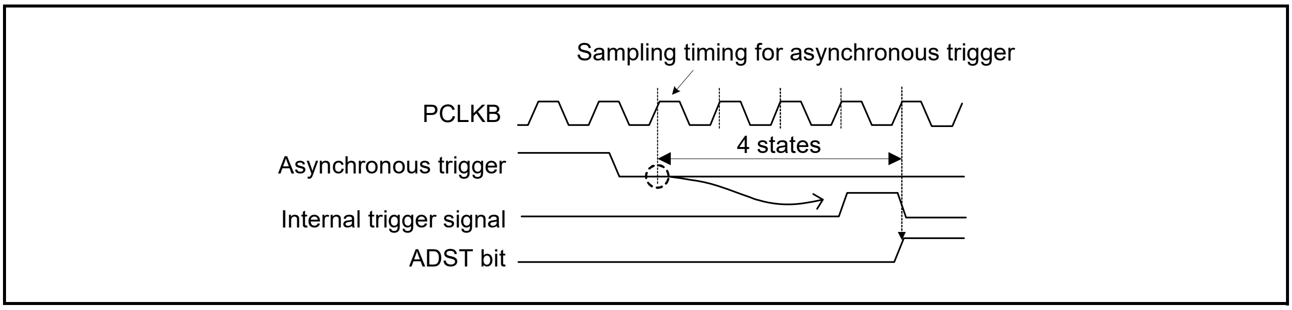


Figure 47.39 Asynchronous trigger input timing

47.3.11 Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module

A/D conversion can be started by a synchronous trigger (ELC). To start the A/D conversion by a synchronous trigger, set the ADCSR.TRGE bit to 1, clear the ADCSR.EXTRG bit to 0, and select the relevant sources in the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

47.3.12 Programmable Gain Amplifiers

Up to three programmable gain amplifiers (PGAs) can be used in each unit. Select a gain in the ADPGAGS.PnGAIN[3:0] bits (n = 000 to 002) and select an operational amplifier to be used in the ADPGACR.PnSEL bits.

These PGAs accept differential inputs. Pins that accept differential inputs are PGAVSS000 for AN000 to AN002 (unit 0), PGAVSS100 for AN100 to AN102 (unit 1). To use differential inputs, set the differential input gain in the ADPGADCR0.PnDG[2:0] bits, enable the differential input gain setting in the ADPGADCR0.PnGEN bits, and then select the differential input amplifier in the ADPGADCR0.PnDEN bits. The PGA register is selectable as shown in Table 47.11.

Table 47.11 Setting of PGA register and available related functions

	Setting of corresponding Register				related function ✓: available x: unavailable			
	PmnPFS	ADPGACR	ADPGAGS0	ADPGADCR0				
		PGA P002: bits [11:8]	bits [11:8]	bits [11:8]		ACMPHS*2		
		PGA P001: bits [7:4]	bits [7:4]	bits [7:4]		IVCMP 2	IVCMP 3	
Selectable value for each condition	ASEL*3	PGA P000: bit [3:0]	bit [3:0]	bit [3:0]	Ports*1			ADC12
When using Ports	0	Leave these bits with initial values			✓	x	x	x
When using ACMPHS or ADC12(PGA bypass)*4	1	9	0	0	x	✓	x	✓
When using PGA Differential input disabled	1	Eh	0 to Eh	0	x	✓	✓	✓
When using PGA Differential input enabled	1	Eh	1, 5, 9, Bh	8 to Bh	x	x	✓	✓

Note 1. Ports: When using input ports.
Note 2. ACMPHS IVCMP2: When using input through the PGA. ACMPHS IVCMP3: When using input of PGA output.
Note 3. For detail on the configuration of PmnPFS registers, see section 20, I/O Ports.
Note 4. Ports and ACMPHS cannot be used at the same time. Ports and ADC12 cannot be used at the same time.

Table 47.12 shows the calculation formula for the PGA output voltage.

Table 47.12 PGA output voltage

Mode	PGA output voltage
Single	Gain × Vin
Differential	Gain (Vin - Vs) + 0.5 × AVCC

Vin: AN000 to AN002, AN100 to AN102
Vs: PGAVSS00, PGAVSS001

47.4 Interrupt Sources and DTC/DMAC Transfer Requests

47.4.1 Interrupt Requests

The ADC12 can send scan end interrupt requests, ADC12i_ADI (i = 0, 1) and ADC12i_GBADI, to the CPU. The module also generates the ADC12i_CMPAI and ADC12i_CMPBI interrupts to the CPU in response to matches with a comparison condition.

An ADC12i_ADI interrupt is always generated. An ADC12i_GBADI interrupt can be generated by setting the ADCSR.GBADIE bit to 1. Similarly, ADC12i_CMPAI and ADC12i_CMPBI interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bit to 1.

In addition, the DTC or DMAC can be started when an ADC12i_ADI or ADC12i_GBADI interrupt, or ADC12i_WCMPM or ADC12i_WCMPUM event is generated. Using these interrupts or events to allow the DTC or DMAC to read the converted data enables continuous conversion without burdening the software.

For details on the DTC settings, see section 18, Data Transfer Controller (DTC), and for details on the DMAC settings, see section 17, DMA Controller (DMAC).

Table 47.13 describes the interrupt sources and ELC events available for the ADC12.

Table 47.13 ADC12 events (1 of 2)

✓: available x: unavailable

Operation		**Interrupt request or ELC event**			**Interrupt request**	**DTC/DMAC activation**	**ELC event request**	**Function**
Scan mode	**Double-trigger mode**	**Compare function Window A/B**	**Unit 0**	**Unit 1**				
Single scan mode	Deselected	Deselected	ADC120_ADI	ADC121_ADI	✓	✓	✓	ADC12i_ADI generated at the end of single scan
		Selected	ADC120_ADI	ADC121_ADI	✓	✓	✓	ADC12i_ADI generated at the end of single scan
			ADC120_CMPAI	ADC121_CMPAI	✓	x	x	ADC12i_CMPAI generated on a match comparison condition of Window A
			ADC120_CMPBI	ADC121_CMPBI	✓	x	x	ADC12i_CMPBI generated on a match comparison condition of Window B
			ADC120_WCMPM	ADC121_WCMPM	x	✓	✓	ADC12i_WCMPM generated on a match condition of the Window A/B compare function
			ADC120_WCMPUM	ADC121_WCMPUM	x	✓	✓	ADC12i_WCMPUM generated on a mismatch condition of the Window A/B compare function
	Selected	Deselected	ADC120_ADI	ADC121_ADI	✓	✓	✓	ADC12i_ADI generated at the end of scans in the even-numbered times
Continuous scan mode	Deselected	Deselected	ADC120_ADI	ADC121_ADI	✓	✓	✓	ADC12i_ADI generated at the end of scan of all selected channels
		Selected	ADC120_CMPAI	ADC121_CMPAI	✓	x	x	ADC12i_CMPAI generated on a match comparison condition of Window A
			ADC120_CMPBI	ADC121_CMPBI	✓	x	x	ADC12i_CMPBI generated on a match comparison condition of Window B

Table 47.13 ADC12 events (2 of 2)

✓: available x: unavailable

Operation		**Interrupt request or ELC event**			**Interrupt request**	**DTC/ DMAC activation**	**ELC event request**	**Function**
Scan mode	**Double-trigger mode**	**Compare function Window A/B**	**Unit 0**	**Unit 1**				
Group scan mode	Deselected	Deselected	ADC120_ADI	ADC121_ADI	✓	✓	✓	ADC12i_ADI generated at the end of Group A scan
			ADC120_GBADI	ADC121_GBADI	✓	✓	x	ADC12i_GBADI dedicated to Group B generated at the end of Group B scan
		Selected	ADC120_ADI	ADC121_ADI	✓	✓	✓	ADC12i_ADI generated at the end of Group A scan
			ADC120_GBADI	ADC121_GBADI	✓	✓	x	ADC12i_GBADI dedicated to Group B generated at the end of Group B scan
			ADC120_CMPAI	ADC121_CMPAI	✓	x	x	ADC12i_CMPAI generated on a match comparison condition of Window A
			ADC120_CMPBI	ADC121_CMPBI	✓	x	x	ADC12i_CMPBI generated on a match comparison condition of Window B
	Selected	Deselected	ADC120_ADI	ADC121_ADI	✓	✓	✓	ADC12i_ADI generated at the end of Group A scans in the even-numbered times
			ADC120_GBADI	ADC121_GBADI	✓	✓	x	ADC12i_GBADI dedicated to Group B generated at the end of Group B scan

Note: i = 0: unit 0, i = 1: unit 1.

47.5 Event Link Function

47.5.1 Event Output to the ELC

The ELC uses the ADC12i_ADI interrupt request signal as an event signal, enabling link operation for the preset module. The ADC12i_GBADI interrupt and ADC12i_CMPAI/ADC12i_CMPBI interrupts cannot be used as event signals. For details, see Table 47.13.

47.5.2 ADC12 Operation through an Event from the ELC

The ADC12 can start A/D conversion by the preset event specified in the ELSRn settings for the ELC as follows:

- Select the ELC_AD00 (unit 0) signal in the ELC.ELSR8 register
- Select the ELC_AD01 (unit 0) signal in the ELC.ELSR9 register
- Select the ELC_AD10 (unit 1) signal in the ELC.ELSR10 register
- Select the ELC_AD11 (unit 1) signal in the ELC.ELSR11 register.

If an ELC_ADi0 or ELC_ADi1 event occurs during A/D conversion, the event is disabled.

47.6 Usage Notes

47.6.1 Constraints on Reading the Data Registers

The following registers must be read in halfword units:

- A/D Data Registers
- A/D Data Duplexing Register
- A/D Data Duplexing Register A
- A/D Data Duplexing Register B
- A/D Temperature Sensor Data Register
- A/D Internal Reference Voltage Register
- A/D Self-Diagnosis Data Register.

If a register is read twice in byte units, that is, the upper byte and bytes are read separately, the A/D-converted value read initially might disagree with the A/D-converted value read subsequently. To prevent this, never read the data registers in byte units.

47.6.2 Constraints on Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger is selected as the condition for starting A/D conversion, follow the procedure shown in Figure 47.40.

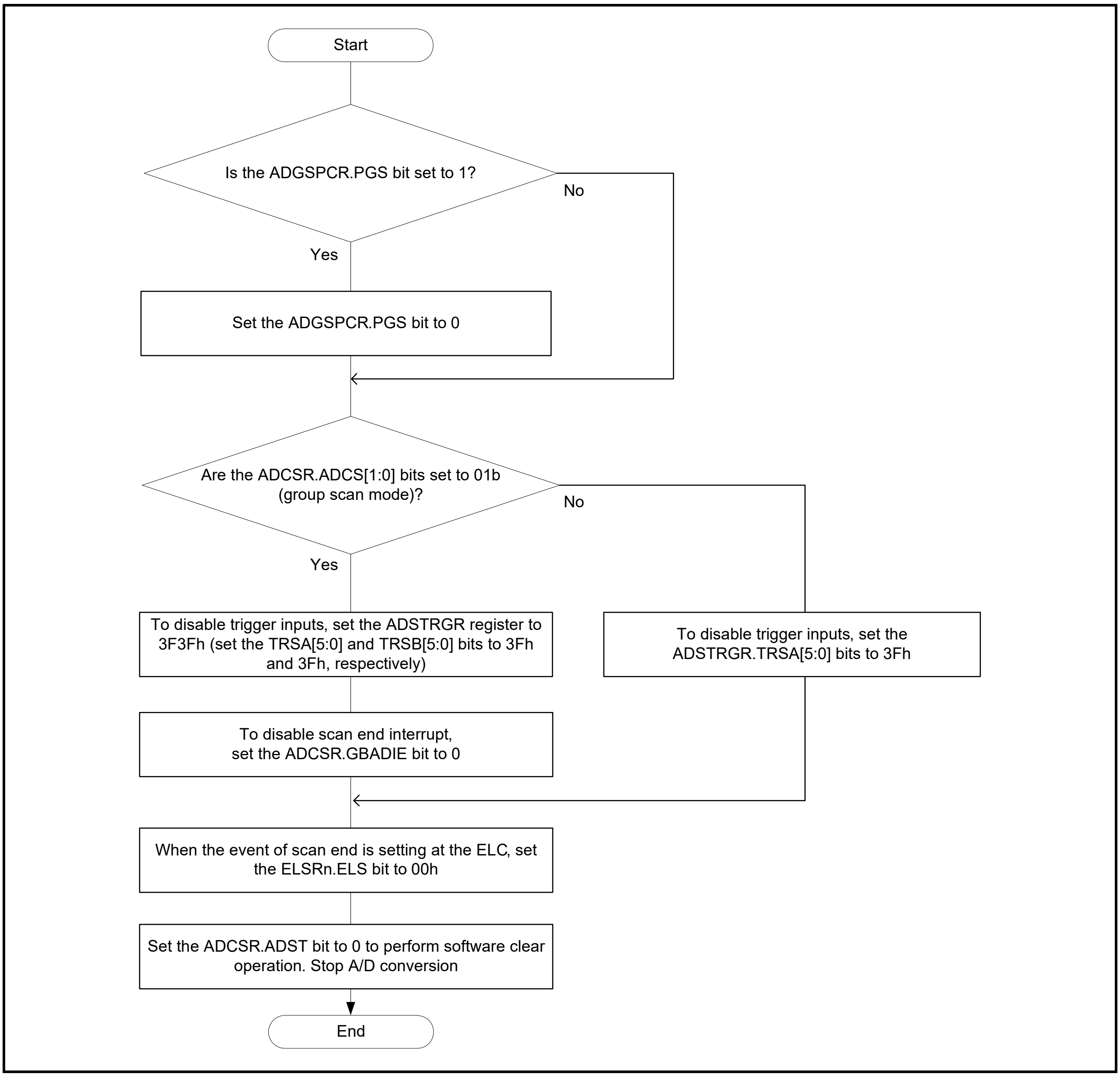


Figure 47.40 Procedures for clearing the ADCSR.ADST bit by software

47.6.3 A/D Conversion Restart and Termination Timing

A maximum of 6 ADCLK cycles is required for the idle analog unit of the ADC12 to restart on setting the ADCSR.ADST bit to 1. A maximum of 2 ADCLK cycles is required for the operating analog unit of the ADC12 to be terminate on setting the ADCSR.ADST bit to 0.

47.6.4 Constraints on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data if the CPU does not finish reading the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

47.6.5 Settings for the Module-Stop Function

ADC12 operation can be disabled or enabled using the Module Stop Control Register. The ADC12 is initially stopped after reset. Releasing the module-stop state enables access to the registers. After release from the module-stop state, wait for at least 1 μs before starting A/D conversion. For details, see section 11.4, Module-Stop Function.

47.6.6 Notes on Entering the Low-Power States

Before entering the module-stop state or Software Standby mode, you must stop A/D conversion. Here, set the ADCSR.ADST bit to 0 and secure a period of time until the analog unit of the ADC12 stops.

To secure this time, follow the procedure shown in Figure 47.40 for clearing the ADCSR.ADST bit by software. Next, wait for 2 clock cycles of ADCLK before entering the module-stop state or Software Standby mode.

47.6.7 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use

Using disconnection detection assistance leads to an error in the absolute accuracy of the ADC12. This error arises because an erroneous voltage is input to the analog input pins because of the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula.

Maximum error in absolute accuracy (LSB) = $(2^{\text{Resolution}} -1) \times \text{Rs} / (\text{Rs} + \text{Rp})$, Resolution = 12, 10, 8

Only use disconnection detection assistance after thorough evaluation.

47.6.8 Available Functions and Register Settings of AN000 to AN002, AN007, AN100 to AN102, and AN107

Table 47.14 shows the available functions and register setting of AN000 to AN002, AN007, AN100 to AN102, and AN107. Figure 47.41 shows the setting procedure of registers. To use each function, set the register value shown in Table 47.14.

When the PGA is used with differential input enabled, a negative voltage can be input for AN000 to AN002 and PGAVSS000 for unit 0, and AN100 to AN102 and PGAVSS100 for unit 1 pin after setting the registers.

When the PGA is used with differential input enabled, all PGA amplifiers in each unit must be set to differential input in the ADPGADCR0 register.

When the PGA is used with differential input disabled, the associated PGAVSS pin must be connected to AVSS0. When the PGA is not used, the associated PGAVSS can be used as an input port.

When transitioning to the ADC module-stop state or Software Standby mode from the state of using PGA or sample-and-hold circuit, if 0 is set to the associated bit of ADPGACR or ADSHCR register of each ADC12 before transitioning, power consumption can be reduced.

The initial value of the ASEL bit of P003 and P007 is 1. When these pins are not used as an analog function, to reduce the input leakage current, set the ASEL bit to 0.

Table 47.14 Available functions and register setting

Available functions						Register setting				
						P0nPFS*6		PGA		S/H*3
Ports*1	IRQ*2	S/H*3	PGA-S*4	PGA-D*5	ADC12	ASEL	ISEL	ADPGADCR0*7	ADPGACR*8	ADSHCR*9
✓	-	-	-	-	-	0	0	x	x	x
-	✓	-	-	-	-	0	1	x	x	x
-	-	-	-	-	✓*10	1	x	0	9h	0
-	-	-	-	-	✓*11	1	x	0*11	0 or 9h	0
-	-	✓	-	-	✓	1	x	0	9h	1 (0*13)
-	-	✓	✓	-	✓	1	x	0	Eh (0h*12)	1 (0*13)
-	-	✓	-	✓	✓	1	x	1	Eh (0h*12	1 (0*13)
-	-	-	✓	-	✓	1	x	0	Eh (0h*12	0
-	-	-	-	✓	✓	1	x	1	Eh (0h*12	0

✓: Available
x: Don’t care

Note 1. Ports: P000 to P007 can be used as port inputs.

Note 2. IRQ: P000 to P002 and P003 to P005 can be used as IRQ pins.

Note 3. S/H: sample-and-hold circuit.

Note 4. PGA-S: When the PGA setting is Differential input disabled. Corresponding PGAVSS must be set as ASEL to 1, and connect to AVSS0.

Note 5. PGA-D: When the PGA setting is Differential input enabled. Corresponding PGAVSS must set as ASEL to 1.

Note 6. P0nPFS: Port 0 Pin Function Select register (n = 00 to 07) corresponding to analog input pin.

Note 7. It indicates the corresponding Differential Input Enable bit (bit [11] or bit [7] or bit [3]) in the ADPGADCR0 register.

Note 8. It indicates the corresponding Amplifier Control bits (bits [11:8] or bits [7:4], or bits [3:0]) in the ADPGACR register.

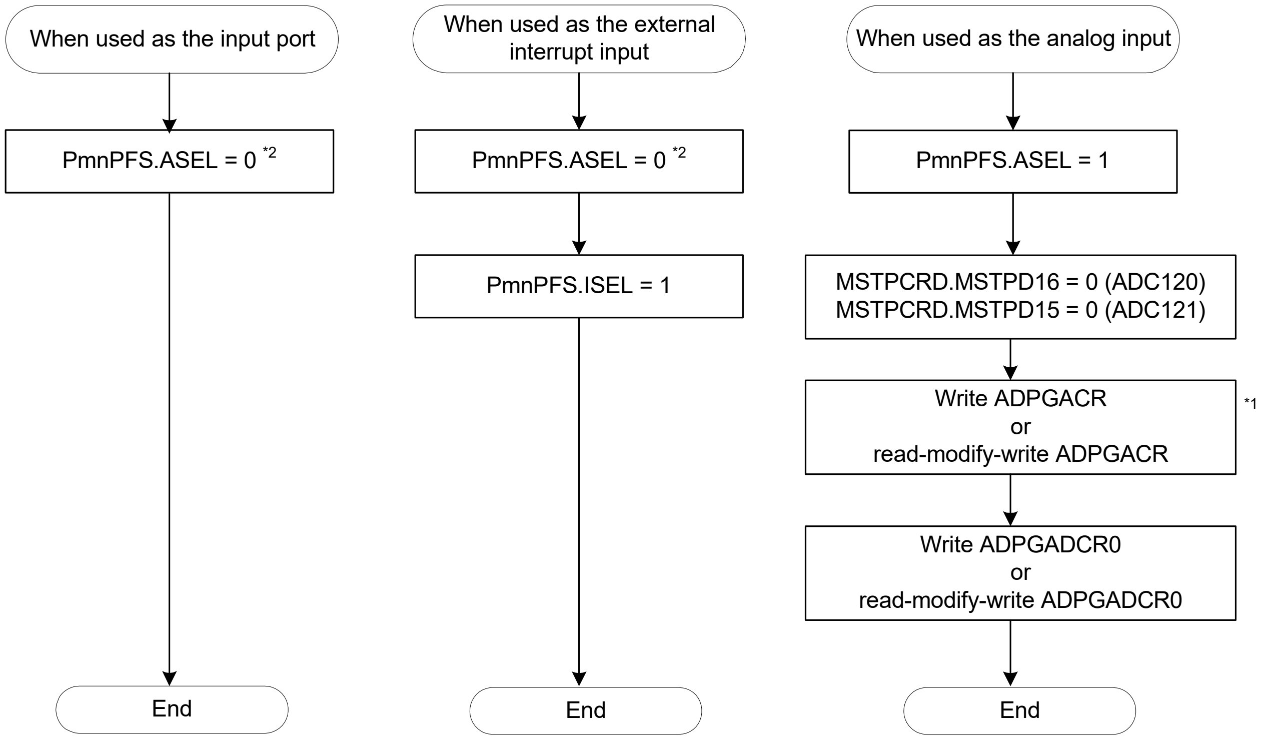
Note 9. It indicates the corresponding Bypass Select bit (bit [10] or bit [9] or bit [8]) in ADSHCR register.

Note 10. When using AN000 to AN002 or AN100 to AN102.

Note 11. When using AN007 or AN107. Set all corresponding bits (ADPGADCR0 bit [11], bit [7], and bit [3] to 0.

Note 12. Power consumption of the PGA can be reduced by setting 0h to the corresponding bit of the ADPGACR register before transitioning to the ADC12 module-stop state or Software Standby mode.

Note 13. Power consumption of the S/H can be reduced by setting 0 to the corresponding bit of the ADSHCR register before transitioning to the ADC12 module-stop state or Software Standby mode.



Note 1. For AN007 and AN107, this step can be skipped.

Note 2. The initial value of the ASEL bit of P003 and P007 is 1. When these pins are not used as an analog function to reduce the input leakage current, set the ASEL bit to 0.

Figure 47.41 Setting procedure of registers

47.6.9 Constraints on Operating Modes and Status Bits

The voltage values in self-diagnosis, the determination of first or second scan in double-trigger mode, and the data buffer pointer and status monitor in the compare function initialize or set it again individually, if required.

- The voltage values in self-diagnosis can be selected in ADCER.DIAGVAL[1:0] after ADCER.DIAGLD is set to 1.
- The double-trigger mode operates as the first scan after ADCSR.DBLE is set to 1 from 0.
- The status monitor bits (MONCMPA, MONCMPB, and MONCOMB) in the compare function are initialized after ADCMPCR.CMPAE and ADCMPCR.CMPBE are cleared to 0.
- The constant sampling function (ADSHMSR.SHMD = 1) is initialized after ADSHMSR.SHMD is cleared to 0. When the constant sampling function is used (setting ADSHMSR.SHMD = 1) again, a wait of 1ADCLK or more waiting is required.

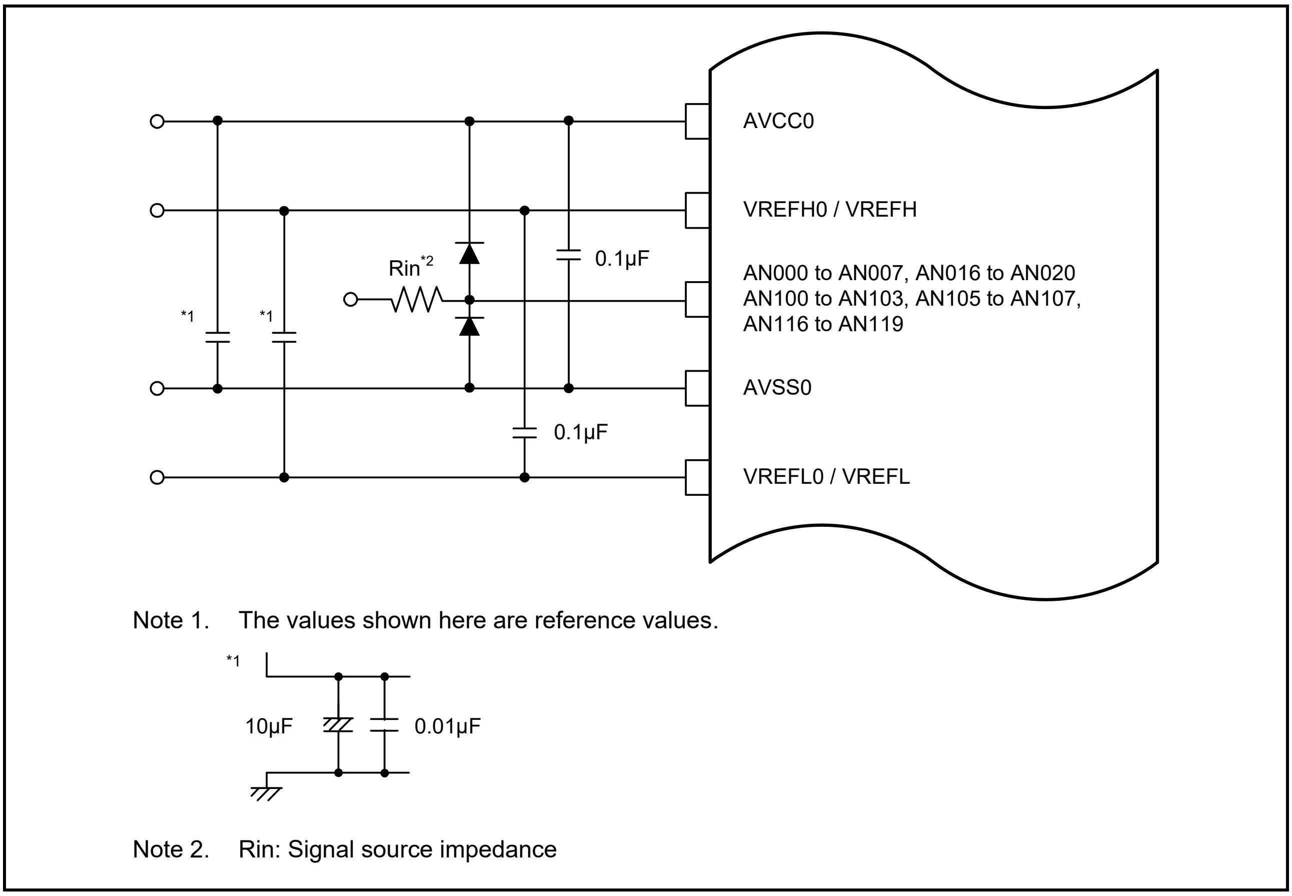
47.6.10 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible.

In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN007, AN016 to AN020, AN100 to AN103, AN105 to AN107, AN116 to AN119), reference power supply pin (VREFH0/VREFH), reference ground pin (VREFL0/VREFL), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

47.6.11 Constraints on Noise Prevention

To prevent the analog input pins (AN000 to AN007, AN016 to AN020, AN100 to AN103, AN105 to AN107, and AN116 to AN119) from being destroyed by abnormal voltage such as excessive surges, insert a capacitor between AVCC0 and AVSS0, between VREFH0 and VREFL0, and between VREFH and VREFL, and connect a protection circuit to protect the analog input pins (AN000 to AN007, AN016 to AN020, AN100 to AN103, AN105 to AN107, and AN116 to AN119) as shown Figure 47.42.



Note 1. The values shown here are reference values.

Note 2. Rin: Signal source impedance

Figure 47.42 Example protection circuit for analog inputs

47.6.12 Port Settings When Using the ADC12 Input

When using the high-precision channels, do not use ports 0 as digital output ports. Renesas also recommends not using the digital output that is also used as the A/D analog input, if normal-precision channels is used. If the digital output that is also used as the A/D analog input is used for output signals, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

47.6.13 Relationship between ADC12 Units 0 and 1 and the ACMPHS

For the A/D conversion targets in Table 47.15, unit 0 and 1 should not perform A/D conversion at the same time.

Table 47.15 A/D conversion targets that are mutually exclusive with each other

A/D conversion target	
Unit 0	**Unit 1**
Temperature sensor	
Internal reference voltage	
AN005/DA0	AN105/DA0
AN006/DA1	AN106/DA1

The A/D conversion targets in Table 47.16 should not be selected as ACMPHS input during A/D conversion, because these pins are multiplexed with the ADC12 and ACMPHS.

Table 47.16 A/D conversion targets that are mutually exclusive with ACMPHS

A/D conversion target		
Unit 0	**Unit 1**	**ACMPHS**
AN000	-	ACMPHS0.IVCMP2
AN001	-	ACMPHS1.IVCMP2
AN002	-	ACMPHS2.IVCMP2
PGA P000 output	-	ACMPHS0.IVCMP3
PGA P001 output	-	ACMPHS1.IVCMP3
PGA P002 output	-	ACMPHS2.IVCMP3
AN005/DA0	-	ACMPHS0 to ACMPHS5.IVREF3
AN006/DA1	-	ACMPHS0 to ACMPHS5.IVCMP1
AN016	-	ACMPHS0 to ACMPHS5.IVREF0
AN017	-	ACMPHS0 to ACMPHS5.IVCMP0
Internal reference voltage	-	ACMPHS0 to ACMPHS5.IVREF2
-	AN100	ACMPHS3.IVCMP2
-	AN101	ACMPHS4.IVCMP2
-	AN102	ACMPHS5.IVCMP2
-	PGA P000 output	ACMPHS3.IVCMP3
-	PGA P001 output	ACMPHS4.IVCMP3
-	PGA P002 output	ACMPHS5.IVCMP3
-	AN105/DA0	ACMPHS3 to ACMPHS5.IVREF3
-	AN106/DA1	ACMPHS3 to ACMPHS5.IVCMP1
-	AN116	ACMPHS0 to ACMPHS5.IVREF1
-	Internal reference voltage	ACMPHS0 to ACMPHS5.IVREF2

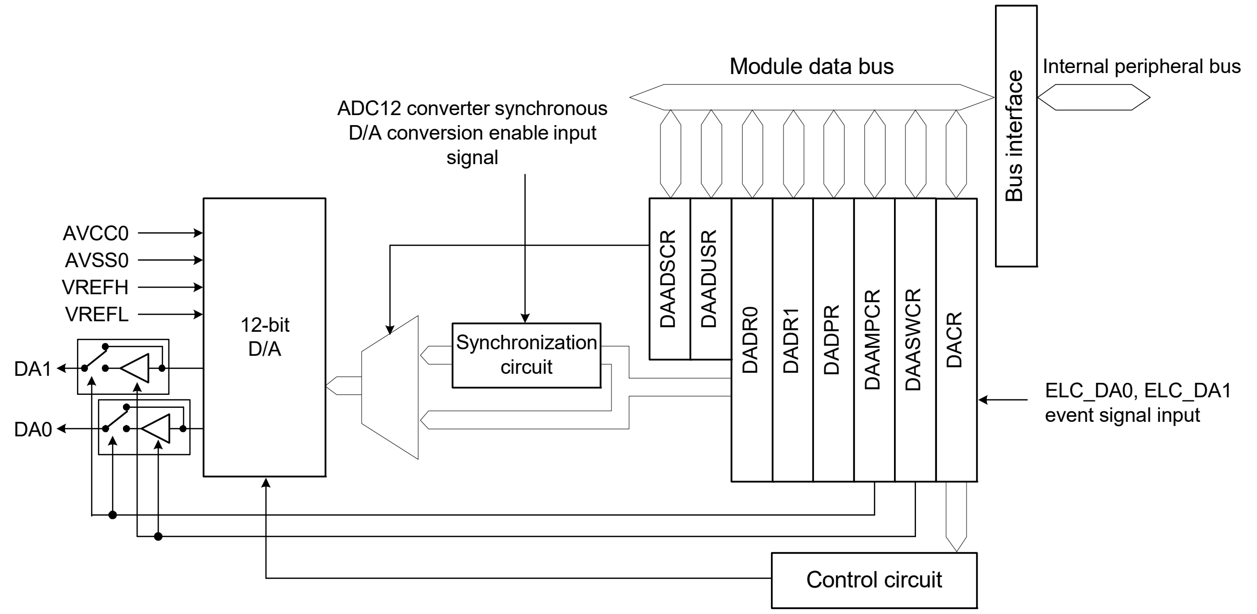
48. 12-Bit D/A Converter (DAC12)

48.1 Overview

The MCU provides a 12-Bit D/A Converter (DAC12) with an output amplifier. Table 48.1 lists the DAC12 specifications, Figure 48.1 shows a block diagram, and Table 48.2 lists the I/O pins.

Table 48.1 DAC12 specifications

Parameter	Specifications
Resolution	12 bits
Output channels	2 channels
Interference reduction between analog modules	Methods provided to minimize interference between D/A and A/D conversion: • D/A converted data update timing is controlled by the ADC12 synchronous D/A conversion enable input signal from the ADC12 (unit 1) • Degradation of A/D conversion accuracy caused by interference is reduced by controlling DAC12 inrush current generation timing with the enable signal
Module-stop function	Module-stop state can be set to reduce power consumption
Event link function (input)	DA0 and DA1 conversion can be started when an event signal is input
D/A output amplifier control function	Controls whether the output amplifier (for both amplifier-through and amplifier-bias controls) is used



DADR0: D/A Data Register 0
DADR1: D/A Data Register 1
DACR: D/A Control Register
DADPR: DADRm Format Select Register
DAADSCR: D/A A/D Synchronous Start Control Register
DAAMPCR: D/A Output Amplifier Control Register
DAASWCR: D/A Amplifier Stabilization Wait Control Register
DAADUSR: D/A A/D Synchronous Unit Select Register

Figure 48.1 DAC12 block diagram

Table 48.2 DAC12 I/O pins (1 of 2)

Pin name	I/O	Function
AVCC0	Input	Analog power supply pin for ADC12, DAC12, and comparator Connect to VCC when these modules are not used.
AVSS0	Input	Analog ground pin for ADC12, DAC12, and comparator Connect to VSS when these modules are not used.

Table 48.2 DAC12 I/O pins (2 of 2)

Pin name	I/O	Function
VREFH	Input	Analog reference top voltage supply pin for the ADC12 (unit 1) and the DAC12
VREFL	Input	Analog reference ground pin for the ADC12 (unit 1) and the DAC12
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

48.2 Register Descriptions

48.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DAC12.DADR0 4005 E000h, DAC12.DADR1 4005 E002h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DADRm registers are 16-bit read/write registers that store data for D/A conversion. Whenever an analog output is enabled, the values in DADRm are converted and output to the analog output pins.

12-bit data can be formatted as left- or right-justified in the DADPR.DPSEL bit setting. In right-justified format (DADPR.DPSEL = 0), the lower 12 bits, [11:0], are valid. In left-justified format (DADPR.DPSEL = 1), the upper 12 bits, [15:4], are valid.

For information on using the output amplifier, see section 48.6.5, Initialization Procedure with the Output Amplifier.

48.2.2 D/A Control Register (DACR)

Address(es): DAC12.DACR 4005 E004h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	DAE	D/A Enable*1	0: Control D/A conversion of channels 0 and 1 individually 1: Control D/A conversion of channels 0 and 1 collectively.	R/W
b6	DAOE0	D/A Output Enable 0	0: Disable analog output of channel 0 (DA0) 1: Enable D/A conversion of channel 0 (DA0).	R/W
b7	DAOE1	D/A Output Enable 1	0: Disable analog output of channel 1 (DA1) 1: Enable D/A conversion of channel 1 (DA1).	R/W

Note 1. This bit controls D/A conversion in combination with the DAOEi bit (i = 0, 1), which controls output of the conversion results. For details, see Table 48.3.

Table 48.3 D/A conversion controls

b5	b7	b6	
DAE	DAOE1	DAOE0	Description
0	0	0	Disable D/A conversion and analog output pins (DA0, DA1)*1
		1	• Enable D/A conversion of channel 0 and disable D/A conversion of channel 1 • Enable analog output of channel 0 (DA0) and disable analog output of channel 1 (DA1)*1
	1	0	• Disable D/A conversion of channel 0 and enable D/A conversion of channel 1 • Disable analog output of channel 0 (DA0)*1 and enable analog output of channel 1 (DA1)
		1	• Enable D/A conversion of channels 0 and 1 • Enable analog output of channels 0 and 1 (DA0, DA1)
1	x	x	• Enable D/A conversion of channels 0 and 1 • Collective enable analog output of channels 0 and 1 (DA0, DA1)

x: Don't care

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

Only set this register while the ADC12 is halted when the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversion is enabled). Only set DACR while the ADCSR.ADST bit is 0 and after selecting the software trigger, for the ADC12 trigger to securely stop the ADC12. This MCU only supports ADC12 unit 1.

DAE bit (D/A Enable)

The DAE bit controls D/A conversion, amplifier operation, and analog output in combination with the DAOEi bit (i = 0, 1) and the DAMMPCR.DAAMPi bit (i = 0, 1). See Table 48.4.

When interference prevention between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the ADCSR.ADST bit of the ADC12 (unit 1) to 0. Then select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1).

DAOEi bit (D/A Output Enable i)

The DAOEi bit (i = 0, 1) controls D/A conversion, amplifier operation, and analog output in combination with the DAE bit and DAMMPCR.DAAMPi bit (i = 0, 1). See Table 48.4.

When both the DAOEi bit (i = 0, 1) and DAE bit are 0, D/A conversion of channel i (i = 0, 1) is not processed, and no conversion result is output.

When interference prevention between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAOEi bit while the ADCSR.ADST bit of the ADC12 (unit 1) is set to 0. Then select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1).

The event link function can be used to set the DAOEi bit to 1. The DAOE0 bit sets to 1 when the event specified in the ELSR12 register of the ELC (ELC_DA0 event) occurs, and output of the D/A conversion results starts. The DAOE1 bit sets to 1 when the event specified in the ELSR13 register of the ELC (ELC_DA1 event) occurs, and output of the D/A conversion results starts.

Table 48.4 D/A conversion and analog output control (1 of 2)

DACR		DAAMPCR			
DAE	DAOEi	DAAMPi	Channel i operation	Amplifier operation of channel i	Analog output of channel i
0	0	0	Stop	Stop	Hi-Z
		1	Stop	Stop	Hi-Z
	1	0	Run	Stop	Amplifier-through
		1	Run	Run	Amplifier output

Table 48.4 D/A conversion and analog output control (2 of 2)

DACR		DAAMPCR	Channel i operation	Amplifier operation of channel i	Analog output of channel i
DAE	DAOEi	DAAMPi			
1	0	0	Run	Stop	Amplifier-through
		1	Run	Run	Amplifier output
	1	0	Run	Stop	Amplifier-through
		1	Run	Run	Amplifier output

i = 0, 1

48.2.3 DADRm Format Select Register (DADPR)

Address(es): DAC12.DADPR 4005 E005h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	DADRm Format Select	0: Right-justified format 1: Left-justified format.	R/W

48.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): DAC12.DAADSCR 4005 E006h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAADST	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DAADST	D/A A/D Synchronous Conversion	0: Do not synchronize DAC12 operation with ADC12 (unit 1) operation (disable interference prevention between D/A and A/D conversion) 1: Synchronize DAC12 operation with ADC12 (unit 1) operation (enable interference prevention between D/A and A/D conversion).	R/W

To minimize interference between D/A and A/D conversion, the DAADSCR register enables synchronization of the start timing of D/A conversion with the ADC12 synchronous D/A conversion enable input signal.

Only set this register while the ADC12 (unit 1) is halted, that is, while the ADCSR.ADST bit is 0 after selecting the software trigger as the ADC12 (unit 1) trigger.

Select unit 1 as the target ADC12 unit before setting the DAADST bit to 1.

Set bit [1] in the DAADUSR register to 1 to select unit 1. The MCU only supports ADC12 unit 1.

DAADST bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value to be converted into analog data at any time. Setting the DAADST bit to 1 allows synchronous D/A conversion with the ADC12 synchronous D/A conversion enable input signal from the ADC12 (unit 1). With this bit set, D/A conversion does not start until the ADC12 (unit 1) completes A/D conversion, even if the DADRm register value is changed.

Set this bit while the ADCSR.ADST bit is set to 0. Then select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1). Set bit [1] in the DAADUSR register to 1 before setting the DAADST bit to 1.

The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR12 and ELSR13 registers of the ELC. The setting of the DAADST bit is shared by channels 0 and 1 of the DAC12.

48.2.5 D/A Output Amplifier Control Register (DAAMPCR)

Address(es): DAC12.DAAMPCR 4005 E008h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAAMP 1	DAAMP 0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	DAAMP0	Amplifier Control 0	0: Do not use channel 0 output amplifier 1: Use channel 0 output amplifier.	R/W
b7	DAAMP1	Amplifier Control 1	0: Do not use channel 1 output amplifier 1: Use channel 1 output amplifier.	R/W

The DAAMPCR register selects D/A output with or without using the amplifier.

DAAMP0 bit (Amplifier Control 0)

When the DAAMP0 bit is 0, analog values are output for D/A output of channel 0 without using the amplifier. When the DAAMP0 bit is 1, analog values are output for D/A output of channel 0 through the amplifier.

When both the DAE and DAOE0 bits are 0, the amplifier is not used regardless of the setting of the DAAMP0 bit. See Table 48.4 for details.

DAAMP1 bit (Amplifier Control 1)

When the DAAMP1 bit is 0, analog values are output for D/A output of channel 1 without using the amplifier. When the DAAMP1 bit is 1, analog values are output for D/A output of channel 1 through the amplifier.

When both the DAE and DAOE1 bits are 0, the amplifier is not used regardless of the setting of the DAAMP1 bit. See Table 48.4 for details.

48.2.6 D/A Amplifier Stabilization Wait Control Register (DAASWCR)

Address(es): DAC12.DAASWCR 4005 E01C

	b7	b6	b5	b4	b3	b2	b1	b0
	DAAS W1	DAAS W0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	DAASW0	D/A Amplifier Stabilization Wait 0	0: Amplifier stabilization wait off (output) for channel 0 1: Amplifier stabilization wait on (high-Z) for channel 0.	R/W
b7	DAASW1	D/A Amplifier Stabilization Wait 1	0: Amplifier stabilization wait off (output) for channel 1 1: Amplifier stabilization wait on (high-Z) for channel 1.	R/W

The DAASWCR register controls D/A output with the output amplifier. This register is used in the initialization procedure to wait for stabilization of the D/A output amplifier. Each bit in DAASWCR should be set to 1 when both the

DAE bit and the DAOEi (i = 0, 1) bit in the DACR register are 0. See section 48.6.5, Initialization Procedure with the Output Amplifier.

DAASW0 bit (D/A Amplifier Stabilization Wait 0)

Set the DAASW0 bit to 1 in the initialization procedure to wait for stabilization of the output amplifier of D/A channel 0. When DAASW0 is set to 1, D/A conversion operates, but the conversion result D/A is not output from channel 0. When the DAASW0 bit is 0, the stabilization wait time stops, and the D/A conversion result of channel 0 is output through the output amplifier.

DAASW1 bit (D/A Amplifier Stabilization Wait 1)

Set the DAASW1 bit to 1 in the initialization procedure to wait for stabilization of the output amplifier of D/A channel 1. When DAASW1 is set to 1, D/A conversion operates, but the conversion result D/A is not output from channel 1. When the DAASW1 bit is 0, the stabilization wait time stops, and the D/A conversion result of channel 1 is output through the output amplifier.

48.2.7 D/A A/D Synchronous Unit Select Register (DAADUSR)

Address(es): DAC12.DAADUSR 4005 F0C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	AMADSEL1	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	AMADSEL1	A/D Unit 1 Select	0: Do not select unit 1 1: Select unit 1.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DAADUSR register selects the target ADC12 unit for D/A and A/D synchronous conversions. Set bit [1] to 1 to select unit 1 as the target synchronous unit for the MCU. When setting the DAADSCR.DAADST bit to 1 for synchronous conversions, select the target unit in this register in advance.

Only set the DAADUSR register while the ADCSR.ADST bit of the ADC12 is set to 0 and the DAADSCR.DAADST bit is set to 0.

48.3 Operation

The DAC12 includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEn bit (n = 0, 1) in DACR is set to 1, DAC12 is enabled and the conversion result is output.

This following is an example of D/A conversion on channel 0. Figure 48.2 shows the timing of this operation.

To process D/A conversion on channel 0:

1. Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time tDCONV elapses. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value (reference) is expressed by the following formula:

$$\frac{\text{Setting in DADRm}}{4096} \times \text{VREFH}$$

3. To start conversion again, write to DADR0 again. The conversion result is output after the conversion time tDCONV elapses.
When the DAADSCR.DAADST bit is 1 (interference prevention between D/A and A/D conversion is enabled), a

maximum of one A/D conversion time is required for D/A conversion to start. When ADCLK is faster than the peripheral clock, a longer time period might be required.

4. To disable analog output, set the DAOE0 bit to 0.

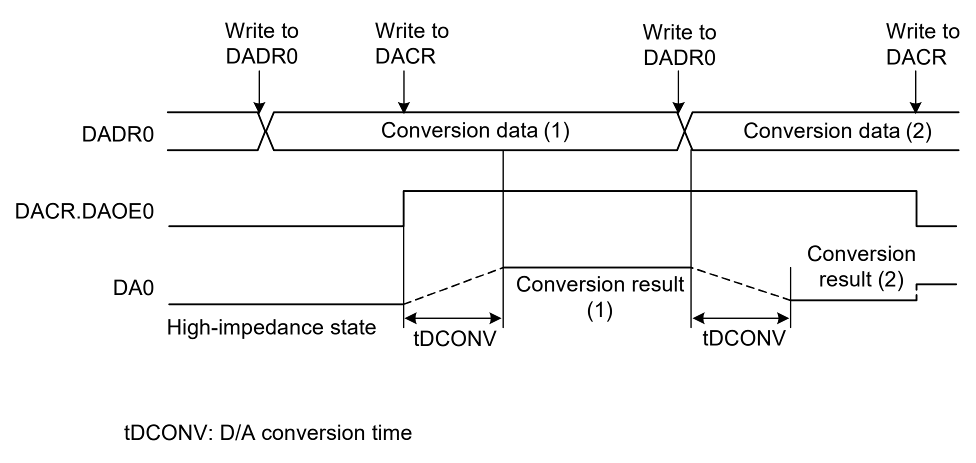


Figure 48.2 Example of DAC12 operation

48.3.1 Reducing Interference between D/A and A/D Conversion

When D/A conversion starts, the DAC12 generates inrush current. Because the same analog power supply is shared by the DAC12 and ADC12 (unit 1), the generated inrush current might interfere with ADC12 (unit 1) operation.

While the DAADSCR.DAADST bit is 1, even if the DADRm register data is changed during ADC12 (unit 1) operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion. A maximum of one A/D conversion time is required for the DADRm register data update to be reflected as the D/A conversion circuit input. Before reflection, the DADRm register value does not correspond to the analog output value.

When this function is enabled, it is impossible to check by any software means whether the DADRm register value was D/A converted. When the DADRm register data is changed while the ADC12 (unit 1) is halted, D/A conversion starts in 1 PCLKB cycle, even if DAADSCR.DAADST is 1.

The following sequence provides an example of channel 0 D/A conversion, in which the DAC12 operates synchronously with the ADC12 (unit 1).

To process D/A conversion on channel 0 in synchronization with the ADC12 (unit 1):

1. Confirm that the ADC12 (unit 1) is halted and set the DAADUSR.AMADSEL1 bit to 1.
2. Confirm that the ADC12 (unit 1) is halted and set the DAADSCR.DAADST bit to 1.
3. Confirm that the ADC12 (unit 1) is halted and set the DACR.DAOE0 bit to 1.
4. Set the DADR0 register. When ADCLK is faster than the peripheral clock, a period longer than one A/D conversion time might be required.

- If the 12-bit A/D conversion is halted (ADCSR.ADST bit = 0) when the DADR0 register is changed, D/A conversion starts in 1 PCLKB cycle.
- If the 12-bit A/D conversion is in progress (ADCSR.ADST bit = 1) when the DADR0 register is changed, D/A conversion starts on A/D conversion completion. If the DADR0 register is changed twice during A/D conversion, the first update might not be converted.

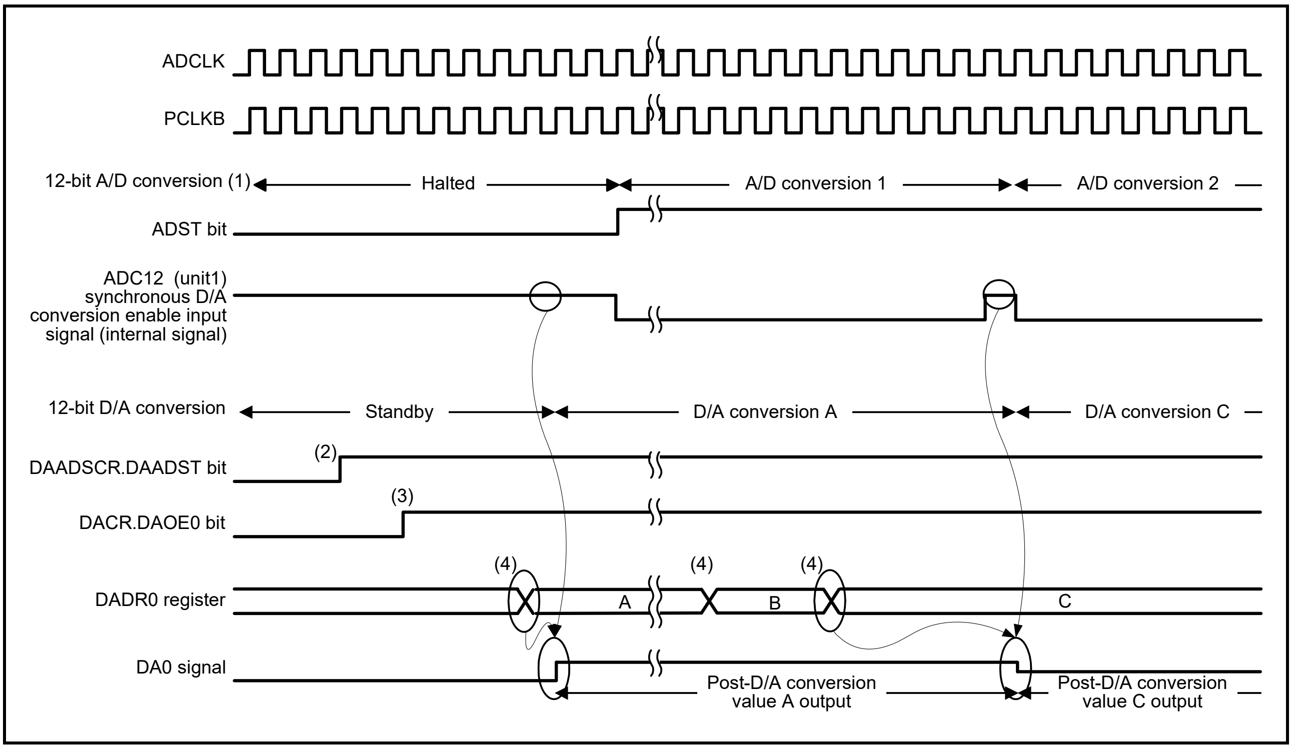


Figure 48.3 Example of conversion when the DAC12 is synchronized with the ADC12

When ADCLK is faster than PCLKB, the DAC12 might not be able to capture an ADC12 (unit 1) synchronous D/A conversion enable input signal for the 1 ADCLK cycle that is output between A/D conversion 1 and A/D conversion 2. Figure 48.4 shows an example of this. In this case, post-D/A conversion value A is continuously output as the DA0 signal.

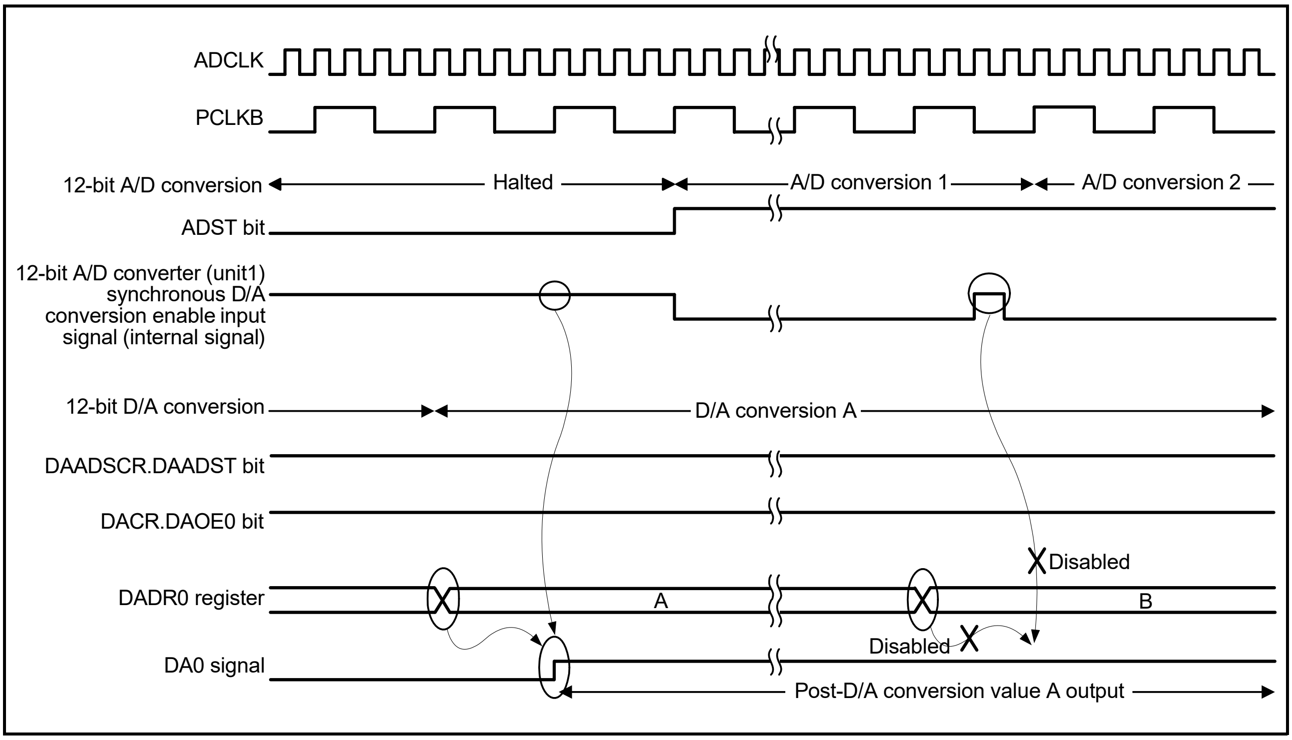


Figure 48.4 Example when the DAC12 cannot capture the synchronous D/A conversion enable input signal from the ADC12 (unit 1)

48.4 Event Link Operation Setting Procedure

This section describes the procedures used in event link operation.

48.4.1 DA0 Event Link Operation Setting Procedure

To set up DA0 event link operation:

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
2. Set the ELC_DA0 event signal to be linked to each peripheral module in the ELSR12 register.
3. Set the ELCR.ELCON bit to 1. This enables event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit sets to 1, and D/A conversion starts on channel 0.
5. Set the ELSR12.ELS[8:0] bits to 000h to stop event link operation on DAC12 channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

48.4.2 DA1 Event Link Operation Setting Procedure

To set up DA1 event link operation:

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR1 register.
2. Set the ELC_DA1 event signal to be linked to each peripheral module in the ELSR13 register.
3. Set the ELCR.ELCON bit to 1. This enables event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE1 bit sets to 1, and D/A conversion starts on channel 1.
5. Set the ELSR13.ELS[8:0] bits to 000h to stop event link operation on DAC12 channel 1. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

48.5 Usage Notes on Event Link Operation

- When the event link function is used, do not use the amplifier output function.
- When the event link function is used, set the DACR.DAE bit to 0.
- When the event specified for the ELC_DA0 event signal is generated while the write cycle is performed on the DACR.DAOE0 bit, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- When the event specified for the ELC_DA1 event signal is generated while the write cycle is performed on the DACR.DAOE1 bit, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 to reduce interference between D/A and A/D conversions.

48.6 Usage Notes

48.6.1 Settings for the Module-Stop Function

DAC12 operation can be disabled or enabled using the Module Stop Control Register. The DAC12 is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

48.6.2 DAC12 Operation in the Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in the module-stop state, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

48.6.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in Software Standby mode, disable D/A conversion by setting the DACR.DAOE1, DAOE0, and DAE bits to 0.

48.6.4 Constraint on Entering Deep Software Standby Mode

When the MCU enters Deep Software Standby mode with D/A conversion enabled, the outputs of the DAC12 are placed in a high impedance state.

48.6.5 Initialization Procedure with the Output Amplifier

Use the following initialization procedures with the output amplifier. The example shows the case for channel 0.

To initialize the DAC12 with the output amplifier:

1. Write 000h to the DADR0 register.
2. Set the DAASWCR.DAASW0 bit to 1.
3. Set the DAAMPCR.DAAMP0 bit to 1.
4. Set the DACR.DAE bit or the DACR.DAOE0 bit to 1 to start operation of the amplifier.
5. Clear the DAASWCR.DAASW0 bit to 0 after waiting D/A conversion time tCONV.
6. Write the value to be converted in the DADR0 register.

While the amplifier is running, clearing the DACR.DAE and DACR.DAOE0 bits to 0 allows the amplifier to stop operation. To use the amplifier again, repeat procedures 1 to 6.

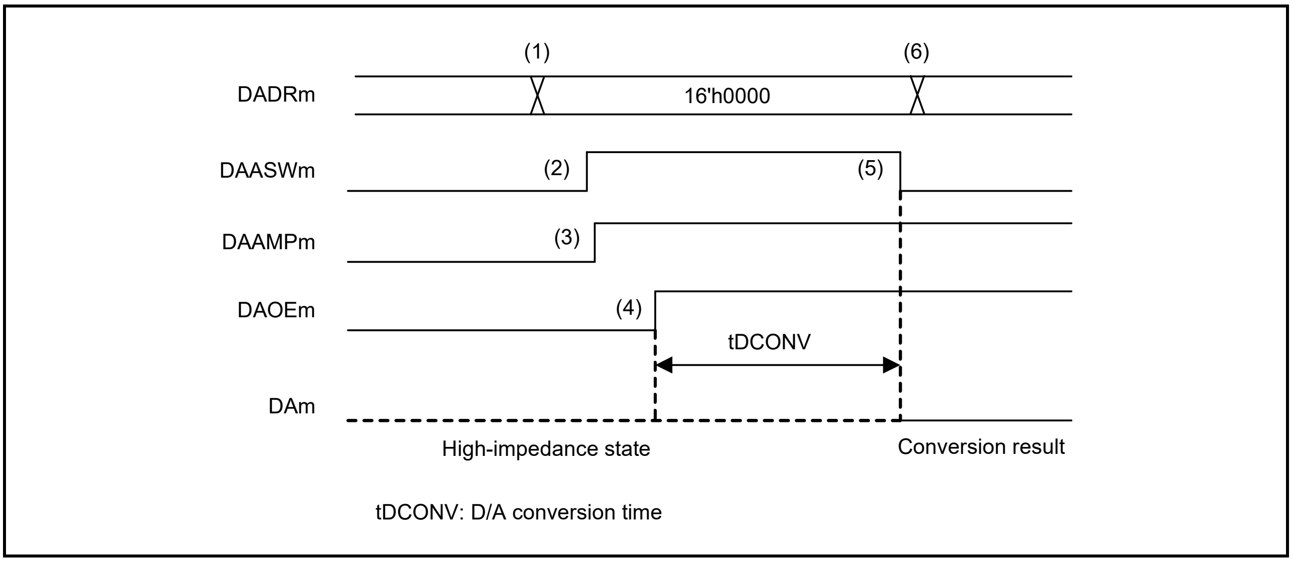


Figure 48.5 Example of the initial flow with the output amplifier in DAC12

48.6.6 Constraints on Usage When Interference Prevention between D/A and A/D Conversion Is Enabled

When the DAADSCR.DAADST bit is 1 (interference prevention between D/A and A/D conversion is enabled), do not place the ADC12 in the module-stop state. This might halt D/A conversion in addition to A/D conversion.

49. Temperature Sensor (TSN)

49.1 Overview

The on-chip temperature sensor can be used to determine and monitor the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can also be used by the end application. Table 49.1 lists the temperature sensor specifications and Figure 49.1 shows a block diagram.

Table 49.1 Temperature sensor specifications

Parameter	Specifications
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-Bit A/D Converter (ADC12).
Module-stop function	Module-stop state can be set to reduce power consumption
Temperature sensor calibration data	Reference data measured for each MCU at factory shipment is stored

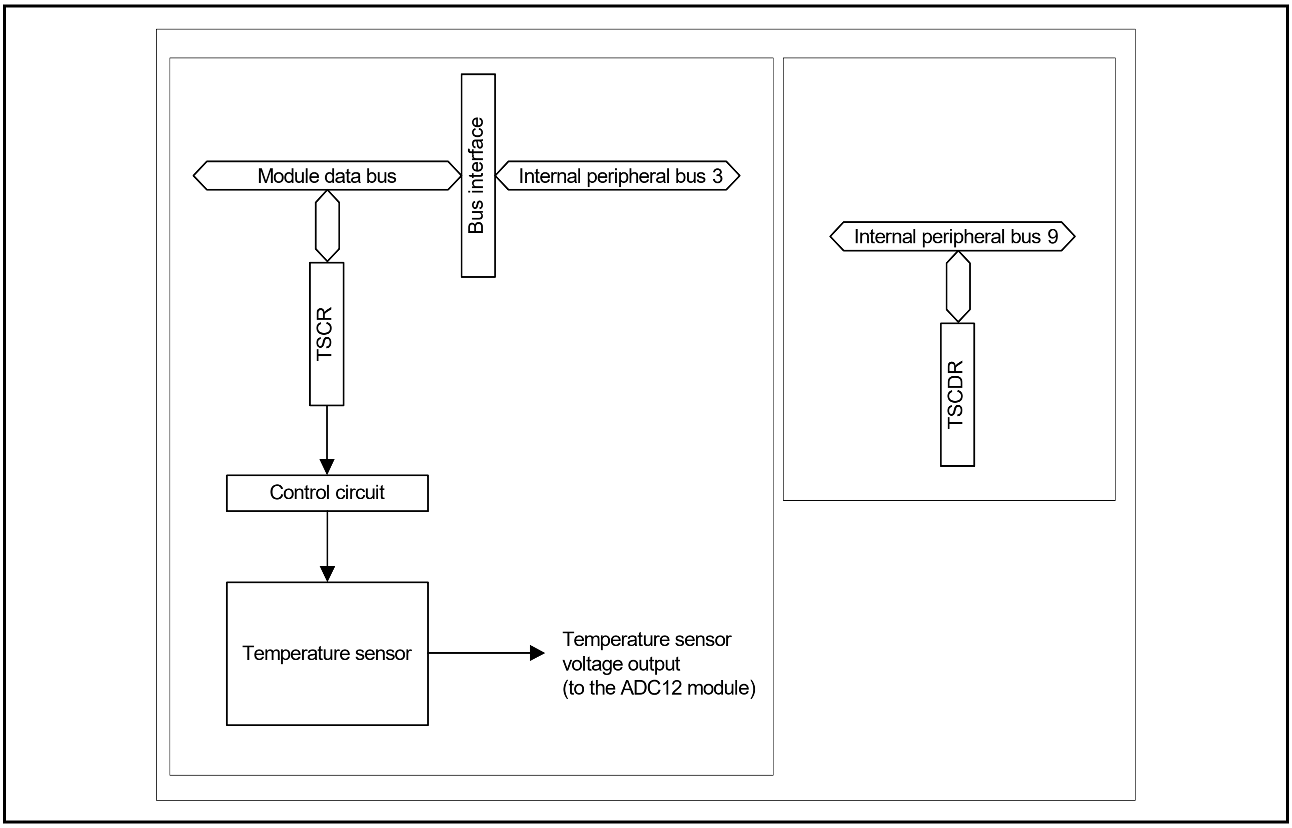


Figure 49.1 Temperature sensor block diagram

49.2 Register Descriptions

49.2.1 Temperature Sensor Control Register (TSCR)

Address(es): TSN.TSCR 4005 D000h

	b7	b6	b5	b4	b3	b2	b1	b0
	TSEN	—	—	TSOE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	**Symbol**	**Bit name**	**Description**	**R/W**
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TSOE	Temperature Sensor Output Enable	0: Disable output from the temperature sensor to the ADC12 1: Enable output from the temperature sensor to the ADC12.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TSEN	Temperature Sensor Enable	0: Stop the temperature sensor 1: Start the temperature sensor.	R/W

The timing constraints shown in Figure 49.3 apply to the settings of the TSCR register.

49.2.2 Temperature Sensor Calibration Data Register(TSCDR)

Address(es): TSD.TSCDR 407F B17Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	Unique value for each MCU											

The TSCDR register stores temperature sensor calibration data measured for each MCU at factory shipment. Temperature sensor calibration data is a digital value obtained using the 12-bit A/D converter unit 0 to convert the voltage output by the temperature sensor under the condition Ta = Tj = 127°C and AVCC0 = 3.3 V.

The TSCDR register is a 32-bit read-only register and should be read in 32-bit units.

49.3 Using the Temperature Sensor

The temperature sensor outputs a voltage which varies with the temperature. This voltage is converted to a digital value by the ADC12. You can then obtain the die temperature by converting the value into the temperature.

49.3.1 Preparation for Using the Temperature Sensor

The temperature (T) is proportional to the sensor voltage output (Vs), so temperature is calculated with the following formula:

$T = (Vs - V1) / Slope + T1$

T: Measured temperature (°C)

Vs: Voltage output by the temperature sensor when temperature is measured (V)

T1: Temperature experimentally measured at one point (°C)

V1: Voltage output by the temperature sensor when T1 is measured (V)

T2: Temperature at the experimental measurement of another point (°C)

V2: Voltage output by the temperature sensor when T2 is measured (V)

Slope: Temperature gradient of the temperature sensor (V / °C), Slope = (V2 - V1) / (T2 - T1)

Determine the values for formula parameter (V1, T1, Slope) measurement. These values vary from sensor to sensor, and Renesas recommends making the following experimental measurement at two different temperatures to determine the values for these parameters:

1. Use the ADC12 to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Use the ADC12 to measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature gradient (Slope = (V2 - V1) / (T2 - T1)) from these results.
3. Obtain subsequent temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs - V1) / Slope + T1).

If you are using the temperature slope given in Table 60.45 of section 60, Electrical Characteristics, use the 12-bit A/D converter unit 0 to measure the voltage V1 output by the temperature sensor at temperature T1, then calculate the temperature characteristic by using the following formula:

T = (Vs - V1) / Slope + T1

T: Measured temperature (°C)

Vs: Voltage output by the temperature sensor when the temperature is measured (V)

T1: Sample temperature measurement at first point (°C)

V1: Voltage output by the temperature sensor when T1 is measured (V)

Slope: Temperature slope given in Table 60.45 ÷ 1000 (V / °C)

In this MCU, the TSCDR register stores the temperature value (CAL127) of the temperature sensor measured under the condition Ta = Tj = 127°C and AVCC0 = 3.3 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.

If V1 is calculated from CAL127:

V1 = 3.3 × CAL127 / 4096 [V]

Using this, the measured temperature can be calculated according to the following formula:

T = (Vs - V1) / Slope + 127 [°C]

T: Measured temperature (°C)

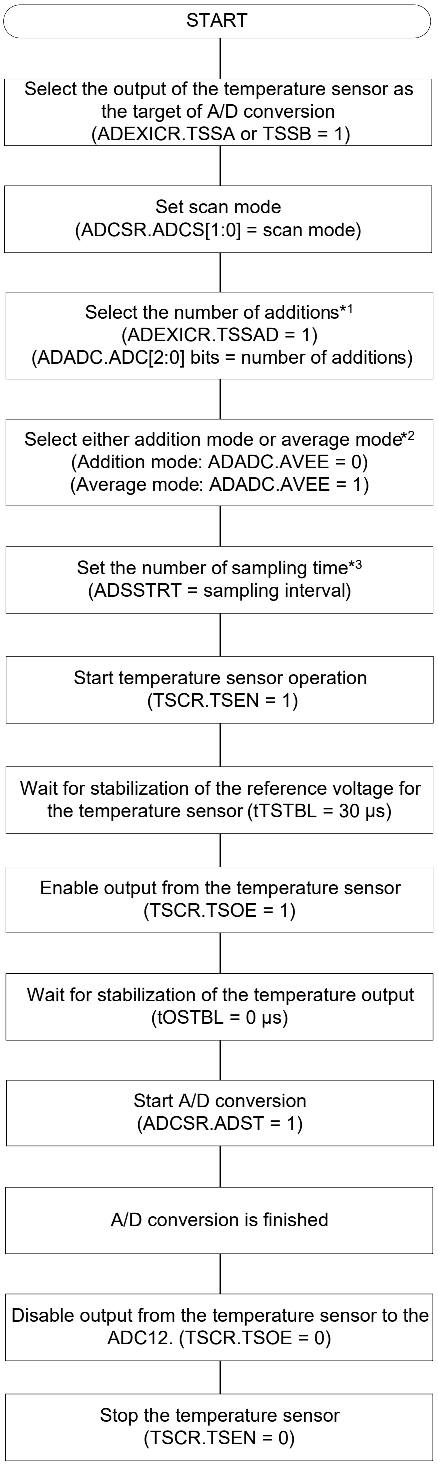
Vs: Voltage output by the temperature sensor when the temperature is measured (V)

V1: Voltage output by the temperature sensor when Ta = Tj = 127°C and AVCC0 = 3.3 V (V)

Slope: Temperature slope given in Table 60.45 ÷ 1000 (V / °C)

49.3.2 Procedures for Using the Temperature Sensor

Figure 49.2 shows the procedure for using the temperature sensor. For the procedure to configure the ADC12, see section 47, 12-Bit A/D Converter (ADC12).



Note 1. The ADADC.ADC[2:0] setting is limited to some values in average mode. For details on available ADADC.ADC[2:0] settings in average mode, see section 47.2.10, A/D-Converted Value Addition/Average Count Select Register (ADADC).

Note 2. This setting is not required if addition/average mode is not set.

Note 3. Set the sampling time to more than the value described in section 60.7, TSN Characteristics.

Figure 49.2 Procedure for using the temperature sensor

Figure 49.3 shows the timing from the start of temperature sensor operation until the completion of A/D conversion when the ADC12 is in single scan mode (the conversion target is the temperature sensor output only). The times shown in the figure are described in Table 49.2.

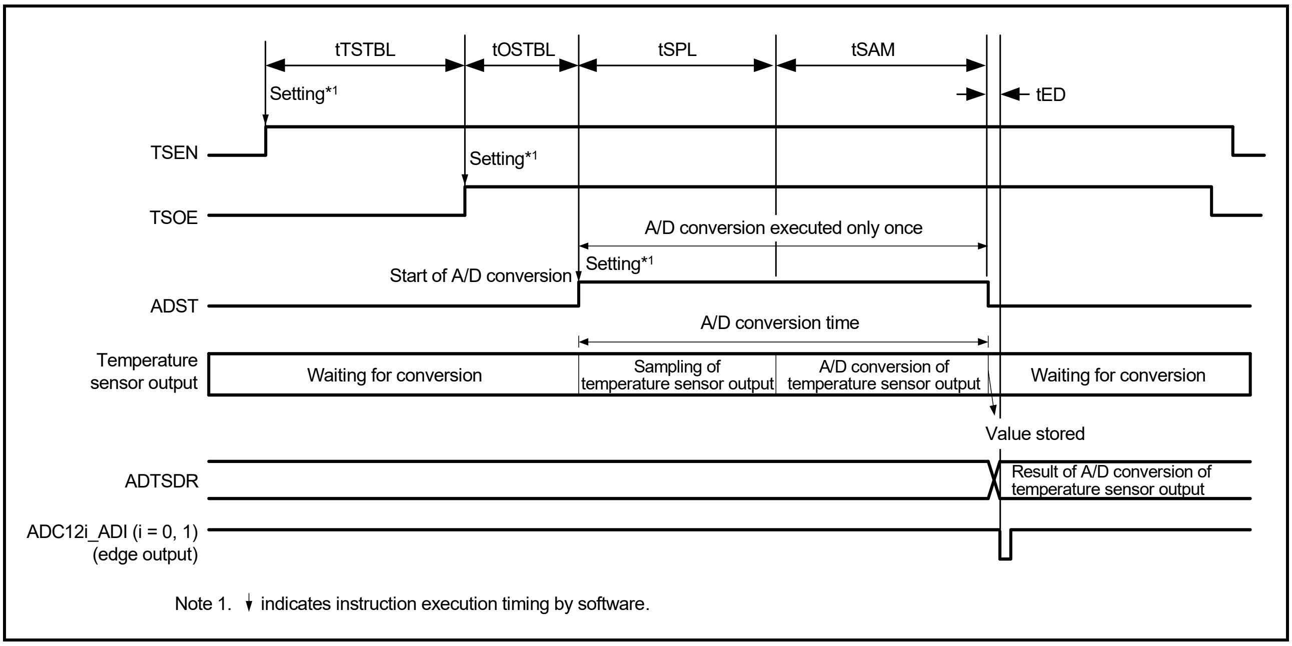


Figure 49.3 Timing from start of temperature sensor operation until completion of A/D conversion

Table 49.2 Time until completion of A/D conversion after start of temperature sensor operation

Parameter	Symbol	Time
Wait time for temperature sensor reference voltage stabilization	tTSTBL	30 µs (min)
Wait time for temperature sensor output stabilization	tOSTBL	0 µs (min)
ADC12 input sampling time	tSPL	ADSSTRT setting × ADCLK cycles
A/D conversion time	tSAM	See Table 47.10, Conversion times during scanning (in numbers of cycles of ADCLK and PCLKB)
Scan conversion end delay	tED	

49.4 Usage Notes

49.4.1 Settings for the Module-Stop Function

Temperature sensor operation can be disabled or enabled using the associated bit in Module Stop Control Register D (MSTPCRD). The temperature sensor is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

49.4.2 Constraints

It is prohibited to use both channels of the ADC12 simultaneously for temperature sensor measurement.

50. High-Speed Analog Comparator (ACMPHS)

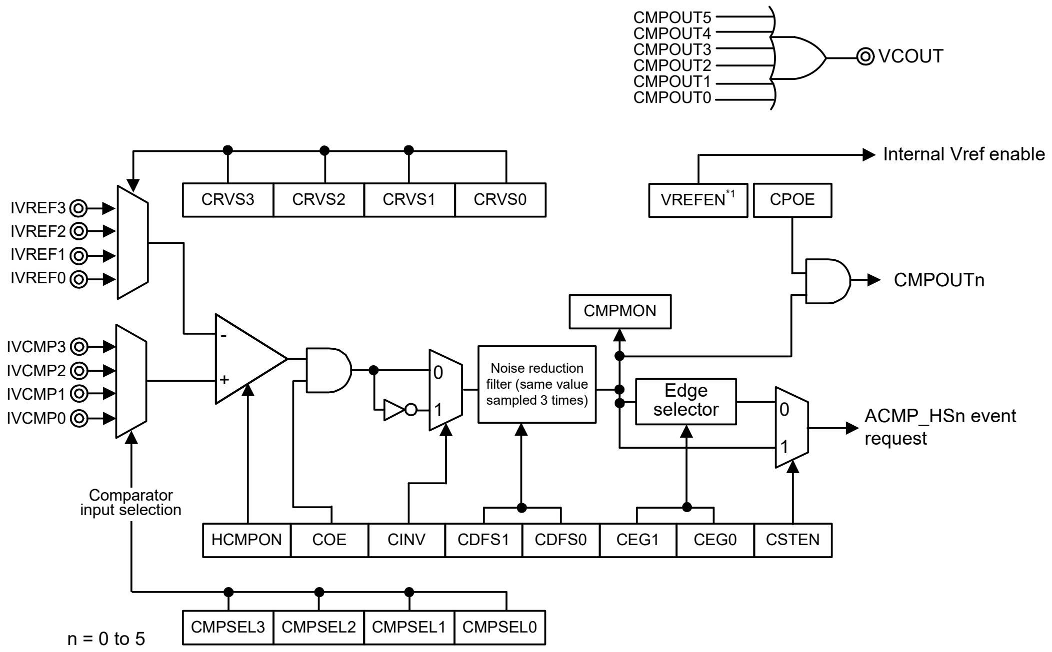
50.1 Overview

The High-Speed Analog Comparator (ACMPHS) can be used to compare a test voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the test voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output and internal reference voltage) and an external source (with or without an internal PGA). Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion.

Table 50.1 lists the ACMPHS specifications, Figure 50.1 shows a block diagram, and Table 50.2 shows the input source configurations.

Table 50.1 ACMPHS 0 to 5 specifications

Parameter	Specifications
Number of channels	6 channels: ACMPHS 0 to ACMPHS 5
Analog input voltage	• Output from internal PGA • Output from internal D/A converter • Input from internal A/D converter input pin (one selectable).
Reference voltage	• Internal reference voltage (Vref) • Output from internal D/A converter • Input from internal A/D converter input pin (one selectable).
ACMPHS output	• Comparison result • Generation of ELC event output • Monitor output from register.
Interrupt request signal	• Interrupt request generated on valid edge detection from comparison result • Selectable to rising edge, falling edge, or both edges.
Digital filter function	• Selectable to one of three sampling frequencies • Not using the filter function is selectable.



CMPSEL3 to CMPSEL0: Bits in CMPSEL0 register
CRVS3 to CRVS0: Bits in CMPSEL1 register
CPOE, VREFEN: Bits in CPIOC register
CMPMON: Bit in CMPMON register
HCMPON, COE, CINV, CDFS1, CDFS0, CEG1, CEG0, CSTEN: Bits in CMPCTL register

Note: VREFEN exists only in ACMPHS0.

Figure 50.1 ACMPHS block diagram

Table 50.2 Input source configuration of the ACMPHS

	Reference voltage input source				Analog voltage input source				
Comparator	**IVREF3**	**IVREF2**	**IVREF1**	**IVREF0**	**IVCMP3**	**IVCMP2**	**IVCMP1**	**IVCMP0**	**Output pin**
ACMPHS0	DA0*1	Vref*2	AN116	AN016	PGA0 output*6	AN000*3, *6	DA1*4	AN017	VCOUT*5
ACMPHS1	DA0*1	Vref*2	AN116	AN016	PGA1 output*6	AN001*3, *6	DA1*4	AN017	
ACMPHS2	DA0*1	Vref*2	AN116	AN016	PGA2 output*6	AN002*3, *6	DA1*4	AN017	
ACMPHS3	DA0*1	Vref*2	AN116	AN016	PGA3 output*6	AN100*3, *6	DA1*4	AN017	
ACMPHS4	DA0*1	Vref*2	AN116	AN016	PGA4 output*6	AN101*3, *6	DA1*4	AN017	
ACMPHS5	DA0*1	Vref*2	AN116	AN016	PGA5 output*6	AN102*3, *6	DA1*4	AN017	

Note 1. When D/A converter 0 output is not used, the signal can be used as AN005/AN105 analog input.
Note 2. Internal voltage reference.
Note 3. Because input is through PGA, the associated Module Stop bit, MSTPCRD.MSTPD16 (unit 0) or MSTPCRD.MSTPD15 (unit 1) should be set to 0.
Note 4. When D/A converter 1 output is not used, the signal can be used as AN006/AN106 analog input.
Note 5. ACMPHS0 to ACMPHS5 compare outputs are bundled with the VCOUT pin.
Note 6. Setting of the ADC12 is required. For details, see section 47.6.8, Available Functions and Register Settings of AN000 to AN002, AN007, AN100 to AN102, and AN107.

50.2 Register Descriptions

50.2.1 Comparator Control Register (CMPCTL)

Address(es): ACMPHS0.CMPCTL 4008 5000h,ACMPHS1.CMPCTL 4008 5100h, ACMPHS2.CMPCTL 4008 5200h,
ACMPHS3.CMPCTL 4008 5300h, ACMPHS4.CMPCTL 4008 5400h, ACMPHS5.CMPCTL 4008 5500h

	b7	b6	b5	b4	b3	b2	b1	b0
	HCMPON	CDFS[1:0]		CEG1	CEG0	CSTEN	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CINV	Comparator Output Polarity Selection*2, *3	0: Do not invert comparator output 1: Invert comparator output.	R/W
b1	COE	Comparator Output Enable	0: Disable comparator output (output signal is low level) 1: Enable comparator output.	R/W
b2	CSTEN	Interrupt Select*1	0: Output through the edge selector 1: Output directly.	R/W
b4, b3	CEG1/CEG0	Selection of Valid Edge (Edge Selector)	b4 b3 0 0: Do not detect edge 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges.	R/W
b6, b5	CDFS[1:0]	Noise Filter Selection*1, *2, *3, *4	b6 b5 0 0: Do not use noise filter. 0 1: Use noise filter sampling frequency of PCLKB/2^3 1 0: Use noise filter sampling frequency of PCLKB/2^4 1 1: Use noise filter sampling frequency of PCLKB/2^5.	R/W
b7	HCMPON	Comparator Operation Control*5	0: Stop operation (comparator outputs a low-level signal) 1: Enable operation (enables input to the comparator pins).	R/W

Note 1. Set the CSTEN bit to 1 and the CDFS[1:0] bits to 00b if the ACMPHS interrupt causes release of Software Standby or Snooze modes. CTSEN is supported only by the ACMPHS0. ACMPHSn.CMPCTL.CTESN (n = 1 to 5) must be set to 0.

Note 2. Disable the ACMPHS output (COE = 0) before changing the CDFS[1:0] and CINV bits.

Note 3. If the CDFS[1:0] and CINV bits are changed, an ACMPHS interrupt request and an ELC event might be generated. Before changing these bits, set the ELSRn register to 0 (the ACMPHS output is not linked). After changing these bits, clear the IR flag in the IELSRn register to 0 to clear the interrupt status.

Note 4. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), perform sampling four times and update the filter output, and then use the ACMPHS interrupt request or the ELC event.

Note 5. A stabilization wait time is required to permit ACMPHS operation after enabling it (HCMPON = 1). The operation stabilization wait time for ACMPHS modules 0 to 5 is 300 ns.

The CMPCTL register controls the ACMPHS operation, enables or disables the ACMPHS output, selects the noise filter, selects the valid edge of the interrupt signal, and selects the interrupt. A reset clears this register to 00h.

50.2.2 Comparator Input Select Register (CMPSEL0)

Address(es): ACMPHS0.CMPSEL0 4008 5004h, ACMPHS1.CMPSEL0 4008 5104h, ACMPHS2.CMPSEL0 4008 5204h, ACMPHS3.CMPSEL0 4008 5304h, ACMPHS4.CMPSEL0 4008 5404h, ACMPHS5.CMPSEL0 4008 5504h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CMPSEL[3:0]	Comparator Input Selection*1	b3 b0 0 0 0 0: Do not input 0 0 0 1: Select IVCMP0*2 0 0 1 0: Select IVCMP1*2 0 1 0 0: Select IVCMP2*2 1 0 0 0: Select IVCMP3.*2 Other settings are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use the following procedure to change the CMPSEL[3:0] bits. Writing a value other than 0000 0000b while the value of the CMPSEL0 register is not 0000 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CMPSEL[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL0 register to 0000 0000b.
3. Set a new value in the CMPSEL[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 50.2.

50.2.3 Comparator Reference Voltage Select Register (CMPSEL1)

Address(es): ACMPHS0.CMPSEL1 4008 5008h, ACMPHS1.CMPSEL1 4008 5108h, ACMPHS2.CMPSEL1 4008 5208h, ACMPHS3.CMPSEL1 4008 5308h, ACMPHS4.CMPSEL1 4008 5408h, ACMPHS5.CMPSEL1 4008 5508h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CRVS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CRVS[3:0]	Reference Voltage Selection*1	b3 b0 0 0 0 0: Do not input 0 0 0 1: Select IVREF0*2 0 0 1 0: Select IVREF1*2 0 1 0 0: Select IVREF2*2 1 0 0 0: Select IVREF3*2 Other settings are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use the following procedure to change the CRVS[3:0] bits. Writing a value other than 0000 0000b while the value of the CMPSEL1 register is not 0000 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CRVS[3:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL1 register to 0000 0000b.
3. Set a new value to the CRVS[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns)
5. Set the CMPCTL.COE bit to 1.

6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 50.2.

50.2.4 Comparator Output Monitor Register (CMPMON)

Address(es): ACMPHS0.CMPMON 4008 500Ch, ACMPHS1.CMPMON 4008 510Ch, ACMPHS2.CMPMON 4008 520Ch, ACMPHS3.CMPMON 4008 530Ch, ACMPHS4.CMPMON 4008 540Ch, ACMPHS5.CMPMON 4008 550Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPMON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPMON	Comparator Output Monitor*1	0: Comparator output is low 1: Comparator output is high.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

Note 1. When ACMPHS operation is enabled (HCMPON = COE = 1) but the noise filter is not in use (CDFS[1:0] = 00b), design the software so that the CMPMON bit is read twice and the values are only used after the two consecutive values match.

50.2.5 Comparator Output Control Register (CPIOC)

Address(es): ACMPHS0.CPIOC 4008 5010h, ACMPHS1.CPIOC 4008 5110h, ACMPHS2.CPIOC 4008 5210h, ACMPHS3.CPIOC 4008 5310h, ACMPHS4.CPIOC 4008 5410h, ACMPHS5.CPIOC 4008 5510h

	b7	b6	b5	b4	b3	b2	b1	b0
	VREFEN	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CPOE	Comparator Output Selection	0: Disable VCOUT pin output of the comparator (output signal is low) 1: Enable VCOUT pin output of the comparator.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	VREFEN	Internal Vref Enable*1	0: Disable internal Vref 1: Enable internal Vref.	R/W

Note 1. For ACMPHS modules 0 to 5, VREFEN exists only in ACMPHS0.CPIOC. When using the internal Vref in COMP0 to COMP5, set the VREFEN bit in ACMPHS0.CPIOC to 1. Bit [7] in ACMPHS1.CPIOC to ACMPHS5.CPIOC should be 0 regardless of whether or not the internal Vref is used.

50.3 Operation

The ACMPHS compares a reference voltage to an analog input voltage. Operation is not guaranteed when the values of registers are changed during ACMPHS operation. Table 50.3 shows the procedures for setting the registers associated with ACMPHS.

Table 50.3 Procedure for setting registers associated with ACMPHSn (n = 0 to 5) (1 of 2)

Step	Register	Bit	Setting
1	Associated MSTPCRD register	MSTPD28 to MSTPD23	0: Input clock supply.
2	Associated Port mn Pin Function Select register (PmnPFS)	ASEL	1: Select the function of pins IVREF and IVCMP.
3	ACMPHS0.CPIOC	VREFEN	1: When using the internal Vref.

Table 50.3 Procedure for setting registers associated with ACMPHSn (n = 0 to 5) (2 of 2)

Step	Register	Bit	Setting
4	Associated D/A convertor		When using the D/A convertor, select in the register.
5	CMPSEL0, CMPSEL1	CMPSEL0 to CMPSEL3, CRVS0 to CRVS3	Select the ACMPHSn input, with 1 set in only one of the bits.
6	CMPCTL	CDFS[1:0], CEG1, CEG0, and CINV	Set up ACMPHSn control.
		HCMPON	1: Enable ACMPHSn operation.
7	Waiting for the ACMPHS stabilization time (minimum 300 ns).		
8	CMPCTL	COE	1: Enable ACMPHSn output.
9	CPIOC	CPOE	Set the VCOUT output
	Associated Port mn Pin Function Select register (PmnPFS)	PSEL, PMR	Select the VCOUT port function.
10	IELSRn	IR, IELS[8:0]	When using an interrupt, select the interrupt status flag and the ICU event link.*1
11	ELSRn	ELS[8:0]	When using an ELC, select the event link.*2
12	Operation started		
13	CMPCTL	COE	0: When changing IVREF or IVCMP, to disable ACMPHSn output.
14	CMPSEL1	CRVS0 to CRVS3	Change the CMPSEL1 bits as follows: 1. Set bits CMPSEL1 to 0000 0000b. 2. Set a new value to the CMPSEL1 bits, with 1 set in only one of the bits.
	CMPSEL0	CMPSEL0 to CMPSEL3	Change the CMPSEL0 bits as follows: 1. Set bits CMPSEL0 to 0000 0000b. 2. Set a new value to the CMPSEL0 bits, with 1 set in only one of the bits.
15	Waiting for the ACMPHS switching stabilization time (minimum 200 ns).		
16	CMPCTL	COE	1: Enable ACMPHSn output.
17	Operation restarted		

Note 1. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the interrupt flag.
Note 2. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the event link select.

Figure 50.2 shows an example of ACMPHS operation. The VCOUT output becomes 1 when the analog input voltage is higher than the ACMPHS reference input voltage, and the VCOUT output becomes 0 when the analog input voltage is lower than the reference voltage. When the ACMPHS output changes, an interrupt request and an ELC event are output.

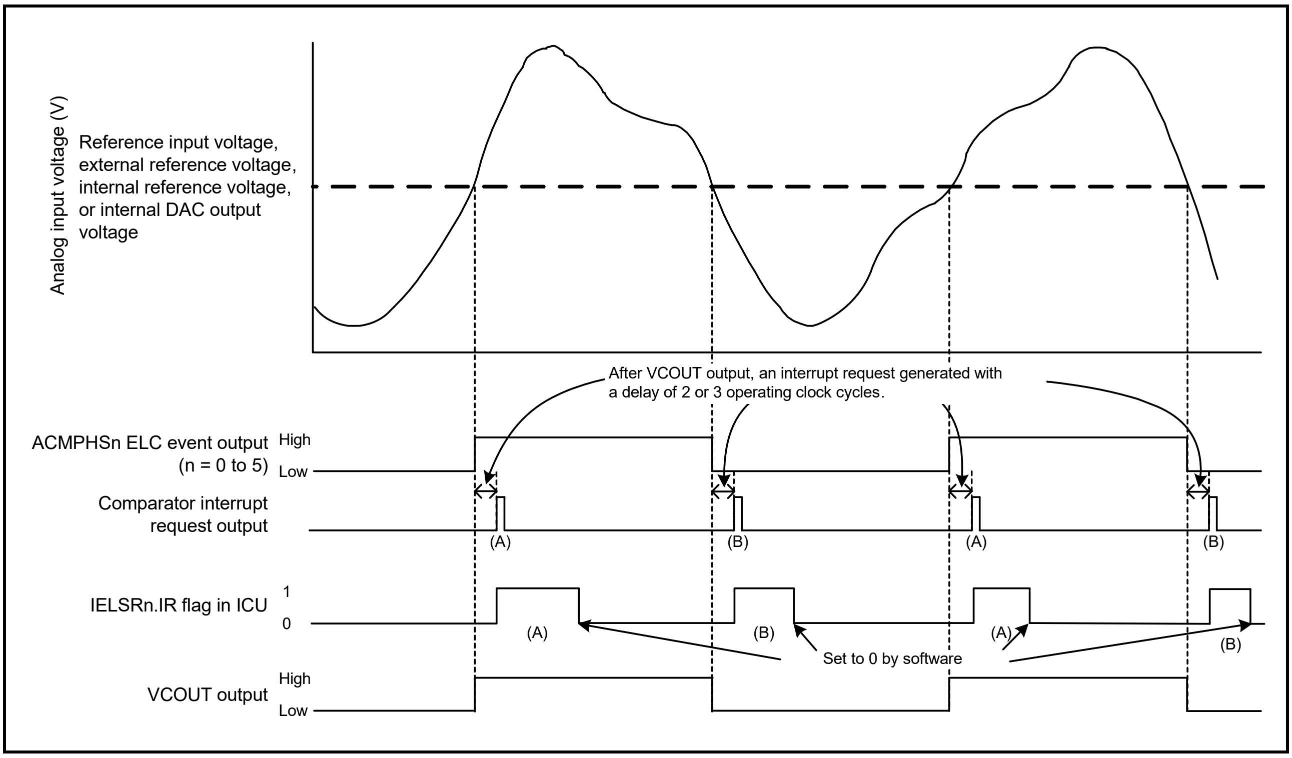


Figure 50.2 ACMPHS operation example

Figure 50.2 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), and CEG1 = CEG0 = 1 (both-edge detection selected). When CINV = 0, CEG0 = 1, and CEG1 = 0 (rising-edge detection selected for non-inversion output signal from the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG0 = 0, and CEG1 = 1 (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only. When CPOE = 1, VCOUT directly outputs the ELC event output.

50.4 Noise Filter

The ACMPHS contains a noise filter. The sampling clock can be selected in the CMPCTL.CDFS[1:0] bits. The ACMPHS signal is sampled every sampling clock, and if the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the ACMPHS output.

Figure 50.3 shows the configuration of the noise filter and edge detector, and Figure 50.4 shows an example of noise filter and interrupt operation.

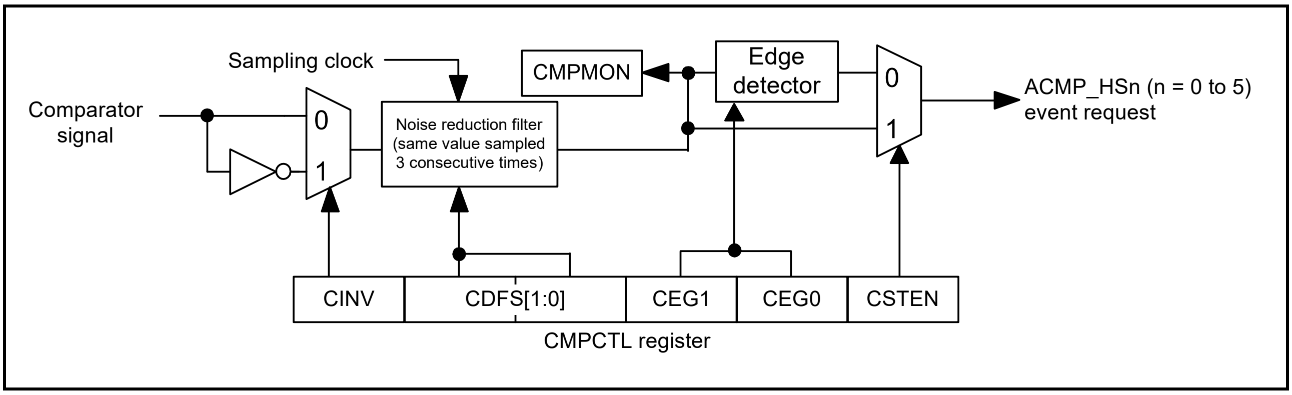


Figure 50.3 Noise filter and edge detection configuration

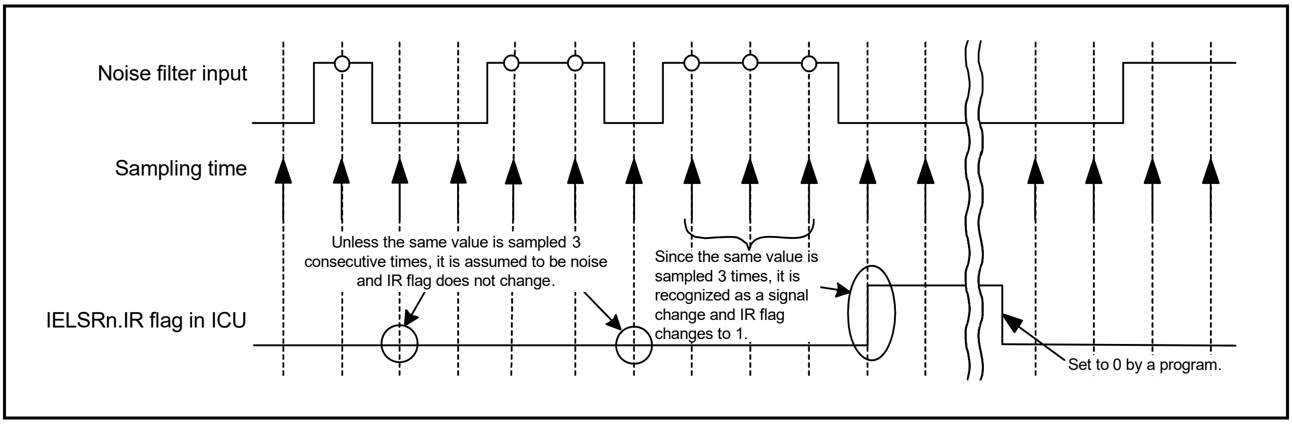


Figure 50.4 Noise filter and interrupt operation example

The operation example in Figure 50.4 applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).

50.5 ACMPHS Interrupts

The ACMPHS generates six interrupt requests from sources ACMPHS0 to ACMPHS5. To use an ACMPHS interrupt, select it in the IELSR register in the Interrupt Controller Unit (ICU). Select the interrupt request in the CMPCTL.CSTEN bit, either through the edge selector, or not.

When using the ACMPHS interrupt through the edge selector, set at least one of the CMPCTL.CEG0 and CMPCTL.CEG1 bits to 1 (to a value other than 00b for no edge selection). In most cases, set the CMPCTL.CSTEN bit to 0 (output through the edge selector). Setting this bit to 1 is only permitted to release Software Standby or Snooze mode.

To use the ACMPHS interrupt in Software Standby or Snooze mode, set the CMPCTL.CSTEN bit to 1 (direct output), set the CMPCTL.CDFS[1:0] bits to 00b (digital noise filter not used), and set CMPCTL.CINV as follows:

- When detecting compare result 1 to 0, set CMPCTL.CINV to 0 (comparator output not inverted)
- When detecting compare result 0 to 1, set CMPCTL.CINV to 1 (comparator output inverted).

An ACMPHS0 interrupt request can be used to release Software Standby or Snooze modes. (ACMPHS1 to ACMPHS5 cannot be used.)

For details on the register settings related to ACMPHS interrupt requests, see section 50.2.1, Comparator Control Register (CMPCTL).

50.6 ACMPHS Output to the Event Link Controller (ELC)

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ACMPHS ELC event, select them in the ELSRn register in the ELC. When using the ELC event request, set the CMPCTL.CSTEN bit to 0 (output through the edge selector). Also set at least one of the CMPCTL.CEG0 and CMPCTL.CEG1 bits to 1 (to a value other than 00b for no edge selection).

50.7 ACMPHS Pin Output

The comparison result from the ACMPHS can be output to external pins. Use the CMPCTL.CINV and CPIOC.CPOE bits to set the output polarity (non-inverted or inverted output) and enable or disable output. To output the ACMPHS comparison result to the VCOUT output pin, set the associated Port mn Pin Function Select register (PmnPFS) in the I/O register.

50.8 Usage Notes

50.8.1 Settings for the Module-Stop Function

ACMPHS operation can be disabled or enabled using the Module Stop Control Register. The ACMPHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

50.8.2 Relationship with the ADC12

Constraints apply on the simultaneous use of ACMPHS analog input and ADC12 analog input. For details, see section 47, Relationship between ADC12 Units 0 and 1 and the ACMPHS.

51. Capacitive Touch Sensing Unit (CTSU)

51.1 Overview

The Capacitive Touch Sensing Unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software that enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with an electrical insulator so that a finger does not come into direct contact with the electrode.

As Figure 51.1 shows, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding insulators. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the value of electrostatic capacitance increases.

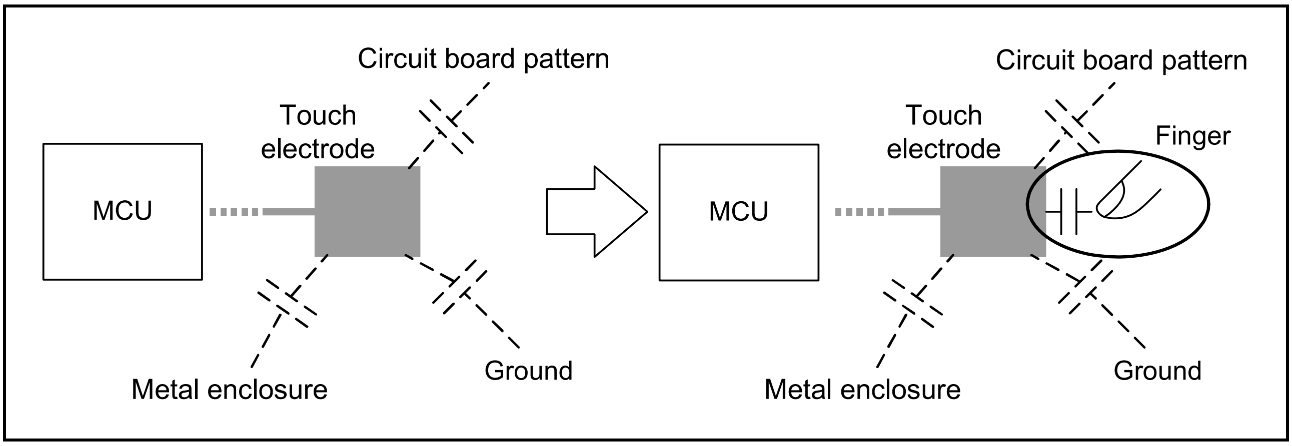


Figure 51.1 Increased electrostatic capacitance because of the presence of a finger

Electrostatic capacitance is detected by the self-capacitance and mutual capacitance methods. In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

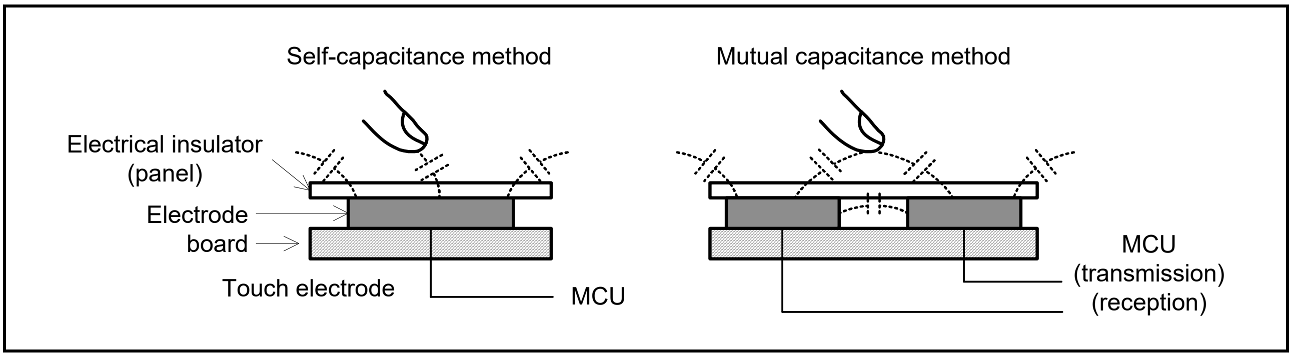


Figure 51.2 Self-capacitance and mutual capacitance methods

Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged or discharged current, for a specified period. For details on the measurement principles of the CTSU, see section 51.3.1, Principles of Measurement Operation. Table 51.1 lists the CTSU specifications and Figure 51.3 shows a block diagram.

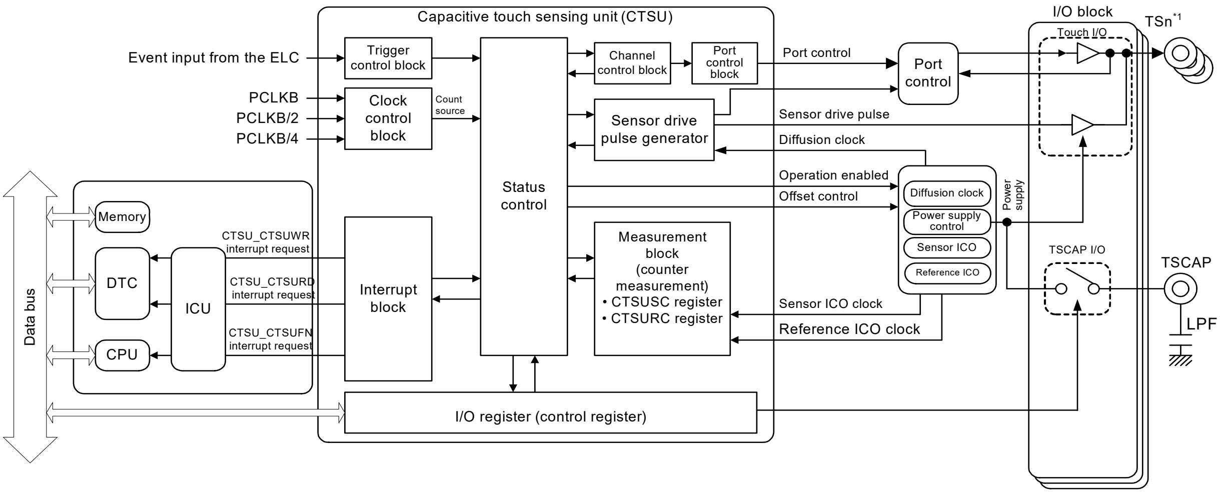
Table 51.1 CTSU specifications (1 of 2)

Parameter	Specifications
Operating clock	PCLKB, PCLKB/2, or PCLKB/4

Table 51.1 CTSU specifications (2 of 2)

Parameter		Specifications
Pins	Electrostatic capacitance measurement	18 channels (TS00 to TS17)
	TSCAP	Low Pass Filter (LPF) connection pin
Measurement modes	Self-capacitance single scan mode	Electrostatic capacitance is measured on one channel using the self-capacitance method
	Self-capacitance multiscan mode	Electrostatic capacitance is measured successively on multiple channels using the self-capacitance method
	Mutual capacitance full scan mode	Electrostatic capacitance is measured successively on multiple channels using the mutual capacitance method
Noise prevention		Synchronous noise prevention, high-pass noise prevention
Measurement start conditions		• Software trigger • External trigger (ELC_CTSU from the Event Link Controller (ELC))

As Figure 51.3 shows, the CTSU consists of a status control block, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt block, and I/O registers.



ICO: Intensity of Current Controlled Oscillator

Note 1. n = 00 to 17

Figure 51.3 CTSU block diagram

Table 51.2 CTSU I/O pins

Pin name	I/O	Function
TS00 to TS17	Input	Electrostatic capacitive measurement pins (touch pins)
TSCAP	-	LPF connection pin

51.2 Register Descriptions

51.2.1 CTSU Control Register 0 (CTSUCR0)

Address(es): CTSU.CTSUCR0 4008 1000h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUTXVSEL	—	—	CTSUINIT	—	CTSUSNZ	CTSUCAP	CTSUSTRT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CTSUSTRT	CTSU Measurement Operation Start	0: Stop measurement operation*1 1: Start measurement operation.	R/W
b1	CTSUCAP	CTSU Measurement Operation Start Trigger Select	0: Software trigger 1: External trigger.	R/W
b2	CTSUSNZ	CTSU Wait State Power-Saving Enable	0: Disable power-saving function during wait state 1: Enable power-saving function during wait state.	R/W
b3	—	Reserved	This bit read as 0. The write value should be 0.	R/W
b4	CTSUINIT	CTSU Control Block Initialization	Writing 1 to this bit initializes the CTSU control block and CTSUSC, CTSURC, CTSUMCH0, CTSUMCH1, and CTSUST registers. This bit is read as 0.	W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CTSUTXVSEL	CTSU Transmission Power Supply Select	0: VCC selected 1: Internal logic power supply selected.	R/W

Note 1. When the CTSU is not used, set the value of this bit to 0.

Only set the CTSUCAP and CTSUSNZ bits when the CTSUSTRT bit is 0. These bits can be set at the same time that measurement operation starts.

CTSUSTRT bit (CTSU Measurement Operation Start)

The CTSUSTRT bit specifies whether CTSU operation starts or stops. When the CTSUCAP bit is 0, measurement starts when the software writes 1 to the CTSUSTRT bit (software trigger) and stops when the hardware clears the CTSUSTRT bit to 0. When the CTSUCAP bit is 1, the CTSU waits for an external trigger by writing 1 to the CTSUSTRT bit, and measurement starts on the rising edge of the external trigger. When measurement is stopped, the CTSU waits for the next external trigger and operation continues.

Table 51.3 lists the CTSU states.

Table 51.3 CTSU states

CTSUSTRT bit	CTSUCAP bit	CTSU state
0	0	Stopped
0	1	Stopped
1	0	Measurement in progress
1	1	Measurement in progress and waiting for an external trigger*1

Note 1. The state can be read from the CTSUST.CTSUSTC[2:0] flags as follows:
During measurement: CTSUST.CTSUSTC[2:0] flags ≠ 000b
While waiting for an external trigger: CTSUST.CTSUSTC[2:0] flags = 000b

If the software sets the CTSUSTRT bit to 1 when the bit is already 1, the write is ignored and operation continues. To force operation to stop through the software when the CTSUSTRT bit is 1, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 at the same time.

CTSUCAP bit (CTSU Measurement Operation Start Trigger Select)

The CTSUCAP bit specifies the measurement start condition. For details, see CTSUSTRT bit (CTSU Measurement Operation Start).

CTSUSNZ bit (CTSU Wait State Power-Saving Enable)

The CTSUSNZ bit enables or disables power-saving operation during a wait state. It can also suspend the CTSU power supply, which decreases power consumption during the wait state. In the suspended state, the CTSU power supply is turned off while the external TSCAP is still charged.

Table 51.4 shows the CTSU power supply state control.

Table 51.4 CTSU power supply state control

CTSUCR1.CTSUPON bit	CTSUSNZ bit	CTSUCAP bit	CTSUSTRT bit	CTSU power supply state
0	0	0	0	Stopped
1	0	—	—	Operating
1	1	0	0	Suspended

Note: Other settings are prohibited.

To start measurement from the suspended state, set the CTSUSNZ bit to 0, and then set the CTSUSTRT bit to 1. To suspend the module after measurement stops, set the CTSUSNZ bit to 1.

CTSUINIT bit (CTSU Control Block Initialization)

Write 1 to the CTSUINIT bit to initialize the internal control registers. To force the current operation to stop, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 at the same time. This stops the operation and initializes the internal control registers.

Do not write 1 to the CTSUINIT bit when the CTSUSTRT bit is 1.

CTSUTXVSEL bit (CTSU Transmission Power Supply Select)

This bit is used to switch the power supply for the transmit buffer in mutual capacitance full scan mode. Set this bit to 0 for any other mode or when the VCC voltage is lower than 2.7 V. This bit switches the power supply for touch I/O which is set for transmission by the CTSUCHTRCn registers. Table 51.5 lists the power supply for TSm pin. When the VCC voltage fluctuates greatly due to the switching of the output buffer, switching to the internal logic power supply can reduce the effect on the voltage fluctuation.

Table 51.5 Power supplied to the TSm pins

Setting of CTSUCHTRCn Register	CTSUTXVSEL bit	Power supply of TSm pins
0 (Reception)	*	VCC
1 (Transmission)	0 (VCC)	
	1 (Internal logic power supply)	Internal logic power supply

51.2.2 CTSU Control Register 1 (CTSUCR1)

Address(es): CTSU.CTSUCR1 4008 1001h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUMD[1:0]		CTSUCLK[1:0]		CTSUATUNE1	—	CTSUCSW	CTSUPON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CTSUPON	CTSU Power Supply Enable	0: Power off the CTSU 1: Power on the CTSU.	R/W
b1	CTSUCSW	CTSU LPF Capacitance Charging Control	This bit controls charging of the LPF capacitance connected to the TSCAP pin. 0: Turn off capacitance switch 1: Turn on capacitance switch.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CTSUATUNE1	CTSU Power Supply Capacity Adjustment	0: Normal output 1: High-current output.	R/W
b5, b4	CTSUCLK[1:0]	CTSU Operating Clock Select	b5 b4 0 0: PCLKB 0 1: PCLKB/2 (PCLKB divided by 2) 1 0: PCLKB/4 (PCLKB divided by 4) 1 1: Setting prohibited.	R/W
b7, b6	CTSUMD[1:0]	CTSU Measurement Mode Select	b7 b6 0 0: Self-capacitance single scan mode 0 1: Self-capacitance multiscan mode 1 0: Setting prohibited 1 1: Mutual capacitance full scan mode.	R/W

Only set the CTSUCR1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPON bit (CTSU Power Supply Enable)

The CTSUPON bit controls the power supply to the CTSU. Set the CTSUPON and CTSUCSW bits to the same value.

CTSUCSW bit (CTSU LPF Capacitance Charging Control)

The CTSUCSW bit controls charging of the LPF capacitor connected to the TSCAP pin by turning the capacitance switch on or off. After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement by setting CTSUCR0.CTSUSTRT to 1. Before starting measurement, use an I/O port to output low to the TSCAP pin, and discharge the existing LPF capacitance. Set the CTSUPON and CTSUCSW bits to the same value.

CTSUATUNE1 bit (CTSU Power Supply Capacity Adjustment)

The CTSUATUNE1 bit sets the capacity of the CTSU power supply. Normally, set this bit to 0.

CTSUCLK[1:0] bits (CTSU Operating Clock Select)

The CTSUCLK[1:0] bits select the operating clock.

CTSUMD[1:0] bits (CTSU Measurement Mode Select)

The CTSUMD[1:0] bits set the measurement mode. For details, see section 51.3.2, Measurement Modes.

51.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS)

Address(es): CTSU.CTSUSDPRS 4008 1002h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUSOFF	CTSUPRMODE[1:0]		CTSUPRRATIO[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CTSUPRRATIO[3:0]	CTSU Measurement Time and Pulse Count Adjustment	These bits set the measurement time and the measurement pulse count setting. Recommended setting: 3 (0011b).	R/W
b5, b4	CTSUPRMODE[1:0]	CTSU Base Period and Pulse Count Setting	These bits set the base pulse count. b5 b4 0 0: 510 pulses 0 1: 126 pulses 1 0: 62 pulses (recommended setting) 1 1: Setting prohibited.	R/W
b6	CTSUSOFF	CTSU High-Pass Noise Reduction Function Off Setting	This bit turns spectrum diffusion on or off to reduce high-pass noise. 0: Turn on 1: Turn off.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Only set the CTSUSDPRS register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPRRATIO[3:0] bits (CTSU Measurement Time and Pulse Count Adjustment)

The CTSUPRRATIO[3:0] bits are used to determine the measurement time and the measurement pulse count. These values are calculated using the following formulas, where the base pulse count is determined by the CTSUPRMODE[1:0] setting:

Measurement pulse count = base pulse count × (CTSUPRRATIO[3:0] bits + 1)

Measurement time = (base pulse count × (CTSUPRRATIO[3:0] bits + 1) + base pulse count - 2) × 0.25 × base clock cycle

Note: For details on the base clock cycle, see section 51.2.17, CTSU Sensor Offset Register 1 (CTSUSO1).

CTSUPRMODE[1:0] bits (CTSU Base Period and Pulse Count Setting)

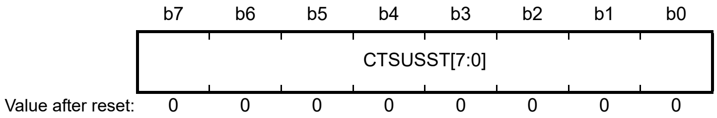
The CTSUPRMODE[1:0] bits select the number of base pulses that occur during measurement.

CTSUSOFF bit (CTSU High-Pass Noise Reduction Function Off Setting)

The CTSUSOFF bit turns on or off the function for reducing high-pass noise. Set this bit to 1 to turn the function off.

51.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST)

Address(es): CTSU.CTSUSST 4008 1003h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUSST[7:0]	CTSU Sensor Stabilization Wait Control	Fix the value of these bits to 00010000b.	R/W

Only set the CTSUSST register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSST[7:0] bits (CTSU Sensor Stabilization Wait Control)

The CTSUSST[7:0] bits set the stabilization wait time for the TSCAP pin voltage. Always fix these bits to 00010000b. If these bits are not set, the TSCAP voltage will be unstable at the start of measurement, and the CTSU will be unable to obtain correct touch measurement results.

51.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0)

Address(es): CTSU.CTSUMCH0 4008 1004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CTSUMCH0[4:0]				
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b4 to b0	CTSUMCH0[4:0]	CTSU Measurement Channel 0	In self-capacitance single scan mode, these bits set the channel to be measured. b4 b0 0 0 0 0 0: TS00 0 0 0 0 1: TS01 0 0 0 1 0: TS02 0 0 0 1 1: TS03 0 0 1 0 0: TS04 0 0 1 0 1: TS05 0 0 1 1 0: TS06 0 0 1 1 1: TS07 0 1 0 0 0: TS08 0 1 0 0 1: TS09 0 1 0 1 0: TS10 0 1 0 1 1: TS11 0 1 1 0 0: TS12 0 1 1 0 1: TS13 0 1 1 1 0: TS14 0 1 1 1 1: TS15 1 0 0 0 0: TS16 1 0 0 0 1: TS17. Other than when specified, starting measurement operation by setting CTSUCR0.CTSUSTRT to 1 is prohibited after these bits are set. In other measurement modes, these bits indicate the channel that is currently being measured. b4 b0 0 0 0 0 0: TS00 0 0 0 0 1: TS01 0 0 0 1 0: TS02 0 0 0 1 1: TS03 0 0 1 0 0: TS04 0 0 1 0 1: TS05 0 0 1 1 0: TS06 0 0 1 1 1: TS07 0 1 0 0 0: TS08 0 1 0 0 1: TS09 0 1 0 1 0: TS10 0 1 0 1 1: TS11 0 1 1 0 0: TS12 0 1 1 0 1: TS13 0 1 1 1 0: TS14 0 1 1 1 1: TS15 1 0 0 0 0: TS16 1 0 0 0 1: TS17 1 1 1 1 1: Measurement is being stopped.	R/W*[1]
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only enabled in self-capacitance single scan mode (CTSUCR1.CTSUMD[1:0] bits = 00b).

Only set the CTSUMCH0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUMCH0[4:0] bits (CTSU Measurement Channel 0)

In self-capacitance single scan mode, the CTSUMCH0[4:0] bits set the channel to be measured. In this mode, only specify enabled channels (00000b to 10001b). In other modes, these indicate the receive channel that is being measured, and writing to these bits has no effect.

51.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1)

Address(es): CTSU.CTSUMCH1 4008 1005h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CTSUMCH1[4:0]				
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b4 to b0	CTSUMCH1[4:0]	CTSU Measurement Channel 1	b4 b0 0 0 0 0 0: TS00 0 0 0 0 1: TS01 0 0 0 1 0: TS02 0 0 0 1 1: TS03 0 0 1 0 0: TS04 0 0 1 0 1: TS05 0 0 1 1 0: TS06 0 0 1 1 1: TS07 0 1 0 0 0: TS08 0 1 0 0 1: TS09 0 1 0 1 0: TS10 0 1 0 1 1: TS11 0 1 1 0 0: TS12 0 1 1 0 1: TS13 0 1 1 1 0: TS14 0 1 1 1 1: TS15 1 0 0 0 0: TS16 1 0 0 0 1: TS17 1 1 1 1 1: Measurement is being stopped.	R
b7 to b5	—	Reserved	These bits are read as 0.	R

CTSUMCH1[4:0] bits (CTSU Measurement Channel 1)

In full scan mode, the CTSUMCH1[4:0] bits indicate the transmit channel that is being measured. They are always 11111b when measurement is stopped, or in self-capacitance single scan and multi-scan modes.

51.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0)

Address(es): CTSU.CTSUCHAC0 4008 1006h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC0[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC0[7:0]	CTSU Channel Enable Control 0	These bits select whether the associated TS pin is measured. 0: Do not measure 1: Measure. These bits specify the TS00 to TS07 pins.	R/W

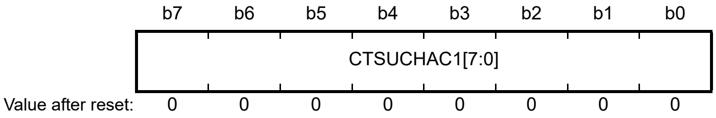
Only set the CTSUCHAC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC0[7:0] bits (CTSU Channel Enable Control 0)

The CTSUCHAC0[7:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC0[0] is associated with TS00 and CTSUCHAC0[7] with TS07.

51.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1)

Address(es): CTSU.CTSUCHAC1 4008 1007h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC1[7:0]	CTSU Channel Enable Control 1	These bits select whether the associated TS pin is measured. 0: Do not measure 1: Measure. These bits specify the TS08 to TS15 pins.	R/W

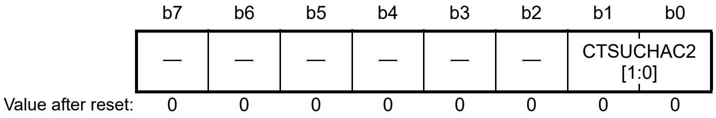
Only set the CTSUCHAC1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC1[7:0] bits (CTSU Channel Enable Control 1)

The CTSUCHAC1[7:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC1[0] is associated with TS08 and CTSUCHAC1[7] with TS15.

51.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2)

Address(es): CTSU.CTSUCHAC2 4008 1008h



Bit	Symbol	Bit name	Description	R/W
b1 to b0	CTSUCHAC2 [1:0]	CTSU Channel Enable Control 2	These bits select whether the associated TS pin is measured. 0: Do not measure 1: Measure. These bits specify the TS16 to TS17 pins.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

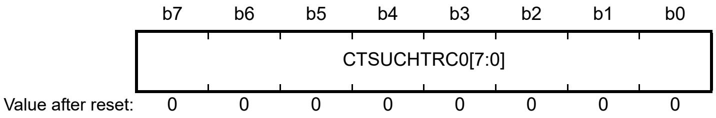
Only set the CTSUCHAC2 register when the CTSUCR0.CTSUSTRT bit = 0.

CTSUCHAC2 [1:0] bits (CTSU Channel Enable Control 2)

The CTSUCHAC2 [1:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC2[0] is associated with TS16 and CTSUCHAC2[1] with TS17.

51.2.10 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0)

Address(es): CTSU.CTSUCHTRC0 4008 100Bh



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC0[7:0]	CTSU Channel Transmit/Receive Control 0	0: Reception 1: Transmission. These bits specify the TS00 to TS07 pins.	R/W

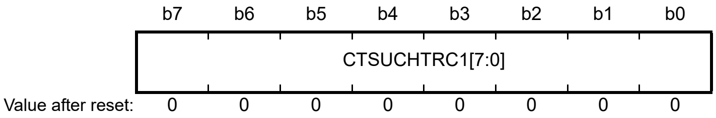
Only set the CTSUCHTRC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC0[7:0] bits (CTSU Channel Transmit/Receive Control 0)

In full scan mode, the CTSUCHTRC0[7:0] bits allocate reception or transmission to the associated TS pins. The setting is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC0[0] is associated with TS00 and CTSUCHTRC0[7] with TS07.

51.2.11 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)

Address(es): CTSU.CTSUCHTRC1 4008 100Ch



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC1[7:0]	CTSU Channel Transmit/Receive Control 1	0: Reception 1: Transmission. These bits specify the TS08 to TS15 pins.	R/W

Only set the CTSUCHTRC1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC1[7:0] bits (CTSU Channel Transmit/Receive Control 1)

In full scan mode, the CTSUCHTRC1[7:0] bits allocate reception or transmission to the associated TS pins. The setting is ignored in self-capacitance single scan and multiscan modes. CTSUCHTRC1[0] is associated with TS08 and CTSUCHTRC1[7] with TS15.

51.2.12 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)

Address(es): CTSU.CTSUCHTRC2 4008 100Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CTSUCHTRC2 [1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1 to b0	CTSUCHTRC2 [1:0]	CTSU Channel Transmit/Receive Control 2	0: Reception 1: Transmission. These bits specify the TS16 to TS17 pins.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Only set the CTSUCHTRC2 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC2 [1:0] bits (CTSU Channel Transmit/Receive Control 2)

In full scan mode, the CTSUCHTRC2 [1:0] bits allocate reception or transmission to the associated TS pins. The setting is ignored in self-capacitance single scan and multiscan modes. CTSUCHTRC2[0] is associated with TS16 and CTSUCHTRC2 [1] with TS17.

51.2.13 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC)

Address(es): CTSU.CTSUDCLKC 4008 1010h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTSUSSCNT[1:0]		—	—	CTSUSSMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CTSUSSMOD[1:0]	CTSU Diffusion Clock Mode Select	Set these bits to 00b.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	CTSUSSCNT[1:0]	CTSU Diffusion Clock Mode Control	Set these bits to 11b.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Only set the CTSUDCLKC register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSSMOD[1:0] bits (CTSU Diffusion Clock Mode Select)

The CTSUSSMOD[1:0] bits set the mode of the spectrum diffusion clock for high-pass noise reduction. When using the high-pass function, always fix these bits to 00b. If these bits are not set, the CTSU will be unable to effectively reduce high-pass noise.

CTSUSSCNT[1:0] bits (CTSU Diffusion Clock Mode Control)

The CTSUSSCNT[1:0] bits adjust the amount of spectrum diffusion applied to reduce high-pass noise. When using the high-pass noise reduction function, always fix these bits to 11b. If these bits are not set, touch measurement might be performed incorrectly.

51.2.14 CTSU Status Register (CTSUST)

Address(es): CTSU.CTSUST 4008 1011h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUPS	CTSUROVF	CTSUSOVF	CTSUDTSR	—	CTSUSTC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CTSUSTC[2:0]	CTSU Measurement Status Counter	These counters indicate the current measurement status. b2 b0 0 0 0: Status 0 0 0 1: Status 1 0 1 0: Status 2 0 1 1: Status 3 1 0 0: Status 4 1 0 1: Status 5.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CTSUDTSR	CTSU Data Transfer Status Flag	This flag indicates whether the measurement result stored in the sensor counter and the reference counter was read. 0: Read 1: Not read.	R
b5	CTSUSOVF	CTSU Sensor Counter Overflow Flag	This flag indicates an overflow on the sensor counter. 0: No overflow occurred 1: Overflow occurred.	R/W
b6	CTSUROVF	CTSU Reference Counter Overflow Flag	This flag indicates an overflow on the reference counter. 0: No overflow occurred 1: Overflow occurred.	R/W
b7	CTSUPS	CTSU Mutual Capacitance Status Flag	This flag indicates the measurement status in mutual capacitance full scan mode. 0: First measurement 1: Second measurement.	R

When using the CTSUCR0.CTSUINIT bit to clear an overflow flag, make sure that the CTSUCR0.CTSUSTRT bit is 0.

CTSUSTC[2:0] flags (CTSU Measurement Status Counter)

The CTSUSTC[2:0] flags are a counter indicating the current measurement status. For details on each status, see section 51.3.2.2, Status counter.

CTSUDTSR flag (CTSU Data Transfer Status Flag)

The CTSUDTSR flag indicates whether the measurement result stored in the sensor counter and the reference counter was read. The flag sets to 1 when measurement completes and 0 when the reference counter is read by software or the DTC. The flag can also be cleared using the CTSUCR0.CTSUINIT bit.

CTSUSOVF flag (CTSU Sensor Counter Overflow Flag)

The CTSUSOVF flag sets to 1 when the sensor counter, CTSUSC, overflows. On overflow, the counter value reads as FFFFh. Measurement processing continues for the specified period.

No interrupt occurs on an overflow. To determine the channel on which the overflow occurred, read the measurement result of each channel after measurement completes, signaled by a measurement end interrupt.

This flag is cleared when 0 is written after 1 is read by software. It can also be cleared using the CTSUCR0.CTSUINIT bit.

CTSUROVF flag (CTSU Reference Counter Overflow Flag)

The CTSUROVF flag sets to 1 when the reference counter, CTSURC, overflows. On overflow, the counter value reads as FFFFh. Measurement processing continues for the specified period.

No interrupt occurs on an overflow. To determine the channel on which the overflow occurred, read the measurement result of each channel after measurement completes, signaled by a measurement end interrupt.

This flag is cleared when 0 is written after 1 is read by software. It can also be cleared using the CTSUCR0.CTSUINIT bit.

CTSUPS flag (CTSU Mutual Capacitance Status Flag)

In mutual capacitance full scan mode, when CTSUCR1.CTSUMD[1:0] = 11b, the CTSUPS flag indicates whether the measurement is the first or second of two measurements for each channel. When measurement is stopped or in other measurement modes, this flag is always 0.

51.2.15 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC)

Address(es): CTSU.CTSUSSC 4008 1012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUSSDIV[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	CTSUSSDIV[3:0]	CTSU Spectrum Diffusion Frequency Division Setting	These bits specify the spectrum diffusion frequency division setting based on the base clock frequency division setting.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CTSUSSDIV[3:0] bits (CTSU Spectrum Diffusion Frequency Division Setting)

The CTSUSSDIV[3:0] bits specify the spectrum diffusion frequency derived from the base clock frequency division setting. To calculate the correct setting for CTSUSSDIV[3:0], see the relationship between base clock frequencies and the settings in Table 51.6.

Table 51.6 Relationship between base clock frequencies and CTSUSSDIV[3:0] bit settings (1 of 2)

Base clock frequency fb (MHz)	CTSUSSDIV[3:0] bit setting
4.00 ≤ fb	0000b
2.00 ≤ fb < 4.00	0001b
1.33 ≤ fb < 2.00	0010b
1.00 ≤ fb < 1.33	0011b
0.80 ≤ fb < 1.00	0100b
0.67 ≤ fb < 0.80	0101b
0.57 ≤ fb < 0.67	0110b
0.50 ≤ fb < 0.57	0111b
0.44 ≤ fb < 0.50	1000b
0.40 ≤ fb < 0.44	1001b
0.36 ≤ fb < 0.40	1010b
0.33 ≤ fb < 0.36	1011b
0.31 ≤ fb < 0.33	1100b

Table 51.6 Relationship between base clock frequencies and CTSUSSDIV[3:0] bit settings (2 of 2)

Base clock frequency fb (MHz)	CTSUSSDIV[3:0] bit setting
0.29 ≤ fb < 0.31	1101b
0.27 ≤ fb < 0.29	1110b
fb < 0.27	1111b

51.2.16 CTSU Sensor Offset Register 0 (CTSUSO0)

Address(es): CTSU.CTSUSO0 4008 1014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUSNUM[5:0]						CTSUSO[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	CTSUSO[9:0]	CTSU Sensor Offset Adjustment	These bits adjust the electronic capacitance when the electrode is not being touched. b9 b0 0 0 0 0 0 0 0 0 0 0: Current offset is 0 0 0 0 0 0 0 0 0 0 1: Current offset is 1 0 0 0 0 0 0 0 0 1 0: Current offset is 2 : : 1 1 1 1 1 1 1 1 1 0: Current offset is 1022 1 1 1 1 1 1 1 1 1 1: Current offset is maximum.	R/W
b15 to b10	CTSUSNUM[5:0]	CTSU Measurement Count Setting	These bits set the number of measurements.	R/W

CTSUSO[9:0] bits (CTSU Sensor Offset Adjustment)

The CTSUSO[9:0] bits offset the sensor ICO input current generated from electrostatic capacitance during touch measurement when the electrode is not being touched. This prevents the CTSU sensor counter from overflowing. Set the TS pin that is to be measured next after a CTSU_CTSUWR interrupt occurs.

CTSUSNUM[5:0] bits (CTSU Measurement Count Setting)

The CTSUSNUM[5:0] bits specify how many times the measurement pulse count specified in the CTSUSDPRS.CTSUPRRATIO[3:0] and CTSUSDPRS.CTSUPRMODE[1:0] bits is repeated during the measurement time. The measurement pulse count is repeated (CTSUSNUM[5:0] bits + 1) times. Set the TS pin that is to be measured next after a CTSU_CTSUWR interrupt is generated.

51.2.17 CTSU Sensor Offset Register 1 (CTSUSO1)

Address(es): CTSU.CTSUSO1 4008 1016h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUICOG[1:0]		CTSUSDPA[4:0]					CTSURICOA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSURICOA[7:0]	CTSU Reference ICO Current Adjustment	These bits adjust the input current of the reference ICO. b7 b0 0 0 0 0 0 0 0 0: Current offset is 0 0 0 0 0 0 0 0 1: Current offset is 1 0 0 0 0 0 0 1 0: Current offset is 2 : : 1 1 1 1 1 1 1 0: Current offset is 254 1 1 1 1 1 1 1 1: Current offset is maximum.	R/W
b12 to b8	CTSUSDPA[4:0]	CTSU Base Clock Setting	These bits are used to generate the base clock. b12 b8 0 0 0 0 0: Operating clock divided by 2*1 0 0 0 0 1: Operating clock divided by 4 0 0 0 1 0: Operating clock divided by 6 0 0 0 1 1: Operating clock divided by 8 0 0 1 0 0: Operating clock divided by 10 0 0 1 0 1: Operating clock divided by 12 0 0 1 1 0: Operating clock divided by 14 0 0 1 1 1: Operating clock divided by 16 0 1 0 0 0: Operating clock divided by 18 0 1 0 0 1: Operating clock divided by 20 0 1 0 1 0: Operating clock divided by 22 0 1 0 1 1: Operating clock divided by 24 0 1 1 0 0: Operating clock divided by 26 0 1 1 0 1: Operating clock divided by 28 0 1 1 1 0: Operating clock divided by 30 0 1 1 1 1: Operating clock divided by 32 1 0 0 0 0: Operating clock divided by 34 1 0 0 0 1: Operating clock divided by 36 1 0 0 1 0: Operating clock divided by 38 1 0 0 1 1: Operating clock divided by 40 1 0 1 0 0: Operating clock divided by 42 1 0 1 0 1: Operating clock divided by 44 1 0 1 1 0: Operating clock divided by 46 1 0 1 1 1: Operating clock divided by 48 1 1 0 0 0: Operating clock divided by 50 1 1 0 0 1: Operating clock divided by 52 1 1 0 1 0: Operating clock divided by 54 1 1 0 1 1: Operating clock divided by 56 1 1 1 0 0: Operating clock divided by 58 1 1 1 0 1: Operating clock divided by 60 1 1 1 1 0: Operating clock divided by 62 1 1 1 1 1: Operating clock divided by 64.	R/W
b14, b13	CTSUICOG[1:0]	CTSU ICO Gain Adjustment	These bits adjust the output frequency gain of the sensor ICO and the reference ICO. b14 b13 0 0: 100% gain 0 1: 66% gain 1 0: 50% gain 1 1: 40% gain.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Do not set the CTSUSDPA[4:0] bits set to 00000b while the high-pass noise reduction function is turned off (CTSUSDPRS.CTSUSOFF bit = 1) in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).

After a CTSU_CTSUWR interrupt occurs, write first to the CTSUSSC register, next to the CTSUSO0 register, and then to the CTSUSO1 register. The write to the CTSUSO1 register causes a transition to Status 3. (See Table 51.7 and Table 51.8.) Set all of the bits in a single operation when writing to the CTSUSO1 register.

CTSURICOA[7:0] bits (CTSU Reference ICO Current Adjustment)

The CTSURICOA[7:0] bits adjust the oscillation frequency using the input current of the reference ICO.

CTSUSDPA[4:0] bits (CTSU Base Clock Setting)

The CTSUSDPA[4:0] bits select the base clock used as the source for the sensor drive pulse by dividing the operating clock. For details on the setting procedure, see section 51.3.2.1, Initial settings flow.

CTSUICOG[1:0] bits (CTSU ICO Gain Adjustment)

The CTSUICOG[1:0] bits adjust the output frequency gain of the sensor ICO and the reference ICO. Normally, set these bits to 00b for the maximum gain. If changes in the capacitance between when the electrode is touched and when it is not touched greatly exceed the dynamic range of the sensor ICO, adjust the gain appropriately with this setting.

51.2.18 CTSU Sensor Counter (CTSUSC)

Address(es): CTSU.CTSUSC 4008 1018h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUSC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	CTSUSC[15:0]	CTSU Sensor Counter	These bits indicate the measurement result of the sensor ICO. They read FFFFh when an overflow occurs.	R

After a CTSU_CTSURD interrupt occurs, read first from the CTSUSC counter and then from the CTSURC counter.

CTSUSC[15:0] bits (CTSU Sensor Counter)

The CTSUSC[15:0] bits are configured as an increment counter for the sensor ICO clock. Read these bits after a CTSU_CTSURD interrupt occurs. After the CTSURC counter is read, these bits are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. They can also be cleared using the CTSUCR0.CTSUINIT bit.

51.2.19 CTSU Reference Counter (CTSURC)

Address(es): CTSU.CTSURC 4008 101Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSURC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	CTSURC[15:0]	CTSU Reference Counter	These bits indicate the measurement result of the reference ICO. They read FFFFh when an overflow occurs.	R

After a CTSU_CTSURD interrupt occurs, read first from the CTSUSC counter and then from the CTSURC counter. Status 3 continues until the CTSURC counter is read, even if the stabilization time specified for Status 3 elapses.

CTSURC[15:0] bits (CTSU Reference Counter)

The CTSURC[15:0] bits are configured as an increment counter for the reference ICO clock. The reference ICO optimizes touch measurement performed by the sensor ICO. There is some deviation depending on the internal sensor ICO and the reference ICO in the CTSU, but both ICOs have almost the same characteristics, including the dynamic range and the current-to-frequency characteristics. The range of current amount that can be set in the reference ICO current adjustment bits is about the same as the dynamic range of both ICOs, and the current amount input to the sensor ICO must be within this dynamic range. To ensure this, use the reference ICO to check the differences between the ICOs and measure the current-to-oscillation frequency characteristics. Because the reference ICO oscillation frequency can be obtained from the reference ICO counter, the ICO oscillation frequency (counter value/measurement time) for the input current amount can be measured by setting the value in the reference ICO current adjustment bits and measuring the reference ICO counter. The reference ICO counter value measured using the maximum value in the reference ICO current adjustment bits is the maximum value of the ICO dynamic range. The current to the sensor ICO must be offset in the offset adjustment bits so that the sensor ICO counter value does not exceed this value.

Read the CTSURC[15:0] bits after a CTSU_CTSURD interrupt occurs. After these bits are read, they are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. They can also be cleared using the CTSUCR0.CTSUINIT bit.

51.2.20 CTSU Error Status Register (CTSUERRS)

Address(es): CTSU.CTSUERRS 4008 101Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUICOMP	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0.	R
b15	CTSUICOMP	TSCAP Voltage Error Monitor	This bit monitors the error status of the TSCAP voltage. 0: Normal TSCAP voltage 1: Abnormal TSCAP voltage.*1	R

Note 1. When CTSUCR1.CTSUPON bit is 0, this bit is set to 1.

CTSUICOMP bit (TSCAP Voltage Error Monitor)

If the offset current amount set in the CTSUSO1 register exceeds the sensor ICO input current during touch measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be performed correctly. The CTSUICOMP bit monitors the TSCAP voltage and it sets to 1 if the voltage becomes abnormal.

If the TSCAP voltage becomes abnormal, the sensor ICO counter value becomes undefined, but touch measurement completes normally, so it is difficult to detect an abnormality by reading the sensor ICO counter value. If the CTSU reference ICO current adjustment bits (CTSURICOA[7:0]) in the CTSUSO1 register are set to any value other than 0, always check this bit when touch measurement completes.

This bit is cleared by writing 0 to the CTSUCR1.CTSUPON bit and turning off the power supply.

51.3 Operation

51.3.1 Principles of Measurement Operation

Figure 51.4 shows the measurement circuit.

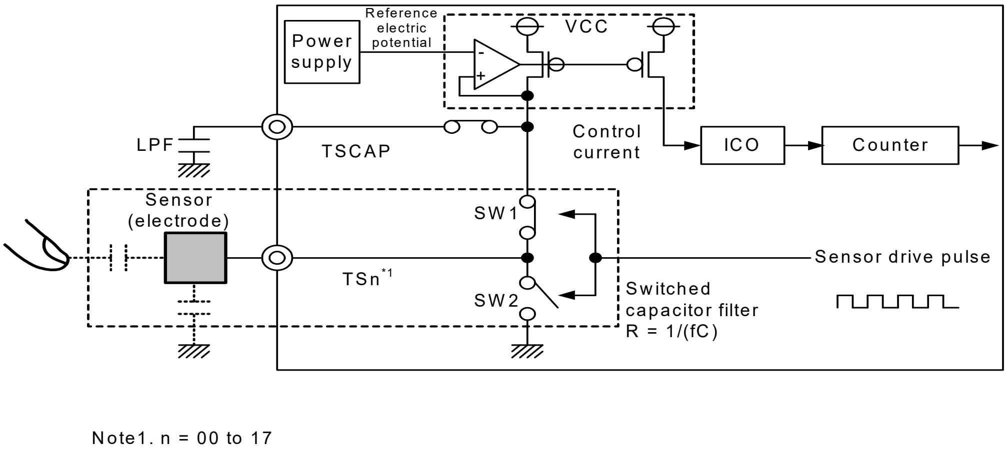


Figure 51.4 Measurement circuit

Figure 51.5 to Figure 51.7 explain the electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method.

The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (Figure 51.5).

The charged capacitance is discharged by turning SW1 off and SW2 on (Figure 51.6).

Current flows to the switched capacitor filter by switching between charging and discharging. At this point, if a finger is in close proximity, the capacitance and the flowing current change. A clock is generated by supplying the control current, which is proportional to the amount of current flowing through the switched capacitor filter, from the circuit that generates the TSCAP power supply to the ICO. The counter measures the clock frequency that changes depending on whether a finger is in close proximity. The software uses the value read from the counter to determine contact with a finger (Figure 51.7).

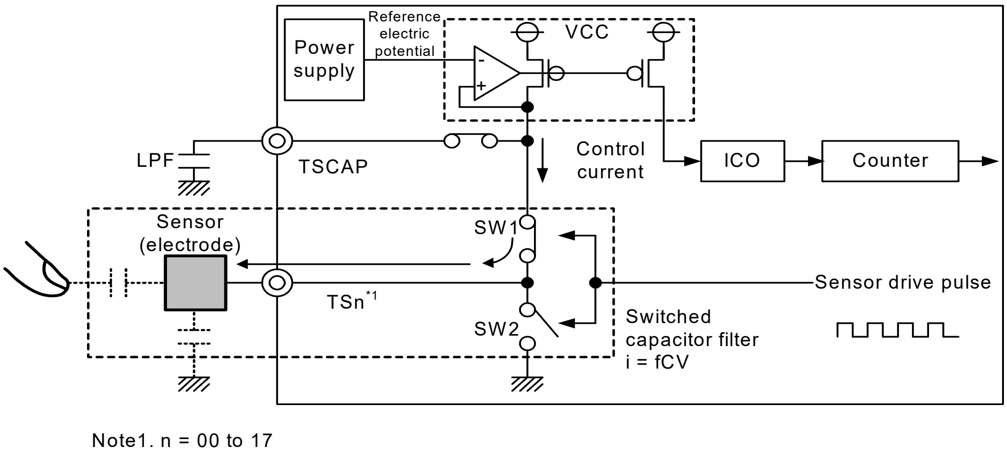


Figure 51.5 Charging operation

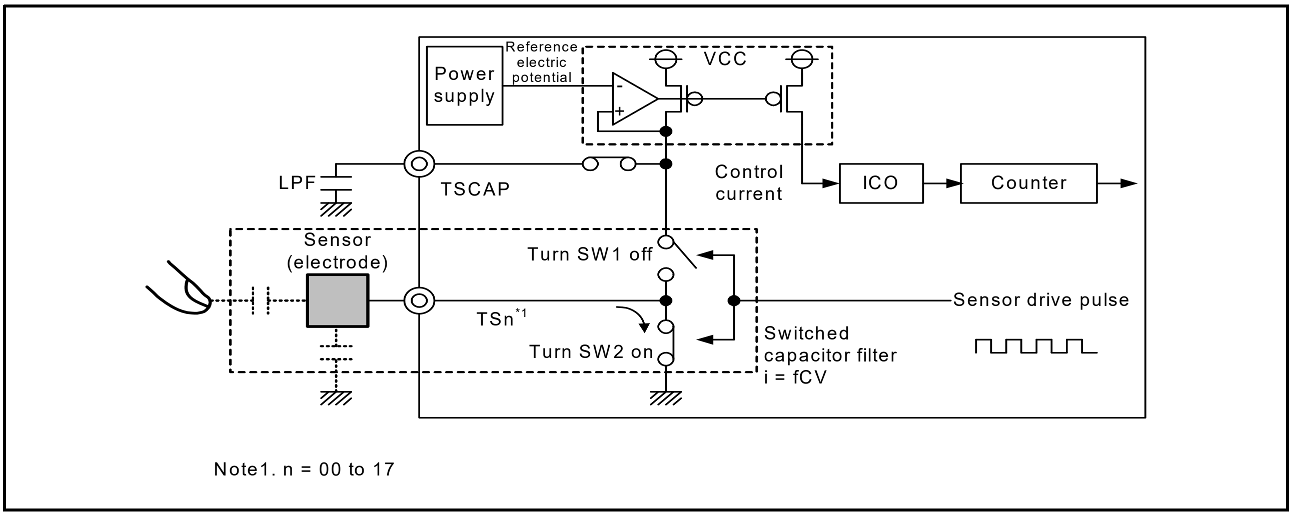


Figure 51.6 Discharging operation

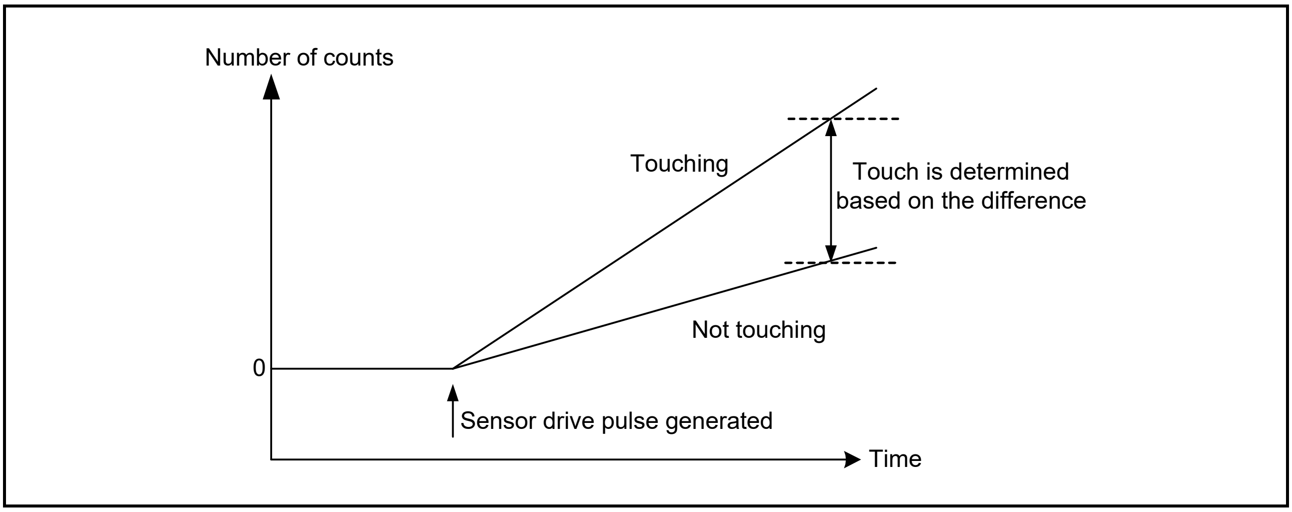


Figure 51.7 Change in measured value when finger is touching and not touching

51.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual capacitance methods. Figure 51.8 illustrates these methods.

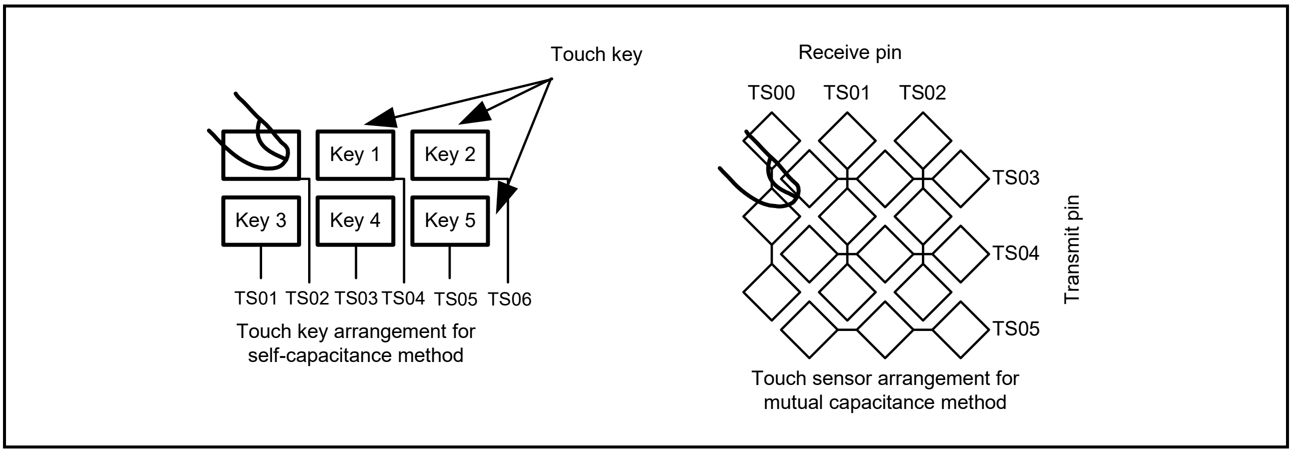
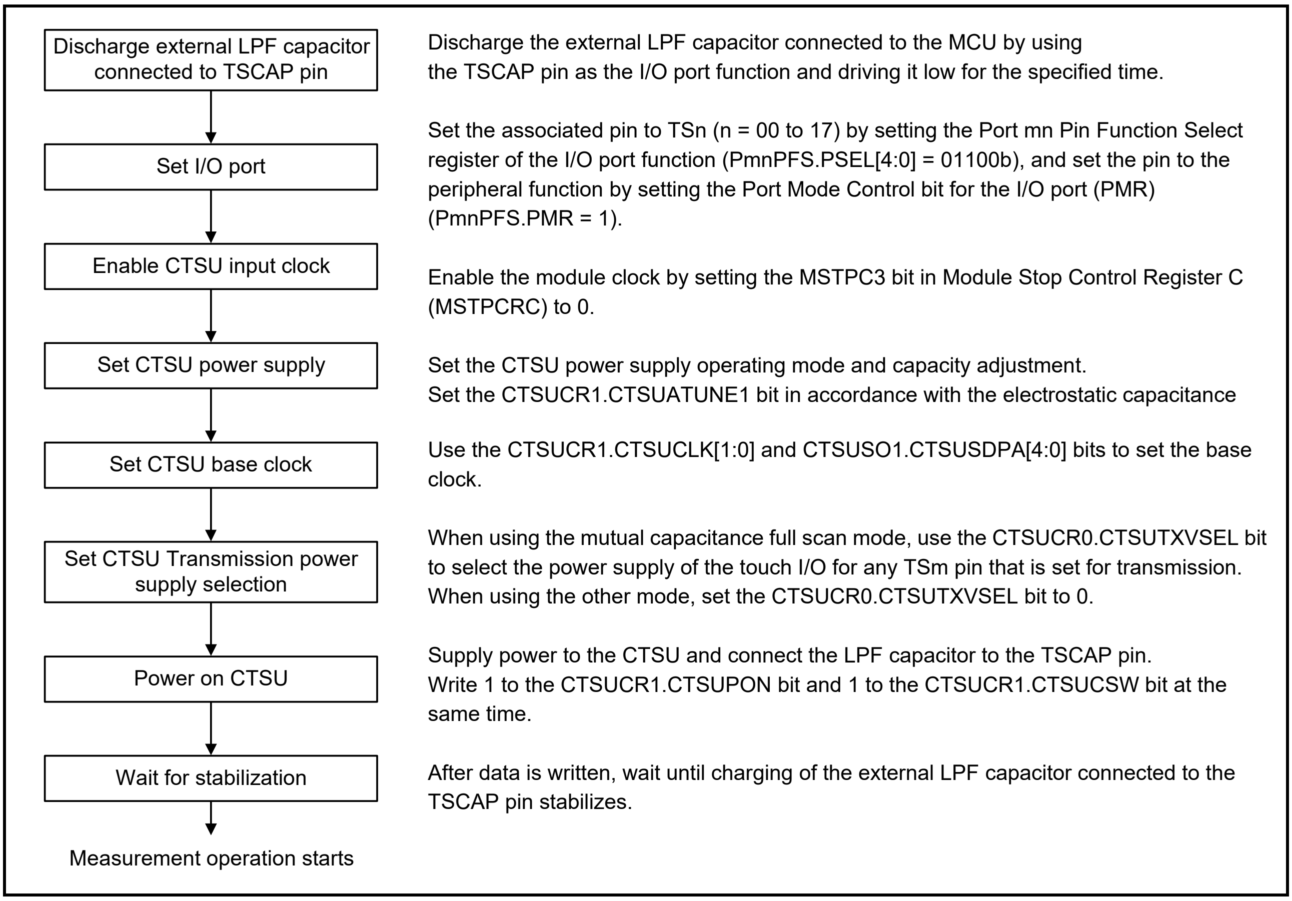


Figure 51.8 Overview of self-capacitance method and mutual capacitance method

In the self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity. In this method, capacitance can be measured in both single scan and multiscan modes. In the mutual capacitance method, the capacitance between two opposing electrodes (transmit and receive pins) is measured.

51.3.2.1 Initial settings flow

Figure 51.9 shows the flow for the CTSU initial settings.



Discharge the external LPF capacitor connected to the MCU by using the TSCAP pin as the I/O port function and driving it low for the specified time.

Set the associated pin to TSn (n = 00 to 17) by setting the Port mn Pin Function Select register of the I/O port function (PmnPFS.PSEL[4:0] = 01100b), and set the pin to the peripheral function by setting the Port Mode Control bit for the I/O port (PMR) (PmnPFS.PMR = 1).

Enable the module clock by setting the MSTPC3 bit in Module Stop Control Register C (MSTPCRC) to 0.

Set the CTSU power supply operating mode and capacity adjustment.
Set the CTSUCR1.CTSUATUNE1 bit in accordance with the electrostatic capacitance

Use the CTSUCR1.CTSUCLK[1:0] and CTSUSO1.CTSUSDPA[4:0] bits to set the base clock.

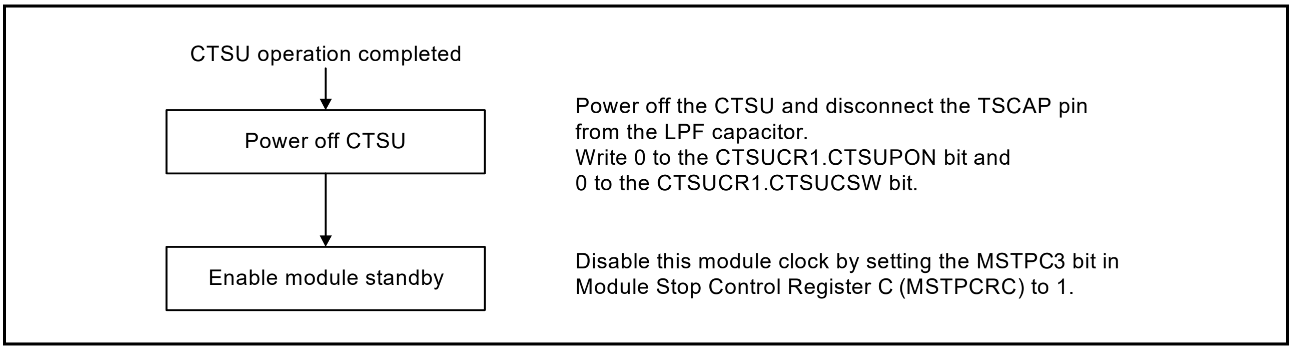
When using the mutual capacitance full scan mode, use the CTSUCR0.CTSUTXVSEL bit to select the power supply of the touch I/O for any TSm pin that is set for transmission. When using the other mode, set the CTSUCR0.CTSUTXVSEL bit to 0.

Supply power to the CTSU and connect the LPF capacitor to the TSCAP pin.
Write 1 to the CTSUCR1.CTSUPON bit and 1 to the CTSUCR1.CTSUCSW bit at the same time.

After data is written, wait until charging of the external LPF capacitor connected to the TSCAP pin stabilizes.

Figure 51.9 CTSU initial settings flow

Figure 51.10 shows the flow for stopping CTSU operation and invoking the standby state.



Power off the CTSU and disconnect the TSCAP pin from the LPF capacitor.
Write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit.

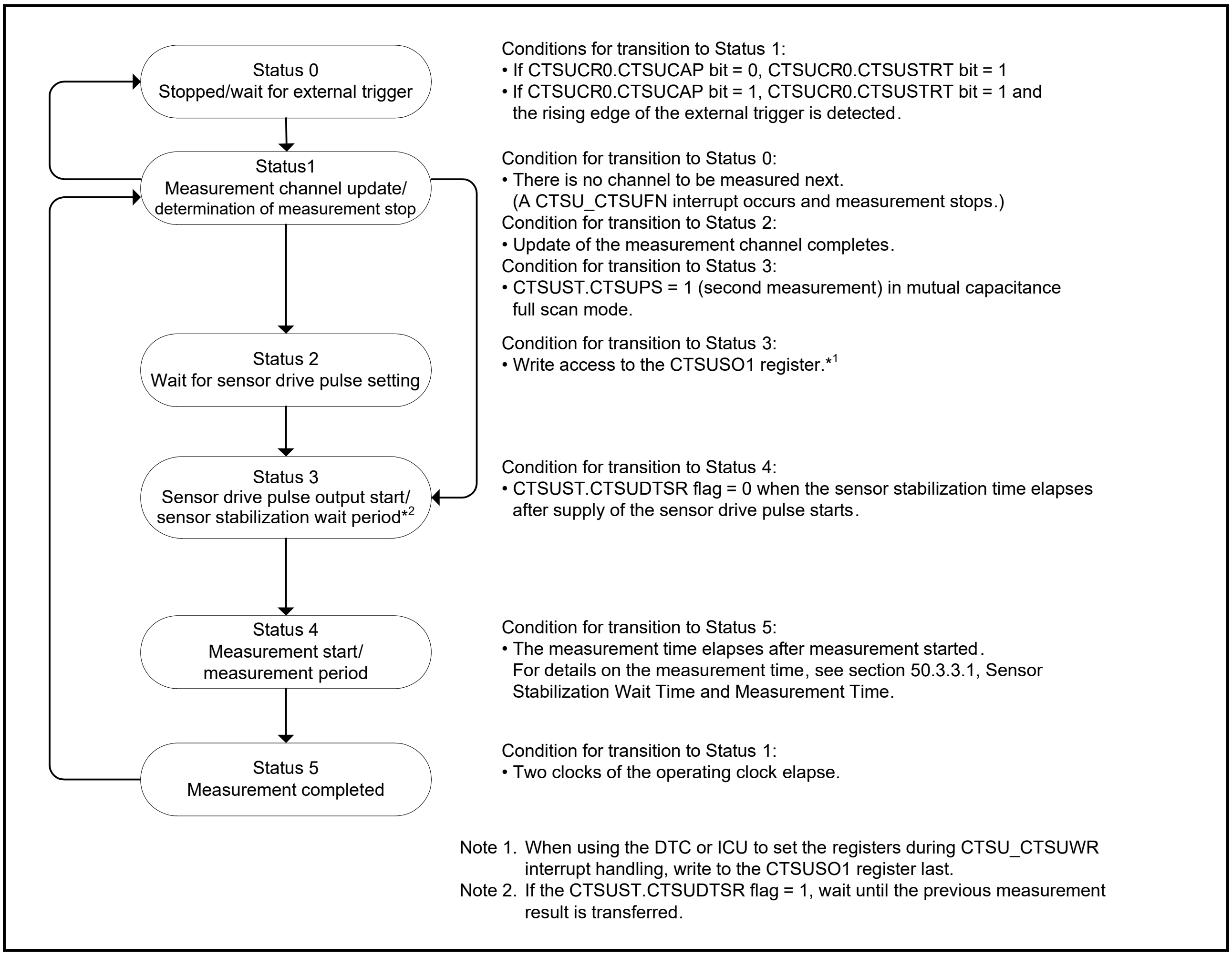
Disable this module clock by setting the MSTPC3 bit in Module Stop Control Register C (MSTPCRC) to 1.

Figure 51.10 CTSU stopping flow

When restarting operation after it stops, follow the initial settings flow shown in Figure 51.9.

51.3.2.2 Status counter

The measurement status counter of the CTSU Status Register (CTSUST) indicates the current measurement status. The measurement status is shared by all three modes. Figure 51.11 shows the status operation transitions.



Conditions for transition to Status 1:
- If CTSUCR0.CTSUCAP bit = 0, CTSUCR0.CTSUSTRT bit = 1
- If CTSUCR0.CTSUCAP bit = 1, CTSUCR0.CTSUSTRT bit = 1 and the rising edge of the external trigger is detected.

Condition for transition to Status 0:
- There is no channel to be measured next. (A CTSU_CTSUFN interrupt occurs and measurement stops.)

Condition for transition to Status 2:
- Update of the measurement channel completes.

Condition for transition to Status 3:
- CTSUST.CTSUPS = 1 (second measurement) in mutual capacitance full scan mode.

Condition for transition to Status 3:
- Write access to the CTSUSO1 register.*1

Condition for transition to Status 4:
- CTSUST.CTSUDTSR flag = 0 when the sensor stabilization time elapses after supply of the sensor drive pulse starts.

Condition for transition to Status 5:
- The measurement time elapses after measurement started. For details on the measurement time, see section 50.3.3.1, Sensor Stabilization Wait Time and Measurement Time.

Condition for transition to Status 1:
- Two clocks of the operating clock elapse.

Note 1. When using the DTC or ICU to set the registers during CTSU_CTSUWR interrupt handling, write to the CTSUSO1 register last.

Note 2. If the CTSUST.CTSUDTSR flag = 1, wait until the previous measurement result is transferred.

Figure 51.11 Status operation transitions

The status counter transitions to Status 0 when all of the specified measurement channels are measured.

The CTSUCR0.CTSUSTRT bit is cleared to 0 by hardware when a software trigger is used. When an external trigger is used, the value of 1 is retained, and the CTSU waits for the next trigger.

When operation is forced to stop during measurement or the trigger wait state, by a simultaneous 0 write to the CTSUCR0.CTSUSTRT bit and a 1 write to the CTSUCR0.CTSUINIT bit, the status transitions to Status 0 and measurement stops.

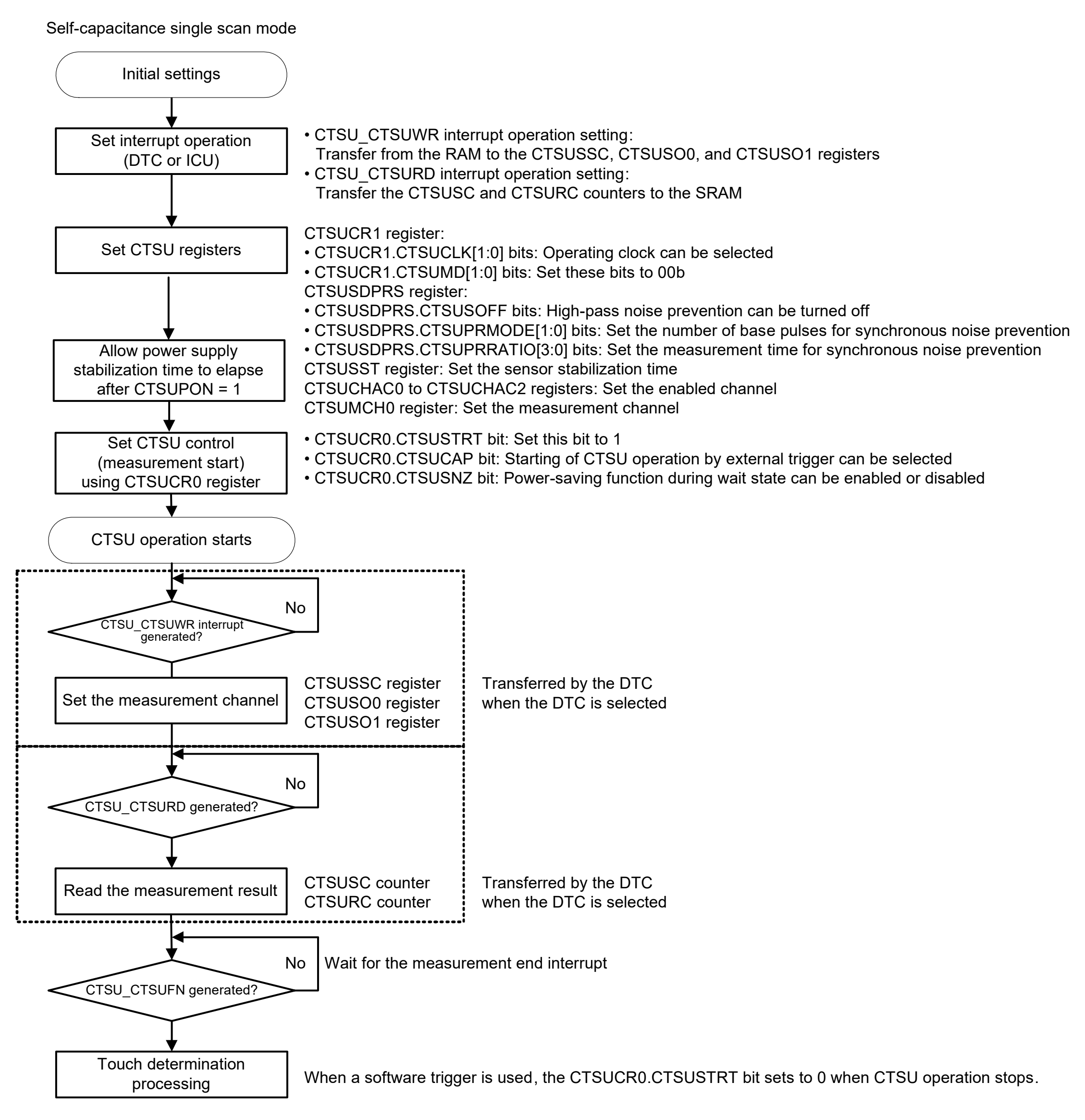
If the channel to be measured is not set in the CTSUCHAC0 to CTSUCHAC2 or CTSUCHTRC0 to CTSUCHTRC2 registers, a CTSU_CTSUFN interrupt occurs immediately after a transition to Status 1, and then the status transitions to Status 0.

In the following situations, there is no channel to be measured:

- No measurement target channel is specified in the CTSUCHAC0 to CTSUCHAC2 registers.
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHAC0 to CTSUCHAC2 registers.
- In full scan modes, there is no transmit channel or receive channel to be measured based on the combined settings of the CTSUCHAC0 to CTSUCHAC2 and CTSUCHTRC0 to CTSUCHTRC2 registers.

51.3.2.3 Self-capacitance single scan mode operation

In self-capacitance single scan mode, electrostatic capacitance on one channel is measured. Figure 51.12 shows the software flow and an operation example, and Figure 51.13 shows the timing.



- CTSU_CTSUWR interrupt operation setting:
  Transfer from the RAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers
- CTSU_CTSURD interrupt operation setting:
  Transfer the CTSUSC and CTSURC counters to the SRAM

CTSUCR1 register:
- CTSUCR1.CTSUCLK[1:0] bits: Operating clock can be selected
- CTSUCR1.CTSUMD[1:0] bits: Set these bits to 00b

CTSUSDPRS register:
- CTSUSDPRS.CTSUSOFF bits: High-pass noise prevention can be turned off
- CTSUSDPRS.CTSUPRMODE[1:0] bits: Set the number of base pulses for synchronous noise prevention
- CTSUSDPRS.CTSUPRRATIO[3:0] bits: Set the measurement time for synchronous noise prevention

CTSUSST register: Set the sensor stabilization time
CTSUCHAC0 to CTSUCHAC2 registers: Set the enabled channel
CTSUMCH0 register: Set the measurement channel

- CTSUCR0.CTSUSTRT bit: Set this bit to 1
- CTSUCR0.CTSUCAP bit: Starting of CTSU operation by external trigger can be selected
- CTSUCR0.CTSUSNZ bit: Power-saving function during wait state can be enabled or disabled

Transferred by the DTC when the DTC is selected

Transferred by the DTC when the DTC is selected

Wait for the measurement end interrupt

When a software trigger is used, the CTSUCR0.CTSUSTRT bit sets to 0 when CTSU operation stops.

Figure 51.12 Software flow and example operation for self-capacitance single scan mode

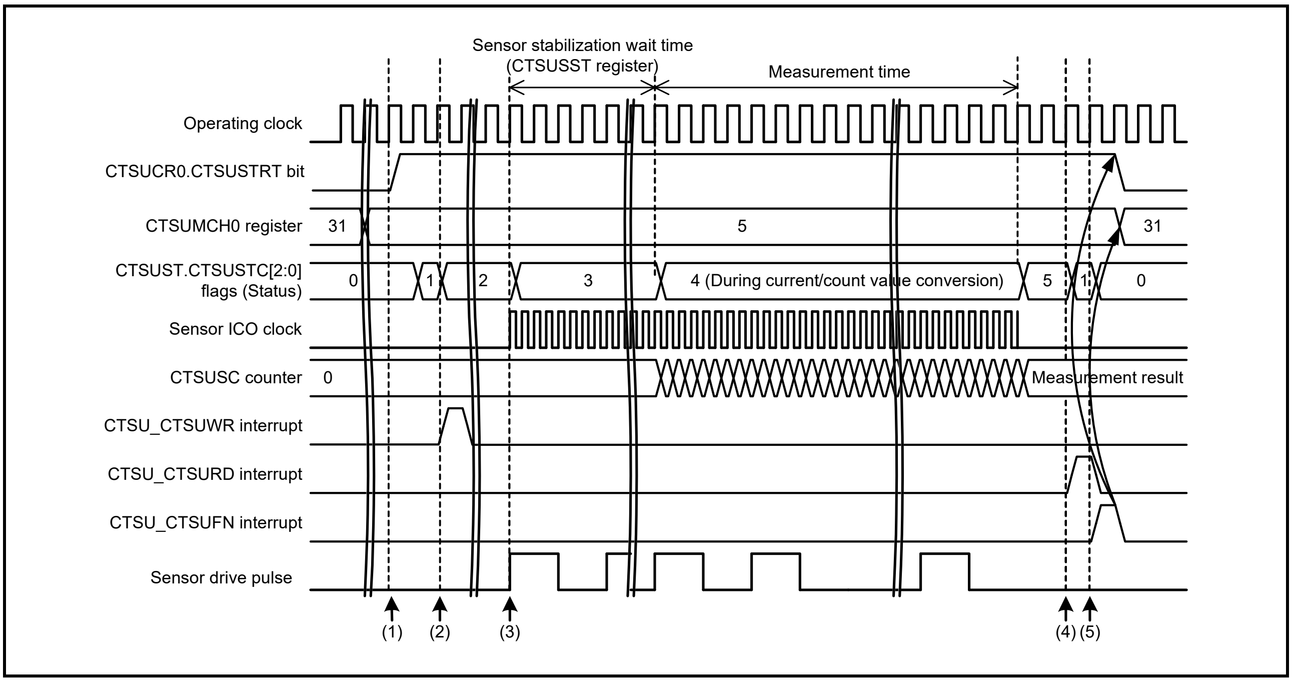


Figure 51.13 Timing of self-capacitance single scan mode when the measurement start condition is a software trigger

The following describes the operation shown in Figure 51.13.

1. After initial settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
2. After the channel to be measured is determined in accordance with the preset conditions, a request for setting the channel (CTSU_CTSUWR) is output.
3. On completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and reference ICO clock operate.
4. After the sensor stabilization wait time and the measurement time elapse, and measurement stops, a measurement result read request (CTSU_CTSURD) is output.
5. A measurement end interrupt (CTSU_CTSUFN) is output and measurement stops (transition to Status 0).

Table 51.7 lists the touch pin states in self-capacitance single scan mode.

Table 51.7 Touch pin states in self-capacitance single scan mode

	Touch pin	
Status	**Measured channel**	**Non-measured channel**
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

51.3.2.4 Self-capacitance multi-scan mode operation

In self-capacitance multi-scan mode, electrostatic capacitance on all channels that are specified as measurement targets in the CTSUCHAC0 to CTSUCHAC2 registers is measured sequentially in ascending order. Figure 51.14 shows the software flow and an operation example, and Figure 51.15 shows the timing.

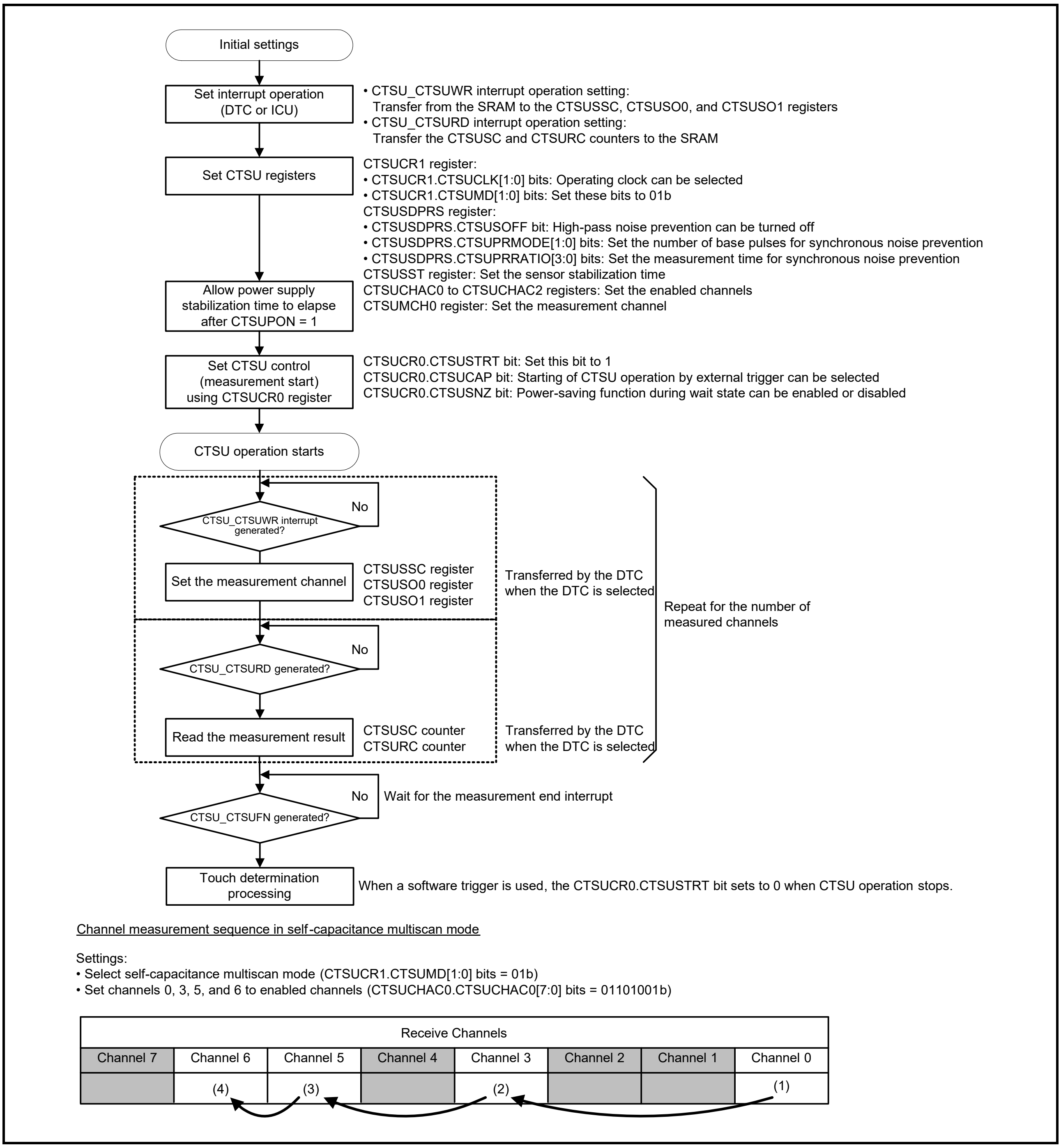


Figure 51.14 Software flow and example operation for self-capacitance multi-scan mode

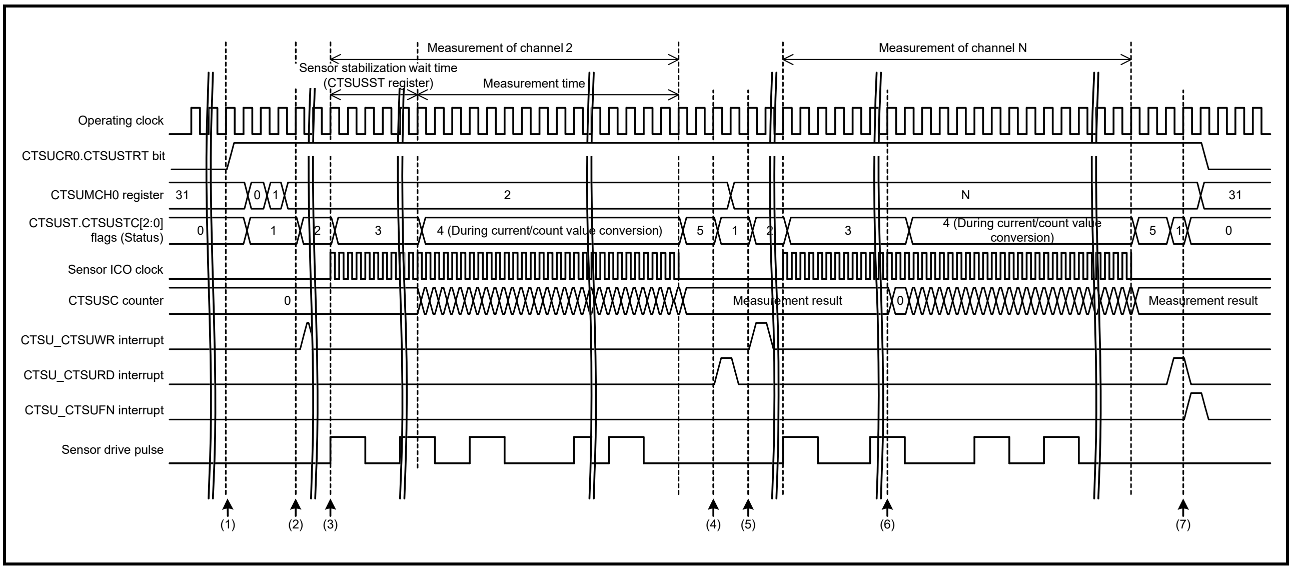


Figure 51.15 Timing of self-capacitance multi-scan mode when the measurement start condition is a software trigger

The following describes the operation shown in Figure 51.15:

1. After initial settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
2. After the channel to be measured is determined in accordance with the preset conditions, a request for setting the channel (CTSU_CTSUWR) is output.
3. On completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
4. After the sensor stabilization wait time and the measurement time elapse, and measurement stops, a measurement result read request (CTSU_CTSURD) is output.
5. After the channel to be measured next is determined, a measurement channel setting request (CTSU_CTSUWR) is output.
6. After the stabilization wait time elapses and when the previous measurement is read, the result is cleared and measurement starts.
7. On completion of all measurement channels, a measurement end interrupt (CTSU_CTSUFN) is output and measurement is stopped (transition to Status 0).

Table 51.8 lists the touch pin states in self-capacitance multi-scan mode.

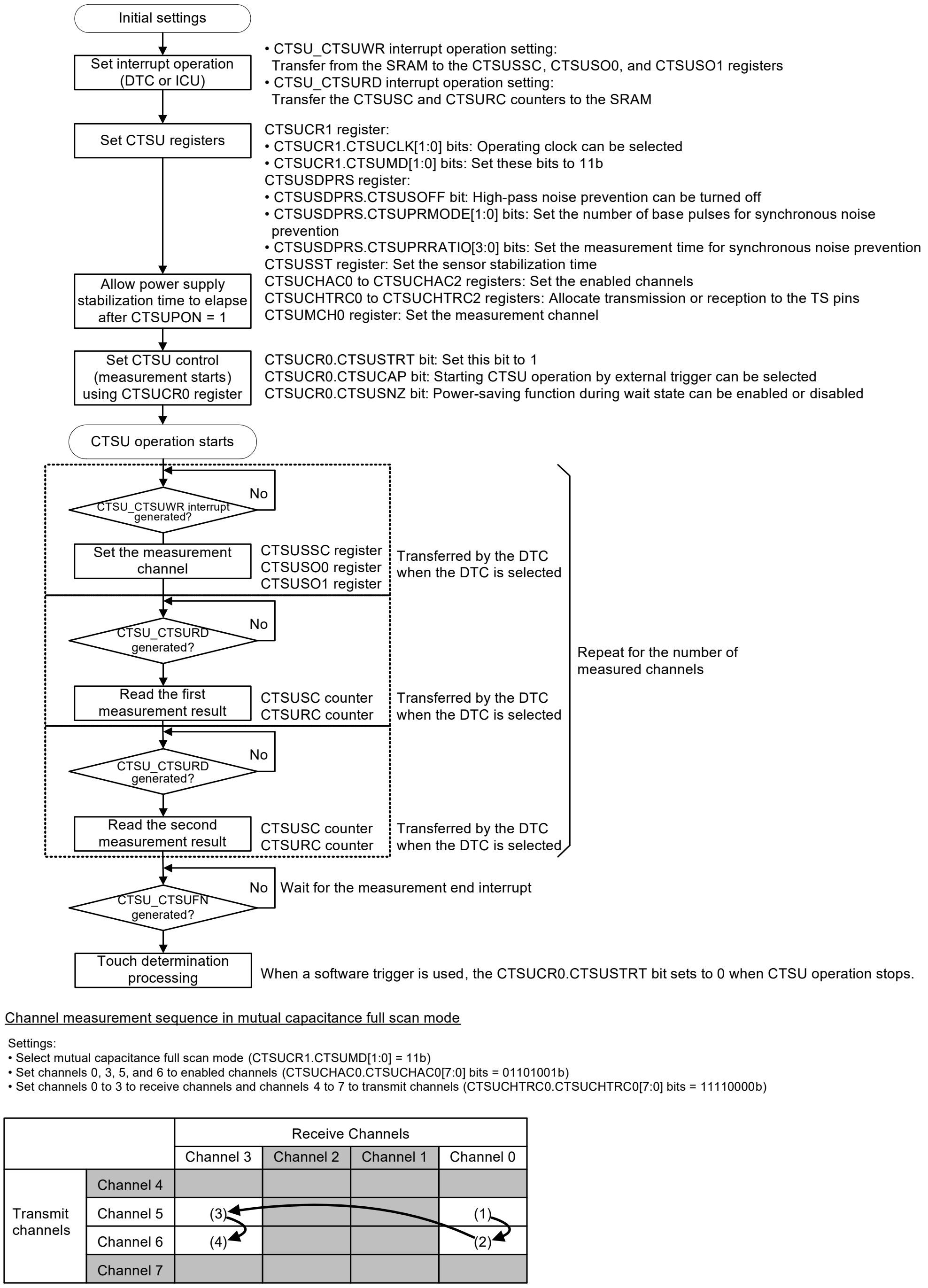
Table 51.8 Touch pin states in self-capacitance multi-scan mode

	Touch pin	
Status	**Measured channel**	**Non-measured channel**
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

51.3.2.5 Mutual capacitance full scan mode operation

In mutual capacitance full scan mode, measurement is performed during the high-level period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, on the rising and falling edges. The difference between the data of these two measurements determines whether or not the electrode was touched, which results in a higher touch sensitivity.

Electrostatic capacitance is measured sequentially on channels set to transmission or reception in the CTSUCHTRC0 to CTSUCHTRC2 registers, and specified as measurement targets in the CTSUCHAC0 to CTSUCHAC2 registers. The capacitance is measured by combining these signals. Figure 51.16 shows the software flow and an operation example, and Figure 51.17 shows the timing.



- CTSU_CTSUWR interrupt operation setting:
  Transfer from the SRAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers
- CTSU_CTSURD interrupt operation setting:
  Transfer the CTSUSC and CTSURC counters to the SRAM

CTSUCR1 register:
- CTSUCR1.CTSUCLK[1:0] bits: Operating clock can be selected
- CTSUCR1.CTSUMD[1:0] bits: Set these bits to 11b

CTSUSDPRS register:
- CTSUSDPRS.CTSUSOFF bit: High-pass noise prevention can be turned off
- CTSUSDPRS.CTSUPRMODE[1:0] bits: Set the number of base pulses for synchronous noise prevention
- CTSUSDPRS.CTSUPRRATIO[3:0] bits: Set the measurement time for synchronous noise prevention

CTSUSST register: Set the sensor stabilization time
CTSUCHAC0 to CTSUCHAC2 registers: Set the enabled channels
CTSUCHTRC0 to CTSUCHTRC2 registers: Allocate transmission or reception to the TS pins
CTSUMCH0 register: Set the measurement channel

CTSUCR0.CTSUSTRT bit: Set this bit to 1
CTSUCR0.CTSUCAP bit: Starting CTSU operation by external trigger can be selected
CTSUCR0.CTSUSNZ bit: Power-saving function during wait state can be enabled or disabled

CTSUSSC register
CTSUSO0 register
CTSUSO1 register
Transferred by the DTC when the DTC is selected

CTSUSC counter
CTSURC counter
Transferred by the DTC when the DTC is selected

CTSUSC counter
CTSURC counter
Transferred by the DTC when the DTC is selected

Repeat for the number of measured channels

Wait for the measurement end interrupt

When a software trigger is used, the CTSUCR0.CTSUSTRT bit sets to 0 when CTSU operation stops.

Channel measurement sequence in mutual capacitance full scan mode

Settings:
- Select mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] = 11b)
- Set channels 0, 3, 5, and 6 to enabled channels (CTSUCHAC0.CTSUCHAC0[7:0] bits = 01101001b)
- Set channels 0 to 3 to receive channels and channels 4 to 7 to transmit channels (CTSUCHTRC0.CTSUCHTRC0[7:0] bits = 11110000b)

		Receive Channels			
		Channel 3	Channel 2	Channel 1	Channel 0
Transmit channels	Channel 4				
	Channel 5	(3)			(1)
	Channel 6	(4)			(2)
	Channel 7				

Figure 51.16 Software flow and example operation for mutual capacitance full scan mode

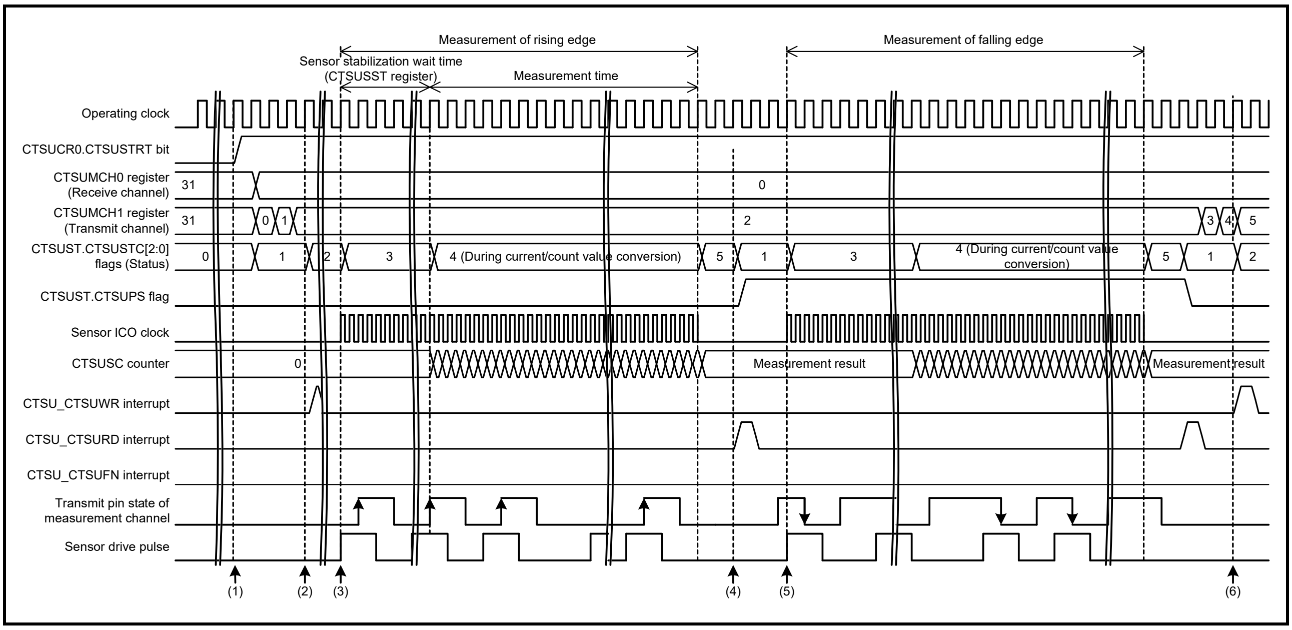


Figure 51.17 Timing of mutual capacitance full scan mode when the measurement start condition is a software trigger

The following describes the operation shown in Figure 51.17.

1. After initial settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
2. After the channel to be measured is determined in accordance with the preset conditions, a request for setting the channel (CTSU_CTSUWR) is output.
3. On completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate. At the same time, a pulse detected on the rising edge is output to the transmit pin on the measurement channel during the high-level period of the sensor drive pulse.
4. After the sensor stabilization wait time and the measurement time elapsed and measurement stops, a measurement result read request (CTSU_CTSURD) is output.
5. The same channel is measured by outputting a pulse detected on the falling edge during the high-level period of the sensor drive pulse.
6. After the same channel is measured twice, the channel to be measured next is determined and measured in the same way.
7. On completion of all measurement channels, a measurement end interrupt (CTSU_CTSUFN) is output and measurement stops (transition to Status 0).

The CTSU mutual capacitance status flag (CTSUST.CTSUPS bit) changes when Status 5 transitions to Status 1.

Table 51.9 lists the touch pin states in mutual capacitance full scan mode.

Table 51.9 Touch pin states in mutual capacitance full scan mode (1 of 2)

	Touch pins for receive channels		Touch pins for transmit channels		
Status	Measured channel	Non-measured channel	Measured channel	Non-measured channel	Remarks
0	Low	Low	Low	Low	-
1	Low	Low	Low/high	Low	-
2	Low	Low	Low	Low	-

Table 51.9 Touch pin states in mutual capacitance full scan mode (2 of 2)

	Touch pins for receive channels		Touch pins for transmit channels		
Status	**Measured channel**	**Non-measured channel**	**Measured channel**	**Non-measured channel**	**Remarks**
3	Pulse	Low	Pulse	Low	The phase pulse is the same as that of the receive channel on the first measurement and opposite on the second measurement.
4	Pulse	Low	Pulse	Low	-
5	Low	Low	Low	Low	-

51.3.3 Parameters Common to Multiple Modes

51.3.3.1 Sensor stabilization wait time and measurement time

Figure 51.18 shows the timing of the sensor stabilization wait and measurement.

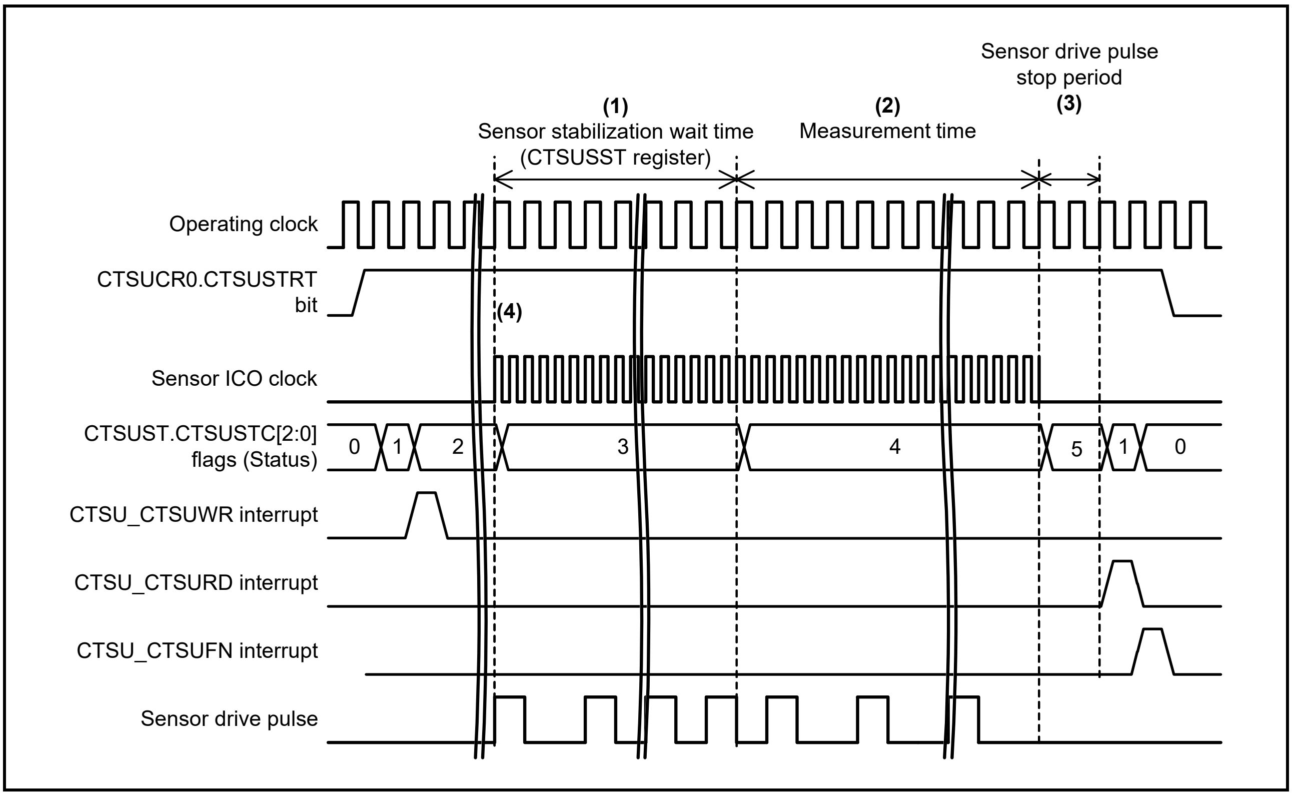


Figure 51.18 Sensor stabilization wait and measurement timing

The following describes the operation shown in Figure 51.18:

1. In response to the CTSU_CTSUWR interrupt request, output of the sensor drive pulse is started by a write access to the CTSUSO1 register. The CTSU waits for the stabilization time set in the CTSUSST register.
2. When the sensor stabilization time elapses and the CTSUST.CTSUDTSR flag clears to 0, measurement starts on transition to Status 4. The measurement time is determined by the base clock cycle setting and the CTSUSDPRS.CTSUPRMODE[1:0], CTSUPRRATIO[3:0], and CTSUSO0.CTSUSNUM[5:0] bits. When the measurement time elapses, measurement of the channel stops.
3. After the measurement time elapses, the status transitions to Status 1 after two operating clock cycles, and a CTSU_CTSURD interrupt occurs. Read the data from the CTSUSC and CTSURC counters. At this time, the sensor drive pulse is output at the low level. When measurement of all specified channels is completed, the CTSUCR0.CTSUSTRT bit clears to 0.
4. The sensor ICO clock oscillates while the CTSUST.CTSUSTC[2:0] flags = 011b (Status 3) or 100b (Status 4).

51.3.3.2 Interrupts

The CTSU supports the following interrupts:

- Write request interrupt for setting registers for each channel (CTSU_CTSUWR)
- Measurement data transfer request interrupt (CTSU_CTSURD)
- Measurement end interrupt (CTSU_CTSUFN).

(1) Write request interrupt for setting registers for each channel (CTSU_CTSUWR)

Store the settings for each measurement channel in the SRAM, and set up the DTC or ICU transfer associated with the CTSU_CTSUWR interrupt in advance. The CTSU_CTSUWR interrupt is output when Status 1 transitions to Status 2. Write the settings for the selected channel from the SRAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers (Figure 51.19). Because write access to the CTSUSO1 register controls the transition to the next status, you must set this register last.

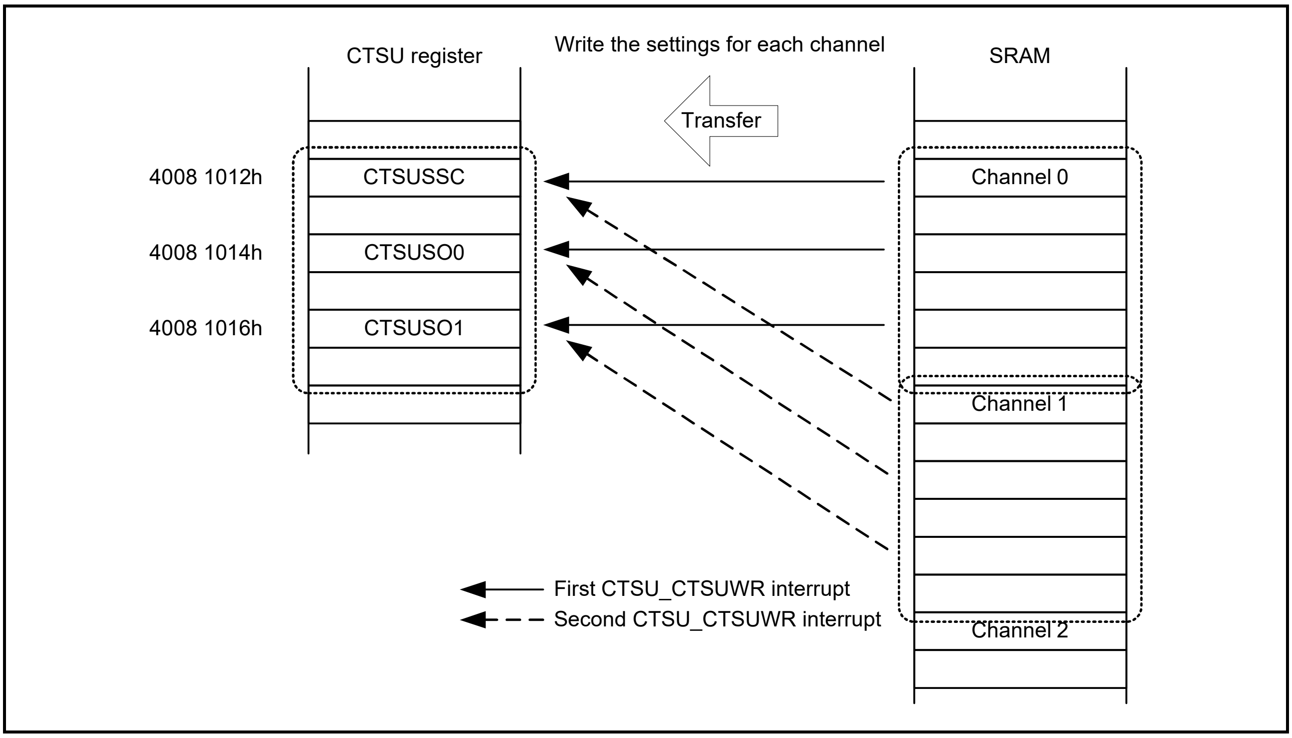


Figure 51.19 Example of DTC transfer operation using the CTSU_CTSUWR interrupt

The registers to be set (CTSUSSC, CTSUSO0, and CTSUSO1) are allocated at sequential addresses. On CTSU_CTSUWR interrupt generation, set up the operation as follows:

- Transfer destination address: CTSUSSC register address
- Handling at the transfer destination address: Transfer 2-byte data three times with a single interrupt. The address of the start byte is fixed
- Transfer source address: CTSUSSC register data storage address for the lowest number channel in the settings stored in the SRAM
- Handling at the transfer source address: Transfer 2-byte data three times with a single interrupt. The address of the first byte is continued from the previous interrupt handling
- Number of transfers per interrupt: Specify the number of measurements.

(2) Measurement data transfer request interrupt (CTSU_CTSURD)

Set up the DTC or ICU transfer associated with the CTSU_CTSURD interrupt in advance. The CTSU_CTSURD interrupt is output when Status 5 transitions to Status 1. Read the measurement result from the CTSUSC and CTSURC counters (Figure 51.20).

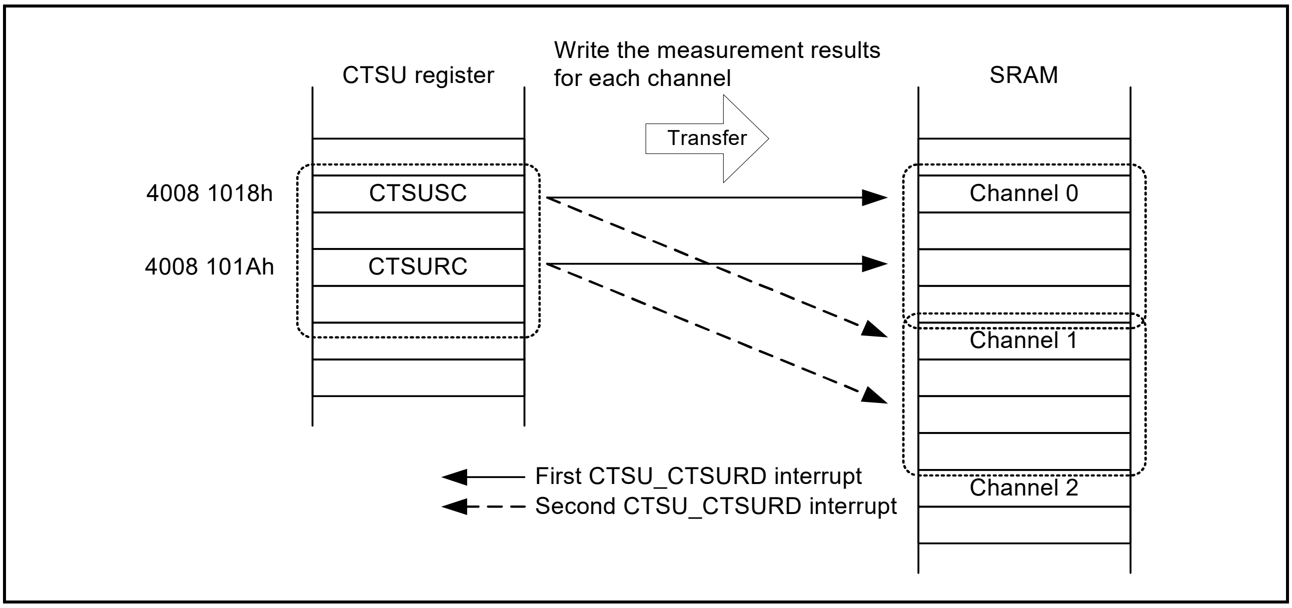


Figure 51.20 Example of DTC transfer operation using the CTSU_CTSURD interrupt

The measurement result registers (CTSUSC and CTSURC counters) used as transfer sources are allocated at sequential addresses. On CTSU_CTSURD interrupt generation, set up the operation as follows:

- Transfer source address: CTSUSC counter address
- Handling at the transfer source address: Transfer 2-byte data twice with a single interrupt. The start address is fixed
- Transfer destination address: CTSUSC counter data storage address for the lowest number channel in the settings stored in the SRAM
- Handling at the transfer destination address: Transfer 2-byte data twice with a single interrupt. The start address is continued from the previous interrupt handling
- Number of transfers per interrupt: Specify the number of measurements.

(3) Measurement end interrupt (CTSU_CTSUFN)

When all channels are measured, an interrupt occurs when Status 1 transitions to Status 0. In the software, check the overflow flags (CTSUST.CTSUSOVF and CTSUROVF) and read the measurement results to determine whether or not the electrode was touched. Interrupt requests are accepted or disabled in the interrupt control block.

51.4 Usage Notes

51.4.1 Measurement Result Data (CTSUSC and CTSURC Counters)

Read access during measurement is prohibited. If the measurement result data is accessed, an incorrect value might be read because of asynchronous operation.

51.4.2 Constraint on Software Trigger

When 10b (PCLKB/4) is selected in the CTSUCR1.CTSUCLK[1:0] bits, to restart measurement by writing 1 to the CTSUR0.CTSUSTRT bit after measurement is complete, wait for at least 3 cycles to elapse after an interrupt occurs, and then write to the CTSUCR0.CTSUSTRT bit.

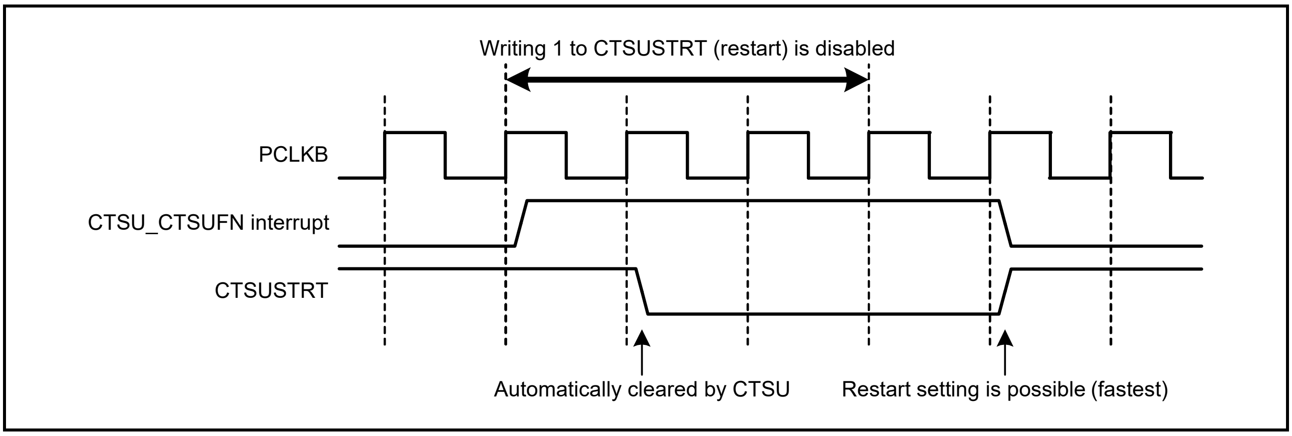


Figure 51.21 Notes on restarting measurement

51.4.3 Constraints on External Triggers

- If an external trigger is input during the measurement time, measurement does not start. The next external event is enabled after 1 cycle of the operating clock when a CTSU_CTSUFN interrupt occurs.
- To stop external trigger mode, write 0 to the CTSUCR0.CTSUSTRT bit and 0 to the CTSUCR0.CTSUINIT bit at the same time (forced stop).

51.4.4 Constraints on Forced Stops

To force the current operation to stop, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time. After this setting, the operation is stopped and the internal control registers are initialized.

When the CTSUCR0.CTSUINIT bit is used for initialization, the following registers are initialized in addition to the initialization of the internal measurement state.

- CTSUMCH0 register
- CTSUMCH1 register
- CTSUST register
- CTSUSC counter
- CTSURC counter.

If operation is forced to stop, an interrupt request might be generated depending on the internal state. After a forced stop, also perform the processing for stopping and disabling the DTC or ICU. If a DTC transfer is stopped in an installed system for some reason, also perform the processing for forcing stop to and initializing the CTSU.

51.4.5 TSCAP Pin

The TSCAP pin requires an external decoupling capacitor to stabilize CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible.

The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output a low level, before turning on the switch (CTSUCR1.CTSUCSW bit = 1) to establish a connection.

51.4.6 Constraints on Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1)

During measurement (CTSUCR0.CTSUSTRT bit = 1), do not use the settings for stopping the peripheral clock or changing the port settings related to the touch pins (TSn and TSCAP pins) and Transmission power supply selection (CTSUCR0.CTSUTXVSEL) in the higher layers of the system.

If control settings non-compliant with these constraints are made, operation is forced to stop (CTSUCR0.CTSUSTRT bit = 0 and CTSUCR0.CTSUINIT bit = 1), write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit at the same time, and set the CTSUCR0.CTSUSNZ bit to 0. Next, restart from the initial settings flow shown in Figure 51.9.

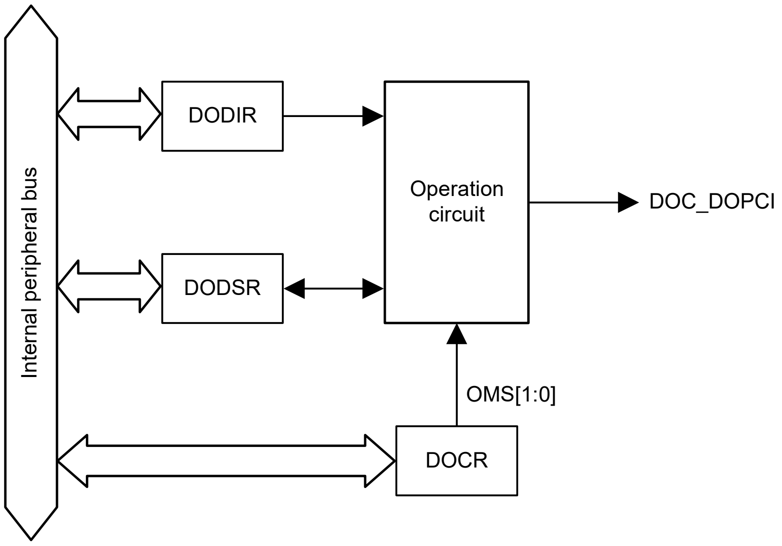
52. Data Operation Circuit (DOC)

52.1 Overview

The Data Operation Circuit (DOC) is used to compare, add, and subtract 16-bit data. When a selected condition applies, 16-bit data is compared and an interrupt can be generated. Table 52.1 lists the DOC specifications and Figure 52.1 shows a block diagram.

Table 52.1 DOC specifications

Parameter	Specifications
Data operation function	16-bit data comparison, addition, and subtraction
Module-stop function	Module-stop state can be set to reduce power consumption
Interrupts and event link function (DOC_DOPCI)	An interrupt occurs on the following conditions: • Compared values either match or mismatch • Data addition result is greater than FFFFh • Data subtraction result is less than 0000h



DOCR: DOC Control Register
DODIR: DOC Data Input Register
DODSR: DOC Data Setting Register

Figure 52.1 DOC block diagram

52.2 Register Descriptions

52.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 4005 4100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	—	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited.	R/W
b2	DCSEL*1	Detection Condition Select	0: Set DOPCF when data mismatch detected 1: Set DOPCF when data match detected.	R/W
b4, b3	—	Reserved	There bits are read as 0. The write value should be 0.	R/W
b5	DOPCF	Data Operation Circuit Flag	Indicates the result of an operation.	R
b6	DOPCFCL	DOPCF Clear	0: Retain DOPCF flag state 1: Clear DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only valid when data comparison mode is selected.

OMS[1:0] bits (Operating Mode Select)

The OMS[1:0] bits select the operating mode of the DOC.

DCSEL bit (Detection Condition Select)

The DCSEL bit selects the detection condition in data comparison mode. This bit is only valid when data comparison mode is selected.

DOPCF flag (Data Operation Circuit Flag)

The DOPCF flag indicates the result of an operation.

[Setting conditions]

- The condition selected in the DCSEL bit is met
- A data addition result is greater than FFFFh
- A data subtraction result is less than 0000h.

[Clearing condition]

- Writing 1 to the DOPCFCL bit.

DOPCFCL bit (DOPCF Clear)

Setting the DOPCFCL bit to 1 clears the DOPCF flag. This bit is read as 0.

52.2.2 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 4005 4102h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DODIR is a 16-bit read/write register that stores 16-bit data used in the operations.

52.2.3 DOC Data Setting Register (DODSR)

Address(es): DOC.DODSR 4005 4104h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DODSR is a 16-bit read/write register that stores 16-bit data used as a reference in data comparison mode. This register also stores the results of operations in data addition and subtraction modes.

52.3 Operation

52.3.1 Data Comparison Mode

Figure 52.2 shows an example of the steps involved in data comparison mode operation by the DOC. The following is an example operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison):

1. Write 00b to the DOCR.OMS[1:0] bits to select data comparison mode.
2. Set 16-bit reference data in DODSR.
3. Write 16-bit data for comparison to DODIR.
4. Continue writing 16-bit data until all data for comparison is written to DODIR.
5. If a value written to DODIR does not match that in DODSR,[*1] the DOCR.DOPCF flag sets to 1.

Note 1. When DOCR.DCSEL = 0

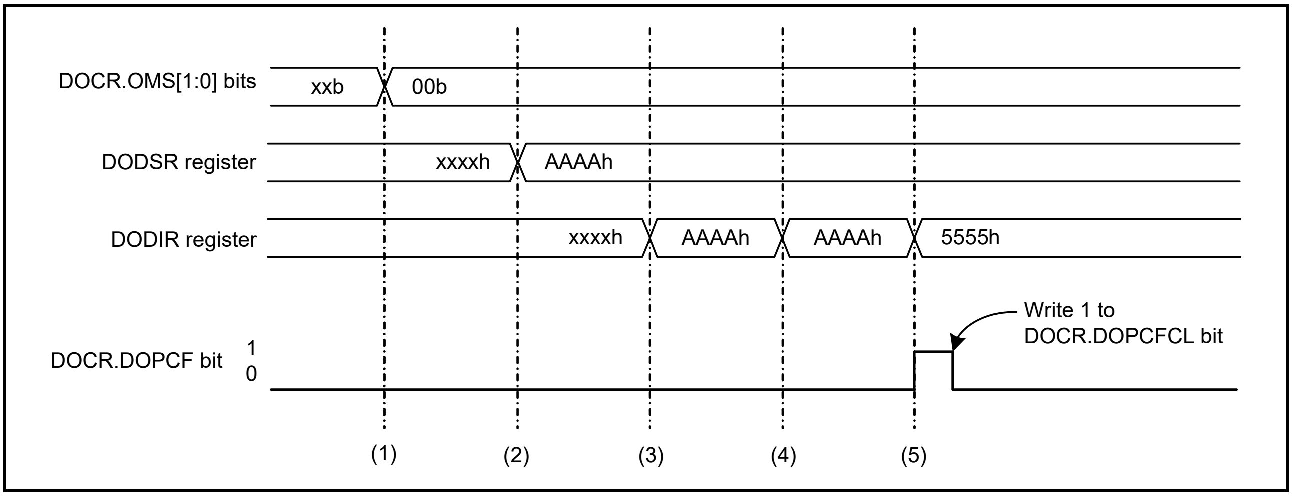


Figure 52.2 Example of operation in data comparison mode

52.3.2 Data Addition Mode

Figure 52.3 shows an example of the steps involved in data addition mode operation by the DOC:

1. Write 01b to the DOCR.OMS[1:0] bits to select data addition mode.
2. Set 16-bit data in the DODSR register as the initial value.
3. Write 16-bit data to be added to DODIR. The result of the operation is stored in DODSR.
4. Continue writing 16-bit data until all data for addition is written to DODIR.
5. If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag sets to 1.

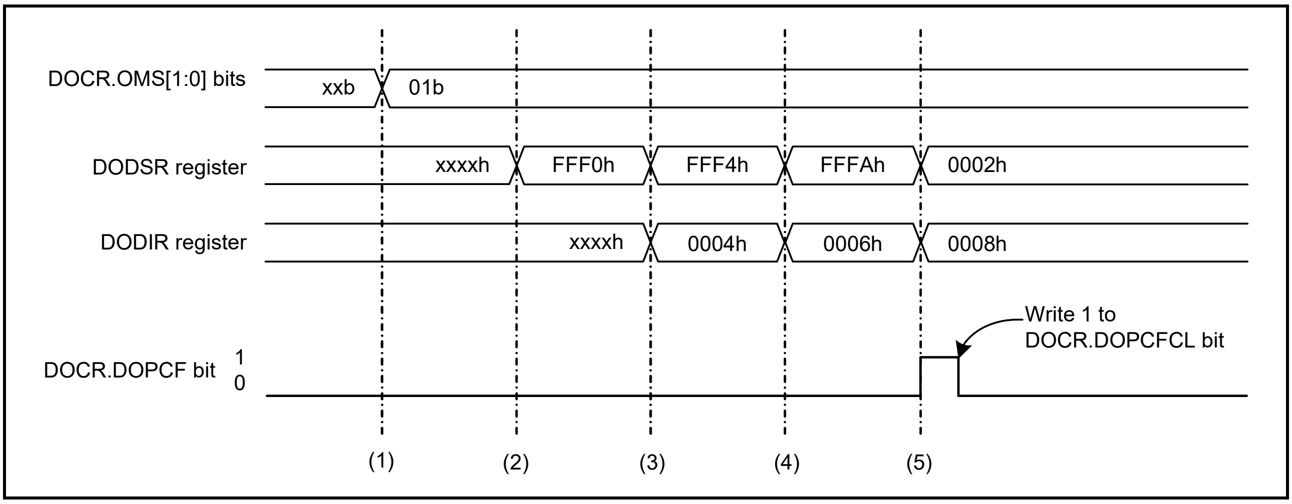


Figure 52.3 Example of operation in data addition mode

52.3.3 Data Subtraction Mode

Figure 52.4 shows an example of the steps involved in data subtraction mode operation by the DOC:

1. Write 10b to the DOCR.OMS[1:0] bits to select data subtraction mode.
2. Set 16-bit data in the DODSR register as the initial value.
3. Write 16-bit data to be subtracted to DODIR. The result of the operation is stored in DODSR.
4. Continue writing 16-bit data until all data for subtraction is written to DODIR.
5. If the result of an operation is less than 0000h, the DOCR.DOPCF flag sets to 1.

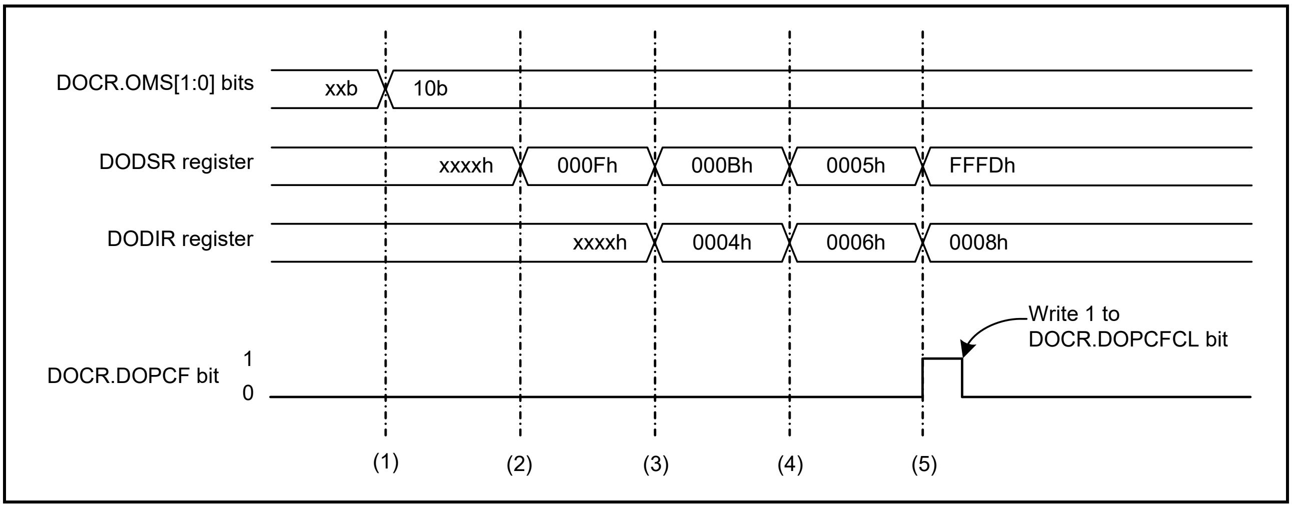


Figure 52.4 Example of operation in data subtraction mode

52.4 Interrupt Request and Output to the Event Link Controller (ELC)

The DOC outputs an event signal for the ELC under the following conditions:

- The compared values either match or mismatch
- The data addition result is greater than FFFFh
- The data subtraction result is less than 0000h.

This signal can be used to initiate operations by other modules selected in advance and also can be used as an interrupt request. When an event signal is generated, the Data Operation Circuit Flag (DOCR.DOPCF) sets to 1.

52.5 Usage Notes

52.5.1 Settings for the Module-Stop Function

DOC operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The DOC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

53. SRAM

53.1 Overview

The MCU provides on-chip high-speed SRAM modules with either parity-bit checking or ECC (Error Correction Code). The area of the first 32 KB of SRAM0 is subject to ECC. Parity check is performed on other areas. Table 53.1 lists the SRAM specifications.

Table 53.1 SRAM specifications

Parameter	Specifications without ECC	Specifications with ECC	SRAMHS specifications
SRAM capacity	SRAM0: 224 KB SRAM1: 256 KB	SRAM0 (ECC area): 32 KB	SRAMHS: 128 KB
SRAM addresses*2	SRAM0: 2000 8000h to 2003 FFFFh SRAM1: 2004 0000h to 2007 FFFFh	SRAM0 (ECC area): 2000 0000h to 2000 7FFFh	SRAMHS: 1FFE 0000h to 1FFF FFFFh
Access*1	Wait states are inserted into the read cycle by default. If the ICLK frequency is faster than 60 (up to 120) MHz, a wait state is required. If the ICLK frequency is 60 MHz or less, a wait state is not required. For details, see section 53.4, Usage Notes.	Wait states are inserted into the read cycle by default. If the ICLK frequency is faster than 60 (up to 120) MHz, wait state is required. If the ICLK frequency is 60 MHz or less, a wait state is not required. For details, see section 53.4, Usage Notes.	Access to the SRAMHS is always no wait state.
Data retention	Not available in Deep Software Standby mode		
Module-stop function	Module-stop state can be set to reduce power consumption		
Parity	Even-parity (data: 8 bits, parity: 1 bit)	No parity	Even-parity (data: 8 bits, parity: 1 bit)
Error checking	Even-parity error check	Detection up to 2-bit errors	Even-parity error check

Note 1. For details, see section 53.3.7, Access Cycles.

Note 2. The Cortex®-M4 processor supports Arm®v7 unaligned accesses. In this product, SRAMHS and SRAM0 are adjacent to each other and there is an access boundary between them. Therefore, when SRAMHS and SRAM0 are used as a continuous area of memory space, access that straddles the boundary must not be produced as this might lead to access of data other than what is intended. For details, see the *ARM® Cortex®-M4 Processor Technical Reference Manual*.

53.2 Register Descriptions

53.2.1 SRAM Parity Error Operation After Detection Register (PARIOAD)

Address(es): SRAM.PARIOAD 4000 2000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OAD	Operation after Detection	1: Reset 0: Non-maskable interrupt.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PARIOAD register controls the operation on detection of a parity error. The SRAM Protection Register (SRAMPRCR) protects this register against writing. Set the SRAMPRCR bit in the SRAMPRCR register to the enabled setting before writing to this register. Do not write to the PARIOAD register while access to the SRAM is in progress.

OAD bit (Operation after Detection)

The OAD bit specifies generation of either a reset or non-maskable interrupt when a parity error is detected. The OAD bit in the PARIOAD register is shared by SRAM0 (without ECC), SRAM1, SRAMHS, and the Standby SRAM.

53.2.2 SRAM Protection Register (SRAMPRCR)

Address(es): SRAM.SRAMPRCR 4000 2004h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							SRAMPRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SRAMPRCR	Register Write Control	0: Disable writes to protected registers 1: Enable writes to protected registers.	R/W
b7 to b1	KW[6:0]	Write Key Code	These bits enable or disable writing of the SRAMPRCR bit.	R/W

SRAMPRCR bit (Register Write Control)

The SRAMPRCR bit controls the write mode of the SRAMWTSC and PARIOAD registers. When the bit is set to 1, writing to the SRAMWTSC and PARIOAD registers is enabled. When you write to this bit, write 78h to the KW[6:0] bits simultaneously.

KW[6:0] bits (Write Key Code)

The KW[6:0] bits enable or disable writes to the SRAMPRCR bit. When you write to the SRAMPRCR bit, write 78h to the KW[6:0] bits simultaneously. When a value other than 78h is written to KW[6:0], the SRAMPRCR bit is not updated. The KW[6:0] bits are always read as 0.

53.2.3 SRAM Wait State Control Register (SRAMWTSC)

Address(es): SRAM.SRAMWTSC 4000 2008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SRAM1WTEN	SRAM0WTEN	SRAM0ERDWTEN	—
Value after reset:	0	0	0	0	1	1	1	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SRAM0ERDWTEN	SRAM0 (ECC area) Read Wait Enable	1: Add wait state in read access cycle to SRAM0 (ECC area) 0: Do not add wait state in read access cycle to SRAM0 (ECC area).	R/W
b2	SRAM0WTEN	SRAM0 Wait Enable	1: Add wait state in read access cycle to SRAM0 0: Do not add wait state in read access cycle to SRAM0.	R/W
b3	SRAM1WTEN	SRAM1 Wait Enable	1: Add wait state in read access cycle to SRAM1 0: Do not add wait state in read access cycle to SRAM1.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SRAMWTSC register controls the wait states in the SRAM modules. The SRAM Protection Register (SRAMPRCR) protects this register against writing. Set the SRAMPRCR bit in the SRAMPRCR register to the enabled setting before writing to this register. Do not write to the SRAMWTSC register while access to the SRAM is in progress.

SRAM0ERDWTEN bit (SRAM0 (ECC area) Read Wait Enable)

The SRAM0ERDWTEN bit enables the wait cycle for reads from the ECC area in SRAM0. When this bit is set to 1, one wait cycle is inserted into the read cycle of the ECC area in SRAM0. When the read access frequency is more than 60 MHz, this one-wait cycle setting is required.

SRAM0WTEN bit (SRAM0 Wait Enable)

The SRAM0WTEN bit enables the wait cycle for reads from SRAM0. When this bit is set to 1, one wait cycle is inserted into the read cycle of SRAM0. When the read access frequency is more than 60 MHz, this one-wait cycle setting is required.

SRAM1WTEN bit (SRAM1 Wait Enable)

The SRAM1WTEN bit enables the wait cycle for reads from SRAM1. When this bit is set to 1, one wait cycle is inserted into the read cycle of SRAM1. When the read access frequency is more than 60 MHz, this one-wait cycle setting is required.

53.2.4 ECC Operating Mode Control Register (ECCMODE)

Address(es): SRAM.ECCMODE 4000 20C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	ECCMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	ECCMOD[1:0]	ECC Operating Mode Select	b1 b0 0 0: Disable ECC function 0 1: Setting prohibited 1 0: Enable ECC function without error checking 1 1: Enable ECC function with error checking.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ECCMODE register specifies the ECC operating mode. The ECC Protection Register (ECCPRCR) protects this register against writing. Set the ECCPRCR bit in the ECCPRCR register to the enabled setting before writing to this register. Do not write to the ECCMODE register while access to the SRAM is in progress.

ECCMOD[1:0] bit (ECC Operating Mode Select)

The ECCMOD[1:0] bit sets the access mode for the ECC area in SRAM0.

53.2.5 ECC 2-Bit Error Status Register (ECC2STS)

Address(es): SRAM.ECC2STS 4000 20C1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ECC2ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ECC2ERR	ECC 2-Bit Error Status	0: No 2-bit ECC error occurred 1: 2-bit ECC error occurred.	R(/W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the bit.

ECC2ERR bit (ECC 2-Bit Error Status)

The ECC2ERR bit indicates whether a 2-bit ECC error occurred in the ECC area of the SRAM. When ECC operations are enabled and error correction is selected, this bit sets to 1 if a 2-bit error is detected. The SRAM error signal is also asserted. Writing 0 to the ECC2ERR bit negates the SRAM error signal triggered by the 2-bit ECC error.

The SRAM error can be specified as a non-maskable interrupt or reset in the ECCOAD register. Do not access the ECC area in the SRAM while writing 0 to this register.

53.2.6 ECC 1-Bit Error Information Update Enable Register (ECC1STSEN)

Address(es): SRAM.ECC1STSEN 4000 20C2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	E1STSEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	E1STSEN	ECC 1-Bit Error Information Update Enable	0: Disable updating of 1-bit ECC error information 1: Enable updating of 1-bit ECC error information.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ECC1STSEN register enables or disables updating of the ECC 1-Bit Error Status Register (ECC1STS) in response to a 1-bit ECC error in the SRAM (ECC area). The ECC Protection Register (ECCPRCR) protects this register against writing. Set the ECCPRCR bit in the ECCPRCR register to the enabled setting before writing to this bit.

E1STSEN bit (ECC 1-Bit Error Information Update Enable)

The E1STSEN bit enables or disables updating of the SRAM (ECC area) 1-Bit Error Status Register (ECC1STS) in response to a 1-bit error in the ECC area of SRAM. The register also functions as an interrupt and reset mask.

53.2.7 ECC 1-Bit Error Status Register (ECC1STS)

Address(es): SRAM.ECC1STS 4000 20C3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ECC1ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ECC1ERR	ECC 1-Bit Error Status	0: No 1-bit ECC error occurred 1: 1-bit ECC error occurred.	R(/W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the bit.

ECC1ERR bit (ECC 1-Bit Error Status)

The ECC1ERR bit indicates whether a 1-bit ECC error occurred in the ECC area of the SRAM. When ECC operations are enabled and error correction is selected, and updating of the 1-bit error information is enabled, this bit sets to 1 if a 1-bit error is detected. The SRAM error signal is also asserted. Writing 0 to the ECC1ERR bit negates the SRAM error signal triggered by the 1-bit ECC error.

The SRAM error can be specified as a non-maskable interrupt or reset in the ECCOAD register. Do not access the ECC area in the SRAM while writing 0 to this register.

53.2.8 ECC Protection Register (ECCPRCR)

Address(es): SRAM.ECCPRCR 4000 20C4h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							ECCPRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ECCPRCR	Registers Write Control	0: Disable writes to protected registers 1: Enable writes to protected registers.	R/W
b7 to b1	KW[6:0]	Write Key Code	These bits enable or disable writes to the ECCPRCR bit.	R/W

ECCPRCR bit (Registers Write Control)

The ECCPRCR bit controls the write mode of the ECCMODE, ECC1STSEN, and ECCOAD registers. When the bit is set to 1, writing to the ECCMODE, ECC1STSEN, and ECCOAD registers is enabled. When you write to this bit, write 78h to the KW[6:0] simultaneously.

KW[6:0] bits (Write Key Code)

The KW[6:0] bits enable or disable writes to the ECCPRCR bit. When you write to the ECCPRCR bit, write 78h to the KW[6:0] bits simultaneously. When a value other than 78h is written to KW[6:0], the ECCPRCR bit is not updated. The KW[6:0] bits are always read as 0.

53.2.9 ECC Test Control Register (ECCETST)

Address(es): SRAM.ECCETST 4000 20D4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TSTBYP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSTBYP	ECC Bypass Select	0: ECC bypass disabled 1: ECC bypass enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ECC Protection Register (ECCPRCR2) protects this register against writing. Change the ECCPRCR2 bit in the ECCPRCR2 register to the enabled setting before writing to this bit. Do not write to the ECCETST register while access to SRAM is in progress.

TSTBYP bit (ECC Bypass Select)

The TSTBYP bit enables direct access to the ECC code by bypassing ECC function. The ECC bypass function is used with the ECCMOD[1:0] bits in the ECCMODE register to 00b. Access the same address with 32-bit access size as the data that is checked by ECC. The lower 7 bits of 32-bit write data can be written as an ECC code when the ECC bypass is enabled. The higher 25 bits in the write data are then ignored. The lower 7 bits of the 32-bit read data can be used as ECC code. The higher 25 bits in the read data are unknown.

Note: For details of ECC test, see section 53.3.4, ECC Decoder Testing.

53.2.10 SRAM ECC Error Operation After Detection Register (ECCOAD)

Address(es): SRAM.ECCOAD 4000 20D8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OAD	Operation after Detection	1: Reset 0: Non-maskable interrupt.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ECCOAD register controls the operation on detection of a ECC error. The ECC Protection Register (ECCPRCR) protects this register against writing. Set the ECCPRCR bit in the ECCPRCR register to the enabled setting before writing to this register. Do not write to the ECCOAD register while access to the SRAM is in progress.

OAD bit (Operation after Detection)

The OAD bit specifies generation of either a reset or a non-maskable interrupt when a ECC error is detected. The OAD bit in the ECCOAD register is used for the SRAM (ECC area).

53.3 Operation

53.3.1 Low-Power Functions

Power consumption can be reduced by setting Module Stop Control Register A (MSTPCRA) to stop supply of the clock signal to the SRAM. The control bits are as follows for each module:

- Setting both the MSTPA0 and MSTPA6 bits in the MSTPCRA to 1 stops supply of the clock signal to SRAM0*1
- Setting the MSTPA1 bit in the MSTPCRA to 1 stops supply of the clock signal to SRAM1
- Setting the MSTPA5 bit in MSTPCRA to 1 stops supply of the clock signal to the SRAMHS
- Setting the MSTPA7 bit in the MSTPCRA to 1 stops supply of the clock signal to the Standby SRAM.

Stopping the clock signal supply places the SRAM in the module-stop state. The SRAM operates after a reset.

The SRAM is not accessible in the module-stop state. Do not transition to the module-stop state while access to the SRAM is in progress. Access to the SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.

For details on the MSTPCRA register, see section 11, Low Power Modes.

Note 1. The settings in the MSTPA0 and MSTPA6 bits in the MSTPCRA register must be the same.

53.3.2 ECC Function

Enabling and disabling of ECC error correction can be selected through ECCMODE register setting. In the initial state, ECC error correction is disabled. The ECC check type is SEC-DED (Single-Error Correction and Double-Error Detection Code).

When ECC function is enabled, 7-bit check bits are appended to 32-bit data for writing. For reading, 39-bit (data: 32 bits, check bits: 7 bits) data is read out from the SRAM (ECC area).

When ECC function is enabled and error checking is enable, error correction is done if a 1-bit error occurs and the ECC1ERR bit in the ECC1STS register is set to 1 if the E1STSEN bit in the ECC1STSEN register is 1; if a 2-bit error occurs, error detection is done and the ECC2ERR bit in the ECC2STS register is set to 1, though error correction is not performed.

When ECC function is enabled and the error checking is disable, error correction is done if a 1-bit error occurs but

ECC1ERR bit in the ECC1STS register is not updated although E1STSEN bit in the ECC1STSEN register is 1; if a 2-bit error occurs, this error is detected but the ECC2ERR bit in the ECC2STS register is not updated, and error correction is not performed.

When ECC function is disable, neither error correction nor error detection is done although 1-bit or 2-bit error occur. So ECC1ERR bit and ECC2ERR bit are not updated.

There is no way to confirm the location where the error was found. Therefore, when after the occurrence of an error, update all the data.

When updating all the data after the occurrence of an error, the only support of 32-bit data writing.

Because the SRAM data is undefined after power on and release from deep software standby mode, accessing the SRAM when ECC function is enabled and error checking is selected causes an ECC error to occur. Therefore, before using ECC function, initial writing with 32-bit data size to the area to be used in the SRAM should be done.

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, please do not perform the read access in a row after the write access.

53.3.3 ECC Error Generation

When ECC function is enabled and error checking is applied to the SRAM (ECC area), an ECC error occurs when either the ECC2ERR bit in the ECC2STS register or the ECC1ERR bit in the ECC1STS register becomes 1 to indicate that ECC checking revealed a 2-bit error or a 1-bit error, respectively.

An ECC error is output with a pulse width of ICLK. When the ECC 1-bit error is to be masked, set the ECC1STSEN.E1STSEN bit to 0 to disable updating of the ECC1ERR bit. An ECC error will not be generated while ECC function is disabled or when ECC function is enabled but error checking is not selected.

ECC error can choose non maskable interrupt or reset by ECCOAD register. When set 1 in the OAD bit of the ECCOAD register, ECC error is output to the reset function. When set 0 in the OAD bit of the ECCOAD register, ECC Error interrupt is output to the ICU as non-maskable interrupt.

53.3.4 ECC Decoder Testing

Figure 53.1 shows the ECC decoder testing.

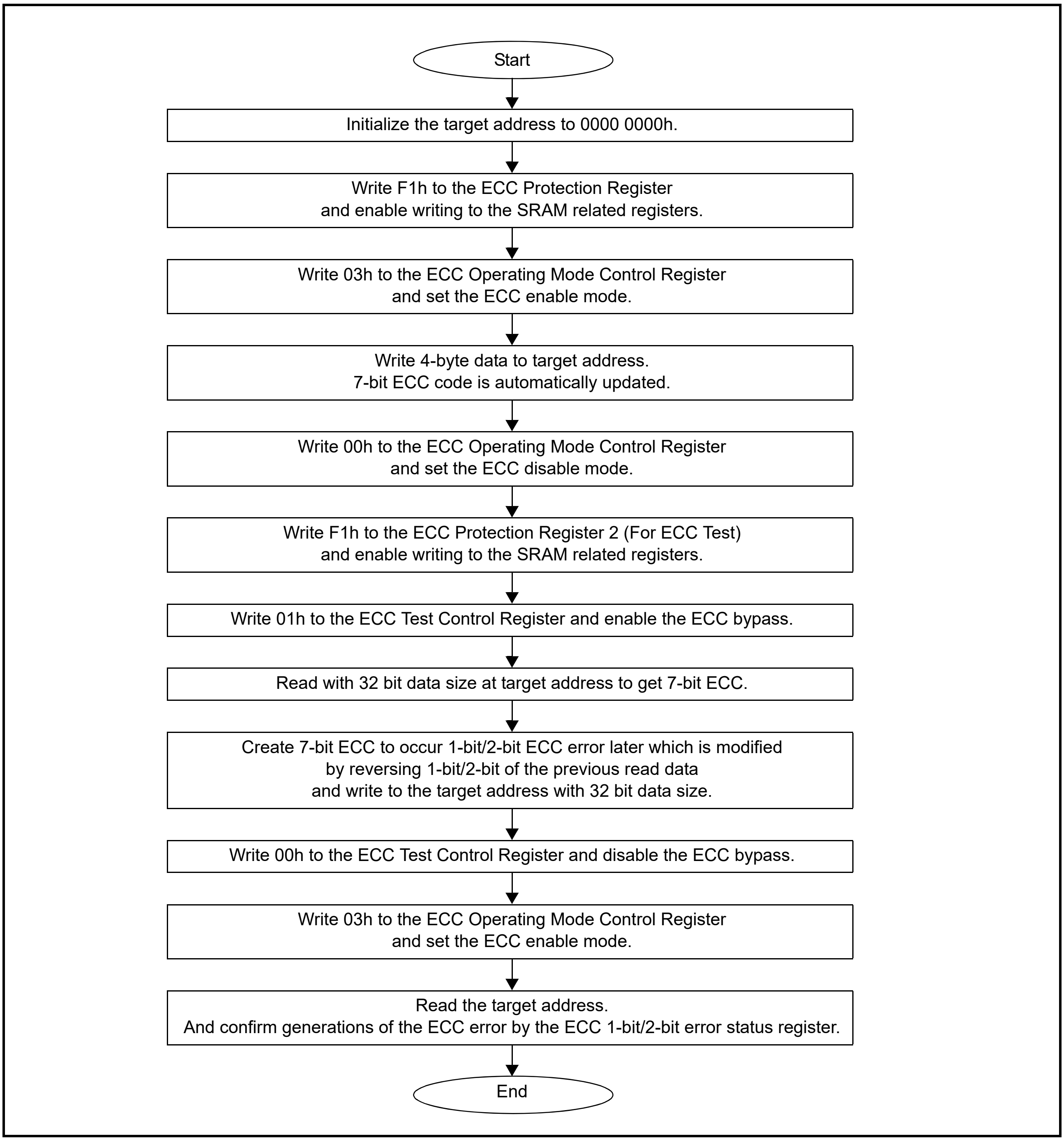


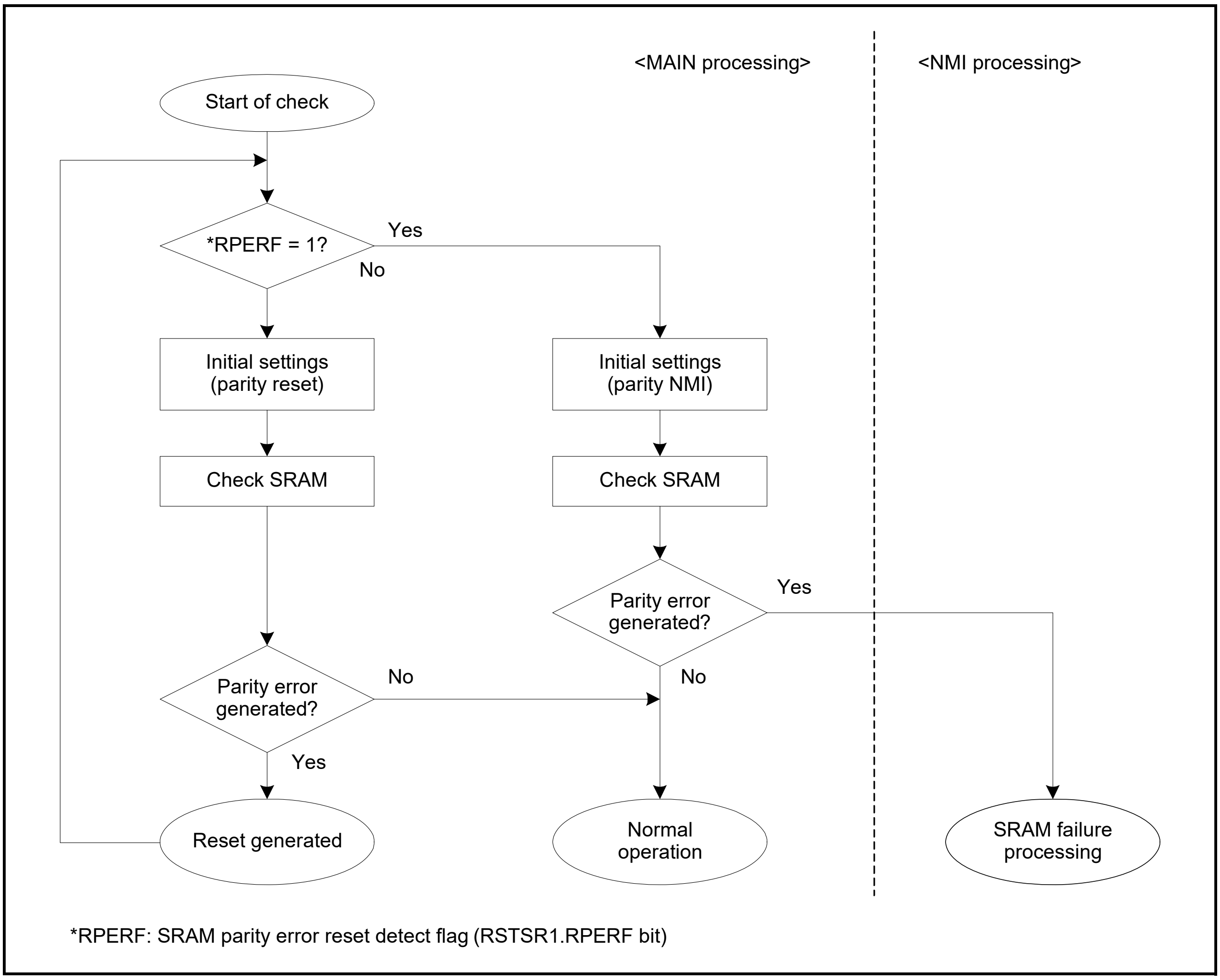
Figure 53.1 ECC Decoder Testing

53.3.5 Parity Calculation Function

The IEC60730 standard requires the checking of SRAM data. A parity bit is therefore added to every 8-bit data unit in the SRAM modules that have 32-bit data width. With this SRAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. When a parity error occurs, a parity-error notification is generated. This function can also be used to trigger a reset. The specification for SRAM0 without ECC, SRAM1, SRAMHS, and Standby SRAM is even parity.

Parity error notification can be specified as either a non-maskable interrupt or reset in the PARIOAD register. When the OAD bit in the PARIOAD register is set to 1, a parity error is output to the reset function. When the OAD bit in the PARIOAD register is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

Parity errors often occur because of noise. To confirm whether the cause is noise or corruption, follow the parity check flows shown in Figure 53.2 and Figure 53.3.



*RPERF: SRAM parity error reset detect flag (RSTSR1.RPERF bit)

Figure 53.2 Flow of SRAM parity check when SRAM parity reset is enabled

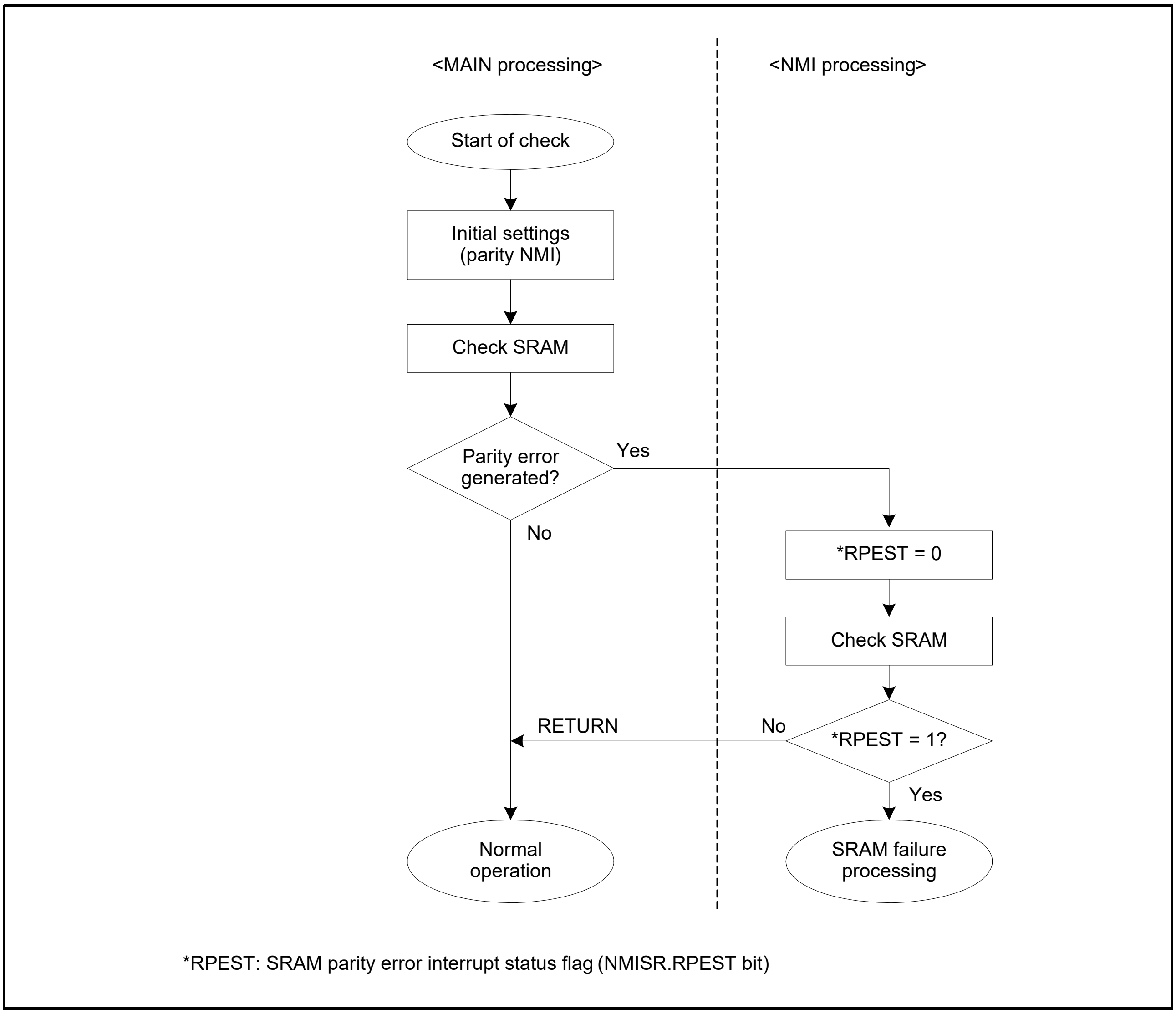


Figure 53.3 Flow of SRAM parity check when SRAM parity interrupt is enabled

53.3.6 SRAM Error Sources

The SRAM error sources are ECC errors and parity errors. ECC errors can be specified as non-maskable interrupts or resets in the OAD bit in the ECCOAD register, and parity errors can be specified as non-maskable interrupts or resets in the PARIOAD register.

Table 53.2 SRAM error sources

Interrupt source	DTC activation	DMAC activation
ECC error (SRAM0 area with ECC)	Not possible	Not possible
Parity error (SRAM0 area without ECC, SRAM1, SRAMHS	Not possible	Not possible

53.3.7 Access Cycles

Table 53.3 SRAMHS (parity area 1FFE 0000h to 1FFF FFFFh)

Read (cycles)		Write (cycles)	
Word access	Halfword/byte access	Word access	Halfword/byte access
2		2	

Table 53.4 SRAM0 (ECC area 2000 0000h to 2000 7FFFh)

Bit settings		Read (cycles)		Write (cycles)	
		Word access	Halfword or byte access	Word access	Halfword or byte access
ECC Off ECCMOD[1] = 0	SRAM0ERDWTEN = 0	2		2	
	SRAM0ERDWTEN = 1	3		2	
ECC On ECCMOD[1] = 1	SRAM0ERDWTEN = 0	2		2	4
	SRAM0ERDWTEN = 1	3		2	4

Table 53.5 SRAM0 (parity area 2000 8000h to 2003 FFFFh)

Bit settings	Read (cycles)		Write (cycles)	
	Word access	Halfword or byte access	Word access	Halfword or byte access
SRAM0WTEN = 0	2		2	
SRAM0WTEN = 1	3		2	

Table 53.6 SRAM1 (parity area 2004 0000h to 2007 FFFFh)

Bit settings	Read (cycles)		Write (cycles)	
	Word access	Halfword or byte access	Word access	Halfword or byte access
SRAM1WTEN = 0	2		2	
SRAM1WTEN = 1	3		2	

53.4 Usage Notes

53.4.1 Wait State Insertion

Set the number of SRAM wait cycles in the SRAMWTSC register based on the following:

- SRAM0, SRAM1
  1 wait: 60 MHz < ICLK ≤ 120 MHz
  No wait: ICLK ≤ 60 MHz

53.4.2 Instruction Fetch from the SRAM Area

When using SRAM0, SRAM1, or the SRAMHS to operate a program, initialize the SRAM area so the CPU can correctly prefetch data. If the CPU prefetches data from an SRAM area that is not initialized, an ECC error or a parity error might occur. Initialize the additional 12-byte area from the end address of the program with the 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

53.4.3 Store Buffer of SRAM

For fast access between SRAM and CPU, a store buffer is used. When a load instruction is executed from the same address after a store instruction to SRAM, the load instruction might read out data from the buffer instead of data from the SRAM. To read data on the SRAM correctly, use either of the following procedures:

- After writing to the SRAM (address = A), use the NOP instruction, then read the SRAM (address = A)
- After writing to the SRAM (address = A), read data from area other than SRAM (address = A), then read the SRAM (address = A).

54. Standby SRAM

54.1 Overview

An on-chip SRAM is provided to retain data in Deep Software Standby mode. Table 54.1 lists the Standby SRAM specifications.

Table 54.1 Standby SRAM specifications

Parameter	Specifications
SRAM capacity	8 KB
SRAM address	200F E000h to 200F FFFFh
Access	The number of access depends on the frequency between ICLK and PCLKB. See section 54.2.4, Access Cycle for details.
Data retention	Data can be retained in Deep Software Standby mode
Parity	Even parity (data: 8 bits, parity: 1 bit)
Module-stop function	Module-stop state can be set to reduce power consumption

54.2 Operation

54.2.1 Data Retention

The power supply to the Standby SRAM in Deep Software Standby mode is enabled by the DPSBYCR.DEEPCUT[1:0] bits. If the DPSBYCR.DEEPCUT[1:0] bits are set to 00b, data in the Standby SRAM is retained in Deep Software Standby mode. See section 11, Low Power Modes, for details on the DPSBYCR.DEEPCUT[1:0] bits.

54.2.2 Low-Power Function

Power consumption can be reduced by setting Module Stop Control Register A (MSTPCRA) to stop supply of the clock signal to the SRAM. Setting the MSTPA7 bit in MSTPCRA to 1 stops supply of the clock signal to the Standby SRAM. The Standby SRAM is then placed in the module-stop state by stopping supply of the clock signals. The Standby SRAM operates after a reset. The Standby SRAM is not accessible if it is in the module-stop state. Do not transition to the module-stop state while access to the Standby SRAM is in progress. For details on the MSTPCRA register, see section 11, Low Power Modes.

54.2.3 Parity Calculation Function

The parity calculation function for Standby SRAM is the same as for SRAM1 or SRAM0 without error correction code (ECC). The function of the OAD bit in the PARIOAD register and the flow of the SRAM parity check are shared by the Standby SRAM modules. For details, see section 53.3.5, Parity Calculation Function and section 53.3.6, SRAM Error Sources.

54.2.4 Access Cycle

Table 54.2 Standby SRAM access cycle

	Read (cycles)		Write (cycles)	
Frequency	Word access	Halfword/byte access	Word access	Halfword/byte access
Same Frequency ICLK = PCLKB	3 ICLK		2 ICLK	
Different Frequency ICLK > PCLKB	1 ICLK + 2 to 3 PCLKB		1 ICLK + 1 to 2 PCLKB	

54.3 Usage Notes

54.3.1 Instruction Fetch from the Standby SRAM area

When using Standby SRAM to operate a program, initialize the Standby SRAM area so that the CPU can correctly prefetch data. A parity error might occur if the CPU prefetches from an area that is not initialized. Initialize the additional 12-byte area from the end address of the program with the 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

55. Flash Memory

55.1 Overview

The MCU provides up to 2-MB code flash memory and 64-KB data flash memory. The Flash Control Unit (FCU) controls programming and erasure of the flash memory. The Flash Application Command Interface (FACI) controls the FCU in accordance with the specified FACI commands.

Table 55.1 lists the specifications of the code flash memory and data flash memory, and Figure 55.1 shows a block diagram of the related modules. Figure 55.2 shows the configuration of the code flash memory, and Figure 55.3 shows the configuration of the data flash memory.

Table 55.1 Specifications of the code flash memory and data flash memory

Parameter	Code flash memory specifications	Data flash memory specifications
Memory capacity	• Up to 2 MB	64 KB
Read cycle	• 80 MHz < ICLK frequency ≤ 120 MHz: Cache hit: 1 cycle Cache miss: 3 cycles • 40 MHz < ICLK frequency ≤ 80 MHz: Cache hit: 1 cycle Cache miss: 2 cycles • ICLK frequency ≤ 40 MHz: Cache hit: 1 cycle Cache miss: 1 cycle	A read operation takes seven cycles of FCLK in words or bytes (FCLK frequency is up to 60 MHz)
Value after erasure	FFh	Undefined
Programming/erasing methods	• Programming and erasing of code and data flash memory handled by FACI commands specified in the FACI command issuing area (407E 0000h) • Programming by dedicated flash-memory programmer transfer through a serial interface (serial programming) • Programming of flash memory by user program (self-programming)	
Security function	Protection against illicit tampering with or reading of data in flash memory	
Protection	Protection against erroneous overwriting of flash memory	
Background operations (BGOs)	• Code flash memory can be read during code flash memory programming*1 • Code flash memory can be read during data flash memory programming • Data flash memory can be read during code flash memory programming	
Units of programming and erasure	• 128-byte units for programming in user area • Block units for erasure in user area	• 4/8/16-byte units for programming in data area • 64/128/256-byte units for erasure in data area
Other functions	Interrupts can be accepted during self-programming	
	An expansion area of flash memory (option bytes) can be set in the initial MCU settings	
On-board programming (four types)	Programming in serial programming mode (SCI boot mode): • Asynchronous serial interface (SCI9) used • Transfer rate adjusted automatically Programming in serial programming mode (USB boot mode): • USBFS used • Dedicated hardware not required, so direct connection to PC is possible Programming in On-chip debug mode: • JTAG or SWD interface used • Dedicated hardware not required Programming by a routine for code and data flash memory programming within the user program: • Allows code and data flash memory programming without resetting the system	

Note 1. Constraints apply to the combinations of ranges in which writing can proceed. See Table 55.11.

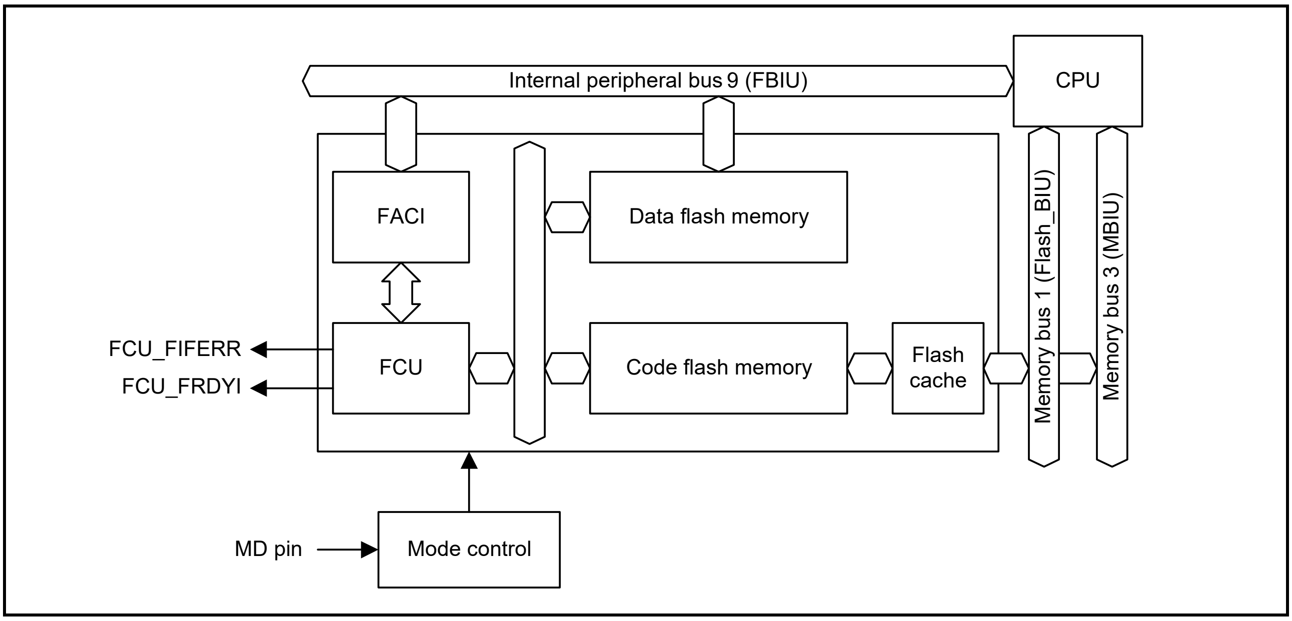


Figure 55.1 Block diagram of flash memory-related modules

55.2 Structure of Memory

Figure 55.2 shows the mapping of the code flash memory, and Table 55.2 shows the read and programming/erasure addresses by product. The user space of the code flash memory is divided into 8- and 32-KB blocks, which serve as the units of erasure. The user area is available for storing the user program.

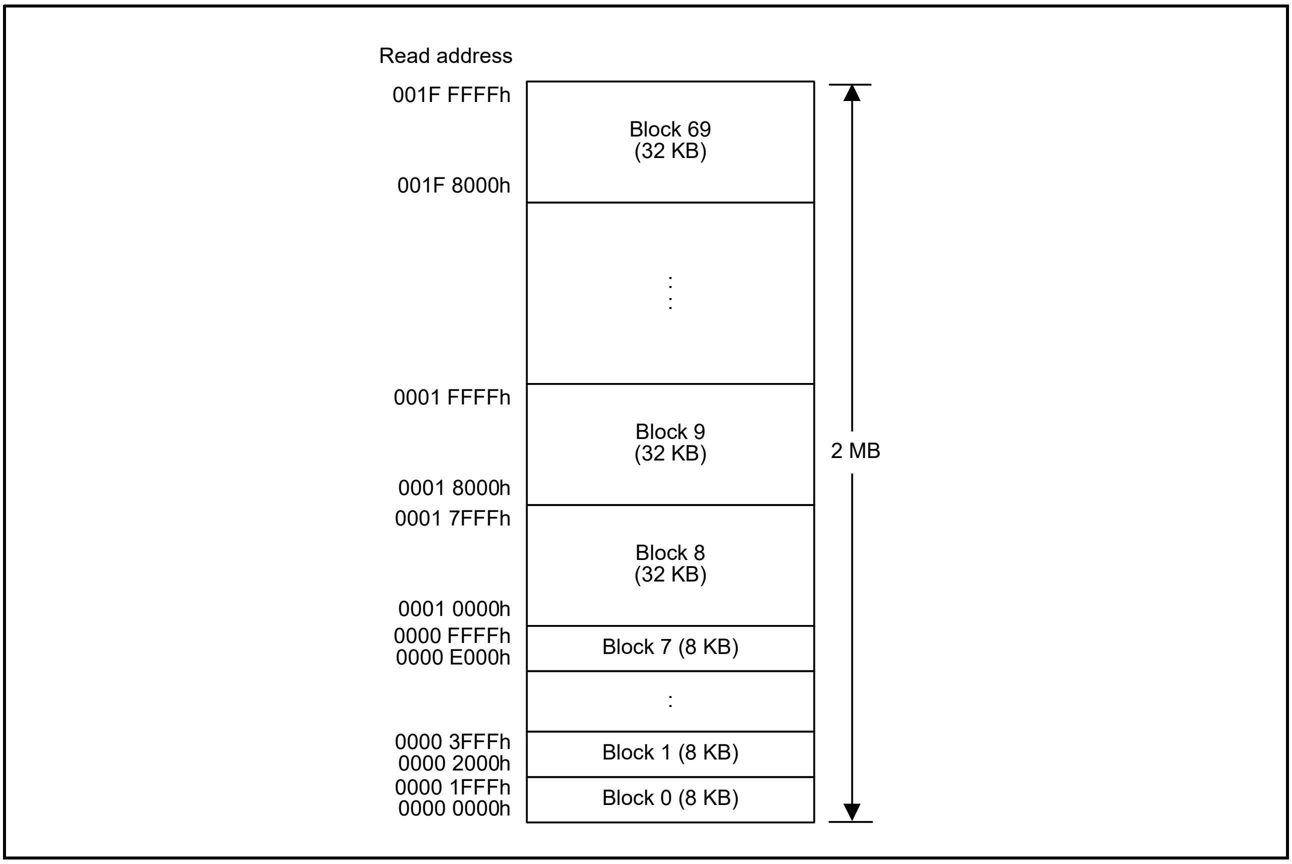


Figure 55.2 Mapping of the code flash memory

Table 55.2 Read and P/E addresses by product for the code flash memory

Product	Read address	P/E address	Number of blocks
2-MB product	0000 0000h to 001F FFFFh	0000 0000h to 001F FFFFh	0 to 69
1-MB product	0000 0000h to 000F FFFFh	0000 0000h to 000F FFFFh	0 to 37

The data area of the data flash memory is divided into 64-byte blocks, with each being a unit for erasure. Figure 55.3 shows the mapping of the data flash memory.

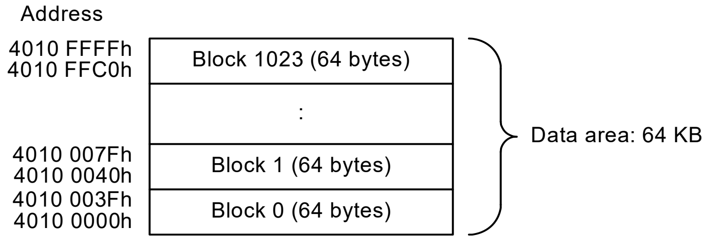


Figure 55.3 Mapping of the data flash memory

55.3 Flash Cache

55.3.1 Overview

The flash cache (FCACHE) speeds up read access from the bus master to the flash memory. The FCACHE includes:

- FCACHE1, for CPU instruction fetches
- FCACHE2, for CPU operand access and DMA
- FLPF, for the prefetch access in CPU instruction fetches.

Table 55.3 Flash cache overview

Parameter	Flash cache 1 (FCACHE1)	Flash cache 2 (FCACHE2)	Prefetch Buffer (FLPF)
Cache target region	0000 0000h - 001F FFFFh	0000 0000h - 001F FFFFh	0000 0000h - 001F FFFFh
Target bus master	CPU instruction fetch	CPU Operand Access and Access from other than CPU	FLPF
Capacity	256 bytes	16 bytes	32 bytes
Associativity	• 8-way set associative • 128 bits/entry (128-bit aligned data) • 2 entries/way	• Fully associative • 128 bits/entry (128-bit aligned data) • 1 entry for FCACHE2	• Fully associative • 128 bits/entry (128-bit aligned data) • 2 entries
Access cycles	• Cache hit: 0 waits • Cache miss: Number of waits set in Flash Wait Cycle Register	• Cache hit: 0 waits • Cache miss: Number of waits set in Flash Wait Cycle Register	• Cache hit: 0 waits • Cache miss: Number of waits set in Flash Wait Cycle Register

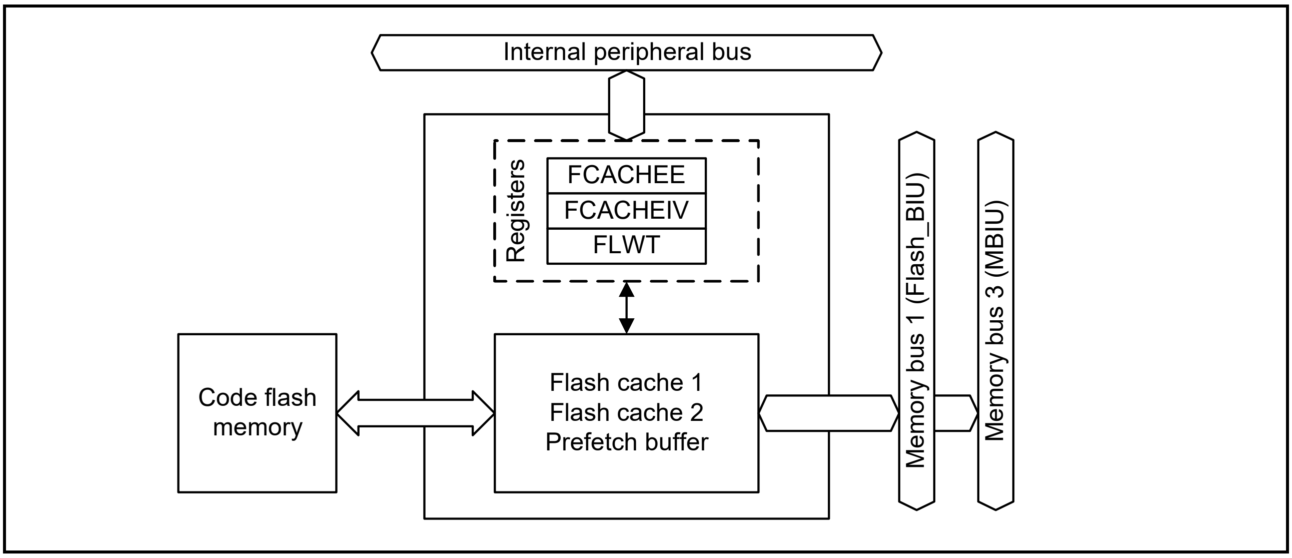


Figure 55.4 **FCACHE block diagram**

55.3.2 Register Descriptions

55.3.2.1 Flash Cache Enable Register (FCACHEE)

Address(es): FCACHE.FCACHEE 4001 C100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCACHEEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FCACHEEN	FCACHE Enable	0: Disable FCACHE 1: Enable FCACHE.	R/W
b15 to b1	—	Reserved	These bits are read as 0.	R

The FCACHEE.FCACHEEN bit enables or disables the flash cache function of FCACHE1, FCACHE2, and FLPF. This bit does not affect FCACHEIV.FCACHEIV. When FCACHE is enabled, the HPROT[3] determines whether it is cacheable or non-cacheable. See section 15.8 for details on HPROT[3].

55.3.2.2 Flash Cache Invalidate Register (FCACHEIV)

Address(es): FCACHE.FCACHEIV 4001 C104h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCACH EIV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FCACHEIV	FCACHE Invalidate	• Reads: 0: Do not invalidate 1: Invalidate. • Writes: When write value is 1, FCACHE is invalidated. When write value is 0, this setting is ignored.	R/W
b15 to b1	—	Reserved	These bits are read as 0.	R

When 1 is written to the FCACHEIV.FCACHEIV bit, the flash cache data in FCACHE1, FCACHE2, and FLPF is invalidated.

55.3.2.3 Flash Wait Cycle Register (FLWT)

Address(es): FCACHE.FLWT 4001 C11Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FLWT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	FLWT[2:0]*1	Flash Wait Cycle	b2 b0 0 0 0: 0 waits (ICLK ≤ 40 MHz) 0 0 1: 1 wait (40 MHz < ICLK ≤ 80 MHz) 0 1 0: 2 waits (80 MHz < ICLK ≤ 120 MHz). Other settings are reserved.	R/W
b15 to b3	—	Reserved	These bits are read as 0.	R

Note 1. Settings other than 000b are prohibited in the SubOSC-speed mode.

The Flash Wait Cycle Register (FLWT) sets the access wait count for the flash memory. For faster clock frequencies, set FLWT.FLWT before changing the clock frequency. For slower clock frequencies, set FLWT.FLWT after changing the clock frequency.

55.4 Operation

Use the FCACHEE register to set up and enable flash operation.

To set up the flash cache and prepare to rewrite the flash memory:

1. Disable the flash cache by resetting FCACHEE.FCACHEEN.*1
2. Invalidate the flash cache by setting FCACHEIV.FCACHEIV.
3. Check that FCACHEIV.FCACHEIV is 0.
4. Enable the flash cache by setting FCACHEE.FCACHEEN.

Note 1. It is not necessary to disable the flash cache on the first setup after reset.

55.4.1 Notice to use Flash Cache

When using Flash cache by access from the CPU, Arm® MPU should also be set to cacheable.

See the ARMv7-M Architecture Reference Manual and the Cortex-M4 Devices Generic User Guide.

55.5 Operating Modes Associated with the Flash Memory

Figure 55.5 shows a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see section 3, Operating Modes.

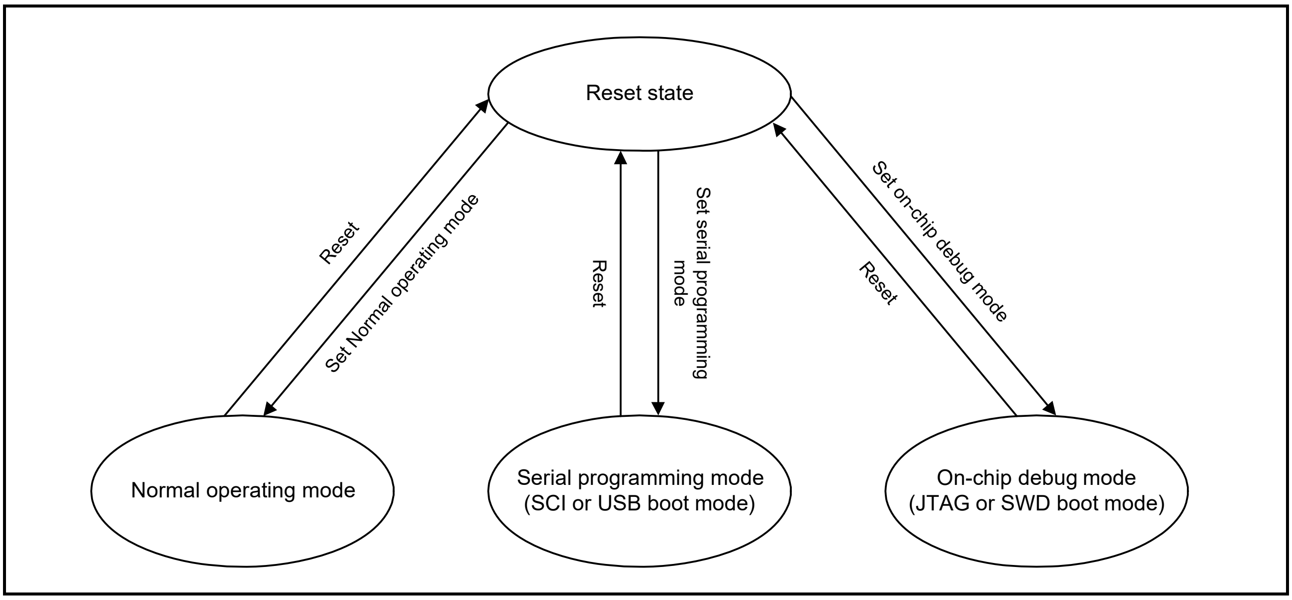


Figure 55.5 Mode transitions associated with flash memory

The flash memory area where programming and erasure are permitted and where the boot program executes after a reset differ with the mode. Table 55.4 shows the differences between the modes.

Table 55.4 Difference between modes

Parameter	Normal operating mode	Serial programming mode (SCI or USB boot mode)	On-chip debug mode (JTAG or SWD boot mode)
Programmable and erasable areas	• Code flash memory • Data flash memory	• Code flash memory • Data flash memory	• Code flash memory • Data flash memory
Erasure in block units	Possible	Possible	Possible
Boot program at a reset	User area program	Embedded program for serial programming	Depends on debug command

55.5.1 ID Code Protection

This function prohibits programming and on-chip debugging. The device validates or invalidates the ID code and determines the ID code based on an ID code stored in the flash memory. When ID code protection is enabled, the ID code sent from the host is compared with the ID code in the flash memory to determine whether they match. Programming and on-chip debugging are enabled only when the two match. The ID code in flash memory consists of four 32-bit words.

ID code bits 127 and 126 determine whether ID code protection is enabled and the method of authentication to use with the host. Table 55.5 shows how the ID code determines the method of authentication.

Table 55.5 Specifications for ID code protection

Operating mode on boot up	ID code	State of protection	Operations on connection with the programmer or on-chip debugger
Serial programming mode (SCI/USB boot mode) On-chip debug mode (JTAG/SWD boot mode)	FFh, …, FFh (All bytes = FFh)	Protection disabled	ID code validation is not performed, the ID code always matches, and connection to the programmer or the on-chip debugger is permitted.
	Bit 127 = 1, Bit 126 = 1, and at least one of the 16 bytes is not FFh	Protection enabled	Matching ID code: Authentication ends and connection to the programmer or on-chip debugger is permitted. Non-matching ID code: Additional transition to the ID code protection waiting state. When the ID code sent from the programmer or the on-chip debugger is "ALeRASE" in ASCII code (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFFh), the content of the user flash (code and data) area and the configuration area are erased. However, forced erasure is not executed when the AWS.FSPR*1 bit is 0.
	Bit 127 = 1 and bit 126 = 0	Protection enabled	Matching ID code: Authentication ends and connection to the programmer or the on-chip debugger is permitted. Non-matching ID code: Additional transition to the ID code protection waiting state.
	Bit 127 = 0	Protection enabled	ID code validation is not performed, the ID code is always non-matching, and connection to the programmer or the on-chip debugger is prohibited.

Note 1. For details on the AWS.FSPR bit, see section 7.2.3, Access Window Setting Register (AWS).

55.6 Overview of Functions

By using a dedicated flash-memory programmer to program the on-chip flash memory through a serial interface (serial programming) or JTAG/SWD interface (on-chip debug mode), the device can be programmed before or after it is mounted on the target system.

Additionally, security functions to prohibit overwriting of the user program written to the on-chip flash memory are incorporated to prevent tampering by third parties.

Programming by the user program (self-programming) is available for applications that might require updating after system manufacturing or shipment. Protection features for safely overwriting the flash memory area are also provided. Additionally, interrupt processing during self-programming is supported so programming can proceed while processing external communications and other functions. Table 55.6 lists the programming methods and the corresponding operating modes.

Table 55.6 Programming methods (1 of 2)

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer through the SCI or USBFS interface enables on-board programming of the flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash-memory programmer through the SCI or USBFS interface and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	

Table 55.6 Programming methods (2 of 2)

Programming method	Functional overview	Operating mode
Self-programming	A user program written to memory in advance of serial programming execution can also program the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data from the code flash memory while the data flash memory is programmed. As a result, a program resident in the code flash memory can program the data flash memory. Background operation can also be used for reading from and writing to the code flash memory by itself, but only when the address ranges of the code flash memory that are the targets for programming and reading satisfy particular conditions (see Table 55.11). When those conditions are met, a program resident in one half of the code flash memory can be executed to program the other half of the code flash memory. For background operations that are not possible, instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being programmed by self-programming. In such cases, a program for programming from the internal SRAM or external memory must be transferred in advance and executed.	Normal operating mode
JTAG or SWD programming	A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD enables on-board programming of the flash memory after the device is mounted on the target system.	On-chip debug mode
	A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	

Table 55.7 lists the functions of the on-chip flash memory. The functions in serial programming are realized by serial programmer commands, while the functions in self-programming are realized by reading of the on-chip flash memory by a FACI command or the user program.

Table 55.7 Basic functions

		Availability	
Function	**Functional overview**	**Serial programming**	**Self-programming**
Blank check	Checks a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing is written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	Not supported	Supported (data flash programming only)
Block erasure	Erases the memory contents in the specified block.	Supported	Supported
Programming	Writes to the specified address.	Supported	Supported
Read	Reads data programmed in the flash memory.	Supported	Not supported (read by user program is possible)
ID code check	Compares the ID code sent by the host with the code stored in the ROM, and if the two match, the FCU enters the wait state for programming and erasure commands from the host.	Supported	Not supported (ID authentication is not performed)
Security configuration	Configures the security function for serial programming.	Supported with conditions (only switching from enabled to disabled configuration is possible)	Supported with conditions (Only switching from enabled to disabled is possible)
Protection configuration	Configures the access window for flash area protection in the code flash memory.	Supported	Supported

The on-chip flash memory supports the ID code security function. Authentication of ID codes is a security function for use with serial programming and with JTAG or SWD programming. Table 55.8 lists the security functions supported by the on-chip flash memory, and Table 55.9 lists available operations and security settings.

Table 55.8 Security functions

Function	Description
ID authentication	Result of ID authentication can be used to control the connection of a serial programmer for serial programming.

Table 55.9 Available operations and security settings

Function	All security settings and erasure, programming, and read operations		Cautions regarding the security setting configuration
	Serial programming and on-chip debug mode	Self-programming mode	Self-programming mode
ID authentication	When ID codes do not match: • Block erasure commands: Not supported • Programming commands: Not supported • Read commands: Not supported • Security configuration commands: Not supported • Protection configuration commands: Not supported When ID codes match: • Block erasure commands: Supported • Programming commands: Supported • Read commands: Supported • Security configuration commands: Supported • Protection configuration commands: Supported	(ID authentication is not performed.) • Blank check: Supported • Block erasure: Supported • Programming: Supported • Security configuration: Supported • Protection configuration: Supported	ID authentication is not performed.

55.6.1 Configuration Area Bit Map

Figure 55.6 shows the configuration area bit map. The boot program must use these bits as hexadecimal data.

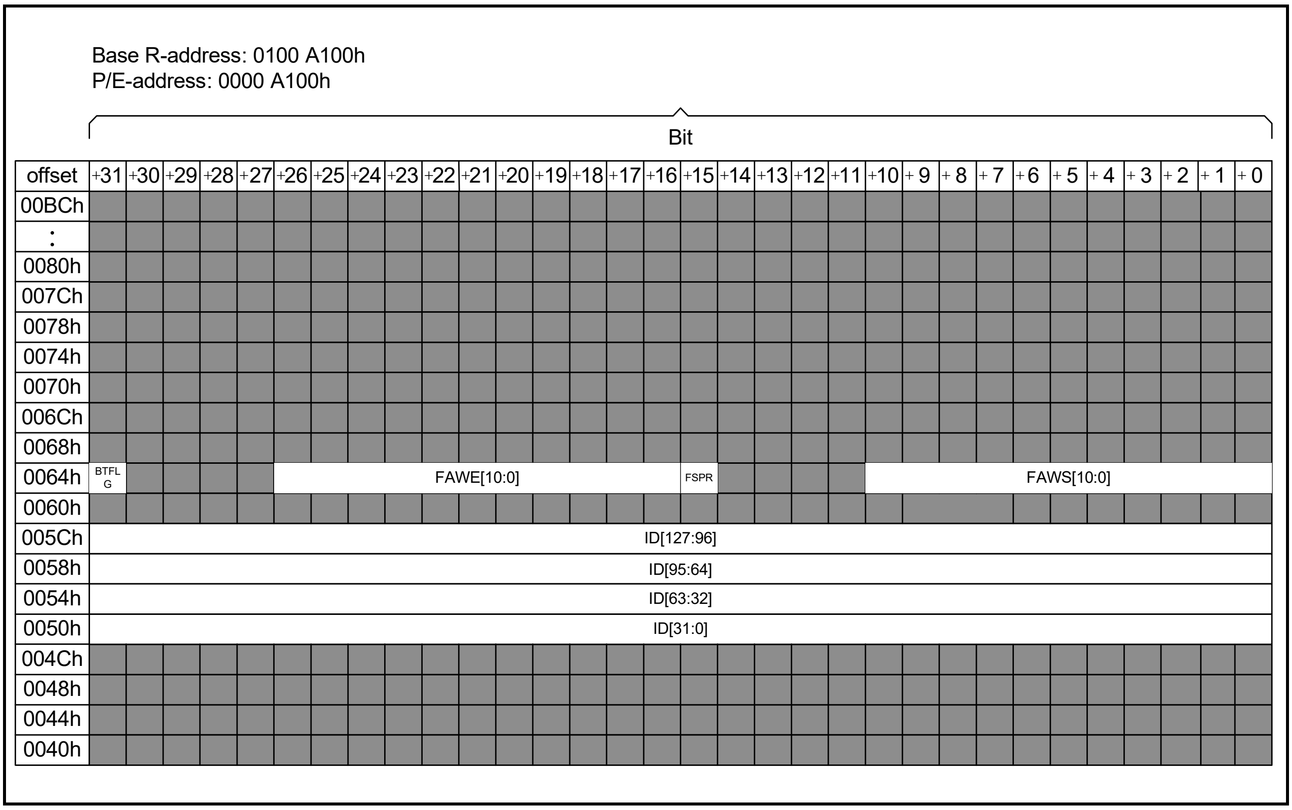
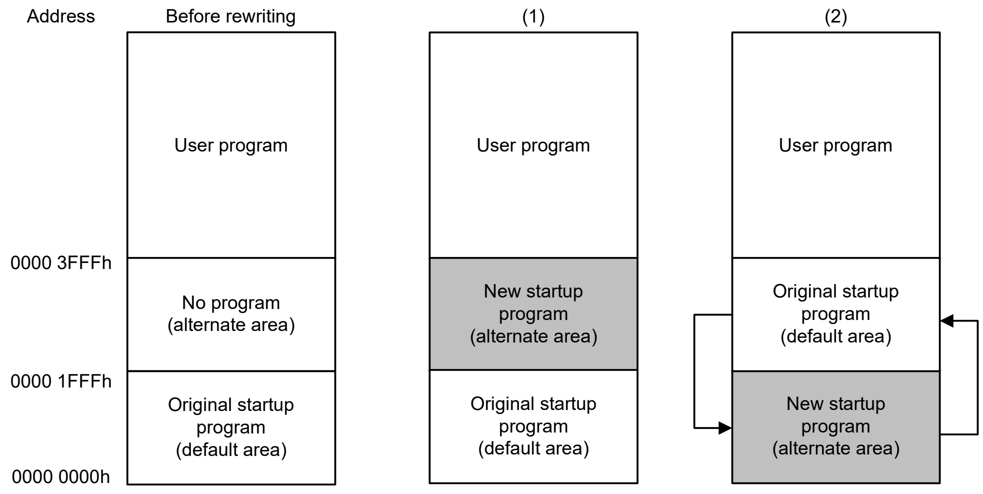


Figure 55.6 Configuration area bit map

55.6.2 Startup Area Select

The startup area select function allows the boot program to be safely updated. The size of the startup area is 8 KB, and the startup area is located in the user area. FACI controls the address of the startup area based on the startup area select flag (AWS.BTFLG) that is located in the configuration area. The startup area can be locked by the AWS.FSPR[*1] bit.

Note 1. For the AWS.FSPR bit, see section 7.2.3, Access Window Setting Register (AWS).



(1) Program a new startup program in the alternate area. If the alternate area fails to be rewritten, the new startup program can be rewritten again after starting up using the default area, because the original startup program is in the default area.
(2) After the alternate area is successfully rewritten, the default and alternate areas are switched using the self-programming library. After that, the program in the alternate area starts after a reset.

Figure 55.7 Overview of startup program protection

55.6.3 Protection by Access Window

Issuing the program or block erase command to a flash memory area outside of the access window results in the command-locked state. The access window is only valid in the user area of the code flash memory. The access window provides protection in self-programming mode, serial programming mode, and on-chip debug mode.

The access window is specified in both the AWS.FAWS[10:0] and AWS.FAWE[10:0][*1] bits. The following describes how to set the FAWS and FAWE bits in different conditions:

- FAWE = FAWS: The P/E command can execute anywhere in the user area of the code flash memory
- FAWE > FAWS: The P/E command can only execute in the window from the block pointed to by the FAWS bits to one block lower than the one pointed to by the FAWE bits
- FSWE < FAWS: The P/E command cannot execute anywhere in the user area of the code flash memory.

Note 1. For information on the AWS.FAWS and AWS.FAWE bits, see section 7.2.4, OCD/Serial Programmer ID Setting Register (OSIS).

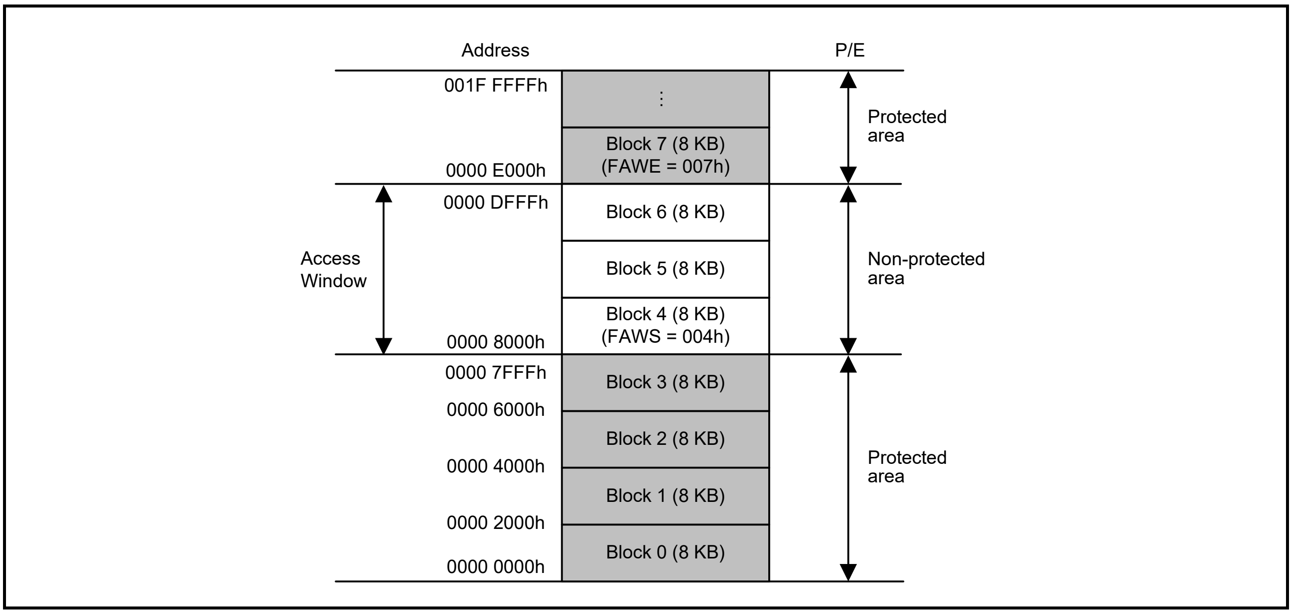


Figure 55.8 Start block address (FAWS) and end block address (FAWE) of access window when the access window only includes the 8-KB block size

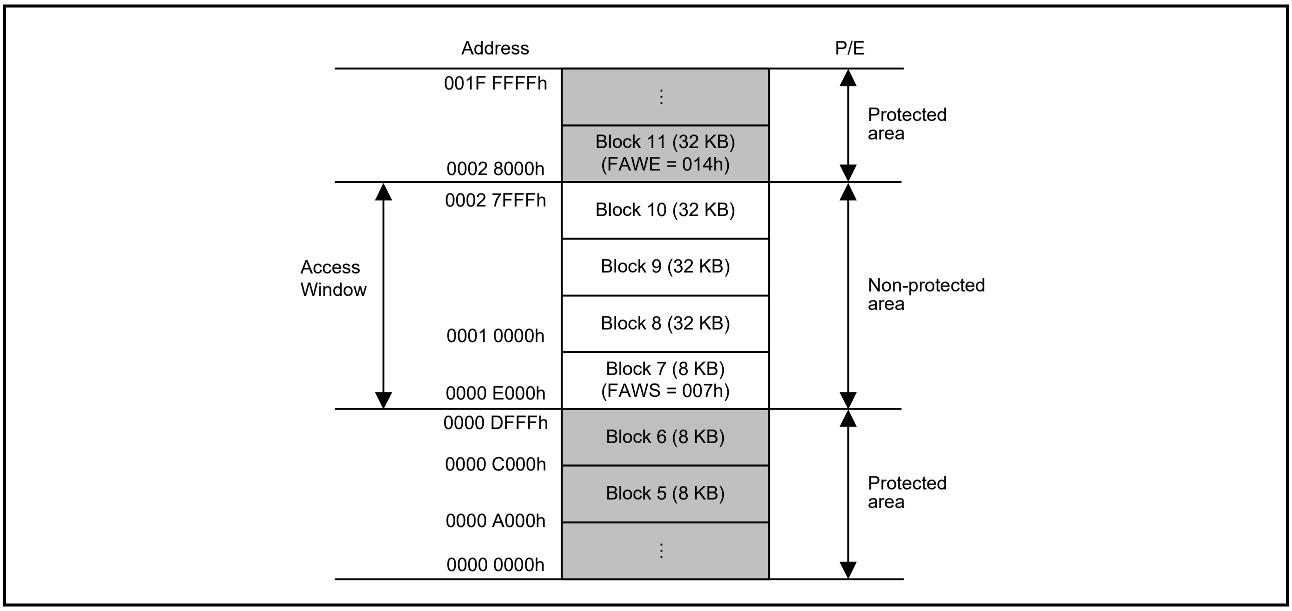


Figure 55.9 Start block address (FAWS) and end block address (FAWE) of access window when the access window includes 8-KB and 32-KB block sizes

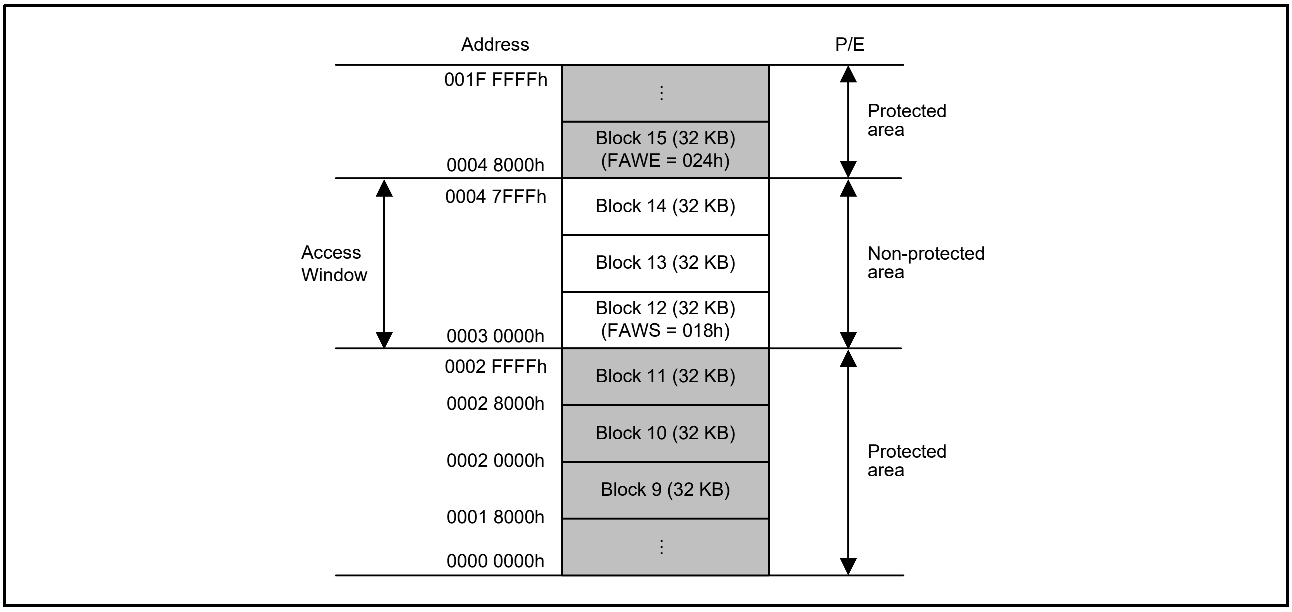


Figure 55.10 Start block address (FAWS) and end block address (FAWE) of access window when the access window only includes the 32-KB block size

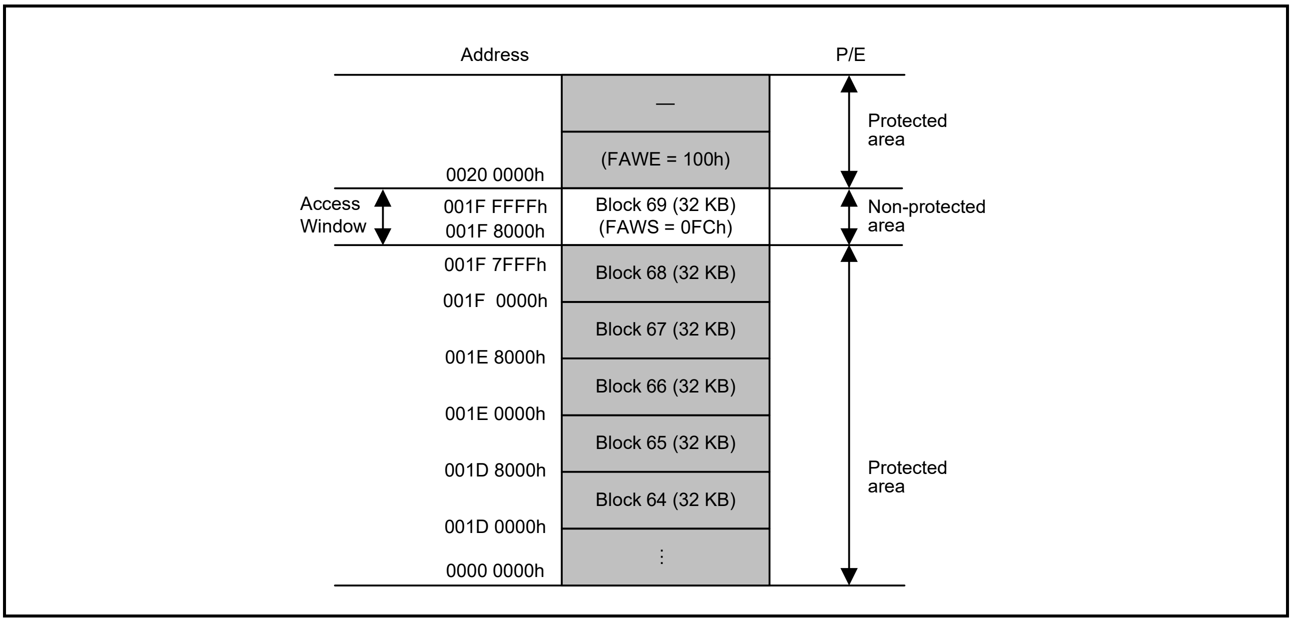


Figure 55.11 Start block address (FAWS) and end block address (FAWE) of access window when the access window only includes the final block

55.7 Programming Commands

The FACI controls the FCU in accordance with the specified FACI commands.

55.8 Suspend Operation

Reading from the code or data flash memory is not possible during programming or erasure when the address ranges do not satisfy the conditions for background operation. When a P/E suspend command is issued to suspend the programming or erasure of the code or data flash memory, reading from the memory is enabled. One suspend command mode is available for programming and two suspend command modes are available for erasure (suspension priority mode and erasure priority mode). The P/E resume command is available for resuming suspended programming or erasure.

55.9 Protection

Provided types of protection include:

- Software protection
- Error protection
- Boot program protection.

55.10 Serial Programming Mode

The serial programming modes include:

- Boot mode with SCI9
- USB boot mode with the USBFS.

Table 55.10 lists the I/O pins for the flash memory-related modules.

Table 55.10 I/O pins for flash memory-related modules

Pin name	I/O	Applicable modes	Function
MD	Input	SCI boot mode USB boot mode (Serial programming mode)	Selection of operating mode
P110/RXD9	Input	SCI boot mode	For host communication, to receive data through SCI
P109/TXD9	Output		For host communication, to transmit data through SCI
USB_DP, USB_DM	I/O	USB boot mode	USB data I/O
USB_VBUS	Input		Detection of connection and disconnection of USB cables

55.10.1 SCI Boot Mode

In boot mode, the host sends control commands and data for programming, and the code and data flash memory area are programmed or erased accordingly. An on-chip SCI handles transfer between the host and the MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When the MCU is activated in boot mode, the embedded program for serial programming is executed. This program automatically adjusts the bit rate of the SCI and controls programming and erasure by receiving control commands from the host. The USB cable must not be connected on reset release.

Figure 55.12 shows the system configuration for operations in boot mode.

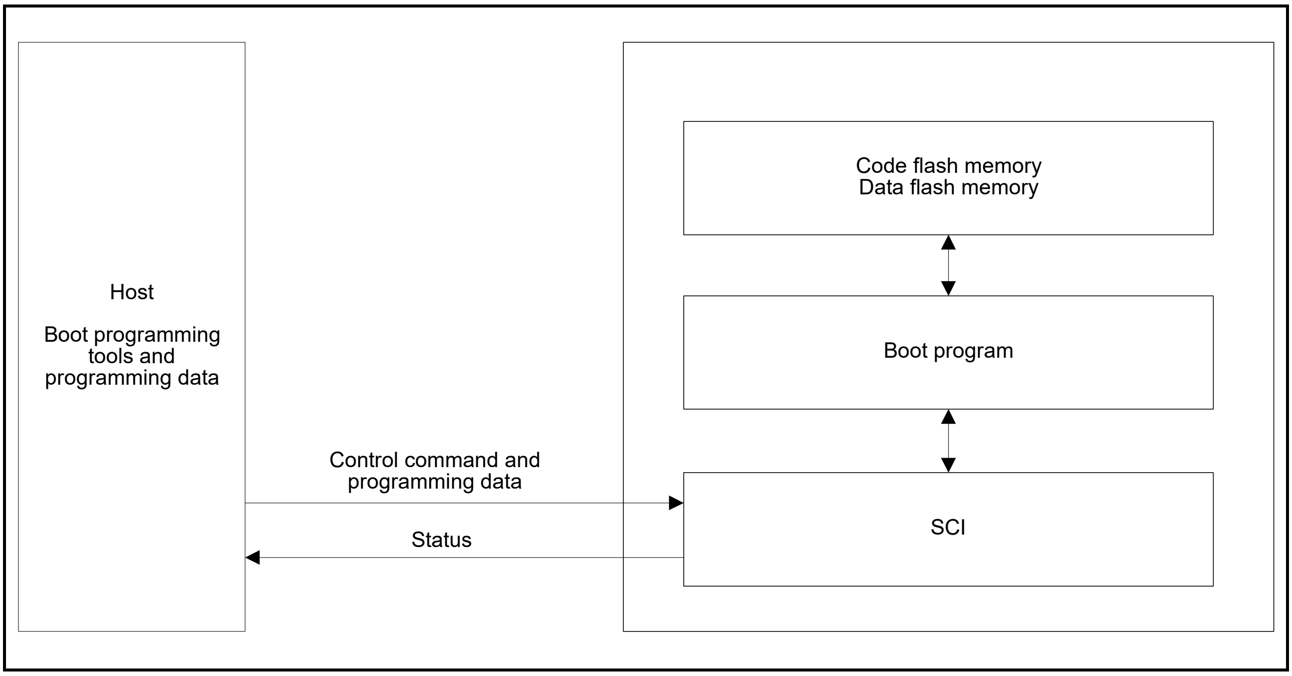


Figure 55.12 System configuration in SCI boot mode

55.10.2 USB Boot Mode

In USB boot mode, the code and data flash memory are programmed or erased by control commands and data for programming transmitted from an externally connected host through the USB interface.

Using USB boot mode requires preparation on the host side of the tools for transmitting control commands and data for programming, and of the data. Figure 55.13 shows the configuration of a system for use in USB boot mode. The USB cable must be connected on reset release.

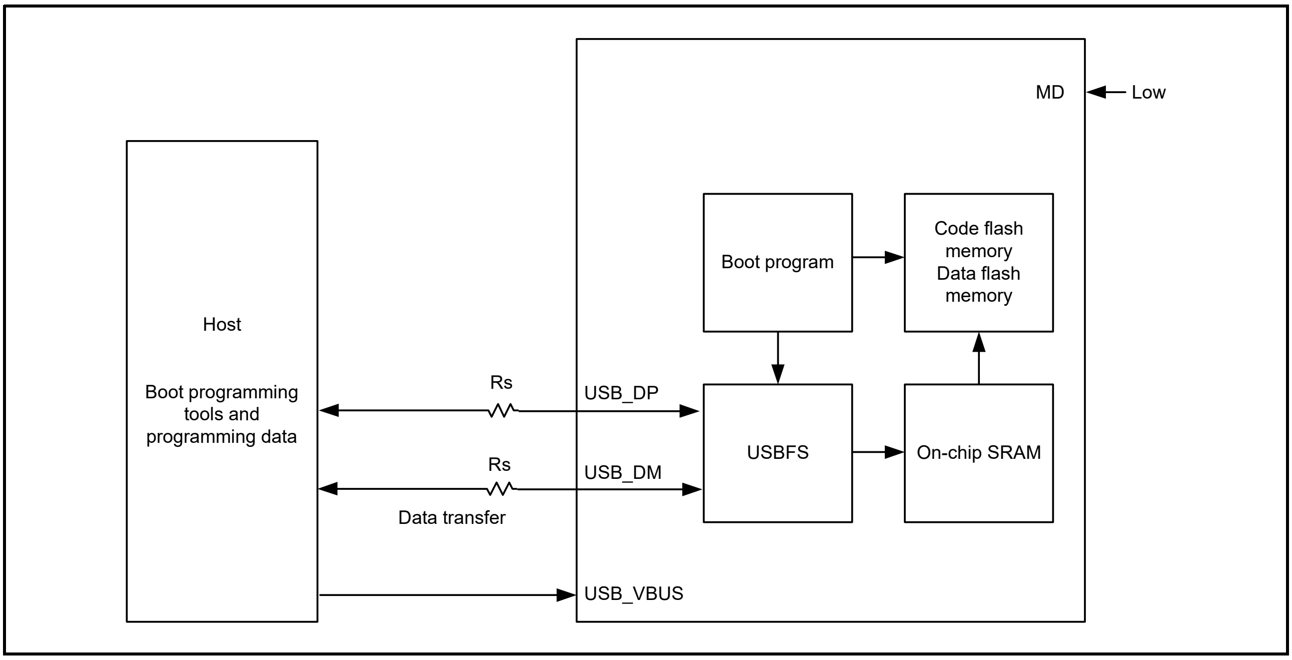


Figure 55.13 System configuration in USB boot mode

55.11 Using a Serial Programmer for Programming

A dedicated flash memory programmer can be used to program the flash memory in serial programming mode.

55.11.1 Serial Programming

The MCU is mounted on the system board for serial programming. A connector to the board allows programming by the flash memory programmer to proceed.

55.11.2 Programming Environments

Figure 55.14 shows the environments that Renesas recommends for programming the flash memory of the MCU with data.

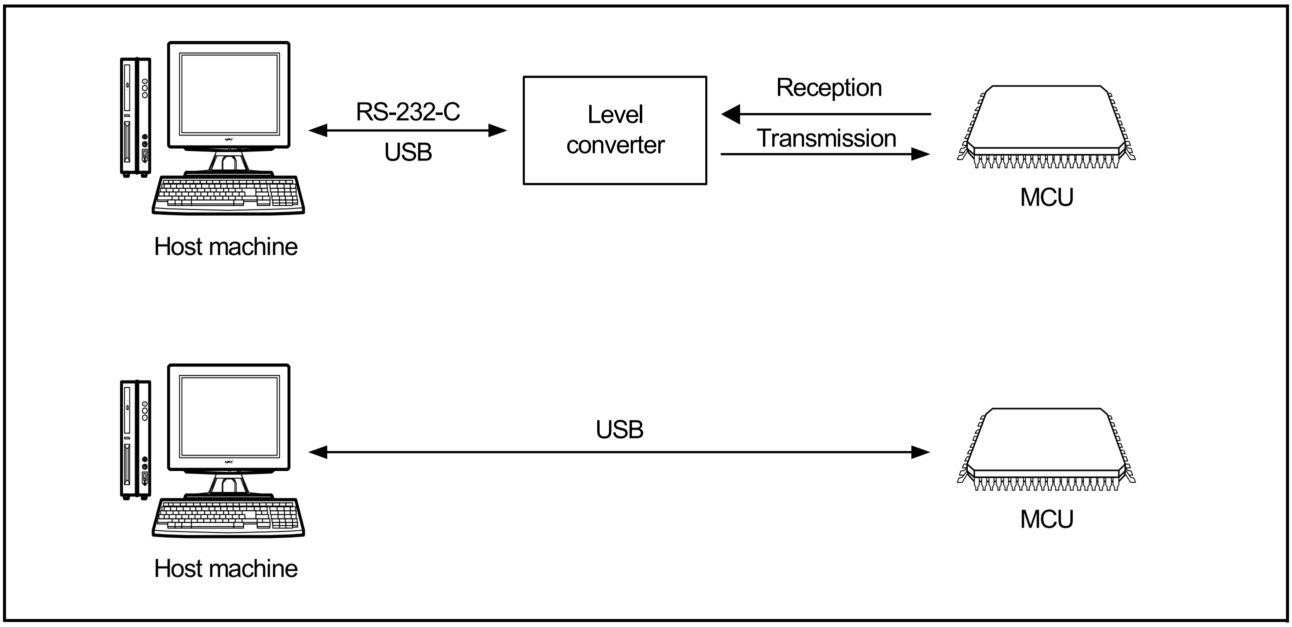


Figure 55.14 Environments for writing programs to the flash memory

55.12 Programming through Self-Programming

55.12.1 Overview

The MCU supports programming of the flash memory by the user program itself. The programming commands can be used with user programs for writing to both the code and data flash memory. This enables upgrading of user programs and overwriting of constant data fields.

For data flash memory programming, the background operation facility makes it possible to program the memory from a programming program in the code flash memory. This programming program can also be copied in advance to and executed from the internal SRAM or external memory.

For code flash memory programming, background operation is available for use when the address ranges of the code flash memory area to be programmed and the code flash memory area to be read satisfy particular conditions (see Table 55.11). For self-programming, a programming program in one half of the code flash memory can be used to program the other half of the code flash memory. This programming program can also be copied in advance to and executed from the internal SRAM or external memory. This is useful when the address ranges do not satisfy the conditions for background operation.

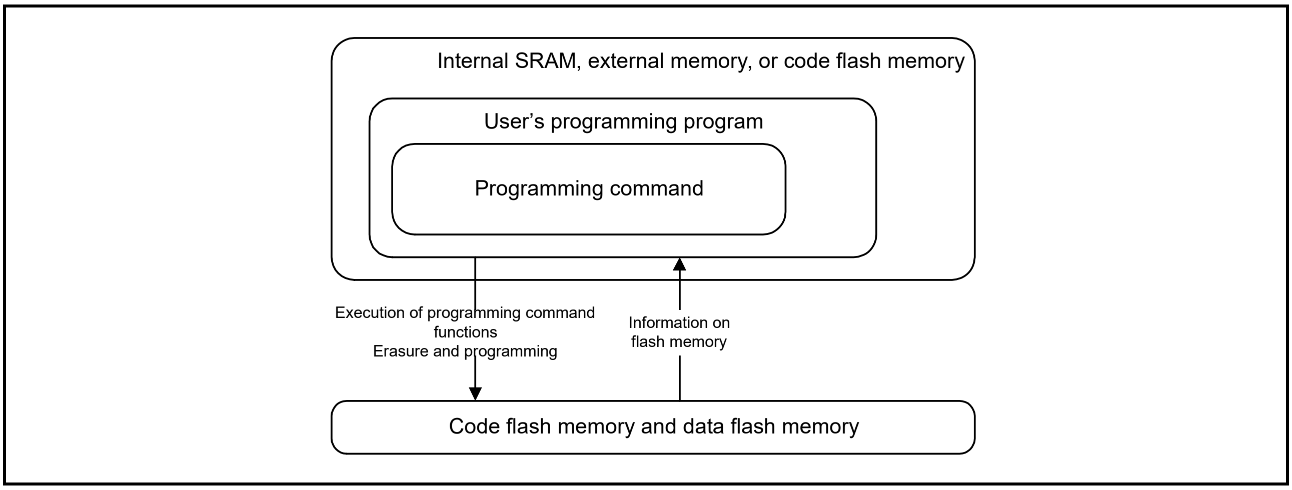


Figure 55.15 Schematic view of self-programming

55.12.2 Background Operation

Background operation can be used when a combination of the flash memory for writing and reading is any of those listed in Table 55.11.

Table 55.11 Conditions under which background operation is available

Product	Range for writing	Range for reading
All products	Data flash memory	Code flash memory
	Code flash memory	Data flash memory
2-MB products	First half (1 MB) of the user area of the code flash memory (addresses 0000 0000h to 000F FFFFh)	Second half (1 MB) of the user area of the code flash memory (addresses 0010 0000h to 001F FFFFh)
	Second half (1 MB) of the user area of the code flash memory (addresses 0010 0000h to 001F FFFFh)	First half (1 MB) of the user area of the code flash memory (addresses 0000 0000h to 000F FFFFh)
1-MB products	First half (0.5 MB) of the user area of the code flash memory (addresses 0000 0000h to 0007 FFFFh)	Second half (0.5 MB) of the user area of the code flash memory (addresses 0008 0000h to 000F FFFFh)
	Second half (0.5 MB) of the user area of the code flash memory (addresses 0008 0000h to 000F FFFFh)	First half (0.5 MB) of the user area of the code flash memory (addresses 0000 0000h to 0007 FFFFh)

55.13 Reading the Flash Memory

55.13.1 Reading the Code Flash Memory

No special settings are required to read the code flash memory in Normal mode. Data can be read through access to addresses in the code flash memory. Values read from code flash memory that was erased but not yet reprogrammed, such as code flash memory in the non-programmed state, are all read as 1s.

55.13.2 Reading the Data Flash Memory

No special settings are required to read the data flash memory in Normal mode. Data can be read through access to addresses in the data flash memory. Values read from data flash memory that was erased but not yet reprogrammed again, such as data flash memory in the non-programmed state, are undefined. Use blank checking to confirm that an area is in the non-programmed state.

55.14 Usage Notes

55.14.1 Reading Areas Where Programming or Erasure Was Interrupted

When programming or erasure of an area of flash memory is interrupted, the data stored in the area become undefined. To avoid reading undefined data, which can be a source of faulty operation, do not fetch instructions or read data from areas where programming or erasure was interrupted.

55.14.2 Constraint on Additional Writes

Other than the configuration area, no other area can be written to twice. After a write to a flash memory area is complete, erase the area before attempting to overwrite data in that area. The configuration area can be overwritten.

55.14.3 Resets during Programming and Erasure

After a reset triggered by a signal assertion on the RES pin during programming and erasure, wait for tRESW until the operating voltage is within the range stipulated in the electrical characteristics before releasing the device from the reset state. For details on tRESW, see section 60.3.3, Reset Timing.

55.14.4 Allocation of Vectors for Interrupts and Other Exceptions during Programming and Erasure

Generation of an interrupt or other exception during programming or erasure might lead to fetching of the vector from the code flash memory. If the vector allocation does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory.

55.14.5 Constraints during Programming and Erasure

During programming and erasure, do not:

- Permit the operating voltage from the power supply to go beyond the allowed range
- Change the frequency of the peripheral clock.

55.14.6 Abnormal Termination of Programming and Erasure

When programming or erasure ends abnormally because of the generation of a reset by the RES pin, the programming or erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming or erasure ends abnormally, the blank check function cannot determine whether the area was erased successfully. Erase the area again to ensure that the corresponding area is completely erased before use.

56. 2D Drawing Engine (DRW)

56.1 Overview

The 2D Drawing Engine (DRW) provides flexible functions that can support almost any object geometry, rather than being bound to only a few specific geometries such as lines, triangles, or circles. The edges of every object can be independently blurred or anti-aliased.

Rasterization is executed on the bounding box of the object from left to right and top to bottom, and performed one pixel per clock. The 2D Drawing Engine can also raster from bottom to top in certain cases, to optimize the performance. In addition, optimization methods are provided to avoid rasterization of many empty pixels of the bounding box.

The distances to the edges of the object are calculated by a set of edge equations for every pixel of the bounding box. These edge equations can be combined to describe the entire object. If a pixel is inside the object, it is selected for rendering, and if it is outside it is discarded. If it is on the edge, an alpha value can be chosen proportional to the distance of the pixel to the nearest edge for anti-aliasing. Every pixel that is selected for rendering can be textured. The resulting aRGB quadruple can be modified by a general raster operation approach independently for each of the four channels. The aRGB quadruples can then be blended with one of the multiple blend modes of the module.

The 2D Drawing Engine provides two inputs (texture read and framebuffer read) and one output (framebuffer write). The internal color format is always aRGB (8888). The color formats from the inputs are converted to the internal format on read and converted back on write.

Figure 56.1 shows examples of objects that can be drawn in hardware with the 2D Drawing Engine, Figure 56.2 shows a simplified rendering pipeline setup, and Figure 56.3 shows a block diagram of the module.

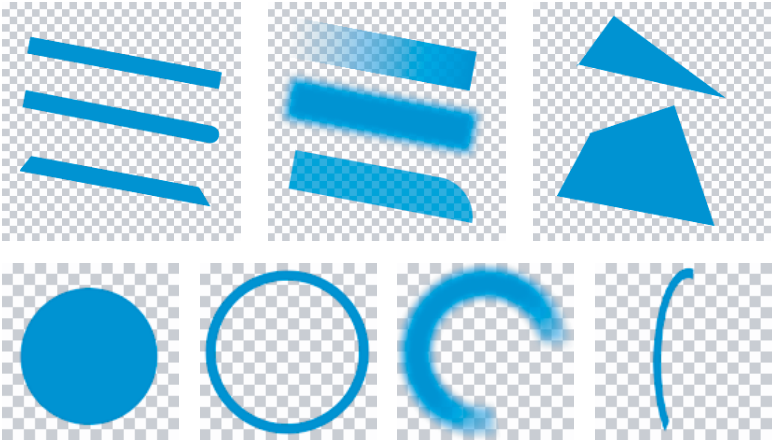


Figure 56.1 Examples of drawing objects

The 2D Drawing Engine also supports a display list mode, which makes it possible to decouple the CPU and graphics controller efficiently and perform rendering in parallel with other CPU activities.

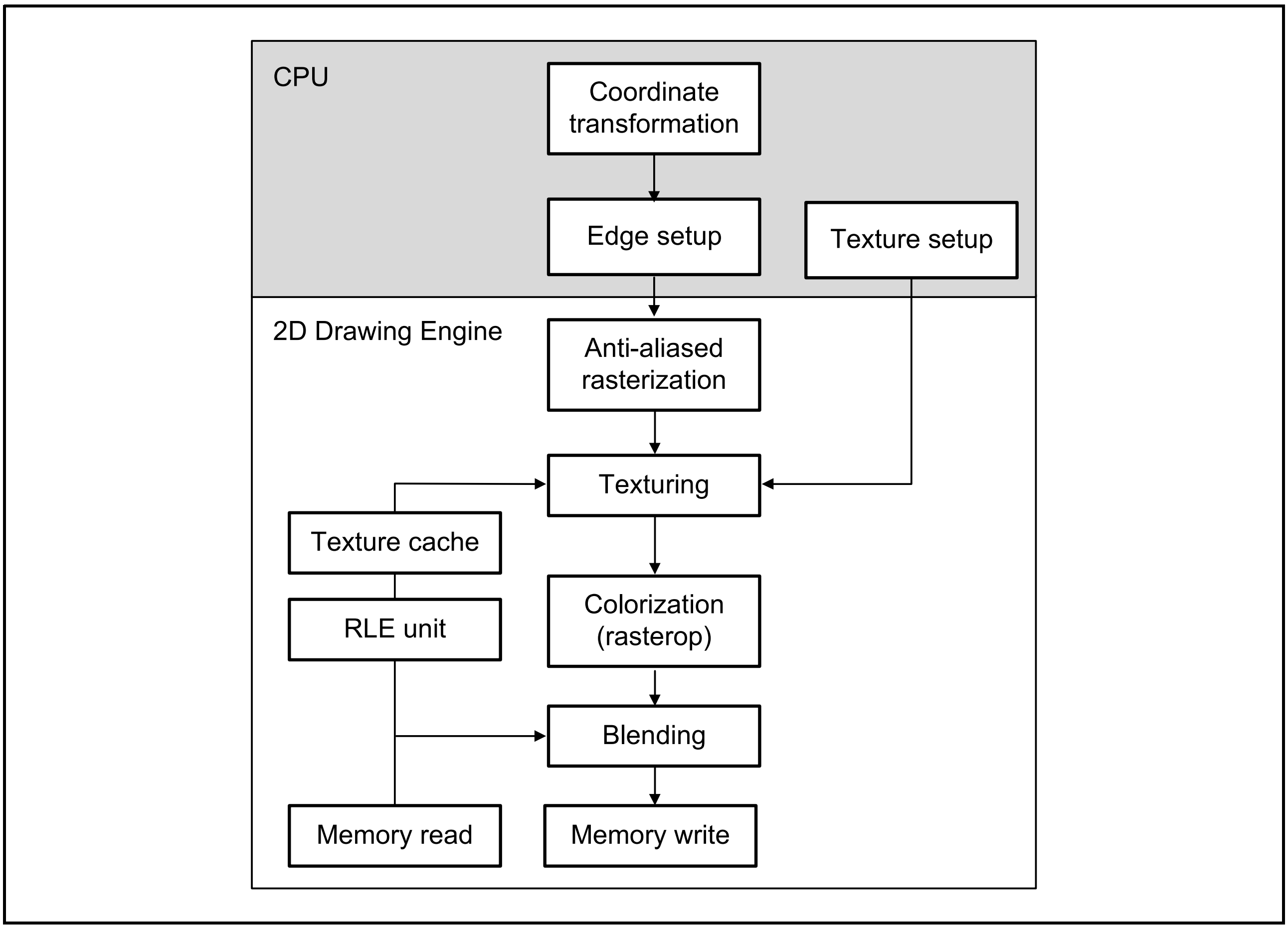


Figure 56.2 Simplified rendering pipeline setup

Note: When using the DRW, set the clock to ICLK = PCLKA.

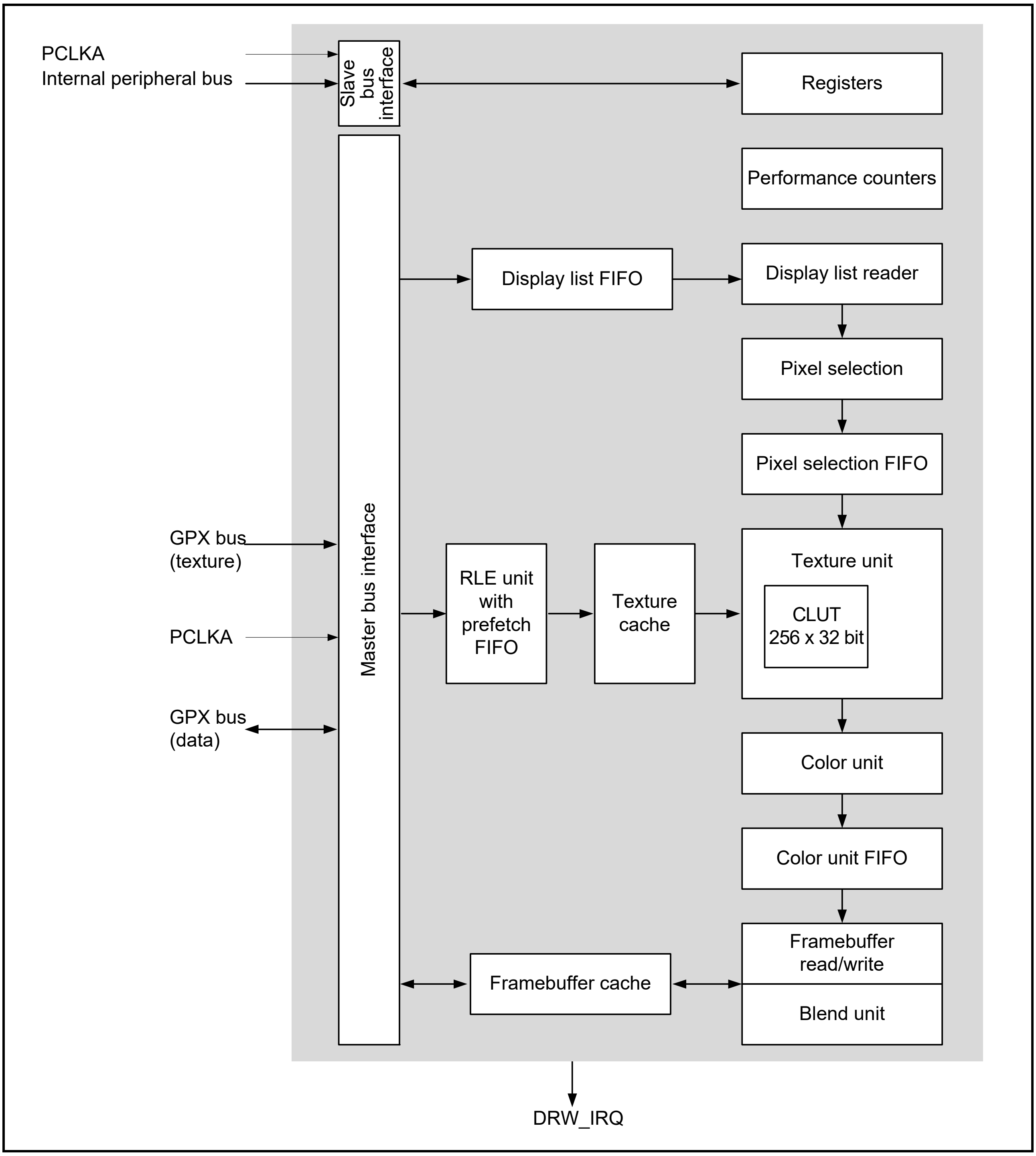


Figure 56.3 Block diagram of the 2D Drawing Engine

The 2D Drawing Engine accesses the GPX bus as bus master through separate caches for:

- Reading and writing pixel data from and to the framebuffer
- Reading textures
- Reading display lists.

The control registers are accessed through the internal peripheral bus interface.

56.2 Register Descriptions

56.2.1 Geometry Control Register (CONTROL)

Address(es): DRW.CONTROL 400E 4000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SPAN STORE	SPAN ABORT	UNION CD	UNION AB	UNION 56	UNION 34	UNION 12	BAND2 ENABL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BAND1 ENABL	LIM6 THRES	LIM5 THRES	LIM4 THRES	LIM3 THRES	LIM2 THRES	LIM1 THRES	QUAD3 ENABL	QUAD2 ENABL	QUAD1 ENABL	LIM6 ENABL	LIM5 ENABL	LIM4 ENABL	LIM3 ENABL	LIM2 ENABL	LIM1 ENABL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	LIM1 ENABLE	Enable Limiter 1	0: Disable 1: Enable.	W
b1	LIM2 ENABLE	Enable Limiter 2	0: Disable 1: Enable.	W
b2	LIM3 ENABLE	Enable Limiter 3	0: Disable 1: Enable.	W
b3	LIM4 ENABLE	Enable Limiter 4	0: Disable 1: Enable.	W
b4	LIM5 ENABLE	Enable Limiter 5	0: Disable 1: Enable.	W
b5	LIM6 ENABLE	Enable Limiter 6	0: Disable 1: Enable.	W
b6	QUAD1 ENABLE	Enable Quadratic Coupling of Limiters 1 and 2	0: Disable 1: Enable.	W
b7	QUAD2 ENABLE	Enable Quadratic Coupling of Limiters 3 and 4	0: Disable 1: Enable.	W
b8	QUAD3 ENABLE	Enable Quadratic Coupling of Limiters 5 and 6	0: Disable 1: Enable.	W
b9	LIM1 THRESHOLD	Enable Limiter 1 Threshold Mode	0: Disable 1: Enable.	W
b10	LIM2 THRESHOLD	Enable Limiter 2 Threshold Mode	0: Disable 1: Enable.	W
b11	LIM3 THRESHOLD	Enable Limiter 3 Threshold Mode	0: Disable 1: Enable.	W
b12	LIM4 THRESHOLD	Enable Limiter 4 Threshold Mode	0: Disable 1: Enable.	W
b13	LIM5 THRESHOLD	Enable Limiter 5 Threshold Mode	0: Disable 1: Enable.	W
b14	LIM6 THRESHOLD	Enable Limiter 6 Threshold Mode	0: Disable 1: Enable.	W
b15	BAND1 ENABLE	Enable Band Post Process for Limiter 1	0: Disable 1: Enable. (See LnBAND.)	W
b16	BAND2 ENABLE	Enable Band Post Process for Limiter 2	0: Disable 1: Enable. (See LnBAND.)	W
b17	UNION12	Combine Limiter 1 and 2 as Union	0: Select minimum/intersect between limiters 1 and 2 1: Select maximum/union between limiters 1 and 2. (Output is called A.)	W

Bit	Symbol	Bit name	Description	R/W
b18	UNION34	Combine Limiter 3 & 4 as Union	0: Select minimum/intersect between limiters 3 and 4 1: Select maximum/union between limiters 3 and 4. (Output is called B.)	W
b19	UNION56	Combine Limiter 5 & 6 as Union	0: Select minimum/intersect between limiters 5 and 6 1: Select maximum/union between limiters 5 and 6. (Output is called D.)	W
b20	UNIONAB	Combine Outputs A & B as union	0: Select minimum/intersect between limiters A and B 1: Select maximum/union between limiters A and B. (Output is called C.)	W
b21	UNIONCD	Combine Outputs C & D as Union	0: Select minimum/intersect between limiters C and D 1: Select maximum/union between limiters C and D. (Output is final.)	W
b22	SPANABORT	Spanabort	0: Disable 1: Enable. Shape is horizontally convex, only a single span per scan line. See Spanabortin 56.6.2.6.	W
b23	SPANSTORE	Spanstore	0: Disable 1: Enable. Next line span start is always equal to or left of current-line span start. See Spanstore in 56.6.2.6.	W
b31 to b24	—	Reserved	The write value should be 0.	W

56.2.2 Surface Control Register (CONTROL2)

Address(es): DRW.CONTROL2 400E 4004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RLEPIXELWIDTH[1:0]		BDIA	BSIA	CLUTFORM	COLKEY	CLUTENABL	RLEENABL	WRITEALPHA[1:0]		WRITEFORMAT[1:0]		READFORMAT[1:0]		TEXTURE	TEXTURE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TEXTURE	TEXTURE	BC2	BDI	BSI	BDF	BSF	WRITEFORMAT2	BDFA	BSFA	READFORMAT[3:2]		USEACB	PATTERN	TEXTURE	PATTERN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

.

Bit	Symbol	Bit name	Description	R/W
b0	PATTERN ENABLE	Pattern Color Enable for Pixel Source	Pixel source is a pattern color (blend of COLOR1 and COLOR2 depending on PATTERN and pattern index). 0: Disable pattern 1: Enable pattern. When patterns are used to fill a primitive an index into the pattern bit mask is generated for each pixel with the U limiter. Depending on the pattern bits the color is selected from COLOR1 and COLOR2 registers. Fractional indices can be interpolated between those two values by using TEXTUREFILTERX = 1. The pattern can be wrapped by using TEXTURECLAMPX = 0, and the mask must be set in the TEXMASK register using the mask for u.	W
b1	TEXTURE ENABLE	Texture Enable for Pixel Source	Pixel source is read from texture and used as an alpha to blend between COLOR1 and COLOR2. 0: Disable texture 1: Enable texture.	W
b2	PATTERN SOURCEL5	Limiter 5 Enable for Pattern Index	Limiter 5 is used as pattern index instead of the default U limiter. Limiter 5 can be combined with limiter 6 to form a quadratic limiter that can be used to make quadratic pattern functions to draw radial patterns.	W
b3	USEACB	Alpha Blend Mode	0: Use WRITEALPHA[1:0] mode 1: Use full alpha channel blending mode.	W

Bit	Symbol	Bit name	Description	R/W
b5, b4	READ FORMAT[3:2]	Texture Format Descriptor	Bits [3] and [2] of the texture buffer format. See the detailed description of the READFORMAT[1:0] bit in this section.	W
b6	BSFA	Blend Source Factor for Alpha Channel	Valid in alpha channel blending mode (USEACB = 1). 0: Use 1.0 as blend source factor for alpha channel 1: Use alpha as blend source factor for alpha channel.	W
b7	BDFA	Blend Destination Factor for Alpha Channel	Valid in alpha channel blending mode (USEACB = 1). 0: Use 1.0 as blend destination factor for alpha channel 1: Use alpha as blend destination factor for alpha channel.	W
b8	WRITE FORMAT2	Writeback Framebuffer Format	Bit [2] of framebuffer pixel format. See the description of WRITEFORMAT[1:0] in this section.	W
b9	BSF	Blend Source Factor	Source factor is alpha (factor is 1 per default). 0: Use 1.0 as blend source factor 1: Use alpha as blend source factor.	W
b10	BDF	Blend Destination Factor	Destination factor is alpha (factor is 1 per default). 0: Use 1.0 as blend destination factor 1: Use alpha as blend destination factor.	W
b11	BSI	Blend Source Factor Inverted	Source factor is inverted (meaning 1-a or 1-1 depending on BSF). 0: Use blend factor as specified through BSF 1: Invert blend source factor (1-x).	W
b12	BDI	Blend Destination Factor Inverted	Destination factor is inverted (meaning 1-a or 1-1 depending on BDF). 0: Use blend factor as specified through BDF 1: Invert blend destination factor (1-x).	W
b13	BC2	Blend color 2	Select of blend color 2 instead of framebuffer pixel. 0: Use pixel from framebuffer as destination (DST) 1: Use color 2 as destination (DST).	W
b14	TEXTURE CLAMPX	Calculating U Limiter Outside Used Texture	This bit describes what happens when the U limiter (x direction in texture space) calculates a u value outside of the used texture. 0: Texture wrap mode: Integer part of the calculated value from the U limiter is AND gated with TEXUMASK, resulting in a repetition of texture in the x/u direction 1: Texture clamp mode: Texture color at the border of the texture is taken, resulting in a repetition of texture border color in the x/u direction.	W
b15	TEXTURE CLAMPY	Calculating V Limiter Outside Used Texture	This bit describes what happens when the V limiter (y direction in texture space) calculates a v value outside of the used texture. 0: Texture wrap mode: Integer part of the calculated value from the V limiter is AND gated with TEXVMASK, resulting in a repetition of texture in the y/v direction. 1: Texture clamp mode: Texture color at the border of the texture is taken, resulting in a repetition of texture border color in the y/v direction.	W
b16	TEXTURE FILTERX	Linear Filtering on Texture U Axis	0: No filtering on texture U axis 1: Linear filtering on texture U axis.	W
b17	TEXTURE FILTERY	Linear Filtering on Texture V Axis	0: No filtering on texture V axis 1: Linear filtering on texture V axis.	W
b19, b18	READ FORMAT[1:0]	Texture Format Descriptor	Pixel format of the texture buffer. b5 b4 b19 b18 0 0 0 0: 8 bpp a (8) 0 0 0 1: 16 bpp RGB (565) 0 0 1 0: 32 bpp aRGB (8888) 0 0 1 1: 16 bpp aRGB (4444) 0 1 0 0: 16 bpp aRGB (1555) 0 1 0 1: 8 bpp aCLUT (44), 4 bit alpha and 4 bit indexed color 1 0 0 1: 8 bpp CLUT (8)/I (8), 8 bit indexed color/luminance 1 0 1 0: 4 bpp CLUT (4)/I (4), 4 bit indexed color/luminance 1 0 1 1: 2 bpp CLUT (2)/I (2), 2 bit indexed color/luminance 1 1 0 0: 1 bpp CLUT (1)/I (1), 1 bit indexed color/luminance. Other settings are prohibited.	W

Bit	Symbol	Bit name	Description	R/W
b21, b20	WRITE FORMAT[1:0]	Writeback Framebuffer Format	Pixel format of the framebuffer. b8 b21 b20 0 0 0: 8 bpp a (8) 0 0 1: 16 bpp RGB (565) 0 1 0: 32 bpp aRGB (8888) 0 1 1: 16 bpp aRGB (4444). Other settings are prohibited.	W
b23, b22	WRITE ALPHA[1:0]	Writeback Alpha Source for Framebuffer	• In non-alpha channel blending mode (USEACB = 0): Sets the alpha source for the framebuffer. b23 b22 0 0: Use alpha from color 2 0 1: Use source alpha (pixel coverage) 1 0: Use 0.0 as alpha 1 1: Use alpha from framebuffer. • In alpha channel blending mode (USEACB = 1): Blends alpha in color 2 instead of framebuffer alpha. 00b: BC2A = 1: Use alpha in color 2 as destination (DST_A) else: BC2A = 0: Use alpha from framebuffer as destination (DST_A).	W
b24	RLE ENABLE	RLE Enable	0: Disable RLE 1: Enable RLE.	W
b25	CLUT ENABLE	CLUT Enable	0: Disable CLUT 1: Enable CLUT. If CLUTENABLE = 0 (CLUT disabled), the index is directly put on the RGB channels.	W
b26	COLKEY ENABLE	Color Keying Enable	0: Disable color keying 1: Enable color keying.	W
b27	CLUT FORMAT	CLUT Format	0: Format CLUT as aRGB (8888) 1: Format CLUT as RGB (565).	W
b28	BSIA	Blend Source Factor Inverted in Alpha Channel	• In alpha channel blending mode (USEACB = 1): 0: Use blend factor as specified through BSFA 1: Invert blend source factor (1-x).	W
b29	BDIA	Blend Destination Factor Inverted in Alpha Channel	• In alpha channel blending mode (USEACB = 1): 0: Use blend factor as specified through BDFA 1: Invert destination blend factor (1-x).	W
b31, 30	RLE PIXEL WIDTH[1:0]	Texel Width for RLE Unit	b31 b30 0 0: 1 byte per texel 0 1: 2 bytes per texel 1 0: 3 bytes per texel 1 1: 4 bytes per texel.	W

56.2.3 Interrupt Control Register (IRQCTL)

Address(es): DRW.IRQCTL 400E 40C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	BUSIRQCLR	BUSIRQEN	DLISTIRQCLR	ENUMIRQCLR	DLISTIRQEN	ENUMIRQEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ENUMIRQEN	ENUMIRQ Interrupt Mask Enable	0: Disable (mask) ENUMIRQ enumeration interrupt 1: Enable (unmask) ENUMIRQ enumeration interrupt.	W
b1	DLISTIRQEN	DLISTIRQ Interrupt Mask Enable	0: Disable (mask) DLISTIRQ display list interrupt 1: Enable (unmask) DLISTIRQ display list interrupt.	W

Bit	Symbol	Bit name	Description	R/W
b2	ENUMIRQCLR	Clear ENUMIRQ	0: Do not clear ENUMIRQ enumeration interrupt 1: Clear ENUMIRQ enumeration interrupt.	W
b3	DLISTIRQCLR	Clear DLISTIRQ	0: Do not clear DLISTIRQ display list interrupt 1: Clear DLISTIRQ display list interrupt.	W
b4	BUSIRQEN	BUSIRQ Interrupt Mask Enable	0: Disable (mask) BUSIRQ bus error interrupt 1: Enable (unmask) BUSIRQ bus error interrupt.	W
b5	BUSIRQCLR	Clear BUSIRQ	0: Do not clear BUSIRQ bus error interrupt 1: Clear BUSIRQ bus error interrupt.	W
b31 to b6	—	Reserved	The write value should be 0.	W

56.2.4 Cache Control Register (CACHECTL)

Address(es): DRW.CACHECTL 400E 40C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	CFLUSHTX	CENABLETX	CFLUSHFX	CENABLE FX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CENABLEFX	Framebuffer Cache Enable	0: Disable the framebuffer cache 1: Enable the framebuffer cache.	W
b1	CFLUSHFX	Flush Framebuffer Cache	0: Do not flush the framebuffer cache 1: Flush the framebuffer cache.	W
b2	CENABLETX	Texture Cache Enable	0: Disable the texture cache 1: Enable the texture cache.	W
b3	CFLUSHTX	Flush Texture Cache	0: Do not flush the texture cache 1: Flush the texture cache.	W
b31 to b4	—	Reserved	The write value should be 0.	W

56.2.5 Status Control Register (STATUS)

Address(es): DRW.STATUS 400E 4000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	BUSERRMDL	BUSERRMTX	BUSERRMFB	—	BUSIRQ	DLISTIRQ	ENUMIRQ	DLISTACTIVE	CACHEDIRTY	BUSYWRITE	BUSYENUM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BUSY ENUM	Enumeration Unit Status	0: Enumeration unit idle 1: Enumeration unit is busy, new primitive cannot be started.	R

Bit	Symbol	Bit name	Description	R/W
b1	BUSY WRITE	Framebuffer Writeback Status	0: Framebuffer writeback finished 1: Framebuffer writeback busy, framebuffer type cannot be changed.	R
b2	CACHE DIRTY	Framebuffer Cache Status	0: Framebuffer cache is not dirty 1: Framebuffer cache is dirty, and frame should not be flipped.	R
b3	DLIST ACTIVE	Display List Reader Status	0: Display list reader is idle 1: Display list reader is busy, and no direct write access to registers allowed.	R
b4	ENUM IRQ	Enumeration Interrupt Triggered	0: Enumeration not finished or interrupt disabled 1: Enumeration finished interrupt triggered.	R
b5	DLIST IRQ	Display List Interrupt Triggered	0: Display list not finished or interrupt disabled 1: Display list finished interrupt triggered.	R
b6	BUS IRQ	Bus Error Interrupt Triggered	0: No bus error occurred or interrupt disabled 1: Bus error interrupt triggered.	R
b7	—	Reserved	This bit is read as 0.	R
b8	BUSERR MFB	Framebuffer Bus Error Interrupt Triggered	0: No framebuffer bus error occurred or interrupt disabled 1: Framebuffer bus error interrupt triggered.	R
b9	BUSERR MTXMRL	Texture Bus Error Interrupt Triggered*1	0: No texture bus error occurred or interrupt disabled 1: Texture bus error interrupt triggered.	R
b10	BUSERR MDL	Display List Bus Error Interrupt Triggered	0: No display list bus error occurred or interrupt disabled 1: Display list bus error interrupt triggered.	R
b31 to b11	—	Reserved	These bits are read as 0.	R

Note 1. Because the RLE unit is also reading data through the texture bus, an error during RLE data access is also reflected in this bit.

56.2.6 Hardware Version and Feature Set ID Register (HWREVISION)

Address(es): DRW.HWREVISION 400E 4004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	ACBLEND	—	COLORKEY	TEXCLUT256	RLEUNIT	—	TEXCLU	PERFCOUNT	TXCACHE	FBCACHE	DLR	—
Value after reset:	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	REV[11:0]											
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b11 to b0	REV[11:0]	Revision Number	Revision number.	R
b16 to b12	—	Reserved	These bits are read as 0.	R
b17	DLR	Display List Reader Available	Display list reader is available.	R
b18	FBCACHE	Framebuffer Cache Available	Framebuffer cache is available.	R
b19	TX CACHE	Texture Cache Available	Texture cache is available.	R
b20	PERF COUNT	Two Performance Counter Available	Two performance counter is available.	R
b21	TEXCLUT	Texture CLUT Available	Texture CLUT is available.	R
b22	—	Reserved	This bit is read as 0.	R
b23	RLEUNIT	RLE Unit Available	RLE unit is available.	R

Bit	Symbol	Bit name	Description	R/W
b24	TEX CLUT256	Texture CLUT with 16 or 256 Entries Available	Texture CLUT with 16 or 256 entries is available.	R
b25	COLORKEY	Color Key Available	Color key is available.	R
b26	—	Reserved	This bit is read as 1.	R
b27	ACBLEND	Alpha Channel Blending Available	Alpha channel blending is available.	R
b31 to b28	—	Reserved	These bits are read as 0.	R

56.2.7 Base Color Register (COLOR1)

Address(es): DRW.COLOR1 400E 4064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	COLOR1A[7:0]								COLOR1R[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	COLOR1G[7:0]								COLOR1B[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	COLOR1B[7:0]	Blue Channel of Color 1	Specifies blue channel of color 1.	W
b15 to b8	COLOR1G[7:0]	Green Channel of Color 1	Specifies green channel of color 1.	W
b23 to b16	COLOR1R[7:0]	Red Channel of Color 1	Specifies red channel of color 1.	W
b31 to b24	COLOR1A[7:0]	Alpha Channel of Color 1	Specifies alpha channel of color 1. 00h: Transparent ... FFh: Opaque.	W

56.2.8 Secondary Color Register (COLOR2)

Address(es): DRW.COLOR2 400E 4068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	COLOR2A[7:0]								COLOR2R[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	COLOR2G[7:0]								COLOR2B[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	COLOR2B[7:0]	Blue Channel of Color 2	Specifies blue channel of color 2.	W
b15 to b8	COLOR2G[7:0]	Green Channel of Color 2	Specifies green channel of color 2.	W
b23 to b16	COLOR2R[7:0]	Red Channel of Color 2	Specifies red channel of color 2.	W
b31 to b24	COLOR2A[7:0]	Alpha Channel of Color 2	Specifies alpha channel of color 2. 00h: Transparent ... FFh: Opaque.	W

56.2.9 Pattern Register (PATTERN)

Address(es): DRW.PATTERN 400E 4074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	PATTERN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	PATTERN[7:0]	Bitmap of the Pattern	Specifies bitmap of the pattern.	W
b31 to b8	—	Reserved	The write value should be 0.	W

56.2.10 Limiter N Start Value Register (LnSTART)

Address(es): DRW.L1START 400E 4010h, DRW.L2START 400E 4014h, DRW.L3START 400E 4018h, DRW.L4START 400E 401Ch, DRW.L5START 400E 4020h, DRW.L6START 400E 4024h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LSTART[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LSTART[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LSTART[31:0]	Start Value of the nth Limiter	Specifies start value of the nth limiter.	W

56.2.11 Limiter N X-Axis Increment Register (LnXADD)

Address(es): DRW.L1XADD 400E 4028h, DRW.L2XADD 400E 402Ch, DRW.L3XADD 400E 4030h, DRW.L4XADD 400E 4034h, DRW.L5XADD 400E 4038h, DRW.L6XADD 400E 403Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LXADD[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LXADD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LXADD[31:0]	X-Axis Increment	Specifies x-axis increment.	W

56.2.12 Limiter N Y-Axis Increment Register (LnYADD)

Address(es): DRW.L1YADD 400E 4040h, DRW.L2YADD 400E 4044h, DRW.L3YADD 400E 4048h, DRW.L4YADD 400E 404Ch, DRW.L5YADD 400E 4050h, DRW.L6YADD 400E 4054h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LYADD[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LYADD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LYADD[31:0]	Y-Axis Increment	Specifies y-axis increment.	W

56.2.13 Limiter M Band Width Parameter Register (LmBAND)

Address(es): DRW.L1BAND 400E 4058h, DRW.L2BAND 400E 405Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LBAND[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LBAND[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LBAND[31:0]	Limiter m Band Width Parameter	Specifies limiter m band width parameter.	W

56.2.14 Texture Base Address Register (TEXORIGIN)

Address(es): DRW.TEXORIGIN 400E 40BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TEXORIGIN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TEXORIGIN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	TEXORIGIN[31:0]	Texture Base Address	Specifies texture base address.	W

56.2.15 Texels Per Texture Line Register (TEXPITCH)

Address(es): DRW.TEXPITCH 400E 40B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TEXPITCH[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TEXPITCH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	TEXPITCH[31:0]	Texels Per Texture Line	Specifies texels per texture line. Valid range: 0 to 2048.	W

56.2.16 Texture Size or Texture Address Mask Register (TEXMASK)

Address(es): DRW.TEXMASK 400E 40B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TEXVMASK[20:5]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TEXVMASK[4:0]					TEXUMASK[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	TEXUMASK[10:0]	U Mask in Texture Mode	Specifies the U mask. Set to texture_width -1. • In texture wrapping mode (CONTROL2.TEXTURECLAMPX = 0): Texture_width must be a power of 2 • In texture clamping mode (CONTROL2.TEXTURECLAMPX = 1): All widths up to 2048 are allowed.	W
b31 to b11	TEXVMASK[20:0]	V Mask in Texture Mode	Specifies the V mask. Set to DRWTEXPITCH (texture_height - 1). • In texture wrapping mode (CONTROL2.TEXTURECLAMPY = 0): Texture_height must be a power of 2 • In texture clamping mode (CONTROL2.TEXTURECLAMPY = 1): All heights up to 1024 are allowed.	W

56.2.17 U Limiter Start Value Register (LUSTART)

Address(es): DRW.LUSTART 400E 4090h

<table>
<tr><td></td><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td></tr>
<tr><td></td><td colspan="16">LUSTART[31:16]</td></tr>
<tr><td>Value after reset:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td></td><td colspan="16">LUSTART[15:0]</td></tr>
<tr><td>Value after reset:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LUSTART[31:0]	U Limiter Start Value	Specifies U limiter start value.	W

56.2.18 U Limiter X-Axis Increment Register (LUXADD)

Address(es): DRW.LUXADD 400E 4094h

<table>
<tr><td></td><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td></tr>
<tr><td></td><td colspan="16">LUXADD[31:16]</td></tr>
<tr><td>Value after reset:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td></td><td colspan="16">LUXADD[15:0]</td></tr>
<tr><td>Value after reset:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LUXADD[31:0]	U Limiter X-Axis Increment	Specifies U limiter x-axis increment.	W

56.2.19 U Limiter Y-Axis Increment Register (LUYADD)

Address(es): DRW.LUYADD 400E 4098h

<table>
<tr><td></td><td>b31</td><td>b30</td><td>b29</td><td>b28</td><td>b27</td><td>b26</td><td>b25</td><td>b24</td><td>b23</td><td>b22</td><td>b21</td><td>b20</td><td>b19</td><td>b18</td><td>b17</td><td>b16</td></tr>
<tr><td></td><td colspan="16">LUYADD[31:16]</td></tr>
<tr><td>Value after reset:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr>
<tr><td></td><td colspan="16">LUYADD[15:0]</td></tr>
<tr><td>Value after reset:</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LUYADD[31:0]	U Limiter Y-Axis Increment	Specifies U limiter y-axis increment.	W

56.2.20 V Limiter Start Value Integer Part Register (LVSTARTI)

Address(es): DRW.LVSTARTI 400E 409Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LVSTARTI[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LVSTARTI[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit nName	Description	R/W
b31 to b0	LVSTARTI[31:0]	V Limiter Start Value Integer Part	Specifies integer part of V limiter start value.	W

56.2.21 V Limiter Start Value Fractional Part Register (LVSTARTF)

Address(es): DRW.LVSTARTF 400E 40A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LVSTARTF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	LVSTARTF[15:0]	V Limiter Start Value Fractional Part	Specifies fractional part of V limiter start value.	W
b31 to b16	—	Reserved	The write value should be 0.	W

56.2.22 V Limiter X-Axis Increment Integer Part Register (LVXADDI)

Address(es): DRW.LVXADDI 400E 40A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LVXADDI[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LVXADDI[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LVXADDI[31:0]	V Limiter X-Axis Increment Integer Part	Specifies integer part of V limiter x-axis increment.	W

56.2.23 V Limiter Y-Axis Increment Integer Part Register (LVYADDI)

Address(es): DRW.LVYADDI 400E 40A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LVYADDI[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LVYADDI[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	LVYADDI[31:0]	V Limiter Y-Axis Increment Integer Part	Specifies integer part of V limiter y-axis increment.	W

56.2.24 V Limiter Increment Fractional Parts Register (LVYXADDF)

Address(es): DRW.LVYXADDF 400E 40ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LVYADDF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LVXADDF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	LVXADDF[15:0]	V Limiter X-Axis Increment Fractional Part	Specifies fractional part of V limiter x-axis increment.	W
b31 to b16	LVYADDF[15:0]	V Limiter Y-Axis Increment Fractional Part	Specifies fractional part of V limiter y-axis increment.	W

56.2.25 CLUT Start Address Register (TEXCLADDR)

Address(es): DRW.TEXCLADDR 400E 40DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CLADDR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CLADDR[7:0]	Texture CLUT Start Address	Specifies texture CLUT start address.	W

Bit	Symbol	Bit name	Description	R/W
b31 to b8	—	Reserved	The write value should be 0.	W

56.2.26 CLUT Data Register (TEXCLDATA)

Address(es): DRW.TEXCLDATA 400E 40E0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CLDATA[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CLDATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	CLDATA[31:0]	Texture CLUT Data	Specifies texture CLUT data.	W

56.2.27 CLUT Offset Register (TEXCLOFFSET)

Address(es): DRW.TEXCLOFFSET 400E 40E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CLOFFSET[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CLOFFSET[7:0]	Texture CLUT Offset	Specifies texture CLUT offset. CLOFFSET[7:0] is OR gated with the original index.	W
b31 to b8	—	Reserved	The write value should be 0.	W

56.2.28 Color Key Register (COLKEY)

Address(es): DRW.COLKEY 400E 40E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	COLKEYR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	COLKEYG[7:0]								COLKEYB[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	COLKEYB[7:0]	Blue Channel of Color Key	Specifies blue channel of color key.	W
b15 to b8	COLKEYG[7:0]	Green Channel of Color Key	Specifies green channel of color key.	W
b23 to b16	COLKEYR[7:0]	Red Channel of Color Key	Specifies red channel of color key.	W
b31 to b24	—	Reserved	The write value should be 0.	W

56.2.29 Bounding Box Dimension Register (SIZE)

Address(es): DRW.SIZE 400E 4078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SIZEY[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SIZEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	SIZEX[15:0]	Bounding Box Width	Specifies width of bounding box in pixels. Valid range: 0 to 1024.	W
b31 to b16	SIZEY[15:0]	Bounding Box Height	Specifies height of bounding box in pixels. Valid range: 0 to 1024.	W

56.2.30 Framebuffer Pitch And Spanstore Delay Register (PITCH)

Address(es): DRW.PITCH 400E 407Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SSD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PITCH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	PITCH[15:0]	Pitch of the Framebuffer	A negative width can be used to render bottom-up instead of top-down.	W
b31 to b16	SSD[15:0]	Spanstore Delay	Specifies number of scan lines to delay spanstore operations.	W

56.2.31 Framebuffer Base Address Register (ORIGIN)

Address(es): DRW.ORIGIN 400E 4080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ORIGIN[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ORIGIN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	ORIGIN[31:0]	Address of the First Pixel in Framebuffer	Writing to ORIGIN triggers the start of rendering.	W

56.2.32 Display List Start Address Register (DLISTSTART)

Address(es): DRW.DLISTSTART 400E 40C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DLISTSTART[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DLISTSTART[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	DLISTSTART[31:0]	Display List Start Address	Setting a new display list base address triggers execution of the new display list. Execution stops only when a new list is set or the current list terminates.	W

56.2.33 Performance Counters Control Register (PERFTRIGGER)

Address(es): DRW.PERFTRIGGER 400E 40D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PERFTRIGGER2[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PERFTRIGGER1[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	PERFTRIGGER1[15:0]	Trigger of Performance Counter 1	Selects the internal event that increments the PERFCOUNT1 register. 0: Disable performance counter 1: Select 2D Drawing Engine active cycles 2: Select framebuffer read access 3: Select framebuffer write access 4: Select texture read access 5: Select invisible pixels (enumerated but selected with alpha 0%) 6: Select invisible pixels while internal FIFO is empty (lost cycles) 7: Select display list reader active cycles 8: Select framebuffer read hits 9: Select framebuffer read misses 10: Select framebuffer write hits 11: Select framebuffer write misses 12: Select texture read hits 13: Select texture read misses 31: Select every clock cycle (for use as timer).	W
b31 to b16	PERFTRIGGER2[15:0]	Trigger of Performance Counter 2	Same as for PERFTRIGGER1, but for performance counter 2.	W

56.2.34 Performance Counter k (PERFCOUNTk) (k = 1, 2)

Address(es): DRW.PERFCOUNT1 400E 40CCh, DRW.PERFCOUNT2 400E 40D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PERFCOUNT[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PERFCOUNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	PERFCOUNT[31:0]	Performance Counter k Value	Specifies counter k value. The counter is reset by writing PERFCOUNTk = 0000_0000h.	R/W

56.3 Drawing Features

56.3.1 Drawing Features Summary

56.3.1.1 Color formats

Supported color formats are:

Framebuffer formats

- 8-bit: a (8)
- 16-bit: RGB (565), aRGB (4444)
- 32-bit: aRGB (8888).

Texture formats

- 1-bit: CLUT (1)/I (1)
- 2-bit: CLUT (2)/I (2)
- 4-bit: CLUT (4)/I (4)
- 8-bit: a (8), CLUT (8)/I (8), aCLUT (44)
- 16-bit: aRGB (4444), aRGB (1555), RGB (565)
- 24-bit: RGB (888) (run length encoded (RLE) unit)
- 32-bit: aRGB (8888).

CLUT formats use a 256-entry color lookup table.

56.3.1.2 BitBLT features

The 2D Drawing Engine supports the BitBLT features using its vector drawing function to draw a rectangle and texture it based on the selected BitBLT function. This approach results in the following BitBLT features:

- Fill
- Copy
- Stretch BitBLT
- Rotate and scale
- Alpha blending

- Bilinear filtering
- Color conversion
- Subpixel exact placement.

56.3.1.3 Vector drawing features

The vector 2D Drawing Engine uses a half-plane rendering approach, which simplifies implementation of edge anti-aliasing and blurring features without much overhead. When combining some of its functional units, the module can draw not only linear primitives such as lines or polygons, but also quadratic equation-based primitives such as circles and ellipsoids. The following primitives are supported:

- Lines
- Polygons
- Circles and ellipses
- Quadratic curves (software driver support)
- 2D texture mapping
- Bilinear filtering of the textures.

56.3.2 Vector Drawing

For a detailed explanation of the algorithms, see section 56.6, Rendering Pipeline. Supported vector drawing includes:

Lines

- Arbitrary width
- Round endpoints
- Truncated endpoints
- Alpha gradients
- Soft edges (blurring)
- Render attribute: color, pattern, or texture.

Polygons

- Triangles and quadrangles (complex polygons are tessellated by software)
- Alpha gradients
- Soft edges (blurring)
- Per edge controls for anti-aliasing
- Render attribute: color, pattern, or texture.

Circles and ellipses

- All conic sections
- Filled or with arbitrary width
- Arcs of 0° to 360°
- Soft edges
- Alpha gradients
- Render attribute: color, pattern, or texture.

Quadratic Bézier

- Approximated by circle arcs
- Arbitrary width

- Round or truncated endpoints
- Outlines, blurring
- Alpha gradients
- Render attribute: color.

Texture mapping

- 2D array of pixels that can be mapped implicitly or explicitly on all primitives provided by the 2D Drawing Engine
- Translation, rotation, and scaling/shearing
- Bilinear filtering of the textures
- 3D-like texturing accomplished with line-by-line mapping, if constant in one axis.

56.3.3 BitBLT

A dedicated BitBLT unit is not required in the 2D Drawing Engine. The rendering pipeline described for vector drawing is used as the BitBLT unit and already provides a 1 pixel/cycle throughput. For details, see section 56.6, Rendering Pipeline.

56.3.3.1 Fill

Any rectangle in the framebuffer can be filled with any value. Possible color formats are any 8-, 16-, or 32-bpp format. The driver optimizes the fill to gain the full benefit of 32-bit parallel rasterization. If the selected color format is less than 32 bpp, the driver corrects the alignment and fills 32 bits per clock, resulting in 2 to 4 times faster fill performance for 8- and 16-bpp formats.

56.3.3.2 Copy

Any rectangle in the framebuffer can be filled with any rectangular data from the texture input. When the texture input points to the framebuffer, copying from framebuffer to framebuffer is possible. To avoid copy problems because of overlapping source and destination areas, the copy start point can be selected from top left to bottom right. Possible color formats are any 8-, 16-, or 32-bpp format.

The driver optimizes the copy to gain the full benefit of 32-bit parallel rasterization. If the selected color format is less than 32 bpp, the driver corrects the alignment and copies 32 bits per clock, resulting in 2 to 4 times faster copy performance for 8- and 16-bpp formats.

56.3.3.3 Stretch BitBLT

This is similar to the normal copy operation. Because the copy is done as a type of texture mapping, the full texture mapping feature set can be used. Any scaling ratios in the x and y directions is selectable, and filtering can be enabled independently for each axis.

56.3.3.4 Rotate and scale

This is similar to the normal copy operation. Because the copy is done as a type of texture mapping, the full texture mapping feature set can be used. Any scaling ratios in the x and y direction and any rotation angle is selectable. The x and y directions can be rotated and scaled independently, and filtering for the scalers can be enabled independently for each axis.

56.3.3.5 Alpha blending

Alpha blending is a fundamental block in the rendering pipeline, so the full alpha blend feature set is available for any BitBLT operation. It is possible to copy an area and blend it over the destination by using any constant global alpha value (register value) or by using an alpha mask. The alpha mask is part of the texture data and can be either a per-pixel value together with a pixel color (aRGB formats) or an alpha-only format using a register color.

In addition to the color channels, the alpha channel can be blended. The formula for the alpha channel can be set independently from the formula for the color channels.

56.3.3.6 Bilinear filtering

The texture unit can be used to scale, rotate, or shear images. The texturing result can be filtered in the x and y directions independently. When selecting both filters, the result is a bilinear filtered texture. Using the unit twice with two independent textures would generate trilinear filtered bitmaps, improving the visual impression for high dynamic scale ratios.

56.3.3.7 Color conversion

Color conversion is required when using different texture formats than the framebuffer format. For saving texture memory, several formats are supported with less bpp than the framebuffer usually has. The 2D Drawing Engine always operates internally with 32-bpp aRGB (8888). All input data is converted into 32 bpp, and finally is converted back into the framebuffer format.

56.4 Input and Output Data Formats

56.4.1 Source and Destination Data

There are two possible inputs, the framebuffer and the texture or pattern input. The output is always the framebuffer.

Every drawing operation is internally rendered in 32 bpp aRGB (8888). If the input color does not provide an alpha channel, the blue channel is taken as the alpha channel. This alpha can be substituted with any alpha (for example, by an external constant) during the colorization step in the 2D Drawing Engine.

56.4.2 Framebuffer Color Formats

Table 56.1 shows the supported framebuffer color formats.

Table 56.1 Framebuffer color formats

Framebuffer memory occupation	Format	Remarks
8 bpp	a (8)	This color format uses 1 byte per pixel. The alpha channel is internally replicated on the red, blue, and green channels and can be substituted with any color during the color step in the 2D Drawing Engine.
16 bpp	RGB (565)	This color format uses 2 bytes per pixel with 5 bits for red and blue and 6 bits for green. The blue color is taken as the alpha channel during color conversion. The alpha can be substituted with any alpha during the colorization step in the 2D Drawing Engine.
	aRGB (4444)	This color format uses 2 bytes per pixel with 4 bits for each color and alpha channel.
32 bpp	aRGB (8888)	This color format uses 4 bytes per pixel with 8 bits for each color and alpha channel.

The framebuffer color format is selected in the Surface Control Register bits, CONTROL2.READFORMAT[2:0].

56.4.3 Texture Color Formats

Table 56.2 shows the supported texture color formats.

Table 56.2 Texture color formats (1 of 2)

Texture memory occupation	Format	Remarks
1 bpp	CLUT (1)/I (1)	In this mode, a 1-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.
2 bpp	CLUT (2)/I (2)	In this mode, a 2-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.
4 bpp	CLUT (4)/I (4)	In this mode, a 4-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.

Table 56.2 Texture color formats (2 of 2)

Texture memory occupation	Format	Remarks
8 bpp	CLUT (8)/I (8)	In this mode, an 8-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value.
	a (8)	This color format uses 1 byte per pixel. The alpha channel is internally replicated on the red, blue, and green channels and can be substituted with any color during the colorization step in the 2D Drawing Engine.
	aCLUT (44)	This color format uses 1 byte per pixel. 4 bits are used as an alpha value and 4 bits are used as an index to a color palette. This approach saves space if 16 colors are sufficient to describe the image, because the next bigger alpha format would be 2-byte aRGB (4444).
16 bpp	RGB(565)	This color format uses 2 bytes per pixel with 5 bits for red and blue and 6 bits for green. The blue color is taken as the alpha channel during color conversion. The alpha can be substituted with any alpha during the colorization step in the 2D Drawing Engine.
	aRGB (4444)	This color format uses 2 bytes per pixel with 4 bits for each color and alpha channel.
	aRGB (1555)	This color format uses 2 bytes per pixel. Every color channel has 5 bits and the topmost single bit is taken as an alpha value. This can be used to hold an image with a transparency mask.
24 bpp	RGB (888)	This color format uses 3 bytes per pixel with 8 bits for each color channel. This format is only available as run length encoded data (RLE compression).
32 bpp	aRGB (8888)	This color format uses 4 bytes per pixel with 8 bits for each color and alpha channel.

The texture color format is selected in the Surface Control Register bits, CONTROL2.WRITEFORMAT[3:0].

56.5 Texture Data Processing

Figure 56.4 shows the processing of texture data.

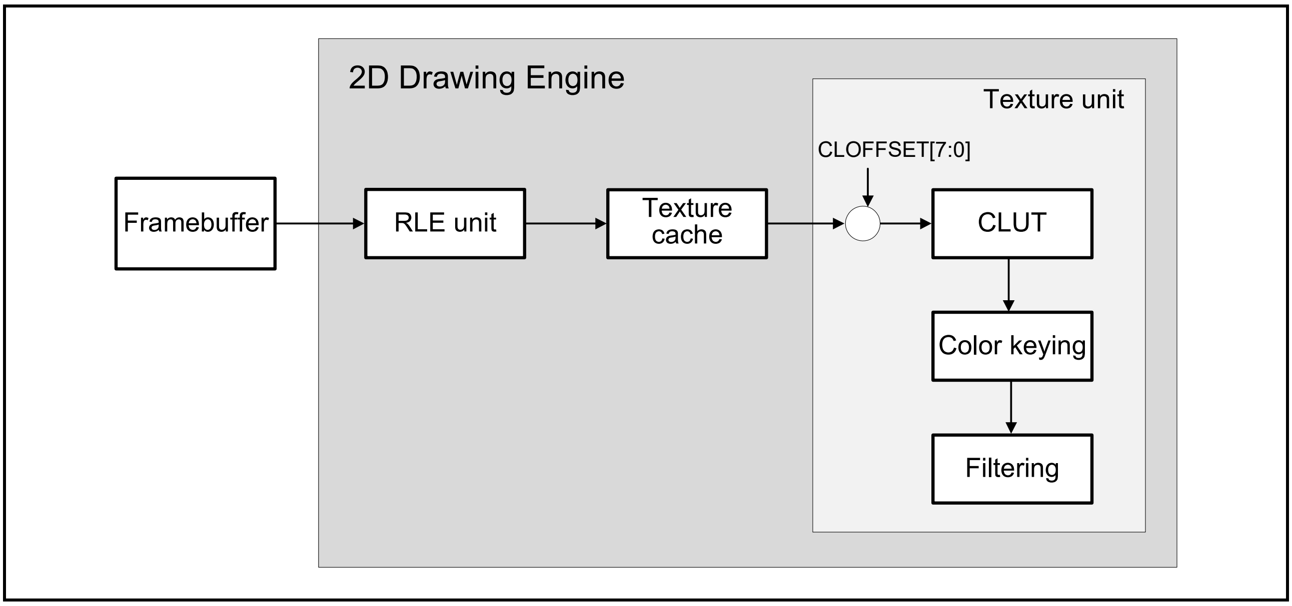


Figure 56.4 Texture data processing

56.5.1 Texture Color Format

Table 56.3 shows the supported texture data formats.

Table 56.3 Texture color formats

Texel bit width	Texel format
32 bits	aRGB (8888)
24 bits	RGB (888)
16 bits	aRGB (4444), aRGB (1555), RGB (565)
8 bits	CLUT (8)I (8), a (8), aCLUT (44)
4 bits	CLUT (4)/I (4)
2 bits	CLUT (2)/I (2)
1 bit	CLUT (1)/I (1)

56.5.2 Run Length Encoded (RLE) Unit

The RLE unit decompresses Targa-like compressed textures and hands the decompressed texel data over to the texture unit. The key features are:

- Support for Targa format
- Avoids the additional Targa limitation to scan lines
- Support for clipping of compressed images, in which the 2D Drawing Engine is allowed to copy only a portion of a larger original texture
- Control of the RLE unit in drawing list operation
- Bypassing of the RLE unit logic if uncompressed textures are fetched from the framebuffer.

Texture cache

The RLE unit feeds the texture cache. The texture cache can be disabled by setting CACHECTL.CENABLETX = 0.

Caution: A texture cache flush operation (CACHECTL.CFLUSHTX = 1) is necessary at the beginning and end of every new RLE texture.

The texture cache and the RLE unit can be bypassed by setting CONTROL2.RLEENABLE = 0.

56.5.2.1 RLE Texel formats

Table 56.4 lists the data formats supported by the RLE unit.

Table 56.4 Texel formats supported by the RLE unit (1 of 2)

Memory texel format	RLE parameters	RLE coded unit format (CONTROL2.RLEPIXELWIDTH[1:0])	Delivered format
32-bit aRGB (8888)	Included in Targa and RLE formats	32 bits (11b)	32 bits
24-bit RGB (888)		24 bits (10b)	32 bits *1
16-bit aRGB (4444), aRGB (1555), RGB (565)		16 bits (01b)	16 bits
8-bit CLUT (8)I (8), a (8), aCLUT (44)		8 bits (00b)	8 bits
4-bit CLUT (4)/I (4)	Optional for RLE *2	8 = 2 x 4 bits	4 + 4 bits

Table 56.4 Texel formats supported by the RLE unit (2 of 2)

Memory texel format	RLE parameters	RLE coded unit format (CONTROL2.RLEPIXELWIDTH[1:0])	Delivered format
2-bit CLUT (2)/I (2)	No RLE		
1-bit CLUT (1)/I (1)			

Note 1. 24-bit RGB (888) encoded texels are delivered as aRGB (8888) with Alpha set to 1.

Note 2. Encoding of textures with 4 bits per texel is not defined by the Targa specification but can be done by:
- Combining two 4-bit texels to one byte
- Padding with 4 zero bits at the end of the file, if the number of texels is odd
- Encoding as with 8-bit texels.

Texel addressing for RLE textures

The address of a texel is the byte address of the first byte of the texel. The origin of the texture is given by the register TEXORIGIN.

Note: The RLE code must begin at a word boundary of the memory.

Caution: When the FIFO is filled, there is no provision to inhibit read access beyond the end of the RLE code. To avoid memory access violations, the RLE code must be padded by 32 memory words, where every bit of each word is set to 1.

56.5.2.2 Targa RLE format

Run-length encoded (RLE) images include two types of data elements:

- Run-length packets
- Raw packets.

The first field (1 byte) of each packet is called the repetition count field. The second field is called the pixel value field (1, 2, 3. or 4 bytes). For run-length packets, the pixel value field contains a single pixel value. For raw packets, the field is a variable number of pixel values.

The highest order bit of the repetition count indicates whether the packet is a raw packet or a run-length packet, as follows:

- If bit [7] of the repetition count is set to 1, the packet is a run-length packet
- If bit [7] of the repetition count is set to zero, the packet is a raw packet.

The lower 7 bits of the repetition count specify how many pixel values are represented by the packet. For a run-length packet, this count indicates how many successive pixels have the pixel value specified in the pixel value field. For raw packets, this count specifies how many pixel values are actually contained in the next field. This 7-bit value is actually encoded as 1 less than the number of pixels in the packet (a value of 0 implies 1 pixel while a value of 7Fh implies 128 pixels).

Run-length packet

Run-length packets are composed of two parts. The first is a repetition count and the second is the pixel value to repeat.

Table 56.5 Run-length packet

Field name	Packet type (must be 1 for run-length)	Pixel count (number of pixels encoded in this packet - 1)	Pixel data (the shared pixel value to be used)
Field size	1 bit	7 bits	Pixel depth (field 5.5)

Raw packet

The raw packet always includes two fields. The first field is the repetition count and the second field is the pixel data field.

Table 56.6 Raw packet

Field name	Packet type (must be 0 for raw packet)	Pixel count (number of pixels encoded by this packet - 1)	Pixel data
Field size	1 bit	7 bits	Pixel depth x pixel count - 1

56.5.3 Color Lookup Table (CLUT)

The color lookup table receives an index that addresses one out of the 256 predefined colors.

The predefined color format can be selected as:

- CONTROL2.CLUTFORMAT = 0: aRGB (8888)
- CONTROL2.CLUTFORMAT = 1: RGB (565).

The CLUT is filled by the use of two registers:

- TEXCLDATA
  The aRGB (8888) color definition is written to this register, while the CLUT address is taken from TEXCLADDR.
- TEXCLADDR.
  This is set to the first address of the CLUT to write to and is automatically incremented after each write to TEXCLDATA.

An offset for indexed formats (CLUT (1), CLUT (2), CLUT (4), and CLUT (8)) can be set up in the TEXCLOFFSET register to allow selecting an offset part of the CLUT. The CLUT index is calculated by CLUT (x) or TEXCLOFFSET.CLOFFSET[7:0].

56.5.3.1 CLUT/I pixel data formats

The following tables explain in which order the pixels are stored within the byte. The left-most pixel is stored at the lowest bit of the memory byte.

CLUT (1)/I (1) format

The CLUT (1)/I (1) format expresses 1 pixel by using a total of 1 bit.

Memory byte	7 (MSB)	6	5	4	3	2	1	0 (LSB)
Pixel	P7	P6	P5	P4	P3	P2	P1	P0

The left-most pixel is stored at lowest bit of the memory byte.

CLUT (2)/I (2) format

The CLUT (2)/I (2) format expresses 1 pixel by using a total of 2 bits.

Memory byte	7 (MSB)	6	5	4	3	2	1	0 (LSB)
Pixel	P3		P2		P1		P0	

The left-most pixel is stored at lowest 2 bits of the memory byte.

CLUT (4)/I (4) format

The CLUT (4)/I (4) format expresses 1 pixel by using a total of 4 bits.

Memory byte	7 (MSB)	6	5	4	3	2	1	0 (LSB)

Pixel	P1	P0

The left-most pixel is stored at lowest 4 bits of the memory byte.

56.5.4 Color Keying

The 2D Drawing Engine provides a color keying unit in front of the texture unit. It operates as follows:

1. If enabled, the incoming color is compared with a transparent color, defined by COLKEY.
2. If the value matches, the alpha and color values are set to 0 to mark the color as transparent and handle it as if alpha was pre-multiplied.
3. If the value does not match, then alpha is set to 1.
4. Additional operations such as alpha_in × alpha_const are still possible.

With this approach, an object such as a round icon can be cut out from a rectangular texture and still can be faded by a constant alpha over the background.

56.6 Rendering Pipeline

56.6.1 Coordinate Transformation

Coordinate transformation such as rotation, translation, projection, and scaling must be done on the application side. This is not part of the 2D Drawing Engine hardware or driver. Because all coordinates fed into the 2D Drawing Engine are in fixed point format, these calculations can be made in fixed point format and do not require a floating point unit.

56.6.2 Rasterization

During rasterization, the vector data of the object must be converted to pixel data. To convert the data, the program sets up the edge interpolation hardware, called a limiter, for each edge of the object that calculates a decision value. The limiter determines which side of the edge the pixel is positioned on. The 2D Drawing Engine includes six internal hardware limiters. In principle, the limiter registers contain the distance between the pixel being processed and the edge.

In the linear setup, a limiter describes a half plane. The intersection of all half planes is the object. If three half planes intersect, a triangle is created as shown in Figure 56.5.

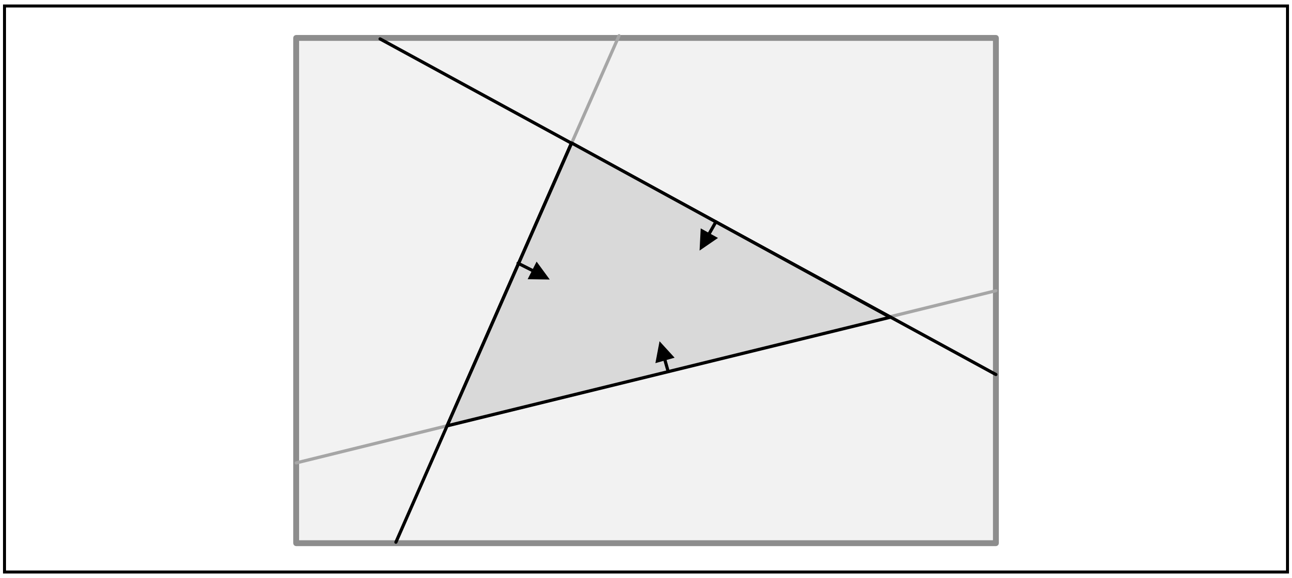


Figure 56.5 Intersection of half planes

The limiter output is clamped to an interval of [0:1]. In limiters 1 and 2 it is possible to apply a band filter before the clamping operation. In this case, the limiter is not describing a half plane but a small band. With this approach, a single limiter can describe a thick line of infinite length.

The output of the different limiters can be combined by the combiner units with a maximum or minimum operation.

Maximum operation describes the union of both half planes, and minimum operation describes the intersection of both half planes. The final output is then used as an alpha value. Edge anti-aliasing can be done with no additional effort with this hardware.

To calculate the decision value for each possible pixel with a limiter, the bounding box of the object must be calculated. Then, the decision value for the top left corner of the bounding box must be calculated for each edge. Finally, the increments for a step in the x direction and a step in the y direction must be calculated. This is done by the CPU in the driver.

With this information, the 2D Drawing Engine scans the whole bounding box and calculates the decision value for every pixel incrementally. For a block diagram of the entire rasterization unit, see section 56.1, Overview.

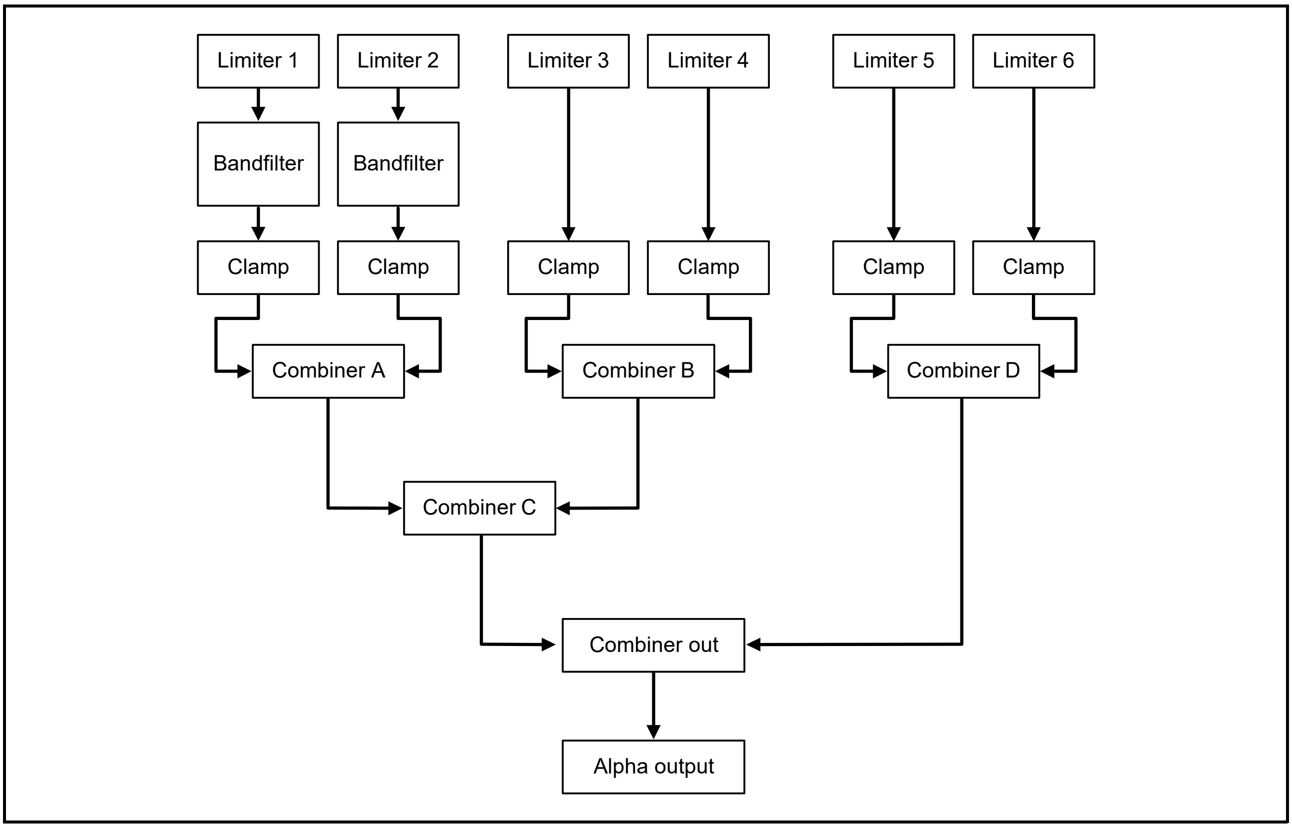


Figure 56.6 Block diagram of rasterization unit

The limiters calculate distances and the combiner units combine them to an alpha value. The combiner units define the conditions for whether or not a pixel is in the bounding box. The alpha value must be greater than 0.

Note: It is possible to have all limiters switched off.

56.6.2.1 Edge setup linear case

(1) Mathematical background

To setup a linear edge, consider the line equation in the classical form:

This can be written as:

$$y = \tilde{a}x + \tilde{b}$$

This can be rewritten as:

$$0 = f(x, y) = \tilde{a}x - y + \tilde{b} = ax + by + c$$

with $a = \tilde{a}, b = -1, c = \tilde{b}$

This is a more general form. Consider a vector form of this equation:

$$\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix}, \vec{n} = \begin{pmatrix} a \\ b \end{pmatrix} \Rightarrow f(x, y) = ax + by + c = \vec{p} \cdot \vec{n} + c$$

If a point $\vec{p}_0$ is on the line:

$$0 = f(x, y) = \vec{p}_0 \cdot \vec{n} + c \Rightarrow c = -\vec{p}_0 \cdot \vec{n}$$

Rewriting the constant, the equation becomes:

$$c = -\vec{p}_0 \cdot \vec{n} \Rightarrow f(x, y) = (\vec{p} - \vec{p}_0) \cdot \vec{n}$$

The vector $\vec{n}$ is called the normal vector and is perpendicular to the line. The setup can be seen in Figure 56.7.

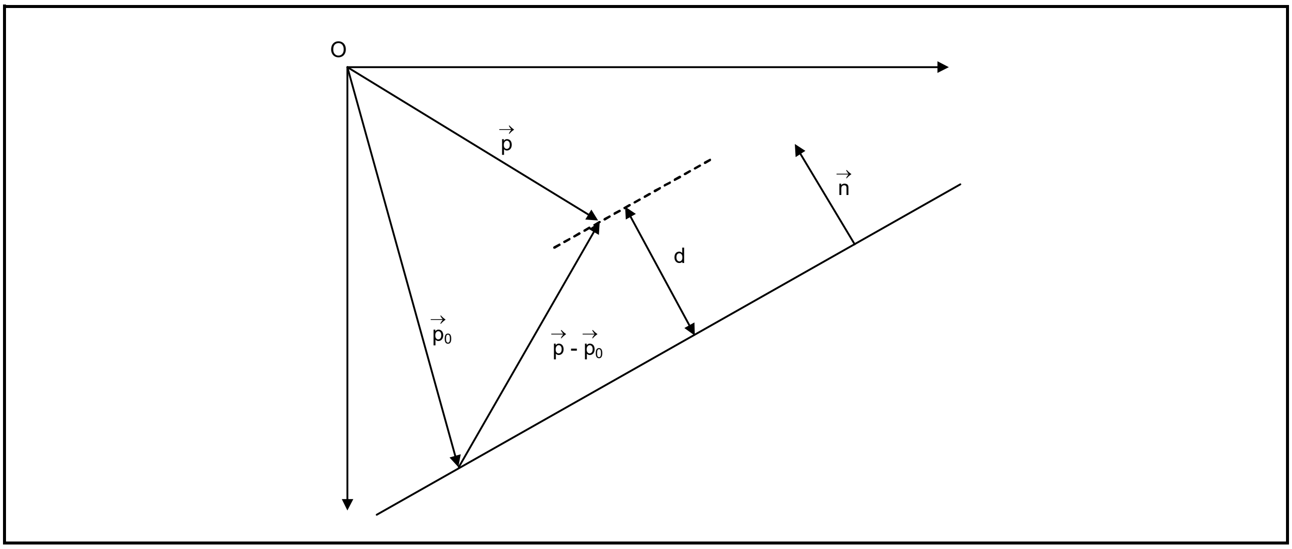


Figure 56.7 Distance of a point to a line

The projection of $\vec{p} - \vec{p}_0$ to $\vec{n}$ is the distance d of the point P to the line. In this case, f (x, y) describes the distance to the line of the pixel with coordinates (x, y).

To calculate the distance of every pixel of the bounding box incrementally, first the distance to the line at origin must be calculated:

$$f(0, 0) = -\vec{p}_0 \cdot \vec{n} = c$$

Next the increments for a step in the x direction and a step in the y direction must be calculated:

$$f(\vec{p} + \vec{e}_x) = (\vec{p} + \vec{e}_x - \vec{p}_0) \cdot \vec{n} = f(\vec{p}) + \vec{e}_x \cdot \vec{n} = f(\vec{p}) + a$$

$$f(\vec{p} + \vec{e}_y) = (\vec{p} + \vec{e}_y - \vec{p}_0) \cdot \vec{n} = f(\vec{p}) + \vec{e}_y \cdot \vec{n} = f(\vec{p}) + b$$

with $\vec{e}_x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \vec{e}_y = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Consequently, the new distance can be calculated from the old distance with the increments a and b. A step in the x direction changes the distance by a, and a step in the y direction changes the distance by b. The distance of the origin to the line is c.

With this information, the entire bounding box can be scanned. If the bounding box top left corner is not at the origin, an offset must be added.

(2) Limiter operation

The 2D Drawing Engine contains six limiters. Each limiter contains three registers:

- LnSTART
- LnXADD
- LnYADD.

See Figure 56.8 for details. It is possible to drive the limiters in a threshold mode, in which all values above 0.5 are set to 1, and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not wanted.

Note: In Figure 56.8, the following abbreviations are used:
start = LnSTART
xadd = LnXADD
yadd = LnYADD

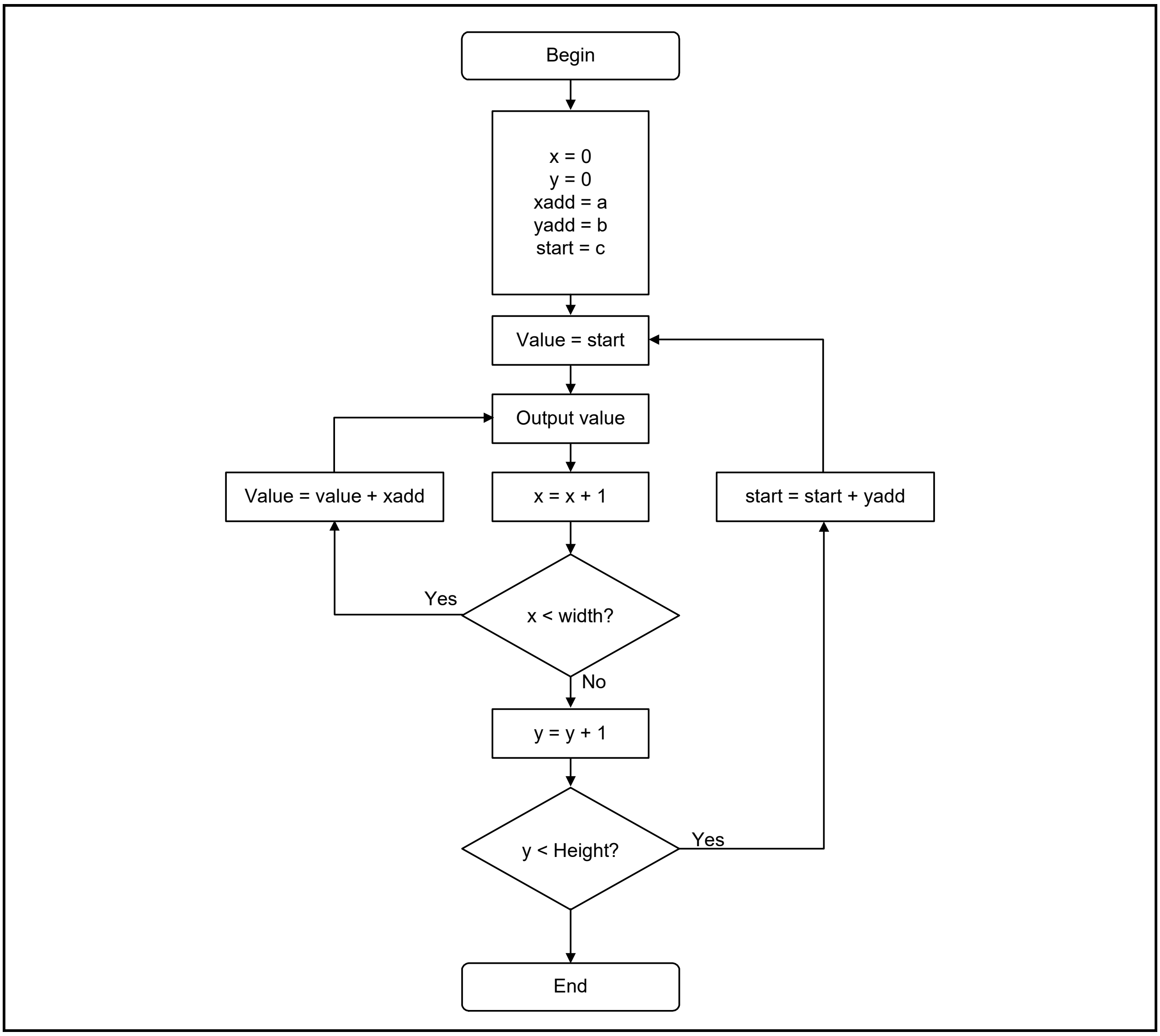


Figure 56.8 Operation flow of the linear limiter

(3) Example

If a straight line is given by the points P0 and P1, then the values are calculated as follows:

$$\Delta\vec{p} = \vec{p}_1 - \vec{p}_0 = \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

The normal vector (perpendicular but not unit size) is then:

$$\vec{n} = \begin{pmatrix} -\Delta y \\ \Delta x \end{pmatrix}$$

The not normalized distance between edge and origin is then:

$$\vec{p}_0 \cdot \vec{n} = -x_0\Delta y + y_0\Delta x$$

The limiter parameters would be:

$$\text{start} = -x_0\Delta y + y_0\Delta x$$

$$\text{xadd} = -\Delta y$$

$$\text{yadd} = \Delta x$$

In the normalized case, the normal vector is:

$$\vec{n} = \begin{pmatrix} -\Delta y \\ \Delta x \end{pmatrix} / (\sqrt{\Delta x^2 + \Delta y^2})$$

The distance between edge and origin is:

$$\vec{p}_0 \cdot \vec{n} = (-x_0 \cdot \Delta y + y_0 \cdot \Delta x) / (\sqrt{\Delta x^2 + \Delta y^2})$$

The limiter parameters would be:

$$\text{start} = (-x_0 \Delta y + y_0 \Delta x) / (\sqrt{\Delta x^2 + \Delta y^2})$$

$$\text{xadd} = -\Delta y / (\sqrt{\Delta x^2 + \Delta y^2})$$

$$\text{yadd} = \Delta x / (\sqrt{\Delta x^2 + \Delta y^2})$$

Normalization is only required if anti-aliasing is used. The driver contains an optimized inverse square root function to speed up the normalization process.

56.6.2.2 Edge setup quadratic case

(1) Mathematical background

It is also possible to set up the limiters to incrementally calculate the following equation:

$$f(x, y) = ax^2 + by^2 + cx + dy + f$$

At the origin, the value is:

$$f(0, 0) = f$$

The step in the x direction is:

$$f(x+1, y)$$

$$= a(x+1)^2 + by^2 + c(x+1) + dy + f$$

$$= ax^2 + 2ax + a + by^2 + cx + c + dy + f$$

$$= f(x, y) + 2ax + c + a$$

$$dx(x) = f(x+1, y) - f(x) = 2ax + c + a$$

The step in the y direction is:

$$f(x, y+1)$$

$$= ax^2 + b(y+1)^2 + cx + d(y+1) + f$$

$$= ax^2 + by^2 + 2by + b + cx + d(y+1) + f$$

$$= f(x, y) + 2bx + b$$

$$dy(y) = f(x+1, y) - f(x) = 2by + d + b$$

In the quadratic case, the increments dx and dy depend on x and y and are not constant. They can be calculated incrementally:

$$d^2x = dx(x+1) - dx(x) = 2a(x+1) + c + 1 - (2ax + c + 1) = 2a$$

$$d^2y = dy(y+1) - dy(y) = 2b(y+1) + d + 1 - (2by + d + 1) = 2b$$

At the origin, the increments are:

$$dx(0) = c + 1 \text{ and } dy(0) = d + 1$$

By incrementing the value by dx and dy for every step in the x and y direction and incrementing dx, dy by d^2x, and d^2y for every step in the x and y direction, the quadratic equation can be easily calculated for the whole bounding box.

(2) Limiter operation

In the quadratic case, two linear limiters are combined to operate as one quadratic limiter, called limiter 1 and limiter 2. The registers are:

- L1START, L1XADD, L1YADD
- L2START, L2XADD, L2YADD.

See Figure 56.9 for details. The gray box is an addition that performs a different operation, as in the linear setup. It is possible to drive the limiters in a threshold mode, in which all values above 0.5 are set to 1, and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not wanted.

Note: In Figure 56.9, the following abbreviations are used:

- start1 = L1START, start2 = L2START
- xadd1 = L1XADD, xadd2 = L2XADD
- yadd1 = L1YADD, yadd2 = L2YADD.

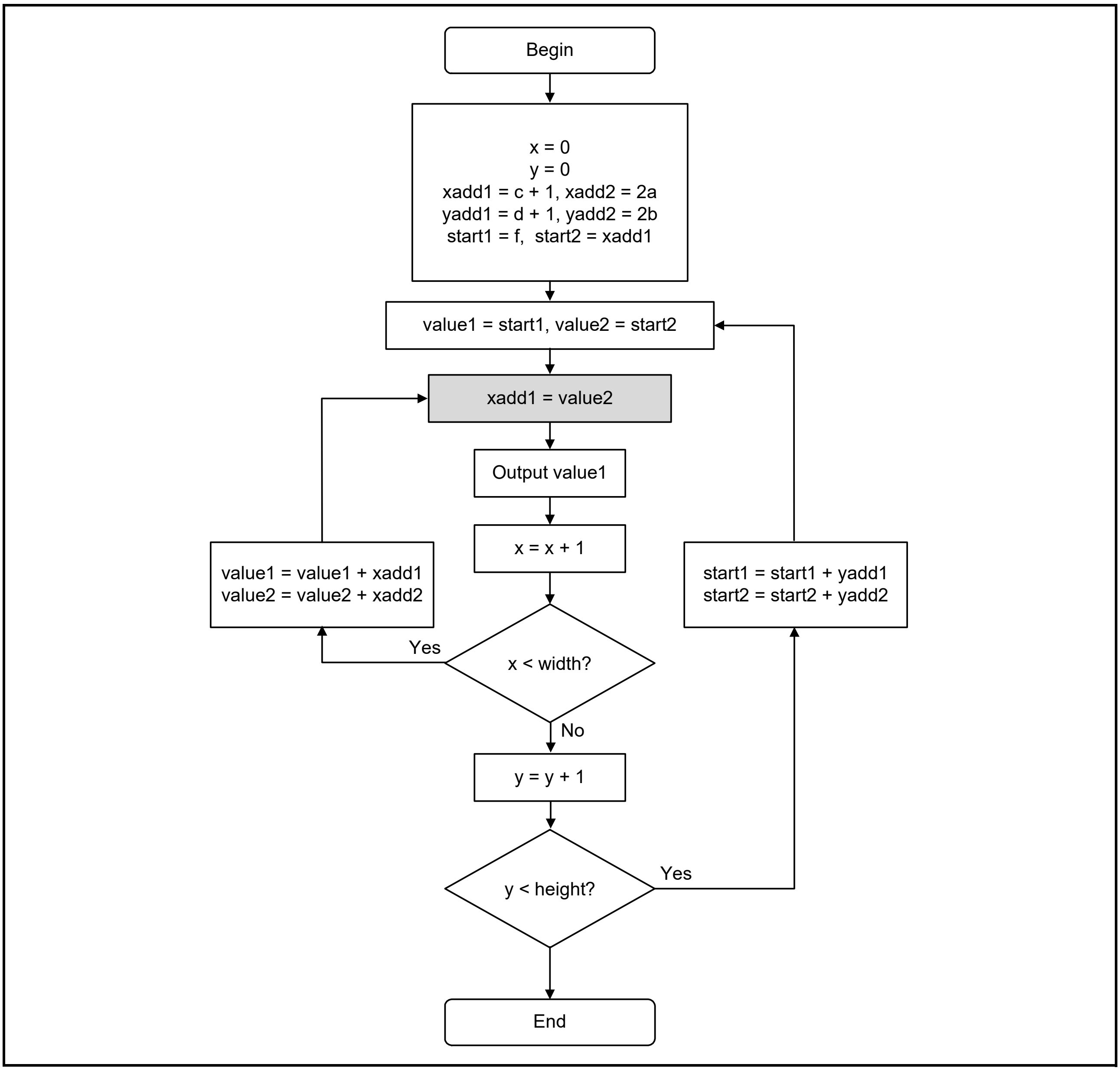


Figure 56.9 Operation flow of the quadratic limiter

(3) Example

Consider the equation for a circle with the center at $\vec{c} = \begin{pmatrix} s \\ t \end{pmatrix}$ and radius r:

$$0 = f(x, y) = (x-s)^2 + (y-t)^2 - r^2$$

This equation can be rewritten as:

$$f(x, y) = x^2 - 2xs + s^2 + y^2 - 2yt + t^2 - r^2$$

This can be sorted to fit to the original equation:

$$f(x, y) = x^2 + y^2 - 2sx - 2ty + (s^2 + t^2 - r^2)$$

With the following assignments, the circle equation can be calculated incrementally:

$a = 1$

$b = 1$

$c = -2s$

$d = -2t$

$f = s^2 + t^2 - r^2$

For the limiters with the results calculated in (1) Mathematical background, this would result in:

$\text{start1} = f = s^2 + t^2 - r^2$

$\text{xadd1} = c + 1 = -2s + 1$

$\text{yadd1} = d + 1 = -2t + 1$

$\text{start2} = \text{xadd1}$

$\text{xadd2} = 2a = 2$

$\text{yadd2} = 2b = 2$

56.6.2.3 Band filter

The output of limiter 1 and 2 can be modified to use a band filter. The band filter has a single filter parameter w.

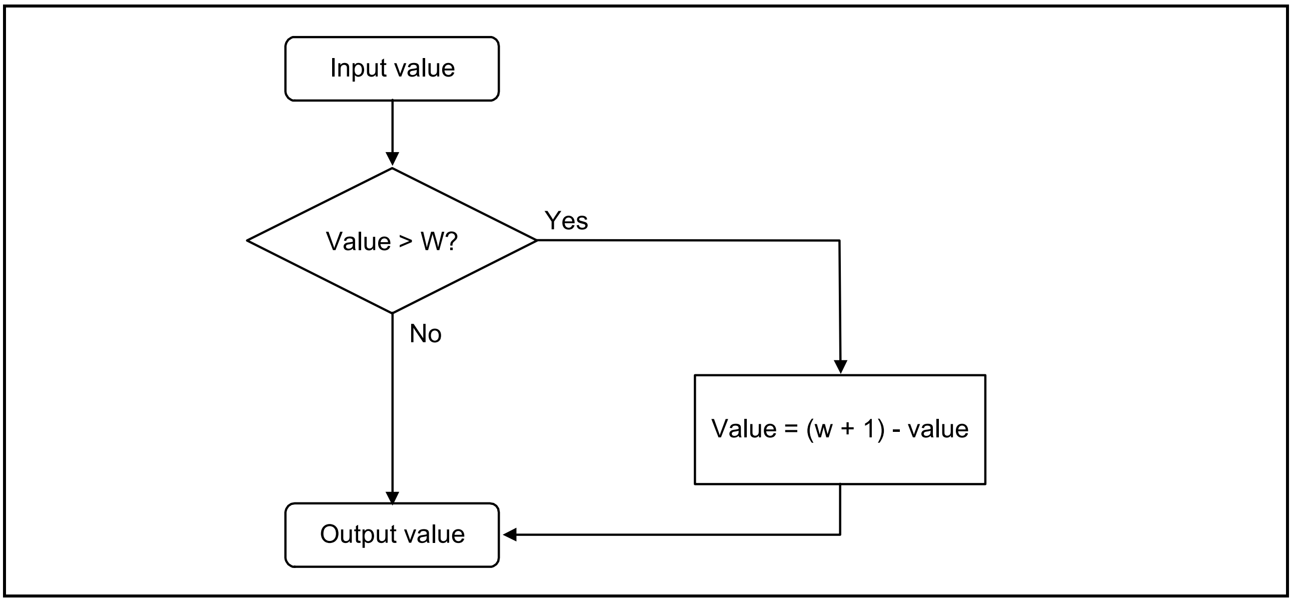


Figure 56.10 Band filter

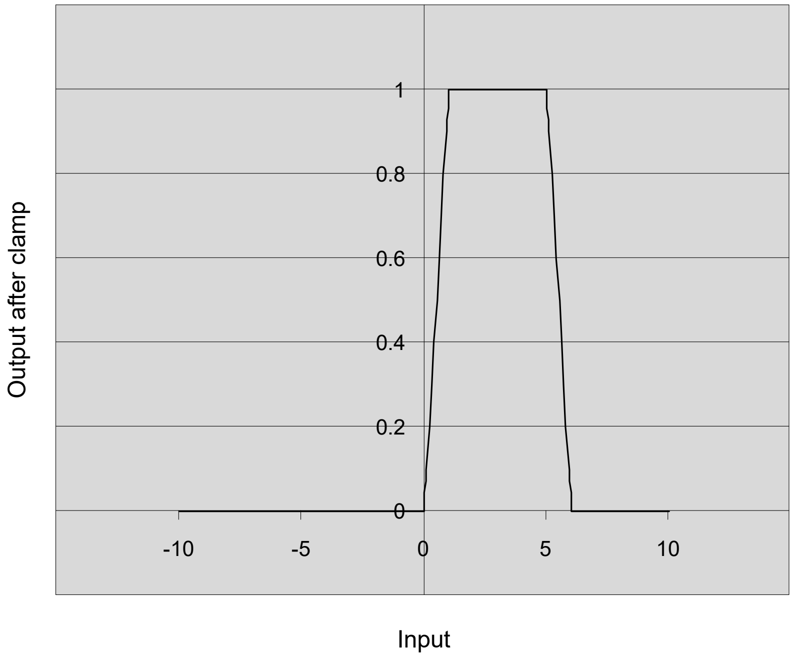


Figure 56.11 Band filter output after clamp with w = 7

56.6.2.4 Clamping unit

The clamping unit cuts the limiter output to the interval [0:1].

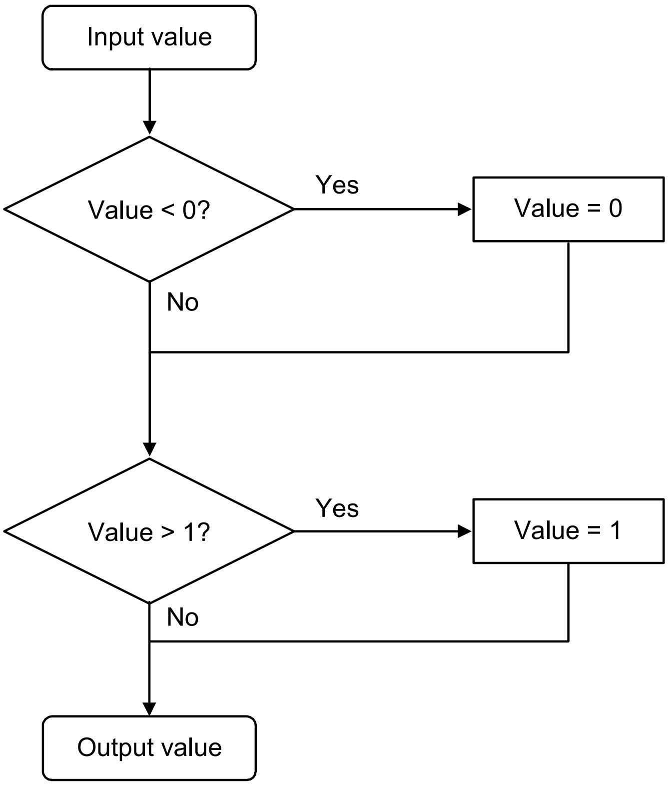


Figure 56.12 Clamping unit

The clamping unit can be put into threshold mode, in which all values greater than 0.5 are set to 1, and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not wanted, such as for shared edges.

56.6.2.5 Combiner unit

The combiner unit can be operated in minimum mode and in maximum mode. In minimum mode the smaller value is output, and in maximum mode the larger value is output. The minimum mode represents the intersection, and the maximum mode represents the union of the two regions.

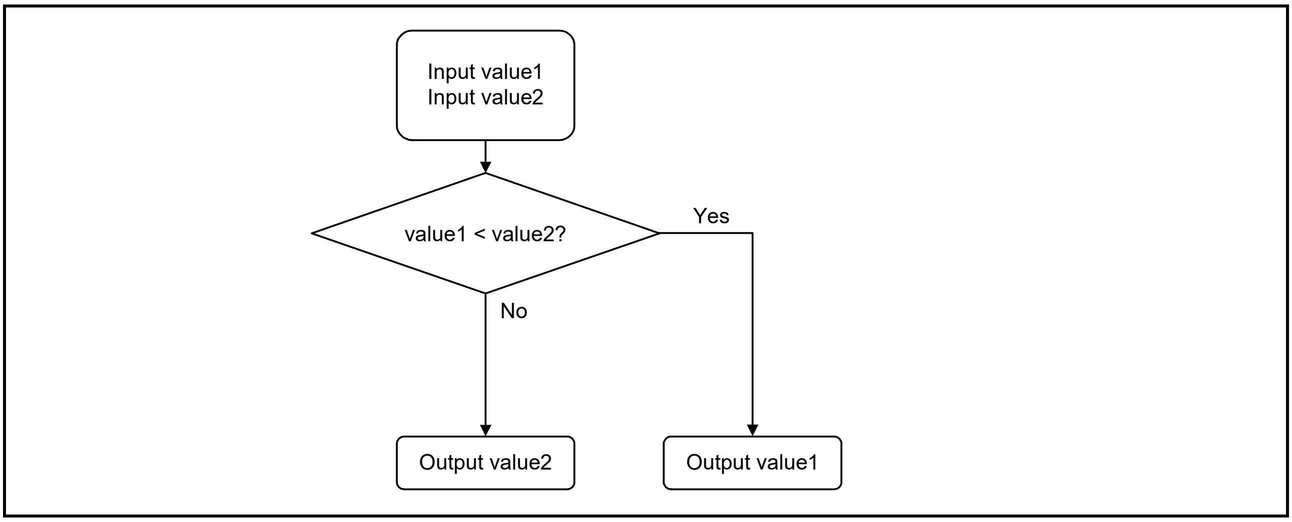


Figure 56.13 Combiner operated in minimum mode with intersection

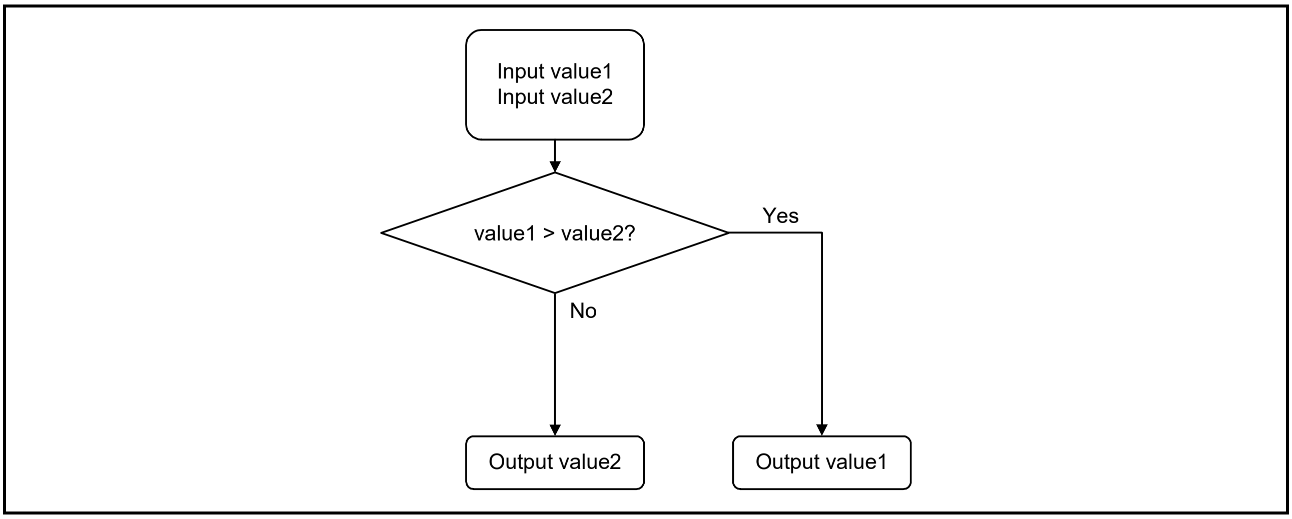


Figure 56.14 Combiner operated in maximum mode with union

56.6.2.6 Rasterization optimization

During rasterization, it is necessary to step through the whole bounding box one pixel at a time. This requirement can lead to an unnecessary number of steps for pixels that are not drawn. The 2D Drawing Engine provides optimization methods designed to reduce the number of steps required during rasterization. One optimization relies on the fact that any convex primitive can have only one span per line (a span is a contiguous horizontal line). This form of optimization detects a span start and saves the information for the next line. Another optimization is to detect a span end and stop rasterization for the current line.

(1) Spanstore

Consider the case in Figure 56.15.

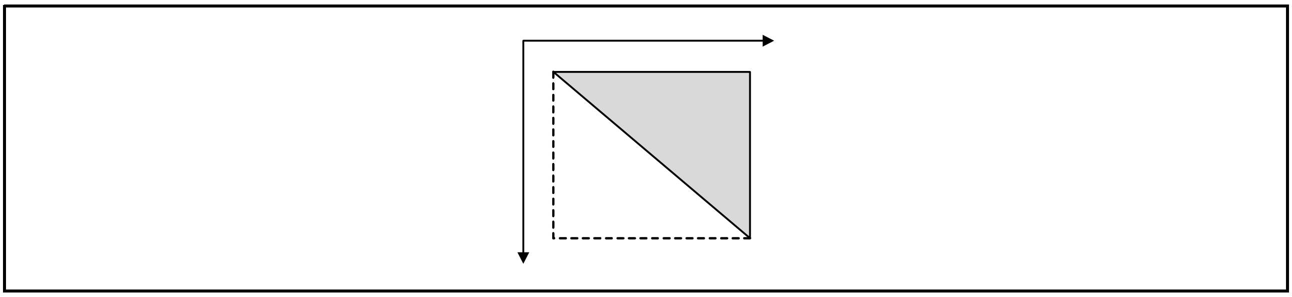


Figure 56.15 Triangle where the first edge is monotone growing

If the gray triangle must be rendered, half of the pixels processed by the rendering engine in the dotted bounding box would not be drawn. This situation can be optimized with the spanstore operation. When spanstore is activated and a span start is detected, the x position of the span start is detected.

In the next line the rendering starts with the stored x position. This only works if the edge is monotonically increasing ($y_1 > y_2 => x_1 \geq x_2$).

Consider the case in Figure 56.16.

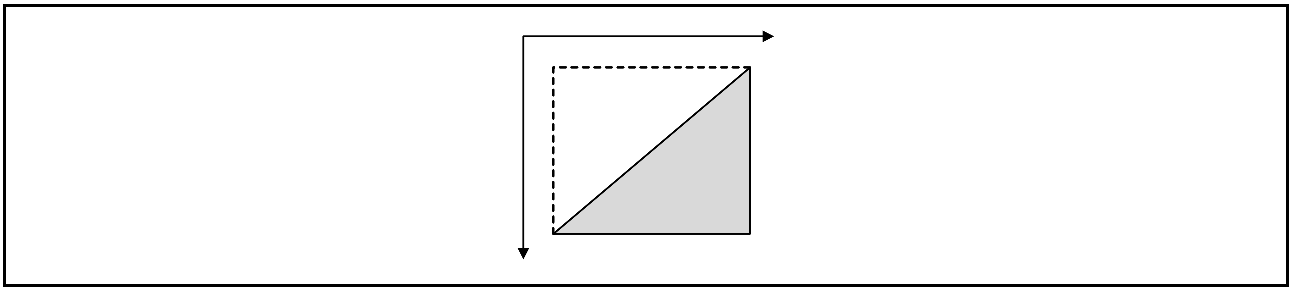


Figure 56.16 Triangle where the first edge is monotone falling

In this case, the normal spanstore operation cannot be performed. For this, the y direction of the rendering is reversed, and spanstore can operate again.

Consider the last case in Figure 56.17.

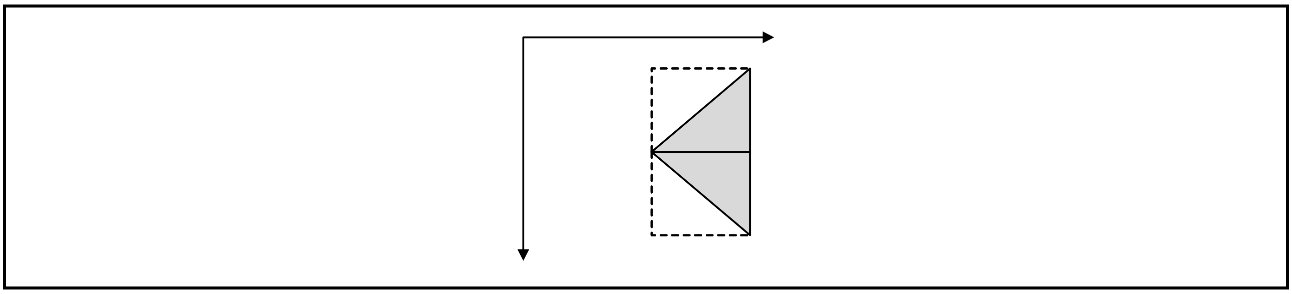


Figure 56.17 Triangle where the first edge is first monotone falling and then monotone growing

In this case, the triangle must be split and rendered as two parts for the spanstore optimization to work.

It is also possible to delay spanstore activation for a number of lines. This approach is used for rasterizing circles, as shown in Figure 56.18.

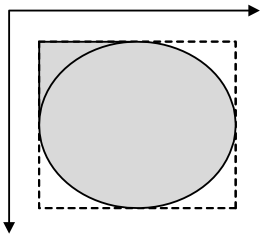


Figure 56.18 Full circle where the first edge is monotone falling for the first half and monotone growing for the second half

In this case, spanstore cannot be activated in the top left corner but can be activated in the bottom left corner. The empty corners in the top right and the bottom right cannot be rasterized because of the spanabort optimization.

(2) Spanabort

The second optimization assumes that the object that must be drawn is convex, which means there is only one span per scan line to be drawn. A non-convex object includes an object such as a triangle that is not filled and only consists of a thick border.

For a convex object, the rasterization can be stopped when the end of the first span is detected. No other constraints apply to this optimization for convex objects.

(3) Optimization efficiency

The efficiency of the optimizations can be seen for a typical case in Figure 56.19. In this case, the triangle is always rendered as single piece and is not separated into multiple triangles for higher optimizations. For this, the spanstore delay is used.

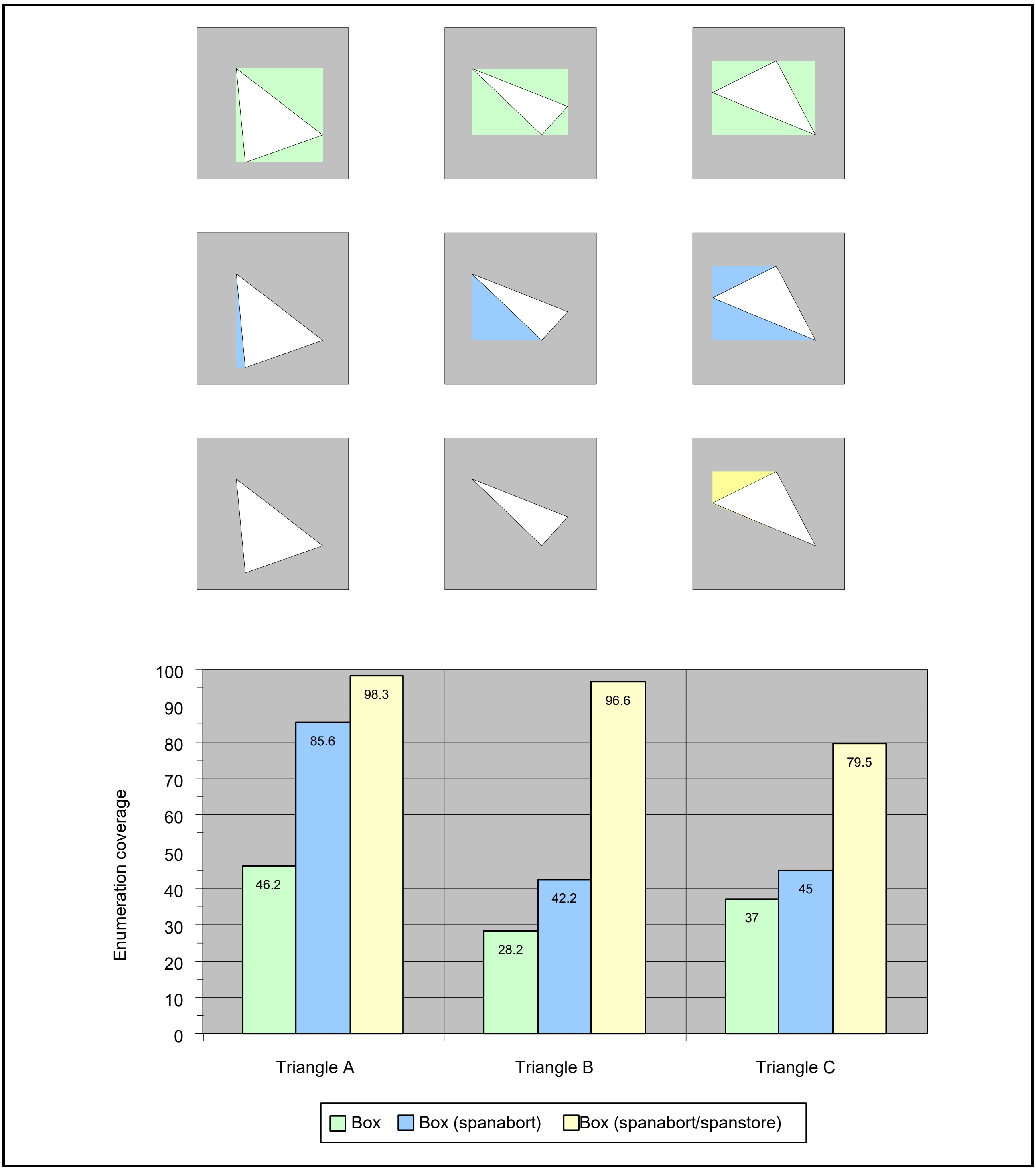


Figure 56.19 Efficiency of spanstore and spanabort optimizations with enumeration coverage equal to {pixels of primitive/pixels of bounding box}

56.6.3 Texturing

The texture unit can cover any primitive with a picture. The picture can be stretched, sheared, rotated, and translated in one step. To avoid aliasing, the result can be filtered bilinear in the u and v directions.

56.6.3.1 Mathematical background

The arbitrary mapping problem is completely determined by a mapping from 3 points in the object space (x, y) to 3 points in the texture space (u, v).

Consider the following mapping:

$$\vec{p}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \Rightarrow (\tilde{\vec{p}}_0) = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\vec{p}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \Rightarrow (\tilde{\vec{p}}_1) = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = \begin{pmatrix} w \\ 0 \end{pmatrix}$$

$$\vec{p}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \Rightarrow (\vec{p}_2) = \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ h \end{pmatrix}$$

where w is the width of the texture and h is the height of the texture.

Examine Figure 56.20 in the object space. To simplify calculations the difference vectors are taken as calculations:

$$\vec{d}_1 = \vec{p}_1 - \vec{p}_0$$

$$\vec{d}_2 = \vec{p}_2 - \vec{p}_0$$

This is equivalent to transforming from coordinate system O to coordinate system O'.

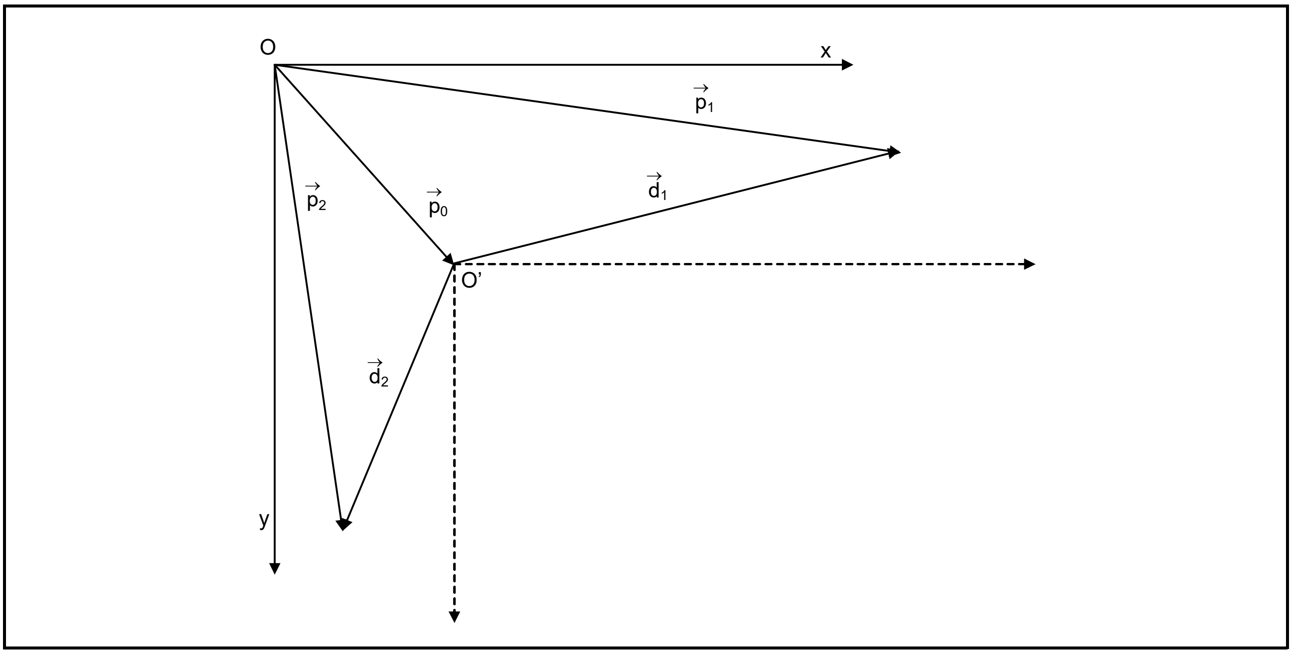


Figure 56.20 **Texture mapping, object space, and transformation from coordinate system O to O' to simplify calculations**

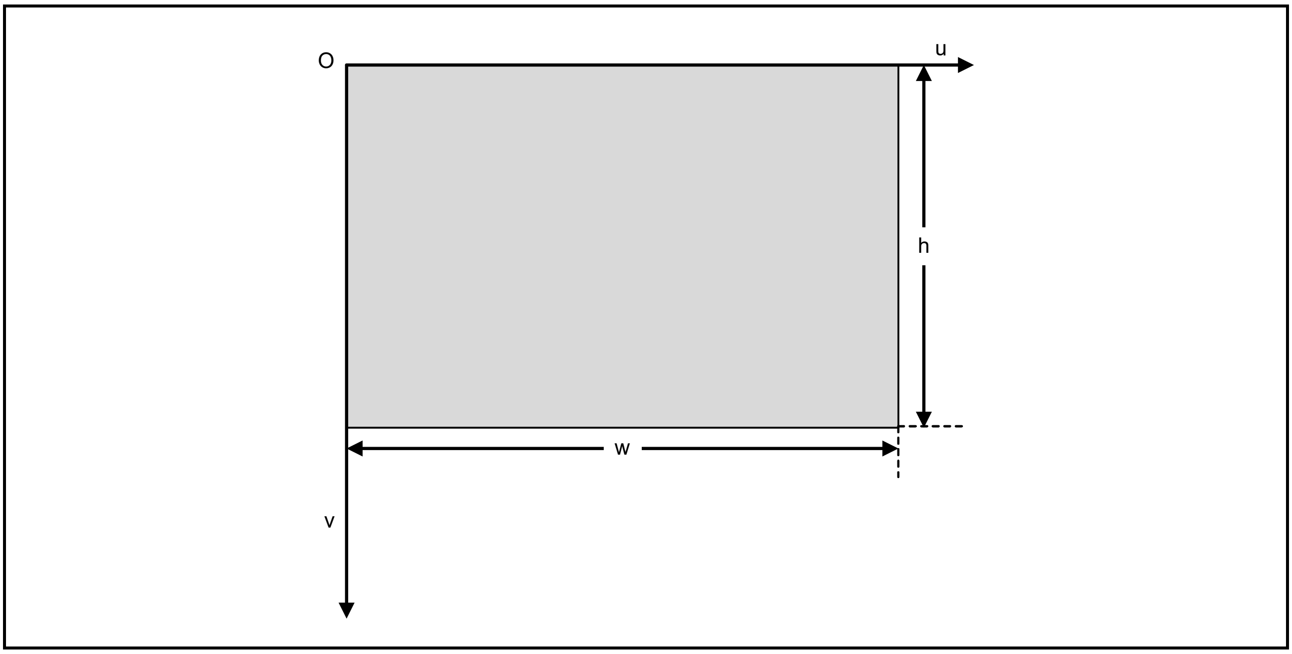


Figure 56.21 Texture mapping, texture space, and texture with width w and height h

In O’, the mapping is:

$$\vec{p}'_0 = \begin{pmatrix}0\\0\end{pmatrix} \Rightarrow (\tilde{\vec{p}}_0) = \begin{pmatrix}u_0\\v_0\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$$

$$\vec{d}_1 = \begin{pmatrix}dx_1\\dy_1\end{pmatrix} \Rightarrow (\tilde{\vec{p}}_1) = \begin{pmatrix}u_1\\v_1\end{pmatrix} = \begin{pmatrix}w\\0\end{pmatrix}$$

$$\vec{d}_2 = \begin{pmatrix}dx_2\\dy_2\end{pmatrix} \Rightarrow (\tilde{\vec{p}}_2) = \begin{pmatrix}u_2\\v_2\end{pmatrix} = \begin{pmatrix}w\\h\end{pmatrix}$$

This is a linear mapping that can be described by a 2 x 2 matrix.

$$(\tilde{\vec{p}}) = M \cdot \vec{p}' = \begin{bmatrix}m_{11} & m_{12}\\m_{21} & m_{22}\end{bmatrix} \cdot \vec{p}'$$

$$\Rightarrow \begin{pmatrix}w\\0\end{pmatrix} = M \cdot \vec{d}_1 \wedge \begin{pmatrix}0\\h\end{pmatrix} = M \cdot \vec{d}_2$$

If the equations are expanded and sorted, the result is two equation systems each with two unknowns. These can be described more easily with a new matrix.

Let $A = \begin{bmatrix}dx_1 & dy_1\\dx_2 & dy_2\end{bmatrix}$ then the equation system can be rewritten as:

$$\begin{pmatrix}w\\0\end{pmatrix} = A \cdot \begin{pmatrix}m_{11}\\m_{12}\end{pmatrix} \wedge \begin{pmatrix}0\\h\end{pmatrix} = A \cdot \begin{pmatrix}m_{21}\\m_{22}\end{pmatrix}$$

This can be easily solved with determinants.

Let $c = \dfrac{1}{\det A} = \dfrac{1}{dx_1 \cdot dy_2 - dx_2 \cdot dy_1}$

The resulting constants are:

$$m_{11} = c \cdot w \cdot dy_2 = \frac{du}{dx}$$

$$m_{12} = -c \cdot w \cdot dx_2 = \frac{du}{dy}$$

$$m_{21} = c \cdot h \cdot dx_1 = \frac{dv}{dx}$$

$$m_{22} = -c \cdot h \cdot dx_2 = \frac{dv}{dy}$$

To calculate the start values for u and v at the top of the bounding box, the transformation from O' to O must be reversed. Let us and vs be the start values, then:

$$\begin{pmatrix} u_s \\ v_s \end{pmatrix} = M \cdot (-\vec{p}_0) = c \cdot \begin{pmatrix} -w \cdot (x_0 \cdot dy_2 - y_0 \cdot dx_2) \\ h \cdot (x_0 \cdot dy_1 - y_0 \cdot dx_1) \end{pmatrix}$$

Examples

U and v are in the texture space. Enter the following into any case:

- Copy case:
  dx1 = 1, dx2 = 0, dy1 = 0, dy1 = 1
- Scaling case x scaling copy case:
  dx1 = f, dx2 = 0, dy1 = 0, dy1 = 1
  with f being the scaling factor in the x direction similar for the y direction
- Rotation case:
  dx1 = cosa, dx2 = -sina, dy1 = sina, dy1 = cosa
  with the angle between d1 and the x axis in the clockwise direction.

56.6.3.2 Limiter operation

The texture limiters operate exactly the same as the spatial limiters shown in Figure 56.8, Operation flow of the linear limiter.

The register layout for the u limiter is the same:

- LUSTART = us
- LUXADD = du/dx
- LUYADD = du/dy.

The register layout for the v limiter is slightly different. TEXPITCH is multiplied to save one hardware multiplier.

- LVSTARTI = floor (vs) • TEXPITCH
  Contains the integer part of the start value.
- LVSTARTF = (vs-floor (vs)) • TEXPITCH
  Contains the fractional part of the start value.
- LVXADDI = floor (dv/dx) • TEXPITCH
  Contains the integer part of dv/dz.
- LVYADD = floor (dv/dy) • TEXPITCH
  Contains the integer part of dv/dy.
- LVYXADDF = (dv/dy - floor (dv/dy)) • TEXPITCH ((dv/dx - floor (dv/dx)) • TEXPITCH)
  Contains the fractional part of dv/dy and dv/dx combined in one register.
- TEXMASK
  Contains a mask for u and v separately to wraparound values of u and v. This is useful for staying inside the limits of the texture or repeat a texture. Wrap around textures have to have a size multiple of 2.
- TEXPITCH
  Contains the width of the texture in pixels in framebuffer memory. This information is required to calculate the new address if stepping to a new line.
- TEXORIGIN.
  Contains the base address of the texture.

56.6.4 Colorization

After a pixel is found to be part of the geometry, its color is calculated. The 2D Drawing Engine supports a very general color calculation scheme, allowing support for several color modes. This color scheme uses an interpolation between two color registers, COLOR1 and COLOR2. See Figure 56.22 for details. COLOR1 and COLOR2 are marked as A, B in the figure for clarity.

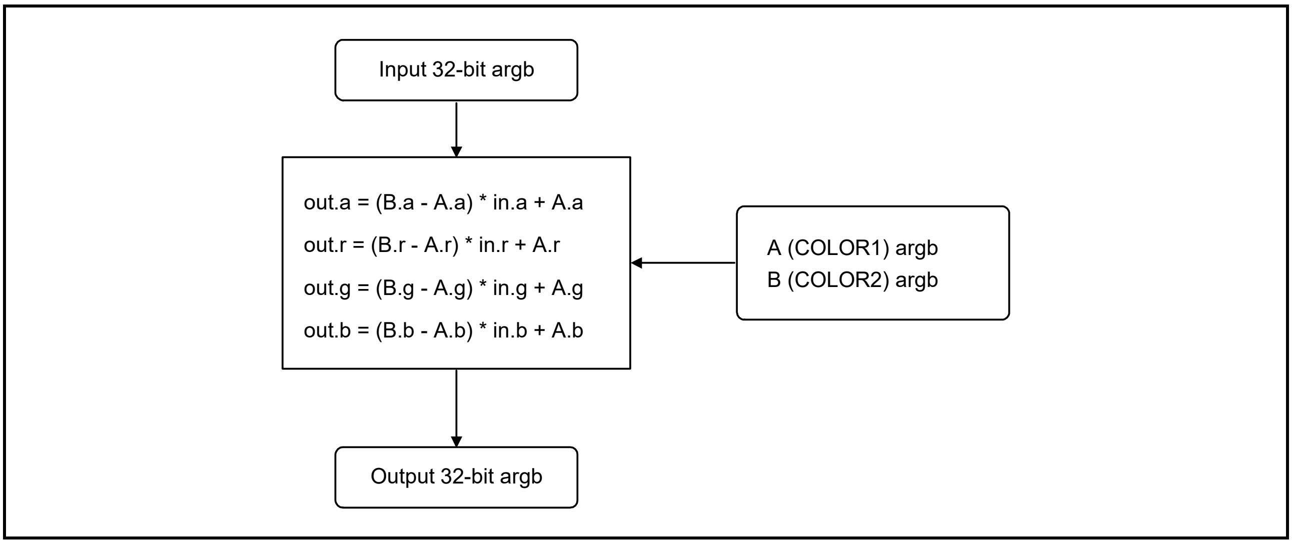


Figure 56.22 Colorization step and interpolation between the two color registers, A (COLOR1) and B (COLOR2)

This general approach can be used to support several different color modes that can be individually applied to any color or alpha channel of the input.

Table 56.7 Colorization operations

Operation	**Settings for A and B** [*1]
Copy	A = 0, B = 0xff
Replace with a constant value v	A = v, B = v
Multiply by a constant value v	A = 0, B = v
Colorize an alpha texture with the RGB value v	A.a = 0, A.r = B.r, A.g = B.g, A.b = B.b, B.a = 0xff,B.r = v.r, B.g = v.g, B.b = v.b
Invert a channel	A = 0xff, B = 0
Invert multiply with v	A = v, B = 0
Interpolate between color v and color u	A = v, B = u

Note 1. A = COLOR1, B = COLOR2

56.6.5 Blending

56.6.5.1 Color channel blending

The last step before the pixel is written to the framebuffer is to blend the pixel with the data that is already written to the framebuffer. If blending is activated, the framebuffer, referred to as DST, must be read. SRC is the output from the colorization unit.

The following color blend modes are supported:

- SRC_ZERO
- SRC_ONE
- SRC_ALPHA

- SRC_ONE_MINUS_ALPHA
- DST_ZERO
- DST_ONE
- DST_ALPHA
- DST_ONE_MINUS_ALPHA.

The selection of the color channel blend modes is performed with the following flags:

- BSF: blend source factor is alpha
- BSI: blend source factor invert
- BDF: blend destination factor is alpha
- BDI: blend destination factor invert.

The formula for the blending is:

$\mathrm{dst} = \mathrm{src} \cdot f_S + \mathrm{dst} \cdot f_D$

where:

$BSF = 0, BSI = 0 \Rightarrow f_S = 1$

$BSF = 1, BSI = 0 \Rightarrow f_S = \alpha$

$BSF = 0, BSI = 1 \Rightarrow f_S = 0$

$BSF = 1, BSI = 1 \Rightarrow f_S = 1 - \alpha$

$BDF = 0, BDI = 0 \Rightarrow f_D = 1$

$BDF = 1, BDI = 0 \Rightarrow f_D = \alpha$

$BDF = 0, BDI = 1 \Rightarrow f_D = 0$

$BDF = 1, BDI = 1 \Rightarrow f_D = 1 - \alpha$.

Table 56.8 lists all possible color channel blend modes.

Table 56.8 Color channel blend modes

Mode	BSF	BSI	BDF	BDI	Blend equation
SRC_ONE DST_ONE	0	0	0	0	SRC + DST
SRC_ONE	0	0	0	1	SRC
SRC_ONE DST_ALPHA	0	0	1	0	SRC + DST × ALPHA
SRC_ONE DST_ONE_MINUS_ALPHA	0	0	1	1	SRC + DST × (1 - ALPHA)
SRC_ZERO DST_ONE	0	1	0	0	DST
SRC_ZERO DST_ZERO	0	1	0	1	0
SRC_ZERO DST_ALPHA	0	1	1	0	DST × ALPHA
SRC_ZERO DST_ONE_MINUS_ALPHA	0	1	1	1	DST × (1 - ALPHA)
SRC_ALPHA DST_ONE	1	0	0	0	SRC × ALPHA + DST
SRC_ALPHA	1	0	0	1	SRC × ALPHA
SRC_ALPHA DST_ALPHA	1	0	1	0	SRC × ALPHA + DST × ALPHA
SRC_ALPHA DST_ONE_MINUS_ALPHA	1	0	1	1	SRC × ALPHA + DST × (1 - ALPHA)
SRC_ONE_MINUS_ALPHA DST_ONE	1	1	0	0	SRC × (1 - ALPHA) + DST
SRC_ONE_MINUS_ALPHA	1	1	0	1	SRC × (1 - ALPHA)
SRC_ONE_MINUS_ALPHA DST_ALPHA	1	1	1	0	SRC × (1 - ALPHA) + DST × ALPHA
SRC_ONE_MINUS_ALPHA DST_ONE_MINUS_ALPHA	1	1	1	1	SRC × (1 - ALPHA) + DST × (1 - ALPHA)

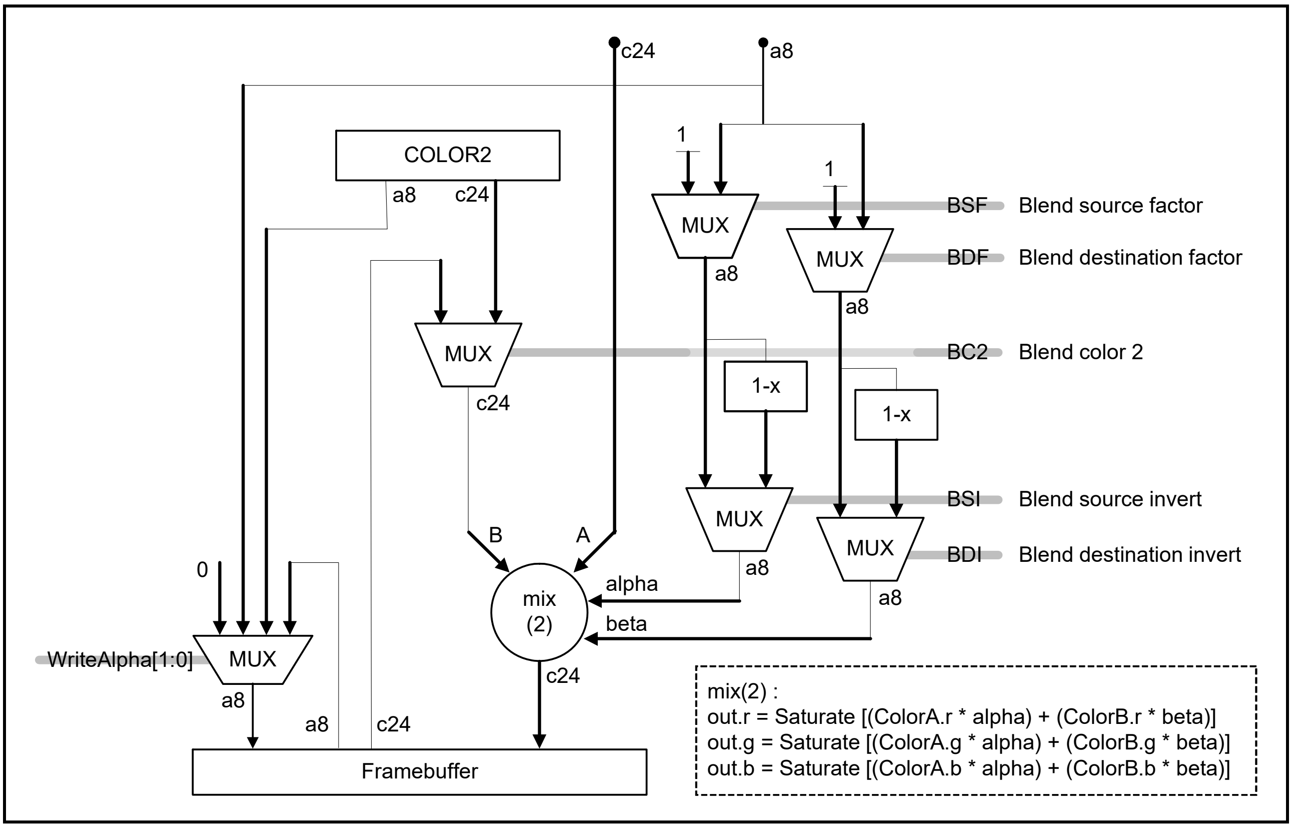


Figure 56.23 Color channel blend unit when CONTROL2.USEACB = 0

56.6.5.2 Alpha channel blending

The alpha channel can be blended in addition to the color channels. Alpha channel blending is enabled by setting CONTROL2.USEACB = 1. Alpha channel blending uses the same formulas and same blend modes as for the color channels. The alpha channel formulas can be set independently from the color channels.

The following alpha channel blend modes are supported:

- SRC_A_ZERO
- SRC_A_ONE
- SRC_A_SRC_A
- SRC_A_ONE_MINUS_SRC_A
- DST_A_ZERO
- DST_A_ONE
- DST_A_SRC_A
- DST_A_ONE_MINUS_SRC_A.

The alpha channel blend modes selected with the following flags:

- BSFA: blend source factor is SRC_A
- BSIA: blend source factor invert
- BDFA: blend destination factor is SRC_A
- BDIA: blend destination factor invert.

The formula for the blending is:

$$\text{dst\_alpha} = \text{src \_a} \cdot f_{S\_a} + \text{dst\_a} \cdot f_{D\_a}$$

where:

BSFA = 0, BSIA = 0 $\Rightarrow f_{S\_a} = 1$

BSFA = 1, BSIA = 0 $\Rightarrow f_{S\_a} = src\_a$

BSFA = 0, BSIA = 1 $\Rightarrow f_{S\_a} = 0$

BSFA = 1, BSIA = 1 $\Rightarrow f_{S\_a} = 1 - src\_a$

BDFA = 0, BDIA = 0 $\Rightarrow f_{D\_a} = 1$

BDFA = 1, BDIA = 0 $\Rightarrow f_{D\_a} = src\_a$

BDFA = 0, BDIA = 1 $\Rightarrow f_{D\_a} = 0$

BDFA = 1, BDIA = 1 $\Rightarrow f_{D\_a} = 1 - src\_a$.

Table 56.9 lists all possible alpha channel blend modes.

Table 56.9 Alpha channel blend modes

Mode	BSF	BSI	BDF	BDI	Blend equation
SRC_A_ONE DST_A_ONE	0	0	0	0	SRC_A + DST_A
SRC_A_ONE	0	0	0	1	SRC_A
SRC_A_ONE DST_A_SRC_A	0	0	1	0	SRC_A + DST_A × SRC_A
SRC_A_ONE DST_A_ONE_MINUS_SRC_A	0	0	1	1	SRC_A + DST_A × (1 - SRC_A)
SRC_A_ZERO DST_A_ONE	0	1	0	0	DST_A
SRC_A_ZERO DST_A_ZERO	0	1	0	1	0
SRC_A_ZERO DST_A_SRC_A	0	1	1	0	DST_A × SRC_A
SRC_A_ZERO DST_A_ONE_MINUS_SRC_A	0	1	1	1	DST_A × (1 - SRC_A)
SRC_A_SRC_A DST_A_ONE	1	0	0	0	SRC_A × SRC_A + DST_A
SRC_A_SRC_A	1	0	0	1	SRC_A × SRC_A
SRC_A_SRC_A DST_A_SRC_A	1	0	1	0	SRC_A × SRC_A + DST_A × SRC_A
SRC_A_SRC_A DST_A_ONE_MINUS_SRC_A	1	0	1	1	SRC_A × SRC_A + DST_A × (1 - SRC_A)
SRC_A_ONE_MINUS_SRC_A DST_A_ONE	1	1	0	0	SRC_A × (1 - SRC_A) + DST_A
SRC_A_ONE_MINUS_SRC_A	1	1	0	1	SRC_A × (1 - SRC_A)
SRC_A_ONE_MINUS_SRC_A DST_A_SRC_A	1	1	1	0	SRC_A × (1 - SRC_A) + DST_A × SRC_A
SRC_A_ONE_MINUS_SRC_A DST_A_ONE_MINUS_SRC_A	1	1	1	1	SRC_A × (1 - SRC_A) + DST_A × (1 - SRC_A)

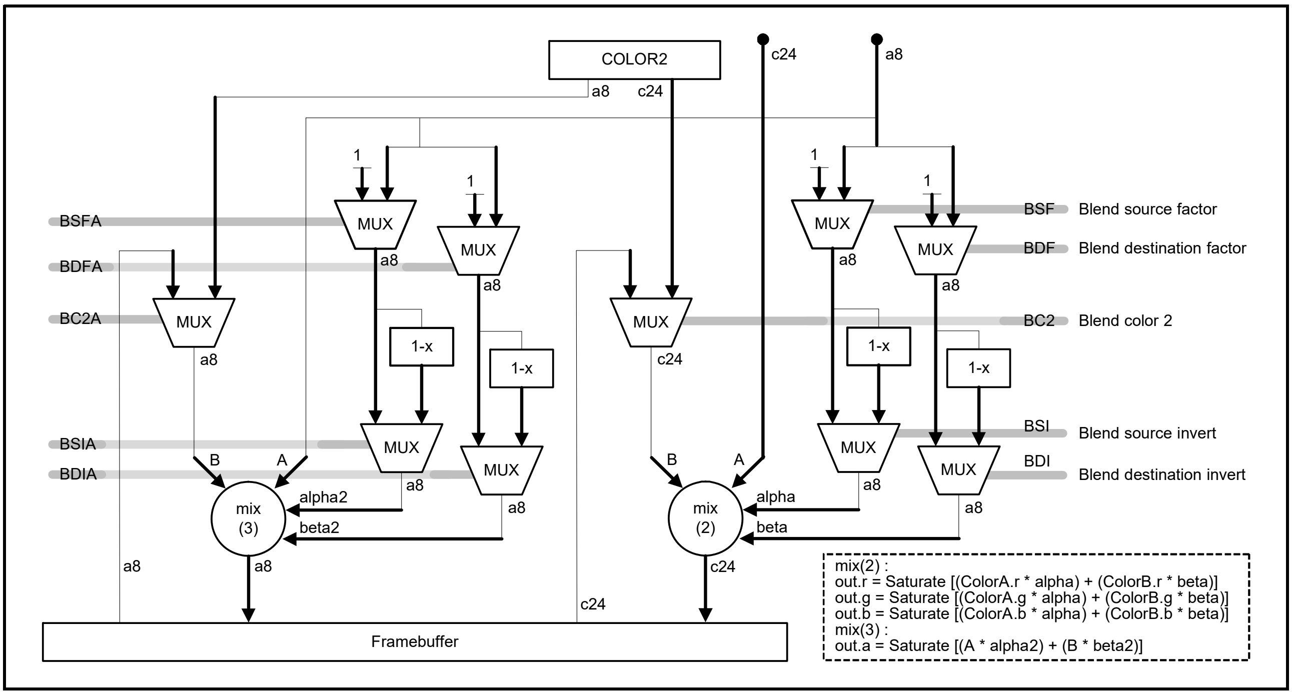


Figure 56.24 Alpha- and color-channel blend unit when CONTROL2.USEACB = 1

56.7 Rendering Modes

The rendering process can be performed in two different modes, register mode and display list mode.

56.7.1 Register Mode

In register mode, when operation is based on register settings, the host CPU configures and initiates each render process separately. To start a new render process, the host CPU must wait until the previous one is completed. In this mode, the host CPU is heavily engaged throughout the entire drawing procedure and is consequently to a large extent unavailable for other tasks.

The host CPU must set up all registers for performing a certain drawing operation before it can start the rendering process. A new register setup can only be started when the previous render process has completed. Before starting a new register setup, make sure that:

- STATUS.DLISTACTIVE = 0: Display list reader is idle
- STATUS.BUSENUM = 0: Pixel selection unit is idle.

Lastly, write the framebuffer start address to the ORIGIN register. This write triggers the 2D Drawing Engine to start rendering.

56.7.2 Display List Mode

In display list mode, the host CPU creates a display list in memory prior to starting the 2D Drawing Engine. Such a display contains a bundle of render operations. When started, the 2D Drawing Engine executes the display list autonomously in parallel with the host CPU, which is not involved with drawing operations most of the time. Use of a display list allows a fully asynchronous operation of the host CPU and the 2D Drawing Engine and offers the best possible system performance.

In this mode, the display list reader reads a memory block containing instructions on how to set the 2D Drawing Engine control registers and executes these control register writes accordingly.

Display list start

To start execution of a display list, which already resides in memory, the start address of the display list is written to the display list start address register DLISTSTART. Because rewriting DLISTSTART also stops any ongoing display list execution, make sure that the previous display list process is completed by either of the following two methods:

- Check that STATUS.DLISTACTIVE = 0, which indicates idle status of the display list reader
- Wait for the display list interrupt DRWDLISTIRQ, which indicates completion of the previous display list process.

Caution: Direct writing to 2D Drawing Engine registers while the display list mode is active (STATUS.DLISTACTIVE = 1) might lead to a 2D Drawing Engine hang-up. To prevent this, always check that the display list reader is idle ((STATUS.DLISTACTIVE = 0) before writing to any 2D Drawing Engine register.

Display list format

Display lists are stored using direct register-to-value mappings, which means that the display list contains a one byte index that addresses a certain register and the value to be written to the register. The register index is derived from the address offset of the register address and can be calculated by dividing the address offset by 4. For the index of each register, see section 56.2, Register Descriptions.

As the 2D Drawing Engine registers are always 32 bits wide, each data unit (called a data word) to be written to a register is of the same size. An address word that contains the indices of the register to be written is stored in a packed notation with up to four indices stored in one 32-bit address word.

A display list command always starts with an address word, followed by up to four data words, one for each register. The indices are read and interpreted from LSB to MSB, so the register of the low byte index is written first.

Example

In the following example:

DWORD 201A 1930h	// start of list address word
DWORD 0000 0013h	// data word 1 (for register 30h)
DWORD FFFF FFAAh	// data word 2 (for register 19h)
DWORD 4033 6480h	// data word 3 (for register 1Ah)
DWORD 0001 0000h	// data word 4 (for register 20h)
DWORD...	// next address word.

This stream of dwords updates the DRW registers as follows:

- Write 0000_0013h to register 30h = 48, which is IRQCTL
- Write FFFF_FFAAh to register 19h = 25, which is COLOR1
- Write 4033_6480h to register 1Ah = 26, which is COLOR2
- Write 0001_0000h to register 20h = 32, which is ORIGIN.

Address word indices

Besides referencing a register, the indices of an address word can also have other meanings, depending on their value.

Table 56.10 Address indices function (1 of 2)

Index	Function	
00h to 7Fh	Register indices Two register indices trigger additional actions:	
	- 20h = 32:	A write to ORIGIN to set a new frame buffer address is delayed until the ongoing frame buffer write-back is complete, when STATUS.BUSYWRITE = 0.
	- 32h = 50:	A write to DLISTSTART sets a new display list start address stops the current display list and starts the new one.
80h	Gap index, which is used to fill unused bytes of an address word. For example, if fewer than four indices are required, the remaining bytes are filled with 80h. In this case, the number of the subsequent data words must be reduced accordingly.	

Table 56.10 Address indices function (2 of 2)

Index	Function	
FFh	If the first index of an address word contains the special index FFh, the subsequent (second) index is interpreted as follows:	
	- Bit [0] set:	Display list end.
	- Bit [1] set:	Issue a full pipeline flush and wait (necessary before flip).
	- Bit [2] set:	Wait for writeback complete (necessary before framebuffer format change).
	- Bits [3:7]	Set all to 0s.
	Bit [1] and [2] settings are mutually exclusive. All indices after the special index FFh are ignored, and the next address word is read, if no display list end (bit [0] = 1) was set.	
	The remaining two indices must be set to 00h.	

Caution: Gap indices 80h must not be placed between other indices, for example as "index1 - 80h - index3 - index4". Always fill all indices after 80h with the gap index.

Caution: If any of the special indices 80h and FFh are used, no register index can follow after them in the same address word.

Table 56.11 2D Drawing Engine registers overview (1 of 2)

Register function	Symbol	Index
Control registers:		
Geometry control 0	CONTROL	0
Surface control	CONTROL2	1
Interrupt control	IRQCTL	48
Cache control	CACHECTL	49
Status control	STATUS	n.a.*1
Hardware version and feature set ID	HWREVISION	n.a.*1
Color registers:		
Base color	COLOR1	25
Secondary color	COLOR2	26
Pattern	PATTERN	29
Limiter registers:		
Limiter 1 start value	L1START	4
Limiter 2 start value	L2START	5
Limiter 3 start value	L3START	6
Limiter 4 start value	L4START	7
Limiter 5 start value	L5START	8
Limiter 6 start value	L6START	9
Limiter 1 x-axis increment	L1XADD	10
Limiter 2 x-axis increment	L2XADD	11
Limiter 3 x-axis increment	L3XADD	12
Limiter 4 x-axis increment	L4XADD	13
Limiter 5 x-axis increment	L5XADD	14
Limiter 6 x-axis increment	L6XADD	15
Limiter 1 y-axis increment	L1YADD	16
Limiter 2 y-axis increment	L2YADD	17
Limiter 3 y-axis increment	L3YADD	18
Limiter 4 y-axis increment	L4YADD	19
Limiter 5 y-axis increment	L5YADD	20
Limiter 6 y-axis increment	L6YADD	21
Limiter 1 band width parameter	L1BAND	22

Table 56.11 2D Drawing Engine registers overview (2 of 2)

Register function	Symbol	Index
Limiter 2 band width parameter	L2BAND	23
Texture registers:		
Texture base address	TEXORIGIN	47
Texels per texture line	TEXPITCH	45
Texture size or texture address mask	TEXMASK	46
U limiter start value	LUSTART	36
U limiter x-axis increment	LUXADD	37
U limiter y-axis increment	LUYADD	38
V limiter start value integer part	LVSTARTI	39
V limiter start value fractional part	LVSTARTF	40
V limiter x-axis increment integer part	LVXADDI	41
V limiter y-axis increment integer part	LVYADDI	42
V limiter increment fractional parts	LVYXADDF	43
Color lookup table start address	TEXCLADDR	55
Write Data to DRWTEXCLADDR; after each data write, DRWTEXCLADDR is incremented by 1.	TEXCLDATA	56
Offset to the index for the indexed texture formats i8, i4, i2, and i1	TEXCLOFFSET	57
Compare value for R,G, B components of internal texel color representation.	COLKEY	58
Miscellaneous registers:		
Bounding box dimension	SIZE	30
Framebuffer pitch and spanstore delay	PITCH	31
Address of the first pixel in framebuffer	ORIGIN	32
Display list start address	DLISTSTART	50
Performance counters control	PERFTRIGGER	53
Performance counter 1	PERFCOUNT1	51
Performance counter 2	PERFCOUNT2	52

Note 1. These registers are read-only and cannot be accessed in display list mode, and they therefore have no index.

56.7.3 Stopping the Render Process

Stopping an ongoing render process requires a specific procedure, which is described in section 56.10, Stopping the 2D Drawing Engine Render Process.

56.8 Interrupts

The 2D Drawing Engine generates three interrupts:

- DRWBUSIRQ
- DRWENUMIRQ
- DRWDLISTIRQ.

56.8.1 Interrupt sources

DRWBUSIRQ

This is the 2D Drawing Engine bus error interrupt. It occurs when the 2D Drawing Engine attempts to access an undefined address range through the following:

- Framebuffer Base Address Register ORIGIN
- Texture Base Address Register TEXORIGIN

- Display List Start Address Register (DLISTSTART).

The bus error interrupt DRWBUSIRQ is then generated. The bus error interrupt only serves for debugging purposes. The interrupt source can be determined by the BUSERRMFB, BUSERRMTXMRL, and BUSERRMDL bits of the STATUS register.

Note: After a DRWBUSIRQ occurrence, you must apply a system reset.

DRWENUMIRQ

This is the current render process finished interrupt.

DRWDLISTIRQ

This is the display list interrupt. It is asserted on completion of a display list process. DRWDLISTIRQ is activated if one of the following is true:

- The entire display list is complete
- The display list processing stops because a new display list start is triggered by a write to the Display List Start Address Register (DLISTSTART).

56.8.2 Interrupt Control

The three 2D Drawing Engine interrupts are combined into a single shared interrupt, DRW_IRQ, to the CPU. Each interrupt can be masked (disabled), or unmasked (enabled) by setting its associated enable bit in the Interrupt Control Register (IRQCTL).

The occurrence of an enabled interrupt (one with its mask bit set to 1 in IRQCTL) is monitored in the Status Control Register (STATUS) with its associated interrupt status bit is set to 1. The shared 2D Drawing Engine interrupt DRW_IRQ is then generated.

To clear the interrupt, the host CPU must write 1 to the interrupt clear bit in IRQCTL. The interrupt clear bit returns to 0 automatically.

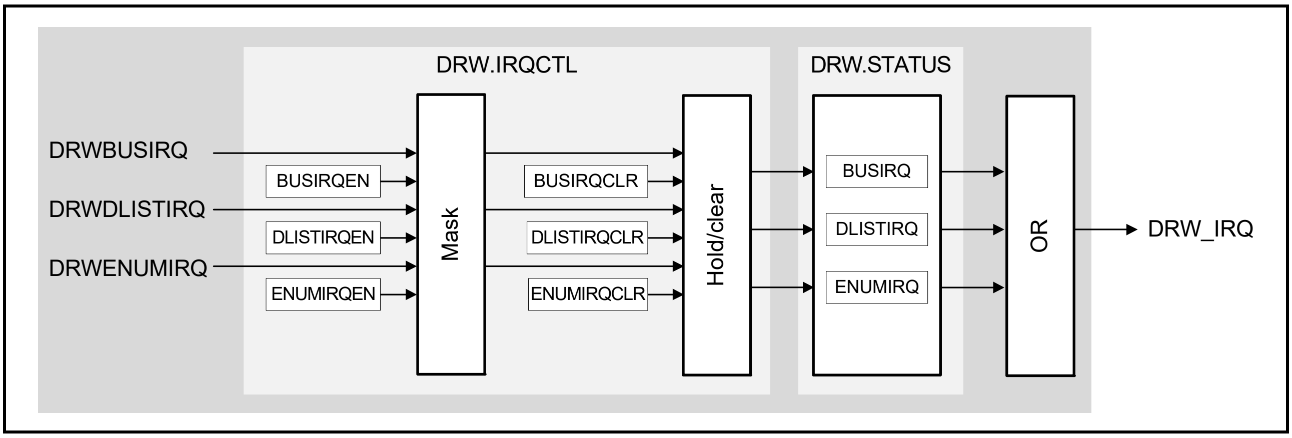


Figure 56.25 Interrupt Controller Unit (ICU)

56.9 Performance Counters

The 2D Drawing Engine features two independent 32-bit performance counter registers (PERFCOUNTk (k = 1, 2)) to count the number of occurrences of a certain event. The events to be counted can be set up independently for each performance counter register with the Performance Counter Control Registers, PERFTRIGGER.PERFTRIGGER2 for PERFCOUNT2 and PERFTRIGGER.PERFTRIGGER1 for PERFCOUNT1.

Table 56.12 lists the performance counter trigger events that can be selected.

Table 56.12 Performance counter trigger events

PERFTRIGGER.PERFTRIGGERk	Event
0	Disable performance counter
1	2D Drawing Engine active cycles
2	Framebuffer read access
3	Framebuffer write access
4	Texture read access
5	Invisible pixels (enumerated but selected with alpha 0%)
6	Invisible pixels while internal FIFO is empty (lost cycles)
7	Display list reader active cycles
8	Framebuffer read hits
9	Framebuffer read misses
10	Framebuffer write hits
11	Framebuffer write misses
12	Texture read hits
13	Texture read misses
31	Every clock cycle (for use as timer)

56.10 Stopping the 2D Drawing Engine Render Process

If a render process has started either in register or display list mode, the 2D Drawing Engine processes the data autonomously until the render process is finished. Depending on the rendering, this process might take several milliseconds.

If the 2D Drawing Engine is to be disabled because, for example, the MCU enters a low power mode, proceed as follows to stop the ongoing rendering:

1. Set the following registers as follows:

   SIZE = 0001 0001h
      Set bounding box dimensions to 1 pixel x 1 line.
   CONTROL2 = 0000 0000h
      Color format a (8), no texture, CLUT.
   ORIGIN = UnmappedAddress
      The UnmappedAddress is an address that is not available for 2D Drawing Engine access.
      The recommended UnmappedAddress is given here under the key word “UnmappedAddress”.
   Alternatively do the same in display list mode:

   ```
   DWORD 8020 011Eh             // start of “address word” list
   DWORD 0001 0001h             // SIZE = 0001 0001h
   DWORD 0000 0000h             // CONTROL2 = 0000 0000h
   DWORD UnmappedAddress        // ORIGIN = UnmappedAddress
   ```
2. Wait for the Bus Error corresponding to the unmapped address violation, which indicates access to an unmapped address and the stop of the render process.
   UnmappedAddress = FFFF FFF0h
3. Disable the 2D Drawing Engine, if wanted.

57. JPEG Codec (JPEG)

57.1 Overview

The JPEG Codec conforms to the JPEG baseline compression and decompression standard to provide high-speed compression of image data and high-speed decoding of JPEG data. Table 57.1 lists the JPEG Codec specifications and Figure 57.1 shows a block diagram.

Table 57.1 JPEG Codec specifications

Parameter	Specifications
Compliant standard	Complies with JPEG Baseline standard within the range described in this document. The JPEG Codec does not support the following features: • Scanning with two elements • Non-interleave scanning with multiple elements
Operational precision	Conforms to JPEG Part 2, ISO-IEC10918-2
Image input/output system	Block interleave method
Pixel format	• Compression: YCbCr422 (H = 2:1:1, V = 1:1:1) • Decompression: YCbCr444 (H = 1:1:1, V = 1:1:1), YCbCr422 (H = 2:1:1, V = 1:1:1), YCbCr411 (H = 4:1:1, V = 1:1:1), YCbCr420 (H = 2:1:1, V = 2:1:1) • Output pixel format to the buffer: ARGB8888, RGB565
Quantization table	• Four tables provided
Huffman table	• Four tables provided (two tables for AC coefficients and two tables for DC coefficients)
Markers supported	SOI (start of image), SOF0 (start of frame type 0), SOS (start of scan), DQT (define quantization tables), DHT (define Huffman tables), DRI (define restart interval), RSTm (restart marks), and EOI (end of image)
Processing unit	8-byte address boundary units can be set
Image sizes that can be processed	Sizes divisible by the minimum coded unit (MCU): 8 lines by 8 pixels in YCbCr444, 8 lines by 16 pixels in YCbCr422, 8 lines by 32 pixels in YCbCr411, 16 lines by 16 pixels in YCbCr420

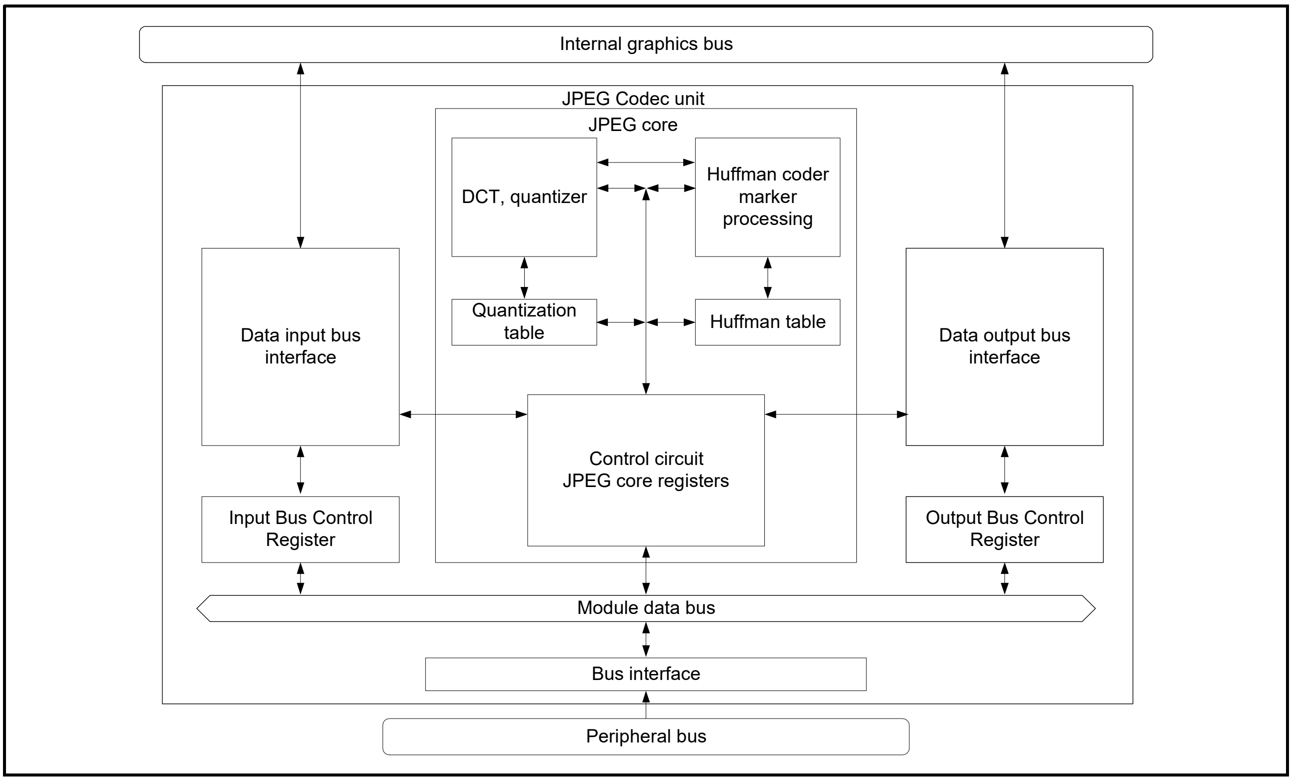


Figure 57.1 JPEG Codec block diagram

Note: When using the JPEG, set the clock to ICLK = PCLKA.

57.2 Register Descriptions

57.2.1 JPEG Code Mode Register (JCMOD)

Address(es): JPEG.JCMOD 400E 6000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	DSP	REDU[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	REDU[2:0]	Pixel Format	- With compression b2 b0 0 0 1: YCbCr422. Other settings are prohibited. - With decompression b2 b0 0 0 0: YCbCr444 0 0 1: YCbCr422 1 1 0: YCbCr411 0 1 0: YCbCr420. Other settings result in an error. The JPEG Codec cannot process them normally.	R/W*2
b3	DSP	Compression/Decompression Set*1	0: Select compression process 1: Select decompression process.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When changing between processing for compression and decompression, you must reset the JPEG Codec in advance by setting the BRST bit in JPEG Code Command Register (JCCMD).

Note 2. In decompression mode, these bits are read-only.

57.2.2 JPEG Code Command Register (JCCMD)

Address(es): JPEG.JCCMD 400E 6001h

	b7	b6	b5	b4	b3	b2	b1	b0
	BRST	—	—	—	—	JEND	JRST	JSRT
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	JSRT	JPEG Core Process Start Command	Set this bit to 1 to start JPEG core processing. Do not write this bit to 1 again while the JPEG Codec is in operation.	W
b1	JRST	JPEG Core Process Stop Clear Command	Set this bit to 1 to clear the process-stopped state caused by requests to read the image size and pixel format (enabled in the INT3 bit in JINTE0).	W*1
b2	JEND	Interrupt Request Clear Command	Set this bit to 1 to clear an interrupt request. This bit is only valid for the interrupt sources associated with the INS6, INS5, and INS3 bits in JINTS0.	W

Bit	Symbol	Bit name	Description	R/W
b6 to b3	—	Reserved	The write value should be 0.	W
b7	BRST	Bus Reset	Set this bit to 1 to reset the JCDTCU, JCDTCM, JCDTCD, JCDERR and JCRST registers. Do not set this bit to 1 while the JPEG Codec is in operation (from setting the JPEG core process start command to writing the last output code and image data). For the bus reset processing, see section 57.5, Bus Reset Processing.	W

Note: Clearing the bits in this register to 0 after setting a command is not required. Multiple commands must not be set simultaneously.

Note 1. In compression mode, this bit is invalid.

57.2.3 JPEG Code Quantization Table Number Register (JCQTN)

Address(es): JPEG.JCQTN 400E 6003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	QT3[1:0]		QT2[1:0]		QT1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	QT1[1:0]	Quantization Table Number for the First Color Component	b1 b0 0 0: Quantization table 0 (JCQTBL0) 0 1: Quantization table 1 (JCQTBL1) 1 0: Quantization table 2 (JCQTBL2) 1 1: Quantization table 3 (JCQTBL3).	R/W*1
b3, b2	QT2[1:0]	Quantization Table Number for the Second Color Component	b3 b2 0 0: Quantization table 0 (JCQTBL0) 0 1: Quantization table 1 (JCQTBL1) 1 0: Quantization table 2 (JCQTBL2) 1 1: Quantization table 3 (JCQTBL3).	R/W*1
b5, b4	QT3[1:0]	Quantization Table Number for the Third Color Component	b5 b4 0 0: Quantization table 0 (JCQTBL0) 0 1: Quantization table 1 (JCQTBL1) 1 0: Quantization table 2 (JCQTBL2) 1 1: Quantization table 3 (JCQTBL3).	R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In decompression mode, these bits are read-only.

57.2.4 JPEG Code Huffman Table Number Register (JCHTN)

Address(es): JPEG.JCHTN 400E 6004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	HTA3	HTD3	HTA2	HTD2	HTA1	HTD1
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	HTD1	Huffman Table Number (DC) for the First Color Component	b1 b0 0 0: DC/AC Huffman table 0 (JCHTBD0 and JCHTBA0) 1 1: DC/AC Huffman table 1 (JCHTBD1 and JCHTBA1). Other settings are prohibited.	R/W*1
b1	HTA1	Huffman Table Number (AC) for the First Color Component		R/W*1
b2	HTD2	Huffman Table Number (DC) for the Second Color Component	b3 b2 0 0: DC/AC Huffman table 0 (JCHTBD0 and JCHTBA0) 1 1: DC/AC Huffman table 1 (JCHTBD1 and JCHTBA1). Other settings are prohibited.	R/W*1
b3	HTA2	Huffman Table Number (AC) for the Second Color Component		R/W*1
b4	HTD3	Huffman Table Number (DC) for the Third Color Component	b5 b4 0 0: DC/AC Huffman table 0 (JCHTBD0 and JCHTBA0) 1 1: DC/AC Huffman table 1 (JCHTBD1 and JCHTBA1). Other settings are prohibited.	R/W*1
b5	HTA3	Huffman Table Number (AC) for the Third Color Component		R/W*1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In decompression mode, these bits are read-only.

57.2.5 JPEG Code DRI Upper Register (JCDRIU)

Address(es): JPEG.JCDRIU 400E 6005h

	b7	b6	b5	b4	b3	b2	b1	b0
	DRIU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DRIU[7:0]	Upper Bytes of MCUs Preceding RST Marker	Valid settings: 00h to FFh (0 to 255).*1	R/W*2

Note 1. When both JCDRIU = 00h and JCDRID = 00h, neither the DRI nor the RST marker is placed.
Note 2. In decompression mode, these bits are invalid.

57.2.6 JPEG Code DRI Lower Register (JCDRID)

Address(es): JPEG.JCDRID 400E 6006h

	b7	b6	b5	b4	b3	b2	b1	b0
	DRIU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DRIU[7:0]	Lower Bytes of MCUs Preceding RST Marker	Valid settings: 00h to FFh (0 to 255).*1	R/W*2

Note 1. When both JCDRIU = 00h and JCDRID = 00h, neither the DRI nor the RST marker is placed.
Note 2. In decompression mode, these bits are invalid.

57.2.7 JPEG Code Vertical Size Upper Register (JCVSZU)

Address(es): JPEG.JCVSZU 400E 6007h

	b7	b6	b5	b4	b3	b2	b1	b0
	VSZU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	VSZU[7:0]	Upper Bytes of Vertical Image Size	Valid settings: 00h to FFh (0 to 255).*2	R/W*1

Note 1. In decompression mode, these bits are read-only.
Note 2. In the decompression process, a downloaded value from the JPEG coded data is set.

57.2.8 JPEG Code Vertical Size Lower Register (JCVSZD)

Address(es): JPEG.JCVSZD 400E 6008h

	b7	b6	b5	b4	b3	b2	b1	b0
	VSZD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	VSZD[7:0]	Lower Bytes of Vertical Image Size	Valid settings: 00h to FFh (0 to 255).*2	R/W*1

Note 1. In decompression mode, these bits are read-only.
Note 2. In the decompression process, a downloaded value from the JPEG coded data is set.

57.2.9 JPEG Code Horizontal Size Upper Register (JCHSZU)

Address(es): JPEG.JCHSZU 400E 6009h

	b7	b6	b5	b4	b3	b2	b1	b0
	HSZU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	HSZU[7:0]	Upper Bytes of Horizontal Image Size	Valid settings: 00h to FFh (0 to 255).*2	R/W*1

Note 1. In decompression mode, these bits are read-only.
Note 2. In the decompression process, a downloaded value from the JPEG coded data is set.

57.2.10 JPEG Coded Horizontal Size Lower Register (JCHSZD)

Address(es): JPEG.JCHSZD 400E 600Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	HSZD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	HSZD[7:0]	Lower Bytes of Horizontal Image Size	Valid settings: 00h to FFh (0 to 255).*2	R/W*1

Note 1. In decompression mode, these bits are read-only.
Note 2. In the decompression process, a downloaded value from the JPEG coded data is set.

57.2.11 JPEG Code Data Count Upper Register (JCDTCU)

Address(es): JPEG.JCDTCU 400E 600Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	DCU[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DCU[7:0]	Upper Bytes of Counted Amount of Data to Be Compressed	Valid settings: 00h to FFh (0 to 255).	R*2

Note 1. The values in this register are reset before compression starts.
Note 2. In decompression mode, these bits are invalid.

57.2.12 JPEG Code Data Count Middle Register (JCDTCM)

Address(es): JPEG.JCDTCM 400E 600Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	DCM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DCM[7:0]	Middle Bytes of Counted Amount of Data to Be Compressed	Valid settings: 00h to FFh (0 to 255).	R*2

Note 1. The values in this register are reset before compression starts.
Note 2. In decompression mode, these bits are invalid.

57.2.13 JPEG Code Data Count Lower Register (JCDTCD)

Address(es): JPEG.JCDTCD 400E 600Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	DCD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DCD[7:0]	Lower Bytes of Counted Amount of Data to Be Compressed	Valid settings: 00h to FFh (0 to 255).	R*2

Note 1. The values in this register are reset before compression starts.
Note 2. In decompression mode, these bits are invalid.

57.2.14 JPEG Interrupt Enable Register 0 (JINTE0)

Address(es): JPEG.JINTE0 400E 600Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	INT7	INT6	INT5	—	INT3	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	INT3	Interrupt Request Signal INS3 Enable Control	Enables interrupt generation when determined that image size and pixel format of compressed data can be read by analyzing the data.	R/W*2
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	INT5	Interrupt Request Last MCU Data Number Error Enable Control	Enables interrupt generation when final number of MCU data units in Huffman-coding segment is not correct in decompression. When this bit is not set to enable, no error code is returned.	R/W*2
b6	INT6	Interrupt Request Total Number Error Enable Control	Enables interrupt generation when total number of data units in Huffman-coding segment is not correct in decompression. When this bit is not set to enable, no error code is returned.	R/W*2

Bit	Symbol	Bit name	Description	R/W
b7	INT7	Interrupt Request Restart Interval Data Number Error Enable Control	Enables interrupt generation when number of data units in restart interval of Huffman-coding segment is not correct in decompression. When this bit is not set to enable, no error code is returned.	R/W*2

Note 1. When any of the bits from INT7 to INT5 sets to 1, the INS5 bit in the JINTS0 register indicates 1 as the error status on occurrence of the compression data error, and the ERR[3:0] bits in the JCDERR register indicate the specific error code.
Note 2. In compression mode, these bits are invalid.

57.2.15 JPEG Interrupt Status Register 0 (JINTS0)

Address(es): JPEG.JINTS0 400E 600Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	INS6	INS5	—	INS3	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	INS3	Request for Reading the Image Size and Pixel Format	This bit sets to 1 when image size and pixel format can be read. When an interrupt occurs, the JPEG Codec stops processing and the state is indicated in the JCRST register. To resume module processing, set the JPEG Core Process Stop Clear Command Bit (JRST) in JCCMD.	R/W*2, *3
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	INS5	JPEG Decompression Error Occurrence	This bit sets to 1 when a compressed data error occurs.	R/W*2, *3
b6	INS6	JPEG Decompression Process End	This bit sets to 1 when the JPEG Codec completes compression process normally.	R/W*2
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The interrupt sources in this register should be cleared by clearing the associated interrupt status bits to 0 and setting the relevant bit in JCCMD appropriately.
Note 2. Clear this bit by writing 0 to it. Do not write 1 to this bit.
Note 3. In compression mode, these bits are invalid.

57.2.16 JPEG Code Decode Error Register (JCDERR)

Address(es): JPEG.JCDERR 400E 6010h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ERR[3:0]			
Value after reset:	0	0	0	0	1	0	1	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	ERR[3:0]	Error Code	See Table 57.3 and Table 57.4. for error codes.	R/W*2
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The values in this register are reset before the JPEG Codec starts decompression.
Note 2. In compression mode, these bits are invalid.

57.2.17 JPEG Code Reset Register (JCRST)

Address(es): JPEG.JCRST 400E 6011h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RST	Operating State	0: All other states 1: Suspended state caused by JINTE0 interrupt sources.	R*2
b7 to b1	—	Reserved	These bits are read as 0.	R

Note 1. To resume processing, set the JPEG core process stop clear command bit (JRST) in JCCMD.
Note 2. In compression mode, this bit is invalid.

57.2.18 JPEG Interface Compression Control Register (JIFECNT)

Address(es): JPEG.JIFECNT 400E 6040h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	JOUTSWAP[2:0]			—	DINRINI	DINRCMD	DINLC	—	DINSWAP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DINSWAP[2:0]	Byte/Halfword/Word Swap	Swaps input image data in compression. b2 b0 0 0 0: (1) (2) (3) (4) (5) (6) (7) (8) 0 0 1: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 0 1 0: (3) (4) (1) (2) (7) (8) (5) (6) [Halfword swap] 0 1 1: (4) (3) (2) (1) (8) (7) (6) (5) [Halfword - byte swap] 1 0 0: (5) (6) (7) (8) (1) (2) (3) (4) [Word swap] 1 0 1: (6) (5) (8) (7) (2) (1) (4) (3) [Word - byte swap] 1 1 0: (7) (8) (5) (6) (3) (4) (1) (2) [Word - halfword swap] 1 1 1: (8) (7) (6) (5) (4) (3) (2) (1) [Word - halfword - byte swap].	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DINLC	Count Mode Setting for Stopping Input Image Data Lines	0: Turn off count mode for stopping input of image data lines 1: Turn on count mode for stopping input of image data lines.	R/W*1
b5	DINRCMD	Input Image Data Lines Resume Command	Set this bit to 1 to resume reading of input image data. This bit is only valid when the count mode for stopping input of image data lines is on. This bit is always read as 0.	W*1
b6	DINRINI	Address Initialization when Resuming Input of Image Data Lines	This bit is only valid when the count mode for stopping input of image data lines is on. Set this bit before writing 1 to the data-line resume command bit. 0: Do not initialize transfer address when input of image data lines restarts 1: Initialize transfer address when input of image data lines restarts.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b10 to b8	JOUTSWAP[2:0]	Byte/Halfword/Word Swap	Swaps output coded data in compression. b10 b8 0 0 0: (1) (2) (3) (4) (5) (6) (7) (8) 0 0 1: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 0 1 0: (3) (4) (1) (2) (7) (8) (5) (6) [Halfword swap] 0 1 1: (4) (3) (2) (1) (8) (7) (6) (5) [Halfword - byte swap] 1 0 0: (5) (6) (7) (8) (1) (2) (3) (4) [Word swap] 1 0 1: (6) (5) (8) (7) (2) (1) (4) (3) [Word - byte swap] 1 1 0: (7) (8) (5) (6) (3) (4) (1) (2) [Word - halfword swap] 1 1 1: (8) (7) (6) (5) (4) (3) (2) (1) [Word - halfword - byte swap].	R/W*1
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In decompression mode, these bits are invalid.

57.2.19 JPEG Interface Compression Source Address Register (JIFESA)

Address(es): JPEG.JIFESA 400E 6044h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ESA[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ESA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	ESA[2:0]	Input Image Data Source Address	The lower three bits should be set to 0. These bits are read as 0.	R/W
b31 to b3	ESA[31:3]		Input image data source address in 8-byte units.	R/W*2

Note 1. This register must be set in 8-byte units.

Note 2. In decompression mode, these bits are invalid.

57.2.20 JPEG Interface Compression Line Offset Register (JIFESOFST)

Address(es): JPEG.JIFESOFST 400E 6048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	ESMW[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	ESMW[2:0]	Input Image Data Lines Offset	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b14 to b3	ESMW[14:3]		Input image data lines offset in 8-byte units.	R/W*2
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register must be set in 8-byte units.

Note 2. In decompression mode, these bits are invalid.

57.2.21 JPEG Interface Compression Destination Address Register (JIFEDA)

Address(es): JPEG.JIFEDA 400E 604Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EDA[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EDA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	EDA[2:0]	Output Coded Data Destination Address	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b31 to b3	EDA[31:3]		Output coded data destination address in 8-byte units.	R/W*2

Note 1. This register must be set in 8-byte units.

Note 2. In decompression mode, these bits are invalid.

57.2.22 JPEG Interface Compression Source Line Count Register (JIFESLC)

Address(es): JPEG.JIFESLC 400E 6050h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LINES[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	LINES[2:0]	Number of Input Image Data Lines to Be Read	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b15 to b3	LINES[15:3]		Number of input image data lines to be read, in 8-line units.	R/W*2
b18 to b16	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31 to b19	—	Reserved	These bits are read as undefined. The write value should be 1	R

Note 1. This register must be set in 8-byte units.

Note 2. In decompression mode, these bits are invalid.

57.2.23 JPEG Interface Decompression Control Register (JIFDCNT)

Address(es): JPEG.JIFDCNT 400E 6058h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	VINTER[1:0]		HINTER[1:0]		OPF[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	JINRINI	JINRCMD	JINC	—	JINSWAP[2:0]			—	DOUTRINI	DOUTRCMD	DOUTLC	—	DOUTSWAP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DOUTSWAP[2:0]	Byte/Halfword/Word Swap	Swaps output image data in decompression. b2 b0 0 0 0: (1) (2) (3) (4) (5) (6) (7) (8) 0 0 1: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 0 1 0: (3) (4) (1) (2) (7) (8) (5) (6) [Halfword swap] 0 1 1: (4) (3) (2) (1) (8) (7) (6) (5) [Halfword - byte swap] 1 0 0: (5) (6) (7) (8) (1) (2) (3) (4) [Word swap] 1 0 1: (6) (5) (8) (7) (2) (1) (4) (3) [Word - byte swap] 1 1 0: (7) (8) (5) (6) (3) (4) (1) (2) [Word - halfword swap] 1 1 1: (8) (7) (6) (5) (4) (3) (2) (1) [Word - halfword - byte swap].	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOUTLC	Count Mode for Stopping Output Image Data Lines	0: Turn off count mode for stopping output of image data lines 1: Turn on count mode for stopping output of image data lines.	R/W*1
b5	DOUTRCMD	Output Image Data Lines Resume Command	Set this bit to 1 to resume writing of image data. This bit is only valid when the count mode for stopping output of image data lines is on. This bit is always read as 0.	W*1
b6	DOUTRINI	Address Initialization When Resuming Output of Image Data Lines	This bit is only valid when the count mode for stopping output of image data lines is on. Set this bit before writing 1 to the data-line resume command bit. 0: Do not initialize transfer address when output of image data lines restarts 1: Initialize transfer address when output of image data lines restarts.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	JINSWAP[2:0]	Byte/Halfword/Word Swap	Swaps input coded data in decompression. b10 b8 0 0 0: (1) (2) (3) (4) (5) (6) (7) (8) 0 0 1: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 0 1 0: (3) (4) (1) (2) (7) (8) (5) (6) [Halfword swap] 0 1 1: (4) (3) (2) (1) (8) (7) (6) (5) [Halfword - byte swap] 1 0 0: (5) (6) (7) (8) (1) (2) (3) (4) [Word swap] 1 0 1: (6) (5) (8) (7) (2) (1) (4) (3) [Word - byte swap] 1 1 0: (7) (8) (5) (6) (3) (4) (1) (2) [Word - halfword swap] 1 1 1: (8) (7) (6) (5) (4) (3) (2) (1) [Word - halfword - byte swap].	R/W*1
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	JINC	Count Mode Setting for Stopping Input Coded Data	0: Turn off count mode for stopping input of coded data 1: Turn on count mode for stopping input of coded data.	R/W*1
b13	JINRCMD	Input Coded Data Resume Command	Set this bit to 1 to resume reading of input coded data. This bit is only valid when the count mode for stopping input of coded data is on. This bit is always read as 0.	W*1

Bit	Symbol	Bit name	Description	R/W
b14	JINRINI	Address Initialization When Input Coded Data is Resumed	This bit is only valid when the count mode for stopping input of coded data is on. Set this bit before writing 1 to the data resume command bit. 0: Do not initialize transfer address when input of coded data restarts 1: Initialize transfer address when input of coded data restarts.	R/W*1
b23 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	OPF[1:0]	Specifies Output Image Data Pixel Format*2	b25 b24 0 0: Setting prohibited 0 1: ARGB8888 1 0: RGB565 1 1: Setting prohibited.	R/W*1
b27, b26	HINTER[1:0]	Horizontal Subsampling	Subsamples horizontal output image data. b27 b26 0 0: No subsampling 0 1: Subsample output data into 1/2 1 0: Subsample output data into 1/4 1 1: Subsample output data into 1/8.	R/W*1
b29, b28	VINTER[1:0]	Vertical Subsampling	Subsamples vertical output image data. b29 b28 0 0: No subsampling 0 1: Subsample output data into 1/2 1 0: Subsample output data into 1/4 1 1: Subsample output data into 1/8.	R/W*1
b31 to b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In compression mode, these bits are invalid.
Note 2. Operation with the initial value is prohibited. Set these bits to 01b or 10b before starting operation.

57.2.24 JPEG Interface Decompression Source Address Register (JIFDSA)

Address(es): JPEG.JIFDSA 400E 605Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DSA[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DSA[2:0]	Input Coded Data Source Address	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b31 to b3	DSA[31:3]		Input coded data source address in 8-byte units.	R/W*2

Note 1. This register must be set in 8-byte units.
Note 2. In compression mode, these bits are invalid.

57.2.25 JPEG Interface Decompression Line Offset Register (JIFDDOFST)

Address(es): JPEG.JIFDDOFST 400E 6060h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	DDMW[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DDMW[2:0]	Output Image Data Lines Offset	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b14 to b3	DDMW[14:3]		Output image data lines offset in 8-byte units.	R/W*2
b31 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register must be set in 8-byte units.
Note 2. In compression mode, these bits are invalid.

57.2.26 JPEG Interface Decompression Destination Address Register (JIFDDA)

Address(es): JPEG.JIFDDA 400E 6064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DDA[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DDA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DDA[2:0]	Output Image Data Destination Address	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b31 to b3	DDA[31:3]		Output image data destination address in 8-byte units.	R/W*2

Note 1. This register must be set in 8-byte units.
Note 2. In compression mode, these bits are invalid.

57.2.27 JPEG Interface Decompression Source Data Count Register (JIFDSDC)

Address(es): JPEG.JIFDSDC 400E 6068h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	JDATAS[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	JDATAS[2:0]	Amount of Input Coded Data to Be Read	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b15 to b3	JDATAS[15:3]		Amount of input coded data to be read, in 8-byte units.	R/W*2
b18 to b16	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31 to b19	—	Reserved	These bits are read as undefined. The write value should be 1.	R

Note 1. This register must be set in 8-byte units.
Note 2. In compression mode, these bits are invalid.

57.2.28 JPEG Interface Decompression Destination Line Count Register (JIFDDLC)

Address(es): JPEG.JIFDDLC 400E 606Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LINES[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	LINES[2:0]	Number of Output Image Data Lines to Be Written	The lower three bits should be set to 0. These bits are read as 0.	R/W*2
b15 to b3	LINES[15:3]		Specifies number of lines of output image data to be written. Set the value so that the output image data line count matches the MCU unit. For YCbCr444, YCbCr422, and YCbCr411 output, the setting value × 1 is equal to the output image data line count. For YCbCr420 output, the setting value × 2 is equal to the output image data line count.	R/W*2
b18 to b16	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31 to b19	—	Reserved	These bits are read as undefined. The write value should be 1.	R

Note 1. Set this register so that the output image data line count matches the MCU unit.
Note 2. In compression mode, these bits are invalid.

57.2.29 JPEG Interface Decompression alpha Set Register (JIFDADT)

Address(es): JPEG.JIFDADT 400E 6070h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ALPHA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	ALPHA[7:0]	Alpha Value for Adding to ARGB8888 Format	Alpha value setting for output in ARGB8888 format.	R/W*1
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In compression mode, these bits are invalid.

57.2.30 JPEG Interrupt Enable Register 1 (JINTE1)

Address(es): JPEG.JINTE1 400E 608Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CBTEN	DINLEN	—	—	DBTEN	JINEN	DOUTLEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DOUTLEN	Interrupt Request Bit DOUTLF Enable Control	Enables or disables a data transfer processing interrupt request (JPEG_JDTI) when the DOUTLF bit in JINTS1 sets to 1. 0: Disable interrupt request 1: Enable interrupt request.	R/W*1
b1	JINEN	Interrupt Request Bit JINF Enable Control	Enables or disables a data transfer processing interrupt request (JPEG_JDTI) when the JINF bit in JINTS1 sets to 1. 0: Disable interrupt request 1: Enable interrupt request.	R/W*1
b2	DBTEN	Interrupt Request Bit DBTF Enable Control	Enables or disables a data transfer processing interrupt request (JPEG_JDTI) when the DBTF bit in JINTS1 sets to 1. 0: Disable interrupt request 1: Enable interrupt request.	R/W*1
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	DINLEN	Interrupt Request Bit DINLF Enable Control	Enables or disables a data transfer processing interrupt request (JPEG_JDTI) when the DINLF bit in JINTS1 sets to 1. 0: Disable interrupt request 1: Enable interrupt request.	R/W*2

Bit	Symbol	Bit name	Description	R/W
b6	CBTEN	Interrupt Request Bit CBTF Enable Control	Enables or disables a data transfer processing interrupt request (JPEG_JDTI) when the CBTF bit in JINTS1 sets to 1. 0: Disable interrupt request 1: Enable interrupt request.	R/W*2
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In compression mode, these bits are invalid.
Note 2. In decompression mode, these bits are invalid.

57.2.31 JPEG Interrupt Status Register 1 (JINTS1)

Address(es): JPEG.JINTS1 400E 6090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CBTF	DINLF	—	—	DBTF	JINF	DOUTLF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DOUTLF	Request for Decompressed Data Output Line Finish	In decompression, this bit sets to 1 when the number of lines of output image data indicated in JIFDDLC is written. This bit is only valid when the DOUTLC bit in JIFDCNT is set to 1.	R/W*1,*2,*3
b1	JINF	Request for JPEG Input Finish	This bit sets to 1 when the amount of input coded data indicated in JIFDSDC is read in decompression. This bit is only valid when the JINC bit in JIFDCNT is set to 1.	R/W*1,*2,*3
b2	DBTF	Request for Decompressed Data Bottom Finish	This bit sets to 1 when the last output image data is written in decompression.	R/W*1,*2,*3
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	DINLF	Request for Data Input Line Finish	This bit sets to 1 when the number of input image data lines indicated in JIFESLC is read in compression. This bit is only valid when the DINLC bit in JIFECNT is set to 1.	R/W*1,*2,*4
b6	CBTF	Request for Compressed Data Bottom Finish	This bit sets to 1 when the last output coded data is written in compression.	R/W*1,*2,*4
b31 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Clear the interrupt sources of this register by writing 0s to this register.
Note 2. When the bit is read as 1, write 0 to clear it. When the bit is read as 0, write 1 to set it.
Note 3. In compression mode, these bits are invalid.
Note 4. In decompression mode, these bits are invalid.

57.3 Operation

57.3.1 Compression

This section describes the compression process flows.

(1) Processing overview

The compression process overview is as follows:

1. The JPEG core is activated. A marker is output, after which, image data can be input. Approximately 30,000 cycles are required (for generating the SOI to SOS markers).

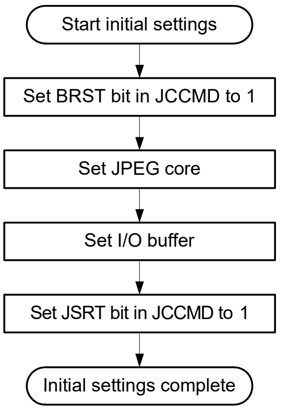
2. Image data is transferred in MCUs from the external buffer to the JPEG Codec.
    - If the count mode for stopping the input of image data lines is on, reading stops each time the number of lines set in JIFESLC is read. To resume reading, set the DINRCMD bit in JIFECNT to 1.
    - When the DINRINI bit in JIFECNT is 0, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
    - When the DINRINI bit is 1, the address set in JIFESA is used on resumption.
    - Reading is also stopped when one frame of image data is completely transferred.
    - If the count mode for stopping the input of image data lines is off, reading continues until one frame of image data is completely transferred.
3. Image data is input to the JPEG core. The input data is processed in MCUs at any time in the JPEG core.
4. Coded data is transferred from the JPEG Codec to the external buffer.
5. Compression is complete after one frame of data is processed completely.

(2) Compression flow

(a) Initial settings

To set the initial settings for compression:

After completing the JPEG core settings and input/output buffer settings and transferring image data to the external buffer, activate the JPEG Codec by setting the JSRT bit in JCCMD to 1. After the JPEG Codec is activated, the JPEG markers (SOI to SOS) are generated and output. Generating the markers takes approximately 30,000 cycles.



JPEG core settings	
Pixel format setting:	REDU bits in JCMOD
Compression setting:	DSP bit in JCMOD
Quantization table number setting:	JCQTN
Huffman table number setting:	JCHTN
DRI setting:	JCDRIU, JCDRID
Vertical image size setting:	JCVSZU, JCVSZD
Horizontal image size setting:	JCHSZU, JCHSZD
Quantization table setting:	JCQTBL0 to JCQTBL3
Huffman table setting:	JCHTBD0, JCHTBD1, JCHTBA0, JCHTBA1
I/O buffer settings	
Byte/halfword/word swap setting:	JOUTSWAP[2:0], DINSWAP[2:0] bits in JIFECNT
Input data line stop count mode setting:	DINLC bit in JIFECNT
Address initialization setting for resumption of input image data:	DINRINI bit in JIFECNT
Source address setting:	JIFESA
Line offset setting:	JIFESOFST
Destination address setting:	JIFEDA
Source line count setting:	JIFESLC
Interrupt settings:	CBTEN, DINLEN bits in JINTE1

Figure 57.2 Flow of compression initial settings

(b) Compression process

The compression process flow is as follows:

1. When JPEG compression process is completed, the INS6 bit in JINTS0 sets to 1. However, the JPEG Codec continues processing, because the coded data remains to be transferred. The CBTF bit in JINTS1 sets to 1 when the last coded data is transferred. The interrupt source is cleared by a 0 write to the INS6 bit. However, the interrupt request asserted by this interrupt source cannot be cleared by a 0 write to the INS6 bit. Set an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request.
2. When the JPEG Codec has completed compression and all coded data is transferred, the CBTF flag in JINTS1 sets to 1. When the CBTEN bit in JINTE1 is 1 here, an interrupt is generated. The interrupt source is cleared by a 0 write to the CBTF flag.

3. If the count mode for stopping image data lines is on, when the specified number of image data lines set in JIFESLC is read, the DINLF flag in JINTS1 sets to 1, and reading stops. When the DINLEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by a 0 write to the DINLEN bit. Setting the DINRCMD bit in JIFECNT to 1 resumes reading.
    - When the DINRINI bit in JIFECNT is 0, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
    - When the DINRINI bit is 1, the address set in JIFESA is used on resumption.

(c) Data correction

After the output coded data is divided by 8, if the remainder is 1 to 6 bytes, transfer of bytes 1 to 6 of the remainder might not complete successfully. If the transfer is unsuccessful, bytes 1 to 6 of the remainder are written to the address specified in the JPEG interface compression destination address register (JIFEDA), overwriting the existing data.*1

For this reason, it is necessary to check whether the output coded data was transferred successfully and, if not, to perform data correction.

Note 1. The JPEG Codec handles the output of coded data in 16-bit units. For this reason, if the coded data has an odd code length, the final code output is D9FFh. (FFh is appended.) When the remainder is 1, 3, or 5 bytes, the remainder data (1, 3, or 5 bytes) + FFh is written to the address specified in JIFEDA, overwriting the existing data.

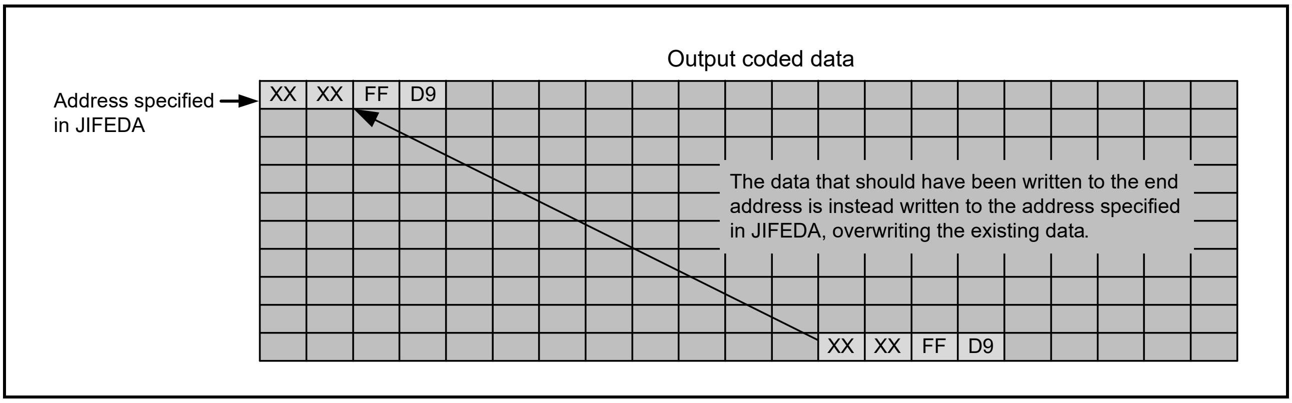


Figure 57.3 Conceptual diagram of abnormal transfer of output coded data

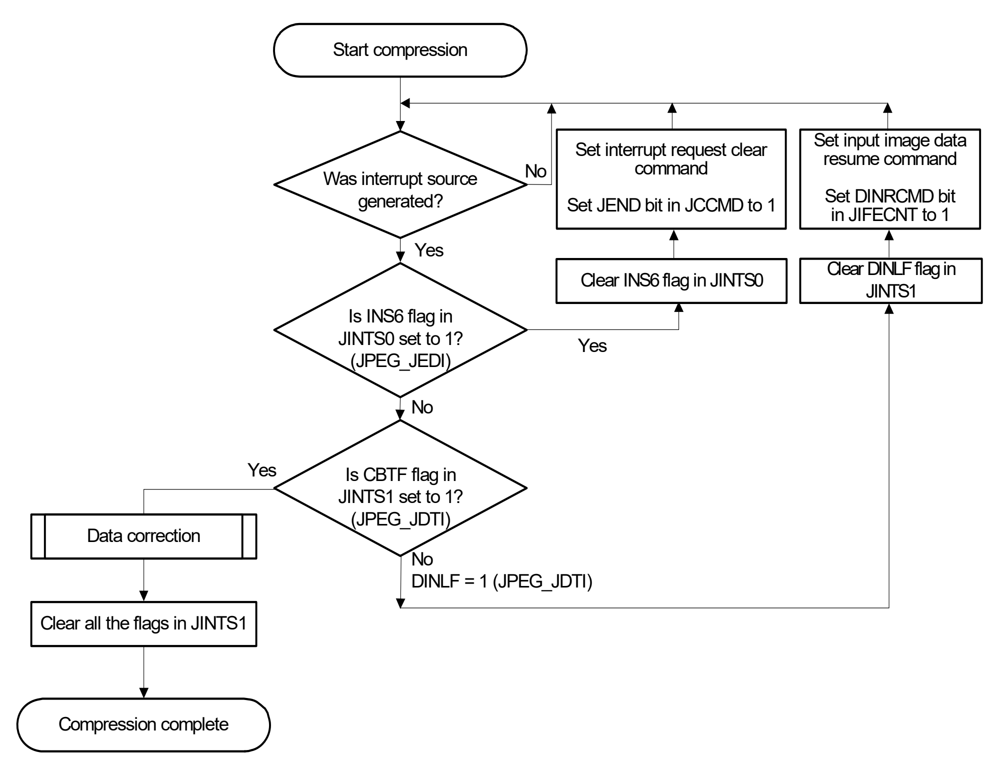
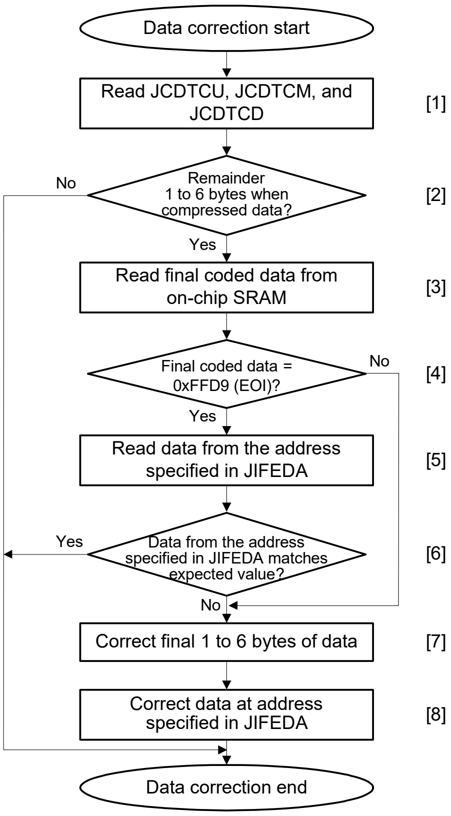


Figure 57.4 Compression process flow



[1] Read the compressed data volume from JCDTCU, JCDTCM, and JCDTCD.

[2] After dividing the compressed data volume by 8, determine whether the remainder is 1 to 6 bytes.

[3] Based on the compressed data volume read in [1], read the data from the address to which the final coded data (0xFFD9) was written.

[4] Determine if the final coded data EOI (0xFFD9) was written.

[5] Read from the address specified in JIFEDA the number of remainder data bytes calculated in [2].

[6] Determine whether the data read from the address specified in JIFEDA matches the expected value. This is the beginning of the JPEG data, so the expected value is 0xFFD8 (SOI), 0xFFDB (DQT), 0x00XX (XX: 0x84 for quantization table 2, 0xC5 for quantization table 3).

[7] If the transfer did not complete successfully, write the 1 to 6 bytes read in [6] to the end address.

[8] Some data is overwritten, so correct the data at the address specified in JIFEDA.

Note 1. If the coded data has an odd code length, the final code output is D9FFh. (FFh is appended.) Therefore, when the remainder is 1, 3, or 5 bytes, this should be corrected to 2, 4, or 6 bytes of data, respectively.

Figure 57.5 Data correction flow

(3) JPEG coded data format

Figure 57.6 shows the data output stream in compression. The amount of coded data from SOI to EOI is indicated in JCDTCU, JCDTCM, and JCDTCD. When both JCDRIU and JCDRID are set to 0000_0000h, the following markers are not output:

- DRI marker
- RST marker (in compressed image data).

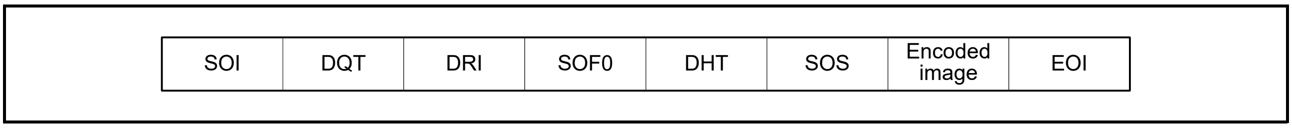


Figure 57.6 JPEG coded data format

- DQT: Not output for unused table
- DHT: Output in the order DC0, AC0, DC1, and AC1. Not output for unused tables
- SOF0: Component identifiers are C1 = first color component, C2 = second color component, and C3 = third color component
- SOS: Scan component selectors are CS1 = first color component, CS2 = second color component, and CS3 = third color component.

Header volume (reference)

- SOI: 2 bytes (FFD8)
- DQT: 134 bytes when two quantization tables are used, 199 bytes when three quantization tables are used (±65 bytes/table increase or decrease)
- DRI: 6 bytes
- SOF0: 19 bytes (4:2:2)
- DHT: 420 bytes (two tables are used)
- SOS: 14 bytes (4:2:2)
- EOI: 2 bytes (FFD9).

(4) Table setting

(a) Quantization table specification

The order of addresses shown in 8 × 8 blocks corresponds to that of the register addresses. Do not access this table while the JPEG Codec is processing.

Table 57.2 Quantization table

00	01	02	03	04	05	06	07
08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27
28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37
38	39	3A	3B	3C	3D	3E	3F

JCQTBL0 (400E 6100h) = 00h
JCQTBL0 (400E 6101h) = 01h
JCQTBL0 (400E 6102h) = 02h

JCQTBL0 (400E 6103h) = 03h
:
JCQTBL0 (400E 613Fh) = 3Fh
JCQTBL1 (400E 6140h) = 40h
JCQTBL1 (400E 6141h) = 41h
JCQTBL1 (400E 6142h) = 42h
JCQTBL1 (400E 6143h) = 43h
:
JCQTBL1 (400E 617Fh) = 7Fh
JCQTBL2 (400E 6180h) = 80h
JCQTBL2 (400E 6181h) = 81h
JCQTBL2 (400E 6182h) = 82h
JCQTBL2 (400E 6183h) = 83h
:
JCQTBL2 (400E 61BFh) = BFh
JCQTBL3 (400E 61C0h) = C0h
JCQTBL3 (400E 61C1h) = C1h
JCQTBL3 (400E 61C2h) = C2h
JCQTBL3 (400E 61C3h) = C3h
:
JCQTBL3 (400E 61FFh) = FFh

(b) Huffman table specification

This section provides examples of the Huffman table specification given in the ITU-T T81 Annex K.3.3 recommended by JPEG. In compression, the following settings must be specified for all the codes so that Huffman codes can be generated for all the group numbers.

- DC Huffman table: The number of codes for each code length is 12, and the group numbers in the order of frequency of occurrence are 12.
- AC Huffman table: The number of codes for each code length is 162, and the zero run length/group numbers in the order of frequency of occurrence are 162.

Do not access the following tables while the JPEG Codec is processing. In particular, read access is prohibited.

- Table K.3/T81

JCHTBD0 (400E 6200h) = 00h
JCHTBD0 (400E 6201h) = 01h
JCHTBD0 (400E 6202h) = 05h
JCHTBD0 (400E 6203h) = 01h
:
JCHTBD0 (400E 621Bh) = 0Bh

- Table K.4/T81

JCHTBD1 (400E 6300h) = 00h
JCHTBD1 (400E 6301h) = 03h
JCHTBD1 (400E 6302h) = 01h
JCHTBD1 (400E 6303h) = 01h
:
JCHTBD1 (400E 631Bh) = 0Bh

- Table K.5/T81

JCHTBA0 (400E 6220h) = 00h

JCHTBA0 (400E 6221h) = 02h
JCHTBA0 (400E 6222h) = 01h
JCHTBA0 (400E 6223h) = 03h
:
JCHTBA0 (400E 62D1h) = FAh

- Table K.6/T81

JCHTBA1 (400E 6320h) = 00h
JCHTBA1 (400E 6321h) = 02h
JCHTBA1 (400E 6322h) = 01h
JCHTBA1 (400E 6323h) = 02h
:
JCHTBA1 (400E 63D1h) = FAh

(5) Input pixel format

Image data in the YCbCr422 format can be input to the JPEG Codec. Allocation of data in the YCbCr422 format can be changed by the DINSWAP[2:0] bits in JIFECNT as shown in the following diagrams.

- When the DINSWAP[2:0] bits = 000b

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y0 8 bits	Cb0 8 bits	Y1 8 bits	Cr0 8 bits	Y2 8 bits	Cb1 8 bits	Y3 8 bits	Cr1 8 bits

- When the DINSWAP[2:0] bits = 001b

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cb0 8 bits	Y0 8 bits	Cr0 8 bits	Y1 8 bits	Cb1 8 bits	Y2 8 bits	Cr1 8 bits	Y3 8 bits

- When the DINSWAP[2:0] bits = 010b

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y1 8 bits	Cr0 8 bits	Y0 8 bits	Cb0 8 bits	Y3 8 bits	Cr1 8 bits	Y2 8 bits	Cb1 8 bits

- When the DINSWAP[2:0] bits = 100b

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y2 8 bits	Cb1 8 bits	Y3 8 bits	Cr1 8 bits	Y0 8 bits	Cb0 8 bits	Y1 8 bits	Cr0 8 bits

- When the DINSWAP[2:0] bits = 101b

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cb1 8 bits	Y2 8 bits	Cr1 8 bits	Y3 8 bits	Cb0 8 bits	Y0 8 bits	Cr0 8 bits	Y1 8 bits

- When the DINSWAP[2:0] bits = 110b

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y3 8 bits	Cr1 8 bits	Y2 8 bits	Cb1 8 bits	Y1 8 bits	Cr0 8 bits	Y0 8 bits	Cb0 8 bits

- When the DINSWAP[2:0] bits = 111b

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cr1 8 bits	Y3 8 bits	Cb1 8 bits	Y2 8 bits	Cr0 8 bits	Y1 8 bits	Cb0 8 bits	Y0 8 bits

(6) Output coded data

For compression, coded data is output. The JPEG Codec handles the output of coded data in 16-bit units. For this reason, if the coded data has an odd code length such as fractional, the final code for output is D9FFh.

The JOUTSWAP[2:0] bits in JIFECNT can be used to alter the arrangement of coded data in output.

57.3.2 Decompression

This section describes the decompression process flows.

(1) Processing overview

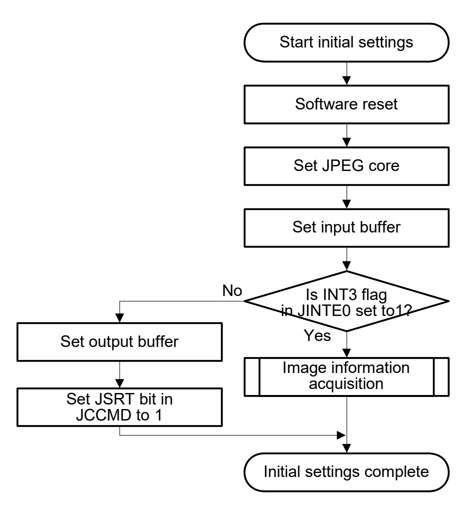
The decompression process overview is as follows:

1. The JPEG core is activated.
2. Coded data is transferred from the external buffer to the JPEG Codec.
    - If the count mode for stopping the input of coded data is on, reading stops each time the amount of coded data set in JIFDSLC is read. To resume reading, set the JINRCMD bit in JIFDCNT to 1.
    - When the JINRINI bit in JIFDCNT is 0, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
    - When the JINRINI bit is 1, the address set in JIFDSA is used on resumption.
    - Reading stops when the end of the coded data is detected.
    - If the count mode for stopping the input of coded data is off, reading continues until the end of code is detected. With this JPEG Codec, more coded data can be read than the coded data size because coded data reading continues until the end of code is detected.
3. Coded data is input to the JPEG core. The input data is processed in MCUs at any time in the JPEG core.
4. Image data is transferred in MCUs from the JPEG Codec to the external buffer.
    - If the count mode for stopping the output of image data lines is on, writing stops each time the number of image data lines set in JIFDDLC is written. To resume writing, set the DOUTRCMD bit in JIFDCNT to 1.
    - When the DOUTRINI bit in JIFDCNT is 0, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
    - When the DOUTRINI bit is 1, the address set in JIFDDA is used on resumption.
    - Writing stops when one frame of image data is completely transferred.
    - If the count mode for stopping the output of image data lines is off, writing continues until one frame of image data is completely transferred.
5. Decompression is complete after one frame of data is processed completely.

(a) Initial settings

To set the initial settings for decompression:

1. When the INT3 bit in JINTE0 is 0:
   After completing the JPEG core settings and input/output buffer settings and transferring coded data to the external buffer, activate the JPEG Codec by setting the JSRT bit in JCCMD to 1.
2. When the INT3 bit in JINTE0 is 1:
    a. After completing the JPEG core settings and input buffer settings and transferring coded data to the external buffer, activate the JPEG Codec by setting the JSRT bit in JCCMD to 1.
    b. When the image size and pixel format become readable after the coded data is decompressed, the INS3 bit in JINTS0 is set. At this time, decompression stops temporarily.
    c. After the image size and pixel format are read, set the output buffer.
    d. After interrupt handling, set the JRST bit in JCCMD to 1 to resume decompression.



Software reset
Reset the JPEG Codec in advance by setting the BRST bit in the JCCMD register.

JPEG core settings

Decompression setting:	DSP bit in JCMOD
Interrupt setting:	INT7 to INT5 and INT3 flags in JINTE0

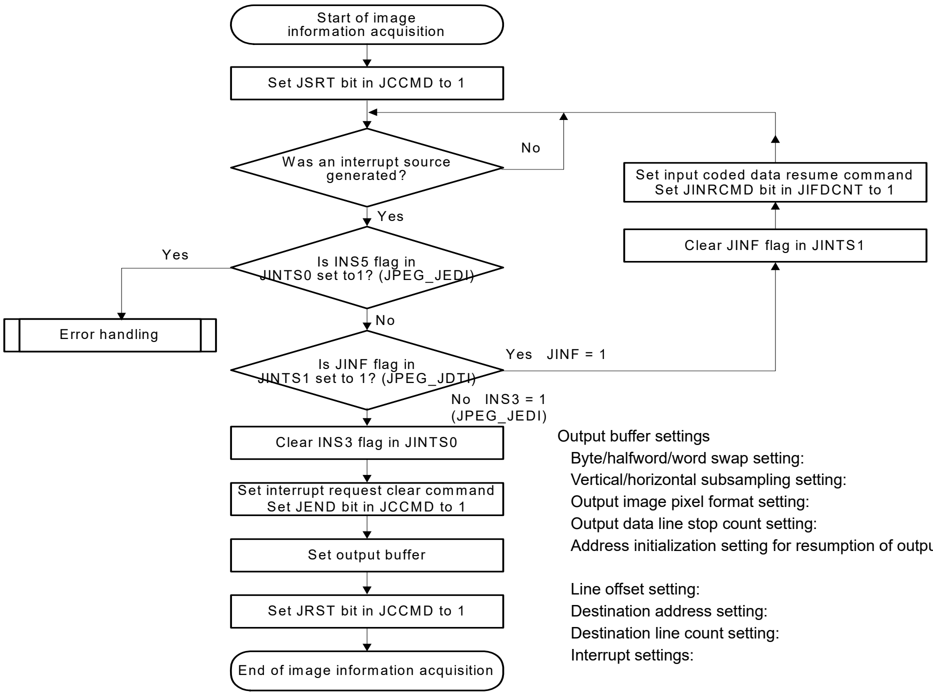
Input buffer settings

Byte/halfword/word swap setting:	JINSWAP bit in JIFDCNT
Input coded data stop count setting:	JINC bit in JIFDCNT
Address initialization setting for resumption of input coded data:	JINRINI bit in JIFDCNT
Source address setting:	JIFDSA
Source count setting:	JIFDSDC
Interrupt setting:	JINEN bit in JINTE0

Output buffer settings

Byte/halfword/word swap setting:	DOUTSWAP bit in JIFDCNT
Vertical/horizontal subsampling setting:	VINTER and HINTER bits in JIFDCNT
Output image pixel format setting:	OPF bits in JIFDCNT
Output data line stop count setting:	DOUTLC bit in JIFDCNT
Address initialization setting for resumption of output image data:	DOUTRINI bit in JIFDCNT
Line offset setting:	JIFDDOFST
Destination address setting:	JIFDDA
Destination line count setting:	JIFDDLC
Interrupt setting:	DBTEN, DOUTLEN bits in JINTE1

Figure 57.7 Flow of decompression initial settings



Output buffer settings

Byte/halfword/word swap setting:	DOUTSWAP bit in JIFDCNT
Vertical/horizontal subsampling setting:	VINTER and HINTER bits in JIFDCNT
Output image pixel format setting:	OPF bits in JIFDCNT
Output data line stop count setting:	DOUTLC bit in JIFDCNT
Address initialization setting for resumption of output image data:	DOUTRINI bit in JIFDCNT
Line offset setting:	JIFDDOST
Destination address setting:	JIFDDA
Destination line count setting:	JIFDDLC
Interrupt settings:	DBTEN, DOUTLEN bits in JINTE1

Figure 57.8 Flow of image information acquisition

(b) Decompression process

The decompression process flow is as follows:

1. When JPEG decompression process is completed, the INS6 bit in JINTS0 sets to 1. However, the JPEG Codec continues processing, because the image data remains to be transferred. The DBTF bit in JINTS1 sets to 1 when the last image data is transferred. The interrupt source is cleared by a 0 write to the INS6 bit. However, the interrupt request asserted by this interrupt source cannot be cleared by a 0 write to the INS6 bit. Set an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request.
2. When the JPEG Codec has completed decompression and all image data is transferred, the DBTF flag in JINTS1 sets to 1. When the DBTEN bit in JINTE1 is 1 here, an interrupt is generated. The interrupt source is cleared by a 0

write to the DBTF flag.

3. If the count mode for stopping input coded data is on, when the specified amount of coded data set in JIFDSDC is read, the JINF flag in JINTS1 sets to 1, and reading stops. When the JINEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by a 0 write to the JINF bit. Setting the JINRCMD bit in JIFDCNT to 1 resumes reading.
   - When the JINRINI bit in JIFDCNT is 0, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
   - When the JINRINI bit is 1, the address set in JIFDSA is used on resumption.
4. If the count mode for stopping the output image data is on, when the specified number of image data lines set in JIFDDLC is written, the DOUTLF flag in JINT1 sets to 1 and writing stops. When the DOUTLEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by a 0 write to the DOUTLF bit. Setting the DOUTRCMD bit in JIFDCNT to 1 resumes writing.
   - When the DOUTRINI bit in JIFDCNT is 0, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
   - When the DOUTRINI bit is 1, the address set in JIFDDA is used on resumption.

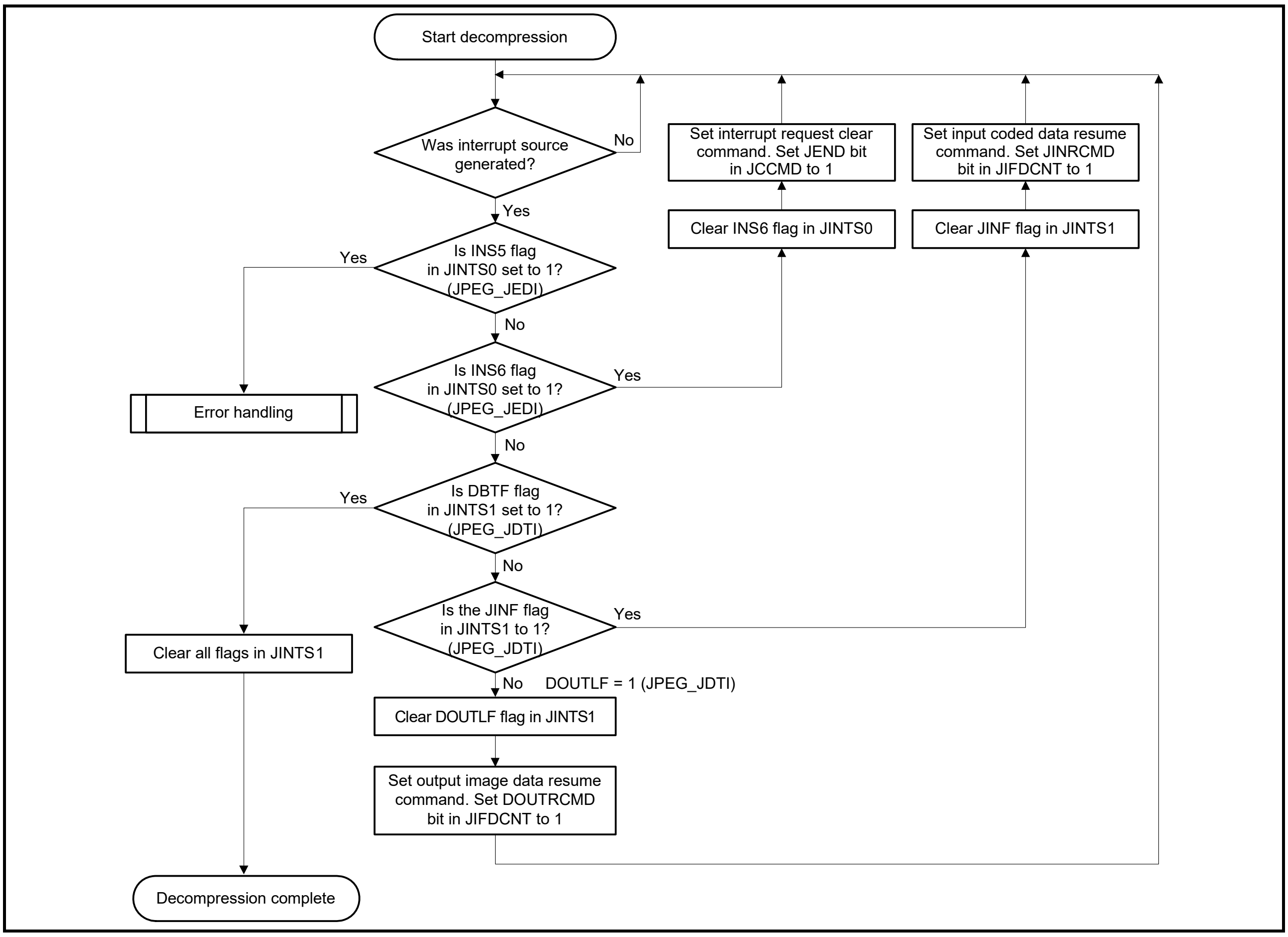


Figure 57.9 Decompression process flow

(c) Error handling

If the INS5 bit in JINTS0 is 1, it indicates that there is an error in the input JPEG coded data and that the decompression process by the JPEG Codec has ended. Read the ERR[3:0] bits in JCDERR to determine the cause of the error. Interrupt signals asserted because of the interrupt source indicated in the INS5 bit cannot be negated by clearing the interrupt status with a 0 write. To clear the interrupt request, you must set the interrupt request clear command (by setting the JEND bit in JCCMD to 1). To perform decompression or compression processing after error handling finishes, start the processing from the initial settings.

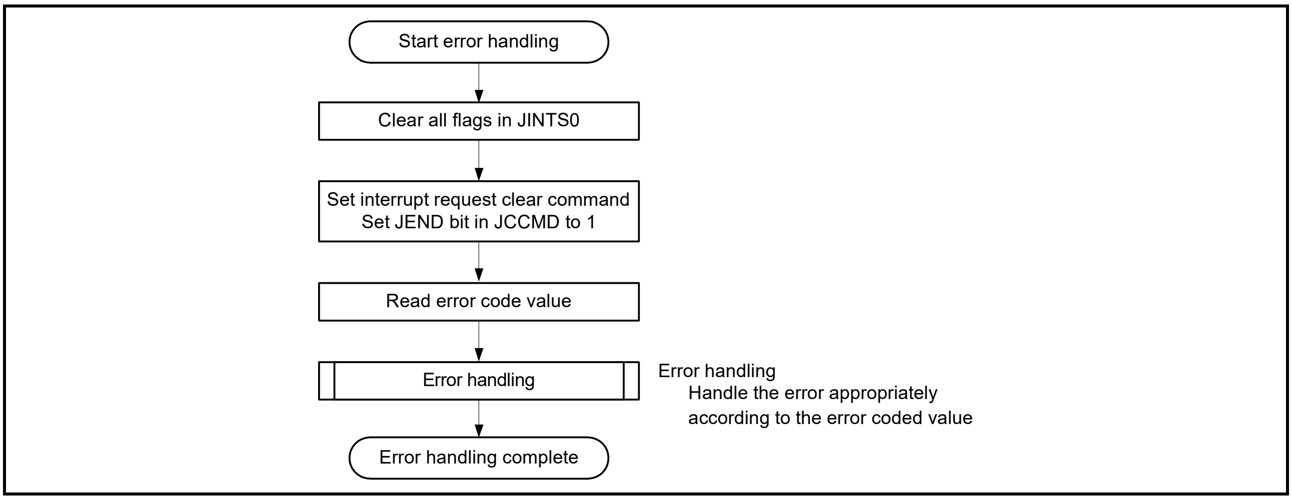


Figure 57.10 Error handling flow

(2) Input JPEG coded data

The markers to be processed in decompression are SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, and EOI. Except for the error markers shown in Table 57.3, other markers are ignored even if they are read.

The JINSWAP[2:0] bits in JIFDCNT can be used to alter the arrangement for the input of coded data.

(3) JPEG decompression errors

(a) Error markers

If a marker error is found during analysis of compressed data for decompression, the code to identify the error type (shown in Table 57.3) is set to the ERR[3:0] bits in JCDERR. When an error is detected, the JPEG Codec generates an interrupt signal and terminates decoding. The stored code value is set to 1010b (default value) at the start of processing of the next frame or after a bus reset.

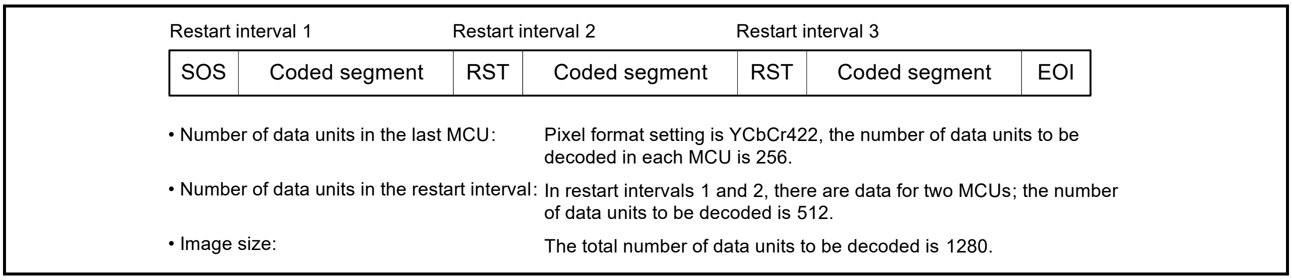
Table 57.3 Decompression error codes

Code	Description
0000b	Normal
0001b	SOI not detected: SOI not detected until EOI detected
0010b	SOF1 to SOFF detected
0011b	Unprovided pixel format detected
0100b	SOF accuracy error: Value other than 8 detected
0101b	DQT accuracy error: Value other than 0 detected
0110b	Component error 1: Number of SOF0 header components detected is value other than 1, 3, or 4
0111b	Component error 2: Number of components differs between SOF0 header and SOS
1000b	SOF0, DQT, and DHT not detected when SOS detected
1001b	SOS not detected: SOS not detected until EOI detected
1010b	EOI not detected (default)
1011b	Restart interval data number error detected
1100b	Image size error detected
1101b	Last MCU data number error detected
1110b	Block data number error detected

(b) Huffman coded segment error

During the compression data analysis in decompression operation, if there is an increase or decrease in the decoded data count because of an error resulting from bit reversal or missing data in the Huffman-coded segment, determine the error type, and set the error code in the ERR[3:0] bits in JCDERR. Table Table 57.4 lists the segment error codes. The error code is set, an interrupt signal is issued, and the process ends only if the INT7 to INT5 bit in JINTE0 associated with the detected error is set to 1. The set code value returns to the default value (1010h) at the start of processing of the next frame or after a bus reset. However, in this error detection if an error in the Huffman-coded segment does not result in an alteration in the decoded data count, the error is undetected.

Figure 57.11 shows an example of the number of data units in a Huffman coded segment with pixel format setting YCbCr422, DRI = 2, X = 80 pixels, and Y = 8 pixels.



- Number of data units in the last MCU: Pixel format setting is YCbCr422, the number of data units to be decoded in each MCU is 256.
- Number of data units in the restart interval: In restart intervals 1 and 2, there are data for two MCUs; the number of data units to be decoded is 512.
- Image size: The total number of data units to be decoded is 1280.

Figure 57.11 Huffman coded segment

Table 57.4 Segment error codes

Code	Description
0000h	Normal
1011h	Restart interval data number error: The number of data units in each interval is compared with the number of data units specified by the DRI marker. If an interval has more or less data than is specified by the DRI marker, the decompression error code (1011) is set. The last interval which is shorter than the restart interval is not compared. If the DRI marker segment is not placed or the specified number is 00, an error is not detected even if the RSTm marker is placed. Also, an m indicating the order of RSTm marker modulo 8 (m = 0 to 7) is exempt from the error detection analysis. When the INT7 bit in JINTE0 sets to 0, this error is not detected.
1100h	Image size error: The number of data units in an image that is calculated from the number of lines specified by the frame parameter and the number of samples per line is compared with the total number of data units from SOS to EOI (in pixel units). If the numbers of data units do not match, the decompression error code (1100) is set. When the INT6 bit in JINTE0 sets to 0, this error is not detected. The number of data units in an image is shown in MCUs. As a result, the number of lines and the number of samples per line for calculation must be shown in MCUs.
1101h	Last MCU data number error: Whether the number of data units in MCUs on EOI detection is shown in MCU units is checked and fractions are detected. If error (1100) occurs simultaneously with another error, error (1100) has priority. When the INT5 bit in JINTE0 sets to 0, this error is not detected.
1110h	Block data number error: Whether a block is an 8 × 8 array is checked and fractions are detected. When the INT7 to INT5 bits in JINTE0 are all set to 0, this error is not detected.

57.3.3 Output Pixel Format in Decompression

The JPEG Codec is capable of decompressing JPEG encoded data created in the YCbCr444, YCbCr422, YCbCr411, and YCbCr420 formats. The pixel format of the output image is ARGB8888 or RGB565. Figure 57.12 shows the flow of converting decompressed data to the given output pixel format.

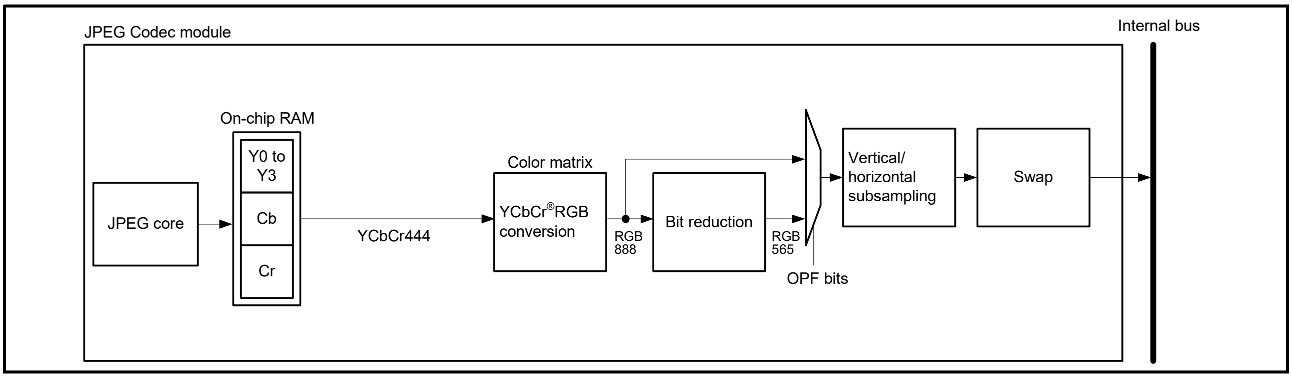


Figure 57.12 Output pixel format conversion in decompression block diagram

(1) On-chip SRAM

Data decoded by the JPEG core is stored in MCUs on SRAM in the JPEG Codec.

(2) YCbCr → RGB conversion

Data in the YCbCr444 format is converted to the RGB888 format. The following formulas are used:

$$R = 1.000Y + 1.402Cr$$
$$G = 1.000Y - 0.344Cb - 0.714Cr$$
$$B = 1.000Y + 1.772Cb$$

(3) Bit reduction

RGB888 data is reduced to RGB565 data. The lower three bits of red and blue, and lower two bits of green are removed.

(4) Output pixel format selection

The pixel format to be output is selected in the OPF[1:0] bits in JIFDCNT. When the DOUTSWAP[2:0] bits in JIFDCNT = 000b, allocation of data in the pixel format is as follows:

- ARGB8888 (32 bits/pixel)

b31 b24	b23 b16	b15 b8	b7 b0
*1	Red 8 bits	Green 8 bits	Blue 8 bits

Note 1. This value is determined by the ALPHA[7:0] bits in JIFDADT.

- RGB565 (16 bits/pixel)

b15 b11	b10 b5	b4 b0
Red 5 bits	Green 6 bits	Blue 5 bits

(5) Vertical and horizontal subsampling

The output data can be horizontally and vertically subsampled based on the VINTER[1:0] and HINTER[1:0] settings in JIFDCNT. Figure 57.13 to Figure 57.15 show the line subsampling modes. In the figures, for the ARGB8888 and RGB565 output formats, one cell represents one pixel. Because sub-sampling is carried out in minimum coded units (MCUs), the numbers of the horizontal and vertical block units vary based on the decompressed pixel format.

Table 57.5 and Table 57.6 show the values of n and m in the figures.

Horizontal

Table 57.5 Number of horizontal blocks (1 of 2)

Compression format	Output format	n
YCbCr444	ARGB8888, RGB565	1
YCbCr422	ARGB8888, RGB565	2

Table 57.5 Number of horizontal blocks (2 of 2)

Compression format	Output format	n
YCbCr411	ARGB8888, RGB565	4
YCbCr420	ARGB8888, RGB565	2

Vertical

Table 57.6 Number of vertical blocks

Compression format	Output format	m
YCbCr444	ARGB8888, RGB565	1
YCbCr422	ARGB8888, RGB565	1
YCbCr411	ARGB8888, RGB565	1
YCbCr420	ARGB8888, RGB565	2

- Sub-sampling into 1/2

  Even lines are skipped by sub-sampling.

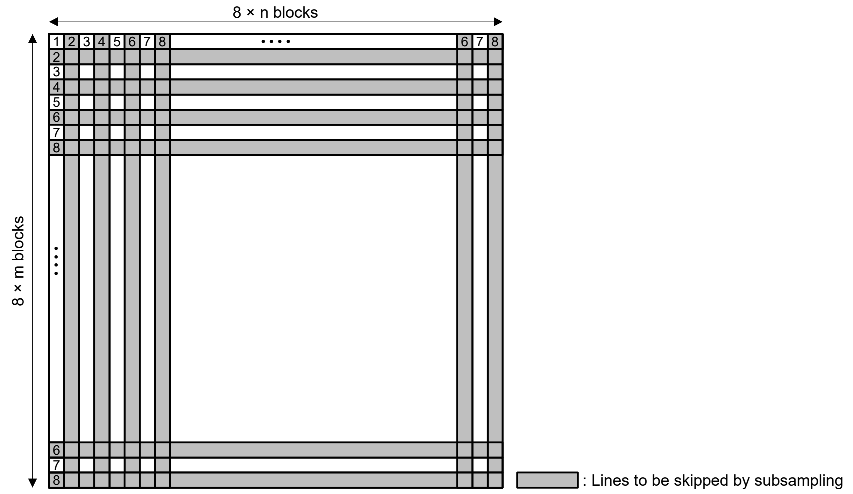


Figure 57.13 MCU when sub-sampling into 1/2 is selected

- Sub-sampling into 1/4

  The second, third, and fourth lines are skipped by sub-sampling.

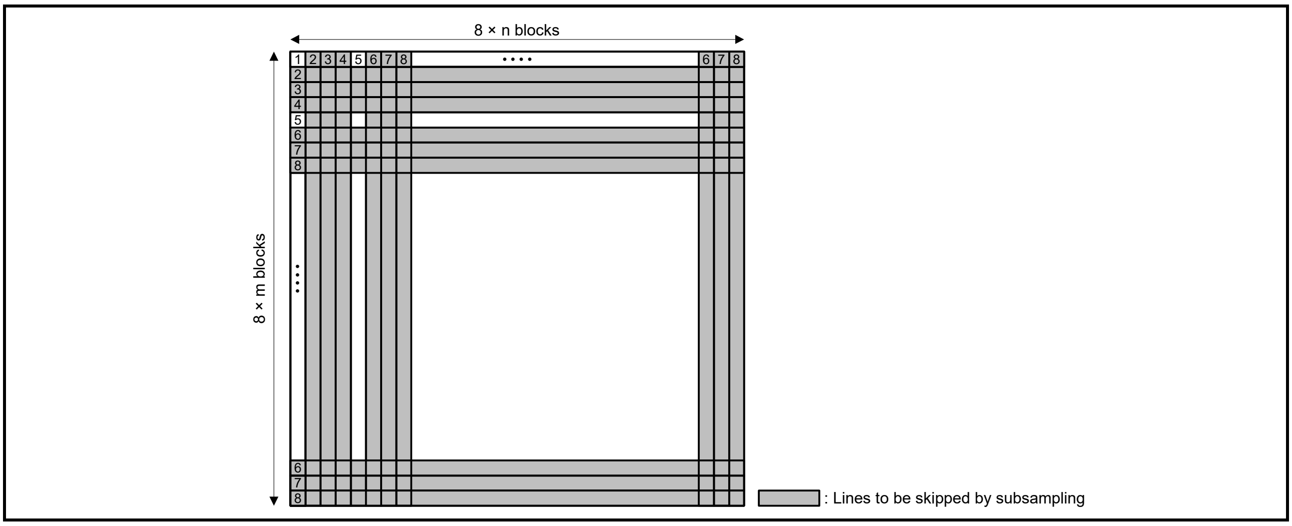


Figure 57.14 MCU when sub-sampling into 1/4 is selected

- Sub-sampling into 1/8

  The second, third, fourth, fifth, sixth, seventh, and eighth lines are skipped by sub-sampling.

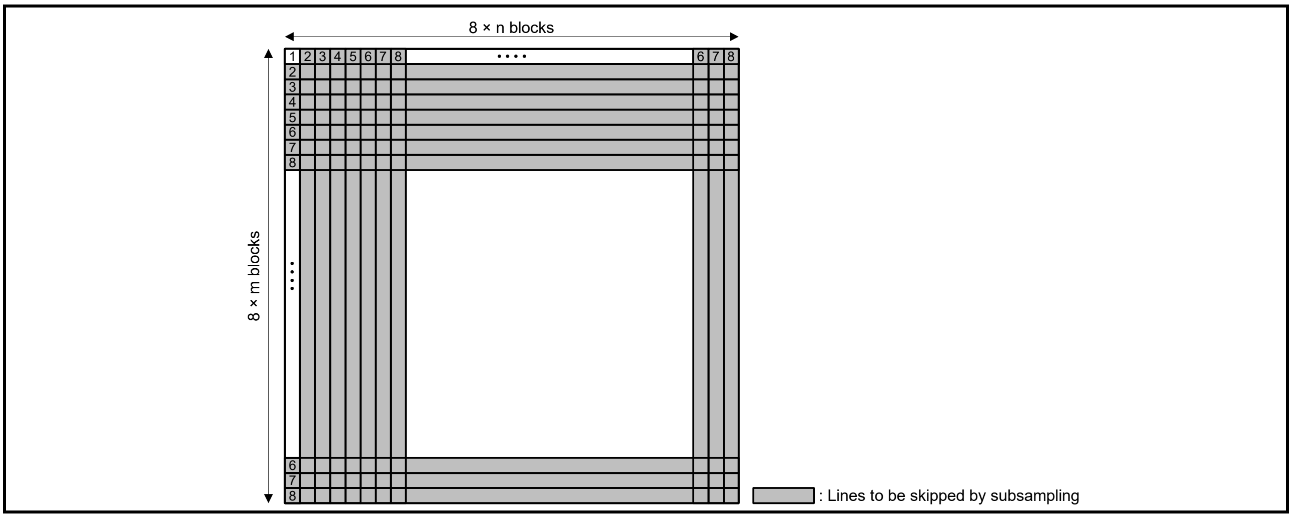


Figure 57.15 MCU when sub-sampling into 1/8 is selected

(6) Swap

Allocation of data can be changed using the DOUTSWAP[2:0] bits in JIFDCNT.

57.3.4 Storing Image Data

Figure 57.16 shows the buffer area for storing the image data.

- Start address
  - Compression: JIFESA
  - Decompression: JIFDDA
- Horizontal size
  - Compression, decompression: JCHSZU, JCHSZD
- Vertical size
  - Compression, decompression: JCVSAU, JCVSZD

- Offset
    - Compression: JIFESOFST
    - Decompression: JIFDDOFST

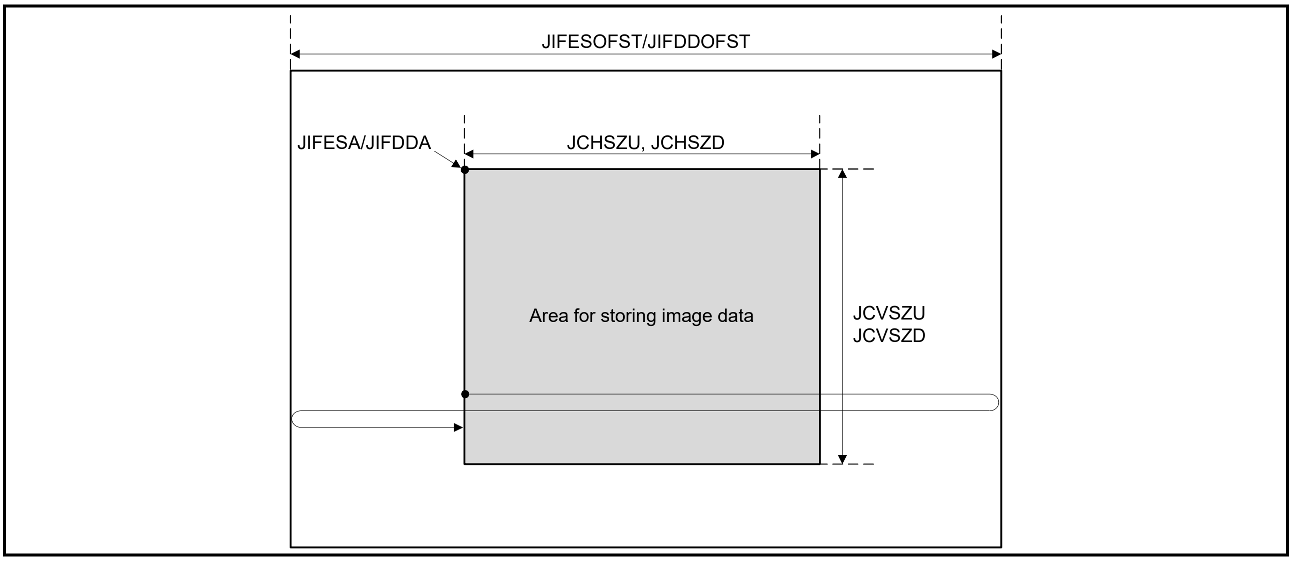


Figure 57.16 Conceptual diagram of image data storing

57.4 Interrupts

Two types of interrupt requests, compression/decompression process interrupt requests (JPEG_JEDI) and data transfer interrupt requests (JPEG_JDTI), are available in the JPEG Codec. The two types of interrupt requests are each related to multiple sources. The interrupt request cancellation methods differ depending on the source of the interrupt request.

57.4.1 Compression/Decompression Process Interrupt Request (JPEG_JEDI)

The flags in JINTS0 indicate compression and decompression-related sources. The interrupt requests asserted by these interrupt sources cannot be negated by clearing the associated interrupt status bits to 0. Issue an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request. When a flag in JINTS0 sets to 1, a compression/decompression process interrupt request is sent to the Interrupt Controller Unit (ICU).

(1) Compression

- JPEG compression process end

  When the INS6 bit in JINTS0 is 1, the JPEG compression process completed successfully. After all of the coded data is transferred, the JPEG Codec completes compression.

(2) Decompression

- JPEG decompression process end

  When the INS6 bit in JINTS0 is 1, the JPEG decompression process completed successfully. After all of the image data is transferred, the JPEG Codec completes decompression.
- JPEG decompression error occurrence

  When the INS5 bit in JINTS0 is 1, the input JPEG coded data has an error and the JPEG Codec has stopped the decompression process. Read the error code (ERR[3:0] bits in JCDERR) and identify the error source. This interrupt occurs when any of the INT7 to INT5 bits in JINTE0 is 1.
- Request for reading the image size and pixel format

  When the INS3 bit in JINTS0 is 1, JPEG coded data is input, and information regarding the image size and pixel format can be read. Because the JPEG decompression process is suspended, resume the JPEG decompression process by setting the process stop clear command after accessing the required registers. This interrupt occurs when the INT3 bit in JINTE0 is 1.

57.4.2 Data Transfer Interrupt Request (JPEG_JDTI)

The flags in JINTS1 are the interrupt sources for transferring the image data and coded data. The interrupt requests asserted by these interrupt sources can be negated by clearing the associated interrupt status bits to 0.

(1) Compression

- Interrupt request generated after the specified number of input image data lines is read

  When the DINLF bit in JINTS1 is 1, the number of image data lines specified in JIFESLC is transferred. Transfer the rest of the image data to the external buffer and resume transferring the data from the external buffer. A data transfer interrupt request is sent when the DINLEN bit in JINTE1 is 1.
- Interrupt request generated after all processes are complete

  When the CBTF bit in JINTS1 is 1, the JPEG Codec has completed compression and transferred all of the coded data. The data transfer interrupt request is sent when the CBTEN bit in JINTE1 is set to 1.

(2) Decompression

- Interrupt request generated after the specified number of output image data lines is written to

  When the DOUTLF bit in the JINTS1 is 1, the number of image data lines specified in JIFDDLC is transferred. Secure a space for the next coded data in the external buffer and resume the transfer process. A data transfer interrupt request is sent when the DOUTLEN bit in JINTE1 is 1.
- Interrupt request generated after the specified amount of input coded data is read

  The JINF bit in JINTS1 sets to 1 when the amount of coded data specified in JIFDSDC is transferred. Secure the next coded data in the external buffer and resume the transfer process. A data transfer interrupt is also sent at this time if the JINEN bit in JINTE1 is 1.
- Interrupt request generated after all processes are completed

  The DBTF bit in JINTS1 sets to 1 when the JPEG Codec has completed decompression and transferred all of the coded data. A data transfer interrupt request is also sent at this time if the DBTEN bit in JINTE1 is set to 1.

57.5 Bus Reset Processing

Issuing the bus reset command (setting the BRST bit in JCCMD to 1) causes a bus reset. Do not issue the bus reset command while the JPEG Codec is operating. The following registers are initialized by a bus reset:

- JPEG Code Data Count Upper Register (JCDTCU)
- JPEG Code Data Count Middle Register (JCDTCM)
- JPEG Code Data Count Lower Register (JCDTCD)
- JPEG Interrupt Status Register 0 (JINTS0)
- JPEG Code Decode Error Register (JCDERR)
- JPEG Code Reset Register (JCRST).

57.6 Usage Notes

57.6.1 Notes on the Decompression Process

Renesas recommends that a timeout detection mechanism be implemented through software or through the operating system to prevent locking of the module in case invalid JPEG code data is detected. This timeout detection mechanism is recommended especially when setting JIFDCNT.JINC bit to 1 because the decompression process might not be complete depending on the timing of the JPEG code data input. If a timeout error occurs while decoding the image, reset the JPEG Codec by setting the BRST bit in the JCCMD register first, then follow the flow for decompression initial settings as shown in Figure 57.7.

58. Graphics LCD Controller (GLCDC)

The multifunctional Graphics LCD Controller (GLCDC) supports multiple data formats and panels. Key GLCDC features include:

- GPX bus master function for accessing graphics data
- Superimposition of three planes (single color background plane, graphics 1 plane, and graphics 2 plane)
- Supports many types of 32- and 16-bit/pixel graphics data and 8-, 4-, and 1-bit LUT data formats
- Digital interface signal output supporting the video image size of WVGA.

Figure 58.1 shows a block diagram of the GLCDC.

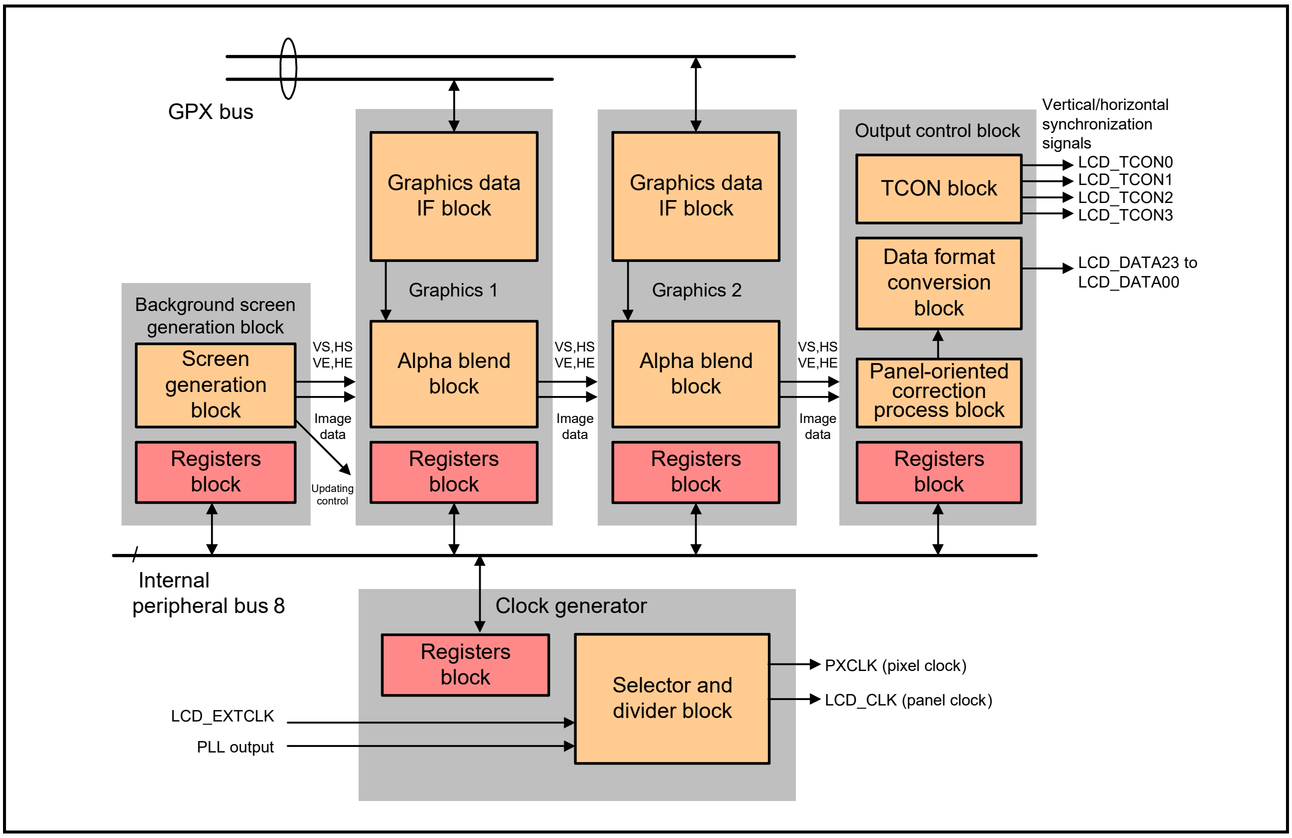


Figure 58.1 GLCDC block diagram

Note: When using the GLCDC, set the clock to ICLK = PCLKA.

Table 58.1 GLCDC I/O pins (1 of 2)

Pin name	I/O	Function
LCD_EXTCLK	Input	Panel clock source input pin
LCD_CLK	Output	Panel clock output pin
LCD_DATA23 to LCD_DATA00	Output	LCD signal output pins RGB888 signal R/G/B: 8 bits each (unsigned) RGB666 signal R/G/B: 6 bits each (unsigned) RGB565 signal R/B: 5 bits each (unsigned) G: 6 bits (unsigned) Serial RGB signal R/G/B/-: 8 bits (unsigned)
LCD_TCON3	Output	LCD_TCON3 output signal pin
LCD_TCON2	Output	LCD_TCON2 output signal pin
LCD_TCON1	Output	LCD_TCON1 output signal pin

Table 58.1 GLCDC I/O pins (2 of 2)

Pin name	I/O	Function
LCD_TCON0	Output	LCD_TCON0 output signal pin

58.1 Functional Overview

Table 58.2 provides a functional overview of GLCDC.

Table 58.2 Functional overview of GLCDC (1 of 2)

Parameter		Function
Graphics	Graphics planes	• Single color background (lowest layer) and two graphics planes • Graphics planes can be alpha-blended with the lower-layer plane
	Pixel format	• RGB-888 progressive format (-: 8 bits, R: 8 bits, G: 8 bits, B: 8 bits; 32 bits in total) • ARGB8888 progressive format (A: 8 bits, R: 8 bits, G: 8 bits, B: 8 bits; 32 bits in total) • RGB565 progressive format (A: None, R: 5 bits, G: 6 bits, B: 5 bits; 16 bits in total) • ARGB1555 progressive format (CLUT: 1 bit, R: 5 bits, G: 5 bits, B: 5 bits; 16 bits in total) • ARGB4444 progressive format (A: 4 bits, R: 4 bits, G: 4 bits, B: 4 bits; 16 bits in total) • CLUT8 progressive format (color palette address: 8 bits) • CLUT4 progressive format (color palette address: 4 bits) • CLUT1 progressive format (color palette address: 1 bit) • CLUT memory: 512 words × 32 bits per graphics plane (ARGB8888)
	Frame buffer control	The following parameters can be set for the frame buffer: • Base address: Start address of the frame buffer, aligned with a 64-byte boundary (burst transfer size) • Macro line offset: Offset address from the start address to the next macro line, aligned with a 64-byte boundary (burst transfer size) • Frame offset: Offset address from the start address to the next frame, aligned with a 64-byte boundary (burst transfer size) • Number of data transfers: Number of data transfers of a macro line • Number of macro lines: Number of macro lines in a single frame
	Alpha blending	• Alpha blending in rectangular area (blending ratio: 256 gradation levels) • Alpha blending in pixel units (blending ratio: 256 gradation levels) • RGB-index chroma key (replaced with the specified color when the object color agrees with the preset value)
Internal video image format		• Total number of vertical lines: Up to 1024 lines • Number of valid vertical lines: 16 to 1020 lines (resolution: 1 line) • Vertical front porch: 2 lines (minimum) • Vertical back porch: 1 line (minimum) • Vertical synchronization (VS) pulse width: 1 line (fixed) • Total number of horizontal pixels: Up to 1024 pixels • Number of valid horizontal pixels: 16 to 1016 pixels (resolution: 2 pixels) • Horizontal front porch: 3 pixels (minimum) • Horizontal back porch: 1 pixel (minimum) • Horizontal synchronization (HS) pulse width: 4 pixels
Data format conversion	Output video image size	• From 16 lines × 16 pixels to 1020 lines × 1016 pixels
	Data format	• RGB888 (parallel 24 bits) • RGB666 (parallel 18 bits) • RGB565 (parallel 16 bits) • RGB888 (serial 8 bits); clock cycle is four times the pixel clock • Bit endian order change and B/R signal swap
	Dither processing	• Reduces 10-bit signals (output after panel-oriented correction) to 8-, 6-, or 5-bit signals (output data format) • Supports the following modes: - Truncate mode - Round-off mode - 2×2 pattern dither mode

Table 58.2 Functional overview of GLCDC (2 of 2)

Parameter		Function
Timing control signal	Signal generation	5 timing signals (STVA, STVB, STHA, STHB, and DE) can be generated from HS / VS: • Vertical synchronization signal (variable) • Horizontal synchronization signal (variable) • Data enable signal
	Signal select	• Signals generated by the signal generation circuit can be output from LCD_TCONn pins (n = 0 to 3)
Output control panel-oriented correction processing	Brightness and contrast	• 10-bit internal processing; the sequence of this processing and gamma correction can be swapped. • Brightness: DC value adjustment range: from -512[LSB] to +512[LSB] • Contrast: gain value adjustment range: from 0 to 2 times (from 0/128 to 255/128)
	Gamma correction	• Sixteen areas; input/output: 10 bits • Gain value adjustment range in the area: From 0 to 2 times (from 0/1024 to 2047/1024)
Interrupts		• Number of specified lines interrupt (GLCDC_VPOS) • Graphics 1 buffer underflow interrupt (GLCDC_L1UNDF) • Graphics 2 buffer underflow interrupt (GLCDC_L2UNDF)

58.1.1 GLCDC Configuration

Figure 58.2 shows the configuration of the GLCDC.

The GLCDC includes the following blocks:

- Background screen generation block: Generates the background screen (including the blanking interval), selects the background color, and generates the synchronization signals for controlling the screens.
- Graphics data interface blocks (2 blocks): Convert the graphics data/CLUT data read through the GPX bus into ARGB (8888) data for internal processing, and transfer the clocks (PCLKA → PXCLK).
- Alpha blending blocks (2 blocks): Superimpose graphics data on the lower-layer screen and perform alpha blending based on the register settings and the alpha blending values for the current screen graphics data.
- TCON block: Generates the vertical and horizontal synchronization and enable signals suited for the specifications of the connected panel, from the internal vertical and horizontal synchronization signals.
- Data format conversion block: Processes data into the specific internal RGB888 format into the data of the specific format suited for the specifications of the panel with dither correction for output image data length.
- Panel-oriented correction processing block: Corrects brightness and contrast, and performs gamma correction suited for the characteristics of the connected panel, allowing either brightness and contrast correction or gamma correction to be performed first.
- Clock generator block: Generates the pixel and the panel clocks on a specific frequency from either LCD_EXTCLK or PLL output.

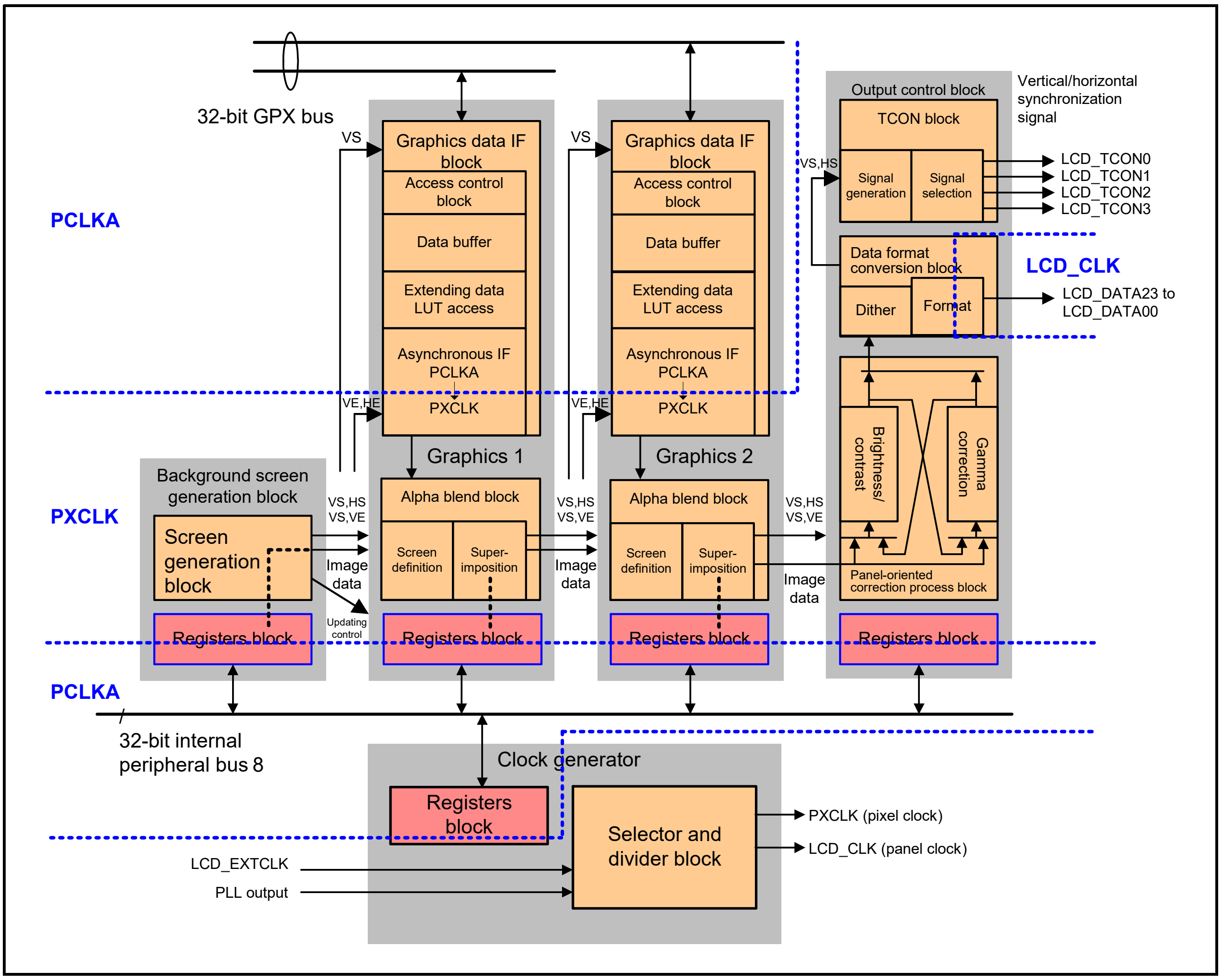


Figure 58.2 GLCDC configuration

58.1.2 Screen Format

Figure 58.3 shows the screen format overview of the GLCDC.

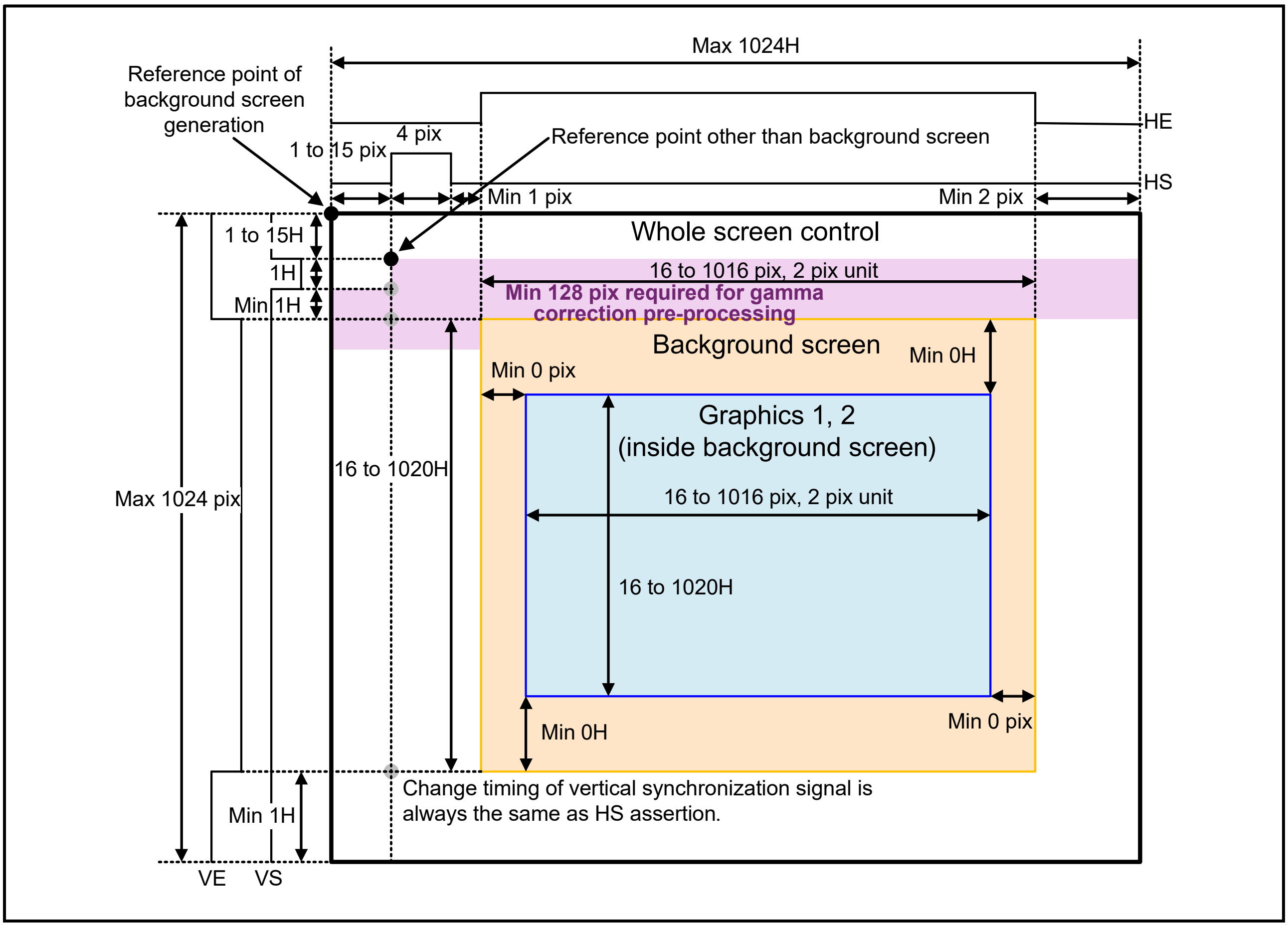


Figure 58.3 Internal screen format

The background screen generation block generates the essential timing signals for operations in the module as a whole. The graphics data interface blocks, alpha blending blocks, gamma correction block, output control block, and TCON block operate based on the synchronization and enable signals, which are sequentially transferred from the background screen generation block.

58.1.3 Graphics and Color Palette (CLUT) Data Formats

The GLCDC handles three display screens, one of which is a background screen. For the background screen, frame data is RGB888 graphics data stored in the registers, and for the other two screens, frame data is stored in the memory connected to the GPX bus as 32- or 16-bit/pixel graphics data or 8-, 4-, or 1-bit/pixel color palette (CLUT) data. The frame data of the relevant screen is read by the graphics data interface block, read into the GLCDC, and extended (converted) into ARGB8888 data for superimposition and blending. Figure 58.4 shows the frame data formats.

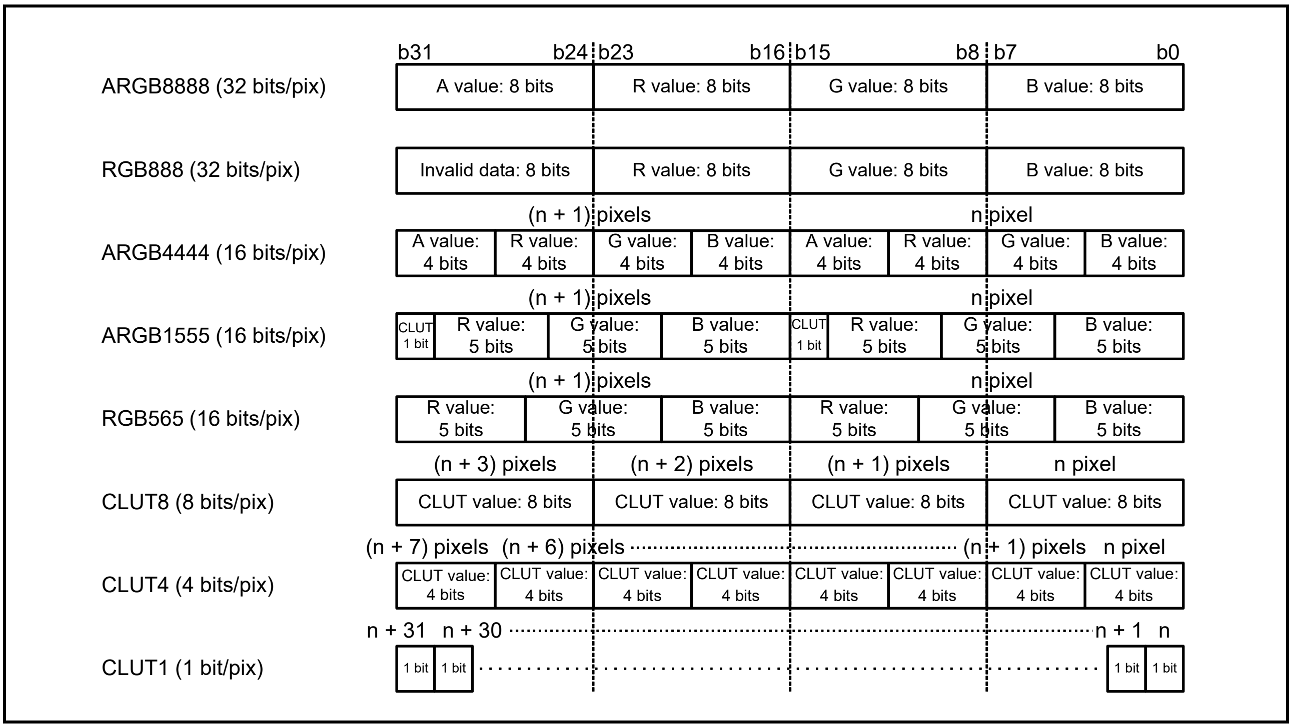


Figure 58.4 Frame data format

The A value (alpha blending value) represents the blending ratio of pixel data for displaying the lower-layer and current graphics screens after superimposition. The blending process is performed in accordance with the equations given in section 58.1.7, Graphics Data Interface. The CLUT value represents the color palette memory address (plane 0 or 1 is selected in the registers), and the data in the color palette is ARGB8888 (32 bits/pixel) graphics data. Addresses 01h/00h are accessed with CLUT1, 0Fh to 00h with CLUT4, and FFh to 00h with CLUT8. Addresses 80h/00h are accessed with ARGB1555 data.

58.1.4 Output Control for Data Format

The GLCDC can output data in accordance with the following formats and register settings:

- Data formats
  - Parallel: RGB888, RGB666, and RGB565
  - Serial: RGB888
  - Lower bits are processed in one of the following modes when 10-bit signals are reduced to 8-, 6-, and 5- bit signals:

    — Truncate mode

    — Round-off mode

    — 2×2 pattern dither processing + truncate mode.
- Pixel arrangement
  - RGB
  - BGR.
- Scan direction select for serial RGB888
  - Forward scan
  - Reverse scan.
- Data output delay of serial RGB888

  - 0 to 3 clock cycles.
- Pin assignment.
  - Little endian
  - Big endian.

For details on dither processing, see section 58.2.48, Output Control Block Panel Dither Correction Register (OUT_PDTHA).

Bit assignment of LCD signals for parallel RGB888 format

Table 58.3 shows RGB signal inputs assigned to the LCD signal outputs for the parallel RGB888 output format.

Table 58.3 Bit assignment of RGB signal inputs for parallel RGB888 format

	Pixel arrangement			
Pin assignment	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA23	R[7]	R[0]	B[7]	B[0]
LCD_DATA22	R[6]	R[1]	B[6]	B[1]
LCD_DATA21	R[5]	R[2]	B[5]	B[2]
LCD_DATA20	R[4]	R[3]	B[4]	B[3]
LCD_DATA19	R[3]	R[4]	B[3]	B[4]
LCD_DATA18	R[2]	R[5]	B[2]	B[5]
LCD_DATA17	R[1]	R[6]	B[1]	B[6]
LCD_DATA16	R[0]	R[7]	B[0]	B[7]
LCD_DATA15	G[7]	G[0]	G[7]	G[0]
LCD_DATA14	G[6]	G[1]	G[6]	G[1]
LCD_DATA13	G[5]	G[2]	G[5]	G[2]
LCD_DATA12	G[4]	G[3]	G[4]	G[3]
LCD_DATA11	G[3]	G[4]	G[3]	G[4]
LCD_DATA10	G[2]	G[5]	G[2]	G[5]
LCD_DATA09	G[1]	G[6]	G[1]	G[6]
LCD_DATA08	G[0]	G[7]	G[0]	G[7]
LCD_DATA07	B[7]	B[0]	R[7]	R[0]
LCD_DATA06	B[6]	B[1]	R[6]	R[1]
LCD_DATA05	B[5]	B[2]	R[5]	R[2]
LCD_DATA04	B[4]	B[3]	R[4]	R[3]
LCD_DATA03	B[3]	B[4]	R[3]	R[4]
LCD_DATA02	B[2]	B[5]	R[2]	R[5]
LCD_DATA01	B[1]	B[6]	R[1]	R[6]
LCD_DATA00	B[0]	B[7]	R[0]	R[7]

Note: R[7:0], G[7:0], and B[7:0] are RGB pixel data that is internally processed.

Bit assignment of LCD signals for parallel RGB666 format

Table 58.4 shows RGB signal inputs assigned to the LCD signal outputs for the parallel RGB666 output format.

Table 58.4 Bit assignment of RGB signal inputs for parallel RGB666 format (1 of 2)

	Pixel arrangement			
Pin assignment	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

Table 58.4 Bit assignment of RGB signal inputs for parallel RGB666 format (2 of 2)

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	R[7]	R[2]	B[7]	B[2]
LCD_DATA16	R[6]	R[3]	B[6]	B[3]
LCD_DATA15	R[5]	R[4]	B[5]	B[4]
LCD_DATA14	R[4]	R[5]	B[4]	B[5]
LCD_DATA13	R[3]	R[6]	B[3]	B[6]
LCD_DATA12	R[2]	R[7]	B[2]	B[7]
LCD_DATA11	G[7]	G[2]	G[7]	G[2]
LCD_DATA10	G[6]	G[3]	G[6]	G[3]
LCD_DATA09	G[5]	G[4]	G[5]	G[4]
LCD_DATA08	G[4]	G[5]	G[4]	G[5]
LCD_DATA07	G[3]	G[6]	G[3]	G[6]
LCD_DATA06	G[2]	G[7]	G[2]	G[7]
LCD_DATA05	B[7]	B[2]	R[7]	R[2]
LCD_DATA04	B[6]	B[3]	R[6]	R[3]
LCD_DATA03	B[5]	B[4]	R[5]	R[4]
LCD_DATA02	B[4]	B[5]	R[4]	R[5]
LCD_DATA01	B[3]	B[6]	R[3]	R[6]
LCD_DATA00	B[2]	B[7]	R[2]	R[7]

Note: R[7:2], G[7:2], and B[7:2] are RGB pixel data that is internally processed.

Bit assignment of LCD signals for parallel RGB565 format

Table 58.5 shows RGB signal inputs assigned to the LCD signal outputs for the parallel RGB565 output format.

Table 58.5 Bit assignment of RGB signal inputs for parallel RGB565 format (1 of 2)

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA16	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA15	R[7]	R[3]	B[7]	B[3]
LCD_DATA14	R[6]	R[4]	B[6]	B[4]
LCD_DATA13	R[5]	R[5]	B[5]	B[5]
LCD_DATA12	R[4]	R[6]	B[4]	B[6]
LCD_DATA11	R[3]	R[7]	B[3]	B[7]

Table 58.5 Bit assignment of RGB signal inputs for parallel RGB565 format (2 of 2)

Pin assignment	Pixel arrangement			
	RGB, little endian	RGB, big endian	BGR, little endian	BGR, big endian
LCD_DATA10	G[7]	G[2]	G[7]	G[2]
LCD_DATA09	G[6]	G[3]	G[6]	G[3]
LCD_DATA08	G[5]	G[4]	G[5]	G[4]
LCD_DATA07	G[4]	G[5]	G[4]	G[5]
LCD_DATA06	G[3]	G[6]	G[3]	G[6]
LCD_DATA05	G[2]	G[7]	G[2]	G[7]
LCD_DATA04	B[7]	B[3]	R[7]	R[3]
LCD_DATA03	B[6]	B[4]	R[6]	R[4]
LCD_DATA02	B[5]	B[5]	R[5]	R[5]
LCD_DATA01	B[4]	B[6]	R[4]	R[6]
LCD_DATA00	B[3]	B[7]	R[3]	R[7]

Note: R[7:3], G[7:2], and B[7:3] are RGB pixel data that is internally processed.

Bit assignment of RGB signal inputs for serial RGB888 format

Table 58.6 and Table 58.7 show RGB signal inputs assigned to the LCD signal outputs for the serial RGB888 output format.

Table 58.6 Bit assignment of RGB signal inputs for serial RGB888 format, RGB arrangement

Pin assignment	Pixel arrangement, scan direction select, and cycle															
	RGB, little endian								RGB, big endian							
	Reverse scan				Forward scan				Reverse scan				Forward scan			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
LCD_DATA23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	⋮				⋮				⋮				⋮			
LCD_DATA08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LCD_DATA07	Undefined	B[7]	G[7]	R[7]	R[7]	G[7]	B[7]	Undefined	Undefined	B[0]	G[0]	R[0]	R[0]	G[0]	B[0]	Undefined
LCD_DATA06	Undefined	B[6]	G[6]	R[6]	R[6]	G[6]	B[6]	Undefined	Undefined	B[1]	G[1]	R[1]	R[1]	G[1]	B[1]	Undefined
LCD_DATA05	Undefined	B[5]	GI5]	R[5]	R[5]	GI5]	B[5]	Undefined	Undefined	B[2]	G[2]	R[2]	R[2]	G[2]	B[2]	Undefined
LCD_DATA04	Undefined	B[4]	G[4]	R[4]	R[4]	G[4]	B[4]	Undefined	Undefined	B[3]	G[3]	R[3]	R[3]	G[3]	B[3]	Undefined
LCD_DATA03	Undefined	B[3]	G[3]	R[3]	R[3]	G[3]	B[3]	Undefined	Undefined	B[4]	G[4]	R[4]	R[4]	G[4]	B[4]	Undefined
LCD_DATA02	Undefined	B[2]	G[2]	R[2]	R[2]	G[2]	B[2]	Undefined	Undefined	B[5]	G[5]	R[5]	R[5]	G[5]	B[5]	Undefined
LCD_DATA01	Undefined	B[1]	G[1]	R[1]	R[1]	G[1]	B[1]	Undefined	Undefined	B[6]	G[6]	R[6]	R[6]	G[6]	B[6]	Undefined
LCD_DATA00	Undefined	B[0]	G[0]	R[0]	R[0]	G[0]	B[0]	Undefined	Undefined	B[7]	G[7]	R[7]	R[7]	G[7]	B[7]	Undefined

Note: R[7:0], G[7:0], and B[7:0] are RGB pixel data that is internally processed.

Table 58.7 Bit assignment of RGB signal inputs for serial RGB888 format, BGR arrangement

Pin assignment	Pixel arrangement, scan direction select, and cycle															
	BGR, little endian								BGR, big endian							
	Reverse scan				Forward scan				Reverse scan				Forward scan			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
LCD_DATA23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮	⋮				⋮				⋮				⋮			
LCD_DATA08	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LCD_DATA07	Undefined	R[7]	G[7]	B[7]	B[7]	G[7]	R[7]	Undefined	Undefined	R[0]	G[0]	B[0]	B[0]	G[0]	R[0]	Undefined
LCD_DATA06	Undefined	R[6]	G[6]	B[6]	B[6]	G[6]	R[6]	Undefined	Undefined	R[1]	G[1]	B[1]	B[1]	G[1]	R[1]	Undefined
LCD_DATA05	Undefined	R[5]	GI5]	B[5]	B[5]	GI5]	R[5]	Undefined	Undefined	R[2]	G[2]	B[2]	B[2]	G[2]	R[2]	Undefined
LCD_DATA04	Undefined	R[4]	G[4]	B[4]	B[4]	G[4]	R[4]	Undefined	Undefined	R[3]	G[3]	B[3]	B[3]	G[3]	R[3]	Undefined
LCD_DATA03	Undefined	R[3]	G[3]	B[3]	B[3]	G[3]	R[3]	Undefined	Undefined	R[4]	G[4]	B[4]	B[4]	G[4]	R[4]	Undefined
LCD_DATA02	Undefined	R[2]	G[2]	B[2]	B[2]	G[2]	R[2]	Undefined	Undefined	R[5]	G[5]	B[5]	B[5]	G[5]	R[5]	Undefined
LCD_DATA01	Undefined	R[1]	G[1]	B[1]	B[1]	G[1]	R[1]	Undefined	Undefined	R[6]	G[6]	B[6]	B[6]	G[6]	R[6]	Undefined
LCD_DATA00	Undefined	R[0]	G[0]	B[0]	B[0]	G[0]	R[0]	Undefined	Undefined	R[7]	G[7]	B[7]	B[7]	G[7]	R[7]	Undefined

Note: R[7:0], G[7:0], and B[7:0] are RGB pixel data that is internally processed.

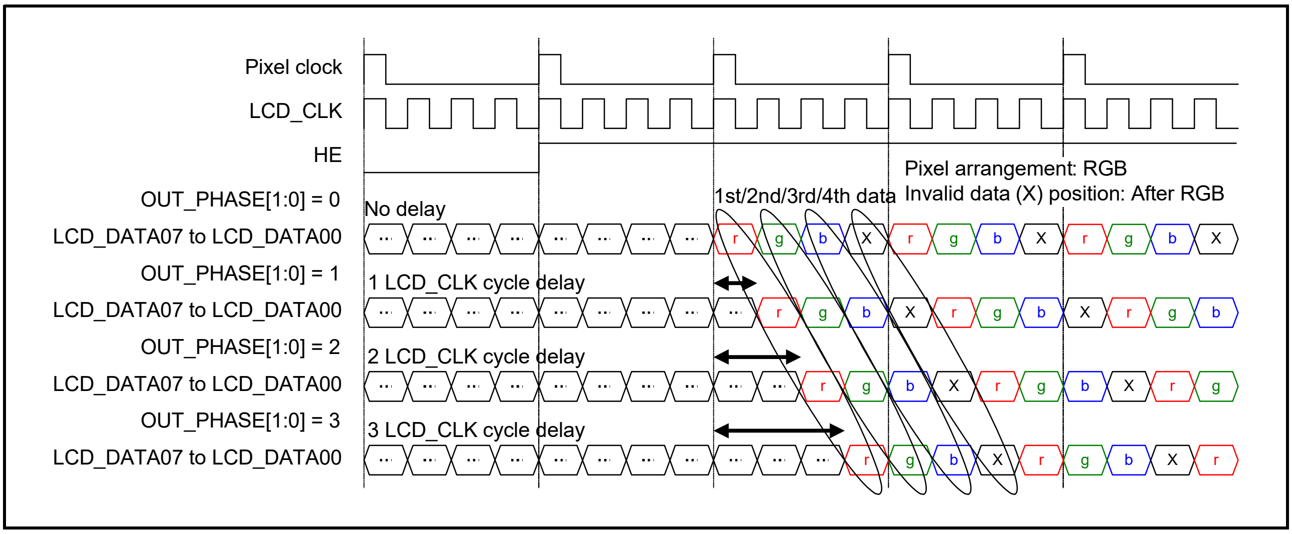


Figure 58.5 Serial RGB888 (4x speed) output timing

58.1.5 Output Control for Panel-Oriented Correction Process

The following panel-oriented correction processes are provided:

- Brightness correction
- Contrast correction
- RGB gamma correction.

Brightness correction always precedes contrast correction, but RGB gamma correction can either precede or follow brightness and contrast correction, based on the register settings. In panel-oriented correction, 10-bit RGB data obtained by extending 8-bit RGB data output from graphics 2 is used, and 10-bit RGB data is also output to the output control (data format conversion) block. Figure 58.6 shows the configuration of the panel-oriented correction circuit.

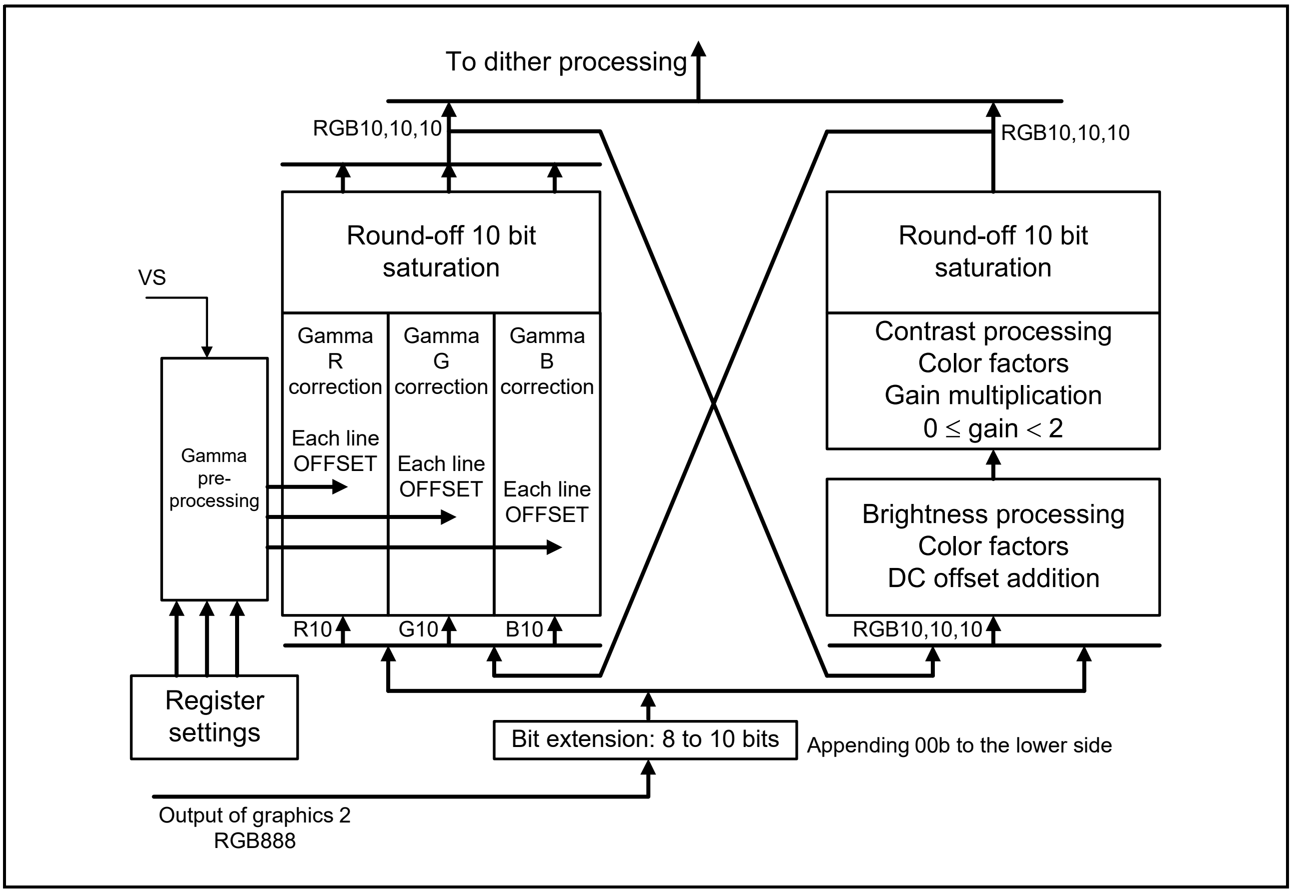


Figure 58.6 Configuration of panel-oriented correction circuit

58.1.6 Output Control for TCON

Any of the four signals generated from the internal vertical and horizontal synchronization signals (STVA, STVB, STHA, and STHB) that have passed through the data format conversion block can be selected for output on four pins, LCD_TCON0, LCD_TCON1, LCD_TCON2, and LCD_TCON3. The generated signals are completely independent of the image data. They are not affected by the output image format or any internal data process, and no register settings for signal generation affect the output image format or any internal data process. The data enable signal DE, which is to be generated by the TCON block, is the logical AND of the two signals STVB and STHB, which were previously generated by the TCON block. Consequently, three signals in total can actually be generated if DE is to be output.

58.1.7 Graphics Data Interface

Two circuit systems are provided for reading graphics data (graphics 1 and 2), each of which incorporates an access control block, data buffer, CLUT memory, data extension block, and asynchronous interface block.

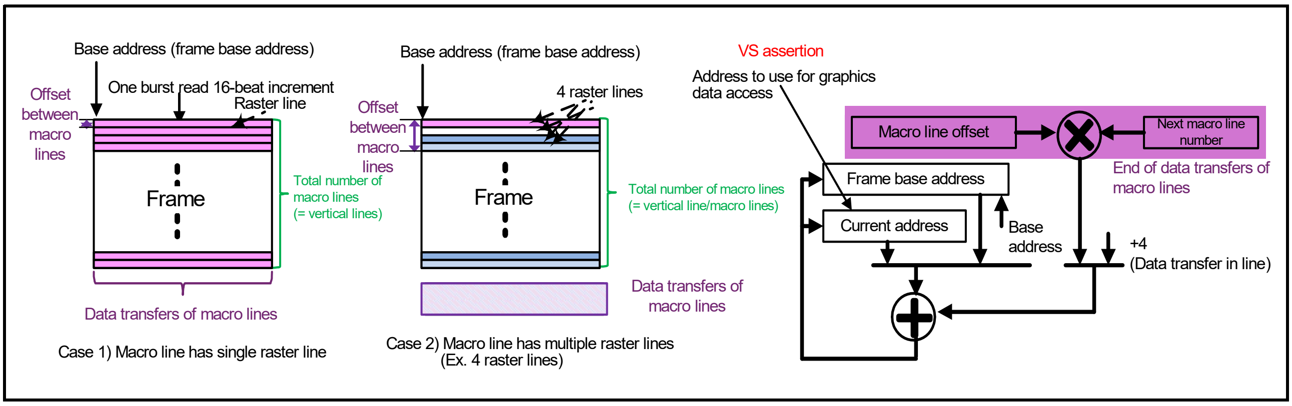


Figure 58.7 Calculation of graphics data access address

Graphics (or CLUT) data is accessed and output to the pixel operation block as ARGB8888 data (32 bits/pixel).

The GPX bus is accessed in word (32-bit) units in 16-beat increment burst reads, in accordance with the preset parameters in the two-dimensional addressing mode, in which the macro line structure is accounted for as shown in Figure 58.7, and data is stored in the data buffer. Even if invalid data is at the macro line end, all the data is stored in the data buffer, and the invalid data is skipped when data is read internally. Regardless of the format of the graphics data, data is extended to ARGB8888 data before being output to the alpha blending blocks.

58.1.8 Blending

The following processes are performed for the graphics areas specified in the registers. The lower-layer graphics plane is output without any processing to the display area outside the graphics area.

- Display plane selection
- RGB-index chroma-key
- Alpha blending.

Figure 58.8 shows the relationship between the graphics display selection and rectangular alpha blending area.

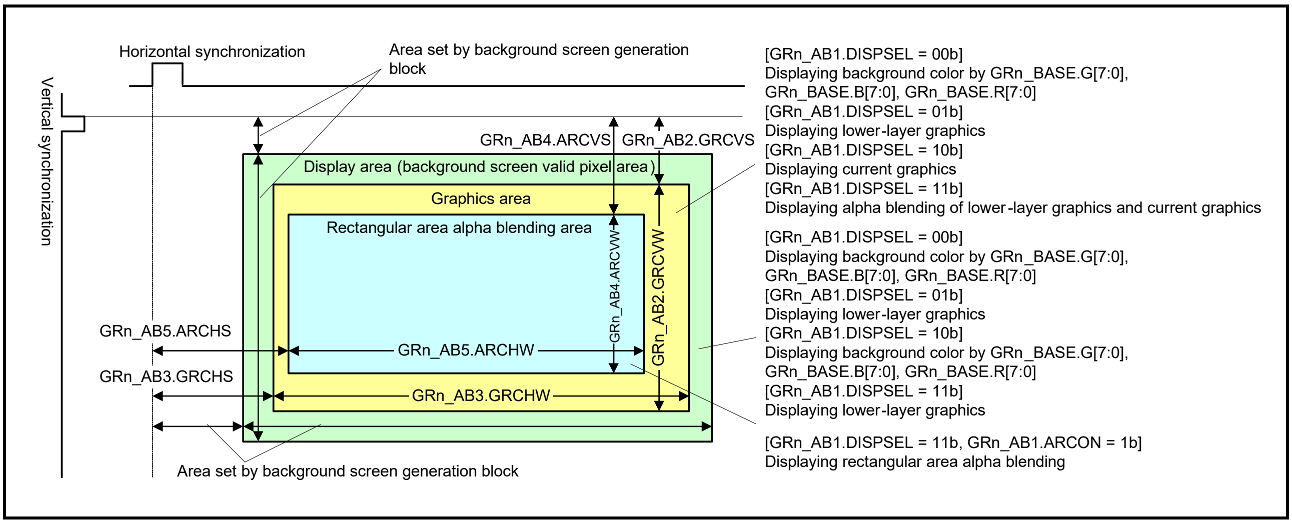


Figure 58.8 Graphics display selection

(1) Display plane selection

Based on the register settings, the following data is output to the graphics area:

- Background color:
  - RGB data specified in the registers.

- Lower-layer graphics:
  - RGB data input from the previous module
  - Output from the background screen generation block for graphics 1
  - Output from graphics 1 for graphics 2.
- Current graphics:
  - RGB data obtained by the graphics data interface block extending the graphics data read by the graphics 1 or 2 module through the GPX bus or CLUT data to ARGB8888 format data.
- Blending:
  - RGB data obtained by blending the lower-layer graphics data and current graphics data obtained by the graphics data interface block extension to ARGB8888 format data, based on the alpha blending values or the register settings.

It is possible to modify the register values related to these functions while display operation is in progress, and to allow reflection of the updates to the internal operations when the VS (vertical synchronization signal) is asserted, if reflection of the register settings to the internal operations on vertical synchronization is enabled.

(2) RGB-index chroma-key

If the RGB value of the ARGB8888 data input from the graphics data interface block agrees with the value set in the GRn_AB8 register, the ARGB8888 data is entirely replaced with the value preset in the GRn_AB9 register. All the ARGB8888 data input from the graphics data interface block is subject to this process. If your application excludes CLUT memory output from this process, you must disable the corresponding process in GRn_AB7.CKON.

(3) Alpha blending

If blending is selected in the selected display plane, lower-layer graphics and current graphics are alpha-blended based on the register settings using either of the following two functions.

(a) Alpha-blending in a rectangular area

The following process is performed for the rectangular area (in the graphics area) preset in the registers:

1. Lower-layer graphics and current graphics planes are blended in accordance with the A value set in the registers.
2. After the number of frames set in the registers passes, the A value is updated to the A value + Δ (register setting).
3. The lower-layer graphics and current graphics planes are blended in accordance with this updated A value.
4. The process of updating the A value after the number of frames set in the registers passes is repeated (A value: min/max value saturation).

(b) Alpha-blending in pixel units

The lower-layer graphics and current graphics planes are blended in accordance with the A value of the ARGB8888 data input from the graphics data interface block.

Figure 58.9 shows the update of the alpha blending value in the rectangular alpha blending area.

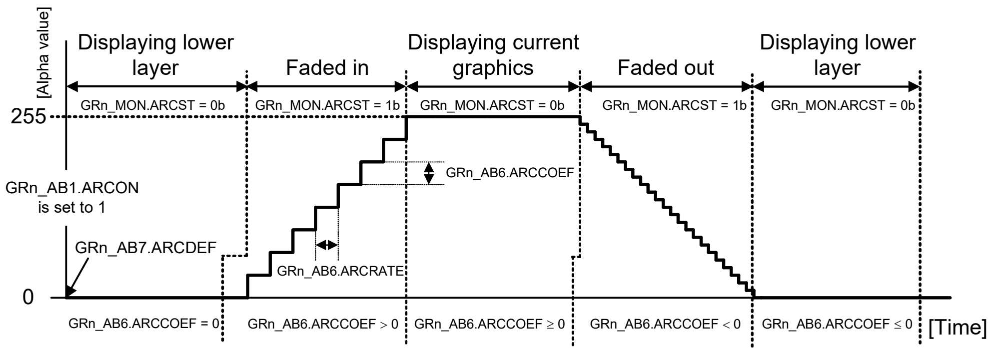


Figure 58.9 Updating of alpha blending value

Alpha blending is based on the following formulas:

When A value = 255
Rout/Gout/Bout = current graphics data
When A value $\neq$ 255
$\mathrm{Rout} = (\mathrm{Rin1} \times A + \mathrm{Rin0} \times (256 - A))/256$
$\mathrm{Gout} = (\mathrm{Gin1} \times A + \mathrm{Gin0} \times (256 - A))/256$
$\mathrm{Bout} = (\mathrm{Bin1} \times A + \mathrm{Bin0} \times (256 - A))/256$

where,

A: alpha blending value
Rin1/Gin1/Bin1: current graphics data
Rin0/Gin0/Bin0: lower-layer graphics data

58.2 Register Descriptions

58.2.1 Color Palette (CLUT)

- Color Palette 0 Plane for Graphics 1 Plane 0 to Color Palette 0 Plane for Graphics 1 Plane 255 (GR1_CLUT0[0] to GR1_CLUT0[255])
  Address(es): GLCDC.GR1_CLUT0[0] 400E 0000h to GLCDC.GR1_CLUT0[255] 400E 03FCh
- Color Palette 1 Plane for Graphics 1 Plane 0 to Color Palette 1 Plane for Graphics 1 Plane 255 (GR1_CLUT1[0] to GR1_CLUT1[255])
  Address(es): GLCDC.GR1_CLUT1[0] 400E 0400h to GLCDC.GR1_CLUT1[255] 400E 07FCh
- Color Palette 0 Plane for Graphics 2 Plane 0 to Color Palette 0 Plane for Graphics 2 Plane 255 (GR2_CLUT0[0] to GR2_CLUT0[255])
  Address(es): GLCDC.GR2_CLUT0[0] 400E 0800h to GLCDC.GR2_CLUT0[255] 400E 0BFCh
- Color Palette 1 Plane for Graphics 2 Plane 0 to Color Palette 1 Plane for Graphics 2 Plane 255 (GR2_CLUT1[0] to GR2_CLUT1[255])
  Address(es): GLCDC.GR2_CLUT1[0] 400E 0C00h to GLCDC.GR2_CLUT1[255] 400E 0FFCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	A[7:0]								R[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	G[7:0]								B[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	B[7:0]	B Value of Color Palette n Plane for Graphics m Plane	B value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W

Bit	Symbol	Bit name	Description	R/W
b15 to b8	G[7:0]	G Value of Color Palette n Plane for Graphics m Plane	G value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W
b23 to b16	R[7:0]	R Value of Color Palette n Plane for Graphics m Plane	R value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W
b31 to b24	A[7:0]	Alpha Blending Value of Color Palette n Plane for Graphics m Plane	Alpha blending value of color palette n plane for graphics m plane. Unsigned 8-bit integer.	R/W

B[7:0] bits (B Value of Color Palette n Plane for Graphics m Plane)

The B[7:0] bits set the B value when this color palette is used.

G[7:0] bits (G Value of Color Palette n Plane for Graphics m Plane)

The G[7:0] bits set the G value when this color palette is used.

R[7:0] bits (R Value of Color Palette n Plane for Graphics m Plane)

The R[7:0] bits set the R value when this color palette is used.

A[7:0] bits (Alpha Blending Value of Color Palette n Plane for Graphics m Plane)

The A[7:0] bits set the alpha blending value when this color palette is used.

All the planes can always be accessed through the register access bus (internal peripheral bus 8), regardless of the plane specified to be used by the graphics data access block. The updates are reflected to the internal operations directly, not in synchronization with the vertical synchronization signal. To keep reflection of the CLUT memory contents to the internal operations in synchronization with the vertical synchronization signal, first write data through the register access bus to the plane that is not being used for internal operations, and then modify the GRn_CLUTINT.SEL bit for controlling the plane that is to be used.

58.2.2 Background Plane Setting Operation Control Register (BG_EN)

Address(es): GLCDC.BG_EN 400E 1000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VEN	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EN	Background Plane Operation Enable	0: Disable background plane operation 1: Enable background plane operation.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	VEN	Control of GLCDC Internal Register Value Reflection to Internal Operations	0: Disable GLCDC register values from being reflected in internal operations at start of screen generation 1: Enable GLCDC register values to be reflected in internal operations at start of screen generation. This bit is cleared to 0 by an internal source.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SWRST	Software Reset Control	0: Place entire module in software reset state 1: Release entire module from software reset state.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

EN bit (Background Plane Operation Enable)

The EN bit enables or disables the operation of the background plane generation module. When the operation is stopped (this bit is cleared to 0) after the operation is enabled (set to 1), the operation is stopped at the end of screen generation, unlike when a software reset occurs.

When setting this bit to 1, set the BG_EN.VEN bit to 1 simultaneously to enable register value reflection to internal operations. When clearing this bit to 0, also clear the BG_EN.VEN bit to 0 simultaneously. Before clearing this bit to 0, confirm that the BG_MON.VEN bit is cleared to 0, to make sure that the signal for controlling reflection of the register values to internal operations is negated. Operation is not guaranteed if this bit and the BG_EN.VEN bit are set to 1, or if the settings in other registers are modified before the BG_MON.EN bit is cleared to 0.

VEN bit (Control of GLCDC Internal Register Value Reflection to Internal Operations)

The VEN bit enables or disables reflection of the GLCDC internal register to the GLCDC internal operation on assertion of the vertical synchronization signal (input). When this bit is set to 1, the signal of GLCDC internal register values reflection control is set to 1 at the immediate start of the screen, and are automatically cleared to 0 at the end of the vertical valid pixel of the same screen. Set this bit to 1 only when 0. Operation is not guaranteed if this bit is set to 1 when 1. Also, while this bit is 1, do not modify any register whose value is reflected to the internal operations at the start of the screen in the GLCDC or on assertion of the vertical synchronization signal (input). Otherwise, operation is not guaranteed.

SWRST bit (Software Reset Control)

The SWRST bit controls a software reset of the entire GLCDC, not only the background plane generation module. When this bit is cleared to 0, the GLCDC enters the reset state from any operation state. This bit must be set to 1 before the registers are set or operation enabled. Although the registers (except the CLUT memory and the some of the operation control registers) can be set while this bit is set to 1 (immediately after), before accessing the CLUT memory, enabling operation, or reflecting the register values to the internal operation by the vertical synchronization signal, confirm that PXCLK/LCD_CLK and PCLKA are supplied and that the BG_MON.SWRST bit, which monitors the entire module software reset state, is set to 1. Operation is not guaranteed if the software accesses CLUT memory, enables operation, or reflects the register vales to the internal operation on the vertical synchronization signal while PXCLK/LCD_CLK and PCLKA are not supplied or the BG_MON.SWRST bit is not set to 1. The peripheral bus clock (PCLKA) must be supplied to the GLCDC when this bit is used to apply or cancel a software reset. If PCLKA is not supplied, writing to this bit is impossible.

The bits in this register control the GLCDC states. The internal states can be read from the associated bits in the status monitor registers and BG_EN.VEN (this register). Because the GLCDC, which operates on multiple clock signals, requires a certain period for state transition, you must confirm that the internal state has stabilized (state transition is complete) when settings are modified. Operation is not guaranteed if settings are modified again before the internal state stabilizes (state transition is complete). However, clearing a software reset to 0 immediately makes the whole GLCDC reset, setting it to 1 releases the reset state. These operations do not require the clock supply of PXCLK.

58.2.3 Background Plane Setting Free-Running Period Register (BG_PERI)

Address(es): GLCDC.BG_PERI 400E 1004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	FV[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FH[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b10 to b0	FH[10:0]	Background Plane Horizontal Synchronization Signal Period	Period based on pixel clocks (PXCLK). 017h: 24 cycles (pixels) 3FFh: 1024 cycles (pixels). Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	FV[10:0]	Background Plane Vertical Synchronization Signal Period	Period based on lines. 013h: 20 lines 3FFh: 1024 lines. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

FH[10:0] bits (Background Plane Horizontal Synchronization Signal Period)

The FH[10:0] bits set the horizontal synchronization signal period of the background plane. This field contains 11 bits and can be set to any number from 000h to 7FFh. However, the valid range is 017h to 3FFh. Operation is not guaranteed if a value outside the valid range is set.

FV[10:0] bits (Background Plane Vertical Synchronization Signal Period)

The FV[10:0] bits set the vertical synchronization signal period of the background plane. This field contains 11 bits and can be set to any number from 000h to 7FFh. However, the valid range is 013h to 3FFh. Operation is not guaranteed if a value outside the valid range is set.

58.2.4 Background Plane Setting Synchronization Position Register (BG_SYNC)

Address(es): GLCDC.BG_SYNC 400E 1008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	VP[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	HP[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HP[3:0]	Background Plane Horizontal Synchronization Signal Assertion Position	Position based on pixel clocks (PXCLK). 0h: Setting prohibited 1h: 1st cycle (pixel) : Fh: 15th cycle (pixel).	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b19 to b16	VP[3:0]	Background Plane Vertical Synchronization Assertion Position	Position based on lines. 0h: Setting prohibited 1h: 1st line : Fh: 15th line.	R/W
b31 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

HP[3:0] bits (Background Plane Horizontal Synchronization Signal Assertion Position)

The HP[3:0] bits set the horizontal synchronization signal assertion position of the background plane. The signal is held asserted for a 4-pixel clock width.

VP[3:0] bits (Background Plane Vertical Synchronization Assertion Position)

The VP[3:0] bits set the vertical synchronization signal assertion position of the background plane. The signal is held asserted for a 1H width, and the assertion timing within a single horizontal line is specified in BG_SYNC.HP[3:0].

58.2.5 Background Plane Setting Full Image Vertical Size Register (BG_VSIZE)

Address(es): GLCDC.BG_VSIZE 400E 100Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	VP[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	VW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	VW[10:0]	Background Plane Vertical Valid Pixel Width	Width based on lines. 010h: 16 lines : 3FCh: 1020 lines. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	VP[10:0]	Background Plane Vertical Valid Pixel Start Position	Position based on of lines. 003h: 3rd line : 3EFh: 1007th lines Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

VW[10:0] bits (Background Plane Vertical Valid Pixel Width)

The VW[10:0] bits set the vertical valid pixel width of the background plane. This field contains 11 bits and can be set to any number from 000h to 7FFh. However, the valid range is 010h to 3FCh. Operation is not guaranteed if a value outside the valid range is set.

VP[10:0] bits (Background Plane Vertical Valid Pixel Start Position)

The VP[10:0] bits set the vertical valid pixel start position of the background plane. This field contains 11 bits and can be set to any number from 000h to 7FFh. However, the valid range is 003h to 3EFh. Operation is not guaranteed if a value outside the valid range is set.

Specify the vertical valid pixel area between the assertion position of the vertical synchronization signal + 2 and the (background plane end - 1)th line. Operation is not guaranteed if the area is specified outside this range.

58.2.6 Background Plane Setting Full Image Horizontal Size Register (BG_HSIZE)

Address(es): GLCDC.BG_HSIZE 400E 1010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	HP[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	HW[10:0]	Background Plane Horizontal Valid Pixel Width	Width based on pixel clocks (PXCLK). 010h: 16 cycles : 3F8h: 1016 cycles. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	HP[10:0]	Background Plane Horizontal Valid Pixel Start Position	Position based on pixel clocks (PXCLK). 006h: 6th cycle (pixel) : 3EEh: 1006th cycle (pixel). Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

HW[10:0] bits (Background Plane Horizontal Valid Pixel Width)

The HW[10:0] bits set the horizontal valid pixel width of the background plane. This field contains 11 bits and can be set to any number from 000h to 7FFh. However, the valid range is 010h to 3F8h. Operation is not guaranteed if a value outside the valid range is set. When serial RGB is selected as the output format for the output control block, add two to the horizontal valid pixel width and set the resulting value to these bits.

HP[10:0] bits (Background Plane Horizontal Valid Pixel Start Position)

The HP[10:0] bits set the horizontal valid pixel start position of the background plane. This field contains 11 bits and can be set to any number from 000h to 7FFh. However, the valid range is 006h to 3EEh. Operation is not guaranteed if a value outside the valid range is set.

Specify the horizontal valid pixel area between the assertion position of the horizontal synchronization signal + 5 and pixel number (line end - 2). Operation is not guaranteed if the area is specified outside this range.

The background plane generation module outputs the values that are specified in the Background Color Register (BG_BGC) for the area defined by the Full Image Vertical Size Register (BG_VSIZE) and Full Image Horizontal Size Register (BG_HSIZE), and it outputs 00h as the RGB values for the blanking interval area.

58.2.7 Background Plane Setting Background Color Register (BG_BGC)

Address(es): GLCDC.BG_BGC 400E 1014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	R[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	G[7:0]								B[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	B[7:0]	Background Plane Valid Pixel Area B Value	B value for background plane valid pixel area. Unsigned 8-bit integer.	R/W
b15 to b8	G[7:0]	Background Plane Valid Pixel Area G Value	G value for background plane valid pixel area. Unsigned 8-bit integer.	R/W
b23 to b16	R[7:0]	Background Plane Valid Pixel Area R Value	R value for background plane valid pixel area. Unsigned 8-bit integer.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

B[7:0] bits (Background Plane Valid Pixel Area B Value)

The B[7:0] bits set the B value of image data to be output to the valid pixel area of the background plane.

G[7:0] bits (Background Plane Valid Pixel Area G Value)

The G[7:0] bits set the G value of image data to be output to the valid pixel area of the background plane.

R[7:0] bits (Background Plane Valid Pixel Area R Value)

The R[7:0] bits set the R value of image data to be output to the valid pixel area of the background plane.

58.2.8 Background Plane Setting Status Monitor Register (BG_MON)

Address(es): GLCDC.BG_MON 400E 1018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VEN	—	—	—	—	—	—	—	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EN	Background Plane Operation Monitor	0: Operation is stopped 1: Operation is in progress.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

Bit	Symbol	Bit name	Description	R/W
b8	VEN	Entire Module Internal Operation Reflection Control Signal Monitor	0: Signal for controlling reflection of the register values to internal operations on assertion of vertical synchronization signal is negated 1: Signal for controlling reflection of the register values to internal operations on assertion of vertical synchronization signal is asserted.	R
b15 to b9	—	Reserved	These bits are read as 0.	R
b16	SWRST	Entire Module SW Reset State Monitor	0: Entire module is in software reset state 1: Entire module is released from software reset state.	R
b31 to b17	—	Reserved	These bits are read as 0.	R

EN bit (Background Plane Operation Monitor)

The EN bit indicates whether the background plane generation module is operating or not. To stop the operation of the background plane generation module by clearing BG_EN.EN to 0, read this bit to confirm that the operation of the background plane is complete. Clearing BG_EN.EN to 0 does not stop the operation until completion of the background plane, unlike when BG_EN.SWRST is cleared to 0.

VEN bit (Entire Module Internal Operation Reflection Control Signal Monitor)

The VEN bit indicates the value of the signal for controlling reflection of the GLCDC internal register values to the internal operations. This signal is asserted at the start of a screen immediately after setting BG_EN.VEN to 1 and negated at the VE negate timing output from the background screen generation block.

SWRST bit (Entire Module SW Reset State Monitor)

The SWRST bit indicates the software reset state of the entire module when PXCLK is supplied. This bit value indicates the result of the peripheral module clock A (PCLKA) resampling the result of the pixel clock (PXCLK) sampling the BG_EN.SWRST bit. Even if PXCLK is not supplied, clearing the BG_EN.SWRST to 0 clears this bit to 0.

58.2.9 Graphics 1 Register Update Control Register (GR1_VEN)
Graphics 2 Register Update Control Register (GR2_VEN)

Address(es): GLCDC.GR1_VEN 400E 1100h, GLCDC.GR2_VEN 400E 1200h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PVEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PVEN	Control of Graphics n Register Value Reflection to Internal Pixel Operations	0: Disable reflection of register values to internal operations on assertion of vertical synchronization signal (VS) 1: Enable reflection of register values to internal operations on assertion of the vertical synchronization signal (VS). This bit is cleared to 0 by an internal source.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PVEN bit (Control of Graphics n Register Value Reflection to Internal Pixel Operations)

The PVEN bit enables or disables reflection of the register values to the internal operations of the pixel operation circuit on assertion of the vertical synchronization signal (input). When this bit is set to 1, the register values are immediately reflected to the internal operations on assertion of the vertical synchronization signal (input), and are automatically cleared to 0. Set this bit to 1 only when 0. Operation is not guaranteed if this bit is set to 1 when 1. Also, if the signal output is asserted from the background plane generation module that controls reflection of the register values to the internal operations of all the modules, the register values are always reflected to the internal operations on assertion of the vertical synchronization signal (input), regardless of the value of this bit. While this bit is 1, do not modify any register whose value is reflected to the internal operations on assertion of the vertical synchronization signal (input) in this block. Otherwise, operation is not guaranteed.

This bit must not be 1 at the same time as the BG_EN.VEN bit (control of background plane register value reflection to internal operations) in the Operation Control Register (BG_EN), one of the background plane setting registers. Otherwise, operation is not guaranteed.

58.2.10 Graphics 1 Frame Buffer Read Control Register (GR1_FLMRD)
Graphics 2 Frame Buffer Read Control Register (GR2_FLMRD)

Address(es): GLCDC.GR1_FLMRD 400E 1104h, GLCDC.GR2_FLMRD 400E 1204h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RENB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RENB	Graphics Data Read Enable	0: Disable reading 1: Enable reading. Graphics data is the frame buffer data.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

RENB bit (Graphics Data Read Enable)

The RENB bit enables or disables reading of the graphics data (frame buffer data in memory connected through the GPX bus). If the current graphics data is required, data read must be enabled (GRn_FLMRD.RENB = 1) before the background panel operation is enabled, register value internal operation reflection control is enabled, or the register value internal operation reflection control of graphics 1 and 2 is enabled.

If the current graphics data is not required, data read must be disabled (GRn_FLMRD.RENB = 0) before the background panel operation is enabled, register value internal operation reflection control is enabled, or the register value internal operation reflection control of graphics 1 and 2 is enabled.

58.2.11 Graphics 1 Frame Buffer Control Register 1 (GR1_FLM1) Graphics 2 Frame Buffer Control Register 1 (GR2_FLM1)

Address(es): GLCDC.GR1_FLM1 400E 1108h, GLCDC.GR2_FLM1 400E 1208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BSTMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	BSTMD[1:0]	Burst Transfer Control for Graphics Data Access	b1 b0 0 0: Setting prohibited 0 1: Setting prohibited 1 0: Setting prohibited 1 1: 16-beat increment burst transfer (64-byte boundary). Graphics data is the frame buffer data.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BSTMD[1:0] bits (Burst Transfer Control for Graphics Data Access)

The BSTMD[1:0] bits control burst transfers for accessing the graphics data (frame buffer data in memory connected to memory through the GPX bus). In this GLCDC, these bits are fixed to 11b. Operation is not guaranteed if these bits are set to any other value.

58.2.12 Graphics 1 Frame Buffer Control Register 2 (GR1_FLM2) Graphics 2 Frame Buffer Control Register 2 (GR2_FLM2)

Address(es): GLCDC.GR1_FLM2 400E 110Ch, GLCDC.GR2_FLM2 400E 120Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BASE[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BASE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b31 to b0	BASE[31:0]	Base Address for Accessing Graphics Data	Start address in the frame buffer where graphics data is to be stored. Fix GRn_FLM2.BASE[5:0] to 0 during 64-byte burst transfer.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BASE[31:0] bits (Base Address for Accessing Graphics Data)

The BASE[31:0] bits specify the base address (start address in the first frame buffer) for graphics data access. Because the GLCDC only supports the 16-beat increment burst transfer mode, in which data is aligned with a 64-byte boundary, the lower 6 bits (GRn_FLM2.BASE[5:0]) must be fixed to 0.

58.2.13 Graphics 1 Frame Buffer Control Register 3 (GR1_FLM3) Graphics 2 Frame Buffer Control Register 3 (GR2_FLM3)

Address(es): GLCDC.GR1_FLM3 400E 1110h, GLCDC.GR2_FLM3 400E 1210h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LNOFF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	LNOFF[15:0]	Macro Line Offset Address for Accessing Graphics Data	Macro line offset address for accessing graphics data (frame buffer data). Signed, 16-bit integer.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

LNOFF[15:0] bits (Macro Line Offset Address for Accessing Graphics Data)

The LNOFF[15:0] bits specify the macro line offset address for accessing graphics data (offset to be added to the current address at the macro line end for calculating the start address of the next macro line). Because the GLCDC only supports the 16-beat increment burst transfer mode, in which data is aligned with a 64-byte boundary, the lower 6 bits (GRn_FLM3.LNOFF[5:0]) must be fixed to 0.

58.2.14 Graphics 1 Frame Buffer Control Register 5 (GR1_FLM5)
Graphics 2 Frame Buffer Control Register 5 (GR2_FLM5)

Address(es): GLCDC.GR1_FLM5 400E 1118h, GLCDC.GR2_FLM5 400E 1218h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	LNNUM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DATANUM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	DATANUM[15:0]	Number of Data Transfer Times Per Line for Accessing Graphics Data	Number of data transfers per macro line for accessing graphics data (frame buffer data), where one transfer is defined as 16-beat burst access (64-byte boundary). 0000h: Once : FFFFh: 65536 times.	R/W
b26 to b16	LNNUM[10:0]	Number of Lines Per Frame for Accessing Graphics Data	Number of macro lines per frame for accessing graphics data (frame buffer data). 000h: 1 macro line : 3FBh: 1020 macro lines. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

DATANUM[15:0] bits (Number of Data Transfer Times Per Line for Accessing Graphics Data)

The DATANUM[15:0] bits specify the number of data transfers per macro line for accessing graphics data, where one transfer is defined as a 16-beat burst access through the GPX bus. For all the data transfers, at least 2 bytes of valid pixel data are required (16 pixels in LUT1 format and 16 pixels or more in the other formats). If the number of bytes per macro line cannot be divided by 64 (4 bytes × 16-beat), DATANUM is obtained by rounding up to the whole number.

LNNUM[10:0] bits (Number of Lines Per Frame for Accessing Graphics Data)

The LNNUM[10:0] bits specify the number of lines per frame for accessing graphics data. When graphics data for the number of lines set to these bits is read, it signals the end of the frame and the base address is loaded.

The following are two use cases for macro lines. In these use cases, the frame size is 480 pixels × 272 lines and the pixel format is RGB565 (16 bpp).

Case 1) One macro line is configured to be equivalent to the frame raster width. The number of macro lines is equivalent to the number of vertical lines.

DATANUM: 000Eh (15 times = 16 × 480 / 512)
LNNUM: 10Fh (272 macro lines = 272 / 1)

Case 2) One macro line is configured to be repeated during the display frame. The number of macro lines is not equivalent to the number of vertical lines. In the example, one macro line has 16 times the raster width.

DATANUM: 00EFh (240 times = 16 × 480 / 512 × 16)
LNNUM: 010h (17 macro lines = 272 / 16)

58.2.15 Graphics 1 Frame Buffer Control Register 6 (GR1_FLM6) Graphics 2 Frame Buffer Control Register 6 (GR2_FLM6)

Address(es): GLCDC.GR1_FLM6 400E 111Ch, GLCDC.GR2_FLM6 400E 121Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	FORMAT[2:0]			—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b27 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b30 to b28	FORMAT[2:0]	Data Format for Accessing Graphics Data	Data format for accessing graphics data (frame buffer data). b30 b28 0 0 0: RGB565 (16 bits/pixel) 0 0 1: RGB888 (32 bits/pixel, 8 bits on the MSB side are invalid) 0 1 0: ARGB1555 (16 bits/pixel, 1 bit of A is LUT data) 0 1 1: ARGB4444 (16 bits/pixel) 1 0 0: ARGB8888 (32 bits/pixel) 1 0 1: CLUT8 (8 bits/pixel) 1 1 0: CLUT4 (4 bits/pixel) 1 1 1: CLUT1 (1 bit/pixel).	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

FORMAT[2:0] bits (Data Format for Accessing Graphics Data)

The FORMAT[2:0] bits specify the data format for accessing graphics data. CLUT1, CLUT4, CLUT8 contain the addresses 01h or 00h, 0Fh to 00h, and FFh to 00h, respectively, for accessing the color palette. ARGB1555 contains the address for accessing the color palette (80h or 00h) in the MSB and RGB data in the other bits. ARGB8888 and ARGB4444 contain the upper 8-bit or 4-bit alpha blending values and RGB data. RGB888 and RGB565 contain RGB data only.

58.2.16 Graphics 1 Alpha Blending Control Register 1 (GR1_AB1) Graphics 2 Alpha Blending Control Register 1 (GR2_AB1)

Address(es): GLCDC.GR1_AB1 400E 1120h, GLCDC.GR2_AB1 400E 1220h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ARCON	—	—	—	ARCDISPON	—	—	—	GRCDISPON	—	—	DISPSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	DISPSEL[1:0]	Graphics Display Plane Control	b1 b0 0 0: Background color display (value set in the GRn_BASE register) 0 1: Lower-layer graphics display 1 0: Current graphics display 1 1: Blended display of lower-layer graphics (input image from the previous stage) and current graphics (data read from the GPX bus).	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	GRCDISPON	Graphics Image Area Border Display Control	0: Turn display off 1: Turn display on.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ARCDISPON	Image Area Border Display Control for Rectangular Area Alpha Blending	0: Turn display off 1: Turn display on.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ARCON	Rectangular Area Alpha Blending Control	0: Turn blending off 1: Turn blending on.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

DISPSEL[1:0] bits (Graphics Display Plane Control)

The DISPSEL[1:0] bits control the graphics display plane. When the plane is selected as the lower-layer graphics, the image input from the previous stage is displayed (the background plane for GR1, and output from GR1 for GR2); as the background color, the background color specified in the GLCDC registers is displayed; and as the current graphics, the ARGB8888 data obtained by expanding the graphics data read by the GLCDC from the GPX bus is displayed. When the current graphics display is selected (these bits are set to 10b), RGB888 data is displayed, regardless of the alpha blending value in the pixel. Table 58.8 and Figure 58.10 show the relationship between the register setting and display area.

GRCDISPON bit (Graphics Image Area Border Display Control)

The GRCDISPON bit turns on or off the border display for the graphics image area. When the display is turned on (this bit is set to 1), the graphics image area is bordered with the preset color. The border is 1 pixel wide on the outermost periphery of the area with the display data set as FFh for each RGB color.

ARCDISPON bit (Image Area Border Display Control for Rectangular Area Alpha Blending)

The ARCDISPON bit turns on or off the border display for the image area where rectangular area alpha blending is performed. When the display is turned on (this bit is set to 1), the image area for the rectangular alpha blending is bordered with the preset color. The border is 1 pixel wide on the outermost periphery of the area and the display data set as FFh for each RGB color.

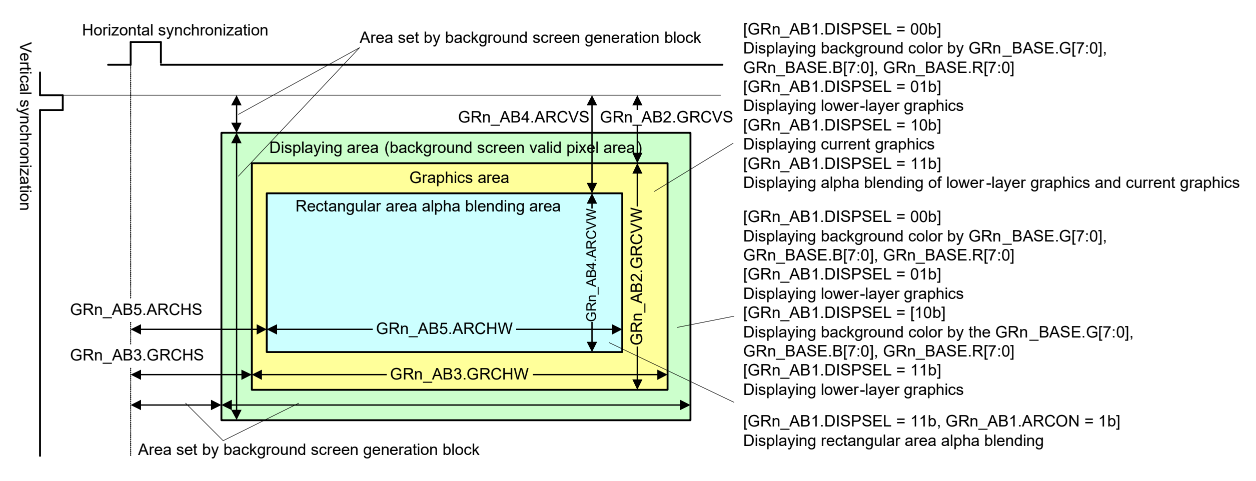
ARCON bit (Rectangular Area Alpha Blending Control)

The ARCON bit turns on or off alpha blending in a rectangular area. When alpha blending is turned on (this bit is set to 1), pixels are processed in accordance with the alpha blending control specified in the relevant registers for the specified rectangular area, not in accordance with the alpha value input from the graphics data interface for each pixel. In the areas outside the rectangular area in the graphics area, pixels are processed in accordance with the alpha value input from the graphics data interface for each pixel.

Table 58.8 Display selections

GRn_AB1. DISPSEL[1:0] (display plane)	GRn_AB1. ARCON (rectangular)	GRn_AB7. CKON (chroma key)	Within rectangular alpha blending area	Outside rectangular alpha blending area within graphics area	Outside graphics area within display area	Outside display area
00b	0	0	-	Background color	Background color	Lower layer R = G = B = 00h
01b	0	0	-	Lower layer	Lower layer	Lower layer R = G = B = 00h
10b	0	0	-	Current	Background color	Lower layer R = G = B = 00h
11b	0	0	-	Current + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 00h
11b	0	1	-	Current + RGB-index chroma key + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 00h
11b	1	0	Current + rectangular alpha blending	Current + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 00h
11b	1	1	Current + rectangular alpha blending	Current + RGB-index chroma key + alpha blending in pixel units	Lower layer	Lower layer R = G = B = 00h

Note: Operation is not guaranteed when any other value is set.



[GRn_AB1.DISPSEL = 00b]
Displaying background color by GRn_BASE.G[7:0], GRn_BASE.B[7:0], GRn_BASE.R[7:0]
[GRn_AB1.DISPSEL = 01b]
Displaying lower-layer graphics
[GRn_AB1.DISPSEL = 10b]
Displaying current graphics
[GRn_AB1.DISPSEL = 11b]
Displaying alpha blending of lower-layer graphics and current graphics

[GRn_AB1.DISPSEL = 00b]
Displaying background color by GRn_BASE.G[7:0], GRn_BASE.B[7:0], GRn_BASE.R[7:0]
[GRn_AB1.DISPSEL = 01b]
Displaying lower-layer graphics
[GRn_AB1.DISPSEL = [10b]
Displaying background color by the GRn_BASE.G[7:0], GRn_BASE.B[7:0], GRn_BASE.R[7:0]
[GRn_AB1.DISPSEL = 11b]
Displaying lower-layer graphics

[GRn_AB1.DISPSEL = 11b, GRn_AB1.ARCON = 1b]
Displaying rectangular area alpha blending

Figure 58.10 Selection of graphics display plane

58.2.17 Graphics 1 Alpha Blending Control Register 2 (GR1_AB2) Graphics 2 Alpha Blending Control Register 2 (GR2_AB2)

Address(es): GLCDC.GR1_AB2 400E 1124h, GLCDC.GR2_AB2 400E 1224h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GRCVS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GRCVW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GRCVW[10:0]	Vertical Width of Graphics Image Area	Width in lines. 010h: 16 lines : 3FCh: 1020 lines. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GRCVS[10:0]	Vertical Start Position of Graphics Image Area	Position in lines. 002h: 2nd line : 3EEh: 1006th line. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GRCVW[10:0] bits (Vertical Width of Graphics Image Area)

The GRCVW[10:0] bits specify the vertical width of the graphics image area.

GRCVS[10:0] bits (Vertical Start Position of Graphics Image Area)

The GRCVS[10:0] bits specify the vertical start position of the graphics image area, in reference to assertion of the vertical synchronization signal (VS). For the relationship with the graphics display plane, see Figure 58.10.

58.2.18 Graphics 1 Alpha Blending Control Register 3 (GR1_AB3) Graphics 2 Alpha Blending Control Register 3 (GR2_AB3)

Address(es): GLCDC.GR1_AB3 400E 1128h, GLCDC.GR2_AB3 400E 1228h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GRCHS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GRCHW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GRCHW[10:0]	Horizontal Width of Graphics Image Area	Width in pixels. 010h: 16 pixels : 3F8h: 1016 pixels. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GRCHS[10:0]	Horizontal Start Position of Graphics Image Area	Position in pixels. 005h: 5th pixel : 3EDh: 1005th pixel. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GRCHW[10:0] bits (Horizontal Width of Graphics Image Area)

The GRCHW[10:0] bits specify the horizontal width of the graphics image area.

GRCHS[10:0] bits (Horizontal Start Position of Graphics Image Area)

The GRCHS[10:0] bits specify the horizontal start position of the graphics image area, in reference to assertion of the horizontal synchronization signal (VS). For the relationship with the graphics display plane, see Figure 58.10.

58.2.19 Graphics 1 Alpha Blending Control Register 4 (GR1_AB4) Graphics 2 Alpha Blending Control Register 4 (GR2_AB4)

Address(es): GLCDC.GR1_AB4 400E 112Ch, GLCDC.GR2_AB4 400E 122Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	ARCVS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ARCVW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	ARCVW[10:0]	Vertical Width of Rectangular Area Alpha Blending Image Area	Width in lines. 001h: 1 line : 3FCh: 1020 lines. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	ARCVS[10:0]	Vertical Start Position of Rectangular Area Alpha Blending Image Area	Position in lines. 002h: 2nd line : 3EEh: 1006th line. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

ARCVW[10:0] bits (Vertical Width of Rectangular Area Alpha Blending Image Area)

The ARCVW[10:0] bits specify the vertical width of the rectangular area alpha blending image area.

ARCVS[10:0] bits (Vertical Start Position of Rectangular Area Alpha Blending Image Area)

The ARCVS[10:0] bits specify the vertical start position of the rectangular area alpha blending image area, in reference to assertion of the vertical synchronization signal (VS). For the relationship with the graphics display plane, see Figure 58.10.

58.2.20 Graphics 1 Alpha Blending Control Register 5 (GR1_AB5)
Graphics 2 Alpha Blending Control Register 5 (GR2_AB5)

Address(es): GLCDC.GR1_AB5 400E 1130h, GLCDC.GR2_AB5 400E 1230h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	ARCHS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	ARCHW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	ARCHW[10:0]	Horizontal Width of Rectangular Area Alpha Blending Image Area	Width in pixels. 001h: 1 pixel : 3F8h: 1016 pixels. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	ARCHS[10:0]	Horizontal Start Position of Rectangular Area Alpha Blending Image Area	Position in pixels. 005h: 5th pixel : 3EDh: 1005th pixel. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

ARCHW[10:0] bits (Horizontal Width of Rectangular Area Alpha Blending Image Area)

The ARCHW[10:0] bits specify the horizontal width of the rectangular area alpha blending image area.

ARCHS[10:0] bits (Horizontal Start Position of Rectangular Area Alpha Blending Image Area)

The ARCHS[10:0] bits specify the horizontal start position of the rectangular area alpha blending image area, in reference to assertion of the horizontal synchronization signal (HS). For the relationship with the graphics display plane, see Figure 58.10.

58.2.21 Graphics 1 Alpha Blending Control Register 6 (GR1_AB6) Graphics 2 Alpha Blending Control Register 6 (GR2_AB6)

Address(es): GLCDC.GR1_AB6 400E 1134h, GLCDC.GR2_AB6 400E 1234h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	ARCCOEF[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ARCRATE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	ARCRATE[7:0]	Frame Rate for Alpha Blending in Rectangular Area	00h: 1 frame : FFh: 256 frames.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24 to b16	ARCCOEF[8:0]	Alpha Coefficient for Alpha Blending in Rectangular Area	Valid settings: -255 to 255. Bit [8]: Sign 0: Add 1: Subtract. Bits [7:0]: Variation, as an absolute value.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

ARCRATE[7:0] bits (Frame Rate for Alpha Blending in Rectangular Area)

The ARCRATE[7:0] bits specify the frame rate for alpha blending in a rectangular area.

ARCCOEF[8:0] bits (Alpha Coefficient for Alpha Blending in Rectangular Area)

The ARCCOEF[8:0] bits specify the alpha coefficient for alpha blending in a rectangular area.

In alpha blending in a rectangular area, current graphics are faded in or out with the GRn_AB7.ARCDEF[7:0], GRn_AB6.ARCCOEF[8:0], and GRn_AB6.ARCRATE[7:0] settings. If the alpha value is set in the GR_ARC_DEF[7:0] bits, the GR_ARC_DEF[7:0] bits and the alpha blending in a rectangular area are turned on. Each time the vertical synchronization signal (VS) rises the number of times set in the GR_ARC_RATE[7:0] bits, the value in GR_ARC_COEF[8:0] is added to or subtracted from the alpha value. Figure 58.11 shows change in the alpha value.

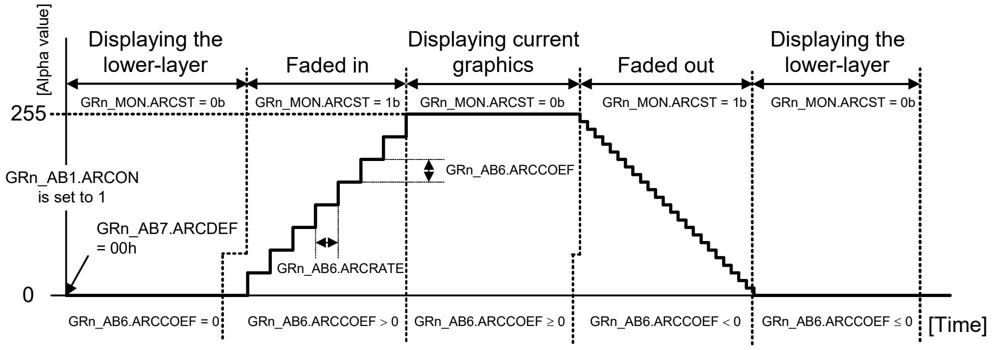


Figure 58.11 Changes in alpha value

58.2.22 Graphics 1 Alpha Blending Control Register 7 (GR1_AB7) Graphics 2 Alpha Blending Control Register 7 (GR2_AB7)

Address(es): GLCDC.GR1_AB7 400E 1138h, GLCDC.GR2_AB7 400E 1238h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	ARCDEF[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CKON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CKON	RGB-Index Chroma-Key Processing Control	0: Disable chroma-key processing 1: Enable chroma-key processing.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	ARCDEF[7:0]	Initial Alpha Value for Alpha Blending in Rectangular Area	Initial alpha value for alpha blending in rectangular area.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

CKON bit (RGB-Index Chroma-Key Processing Control)

The CKON bit turns RGB-index chroma-key processing on or off. This function is enabled only if this bit is 1 when the graphics display plane is the blended display (GRn_AB1.DISPSEL[1:0] = 11b). And it is reflected to the graphics area except alpha blending in a rectangular area. For details, see Table 58.8.

ARCDEF[7:0] bits (Initial Alpha Value for Alpha Blending in Rectangular Area)

The ARCDEF[7:0] bits specify the initial alpha value for alpha blending in a rectangular area. For changes in the alpha value during fade-in or fade-out of the current graphics using this bit, see Figure 58.11.

58.2.23 Graphics 1 Alpha Blending Control Register 8 (GR1_AB8) Graphics 2 Alpha Blending Control Register 8 (GR2_AB8)

Address(es): GLCDC.GR1_AB8 400E 113Ch, GLCDC.GR2_AB8 400E 123Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	CKKG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CKKB[7:0]								CKKR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CKKR[7:0]	R Signal for RGB-Index Chroma-Key Processing	R signal for RGB-index chroma-key processing. Unsigned 8-bit value.	R/W
b15 to b8	CKKB[7:0]	B Signal for RGB-Index Chroma-Key Processing	B signal for RGB-index chroma-key processing. Unsigned 8-bit value.	R/W
b23 to b16	CKKG[7:0]	G Signal for RGB-Index Chroma-Key Processing	G signal for RGB-index chroma-key processing. Unsigned 8-bit value.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

If RGB-index chroma-key processing is enabled when the RGB values of the current bit graphics agree with the values of the associated chroma-key bits, the image data of the current graphics (ARGB8888, including the alpha blending values) is replaced by the values in the GRn_AB9 register. In alpha blending in pixel units at later stages, the latest alpha values are used.

CKKR[7:0] bits (R Signal for RGB-Index Chroma-Key Processing)

The CKKR[7:0] bits specify the value to be compared with the R value of the current graphics in the RGB-index chroma-key processing.

CKKB[7:0] bits (B Signal for RGB-Index Chroma-Key Processing)

The CKKB[7:0] bits specify the value to be compared with the B value of the current graphics in the RGB-index chroma-key processing.

CKKG[7:0] bits (G Signal for RGB-Index Chroma-Key Processing)

The CKKG[7:0] bits specify the value to be compared with the G value of the current graphics in the RGB-index chroma-key processing.

58.2.24 Graphics 1 Alpha Blending Control Register 9 (GR1_AB9)
Graphics 2 Alpha Blending Control Register 9 (GR2_AB9)

Address(es): GLCDC.GR1_AB9 400E 1140h, GLCDC.GR2_AB9 400E 1240h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CKA[7:0]								CKG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CKB[7:0]								CKR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CKR[7:0]	R Value after RGB-Index Chroma-Key Processing Replacement	R value after RGB-index chroma-key processing replacement. Unsigned 8-bit value.	R/W
b15 to b8	CKB[7:0]	B Value after RGB-Index Chroma-Key Processing Replacement	B value after RGB-index chroma-key processing replacement. Unsigned 8-bit value.	R/W
b23 to b16	CKG[7:0]	G Value after RGB-Index Chroma-Key Processing Replacement	G value after RGB-index chroma-key processing replacement. Unsigned 8-bit value.	R/W
b31 to b24	CKA[7:0]	A Value after RGB-Index Chroma-Key Processing Replacement	A value after RGB-index chroma-key processing replacement.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

If the RGB-index chroma-key processing is enabled and the pixel data is to be replaced (the RGB values of the current graphics agree with the GRn_AB8 values), alpha blending in pixel units is performed in later stages in accordance with this image data.

CKR[7:0] bits (R Value after RGB-Index Chroma-Key Processing Replacement)

The CKR[7:0] bits specify the R value after RGB-index chroma-key processing replacement.

CKB[7:0] bits (B Value after RGB-Index Chroma-Key Processing Replacement)

The CKB[7:0] bits specify the B value after RGB-index chroma-key processing replacement.

CKG[7:0] bits (G Value after RGB-Index Chroma-Key Processing Replacement)

The CKG[7:0] bits specify the G value after RGB-index chroma-key processing replacement.

CKA[7:0] bits (A Value after RGB-Index Chroma-Key Processing Replacement)

The CKA[7:0] bits specify the A value after RGB-index chroma-key processing replacement.

58.2.25 Graphics 1 Background Color Control Register (GR1_BASE) Graphics 2 Background Color Control Register (GR2_BASE)

Address(es): GLCDC.GR1_BASE 400E 114Ch, GLCDC.GR2_BASE 400E 124Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	G[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	B[7:0]								R[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	R[7:0]	Background Color R Value	Background color R value. Unsigned 8-bit value.	R/W
b15 to b8	B[7:0]	Background Color B Value	Background color B value. Unsigned 8-bit value.	R/W
b23 to b16	G[7:0]	Background Color G Value	Background color G value. Unsigned 8-bit value.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

When the background color is selected in the display selection (GRn_AB1.DISPSEL[1:0] = 00b), this RGB data is output to the entire display image area. When the current graphics setting is selected (GRn_AB1.DISPSEL[1:0] = 10b), the RGB data is output to the outside of the graphics image area within the display image area.

R[7:0] bits (Background Color R Value)

The R[7:0] bits specify the background color R value.

B[7:0] bits (Background Color B Value)

The B[7:0] bits specify the background color B value.

G[7:0] bits (Background Color G Value)

The G[7:0] bits specify the background color G value.

58.2.26 Graphics 1 CLUT Table Interrupt Control Register (GR1_CLUTINT) Graphics 2 CLUT Table Interrupt Control Register (GR2_CLUTINT)

Address(es): GLCDC.GR1_CLUTINT 400E 1150h, GLCDC.GR2_CLUTINT 400E 1250h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LINE[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	LINE[10:0]	Number of Detection Lines	000h: 1 line : 3FFh: 1024 lines. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SEL	CLUT Table Control	0: Select CLUT table 0 1: Select CLUT table 1.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GRn_VEN.PVEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

LINE[10:0] bits (Number of Detection Lines)

The LINE[10:0] bits specify the number of lines to be detected. When the number of lines specified in this bit are detected, the event is recognized outside the module on the HS assertion. To retain the status of the recognized event and assert the GLCDC interrupt request signal, set the prescribed value to the State Detection Control Register (SYSCNT_DTCTEN) and Interrupt Request Enable Control Register (SYSCNT_INTEN), which are system control registers. Although this function is provided to both graphics 1 and 2, it is only enabled in graphics 2 in this GLCDC.

SEL bit (CLUT Table Control)

The SEL bit controls the CLUT plane to be used for internal operations. Access to the color palette (CLUT) through the register access bus is always valid for both planes 0 and 1, regardless of the setting in this bit, and the written value is immediately reflected to the internal operations (not in synchronization with the vertical synchronization signal).

To keep reflection of the CLUT memory contents to the internal operations in synchronization with the vertical synchronization signal, first write data through the register access bus to the plane that is not being used for the internal operations, and then modify the bits intended for controlling the plane that is to be used for the internal operations.

58.2.27 Graphics 1 Status Monitor Register (GR1_MON) Graphics 2 Status Monitor Register (GR2_MON)

Address(es): GLCDC.GR1_MON 400E 1154h, GLCDC.GR2_MON 400E 1254h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UNDFLST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARCST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ARCST	Status Monitor for Alpha Blending in Rectangular Area	0: Fade-in/fade-out not in progress 1: Fade-in/fade-out in progress.	R
b15 to b1	—	Reserved	These bits are read as 0.	R
b16	UNDFLST	Underflow Status Monitor	0: No underflow occurred in internal operations 1: Underflow occurred in internal operations.	R
b31 to b17	—	Reserved	These bits are read as 0.	R

ARCST bit (Status Monitor for Alpha Blending in Rectangular Area)

The ARCST bit indicates whether or not alpha blending (fade-in/fade-out) in a rectangular area is in progress. When alpha blending in a rectangular area is turned on (GRn_AB1.ARCON is set to 1) and the register value is to be reflected to the internal operations on assertion of the vertical synchronization signal (VS), this bit sets to 1 immediately on assertion of the vertical synchronization signal. When alpha blending in a rectangular area is turned off (GRn_AB1.ARCON cleared to 0) or when the alpha blending (fade-in/fade-out) in a rectangular area is complete (alpha blended value reaches the minimum or maximum value), this bit clears to 0. If the alpha coefficient for the alpha blending in a rectangular area (GRn_AB6.ARCCOEF[8:0]) is set to 000h and the initial alpha value for alpha blending in a rectangular area (GRn_AB7.ARCDEF[7:0]) is set to any value other than FFh or 00h, the alpha blending value does not reach the minimum or maximum value, and this bit remains 1 (no timeout processing is performed).

UNDFLST bit (Underflow Status Monitor)

The UNDFLST bit indicates whether or not an underflow has occurred in the internal operations. The underflow interrupt request flag sets when an underflow occurs, and retains its value until it is cleared by software. However, this bit monitors the internal status, and so the flag automatically clears to 0 when the graphics data bus interface initializes on assertion of the vertical synchronization signal. Even when the current graphics data is not required (GRn_AB1.DISPSEL[1:0] = 0xb), this bit sets to 1 during the period from the graphics image valid area start to the next frame vertical synchronization signal (VS) assertion timing set in the registers.

58.2.28 Gamma G Register Update Control Register (GAMG_LATCH)
Gamma B Register Update Control Register (GAMB_LATCH)
Gamma R Register Update Control Register (GAMR_LATCH)

Address(es): GLCDC.GAMG_LATCH 400E 1300h, GLCDC.GAMB_LATCH 400E 1340h, GLCDC.GAMR_LATCH 400E 1380h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VEN	Control of Gamma Correction x Module Register Value Reflection to Internal Operations	0: Disable reflection of register values to internal operations on assertion of vertical synchronization signal (VS) 1: Enable reflection of register values to internal operations on assertion of the vertical synchronization signal (VS).	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VEN bit (Control of Gamma Correction x Module Register Value Reflection to Internal Operations)

The VEN bit enables or disables reflection of the register values to the internal operations in the gamma correction circuit on assertion of the vertical synchronization signal (input). When this bit is set to 1, the register values are immediately reflected to the internal operations on assertion of the vertical synchronization signal (input), and then this bit automatically clears to 0. Also, if the signal is asserted that controls reflection of the register values to the internal operations of all the modules output from the background plane generation module, the register values are reflected to the internal operations on assertion of the vertical synchronization signal (input), regardless of the value of this bit. While this bit is 1, do not modify any register whose value is reflected to the internal operations on assertion of the vertical synchronization signal (input) in the GLCDC. Otherwise, operation is not guaranteed.

This bit must not be 1 at the same time as the BG_EN.VEN bit (control of background plane register value reflection to internal operations) in the Operation Control Register (BG_EN), one of the background plane setting registers. Otherwise, operation is not guaranteed.

Although there are three VEN bits, one for each G, R, and B color, only the GAMG_LATCH.VEN bit controls the gamma correction with the reflection of the GAM_SW.GAMON bit. To enable gamma correction, set the GAMG_LATCH.VEN bit once after setting the GAM_SW.GAMON bit.

The VEN bit is set to 1 by writing 1, and automatically clears to 0 immediately on assertion of the vertical synchronization signal.

58.2.29 Gamma Correction Block Function Switch Register (GAM_SW)

Address(es): GLCDC.GAM_SW 400E 1304h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	GAMON	Gamma Correction On/Off Control	0: Turn off gamma correction 1: Turn on gamma correction.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMG_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAMON bit (Gamma Correction On/Off Control)

The GAMON bit turns on or off gamma correction.

58.2.30 Gamma G Correction Block Table Setting Register 1 (GAMG_LUT1) Gamma B Correction Block Table Setting Register 1 (GAMB_LUT1) Gamma R Correction Block Table Setting Register 1 (GAMR_LUT1)

Address(es): GLCDC.GAMG_LUT1 400E 1308h, GLCDC.GAMB_LUT1 400E 1348h, GLCDC.GAMR_LUT1 400E 1388h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN00[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN01[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN01[10:0]	Gain Value of Area 1	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN00[10:0]	Gain Value of Area 0	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W

Bit	Symbol	Bit name	Description	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

Gamma correction for each color is carried out as follows:

- Din (input signal): 10 bits. To correct the output of graphics 2, 00b is appended as the lower 2 bits for extension
- Dout (output signal): 10 bits
- TH (threshold): 10 bits (register setting); up to 15 can be set
- GAIN (gain): 0/1024 to 2047/1024 for each area (register setting), can be set for up to 16 areas
- OFFSET (offset value): 21 bits (result of internal calculation; calculation of up to 15 points)

The following is automatically calculated from the assertion of the vertical synchronization signal (VS) to the start of valid pixel data internally:

- $\mathrm{Dout} = ((\mathrm{Din} - \mathrm{TH}(n)) \times \mathrm{GAIN}(n) + \mathrm{OFFSET}(n))$
- $\mathrm{OFFSET}(n) = \mathrm{OFFSET}(n - 1) + (\mathrm{TH}(n) - \mathrm{TH}(n - 1)) \times \mathrm{GAIN}(n - 1)$, where $\mathrm{OFFSET}(0) = 0$

Note: Because the gain is a positive number (≥ 0), the correction line shows a monotonic increase.

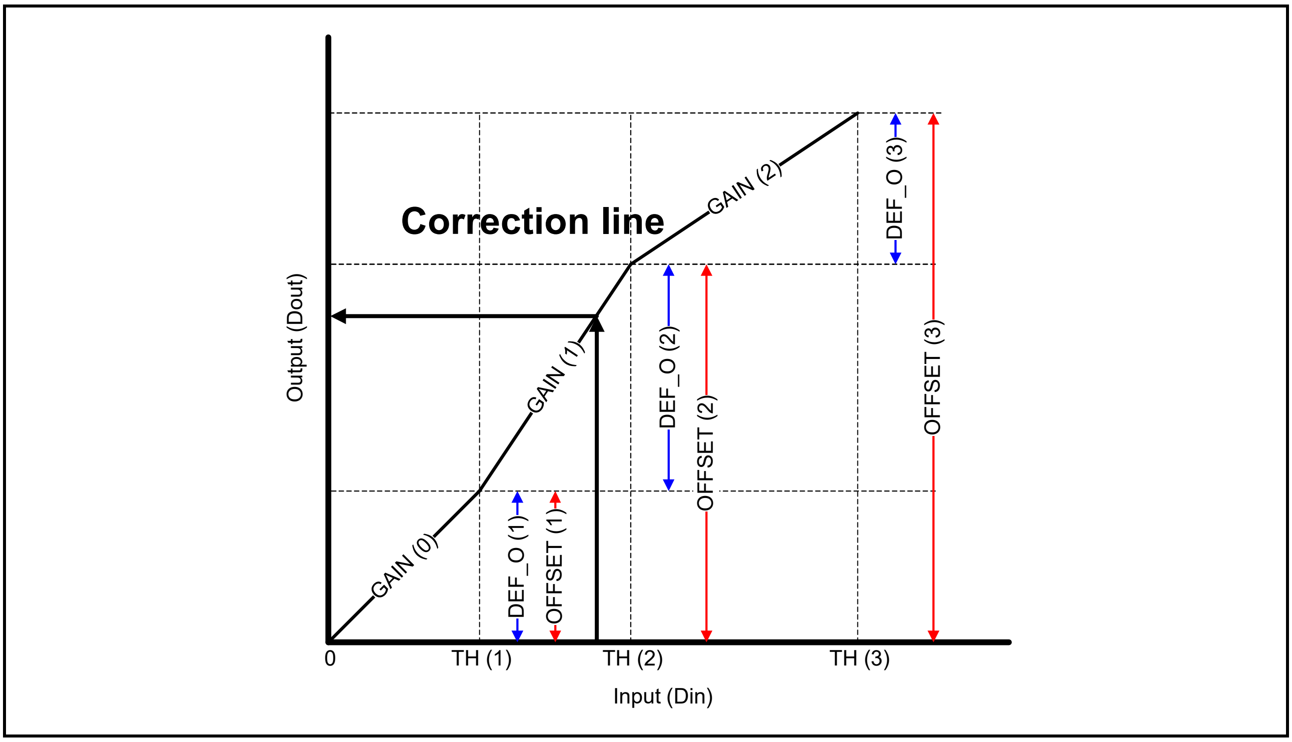


Figure 58.12 Calculation of gamma correction value

GAIN01[10:0] bits (Gain Value of Area 1)

The GAIN01[10:0] bits specify the gain value of area 1 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN00[10:0] bits (Gain Value of Area 0)

The GAIN00[10:0] bits specify the gain value of area 0 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.31 Gamma G Correction Block Table Setting Register 2 (GAMG_LUT2) Gamma B Correction Block Table Setting Register 2 (GAMB_LUT2) Gamma R Correction Block Table Setting Register 2 (GAMR_LUT2)

Address(es): GLCDC.GAMG_LUT2 400E 130Ch, GLCDC.GAMB_LUT2 400E 134Ch, GLCDC.GAMR_LUT2 400E 138Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN02[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN03[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN03[10:0]	Gain Value of Area 3	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN02[10:0]	Gain Value of Area 2	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN03[10:0] bits (Gain Value of Area 3)

The GAIN03[10:0] bits specify the gain value of area 3 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN02[10:0] bits (Gain Value of Area 2)

The GAIN02[10:0] bits specify the gain value of area 2 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.32 Gamma G Correction Block Table Setting Register 3 (GAMG_LUT3) Gamma B Correction Block Table Setting Register 3 (GAMB_LUT3) Gamma R Correction Block Table Setting Register 3 (GAMR_LUT3)

Address(es): GLCDC.GAMG_LUT3 400E 1310h, GLCDC.GAMB_LUT3 400E 1350h, GLCDC.GAMR_LUT3 400E 1390h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN04[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN05[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN05[10:0]	Gain Value of Area 5	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN04[10:0]	Gain Value of Area 4	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN05[10:0] bits (Gain Value of Area 5)

The GAIN05[10:0] bits specify the gain value of area 5 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN04[10:0] bits (Gain Value of Area 4)

The GAIN04[10:0] bits specify the gain value of area 4 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.33 Gamma G Correction Block Table Setting Register 4 (GAMG_LUT4) Gamma B Correction Block Table Setting Register 4 (GAMB_LUT4) Gamma R Correction Block Table Setting Register 4 (GAMR_LUT4)

Address(es): GLCDC.GAMG_LUT4 400E 1314h, GLCDC.GAMB_LUT4 400E 1354h, GLCDC.GAMR_LUT4 400E 1394h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN06[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN07[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN07[10:0]	Gain Value of Area 7	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN06[10:0]	Gain Value of Area 6	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN07[10:0] bits (Gain Value of Area 7)

The GAIN07[10:0] bits specify the gain value of area 7 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN06[10:0] bits (Gain Value of Area 6)

The GAIN06[10:0] bits specify the gain value of area 6 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.34 Gamma G Correction Block Table Setting Register 5 (GAMG_LUT5) Gamma B Correction Block Table Setting Register 5 (GAMB_LUT5) Gamma R Correction Block Table Setting Register 5 (GAMR_LUT5)

Address(es): GLCDC.GAMG_LUT5 400E 1318h, GLCDC.GAMB_LUT5 400E 1358h, GLCDC.GAMR_LUT5 400E 1398h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN08[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN09[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN09[10:0]	Gain Value of Area 9	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN08[10:0]	Gain Value of Area 8	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN09[10:0] bits (Gain Value of Area 9)

The GAIN09[10:0] bits specify the gain value of area 9 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN08[10:0] bits (Gain Value of Area 8)

The GAIN08[10:0] bits specify the gain value of area 8 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.35 Gamma G Correction Block Table Setting Register 6 (GAMG_LUT6) Gamma B Correction Block Table Setting Register 6 (GAMB_LUT6) Gamma R Correction Block Table Setting Register 6 (GAMR_LUT6)

Address(es): GLCDC.GAMG_LUT6 400E 131Ch, GLCDC.GAMB_LUT6 400E 135Ch, GLCDC.GAMR_LUT6 400E 139Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN10[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN11[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN11[10:0]	Gain Value of Area 11	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN10[10:0]	Gain Value of Area 10	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN11[10:0] bits (Gain Value of Area 11)

The GAIN11[10:0] bits specify the gain value of area 11 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN10[10:0] bits (Gain Value of Area 10)

The GAIN10[10:0] bits specify the gain value of area 10 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.36 Gamma G Correction Block Table Setting Register 7 (GAMG_LUT7) Gamma B Correction Block Table Setting Register 7 (GAMB_LUT7) Gamma R Correction Block Table Setting Register 7 (GAMR_LUT7)

Address(es): GLCDC.GAMG_LUT7 400E 1320h, GLCDC.GAMB_LUT7 400E 1360h, GLCDC.GAMR_LUT7 400E 13A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN12[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN13[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN13[10:0]	Gain Value of Area 13	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN12[10:0]	Gain Value of Area 12	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN13[10:0] bits (Gain Value of Area 13)

The GAIN13[10:0] bits specify the gain value of area 13 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN12[10:0] bits (Gain Value of Area 12)

The GAIN12[10:0] bits specify the gain value of area 12 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.37 Gamma G Correction Block Table Setting Register 8 (GAMG_LUT8) Gamma B Correction Block Table Setting Register 8 (GAMB_LUT8) Gamma R Correction Block Table Setting Register 8 (GAMR_LUT8)

Address(es): GLCDC.GAMG_LUT8 400E 1324h, GLCDC.GAMB_LUT8 400E 1364h, GLCDC.GAMR_LUT8 400E 13A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	GAIN14[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	GAIN15[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	GAIN15[10:0]	Gain Value of Area 15	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	GAIN14[10:0]	Gain Value of Area 14	Unsigned 11-bit fixed point value. 000h: 0.000 (0/1024) : 400h: 1.000 (1024/1024) : 7FFh: 1.999 (2047/1024).	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

GAIN15[10:0] bits (Gain Value of Area 15)

The GAIN15[10:0] bits specify the gain value of area 15 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

GAIN14[10:0] bits (Gain Value of Area 14)

The GAIN14[10:0] bits specify the gain value of area 14 to be used for gamma correction in terms of an unsigned 11-bit fixed point (0/1024 to 2047/1024). The location of the decimal point is between bits [10] and [11].

58.2.38 Gamma G Correction Block Area Setting Register 1 (GAMG_AREA1) Gamma B Correction Block Area Setting Register 1 (GAMB_AREA1) Gamma R Correction Block Area Setting Register 1 (GAMR_AREA1)

Address(es): GLCDC.GAMG_AREA1 400E 1328h, GLCDC.GAMB_AREA1 400E 1368h, GLCDC.GAMR_AREA1 400E 13A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TH01[9:0]										TH02[9:6]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TH02[5:0]						TH03[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	TH03[9:0]	Start Threshold of Area 3	Start threshold of area 3. Unsigned 10-bit integer.	R/W
b19 to b10	TH02[9:0]	Start Threshold of Area 2	Start threshold of area 2. Unsigned 10-bit integer.	R/W
b29 to b20	TH01[9:0]	Start Threshold of Area 1	Start threshold of area 1. Unsigned 10-bit integer.	R/W
b31 to b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

Set the start threshold TH(n) of area n (n = 0 to 15) to satisfy the following conditions. Otherwise, operation is not guaranteed.

TH (n) < TH (n + 1)
n = 0 to 15 and TH (0) = 000h, TH (16) = 3FFh
TH (n) = TH (n + 1) is valid only if TH (n) = 3FFh.

For details on calculation of the gamma correction value, see section 58.2.30, Gamma G Correction Block Table Setting Register 1 (GAMG_LUT1) Gamma B Correction Block Table Setting Register 1 (GAMB_LUT1) Gamma R Correction Block Table Setting Register 1 (GAMR_LUT1), and Figure 58.12.

TH03[9:0] bits (Start Threshold of Area 3)

The TH03[9:0] bits specify the start threshold of area 3 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH02[9:0] bits (Start Threshold of Area 2)

The TH02[9:0] bits specify the start threshold of area 2 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH01[9:0] bits (Start Threshold of Area 1)

The TH01[9:0] bits specify the start threshold of area 1 to be used for gamma correction in terms of an unsigned 10-bit integer.

58.2.39 Gamma G Correction Block Area Setting Register 2 (GAMG_AREA2)
Gamma B Correction Block Area Setting Register 2 (GAMB_AREA2)
Gamma R Correction Block Area Setting Register 2 (GAMR_AREA2)

Address(es): GLCDC.GAMG_AREA2 400E 132Ch, GLCDC.GAMB_AREA2 400E 136Ch, GLCDC.GAMR_AREA2 400E 13ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TH04[9:0]										TH05[9:6]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TH05[5:0]						TH06[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	TH06[9:0]	Start Threshold of Area 6	Start threshold of area 6. Unsigned 10-bit integer.	R/W
b19 to b10	TH05[9:6], TH08[5:0]	Start Threshold of Area 5	Start threshold of area 5. Unsigned 10-bit integer.	R/W
b29 to b20	TH04[9:0]	Start Threshold of Area 4	Start threshold of area 4. Unsigned 10-bit integer.	R/W
b31 to b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH06[9:0] bits (Start Threshold of Area 6)

The TH06[9:0] bits specify the start threshold of area 6 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH05[9:0] bits (Start Threshold of Area 5)

The TH05[9:0] bits specify the start threshold of area 5 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH04[9:0] bits (Start Threshold of Area 4)

The TH04[9:0] bits specify the start threshold of area 4 to be used for gamma correction in terms of an unsigned 10-bit integer.

58.2.40 Gamma G Correction Block Area Setting Register 3 (GAMG_AREA3) Gamma B Correction Block Area Setting Register 3 (GAMB_AREA3) Gamma R Correction Block Area Setting Register 3 (GAMR_AREA3)

Address(es): GLCDC.GAMG_AREA3 400E 1330h, GLCDC.GAMB_AREA3 400E 1370h, GLCDC.GAMR_AREA3 400E 13B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TH07[9:0]										TH08[9:6]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TH08[5:0]						TH09[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	TH09[9:0]	Start Threshold of Area 9	Start threshold of area 9. Unsigned 10-bit integer.	R/W
b19 to b10	TH05[9:6], TH08[5:0]	Start Threshold of Area 8	Start threshold of area 8. Unsigned 10-bit integer.	R/W
b29 to b20	TH07[9:0]	Start Threshold of Area 7	Start threshold of area 7. Unsigned 10-bit integer.	R/W
b31 to b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH09[9:0] bits (Start Threshold of Area 9)

The TH09[9:0] bits specify the start threshold of area 9 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH08[5:0] bits (Start Threshold of Area 8)

The TH08[9:0] bits specify the start threshold of area 8 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH07[9:0] bits (Start Threshold of Area 7)

The TH07[9:0] bits specify the start threshold of area 7 to be used for gamma correction in terms of an unsigned 10-bit integer.

58.2.41 Gamma G Correction Block Area Setting Register 4 (GAMG_AREA4) Gamma B Correction Block Area Setting Register 4 (GAMB_AREA4) Gamma R Correction Block Area Setting Register 4 (GAMR_AREA4)

Address(es): GLCDC.GAMG_AREA4 400E 1334h, GLCDC.GAMB_AREA4 400E 1374h, GLCDC.GAMR_AREA4 400E 13B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TH10[9:0]										TH11[9:6]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TH11[5:0]						TH12[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	TH12[9:0]	Start Threshold of Area 12	Start threshold of area 12. Unsigned 10-bit integer.	R/W
b19 to b10	TH08[9:6], TH11[5:0]	Start Threshold of Area 11	Start threshold of area 11. Unsigned 10-bit integer.	R/W
b29 to b20	TH10[9:0]	Start Threshold of Area 10	Start threshold of area 10. Unsigned 10-bit integer.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH12[9:0] bits (Start Threshold of Area 12)

The TH12[9:0] bits specify the start threshold of area 12 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH11[5:0] bits (Start Threshold of Area 11)

The TH11[9:0] bits specify the start threshold of area 11 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH10[9:0] bits (Start Threshold of Area 10)

The TH10[9:0] bits specify the start threshold of area 10 to be used for gamma correction in terms of an unsigned 10-bit integer.

58.2.42 Gamma G Correction Block Area Setting Register 5 (GAMG_AREA5) Gamma B Correction Block Area Setting Register 5 (GAMB_AREA5) Gamma R Correction Block Area Setting Register 5 (GAMR_AREA5)

Address(es): GLCDC.GAMG_AREA5 400E 1338h, GLCDC.GAMB_AREA5 400E 1378h, GLCDC.GAMR_AREA5 400E 13B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TH13[9:0]										TH14[9:6]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TH14[5:0]						TH15[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	TH15[9:0]	Start Threshold of Area 15	Start threshold of area 15. Unsigned 10-bit integer.	R/W
b19 to b10	TH11[9:6], TH14[5:0]	Start Threshold of Area 14	Start threshold of area 14. Unsigned 10-bit integer.	R/W
b29 to b20	TH13[9:0]	Start Threshold of Area 13	Start threshold of area 13. Unsigned 10-bit integer.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when GAMx_LATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TH15[9:0] bits (Start Threshold of Area 15)

The TH15[9:0] bits specify the start threshold of area 15 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH14[9:0] bits (Start Threshold of Area 14)

The TH14[9:0] bits specify the start threshold of area 14 to be used for gamma correction in terms of an unsigned 10-bit integer.

TH13[9:0] bits (Start Threshold of Area 13)

The TH13[9:0] bits specify the start threshold of area 13 to be used for gamma correction in terms of an unsigned 10-bit integer.

58.2.43 Output Control Block Register Update Control Register (OUT_VLATCH)

Address(es): GLCDC.OUT_VLATCH 400E 13C0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VEN	Control of Output Control Module Register Value Reflection to Internal Operations	0: Disable reflection of register values to internal operations on assertion of vertical synchronization signal (VS) 1: Enable reflection of register values to internal operations on assertion of vertical synchronization signal (VS).	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VEN bit (Control of Output Control Module Register Value Reflection to Internal Operations)

The VEN bit enables or disables reflection of the register values to the internal operations in the output control circuit on assertion of the vertical synchronization signal (input). When this bit is set to 1, the register values are immediately reflected to the internal operations on assertion of the vertical synchronization signal (input), and then this bit automatically clears to 0. Also, if the signal is asserted that controls reflection of the register values to the internal operations of all the modules output from the ground plane generation module, the register values are reflected to the internal operations on assertion of the vertical synchronization signal (input), regardless of the value of this bit. While this bit is 1, do not modify any register whose value is reflected to the internal operations on assertion of the vertical synchronization signal (input) in the GLCDC. Otherwise, operation is not guaranteed.

This bit must not be 1 at the same time as the BG_EN.VEN bit (control of background plane register value reflection to internal operations) in the Operation Control Register (BG_EN), one of the background plane setting registers. Otherwise, operation is not guaranteed.

58.2.44 Output Control Block Output Interface Register (OUT_SET)

Address(es): GLCDC.OUT_SET 400E 13C4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	ENDIANON	—	—	—	SWAPON	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FORMAT[1:0]		—	—	FRQSEL[1:0]		—	—	—	DIRSEL	—	—	PHASE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PHASE[1:0]	Data Output Delay Control in Serial RGB Format	Data delay in LCD_CLK cycles. b1 b0 0 0: 0 cycle 0 1: 1 cycle 1 0: 2 cycles 1 1: 3 cycles.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIRSEL	Scan Direction Select of Serial RGB Format	0: Forward scan 1: Reverse scan.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	FRQSEL[1:0]	Clock Frequency Division Control	b9 b8 0 0: No frequency division, parallel RGB 0 1: Setting prohibited 1 0: Quarter frequency (serial RGB) 1 1: Setting prohibited.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	FORMAT[1:0]	Output Format Select	b13 b12 0 0: RGB888; select RGB888 as dither output format 0 1: RGB666; select RGB666 as dither output format 1 0: RGB565; select RGB565 as dither output format 1 1: Serial RGB; select RGB888 as dither output format. Select dither output format in OUT_PDTHA.FORM[1:0].	R/W
b23 to b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	SWAPON	Pixel Order Control	0: RGB order 1: BGR order.	R/W
b27 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	ENDIANON	Bit Endian Control	0: Descending order (little endian) 1: Ascending order (big endian).	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the settings before operation is enabled. Operation is not guaranteed if the bit is rewritten during operation.

For details on assignment of pixel data to the output pins (LCD_DATA23 to LCD_DATA00) by setting these bits (except for OUT_SET.FRQSEL[1:0]), see Figure 58.3 to Figure 58.6.

PHASE[1:0] bits (Data Output Delay Control in Serial RGB Format)

The PHASE[1:0] bits control data output delay in serial RGB format. When the delay is 0 cycles (these bits are 00b), pixel data (R, B, or invalid data, depending on the setting in this register) is output one pixel clock (PXCLK) cycle after the horizontal data enable signal (HE). When any value other than 00b is set to OUT_SET.PHASE[1:0], pixel data output is delayed for a preset number of LCD_CLK cycles.

DIRSEL bit (**Scan Direction Select of Serial RGB Format**)

The DIRSEL bit controls the data arrangement of the serial RGB format. When this bit is set to 1, the serial RGB data is arranged in reverse direction, and when it is 0, the serial RGB is arranged in forward direction.

FRQSEL[1:0] bits (**Clock Frequency Division Control**)

The FRQSEL[1:0] bits control clock frequency division of LCD_CLK (panel output clock) and PXCLK (pixel clock for internal operations). Set these bits to 10b only for the serial RGB format (OUT_SET.FORMAT[1:0] = 11b), so that PXCLK has a quarter frequency of the LCD_CLK frequency and synchronizes with LCD_CLK. Set these bits to 00b for the parallel RGB format (OUT_SET.FORMAT[1:0] = 10b, 01b, or 00b), so that PXCLK has the same frequency as the LCD_CLK frequency and synchronizes with LCD_CLK. Otherwise, operation is not guaranteed.

FORMAT[1:0] bits (**Output Format Select**)

The FORMAT[1:0] bits select the output format of RGB data. Set these bits in accordance with the output format select bits in the Panel Dither Correction Register (OUT_PDTHA.FORM[1:0]). For serial RGB format (these bits are 11b), set OUT_PDTHA.FORM[1:0] to 00b. Otherwise, operation is not guaranteed.

SWAPON bit (**Pixel Order Control**)

The SWAPON bit controls the pixel order of RGB data output. When this bit is set to 1, internally processed data is assigned to the output pins in BGR order, and when this bit is 0, data is assigned in the RGB order. Data is assigned to the output pins (LCD_DATA23 to LCD_DATA00) with the MSB first for the RGB parallel format, and serially for the RGB serial format.

ENDIANON bit (**Bit Endian Control**)

The ENDIANON bit controls the bit order of RGB data output. When this bit is set to 1, internally processed data is assigned to the output pins in ascending order (big endian), and when this bit is 0, data is assigned in descending order (little endian).

58.2.45 Output Control Block Brightness Correction Register 1 (OUT_BRIGHT1)

Address(es): GLCDC.OUT_BRIGHT1 400E 13C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BRTG[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	BRTG[9:0]	Brightness Adjustment of G Signal	Brightness (DC) adjustment of G signal. Unsigned 10-bit integer; +512 with offset.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BRTG[9:0] bits (**Brightness Adjustment of G Signal**)

The BRTG[9:0] bits specify the brightness (DC) adjustment of the G signal.

Brightness correction of the G signal is performed as follows:

- Gout: Output of brightness correction (input of contrast correction); unsigned; 10 bits

- Gin: Input of brightness correction; unsigned; 10 bits
- BRTG: Setting in this bit
- Gout = Gin + BRTG - 512.

58.2.46 Output Control Block Brightness Correction Register 2 (OUT_BRIGHT2)

Address(es): GLCDC.OUT_BRIGHT2 400E 13CCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	BRTB[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BRTR[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	BRTR[9:0]	Brightness Adjustment of R Signal	Brightness (DC) adjustment of R signal. Unsigned 10-bit integer; +512 with offset.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	BRTB[9:0]	Brightness Adjustment of B Signal	Brightness (DC) adjustment of B signal. Unsigned 10-bit integer; +512 with offset.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

BRTR[9:0] bits (Brightness Adjustment of R Signal)

The BRTR[9:0] bits specify the brightness (DC) adjustment of the R signal.

Brightness correction of the R signal is performed as follows:

- Rout: Output of brightness correction (input of contrast correction); unsigned; 10 bits
- Rin: Input of brightness correction; unsigned; 10 bits
- BRTR: Setting in this bit
- Rout = Rin + BRTR - 512.

BRTB[9:0] bits (Brightness Adjustment of B Signal)

The BRTB[9:0] bits specify the brightness (DC) adjustment of the B signal.

Brightness correction of the B signal is performed as follows:

- Bout: Output of brightness correction (input of contrast correction); unsigned; 10 bits
- Bin: Input of brightness correction; unsigned; 10 bits
- BRTB: Setting in this bit
- Bout = Bin + BRTB - 512.

58.2.47 Output Control Block Contrast Correction Register (OUT_CONTRAST)

Address(es): GLCDC.OUT_CONTRAST 400E 13D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	CONTG[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CONTB[7:0]								CONTR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CONTR[7:0]	Contrast Adjustment of R Signal	Unsigned 8-bit fixed point value adjusting GAIN on R signal. 00h: 0/128 = 0.000 : 80h: 128/128 = 1.000 : FFh: 255/128 = 1.992.	R/W
b15 to b8	CONTB[7:0]	Contrast Adjustment of B Signal	Unsigned 8-bit fixed point value adjusting GAIN on B signal. 00h: 0/128 = 0.000 : 80h: 128/128 = 1.000 : FFh: 255/128 = 1.992.	R/W
b23 to b16	CONTG[7:0]	Contrast Adjustment of G Signal	Unsigned 8-bit fixed point value adjusting GAIN on G signal. 00h: 0/128 = 0.000 : 80h: 128/128 = 1.000 : FFh: 255/128 = 1.992.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

CONTR[7:0] bits (Contrast Adjustment of R Signal)

The CONTR[7:0] bits specify the contrast (GAIN) adjustment of R signal. The location of the decimal point is between bit [7] and [6].

CONTB[7:0] bits (Contrast Adjustment of B Signal)

The CONTB[7:0] bits specify the contrast (GAIN) adjustment of B signal. The location of the decimal point is between bit [7] and [6].

CONTG[7:0] bits (Contrast Adjustment of G Signal)

The CONTG[7:0] bits specify the contrast (GAIN) adjustment of G signal. The location of the decimal point is between bit [7] and [6].

Contrast correction of each pixel is performed as follows (x = R, G, B):

- Dxout: Output of contrast correction; unsigned; 10 bits
- Dxin: Input of contrast correction (output of brightness correction); unsigned; 10 bits
- CONTx: Setting in this bit

- Dxout = Dxin × CONTx.

58.2.48 Output Control Block Panel Dither Correction Register (OUT_PDTHA)

Address(es): GLCDC.OUT_PDTHA 400E 13D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	SEL[1:0]		—	—	FORM[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PA[1:0]		—	—	PB[1:0]		—	—	PC[1:0]		—	—	PD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PD[1:0]	Pattern Value (D) of 2×2 Pattern Dither	Pattern value (D) of 2×2 pattern dither. Unsigned 2-bit integer.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	PC[1:0]	Pattern Value (C) of 2×2 Pattern Dither	Pattern value (C) of 2×2 pattern dither. Unsigned 2-bit integer.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	PB[1:0]	Pattern Value (B) of 2×2 Pattern Dither	Pattern value (B) of 2×2 pattern dither. Unsigned 2-bit integer.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	PA[1:0]	Pattern Value (A) of 2×2 Pattern Dither	Pattern value (A) of 2×2 pattern dither. Unsigned 2-bit integer.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	FORM[1:0]	Output Format Select	b17 b16 0 0: RGB888; select RGB888 or serial RGB as output interface format 0 1: RGB666; select RGB666 as output interface format 1 0: RGB565; select RGB565 as output interface format 1 1: Setting prohibited. Select output interface format in OUT_SET.FORMAT[1:0].	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b21, b20	SEL[1:0]	Operation Mode	b21 b20 0 0: Truncate 0 1: Round-off 1 0: 2×2 pattern dither 1 1: Setting prohibited.	R/W
b31 to b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

PA[1:0], PB[1:0], PC[1:0], PD[1:0] bits (Pattern Value (A, B, C, D) of 2×2 Pattern Dither)

The PA[1:0], PB[1:0], PC[1:0], and PD[1:0] bits specify the pattern value A, B, C, and D of 2×2 pattern dither. Figure 58.13 shows the configuration of the dither correction block.

FORM[1:0] bits (Output Format Select)

The FORM[1:0] bits specify the output format of the dither process. These bits must be set in accordance with the OUT_SET.FORMAT[1:0] bits of the Output Interface Register. For serial RGB (OUT_SET.FORMAT[1:0] = 11b), set these bits to 00b. Otherwise, operation is not guaranteed.

SEL[1:0] bits (Operation Mode)

The SEL[1:0] bits specify the dither operation mode. The dither process is performed for the bits equal to or shorter than the pixel data length selected in the output format select bits (OUT_PDTHA.FORM[1:0]). OUT_PDTHA.PA[1:0], PB[1:0], PC[1:0], and PD[1:0] are used for 2×2 pattern dither.

Figure 58.13 shows the configuration of the dither correction block.

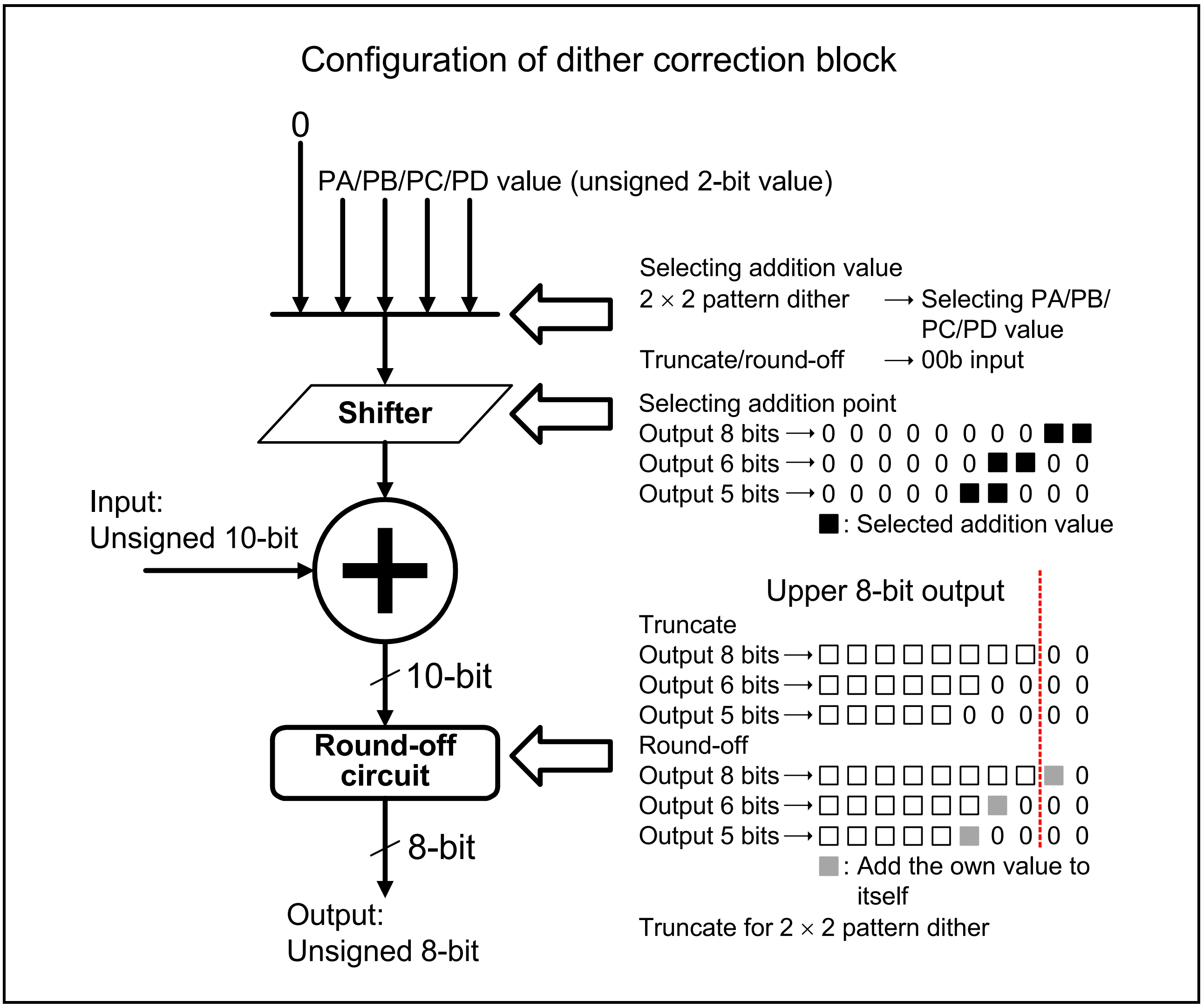


Figure 58.13 Configuration of dither correction block

Figure 58.14 shows the addition value selection method for 2×2 pattern dither.

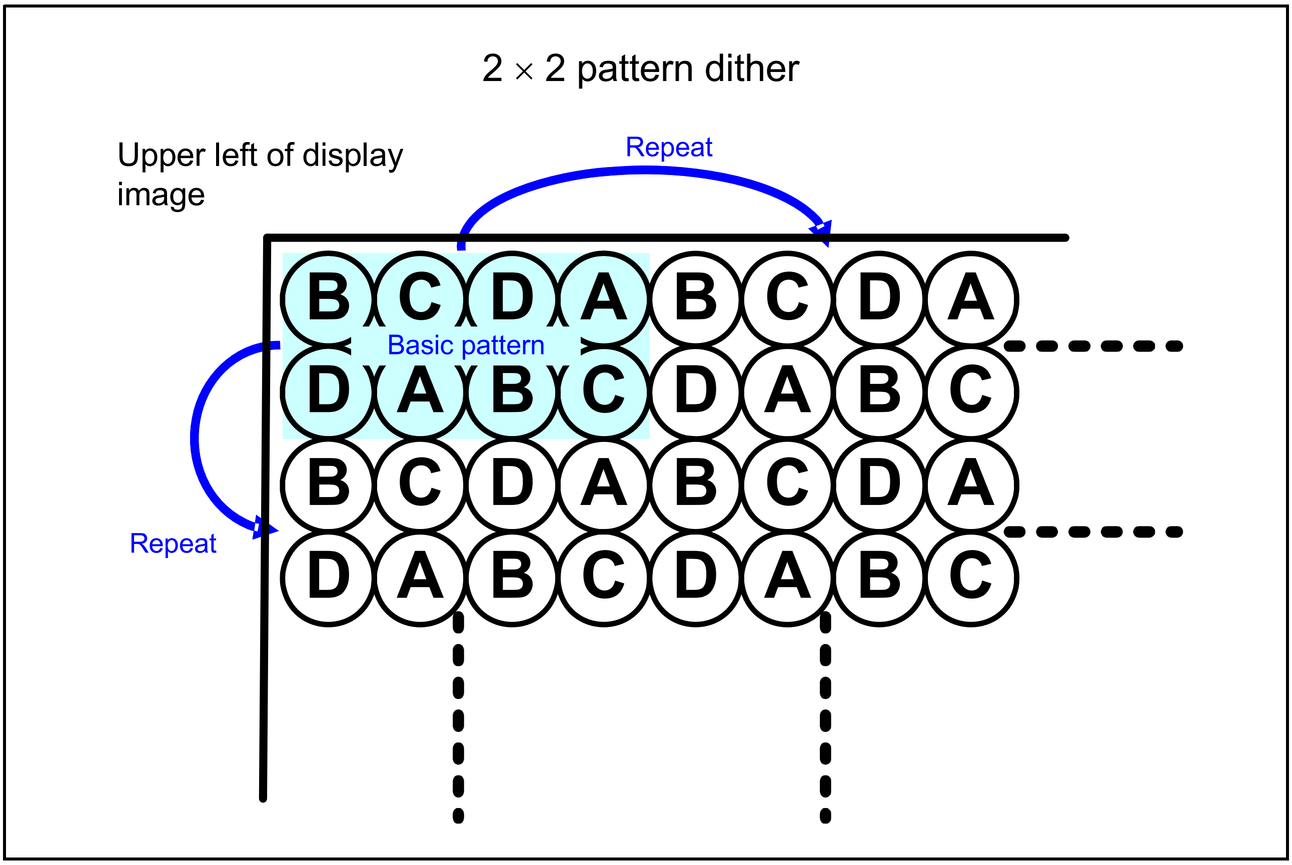


Figure 58.14 Addition value selection method for 2×2 pattern dither

A basic pattern is repeated as follows. Four frames constitute 1 cycle.

BCDA →	CDAB →	DABC →	ABCD	Shift to the left by one pixel
DABC →	ABCD →	BCDA →	CDAB	Shift to the left by one pixel
1st Frame	2nd Frame	3rd Frame	4th Frame	

- A: Value obtained after OUT_PDTHA.PA[1:0] bit value is shifted in accordance with the output format
- B: Value obtained after OUT_PDTHA.PB[1:0] bit value is shifted in accordance with the output format
- C: Value obtained after OUT_PDTHA.PC[1:0] bit value is shifted in accordance with the output format
- D: Value obtained after OUT_PDTHA.PD[1:0] bit value is shifted in accordance with the output format

Renesas recommends setting the bits as follows: PA[1:0] = 11b, PB[1:0] = 00b, PC[1:0] = 10b, PD[1:0] = 01b

When 2×2 pattern dither (OUT_PDTHA.SEL[1:0] = 10b) is to be set, the valid pixel area of the background plane must be an integer multiple of the basic pattern. If serial RGB is selected as the output format for the output control block, add two to the horizontal valid pixel width and set the resulting value to the background plane horizontal valid pixel width bits (BG_HSIZE.HW[10:0]).

58.2.49 Output Control Block Output Phase Control Register (OUT_CLKPHASE)

Address(es): GLCDC.OUT_CLKPHASE 400E 13E4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FRONTGAM	—	—	—	LCDEDGE	—	TCON0EDGE	TCON1EDGE	TCON2EDGE	TCON3EDGE	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	TCON3EDGE	LCD_TCON3 Output Phase Control	0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK.	R/W
b4	TCON2EDGE	LCD_TCON2 Output Phase Control	0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK.	R/W
b5	TCON1EDGE	LCD_TCON1 Output Phase Control	0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK.	R/W
b6	TCON0EDGE	LCD_TCON0 Output Phase Control	0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	LCDEDGE	LCD_DATA Output Phase Control	0: Synchronize output with rising edge of LCD_CLK 1: Synchronize output with falling edge of LCD_CLK.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	FRONTGAM	Correction Control	0: Process brightness/contrast correction followed by gamma correction 1: Process gamma correction followed by brightness/contrast correction.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This setting is reflected to the internal operations on assertion of the vertical synchronization signal (input) when OUT_VLATCH.VEN = 1 or when the register value reflection control signal to internal operations for all the modules is asserted.

TCON3EDGE bit (LCD_TCON3 Output Phase Control)

The TCON3EDGE bit controls the output phase of LCD_TCON3. When this bit is set to 1, LCD_TCON3 is output in synchronization with the falling edge of LCD_CLK, and when it is cleared to 0, LCD_TCON3 is output in synchronization with the rising edge.

TCON2EDGE bit (LCD_TCON2 Output Phase Control)

The TCON2EDGE bit controls the output phase of LCD_TCON2. When this bit is set to 1, LCD_TCON2 is output in synchronization with the falling edge of LCD_CLK, and when it is cleared to 0, LCD_TCON2 is output in synchronization with the rising edge.

TCON1EDGE bit (LCD_TCON1 Output Phase Control)

The TCON1EDGE bit controls the output phase of LCD_TCON1. When this bit is set to 1, LCD_TCON1 is output in synchronization with the falling edge of LCD_CLK, and when it is cleared to 0, LCD_TCON1 is output in synchronization with the rising edge.

TCON0EDGE bit (LCD_TCON0 Output Phase Control)

The TCON0EDGE bit controls the output phase of LCD_TCON0. When this bit is set to 1, LCD_ TCON0 is output in synchronization with the falling edge of LCD_CLK, and when it is cleared to 0, LCD_TCON0 is output in synchronization with the rising edge.

LCDEDGE bit (LCD_DATA Output Phase Control)

The LCDEDGE bit controls the output phase of the LCD_DATA pins (LCD_DATA23 to LCD_DATA00). When this bit is set to 1, the LCD_DATA pins are output in synchronization with the falling edge of LCD_CLK, and when it is cleared to 0, the LCD_DATA pins are output in synchronization with the rising edge.

FRONTGAM bit (Correction Control)

The FRONTGAM bit controls the correction sequence. When this bit is set to 1, gamma correction is followed by brightness and contrast correction, and when it is cleared to 0, gamma correction follows brightness and contrast correction. In both cases, each RGB data output from the graphics 2 module is extended from 8 bits to 10 bits (with 00b appended to the lower side), and is input to the preceding stage of the correction circuit. The output is rounded to 10 bits and is input to the dither correction circuit. Although the sequence of RGB gamma correction and brightness/contrast correction can be reversed using this bit, brightness correction always precedes contrast correction.

58.2.50 TCON Reference Timing Setting Register (TCON_TIM)

Address(es): GLCDC.TCON_TIM 400E 1404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	HALF[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OFFSET[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	OFFSET[10:0]	Horizontal Synchronization Signal Generation Reference Timing	Offset from the assertion of the internal horizontal synchronization signal in pixels. 000h: 1 pixel : 3FFh: 1024 pixels. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	HALF[10:0]	Vertical Synchronization Signal Generation Change Timing	Delay from the assertion of the internal horizontal synchronization signal in pixels. 000h: 1 pixel (no delay) : 3FFh: 1024 pixels. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

OFFSET[10:0] bits (Horizontal Synchronization Signal Generation Reference Timing)

The OFFSET[10:0] bits specify the reference timing to be used when the horizontal synchronization signal is generated in the TCON. Set the offset from the assertion of the internal horizontal synchronization signal (HS) in terms of pixels. Figure 58.15 shows the horizontal synchronization signal generation reference timing in the TCON.

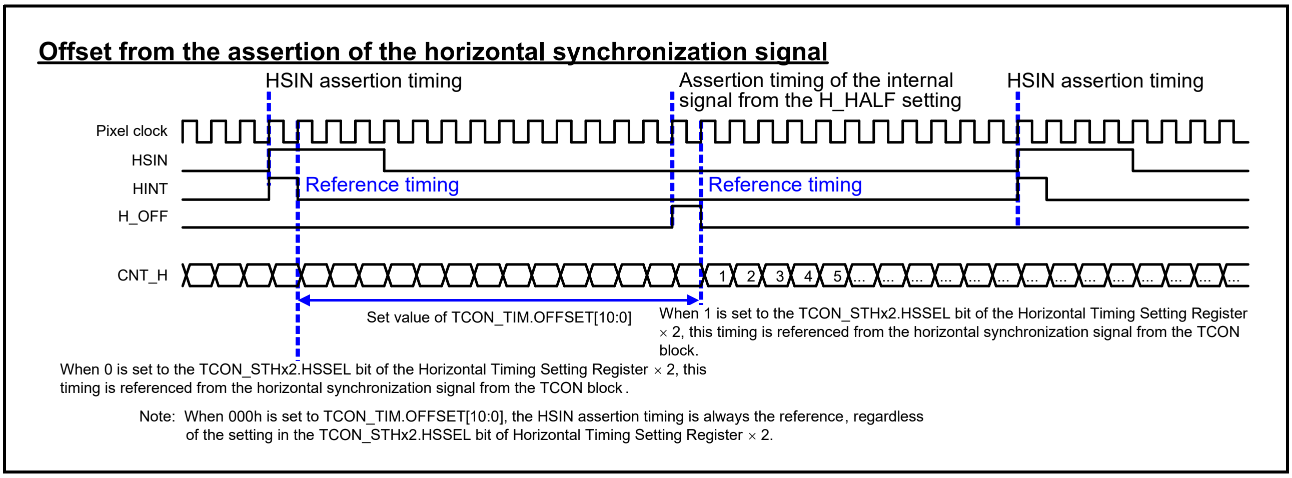


Figure 58.15 Reference timing in the TCON for horizontal synchronization signal generation

HALF[10:0] bits (Vertical Synchronization Signal Generation Change Timing)

The HALF[10:0] bits specify the vertical synchronization signal change timing when the signal is generated in the TCON. Set the change timing as a delay from the assertion of the internal horizontal synchronization signal (HS) in terms of pixels. Figure 58.16 shows the vertical synchronization signal change timing in the TCON.

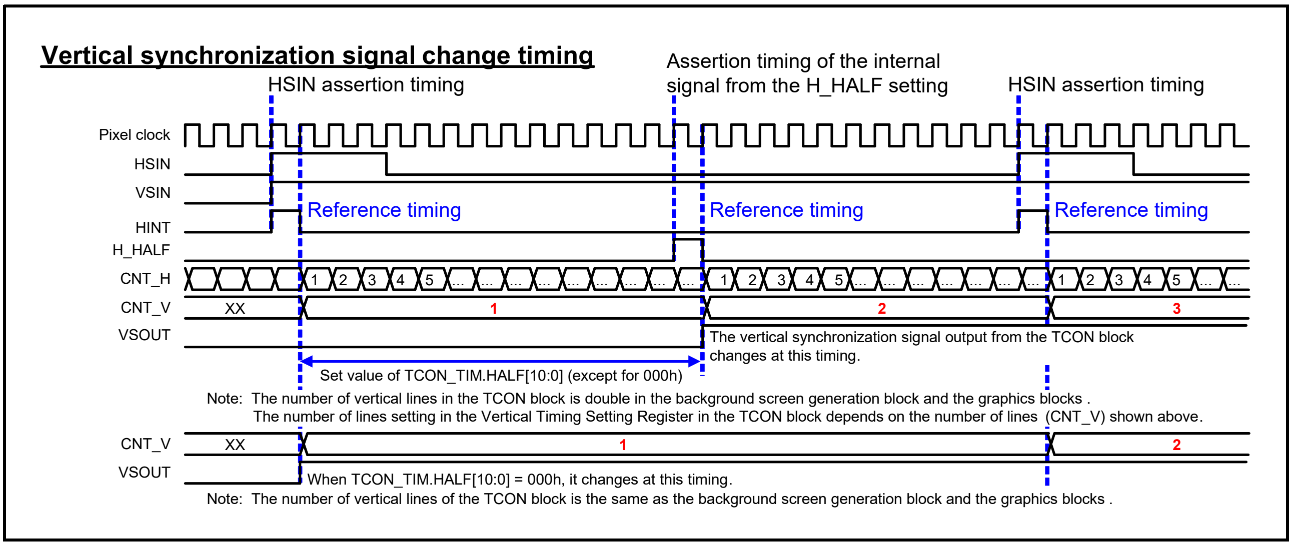


Figure 58.16 Timing in the TCON for vertical synchronization signal change

It is possible to change the vertical synchronization signal at any time within a horizontal line by setting this bit to the appropriate value.

Also, Figure 58.17 shows the relationship between the TCON block and the output format block in the output control block. These blocks are based on shared control signals (same timing) and image data, and the internal delay is the same. The delay difference specified in the register settings is the source of the timing differences on the external pins.

- TCON block: 3 cycles delay of the pixel clock (PXCLK)
- Data Format Block:
  - Parallel RGB: 3 cycles delay of the pixel clock (PXCLK)
  - Serial RGB: 3 cycles delay of the pixel clock (PXCLK)

    - The delay of the pixel head data including invalid data

    - No delay of the serial RGB data.

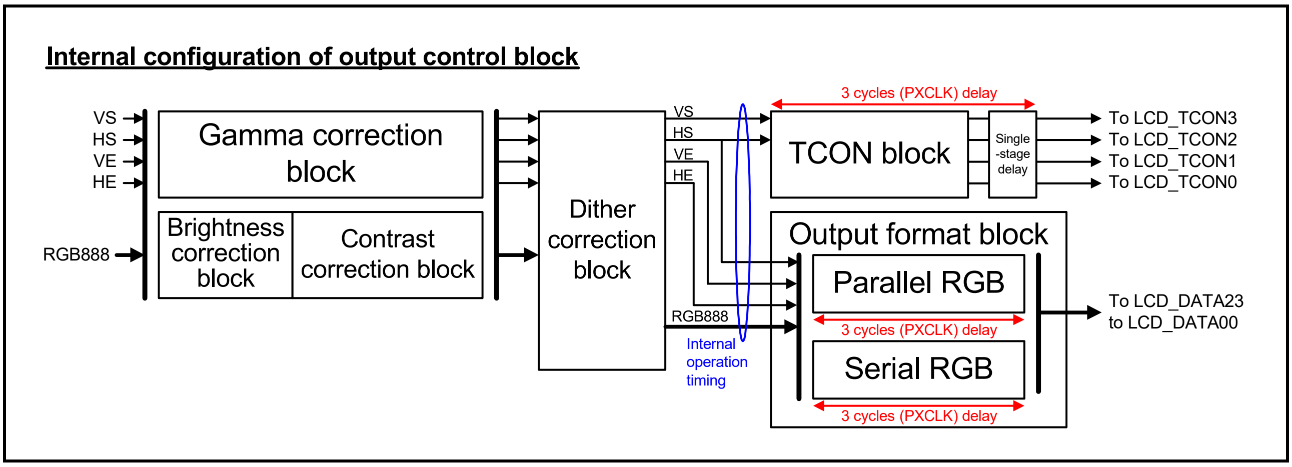


Figure 58.17 Internal configuration of output control block

58.2.51 TCON Vertical Timing Setting Register A1 (TCON_STVA1) TCON Vertical Timing Setting Register B1 (TCON_STVB1)

Address(es): GLCDC.TCON_STVA1 400E 1408h, GLCDC.TCON_STVB1 400E 1410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	VS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	VW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	VW[10:0]	Vertical Synchronization Signal STVx1 Second Change Timing	Signal assertion width in lines. 000h: 0 line (no vertical synchronization signal assertion) : 7FFh: 2047 lines.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	VS[10:0]	Vertical Synchronization Signal STVx1 First Change Timing	Signal delay in lines. 000h: 0 line (no delay) : 7FFh: 2047 lines.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The vertical timing setting registers (TCON_STVA1/TCON_STVB1 and TCON_STVA2/TCON_STVB2) have the same configuration, and x is either A or B in the descriptions.

VW[10:0] bits (Vertical Synchronization Signal STVx1 Second Change Timing)

The VW[10:0] bits specify the second change (negation) timing of the vertical synchronization signal STVx1, which is generated in the TCON. Set the second change timing as a delay from the first change point in terms of lines. The change position in a horizontal line is defined in TCON_TIM.HALF[10:0] in the Reference Timing Setting Register (TCON_TIM), as is the first change timing.

VS[10:0] bits (Vertical Synchronization Signal STVx1 First Change Timing)

The VS[10:0] bits specify the first change (assertion) timing of the vertical synchronization signal STVx1, which is generated in the TCON. Set the change timing as a delay from the input vertical synchronization signal (VSIN) in terms of lines. The change position in a horizontal line is defined in TCON_TIM.HALF[10:0] in the Reference Timing Setting Register (TCON_TIM), as is the first change timing.

Figure 58.18 shows the change timing of vertical synchronization signal to be generated.

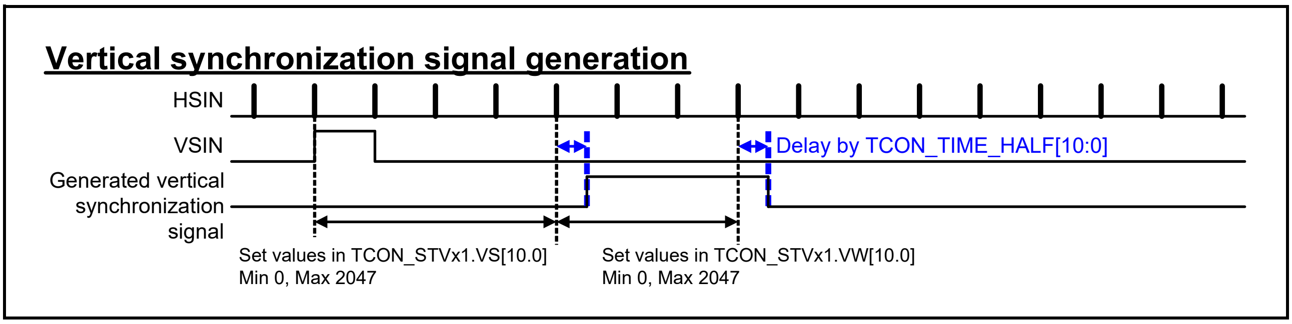


Figure 58.18 Generation of vertical synchronization signal

58.2.52 TCON Vertical Timing Setting Register A2 (TCON_STVA2) TCON Vertical Timing Setting Register B2 (TCON_STVB2)

Address(es): GLCDC.TCON_STVA2 400E 140Ch, GLCDC.TCON_STVB2 400E 1414h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	INV	—	SEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SEL[2:0]	Output Signal Select Control for VSOUT/VEOUT Pin	Output signal select for LCD_TCON0 pin (controlled in TCON_STVA2 register) and LCD_TCON1 pin (controlled in TCON_STVB2 register). b2 b0 0 0 0: STVA 0 0 1: STVB 0 1 0: STHA 0 1 1: STHB 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: DE.	R/W
b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	INV	Vertical Synchronization Signal STVx Polarity Inversion Control	0: Do not invert 1: Invert.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The vertical timing setting registers (TCON_STVA1/TCON_STVB1 and TCON_STVA2/TCON_STVB2) have the same configuration, and x is either A or B in the descriptions.

SEL[2:0] bits (Output Signal Select Control for VSOUT/VEOUT Pin)

The SEL[2:0] bits control output signal select for the LCD_TCON0/LCD_TCON1 pin. Figure 58.19 shows the configuration of the inversion control and output signal select.

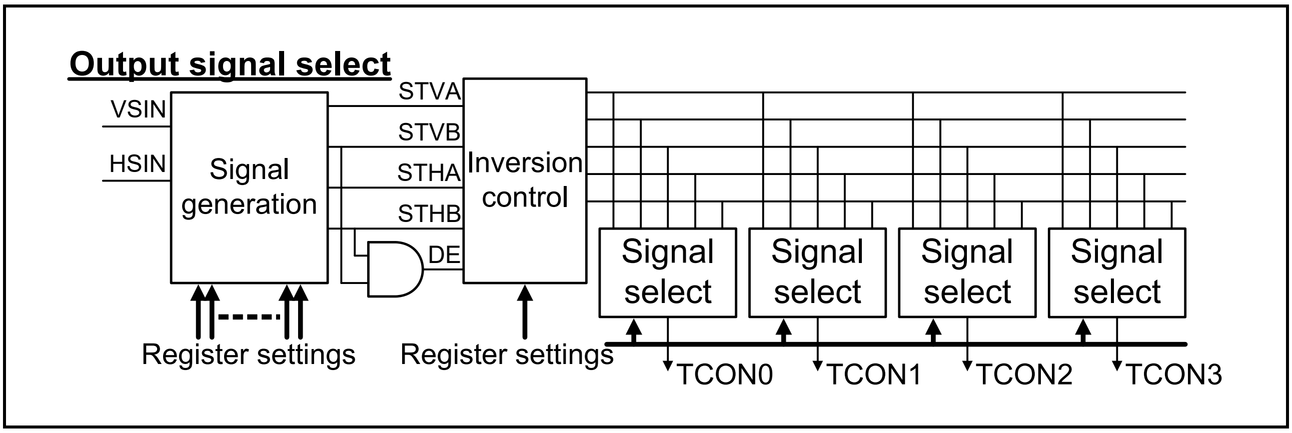


Figure 58.19 Configuration of inversion control and output signal selects

INV bit (Vertical Synchronization Signal STVx Polarity Inversion Control)

The INV bit controls polarity inversion of the vertical synchronization signal (STVx).

58.2.53 TCON Horizontal Timing Setting Register STHA1 (TCON_STHA1) TCON Horizontal Timing Setting Register STHB1 (TCON_STHB1)

Address(es): GLCDC.TCON_STHA1 400E 1418h, GLCDC.TCON_STHB1 400E 1420h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	HS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HW[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	HW[10:0]	Horizontal Synchronization Signal STHx1 Second Change Timing	Signal assertion width in pixels. 000h: 0 pixel (no horizontal synchronization signal assertion) : 3FFh: 1023 pixels. Other settings are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b16	HS[10:0]	Horizontal Synchronization Signal STHx1 First Change Timing	Signal delay in pixels. 000h: 0 pixel (no delay) : 3FFh: 1023 pixels. Other settings are prohibited.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The horizontal timing setting registers (TCON_STHA1/TCON_STHB1 and TCON_STHA2/TCON_STHB2) have the same configuration, and x is either A or B in the descriptions.

HW[10:0] bits (Horizontal Synchronization Signal STHx1 Second Change Timing)

The HW[10:0] bits specify the second change (negation) timing of the horizontal synchronization signal STHx1, which is generated in the TCON. Set the second change timing as a distance from the first change point in terms of pixels.

HS[10:0] bits (Horizontal Synchronization Signal STHx1 First Change Timing)

The HS[10:0] bits specify the first change (assertion) timing of the horizontal synchronization signal STHx1, which is generated in the TCON. Set the change timing as a distance from the input horizontal synchronization signal (HSIN) or the reference timing based on the offset specified in the TCON_TIM.OFFSET[10:0] bit (horizontal synchronization signal generation reference timing) in terms of pixels.

Figure 58.20 shows the horizontal synchronization signal generation timing if the input horizontal synchronization signal (HSIN) is based on the negated edge reference of the HINT signal. Figure 58.21 shows the horizontal synchronization signal generation timing after offset. By setting the TCON_TIM.OFFSET[10:0] bit (horizontal synchronization generation reference timing) and these horizontal synchronization bits appropriately, it is possible to generate a signal that is asserted before HSIN and negated after HSIN, where HSIN is the horizontal synchronization signal input to the TCON.

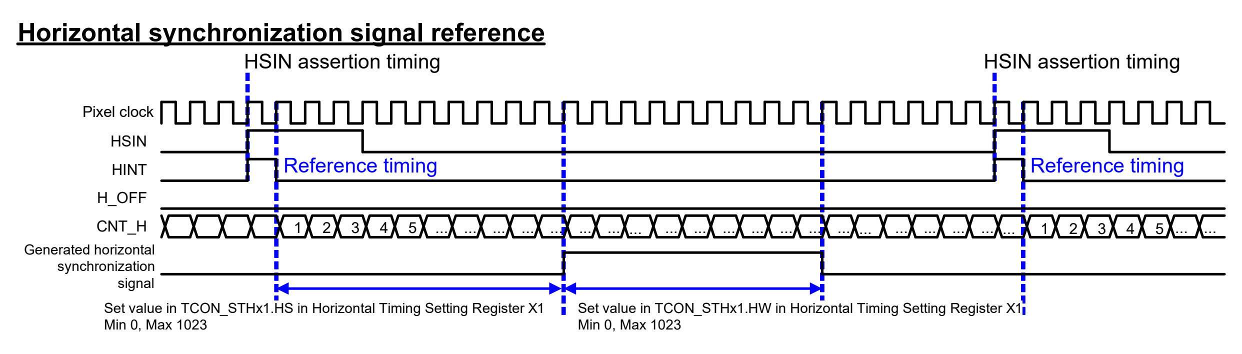


Figure 58.20 Signal generation based on input horizontal synchronization signal (HSIN)

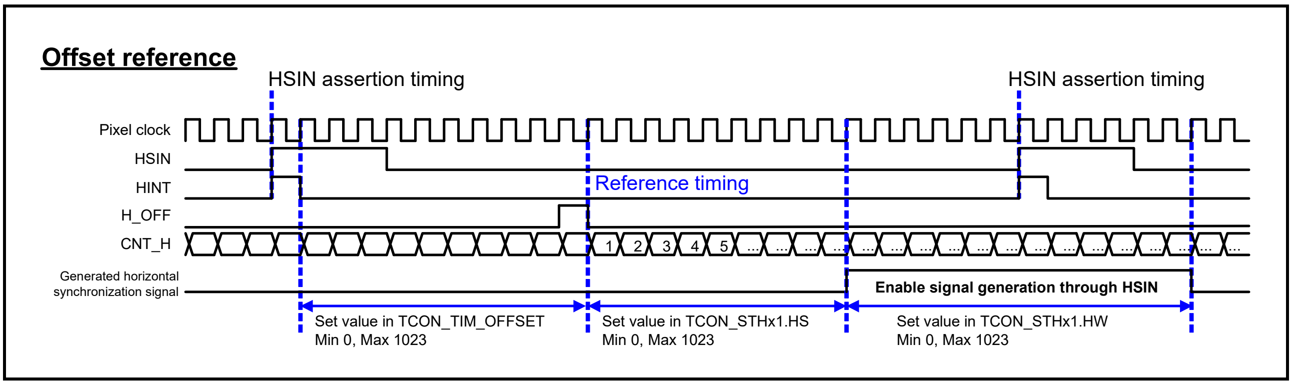


Figure 58.21 Horizontal synchronization signal generation based on offset

When generating the signal across HSIN on the offset reference, the horizontal synchronization signal of the last line of the frame spans the first line of the next frame. Even if the BG_EN.EN bit is cleared to 0 and the GLCDC operation is stopped, the horizontal synchronization signal across HSIN is not cleared at the frame end and retains the predetermined value until the second change timing set in the registers. If BG_EN.SWRST is cleared to 0, the signal returns to the initial state immediately.

58.2.54 TCON Horizontal Timing Setting Register STHA2 (TCON_STHA2)
TCON Horizontal Timing Setting Register STHB2 (TCON_STHB2)

Address(es): GLCDC.TCON_STHA2 400E 141Ch, GLCDC.TCON_STHB2 400E 1424h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HSSEL	—	—	—	INV	—	SEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SEL[2:0]	Output Signal Select Control for LCD_TCON2/LCD_TCON3 Pin	Output signal select for LCD_TCON2 pin (controlled in TCON_STHA2 register) and LCD_TCON3 pin (controlled in TCON_STHB2 register). b2 b0 0 0 0: STVA 0 0 1: STVB 0 1 0: STHA 0 1 1: STHB 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: DE.	R/W
b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	INV	Horizontal Synchronization Signal STHx Polarity Inversion Control	0: Do not invert 1: Invert.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	HSSEL	Horizontal Synchronization Signal STHx Reference Timing Control	0: Select input horizontal synchronization signal (HSIN) as reference for signal generation 1: Select offset specified in TCON_TIM.OFFSET[10:0] (horizontal synchronization generation reference timing) as reference for signal generation.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

The horizontal timing setting registers (TCON_STHA1/TCON_STHB1 and TCON_STHA2/TCON_STHB2) have the same configuration, and x is either A or B in the descriptions.

SEL[2:0] bits (Output Signal Select Control for LCD_TCON2/LCD_TCON3 Pin)

The SEL[2:0] bits control the output signal select for the LCD_TCON2/LCD_TCON3 pins.

INV bit (Horizontal Synchronization Signal STHx Polarity Inversion Control)

The INV bit controls polarity inversion of the horizontal synchronization signal (STHx).

HSSEL bit (Horizontal Synchronization Signal STHx Reference Timing Control)

The HSSEL bit selects the reference timing for generating the horizontal synchronization signal STHx. For details on the signal to be generated, see Figure 58.20 and Figure 58.21. For the configuration of the inversion control and output signal select, see Figure 58.19.

58.2.55 TCON Data Enable Polarity Setting Register (TCON_DE)

Address(es): GLCDC.TCON_DE 400E 1428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	INV	Data Enable Signal DE Polarity Inversion Control	0: Do not invert 1: Invert.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Rewriting these bits is prohibited during operation. Set the required settings before enabling operation. Operation is not guaranteed if the bit is rewritten during operation.

INV bit (Data Enable Signal DE Polarity Inversion Control)

The INV bit controls polarity inversion of the data enable signal DE. The data enable signal DE generated in the TCON is the logical AND of the STVB and STHB signals.

58.2.56 System Control Block State Detection Control Register (SYSCNT_DTCTEN)

Address(es): GLCDC.SYSCNT_DTCTEN 400E 1440h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDFDTC	L1UNDFDTC	VPOSDTC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VPOSDTC	Specified Line Detection Control	0: Disable detection of specified line 1: Enable detection of specified line*1.	R/W
b1	L1UNDFDTC	Graphics 1 Underflow Detection Control	0: Disable detection of graphics 1 underflow 1: Enable detection of graphics 1 underflow*2.	R/W
b2	L2UNDFDTC	Graphics 2 Underflow Detection Control	0: Disable detection of graphics 2 underflow 1: Enable detection of graphics 2 underflow*2.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set the VPOSDTC bit to 1 after setting the BG_EN.EN bit to 1.

Note 2. When setting the LnUNDFDTC (n = 1, 2) bit to 1 and when the BG_SYNC.VP[3:0] bits are set to a value greater than 5h, an unexpected GLCDC_LnUNDF (n = 1, 2) interrupt is generated after the GLCDC starts.Therefore, set the SYSCNT_STCLR.LnUNDFCLR (n = 1, 2) bit to 1, then set the SYSCNT_STMON.LnUNDF (n = 1, 2) bit to 0 to clear the unexpected GLCDC_LnUNDF (n = 1, 2) interrupt.

VPOSDTC bit (Specified Line Detection Control)

The VPOSDTC bit enables or disables detection of the specified line. When this bit is set to 1, the associated bit in the SYSCNT_STMON register sets to 1 on event notification from graphics 2. When it is cleared to 0, the associated bit in the SYSCNT_STMON register does not set to 1 even on event notification from graphics 2.

L1UNDFDTC bit (Graphics 1 Underflow Detection Control)

The L1UNDFDTC bit enables or disables detection of the graphics 1 underflow. When this bit is set to 1, the associated bit in the SYSCNT_STMON register sets to 1 when an underflow occurs in graphics 1. When it is cleared to 0, the associated bit in the SYSCNT_STMON register does not set to 1 even when an underflow occurs in graphics 1. The underflow state in graphics 1 is automatically cleared on assertion of the vertical synchronization signal (VS) regardless of the value of this bit, and normal operation is recovered.

L2UNDFDTC bit (Graphics 2 Underflow Detection Control)

The L2UNDFDTC bit enables or disables detection of the graphics 2 underflow. When this bit is set to 1, the associated bit in the SYSCNT_STMON register sets to 1 when an underflow occurs in graphics 2. When it is cleared to 0, the associated bit in the SYSCNT_STMON register does not set to 1 even when an underflow occurs in graphics 2. The underflow state in graphics 2 is automatically cleared on assertion of the vertical synchronization signal (VS) regardless of the value of this bit, and normal operation is recovered.

58.2.57 System Control Block Interrupt Request Enable Control Register (SYSCNT_INTEN)

Address(es): GLCDC.SYSCNT_INTEN 400E 1444h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDFINTEN	L1UNDFINTEN	VPOSINTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VPOSINTEN	Interrupt Request Signal GLCDC_VPOS Enable Control	0: Disable GLCDC_VPOS output 1: Enable GLCDC_VPOS output.	R/W
b1	L1UNDFINTEN	Interrupt Request Signal GLCDC_L1UNDF Enable Control	0: Disable GLCDC_L1UNDF output 1: Enable GLCDC_L1UNDF output.	R/W
b2	L2UNDFINTEN	Interrupt Request Signal GLCDC_L2UNDF Enable Control	0: Disable GLCDC_L2UNDF output 1: Enable GLCDC_L2UNDF output.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VPOSINTEN bit (Interrupt Request Signal GLCDC_VPOS Enable Control)

The VPOSINTEN bit enables or disables the interrupt request signal GLCDC_VPOS. When this bit is set to 1, the interrupt request signal GLCDC_VPOS is asserted when the associated status monitor flag SYSCNT_STMON[0] sets. When it is cleared to 0, the interrupt request signal GLCDC_VPOS is not asserted even when the associated status monitor flag SYSCNT_STMON[0] sets. If this bit is cleared during GLCDC_VPOS assertion, the associated status monitor flag SYSCNT_STMON[0] does not change, but the interrupt request signal GLCDC_VPOS is negated.

L1UNDFINTEN bit (Interrupt Request Signal GLCDC_L1UNDF Enable Control)

The L1UNDFINTEN bit enables or disables the interrupt request signal GLCDC_L1UNDF. When this bit is set to 1, the interrupt request signal GLCDC_L1UNDF is asserted when the associated status monitor flag SYSCNT_STMON[1] sets. When it is cleared to 0, the interrupt request signal GLCDC_L1UNDF is not asserted even when the associated status monitor flag SYSCNT_STMON[1] sets. If this bit is cleared during GLCDC_L1UNDF assertion, the associated status monitor flag SYSCNT_STMON[1] does not change, but the interrupt request signal GLCDC_L1UNDF is negated.

L2UNDFINTEN bit (Interrupt Request Signal GLCDC_L2UNDF Enable Control)

The L2UNDFINTEN bit enables or disables the interrupt request signal GLCDC_L2UNDF. When this bit is set to 1, the interrupt request signal GLCDC_L2UNDF is asserted when the associated status monitor flag SYSCNT_STMON[2] sets. When it is cleared to 0, the interrupt request signal GLCDC_L2UNDF is not asserted even when the associated status monitor flag SYSCNT_STMON[2] sets. If this bit is cleared during GLCDC_L2UNDF assertion, the associated status monitor flag SYSCNT_STMON[2] does not change, but the interrupt request signal GLCDC_L2UNDF is negated.

58.2.58 System Control Block Status Clear Register (SYSCNT_STCLR)

Address(es): GLCDC.SYSCNT_STCLR 400E 1448h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDFCLR	L1UNDFCLR	VPOSCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VPOSCLR	Graphics 2 Specified Line Detection Flag Clear	0: No operation 1: Clear the graphics 2 specified line detection flag.	R/W*1
b1	L1UNDFCLR	Graphics 1 Underflow Detection Flag Clear	0: No operation 1: Clear the graphics 1 underflow detection flag.	R/W*1
b2	L2UNDFCLR	Graphics 2 Underflow Detection Flag Clear	0: No operation 1: Clears the graphics 2 underflow detection flag.	R/W*1
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are read as 0.

VPOSCLR bit (Graphics 2 Specified Line Detection Flag Clear)

Writing 1 to the VPOSCLR bit clears the graphics 2 specified line detection flag. Clearance of the flag by this bit is only valid for the associated flag, and does not directly affect the other states and interrupt request signals. However, an interrupt request signal might be negated by clearing the detection flag.

L1UNDFCLR bit (Graphics 1 Underflow Detection Flag Clear)

Writing 1 to the L1UNDFCLR bit clears the graphics 1 underflow detection flag. Clearance of the flag by this bit is only valid for the associated flag, and does not directly affect the other states and interrupt request signals. However, an interrupt request signal might be negated by clearing the detection flag.

L2UNDFCLR bit (Graphics 2 Underflow Detection Flag Clear)

Writing 1 to the L2UNDFCLR bit clears the graphics 2 underflow detection flag. Clearance of the flag by this bit is only valid for the associated flag, and does not directly affect the other states and interrupt request signals. However, an interrupt request signal might be negated by clearing the detection flag.

58.2.59 System Control Block Status Monitor Register (SYSCNT_STMON)

Address(es): GLCDC.SYSCNT_STMON 400E 144Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	L2UNDF	L1UNDF	VPOS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VPOS	Graphics 2 Specified Line Detection Flag	0: Specified line notification not detected in graphics 2 1: Specified line notification detected in graphics 2.	R
b1	L1UNDF	Graphics 1 Underflow Detection Flag	0: No underflow detected in graphics 1 1: Underflow detected in graphics 1.	R
b2	L2UNDF	Graphics 2 Underflow Detection Flag	0: No underflow detected in graphics 2 1: Underflow detected in graphics 2.	R
b31 to b3	—	Reserved	These bits are read as 0.	R

VPOS flag (Graphics 2 Specified Line Detection Flag)

The VPOS flag indicates that the specified line notification was detected in graphics 2. When this flag is 1, it indicates that the specified line notification was detected in graphics 2 at some time in the past. It does not necessarily mean that graphics 2 is currently processing the specified line. When this flag is 0, it indicates that no specified line notification was detected after the module operation was enabled. When the interrupt request signal is enabled and is cleared while this flag is 1, the associated interrupt request signal GLCDC_VPOS is negated, but this does not affect the state of graphics 2.

L1UNDF flag (Graphics 1 Underflow Detection Flag)

The L1UNDF flag indicates that an underflow was detected in graphics 1. When this flag is 1, it indicates that an underflow was detected in graphics 1 at some time in the past. It does not necessarily mean that graphics 1 is currently in the underflow state. When this flag is 0, it indicates that no underflow was detected after the module operation was enabled. When the interrupt request signal is enabled and is cleared while this flag is 1, the associated interrupt request signal GLCDC_L1UNDF is negated, but this does not affect the state of graphics 1.

Even if the current graphics data is not required (GR1_AB1.DISPSEL[1:0] = 0xb), if the SYSCNT_DTCTEN.L1UNDFDTC flag (graphics 1 underflow detection control) is 1 and detection is enabled, this flag sets to 1 at the start of the graphics image valid area. To avoid unnecessary detection flag settings or interrupt request signal assertions, when the display does not require the current graphics data, clear the SYSCNT_DTCTEN.L1UNDFDTC flag and the SYSCNT_INTEN.L1UNDFINTEN bit (interrupt request signal GLCDC_L1UNDF enable control) to 0.

L2UNDF flag (Graphics 2 Underflow Detection Flag)

The L2UNDF flag indicates that an underflow was detected in graphics 2. When this flag is 1, it indicates that an underflow was detected in graphics 2 at some time in the past. It does not necessarily mean that graphics 2 is currently in the underflow state. When this flag is 0, it indicates that no underflow was detected after the module operation was enabled. When the interrupt request signal is enabled and is cleared while this flag is 1, the associated interrupt request signal GLCDC_L2UNDF is negated, but this does not affect the state of graphics 2.

Even if the current graphics data is not required (GR2_AB1.DISPSEL[1:0] = 0xb), if the SYSCNT_DTCTEN.L2UNDFDTC flag (graphics 2 underflow detection control) is 1 and detection is enabled, this flag sets to 1 at the start of the graphics image valid area. To avoid the unnecessary detection flag settings or interrupt request signal assertions, when the display does not require the current graphics data, clear the SYSCNT_DTCTEN.L2UNDFDTC flag and the SYSCNT_INTEN.L2UNDFINTEN bit (interrupt request signal

GLCDC_L2UNDF enable control) to 0.

58.2.60 System Control Block Version and Panel Clock Control Register (SYSCNT_PANEL_CLK)

Address(es): GLCDC.SYSCNT_PANEL_CLK 400E 1450h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	VER[15:0]															
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PIXSEL	—	—	—	CLKSEL	—	CLKEN	DCDR[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	DCDR[5:0]	Clock Division Ratio Setting Control	See Table 58.9 for details on these settings.	R/W
b6	CLKEN	Panel Clock Output Enable Control	0: Disable panel clock output 1: Enable panel clock output. Before changing the PIXSEL, CLKSEL, or DCDR bit, this bit must be set to 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CLKSEL	Panel Clock Supply Source Control	0: Select external clock (LCD_EXTCLK) 1: Select PLL output.	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	PIXSEL	Pixel Clock Select Control	0: Select no frequency division, parallel RGB 1: Select quarter frequency, serial RGB. This setting must have the same value as OUT_SET.FRQSEL[1].	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b16	VER[15:0]	Version Information	Version information of the GLCDC.	R

The configuration of the pixel and panel clock generator circuits are shown in Figure 9.1, Clock generation circuit block diagram.

DCDR[5:0] bits (Clock Division Ratio Setting Control)

The DCDR[5:0] bits control the setting of the panel clock division ratio. The division ratio bit can only be set to the values listed in Table 58.9. Operation is not guaranteed for values not listed.

Table 58.9 Input clock division (1 of 2)

DCDR[5:0]	Clock division ratio	CLKSEL = 0, LCD_EXTCLK(≤ 60 MHz)	CLKSEL = 1, PLL output (120 to 240 MHz)
000000b	1/2	LCD_EXTCLK/2	PLL output/2*1
000001b	1/1	LCD_EXTCLK*1	PLL output*1
000010b	1/2	LCD_EXTCLK/2	PLL output/2*1
000011b	1/3	LCD_EXTCLK/3	PLL output/3*1
000100b	1/4	LCD_EXTCLK/4	PLL output/4*1
000101b	1/5	LCD_EXTCLK/5	PLL output/5
000110b	1/6	LCD_EXTCLK/6	PLL output/6
000111b	1/7	LCD_EXTCLK/7	PLL output/7
001000b	1/8	LCD_EXTCLK/8	PLL output/8

Table 58.9 Input clock division (2 of 2)

DCDR[5:0]	Clock division ratio	CLKSEL = 0, LCD_EXTCLK(≤ 60 MHz)	CLKSEL = 1, PLL output (120 to 240 MHz)
001001b	1/9	LCD_EXTCLK/9	PLL output/9
001100b	1/12	LCD_EXTCLK/12	PLL output/12
010000b	1/16	LCD_EXTCLK/16	PLL output/16
011000b	1/24	LCD_EXTCLK/24	PLL output/24
100000b	1/32	LCD_EXTCLK/32	PLL output/32

Note 1. The panel clock is output as the LCD_CLK output clock. This setting may be prohibited because LCD_EXTCLK and LCD_CLK have limited frequencies. See section 60, Electrical Characteristics.

To set the panel clock:

1. After setting the input source of the panel clock in the CLKSEL bit, set the division ratio in the DCDR[5:0] bits and the pixel clock selection.
2. Set the CLKEN bit to 1.

CLKEN bit (Panel Clock Output Enable Control)

The CLKEN bit enables or disables the panel clock output. When enabling the panel clock output and operating the panel clock block, set this bit to 1. When changing the PIXSEL, CLKSEL, or DCDR bit, you must set this bit to 0 once and stop the panel clock output. Operation is not guaranteed if any setting is changed while the panel clock is being output.

CLKSEL bit (Panel Clock Supply Source Control)

The CLKSEL bit controls the selection of the panel clock supply source from either the external clock pin (LCD_EXTCLK) or PLL output. When the external clock is selected, set this bit to 0. When PLL output is selected, set this bit to 1.

PIXSEL bit (Pixel Clock Select Control)

The PIXSEL bit controls the selection of the pixel clock output. When selecting parallel RGB, set this bit to 0 to output the same frequency as the panel clock (no frequency division). When selecting serial RGB, set this bit to 1 to output the quarter frequency of the panel clock as the pixel clock.

This bit must be synchronized with the OUT_SET.FRQSEL[1:0] setting. You must set the same value as in FRQSEL[1]. Otherwise, operation is not guaranteed.

VER[15:0] bits (Version Information)

The VER[15:0] bits provide GLCDC version information.

58.3 Operation

58.3.1 Overall Control

The GLCDC consists of six modules as shown in Figure 58.22, each of which functions independently. The four modules handling image data are interconnected by the vertical and horizontal synchronization signals VS, HS, VE, and HE, and image data (RGB888), as shown in Figure 58.23. The processing of image data is carried out with the pixel clock (PXCLK). LCD_CLK synchronizes with PXCLK (also in phase with each other), and has the same or quadruple frequency (for serial RGB888 output). The registers controlling operation and setting parameters are connected with the internal peripheral bus 8 and operate on PCLKA. The circuits, including the data buffer and the CLUT memory, operate on PCLKA to read graphics data from the GPX bus, access the color palette (CLUT) memory, and expand graphics data into the ARGB8888 format.

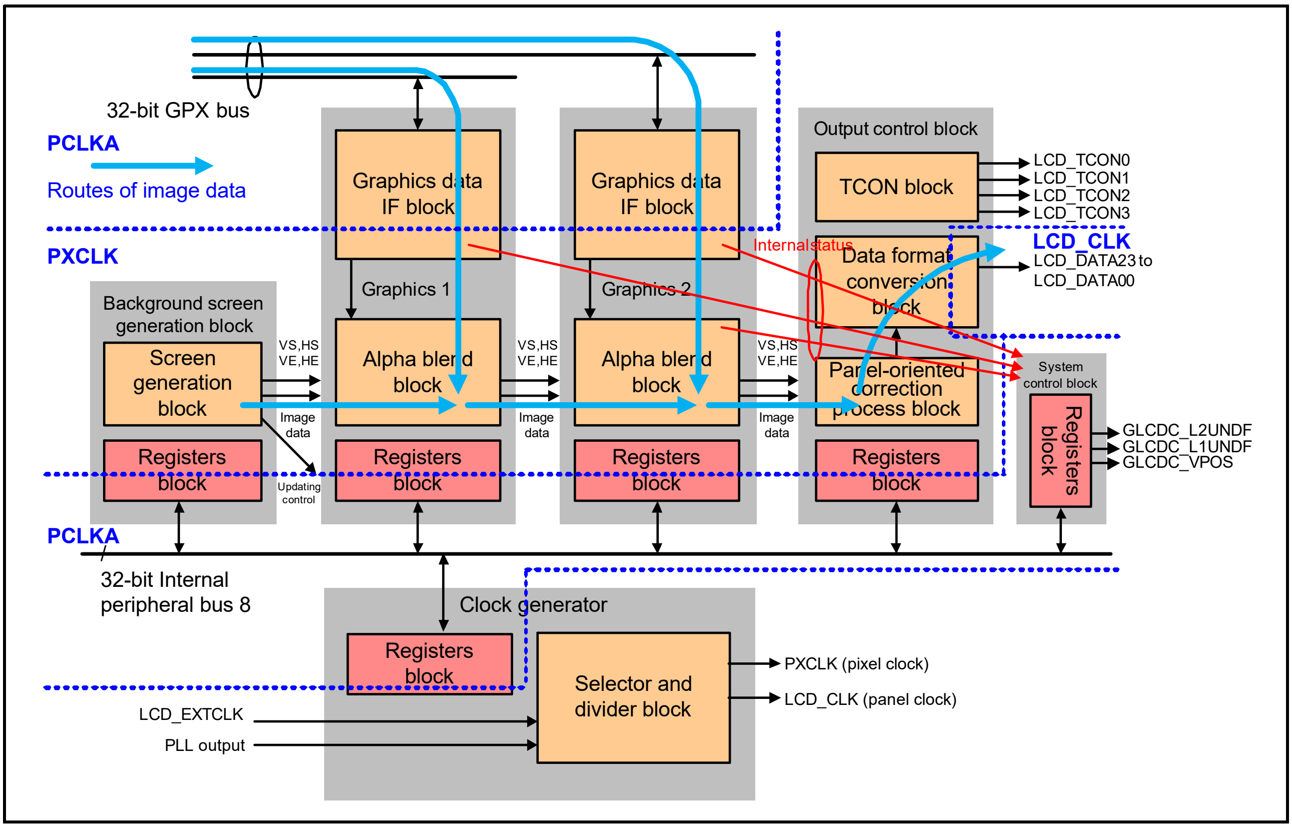


Figure 58.22 GLCDC configuration

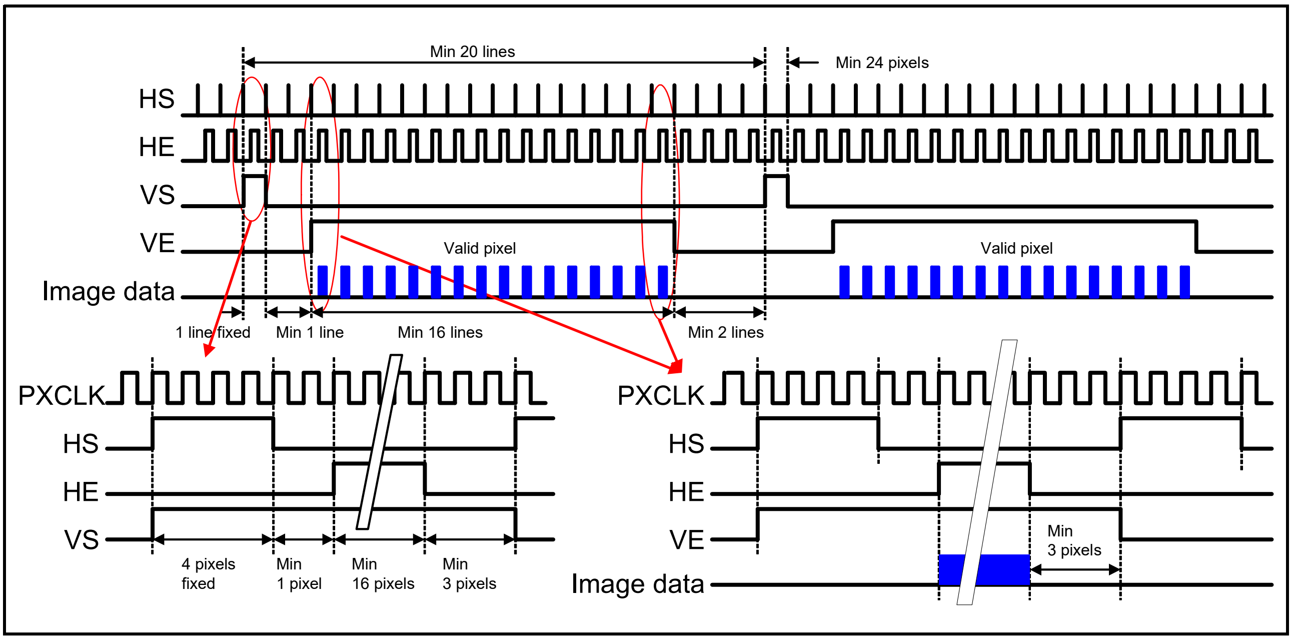


Figure 58.23 Image data carrier signals in the GLCDC

The graphics 1, graphics 2, and output control modules have no individual operation enable bits. By clearing the software reset state in the background screen generation module, these modules wait for vertical and horizontal synchronization signals VS, HS,VE, and HE, and image data (RGB888) to be input, detect assertion of the vertical synchronization signal VS, and start operation for each frame based on the values preset in the registers. For the system control module, clearing the software reset state in the background screen generation module allows it to monitor the status of graphics 1 and 2 in accordance with the register settings and assert interrupt request signals. These status flags are intended for observing the internal status and exert no influence on the internal operation. The operation between startup and stop of the GLCDC is outlined in this section.

(1) Startup

To start GLCDC operation:

1. Confirm that PCLKA is supplied to the GLCDC and that the reset is negated. When setting LCD_CLK to the same frequency as PXCLK or LCD_CLK to the quadruple speed clock, the setting must be consistent with the value in the register specified in the subsequent stage. The quadruple speed clock is required only when the output format is serial RGB888. Operation is not guaranteed if the quadruple speed clock is supplied to LCD_CLK in other output formats than serial RGB888.
2. Set 1 to the BG_EN.SWRST bit to release the entire GLCDC from a software reset.
3. Set parameters necessary for operation in each register. Although registers of any module can be set first, while the BG_EN.VEN bit (control of GLCDC internal register value reflection to internal operations), BG_EN.EN bit (background plane operation enable) of the background screen generation module, and the VEN bit in the register update control register of each module remain cleared to 0, confirm that PCLKA and PXCLK/LCD_CLK are supplied and the BG_MON.SWRST bit (entire module software reset state monitor) is set to 1.
4. When displaying the graphics data read by graphics 1 and 2 through the GPX bus, set 1 to the GRn_FLMRD.RENB bit (graphics data read enable).
5. Write the color palette data of graphics 1 and 2 (0 and 1 planes) to the CLUT memory through the register access bus (internal peripheral bus 8) as required. This is necessary for the LUT1, LUT4, LUT8, and ARGB1555 data formats. ARGB1555 uses addresses 80h and 00h on the two planes, LUT1 uses 01h and 00h, LUT4 uses the addresses from 0Fh to 00h, and LUT8 uses the addresses from FFh to 00h.
6. Set 1 to the BG_EN.VEN bit (control of background plane register value reflection to internal operations) and the BG_EN.EN bit (background plane operation enable) of the background screen generation module at the same time. This setting allows output of the vertical and horizontal synchronization signals VS, HS, VE, and HE, and the image data (RGB888) from the background screen generation module. When the pixel data is valid (both VE and HE are 1), output from the background screen generation module uses the value in the BG_BGC register. The data value is 00 0000h for pixels that are not valid.
7. Each module detects the assertion of the vertical synchronization signal (VS) output from its previous stage (background screen generation module for graphics 1, graphics 1 for graphics 2, and graphics 2 for the output control module) and starts operation in accordance with its register settings. All the modules control the operation in frame units. When the assertion of the vertical synchronization signal (VS) is detected, the current frame is taken as the frame head and the status is initialized. If necessary (when the VEN bits in the register update control registers are 1), the register values read through the register access bus are reflected to the internal operation.

(2) Changing parameters during operation

With the GLCDC, it is possible to update parameters of each module and reflect the updates to the internal operations during operation, without preventing graphics data to be read by the background screen generation module, graphics 1, or graphics 2. By setting 1 to the VEN bits of the modules, including the background screen generation module, almost all parameters are reflected to the internal operations at the start of the following frame (at the start of the control screen for the background screen generation module, and immediately after the VS assertion of the previous stage for the other modules). However, if the CLUT plane being used for internal operations (determined in the GRn_CLUTINT.SEL[1:0] bits) is modified, the updates are reflected to the internal operations immediately without waiting for the following VS assertion. To circumvent this immediate reflection of the CLUT plane modification to the internal operations, first write all the image data (ARGB8888) necessary for the CLUT plane that is not being used for internal operations, next modify the GRn_CLUTINT.SEL[1:0] bits, and finally set 1 to the VEN bit of the background screen generation module or the target module.

To modify parameters during operation:

1. Confirm that the VEN bit of each module is 0. Operation is not guaranteed if the target registers are modified when the VEN bit of the module to which the register values are to be reflected to the internal operations, or the VEN bit of the background screen generation module is 1.
2. Modify the value of the target registers.
3. If only a particular module is to be the target, set 1 to the VEN bit of the target module. If multiple modules or the

background screen generation module are to be the targets, set 1 to the VEN bit of the background screen generation module. In this case, all the modules are included as targets, not only the background screen generation module.

4. Confirm that the VEN bit to which 1 was set is 0. If the bit is cleared to 0, the target register contents are reflected to the internal operations. If the bit remains 1, however, the target register contents might not yet be reflected to the internal operations. The VEN bit of each module is cleared to 0 immediately after the target register values are internally reflected. However, the VEN bit of the background screen generation module is not cleared to 0 until the module output VE is negated (with all the modules, sufficient delay is secured in reference to the VS assertion of the background screen generation module so that the register values are reflected to the internal operations for the same frame). Figure 58.24 shows the operation of the signals output by the background screen generation module and the monitor bits.

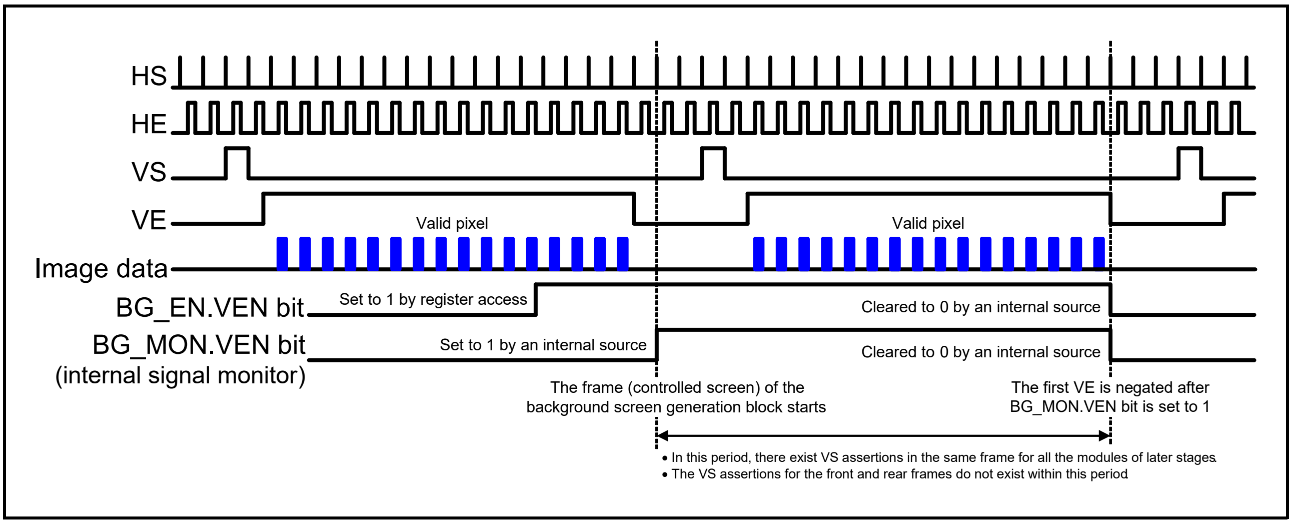


Figure 58.24 Control signals for register value reflection to internal operations

(3) Stopping and restarting in normal operation

To stop the GLCDC:

1. Confirm that the VEN bit of each module is cleared to 0.
2. Clear the BG_EN.EN bit (background plane operation enable) of the background screen generation module to 0.
3. Confirm that the BG_MON.EN bit (background plane operation monitor) of the background screen generation module has changed to 0. This bit is cleared to 0 only after operation has stopped. It is cleared at the frame end of the background screen generation module, not when the operations of all the modules are complete (not at the frame end of the output control module). If it is necessary to wait for all the modules to complete operations, a certain period (for example, a period equivalent to one line) is required.
4. Usually, a software reset can be safely applied (clearing of the BG_EN.SWRST bit) after confirming that the BG_MON.EN bit of the background screen generation module has changed to 0. (Even if the output signal returns to the initial value, no problems occur because the GLCDC has already entered the vertical blanking interval.)
5. When restarting the GLCDC by setting the relevant registers without applying a software reset, wait until sufficient time elapses after the BG_MON.EN bit becomes 0 (when the output control module output is the frame end) before starting the GLCDC. The GLCDC itself is not affected by this because the GLCDC starts operating after recognizing the assertion of the VS of the previous stage as the frame head. However, operation of some connected devices might be affected if a blanking interval or a period equivalent to one line is too short. For details, check the specifications of the connected device. When a software reset is applied, the register values are also initialized and almost all of the registers must be set again. Only the color palette data (CLUT memory content) is retained (only for a software reset after normal end).

Figure 58.25 shows the changes in signal lines and bits for the stop and restart in normal operation. Even if the background screen generation module is stopped by a clearing of the BG_EN.EN bit to 0, the GLCDC stops because of an abnormal operation sequence, not normal operation, if the GPX bus access is not complete at the frame end because of an underflow in graphics 1 or 2, inappropriate setting of graphics data access, or other undesirable conditions.

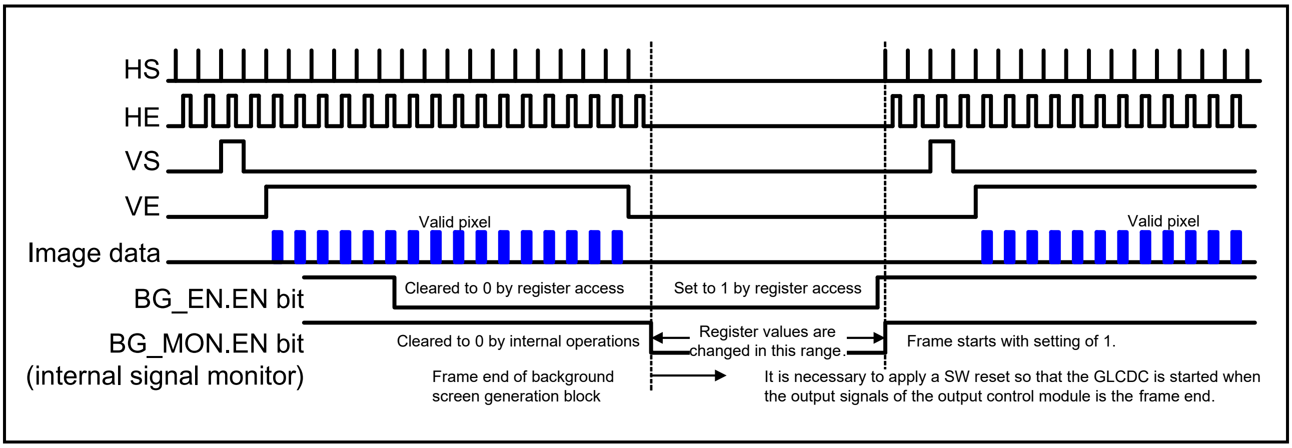


Figure 58.25 Stop and restart in normal operation

(4) Stopping and restarting in abnormal operation

Stop in abnormal operation occurs in the following cases:

- When during operation (BG_EN.EN or BG_MON.EN is set to 1), a software reset (BG_EN.SWRST bit is cleared to 0) or a reset is applied
- When unnecessary (unintended) access to the GPX bus occurs and a data cycle is not completed, even though the BG_EN.EN bit is cleared to 0 and so the BG_MON.EN bit is also cleared to 0.

In both cases, the GLCDC is internally initialized and the register access bus can be accessed normally (except for a hardware reset). However, the GPX bus might write unintended graphics data to the GLCDC. Therefore, it is advisable to apply a software reset (clear the BG_EN.SWRST bit to 0) and confirm that there is no unintended access to the GPX bus, even after confirming that both the BG_EN.EN and BG_MON.EN bits are cleared to 0. Next release the GLCDC from the software reset (set the BG_EN.SWRST bit to 1), set the relevant registers, and set the BG_EN.En and BG_EN.VEN bits in the background screen generation module to 1 to restart the GLCDC. For the procedure for checking the GPX bus state, see the *Arm® AHB Specification*. Use this procedure even after a reset when the GPX bus, a target device, or a target controller is not initialized. If these are initialized on a reset assertion for the GLCDC, a normal startup procedure can be used without checking the GPX bus state.

58.3.2 Screen Definition

The essential signals for GLCDC operations are generated by the background screen generation module, and the graphics 1 and 2 modules and output control module operate based on the sequentially transferred vertical and horizontal synchronization signals (VS and HS) and vertical and horizontal pixel enable signals (VE and HE). The reference point (frame start point) of the background screen cannot be determined by the output signals. The point shown in the figures is virtual and provided only for register settings. The reference points (frame start points) of the graphics 1 and 2 modules and output control module are the assertion of the vertical synchronization signal (VS), which is input (output) from the previous stage. Each module defines the valid pixel display area and special processing area for pixel data (rectangular areas for graphics 1 and 2) according to the position and width based on this reference point. Figure 58.26 shows the definition of the background screen, and Figure 58.27 shows the definition of the graphics 1 and 2 screen. The output control module performs correction for the entire valid pixel area (when both VE and HE are 1) output from the previous stage (graphics 2). (No registers are provided in the output control module for setting the correction area.) TCON in the output control module generates the control signals to be output based on the internal vertical and horizontal synchronization signals (VS and HS), and the change timing of the output signals can be freely modified within the valid setting range specifiable in the registers.

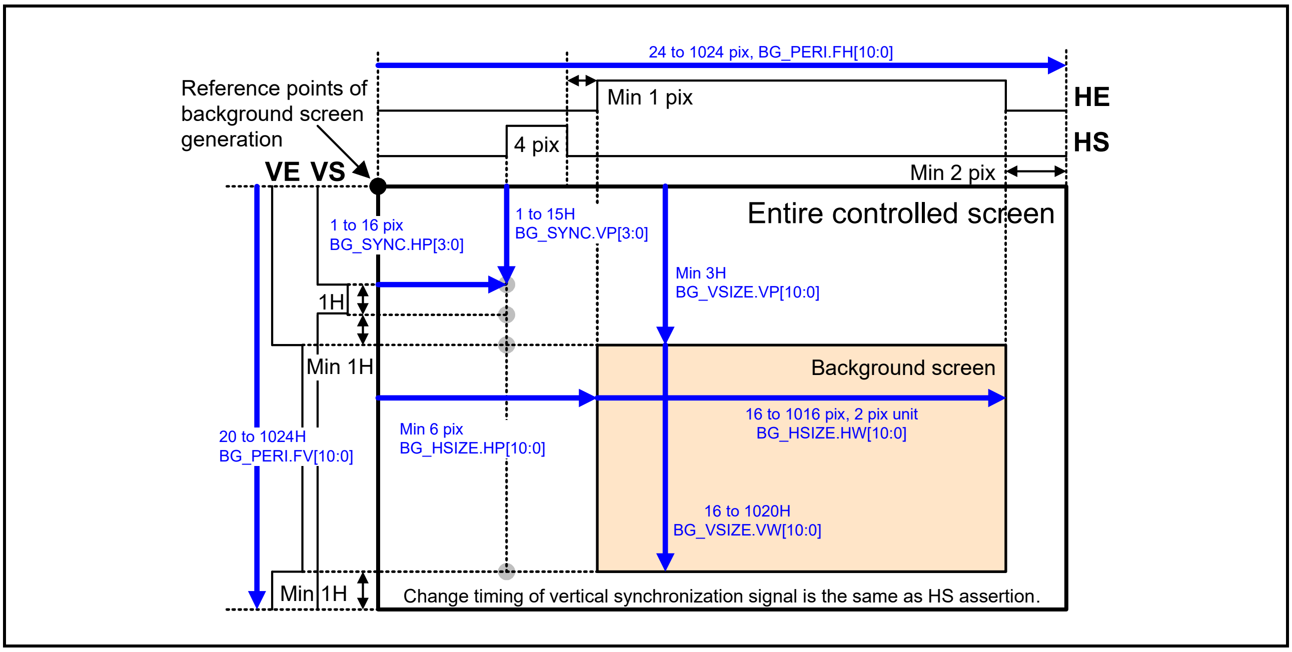


Figure 58.26 Internal definition of background screen

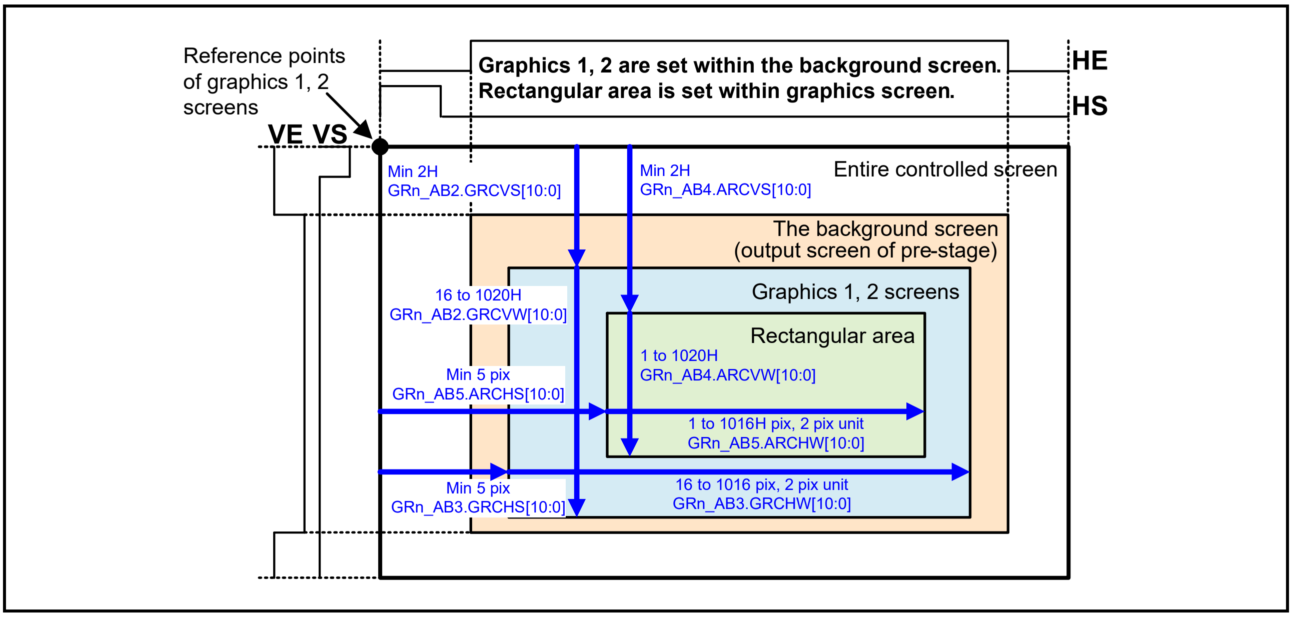


Figure 58.27 Internal definition of graphics 1 and 2 screens

58.3.3 Underflow and Interrupts

The GLCDC can detect the three types of status conditions described in this section. To detect each status condition, set the associated bit in the Status Detection Control Register (SYSCNT_DTCTEN) to 1.

58.3.3.1 Graphics 2 underflow detection

This function detects underflows in the graphics data interface block for graphics 2. The SYSCNT_STMON.L2UNDF flag sets to 1 if the graphics data cannot be read from the graphics data interface block (if valid data is not stored in the 4-stage ring buffer). The underflow is cleared as an internal state of graphics 2 on the VS (vertical synchronization) signal assertion of the previous stage. However, to clear the relevant bit in the Status Monitor Register of the system control block, the software must set the associated bit in the Status Clear Register to 1.

58.3.3.2 Graphics 1 underflow detection

This function detects underflows in the graphics data interface block for graphics 1. The SYSCNT_STMON.L1UNDF flag is set to 1 if the graphics data cannot be read from the graphics data interface block (if valid data is not stored in the 4-stage ring buffer). The underflow is cleared as an internal state of graphics 1 on the VS (vertical synchronization) signal assertion of the previous stage. However, to clear the relevant bit in the Status Monitor Register of the system control block, the software must set the associated bit in the Status Clear Register to 1.

58.3.3.3 Graphics 2 line detection

This function detects that the number of lines specified in the GR2_CLUTINT.LINE[10:0] bit was processed. The detection is performed on the HS (horizontal synchronization) signal assertion of the previous stage, not when valid pixels start to be processed. Each time the number of detected lines reach the value specified for graphics 2, the SYSCNT_STMON.VPOS flag sets to 1. As with underflow detection, to clear the relevant bit in the Status Monitor Register of the system control block, the software must set the associated bit in the Status Clear Register to 1.

58.3.3.4 Interrupts

The GLCDC provides three interrupt request output signals (GLCDC_L2UNDF, GLCDC_L1UNDF, and GLCDC_VPOS) that correspond to detection of the three status conditions. The GLCDC_L2UNDF, GLCDC_L1UNDF, and GLCDC_VPOS signals are associated with graphics 2 underflow detection, graphics 1 underflow detection, and graphics 2 line detection, respectively. Each of the interrupt request signals is asserted by setting the associated bit in the Interrupt Request Enable Register (SYSCNT_INTEN) to 1. Detecting the status and enabling the associated interrupt request can be controlled separately. Even if detection is enabled, the interrupt request signal is not asserted unless the interrupt request output signal is enabled. In addition, if the Status Monitor Register (SYSCNT_STMON) clears while the interrupt request output signal is asserted (the associated bit in the Status Clear Register (SYSCNT_STCLR) is set to 1), or if the associated bit in the Interrupt Request Enable Register (SYSCNT_INTEN) is cleared to 0, the interrupt request signal is negated. The interrupt request signal generation circuit is configured as shown in Figure 58.28. This circuit is glitch-free except for reset-induced glitches.

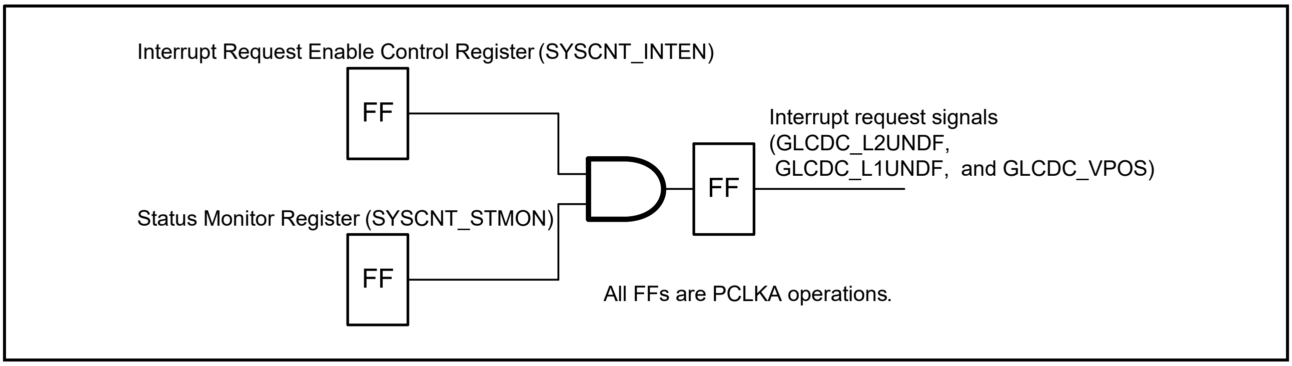


Figure 58.28 Interrupt request signal generation circuit

59. Internal Voltage Regulator

59.1 Overview

The MCU includes a linear regulator (LDO) that supplies voltage to the internal circuits and memory except for I/O, analog, USB, and battery backup power domain.

59.2 Operation

Table 59.1 lists the setting descriptions for the LDO mode pins, and Figure 59.1 shows the LDO mode settings. In LDO mode, the internal voltage is generated from VCC.

Table 59.1 Setting descriptions for the LDO mode pins

Parameter	Description
All VCC pins	• Connect each pin to the system power supply. • Connect each pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCL and VCL0 pins	Connect each pin to VSS through a 0.1-µF multilayer ceramic capacitor. Place the capacitor close to the pin.

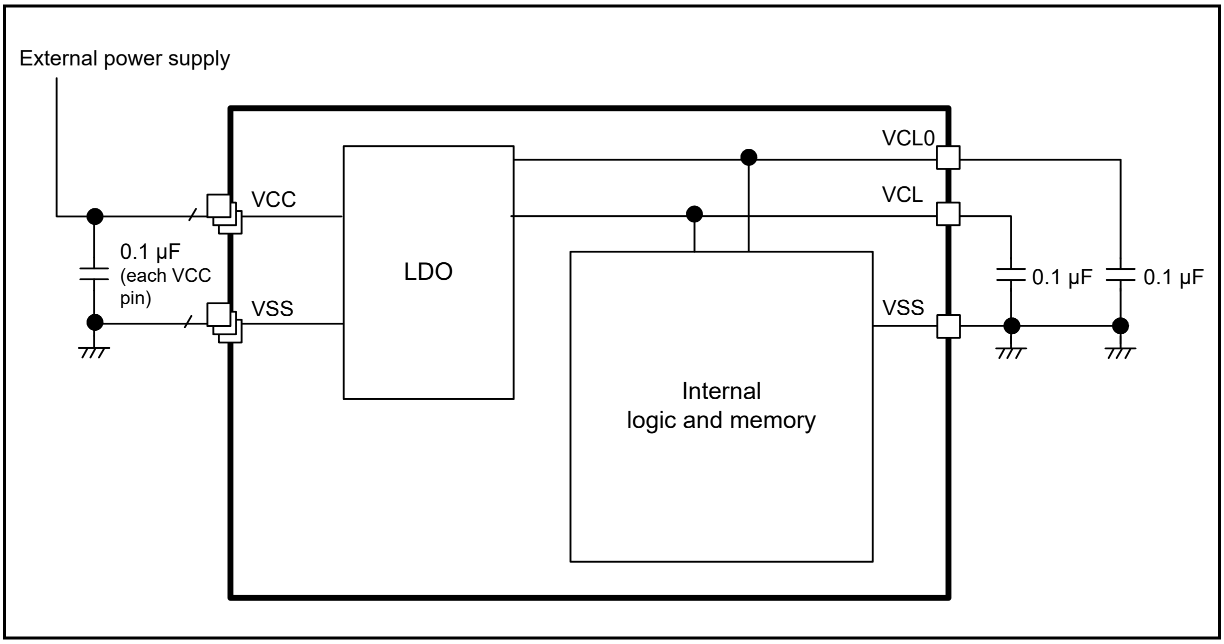


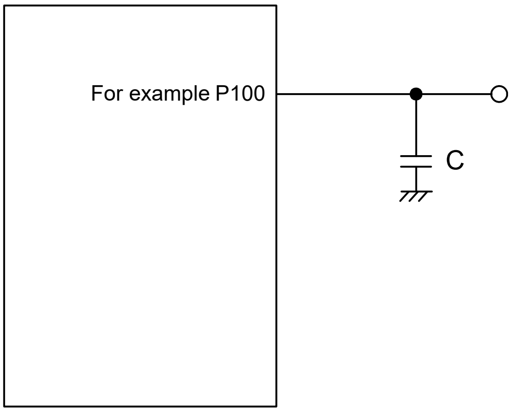
Figure 59.1 LDO mode settings

60. Electrical Characteristics

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

VCC = AVCC0 = VCC_USB = VBATT = 2.7 to 3.6 V, 2.7 ≤ VREFH0/VREFH ≤ AVCC0, VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V, VSS = AVSS0 = VREFL0/VREFL = VSS_USB = VSS1_USBHS = VSS2_USBHS = PVSS_USBHS = AVSS_USBHS = 0 V, Ta = Topr.

Figure 60.1 shows the timing conditions.



V_{OH} = VCC × 0.7, V_{OL} = VCC × 0.3
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
Load capacitance C = 30pF

Figure 60.1 Input or output timing measurement conditions

The measurement conditions of timing specification in each peripherals are recommended for the best peripheral operation, however make sure to adjust driving abilities of each pins to meet your conditions.

60.1 Absolute Maximum Ratings

Table 60.1 Absolute maximum ratings

Parameter	Symbol	Value	Unit
Power supply voltage	VCC, VCC_USB *2	-0.3 to +4.0	V
VBATT power supply voltage	VBATT	-0.3 to +4.0	V
Input voltage (except for 5V-tolerant ports*1)	V_{in}	-0.3 to VCC + 0.3	V
Input voltage (5V-tolerant ports*1)	V_{in}	-0.3 to + VCC + 4.0 (max 5.8)	V
Reference power supply voltage	VREFH/VREFH0	-0.3 to AVCC0 + 0.3	V
Analog power supply voltage	AVCC0 *2	-0.3 to +4.0	V
USBHS power supply voltage	VCC_USBHS	-0.3 to +4.0	V
USBHS analog power supply voltage	AVCC_USBHS	-0.3 to +4.0	V
Analog input voltage (except for P000 to P007)	V_{AN}	-0.3 to AVCC0 + 0.3	V
Analog input voltage (P000 to P007) when PGA differential input is disabled	V_{AN}	-0.3 to AVCC0 + 0.3	V
Analog input voltage (P000 to P002, P004 to P006) when PGA differential input is enabled	V_{AN}	-1.3 to AVCC0 + 0.3	V
Analog input voltage (P003, P007) when PGA differential input is enabled	V_{AN}	-0.8 to AVCC0 + 0.3	V
Operating temperature*3,*4,*5	T_{opr}	-40 to +85 -40 to +105	°C
Storage temperature	T_{stg}	-55 to +125	°C

Caution: Permanent damage to the MCU might result if absolute maximum ratings are exceeded.

Note 1. Ports P205, P206, P400, P401, P407 to P415, P511, P512, P708 to P713, and PB01 are 5V-tolerant.

Note 2. Connect AVCC0 and VCC_USB to VCC.

Note 3. See section 60.2.1, T_j/T_a Definition.

Note 4. Contact a Renesas Electronics sales office for information on derating operation when T_a = +85°C to +105°C. Derating is the systematic reduction of load for improved reliability.

Note 5. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, see section 1.3, Part Numbering.

Table 60.2 Recommended operating conditions

Parameter	Symbol	Value	Min	Typ	Max	Unit
Power supply voltages	VCC	When USB/SDRAM is not used	2.7	-	3.6	V
		When USB/SDRAM is used	3.0	-	3.6	V
	VSS		-	0	-	V
USB power supply voltages	VCC_USB, VCC_USBHS		-	VCC	-	V
	VSS_USB, AVSS_USBHS, PVSS_USBHS, VSS1_USBHS, VSS2_USBHS		-	0	-	V
VBATT power supply voltage	VBATT		1.8	-	3.6	V
Analog power supply voltages	AVCC0*1		-	VCC	-	V
	AVSS0		-	0	-	V

Note 1. Connect AVCC0 to VCC. When neither the A/D converter nor the D/A converter nor the comparator is in use, do not leave the AVCC0, VREFH/VREFH0, AVSS0, and VREFL/VREFL0 pins open. Connect the AVCC0 and VREFH/VREFH0 pins to VCC, and the AVSS0 and VREFL/VREFL0 pins to VSS, respectively.

60.2 DC Characteristics

60.2.1 T_j/T_a Definition

Table 60.3 DC characteristics

Conditions: Products with operating temperature (T_a) -40 to +105°C

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible junction temperature	T_j	-	125	°C	High-speed mode Low-speed mode Subosc-speed mode
			105*1		

Note: Make sure that $T_j = T_a + θja \times$ total power consumption (W), where total power consumption = $(VCC - V_{OH}) \times ΣI_{OH} + V_{OL} \times ΣI_{OL} + I_{CC}max \times VCC$.

Note 1. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, see section 1.3, Part Numbering. If the part number shows the operation temperature to 85°C, then Tj max is 105°C, otherwise, 125°C.

60.2.2 I/O V_{IH}, V_{IL}

Table 60.4 I/O V_{IH}, V_{IL}

Parameter					Symbol	Min	Typ	Max	Unit
Input voltage (except for Schmitt trigger input pins)	Peripheral function pin	EXTAL(external clock input), WAIT, SPI (except RSPCK)			V_{IH}	VCC × 0.8	-	-	V
					V_{IL}	-	-	VCC × 0.2	
		D00 to D15, DQ00 to DQ15			V_{IH}	VCC × 0.7	-	-	
					V_{IL}	-	-	VCC × 0.3	
		ETHERC			V_{IH}	2.3	-	-	
					V_{IL}	-	-	VCC × 0.2	
		IIC (SMBus)*1			V_{IH}	2.1	-	-	
					V_{IL}	-	-	0.8	
		IIC (SMBus)*2			V_{IH}	2.1	-	VCC + 3.6 (max 5.8)	
					V_{IL}	-	-	0.8	
Schmitt trigger input voltage		IIC (except for SMBus)*1			V_{IH}	VCC × 0.7	-	-	
					V_{IL}	-	-	VCC × 0.3	
					ΔV_T	VCC × 0.05	-	-	
		IIC (except for SMBus)*2			V_{IH}	VCC × 0.7	-	VCC + 3.6 (max 5.8)	
					V_{IL}	-	-	VCC × 0.3	
					ΔV_T	VCC × 0.05	-	-	
		5V-tolerant ports*3, *7			V_{IH}	VCC × 0.8	-	VCC + 3.6 (max 5.8)	
					V_{IL}	-	-	VCC × 0.2	
					ΔV_T	VCC × 0.05	-	-	
		RTCIC0, RTCIC1, RTCIC2	When using the Battery Backup Function	When VBATT power supply is selected	V_{IH}	V_{BATT} × 0.8	-	V_{BATT} + 0.3	
					V_{IL}	-	-	V_{BATT} × 0.2	
					ΔV_T	V_{BATT} × 0.05	-	-	
				When VCC power supply is selected	V_{IH}	VCC × 0.8	-	Higher voltage either VCC + 0.3 V or V_{BATT} + 0.3 V	
					V_{IL}	-	-	VCC × 0.2	
					ΔV_T	VCC × 0.05	-	-	
			When not using the Battery Backup Function		V_{IH}	VCC × 0.8	-	VCC + 0.3	
					V_{IL}	-	-	VCC × 0.2	
					ΔV_T	VCC × 0.05	-	-	
		Other input pins*4			V_{IH}	VCC × 0.8	-	-	
					V_{IL}	-	-	VCC × 0.2	
					ΔV_T	VCC × 0.05	-	-	
	Ports	5V-tolerant ports*5, *7			V_{IH}	VCC × 0.8	-	VCC + 3.6 (max 5.8)	
					V_{IL}	-	-	VCC × 0.2	
		Other input pins*6			V_{IH}	VCC × 0.8	-	-	
					V_{IL}	-	-	VCC × 0.2	

Note 1. SCL0_B (P204), SCL1_B, SDA1_B (total 3 pins).

Note 2. SCL0_A, SDA0_A, SCL0_B (P408), SDA0_B, SCL1_A, SDA1_A, SCL2, SDA2 (total 8 pins).

Note 3. RES and peripheral function pins associated with P205, P206, P400, P401, P407 to P415, P511, P512, P708 to P713, PB01 (total 23 pins).

Note 4. All input pins except for the peripheral function pins already described in the table.

Note 5. P205, P206, P400, P401, P407 to P415, P511, P512, P708 to P713, PB01 (total 22 pins).

Note 6. All input pins except for the ports already described in the table.

Note 7. When VCC is less than 2.7 V, the input voltage of 5 V-tolerant ports should be less than 3.6 V, otherwise breakdown might occur because the 5 V-tolerant ports are electrically controlled to not violate the breakdown voltage.

60.2.3 I/O I_{OH}, I_{OL}

Table 60.5 I/O I_{OH}, I_{OL}

Parameter			Symbol	Min	Typ	Max	Unit
Permissible output current (average value per pin)	Ports P008 to P010, P201	-	I_{OH}	-	--	-2.0	mA
			I_{OL}	-	-	2.0	mA
	Ports P014, P015	-	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
	Ports P205, P206, P407 to P415, P602, P708 to P713, PB01 (total 19 pins)	Low drive*1	I_{OH}	-	-	-2.0	mA
			I_{OL}	-	-	2.0	mA
		Middle drive*2	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		High drive*3	I_{OH}	-	-	-20	mA
			I_{OL}	-	-	20	mA
	Other output pins*4	Low drive*1	I_{OH}	-	-	-2.0	mA
			I_{OL}	-	-	2.0	mA
		Middle drive*2	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		High drive*3	I_{OH}	-	-	-16	mA
			I_{OL}	-	-	16	mA
Permissible output current (max value per pin)	Ports P008 to P010, P201	-	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
	Ports P014, P015	-	I_{OH}	-	-	-8.0	mA
			I_{OL}	-	-	8.0	mA
	Ports P205, P206, P407 to P415, P602, P708 to P713, PB01 (total 19 pins)	Low drive*1	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2	I_{OH}	-	-	-8.0	mA
			I_{OL}	-	-	8.0	mA
		High drive*3	I_{OH}	-	-	-40	mA
			I_{OL}	-	-	40	mA
	Other output pins*4	Low drive*1	I_{OH}	-	-	-4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2	I_{OH}	-	-	-8.0	mA
			I_{OL}	-	-	8.0	mA
		High drive*3	I_{OH}	-	-	-32	mA
			I_{OL}	-	-	32	mA
Permissible output current (max value total pins)	Maximum of all output pins		$\Sigma I_{OH\ (max)}$	-	-	-80	mA
			$\Sigma I_{OL\ (max)}$	-	-	80	mA

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in this table. The average output current indicates the average value of current measured during 100 µs.

Note 1. This is the value when low driving ability is selected in the port drive capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.

Note 2. This is the value when middle driving ability is selected in the port drive capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.

Note 3. This is the value when high driving ability is selected in the port drive capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.

Note 4. Except for P000 to P007, P200, which are input ports.

60.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 60.6 I/O V_{OH}, V_{OL}, and other characteristics

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	IIC		V_{OL}	-	-	0.4	V	I_{OL} = 3.0 mA
			V_{OL}	-	-	0.6		I_{OL} = 6.0 mA
	IIC*1		V_{OL}	-	-	0.4		I_{OL} = 15.0 mA (ICFER.FMPE = 1)
			V_{OL}	-	0.4	-		I_{OL} = 20.0 mA (ICFER.FMPE = 1)
	ETHERC		V_{OH}	VCC - 0.5	-	-		I_{OH} = -1.0 mA
			V_{OL}	-	-	0.4		I_{OL} = 1.0 mA
	Ports P205, P206, P407 to P415, P602, P708 to P713, PB01 (total 19 pins)*2		V_{OH}	VCC - 1.0	-	-		I_{OH} = -20 mA VCC = 3.3 V
			V_{OL}	-	-	1.0		I_{OL} = 20 mA VCC = 3.3 V
	Other output pins		V_{OH}	VCC - 0.5	-	-		I_{OH} = -1.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 1.0 mA
Input leakage current	RES		$\|I_{in}\|$	-	-	5.0	μA	V_{in} = 0 V V_{in} = 5.5 V
	Ports P000 to P002, P004 to P006, P200			-	-	1.0		V_{in} = 0 V V_{in} = VCC
	Ports P003, P007	Before initialization*3		-	-	45.0		V_{in} = 0 V V_{in} = VCC
		After initialization*4		-	-	1.0		V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5V-tolerant ports		$\|I_{TSI}\|$	-	-	5.0	μA	V_{in} = 0 V V_{in} = 5.5 V
	Other ports (except for ports P000 to P007, P200)			-	-	1.0		V_{in} = 0 V V_{in} = VCC
Input pull-up MOS current	Ports P0 to PB (except for ports P000 to P007)		I_p	-300	-	-10	μA	VCC = 2.7 to 3.6 V V_{in} = 0 V
Input capacitance	USB_DP, USB_DM, and ports P003, P007, P014, P015, P400, P401, P511, P512		C_{in}	-	-	16	pF	Vbias = 0V Vamp = 20mV f = 1 MHz T_a = 25°C
	Other input pins			-	-	8		

Note 1. SCL0_A, SDA0_A (total 2 pins).
Note 2. This is the value when high driving ability is selected in the port drive capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.
Note 3. P0nPFS.ASEL (n = 3 or 7) = 1.
Note 4. P0nPFS.ASEL (n = 3 or 7) = 0.

60.2.5 Operating and Standby Current

Table 60.7 Operating and standby current (1 of 2)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Supply current*1	High-speed mode	Maximum*2		I_{CC}*3	-	-	137*2	mA	ICLK = 120 MHz PCLKA = 120 MHz*7 PCLKB = 60 MHz PCLKC = 60 MHz PCLKD = 120 MHz FCLK = 60 MHz BCLK = 120 MHz
		CoreMark®*5			-	21	-		
		Normal mode	All peripheral clocks enabled, while (1) code executing from flash*4		-	34	-		
			All peripheral clocks disabled, while (1) code executing from flash*5, *6		-	14	-		
		Sleep mode*5, *6			-	12	46		
		Increase during BGO operation	Data flash P/E		-	6	-		
			Code flash P/E		-	8	-		
	Low-speed mode*5				-	2.4	-		ICLK = 1 MHz
	Subosc-speed mode*5				-	2	-		ICLK = 32.768 kHz
	Software Standby mode				-	1.8	18		Ta ≤ 85°C
					-	1.8	28		Ta ≤ 105°C
	Deep Software Standby mode	Power supplied to Standby SRAM and USB resume detecting unit			-	30	79	μA	Ta ≤ 85°C
					-	30	113	μA	Ta ≤ 105°C
		Power not supplied to SRAM or USB resume detecting unit	Power-on reset circuit low-power function disabled		-	13	33	μA	Ta ≤ 85°C
					-	13	40		Ta ≤ 105°C
			Power-on reset circuit low-power function enabled		-	6.3	28		Ta ≤ 85°C
					-	6.3	34		Ta ≤ 105°C
		Increase when the RTC and AGT are operating	When the low-speed on-chip oscillator (LOCO) is in use		-	5	-		-
			When a crystal oscillator for low clock loads is in use		-	1.0	-		-
			When a crystal oscillator for standard clock loads is in use		-	1.5	-		-
	RTC operating while VCC is off (with the battery backup function, only the RTC and sub-clock oscillator operate)		When a crystal oscillator for low clock loads is in use		-	0.9	-		V_{BATT} = 1.8 V, VCC = 0 V
					-	1.3	-		V_{BATT} = 3.3 V, VCC = 0 V
			When a crystal oscillator for standard clock loads is in use		-	1.1	-		V_{BATT} = 1.8 V, VCC = 0 V
					-	1.8	-		V_{BATT} = 3.3 V, VCC = 0 V
Analog power supply current	During 12-bit A/D conversion			AI_{CC}	-	0.8	1.1	mA	-
	During 12-bit A/D conversion with S/H amp				-	2.3	3.3	mA	-
	PGA (1ch)				-	1	3	mA	-
	ACMPHS (1unit)				-	100	150	μA	-
	Temperature sensor				-	0.1	0.2	mA	-
	During D/A conversion (per unit)		Without AMP output		-	0.1	0.2	mA	-
			With AMP output		-	0.6	1.1	mA	-
	Waiting for A/D, D/A conversion (all units)				-	0.9	1.6	mA	-
	ADC12, DAC12 in standby modes (all units)*8				-	2	8	μA	-
Reference power supply current (VREFH0)	During 12-bit A/D conversion (unit 0)			AI_{REFH0}	-	70	120	μA	-
	Waiting for 12-bit A/D conversion (unit 0)				-	0.07	0.5	μA	-
	ADC12 in standby modes (unit 0)				-	0.07	0.5	μA	-
Reference power supply current (VREFH)	During 12-bit A/D conversion (unit 1)			AI_{REFH}	-	70	120	μA	-
	During D/A conversion (per unit)		Without AMP output		-	0.1	0.4	mA	-
			With AMP ouput		-	0.1	0.4	mA	-
	Waiting for 12-bit A/D (unit 1), D/A (all units) conversion				-	0.07	0.8	μA	-
	ADC12 unit 1 in standby modes				-	0.07	0.8	μA	-

Table 60.7 Operating and standby current (2 of 2)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
USB operating current	Low speed	USB	$I_{CCUSBLS}$	-	3.5	6.5	mA	VCC_USB
		USBHS		-	10.5	13.5	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 0)
		USBHS		-	2.8	3.6	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 1)
	Full speed	USB	$I_{CCUSBFS}$	-	4.0	10.0	mA	VCC_USB
		USBHS		-	14	22	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 0)
		USBHS		-	6.5	13.0	mA	VCC_USBHS = AVCC_USBHS (PHYSET.HSEB = 1)
	High speed	USBHS	$I_{CCUSBHS}$	-	50	65	mA	VCC_USBHS = AVCC_USBHS
	Standby mode (direct power down)	USBHS	$I_{CCUSBSBY}$	-	0.5	4.5	μA	VCC_USBHS = AVCC_USBHS

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOS transistors in the off state.

Note 2. Measured with clocks supplied to the peripheral functions. This does not include the BGO operation.

Note 3. ICC depends on f (ICLK) as follows. (ICLK:PCLKA:PCLKB:PCLKC:PCLKD:BCK:EBCLK = 2:2:1:1:2:1:1)
ICC Max. = 0.84 × f + 37 (max. operation in High-speed mode)
ICC Typ. = 0.09 × f + 3.7 (normal operation in High-speed mode)
ICC Typ. = 0.6 × f + 1.8 (Low-speed mode 1)
ICC Max. = 0.08 × f + 37 (Sleep mode).

Note 4. This does not include the BGO operation.

Note 5. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

Note 6. FCLK, BCLK, PCLKA, PCLKB, PCLKC, and PCLKD are set to divided by 64 (3.75 MHz).

Note 7. When using ETHERC, GLCDC, DRW, and JPEG, PCLKA frequency is such that PCLKA = ICLK.

Note 8. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD16 (ADC120 Module Stop bit) and MSTPCRD.MSTPD15 (ADC121 Module Stop bit) are in the module-stop state. See section 47.6.8, Available Functions and Register Settings of AN000 to AN002, AN007, AN100 to AN102, and AN107.

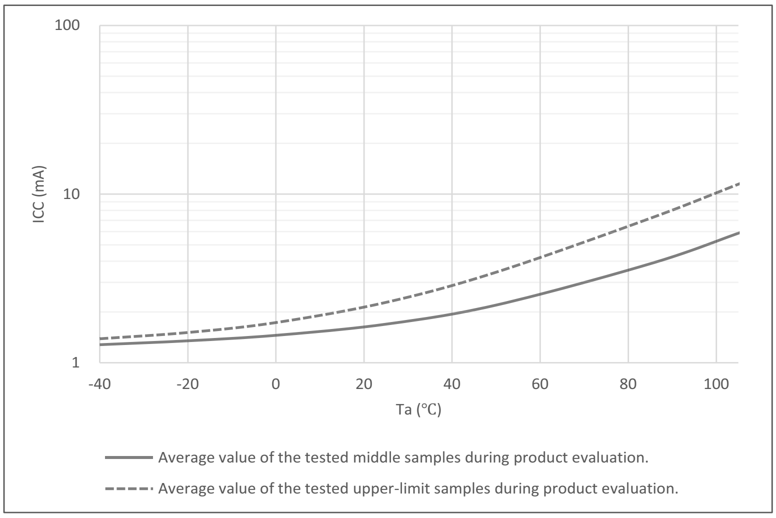


Figure 60.2 Temperature dependency in Software Standby mode (reference data)

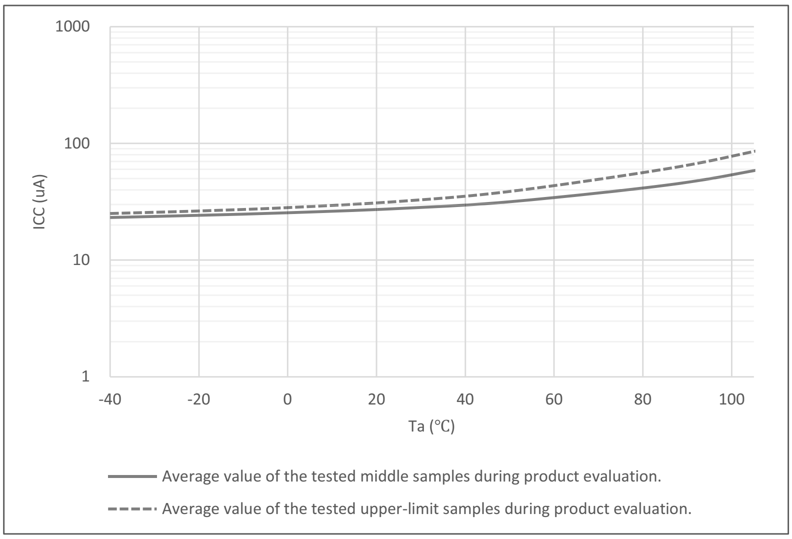


Figure 60.3 Temperature dependency in Deep Software Standby mode, power supplied to standby SRAM and USB resume detecting unit (reference data)

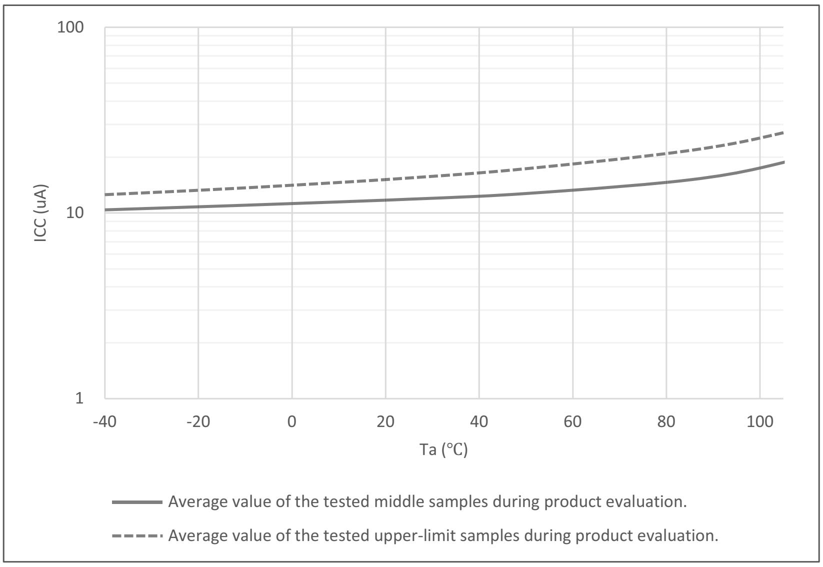


Figure 60.4 Temperature dependency in Deep Software Standby mode, power not supplied to SRAM or USB resume detecting unit, power-on reset circuit low-power function disabled (reference data)

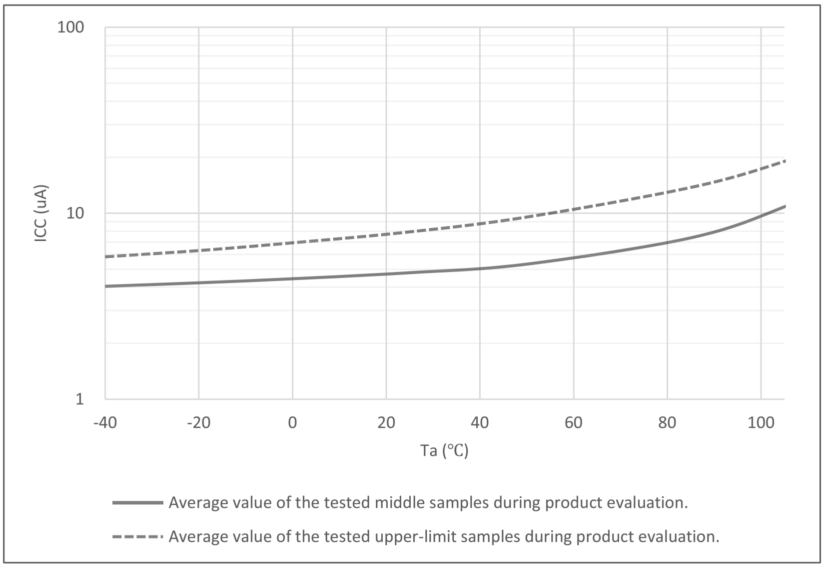


Figure 60.5 Temperature dependency in Deep Software Standby mode, power not supplied to SRAM or USB resume detecting unit, power-on reset circuit low-power function enabled (reference data)

60.2.6 VCC Rise and Fall Gradient and Ripple Frequency

Table 60.8 Rise and fall gradient characteristics

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
VCC rising gradient	Voltage monitor 0 reset disabled at startup	SrVCC	0.0084	-	20	ms/V	-
	Voltage monitor 0 reset enabled at startup		0.0084	-	-		-
	SCI/USB boot mode*1		0.0084	-	20		-
VCC falling gradient*2		SfVCC	0.0084	-	-	ms/V	-

Note 1. At boot mode, the reset from voltage monitor 0 is disabled regardless of the value of OFS1.LVDAS bit.
Note 2. This applies when VBATT is used.

Table 60.9 Rise and fall gradient and ripple frequency characteristics

The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (3.6 V) and lower limit (2.7 V). When the VCC change exceeds VCC ±10%, the allowable voltage change rising and falling gradient dt/dVCC must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r\ (VCC)}$	-	-	10	kHz	Figure 60.6 $V_{r\ (VCC)} \leq VCC \times 0.2$
		-	-	1	MHz	Figure 60.6 $V_{r\ (VCC)} \leq VCC \times 0.08$
		-	-	10	MHz	Figure 60.6 $V_{r\ (VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC	1.0	-	-	ms/V	When VCC change exceeds VCC ±10%

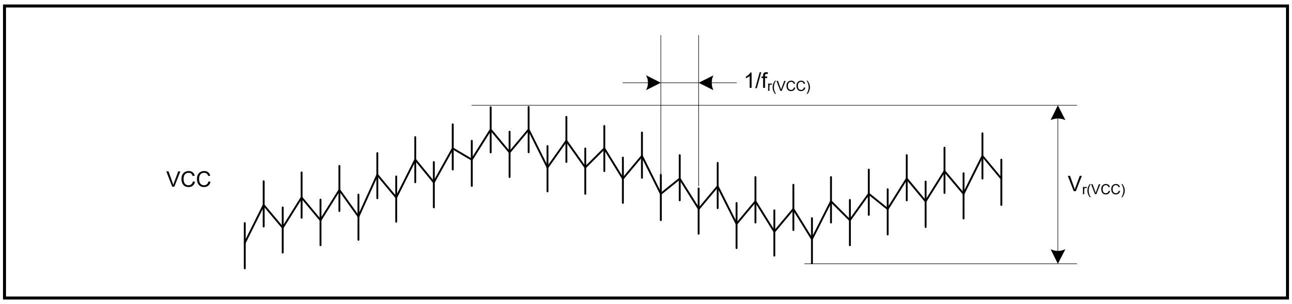


Figure 60.6 Ripple waveform

60.3 AC Characteristics

60.3.1 Frequency

Table 60.10 Operation frequency value in high-speed mode

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK*2)		f	-	-	120	MHz
	Peripheral module clock (PCLKA)*2			-	-	120	
	Peripheral module clock (PCLKB)*2			-	-	60	
	Peripheral module clock (PCLKC)*2			-*3	-	60	
	Peripheral module clock (PCLKD)*2			-	-	120	
	Flash interface clock (FCLK)*2			-*1	-	60	
	External bus clock (BCLK)*2			-	-	120	
	EBCLK pin output			-	-	60	
	SDCLK pin output	VCC ≥ 3.0 V		-	-	120	

Note 1. FCLK must run at a frequency of at least 4 MHz when programming or erasing the flash memory.
Note 2. See section 9, Clock Generation Circuit for the relationship between the ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK frequencies.
Note 3. When the ADC12 is used, the PCLKC frequency must be at least 1 MHz.

Table 60.11 Operation frequency value in low-speed mode

Parameter		Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*2	f	-	-	1	MHz
	Peripheral module clock (PCLKA)*2		-	-	1	
	Peripheral module clock (PCLKB)*2		-	-	1	
	Peripheral module clock (PCLKC)*2,*3		-*3	-	1	
	Peripheral module clock (PCLKD)*2		-	-	1	
	Flash interface clock (FCLK)*1, *2		-	-	1	
	External bus clock (BCLK)		-	-	1	
	EBCLK pin output		-	-	1	

Note 1. Programming or erasing the flash memory is disabled in low-speed mode.
Note 2. See section 9, Clock Generation Circuit for the relationship between the ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK frequencies.
Note 3. When the ADC12 is used, the PCLKC frequency must be set to at least 1 MHz.

Table 60.12 Operation frequency value in Subosc-speed mode

Parameter		Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*2	f	27.8	-	37.7	kHz
	Peripheral module clock (PCLKA)*2		-	-	37.7	
	Peripheral module clock (PCLKB)*2		-	-	37.7	
	Peripheral module clock (PCLKC)*2,*3		-	-	37.7	
	Peripheral module clock (PCLKD)*2		-	-	37.7	
	Flash interface clock (FCLK)*1, *2		27.8	-	37.7	
	External bus clock (BCLK)*2		-	-	37.7	
	EBCLK pin output		-	-	37.7	

Note 1. Programming or erasing the flash memory is disable in Subosc-speed mode.
Note 2. See section 9, Clock Generation Circuit for the relationship between the ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK frequencies.
Note 3. The ADC12 cannot be used.

60.3.2 Clock Timing

Table 60.13 Clock timing except for sub-clock oscillator (1 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
EBCLK pin output cycle time	t_{Bcyc}	16.6	-	-	ns	Figure 60.7
EBCLK pin output high pulse width	t_{CH}	3.3	-	-	ns	
EBCLK pin output low pulse width	t_{CL}	3.3	-	-	ns	
EBCLK pin output rise time	t_{Cr}	-	-	5.0	ns	
EBCLK pin output fall time	t_{Cf}	-	-	5.0	ns	
SDCLK pin output cycle time	t_{SDcyc}	8.33	-	-	ns	
SDCLK pin output high pulse width	t_{CH}	1.0	-	-	ns	
SDCLK pin output low pulse width	t_{CL}	1.0	-	-	ns	
SDCLK pin output rise time	t_{Cr}	-	-	3.0	ns	
SDCLK pin output fall time	t_{Cf}	-	-	3.0	ns	

Table 60.13 Clock timing except for sub-clock oscillator (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time		t_{EXcyc}	41.66	-	-	ns	Figure 60.8
EXTAL external clock input high pulse width		t_{EXH}	15.83	-	-	ns	
EXTAL external clock input low pulse width		t_{EXL}	15.83	-	-	ns	
EXTAL external clock rise time		t_{EXr}	-	-	5.0	ns	
EXTAL external clock fall time		t_{EXf}	-	-	5.0	ns	
Main clock oscillator frequency		f_{MAIN}	8	-	24	MHz	-
Main clock oscillation stabilization wait time (crystal) *1		$t_{MAINOSCWT}$	-	-	-*1	ms	Figure 60.9
LOCO clock oscillation frequency		f_{LOCO}	27.8528	32.768	37.6832	kHz	-
LOCO clock oscillation stabilization wait time		t_{LOCOWT}	-	-	60.4	μs	Figure 60.10
ILOCO clock oscillation frequency		f_{ILOCO}	12.75	15	17.25	kHz	-
MOCO clock oscillation frequency		F_{MOCO}	6.8	8	9.2	MHz	-
MOCO clock oscillation stabilization wait time		t_{MOCOWT}	-	-	15.0	μs	-
HOCO clock oscillator oscillation frequency	Without FLL	f_{HOCO16}	15.78	16	16.22	MHz	-20 ≤ Ta ≤ 105°C
		f_{HOCO18}	17.75	18	18.25		
		f_{HOCO20}	19.72	20	20.28		
		f_{HOCO16}	15.71	16	16.29		-40 ≤ Ta ≤ -20°C
		f_{HOCO18}	17.68	18	18.32		
		f_{HOCO20}	19.64	20	20.36		
	With FLL	f_{HOCO16}	15.955	16	16.045		-40 ≤ Ta ≤ 105°C Sub-clock frequency accuracy is ±50 ppm.
		f_{HOCO18}	17.949	18	18.051		
		f_{HOCO20}	19.944	20	20.056		
HOCO clock oscillation stabilization wait time*2		t_{HOCOWT}	-	-	64.7	μs	-
FLL stabilization wait time		t_{FLLWT}	-	-	1.8	ms	-
PLL clock frequency		f_{PLL}	120	-	240	MHz	-
PLL clock oscillation stabilization wait time		t_{PLLWT}	-	-	174.9	μs	Figure 60.11

Note 1. When setting up the main clock oscillator, ask the oscillator manufacturer for an oscillation evaluation and use the results as the recommended oscillation stabilization time. Set the MOSCWTCR register to a value equal to or greater than the recommended value.
After changing the setting in the MOSCCR.MOSTP bit to start main clock operation, read the OSCSF.MOSCSF flag to confirm that it is 1, and then start using the main clock oscillator.

Note 2. This is the time from release from reset state until the HOCO oscillation frequency (fHOCO) reaches the range for guaranteed operation.

Table 60.14 Clock timing for the sub-clock oscillator

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Sub-clock frequency	f_{SUB}	-	32.768	-	kHz	-
Sub-clock oscillation stabilization wait time	$t_{SUBOSCWT}$	-	-	*1	s	Figure 60.12

Note 1. When setting up the sub-clock oscillator, ask the oscillator manufacturer for an oscillation evaluation and use the results as the recommended oscillation stabilization time.
After changing the setting in the SOSCCR.SOSTP bit to start sub-clock operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization time elapses with an adequate margin. Two times the value shown is recommended.

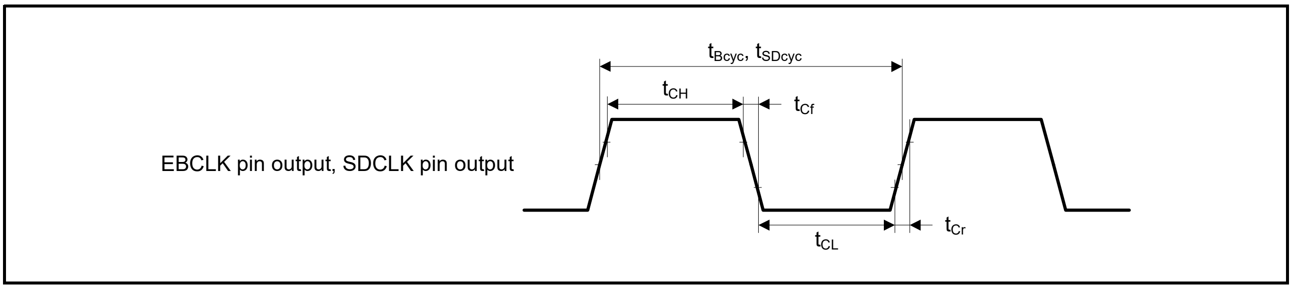


Figure 60.7 EBCLK and SDCLK output timing

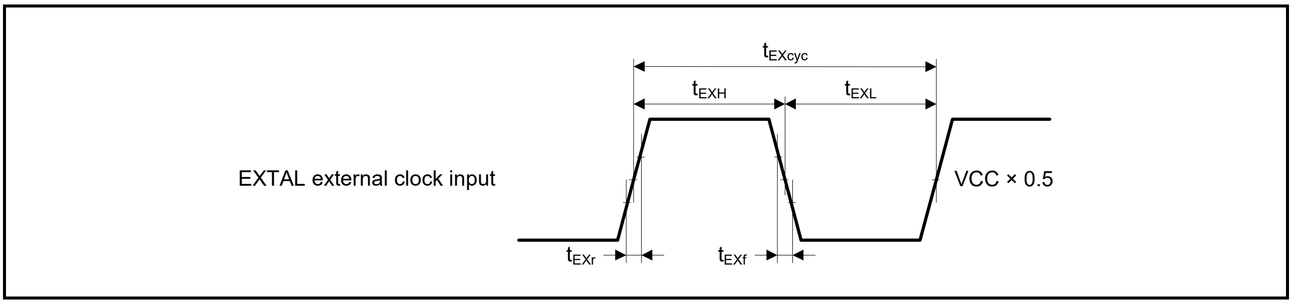


Figure 60.8 EXTAL external clock input timing

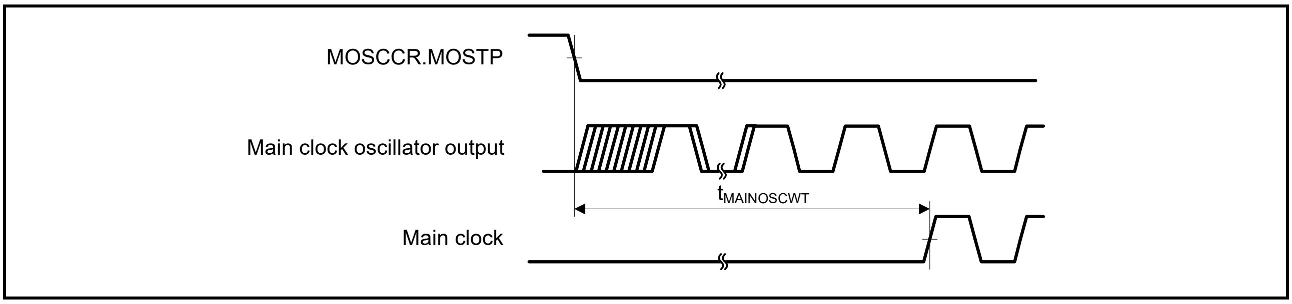


Figure 60.9 Main clock oscillation start timing

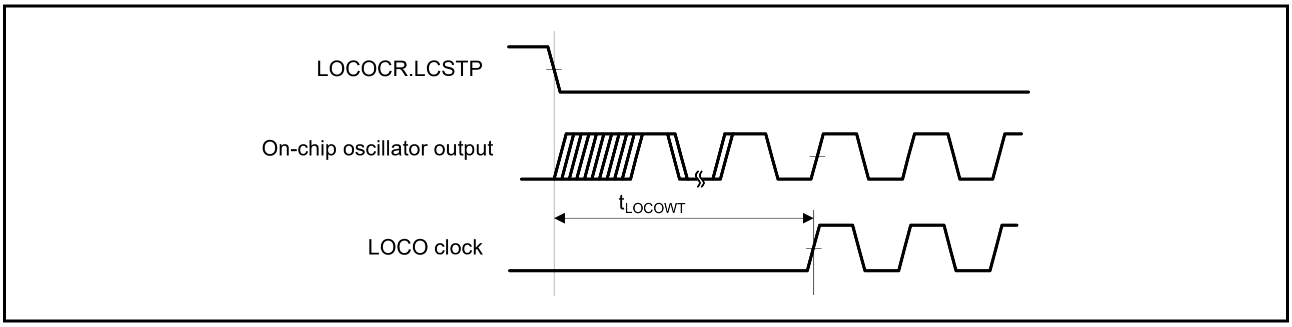


Figure 60.10 LOCO clock oscillation start timing

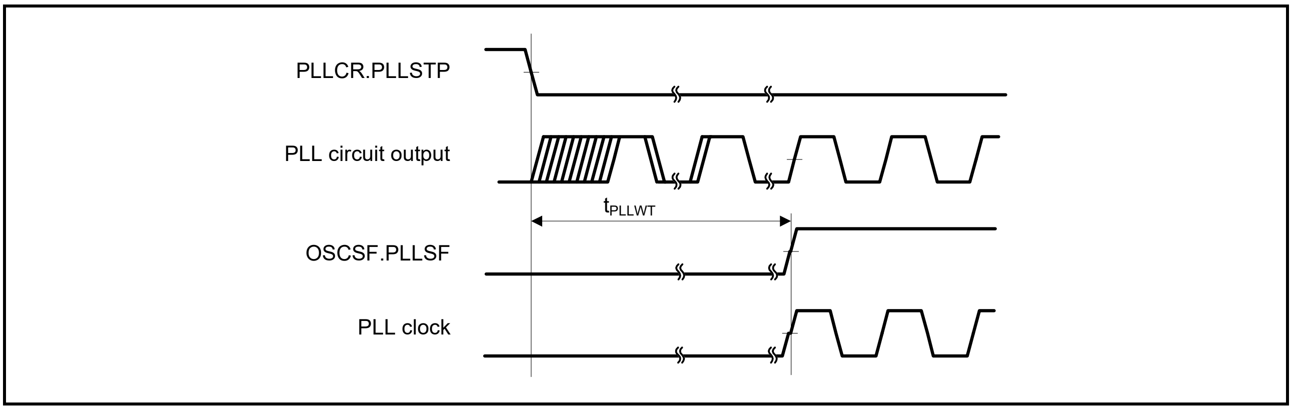


Figure 60.11 PLL clock oscillation start timing

Note: Only operate the PLL is operated after main clock oscillation has stabilized.

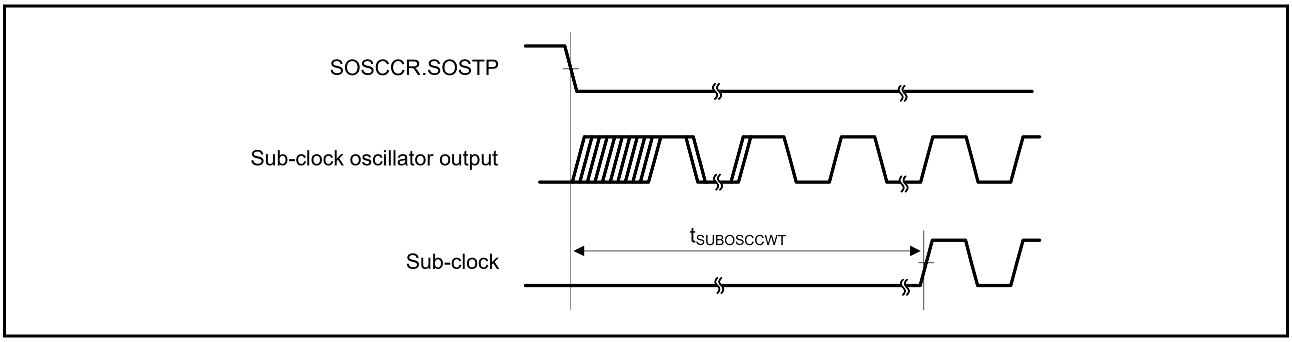


Figure 60.12 Sub-clock oscillation start timing

60.3.3 Reset Timing

Table 60.15 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	Power-on	t_{RESWP}	1	-	-	ms	Figure 60.13
	Deep Software Standby mode	t_{RESWD}	0.6	-	-	ms	Figure 60.14
	Software Standby mode, Subosc-speed mode	t_{RESWS}	0.3	-	-	ms	
	All other	t_{RESW}	200	-	-	μs	
Wait time after RES cancellation		t_{RESWT}	-	29	33	μs	Figure 60.13
Wait time after internal reset cancellation (IWDT reset, WDT reset, software reset, SRAM parity error reset, SRAM ECC error reset, bus master MPU error reset, bus slave MPU error reset, stack pointer error reset)		t_{RESW2}	-	320	408	μs	-

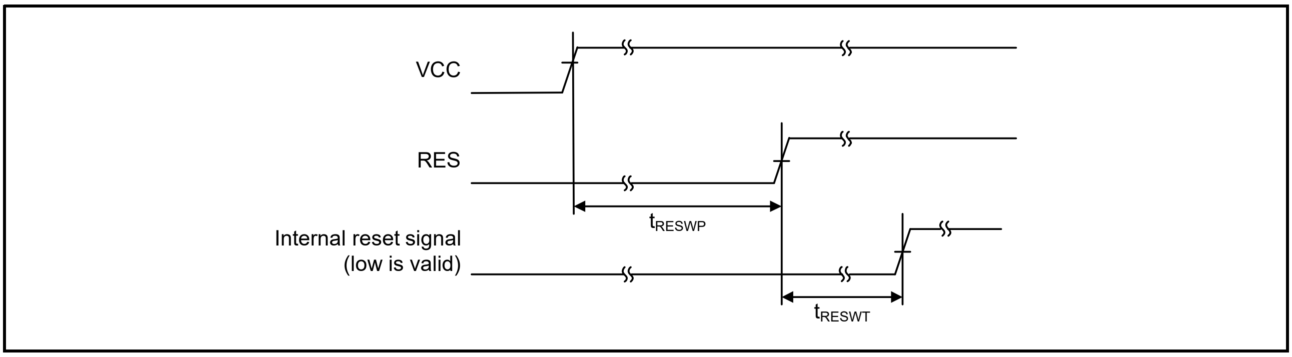


Figure 60.13 Power-on reset timing

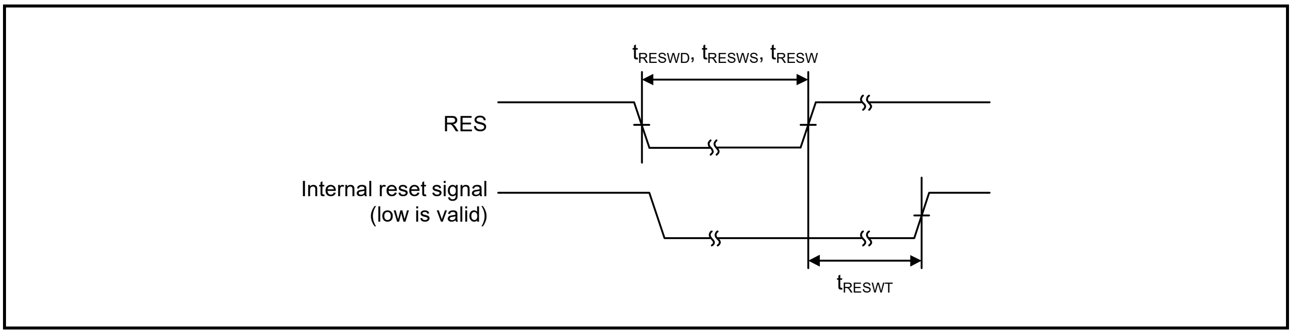


Figure 60.14 Reset input timing

60.3.4 Wakeup Timing

Table 60.16 Timing of recovery from low power modes

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator*2	t_{SBYMC}	-	2.4*9	2.8*9	ms	Figure 60.15 The division ratio of all oscillators is 1.
		System clock source is PLL with main clock oscillator*3	t_{SBYPC}	-	2.7*9	3.2*9	ms	
	External clock input to main clock oscillator	System clock source is main clock oscillator*4	t_{SBYEX}	-	230*9	280*9	µs	
		System clock source is PLL with main clock oscillator*5	t_{SBYPE}	-	570*9	700*9	µs	
	System clock source is sub-clock oscillator*8		t_{SBYSC}	-	1.2*9	1.3*9	ms	
	System clock source is LOCO*8		t_{SBYLO}	-	1.2*9	1.4*9	ms	
	System clock source is HOCO clock oscillator*6		t_{SBYHO}	-	240*9, *10	310 *9, *10	µs	
	System clock source is MOCO clock oscillator*7		t_{SBYMO}	-	220*9	300*9	µs	
Recovery time from Deep Software Standby mode			t_{DSBY}	-	0.65	1.0	ms	Figure 60.16
Wait time after cancellation of Deep Software Standby mode			t_{DSBYWT}	34	-	35	t_{cyc}	
Recovery time from Software Standby mode to Snooze mode	High-speed mode when system clock source is HOCO (20 MHz)		t_{SNZ}	-	35*9, *10	71 *9, *10	µs	Figure 60.17
	High-speed mode when system clock source is MOCO (8 MHz)		t_{SNZ}	-	11*9	14*9	µs	

Note 1. The recovery time is determined by the system clock source. When multiple oscillators are active, the recovery time can be determined with the following equation:
Total recovery time = recovery time for an oscillator as the system clock source + the longest oscillation stabilization time of any oscillators requiring longer stabilization times than the system clock source + 2 LOCO cycles (when LOCO is operating) + 3 SOSC cycles (when Subosc is oscillating and MSTPC0 = 0 (CAC module stop)).

Note 2. When the frequency of the crystal is 24 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 05h). For other settings (MOSCWTCR is set to Xh), the recovery time can be determined with the following equation:
t_{SBYMC} (MOSCWTCR = Xh) = t_{SBYMC} (MOSCWTCR = 05h) + ($t_{MAINOSCWT}$ (MOSCWTCR = Xh) - $t_{MAINOSCWT}$ (MOSCWTCR = 05h))

Note 3. When the frequency of PLL is 240 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 05h). For other settings (MOSCWTCR is set to Xh), the recovery time can be determined with the following equation:
t_{SBYMC} (MOSCWTCR = Xh) = t_{SBYMC} (MOSCWTCR = 05h) + ($t_{MAINOSCWT}$ (MOSCWTCR = Xh) - $t_{MAINOSCWT}$ (MOSCWTCR = 05h))

Note 4. When the frequency of the external clock is 24 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 00h). For other settings (MOSCWTCR is set to Xh), the recovery time can be determined with the following equation:
t_{SBYMC} (MOSCWTCR = Xh) = t_{SBYMC} (MOSCWTCR = 00h) + ($t_{MAINOSCWT}$ (MOSCWTCR = Xh) - $t_{MAINOSCWT}$ (MOSCWTCR = 00h))

Note 5. When the frequency of PLL is 240 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 00h). For other settings (MOSCWTCR is set to Xh), the recovery time can be determined with the following equation:
t_{SBYMC} (MOSCWTCR = Xh) = t_{SBYMC} (MOSCWTCR = 00h) + ($t_{MAINOSCWT}$ (MOSCWTCR = Xh) - $t_{MAINOSCWT}$ (MOSCWTCR = 00h))

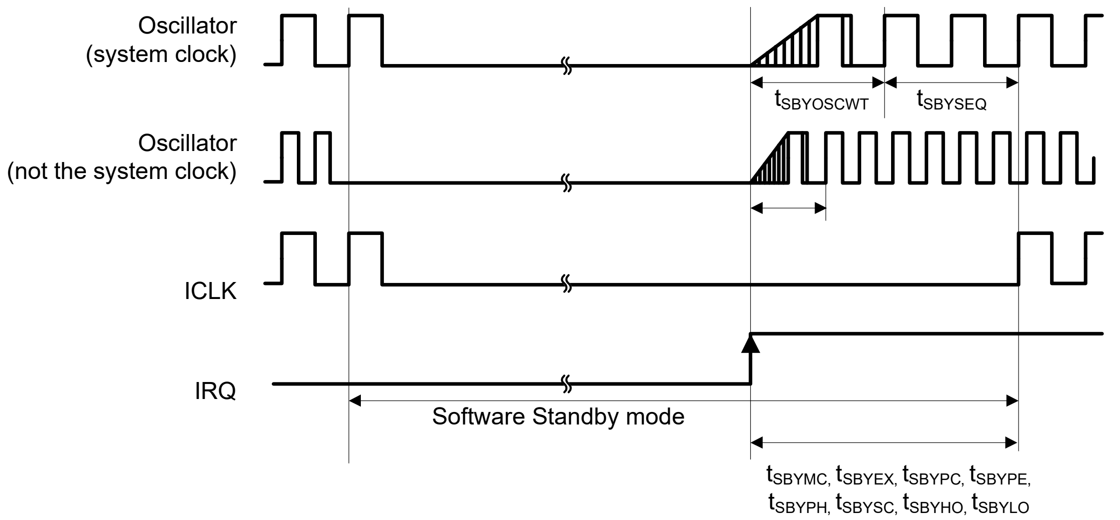
Note 6. The HOCO frequency is 20 MHz.

Note 7. The MOCO frequency is 8 MHz.

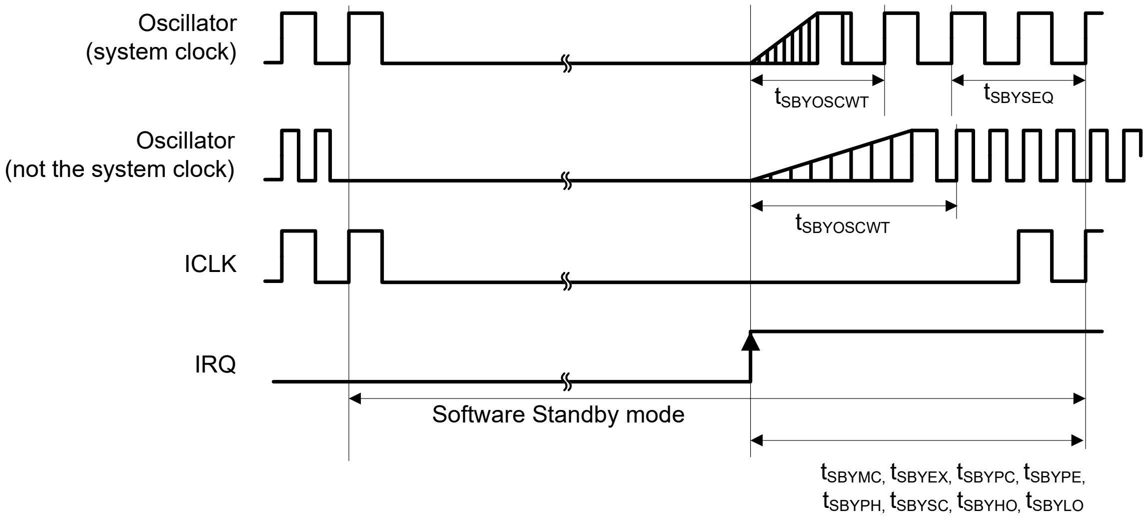
Note 8. In Subosc-speed mode, the sub-clock oscillator or LOCO continues oscillating in Software Standby mode.

Note 9. When the SNZCR.RXDREQEN bit is set to 0, the following time is added as the power supply recovery time:
STCONR.STCON[1:0] = 00b:16 µs (typical), 34 µs (maximum)
STCONR.STCON[1:0] = 11b:16 µs (typical), 104 µs (maximum).

Note 10. When the SNZCR.RXDREQEN bit is set to 0, 16 µs (typical) or 18 µs (maximum) is added as the HOCO wait time.



When stabilization of the system clock oscillator is slower



When stabilization of an oscillator other than the system clock is slower

Figure 60.15 Software Standby mode cancellation timing

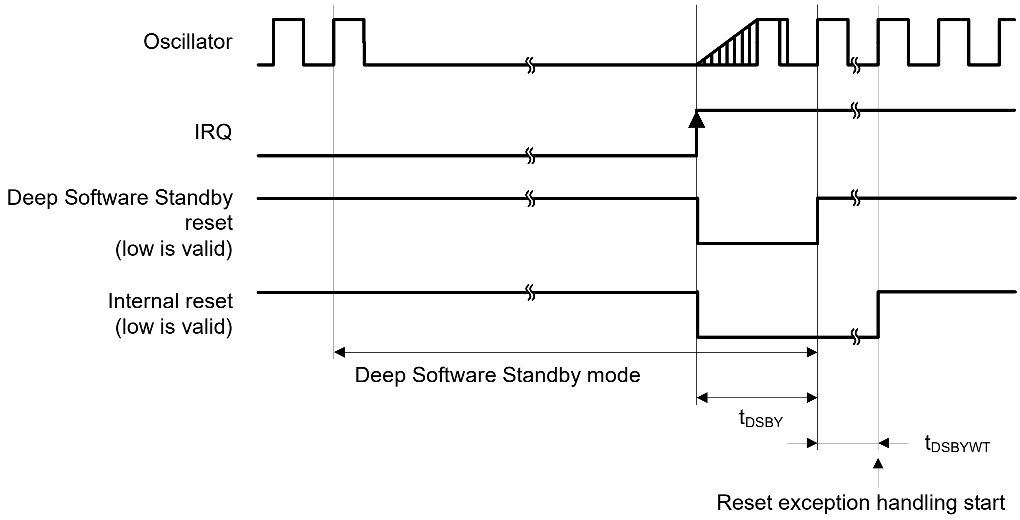
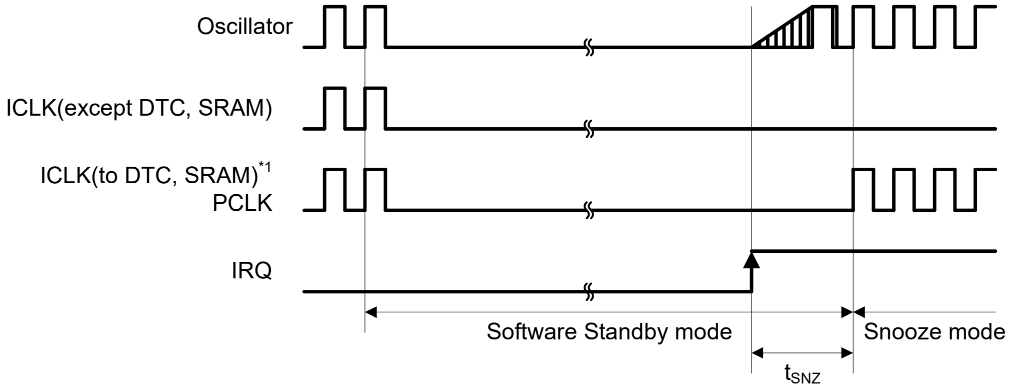


Figure 60.16 Deep Software Standby mode cancellation timing



Note 1. When SNZCR.SNZDTCEN is set to 1, ICLK is supplied to DTC and SRAM.

Figure 60.17 Recovery timing from Software Standby mode to Snooze mode

60.3.5 NMI and IRQ Noise Filter

Table 60.17 NMI and IRQ noise filter

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
NMI pulse width	t_{NMIW}	200	-	-	ns	NMI digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$[*1]	-	-			$t_{Pcyc} \times 2 > 200$ ns
		200	-	-		NMI digital filter enabled	$t_{NMICK} \times 3 \leq 200$ ns
		$t_{NMICK} \times 3.5$[*2]	-	-			$t_{NMICK} \times 3 > 200$ ns
IRQ pulse width	t_{IRQW}	200	-	-	ns	IRQ digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$[*1]	-	-			$t_{Pcyc} \times 2 > 200$ ns
		200	-	-		IRQ digital filter enabled	$t_{IRQCK} \times 3 \leq 200$ ns
		$t_{IRQCK} \times 3.5$[*3]	-	-			$t_{IRQCK} \times 3 > 200$ ns

Note: 200 ns minimum in Software Standby mode.
Note: If the clock source is switched, add 4 clock cycles of the switched source.

Note 1. t_{Pcyc} indicates the PCLKB cycle.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock.

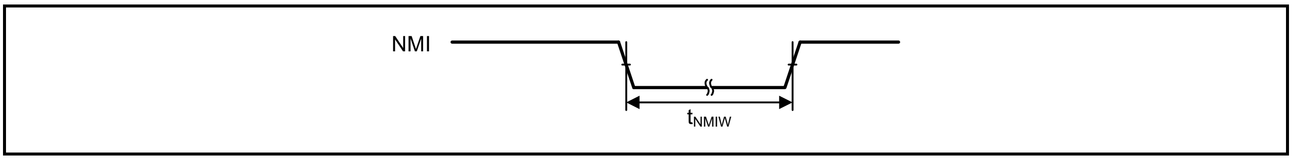


Figure 60.18 NMI interrupt input timing

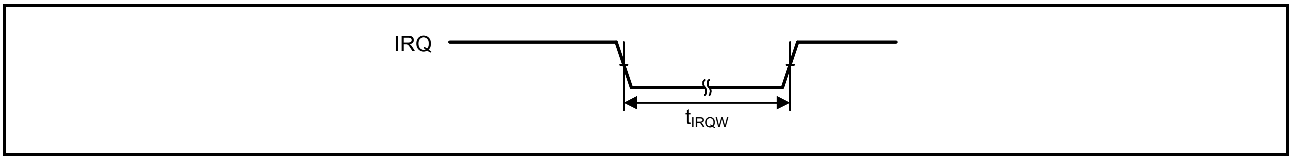


Figure 60.19 IRQ interrupt input timing

60.3.6 Bus Timing

Table 60.18 Bus timing (1 of 2)

Condition 1: When using the CS area controller (CSC).
BCLK = 8 to 120 MHz, EBCLK = 8 to 60 MHz
VCC = AVCC0 = VCC_USB = VBATT = 2.7 to 3.6 V, VREFH/VREFH0 = 2.7 V to AVCC0,
VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 30 pF
EBCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 2: When using the SDRAM area controller (SDRAMC).
BCLK = SDCLK = 8 to 120 MHz
VCC = AVCC0 = VCC_USB = VBATT = 3.0 to 3.6 V, VREFH/VREFH0 = 3.0 V to AVCC0,
VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
High drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 3: When using the SDRAM area controller (SDRAMC) and CS area controller (CSC) simultaneously.
BCLK = SDCLK = 8 to 60 MHz
VCC = AVCC0 = VCC_USB = VBATT = 3.0 to 3.6 V, VREFH/VREFH0 = 3.0 V to AVCC0,
VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
High drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter	Symbol	Min	Max	Unit	Test conditions
Address delay	t_{AD}	-	12.5	ns	Figure 60.20 to Figure 60.25
Byte control delay	t_{BCD}	-	12.5	ns	
CS delay	t_{CSD}	-	12.5	ns	
ALE delay time	t_{ALED}	-	12.5	ns	
RD delay	t_{RSD}	-	12.5	ns	
Read data setup time	t_{RDS}	12.5	-	ns	
Read data hold time	t_{RDH}	0	-	ns	
WR/WRn delay	t_{WRD}	-	12.5	ns	
Write data delay	t_{WDD}	-	12.5	ns	
Write data hold time	t_{WDH}	0	-	ns	
WAIT setup time	t_{WTS}	12.5	-	ns	Figure 60.26
WAIT hold time	t_{WTH}	0	-	ns	

Table 60.18 Bus timing (2 of 2)

Condition 1: When using the CS area controller (CSC).
BCLK = 8 to 120 MHz, EBCLK = 8 to 60 MHz
VCC = AVCC0 = VCC_USB = VBATT = 2.7 to 3.6 V, VREFH/VREFH0 = 2.7 V to AVCC0,
VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 30 pF
EBCLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
Others: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 2: When using the SDRAM area controller (SDRAMC).
BCLK = SDCLK = 8 to 120 MHz
VCC = AVCC0 = VCC_USB = VBATT = 3.0 to 3.6 V, VREFH/VREFH0 = 3.0 V to AVCC0,
VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
High drive output is selected in the port drive capability bit in the PmnPFS register.

Condition 3: When using the SDRAM area controller (SDRAMC) and CS area controller (CSC) simultaneously.
BCLK = SDCLK = 8 to 60 MHz
VCC = AVCC0 = VCC_USB = VBATT = 3.0 to 3.6 V, VREFH/VREFH0 = 3.0 V to AVCC0,
VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V
Output load conditions: VOH = VCC × 0.5, VOL = VCC × 0.5, C = 15 pF
High drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter	Symbol	Min	Max	Unit	Test conditions
Address delay 2 (SDRAM)	t_{AD2}	0.8	6.8	ns	Figure 60.27 to Figure 60.33
CS delay 2 (SDRAM)	t_{CSD2}	0.8	6.8	ns	
DQM delay (SDRAM)	t_{DQMD}	0.8	6.8	ns	
CKE delay (SDRAM)	t_{CKED}	0.8	6.8	ns	
Read data setup time 2 (SDRAM)	t_{RDS2}	2.9	-	ns	
Read data hold time 2 (SDRAM)	t_{RDH2}	1.5	-	ns	
Write data delay 2 (SDRAM)	t_{WDD2}	-	6.8	ns	
Write data hold time 2 (SDRAM)	t_{WDH2}	0.8	-	ns	
WE delay (SDRAM)	t_{WED}	0.8	6.8	ns	
RAS delay (SDRAM)	t_{RASD}	0.8	6.8	ns	
CAS delay (SDRAM)	t_{CASD}	0.8	6.8	ns	

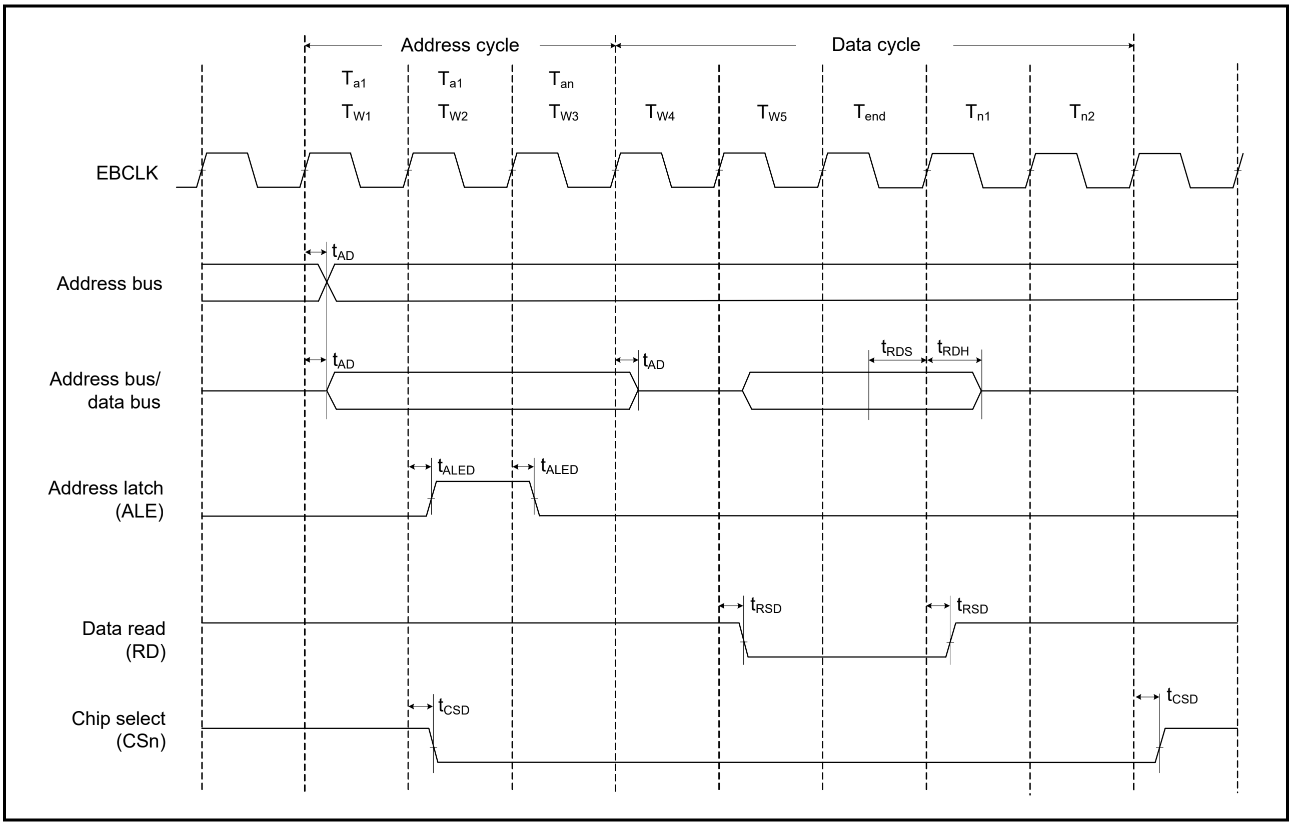


Figure 60.20 Address/data multiplexed bus read access timing

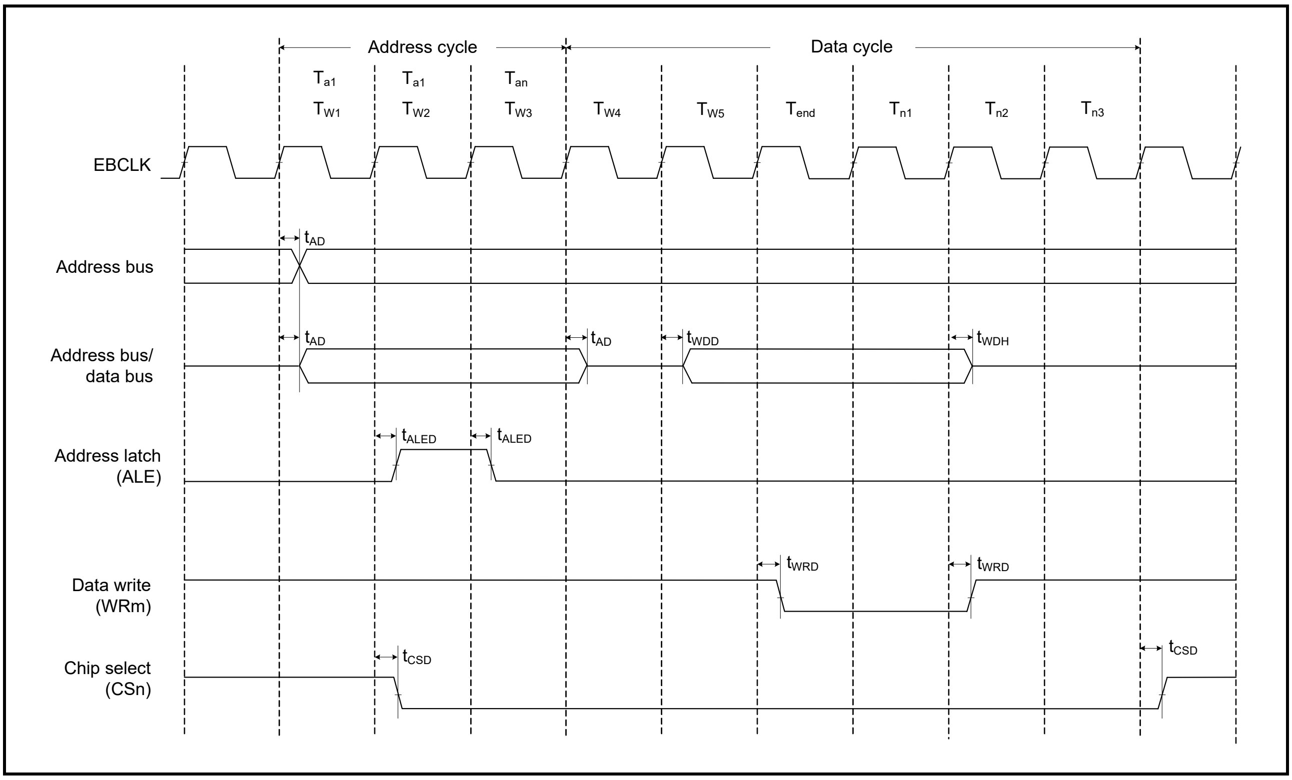


Figure 60.21 Address/data multiplexed bus write access timing

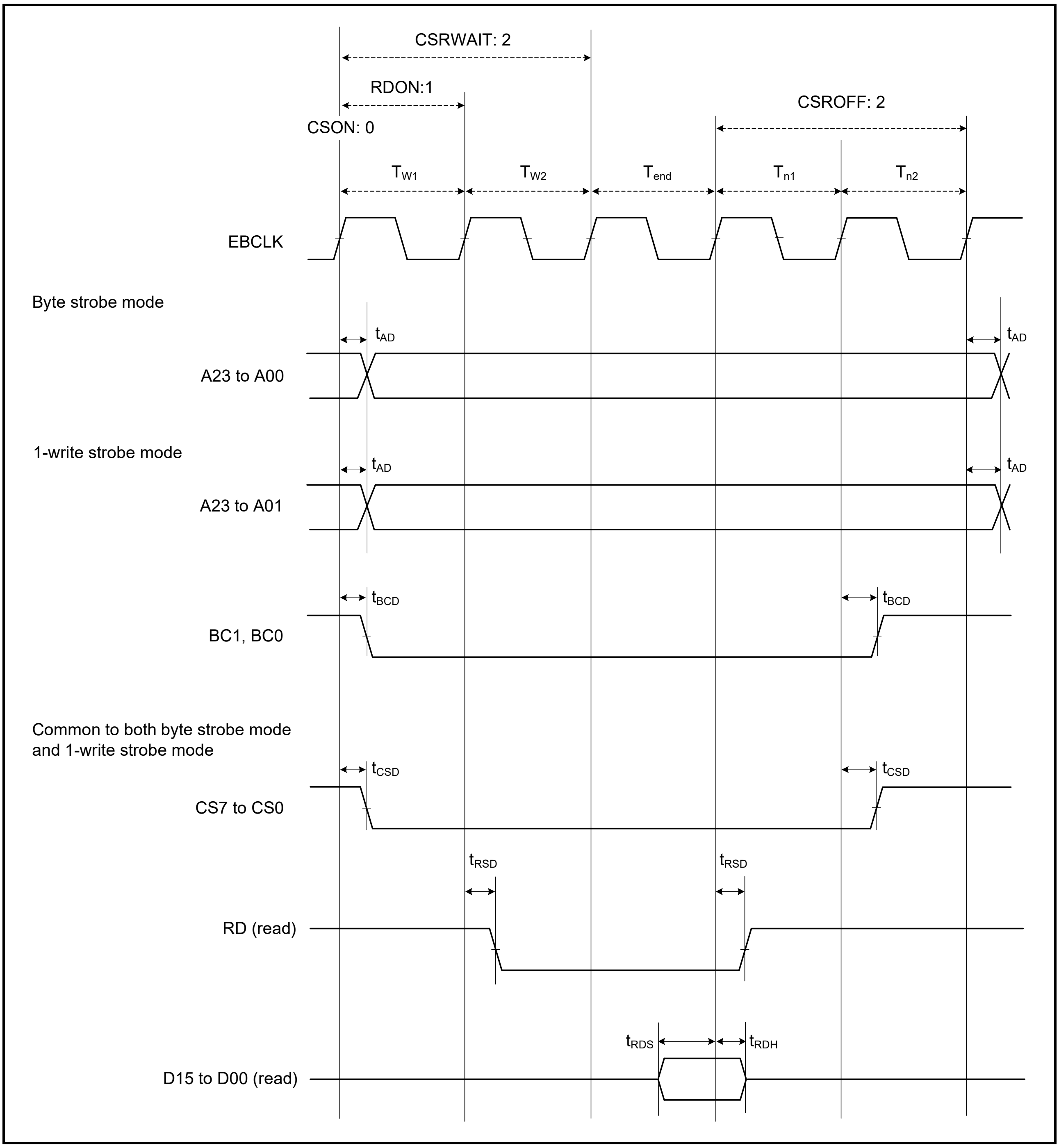
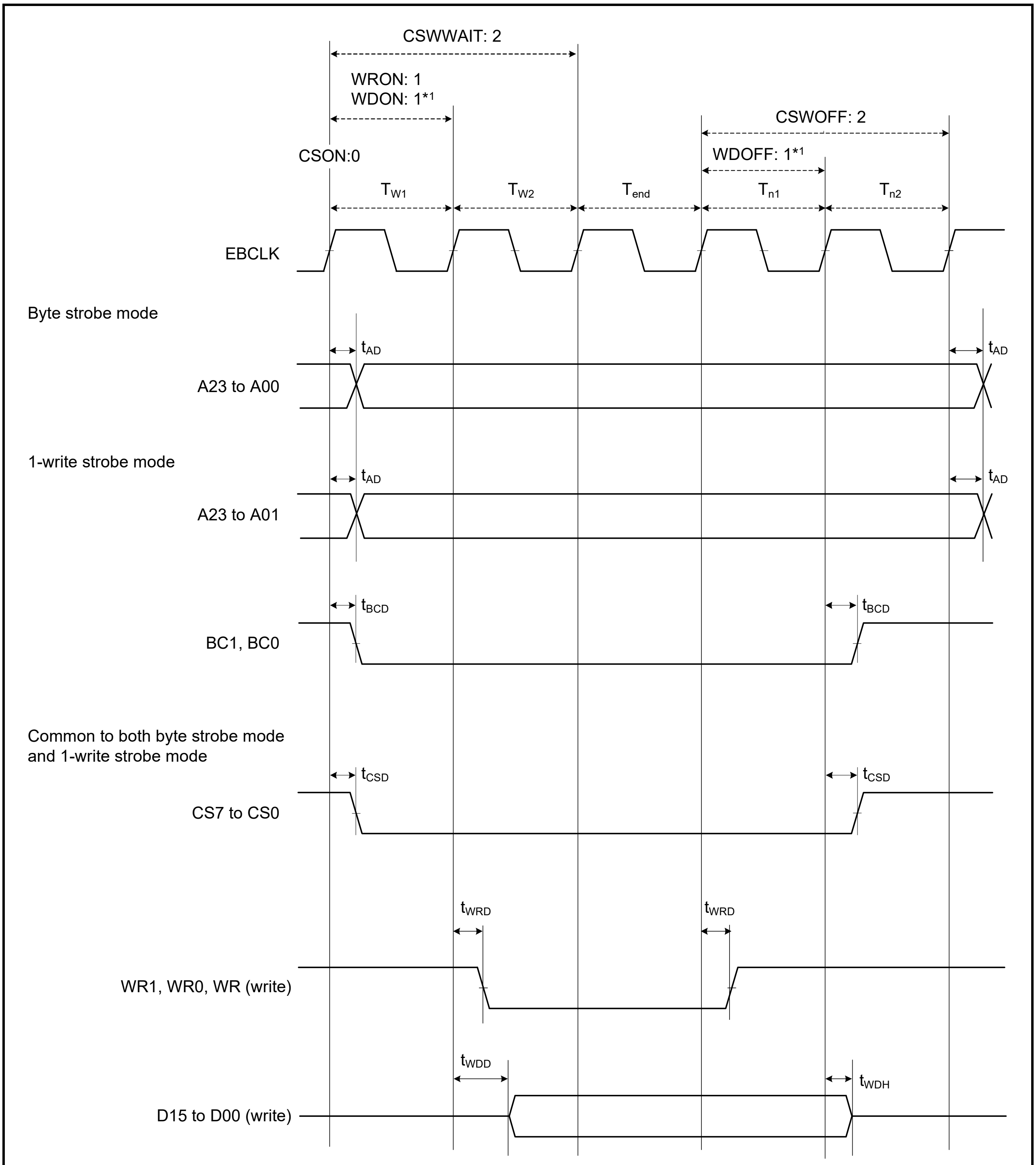


Figure 60.22 External bus timing for normal read cycle with bus clock synchronized



Note 1. Always specify WDON and WDOFF as at least one EBCLK cycle.

Figure 60.23 External bus timing for normal write cycle with bus clock synchronized

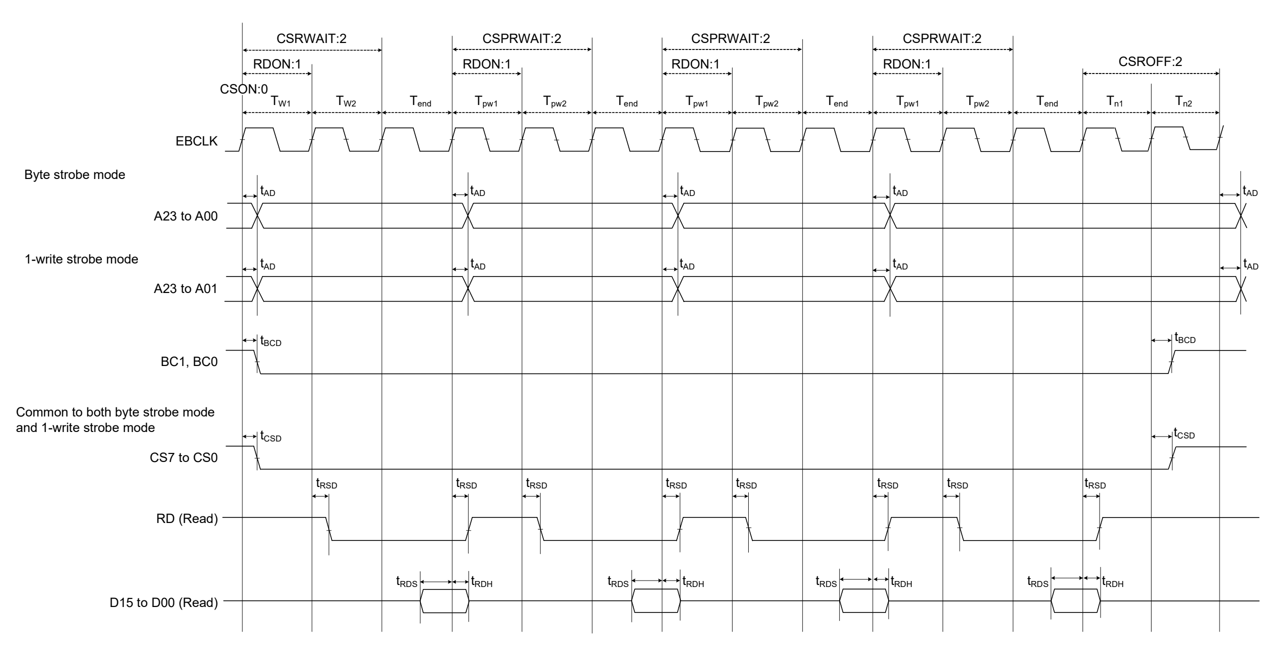
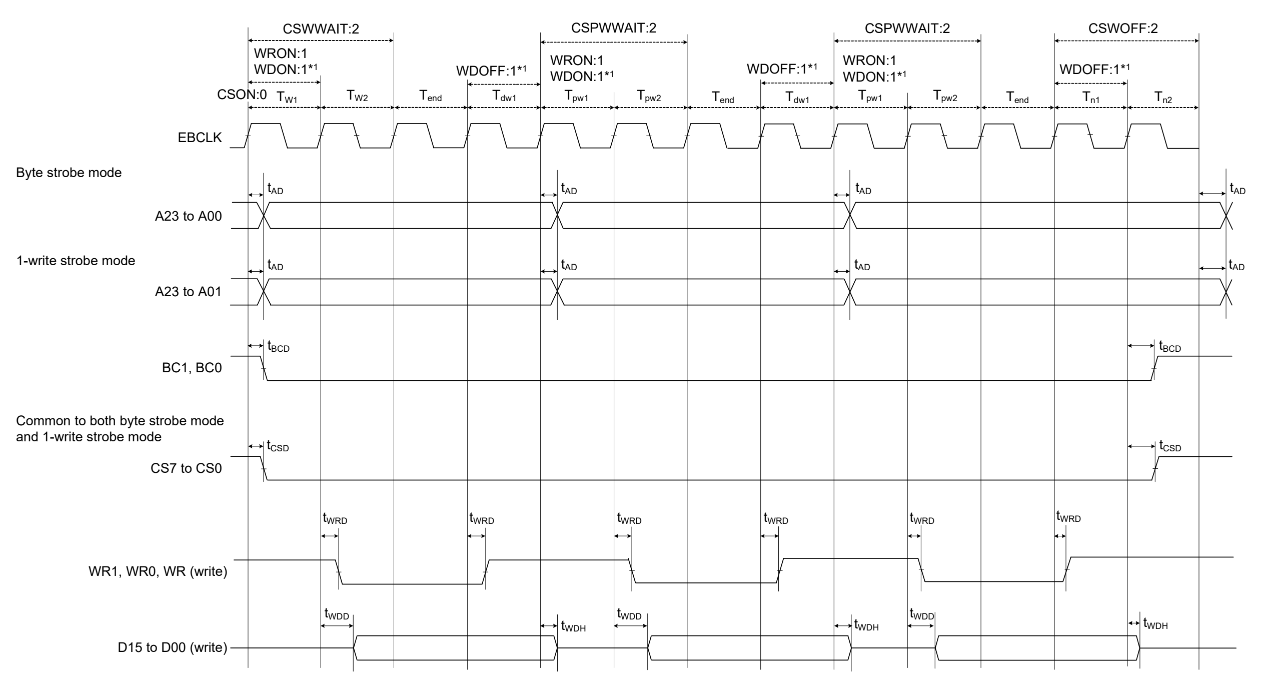


Figure 60.24 External bus timing for page read cycle with bus clock synchronized



Note 1. Always specify WDON and WDOFF as at least one EBCLK cycle.

Figure 60.25 External bus timing for page write cycle with bus clock synchronized

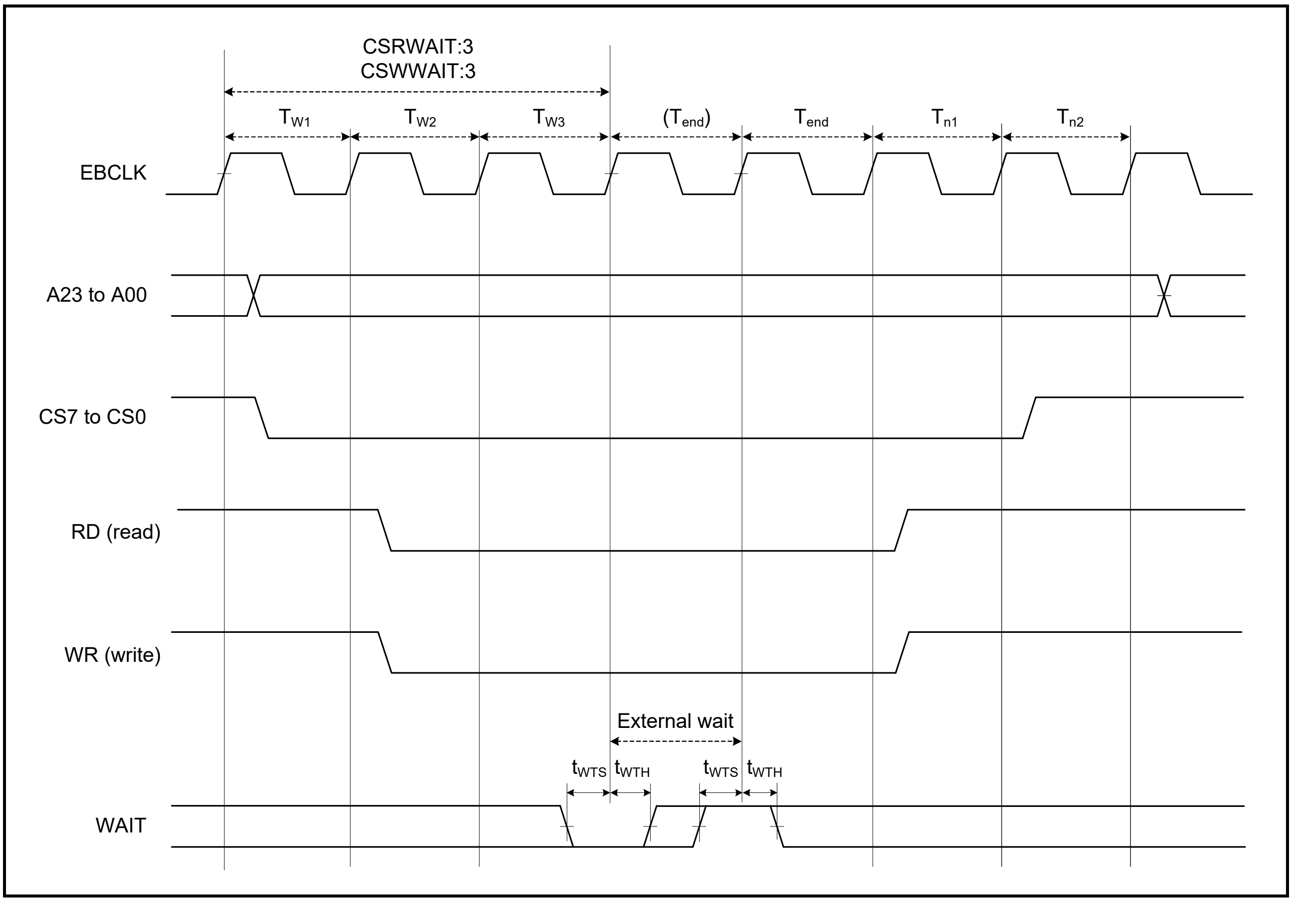
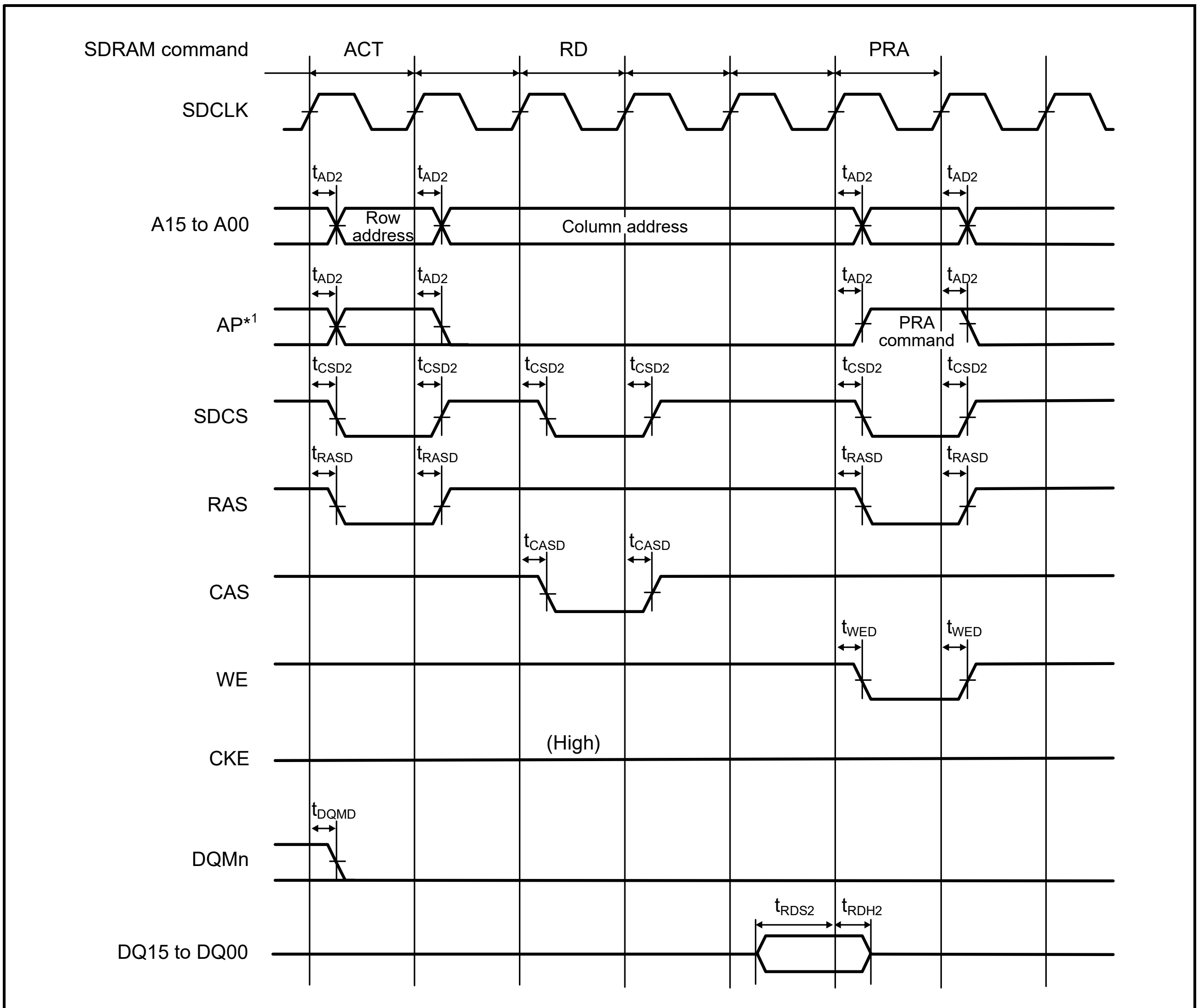
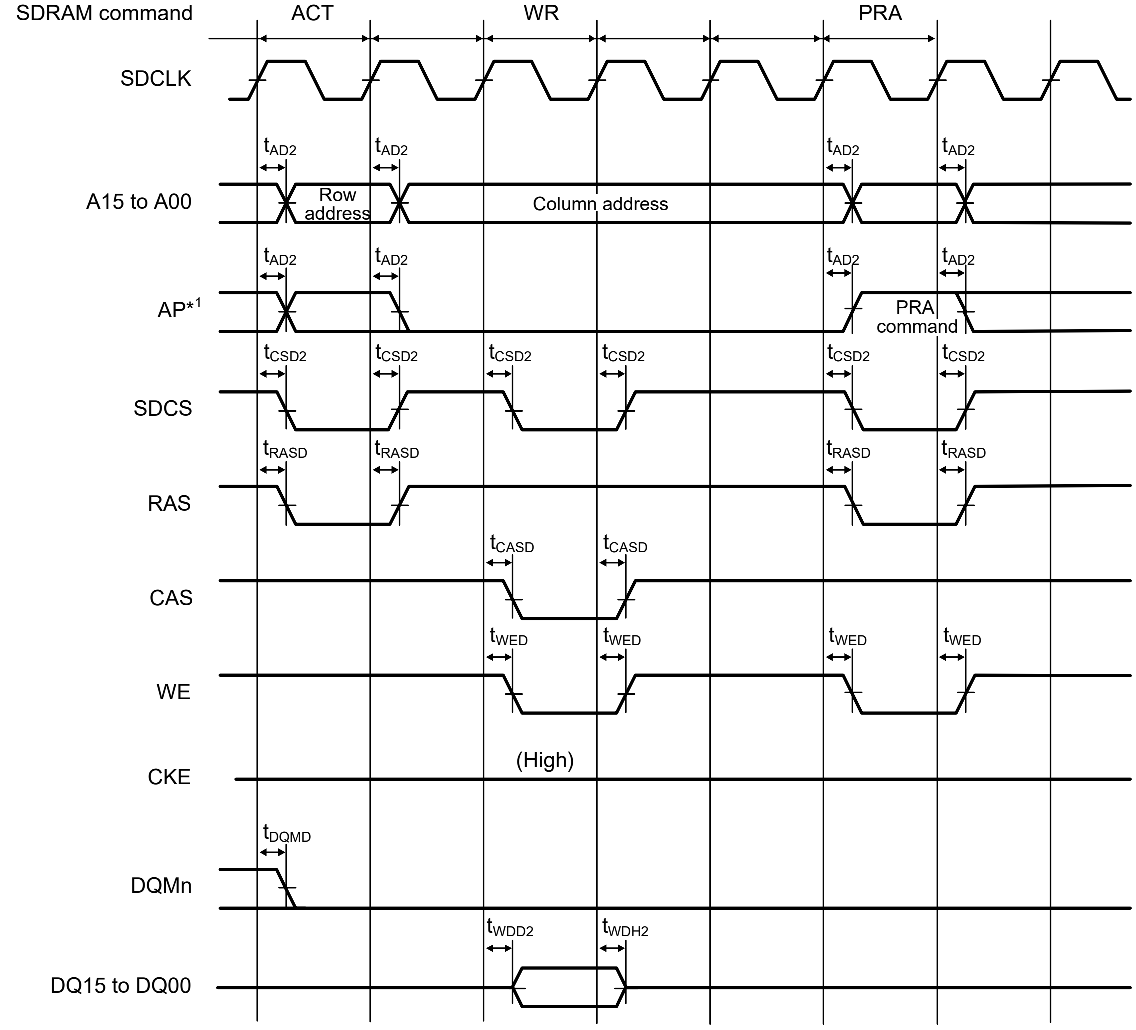


Figure 60.26 External bus timing for external wait control



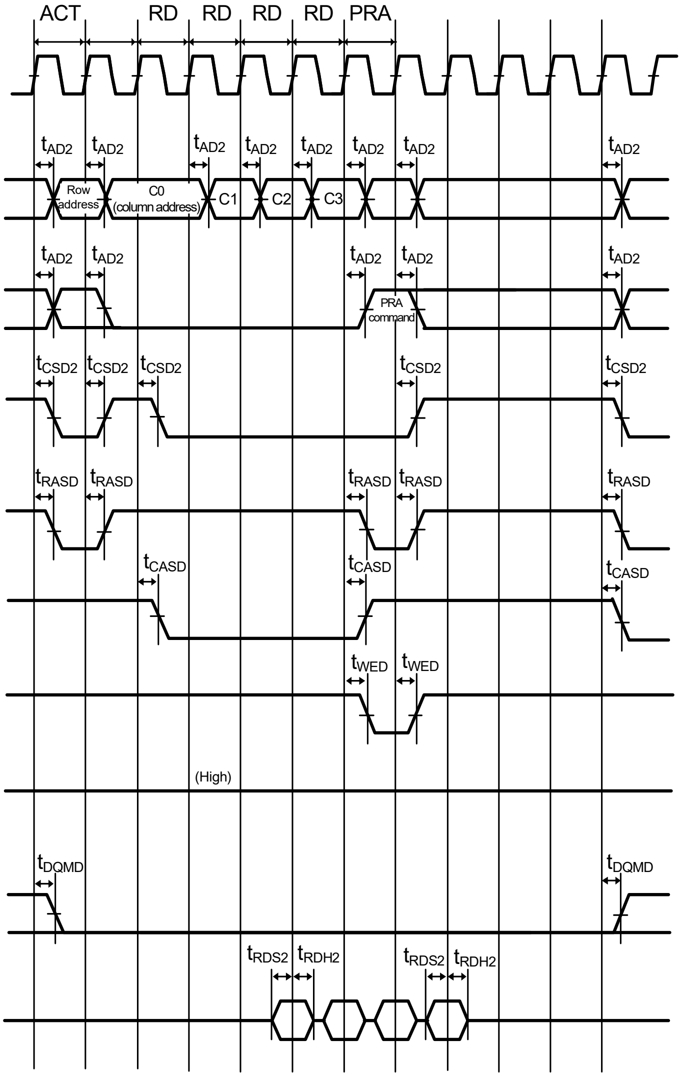
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.27 SDRAM single read timing



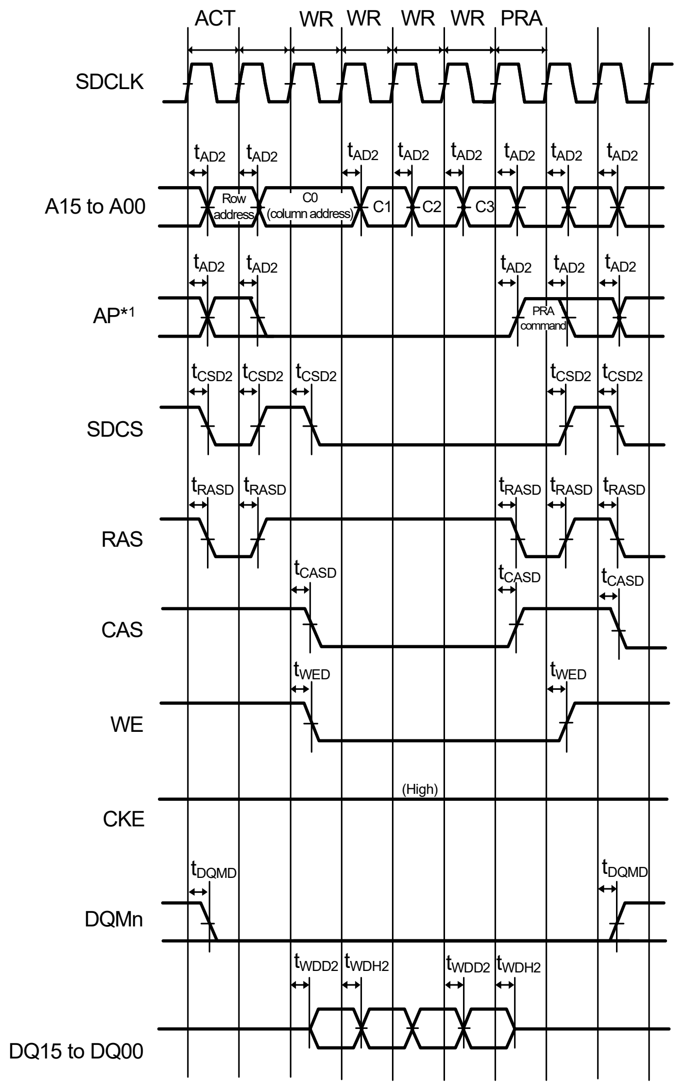
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.28 SDRAM single write timing



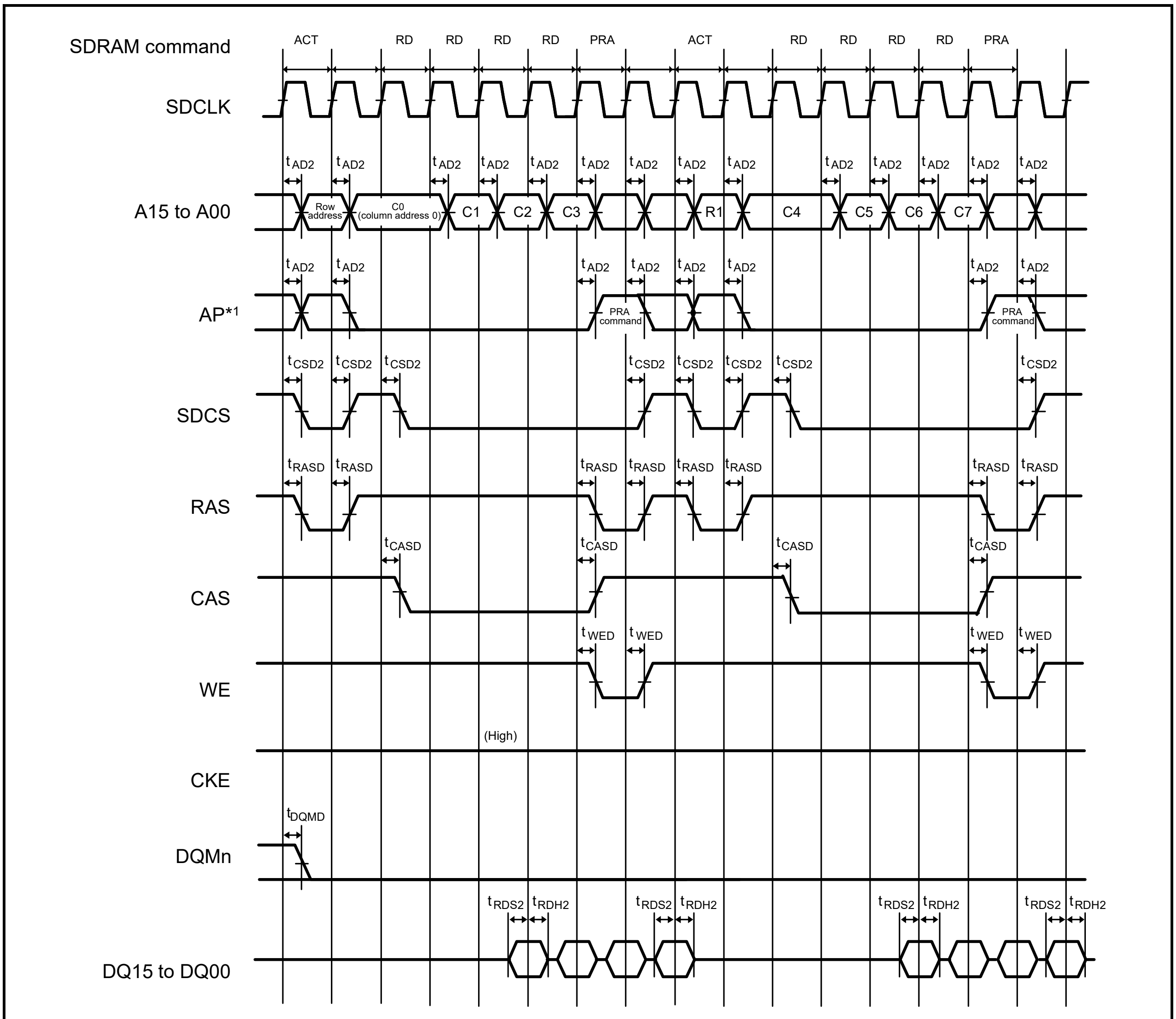
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.29 SDRAM multiple read timing



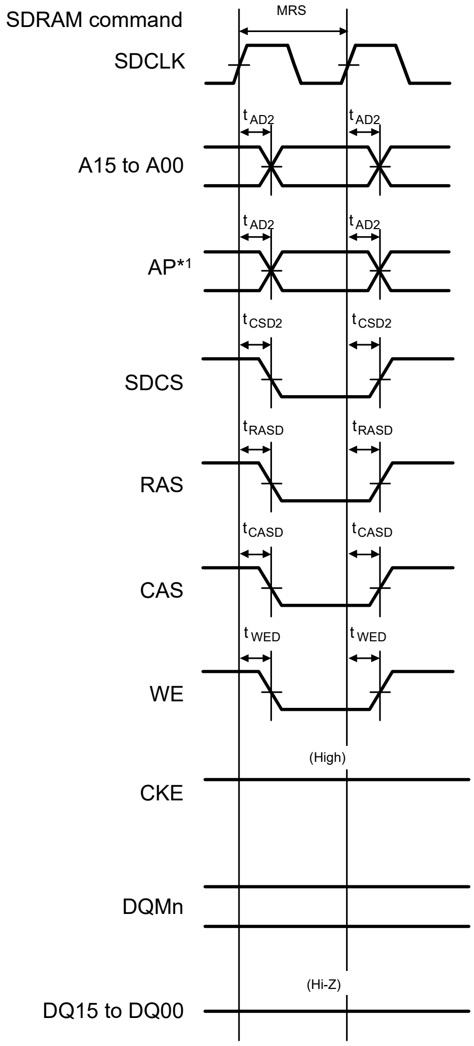
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.30 SDRAM multiple write timing



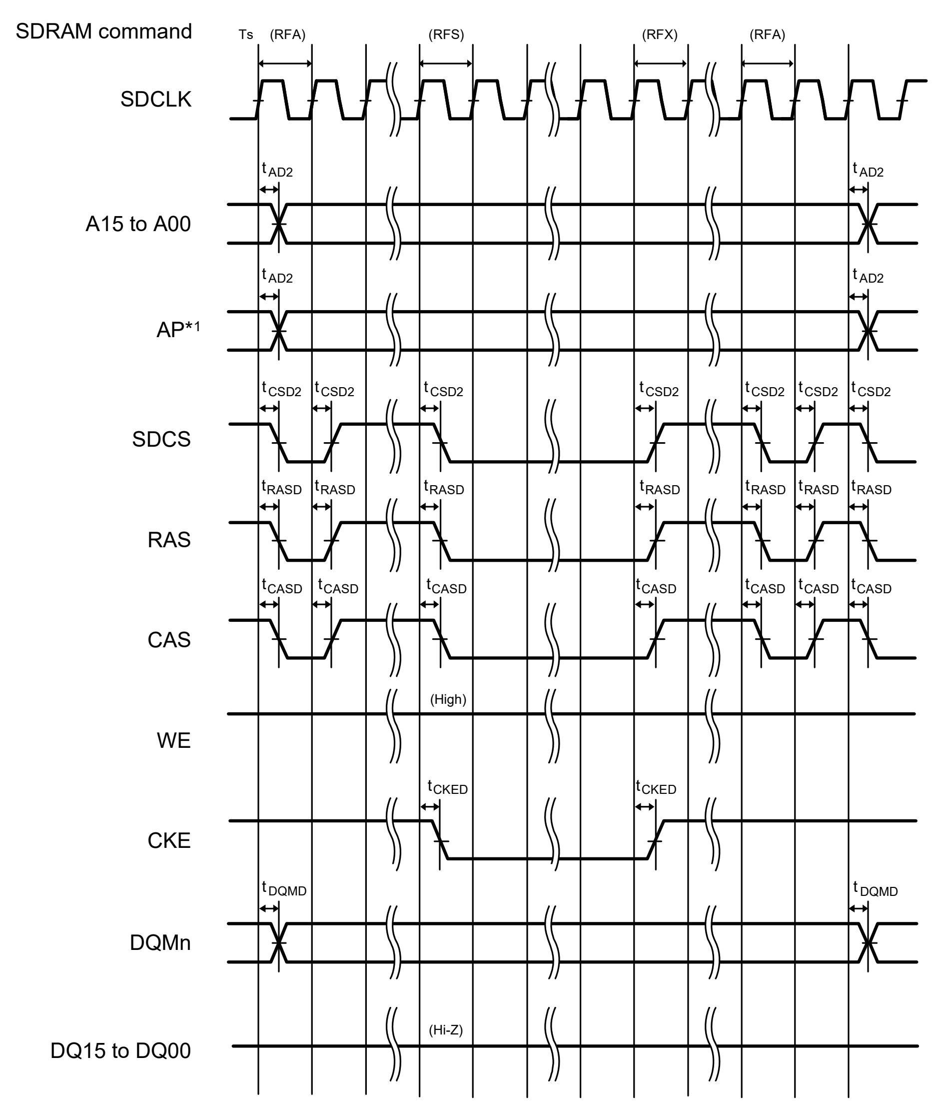
Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.31 SDRAM multiple read line stride timing



Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.32 SDRAM mode register set timing



Note 1. Address pins are for output of the precharge-select command (Precharge-sel) for the SDRAM.

Figure 60.33 SDRAM self-refresh timing

60.3.7 I/O Ports, POEG, GPT32, AGT, KINT, and ADC12 Trigger Timing

Table 60.19 I/O ports, POEG, GPT32, AGT, KINT, and ADC12 trigger timing (1 of 2)

GPT32 Conditions:
High drive output is selected in the port drive capability bit in the PmnPFS register.

AGT Conditions:
Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	-	t_{Pcyc}	Figure 60.34
POEG	POEG input trigger pulse width	t_{POEW}	3	-	t_{Pcyc}	Figure 60.35

Table 60.19 I/O ports, POEG, GPT32, AGT, KINT, and ADC12 trigger timing (2 of 2)

GPT32 Conditions:
High drive output is selected in the port drive capability bit in the PmnPFS register.

AGT Conditions:
Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit	Test conditions
GPT32	Input capture pulse width	Single edge	t_{GTICW}	1.5	-	t_{PDcyc}	Figure 60.36
		Dual edge		2.5	-		
	GTIOCxY output skew (x = 0 to 7, Y= A or B)	Middle drive buffer	t_{GTISK}*1	-	4	ns	Figure 60.37
		High drive buffer		-	4		
	GTIOCxY output skew (x = 8 to 13, Y = A or B)	Middle drive buffer		-	4		
		High drive buffer		-	4		
	GTIOCxY output skew (x = 0 to 13, Y = A or B)	Middle drive buffer		-	6		
		High drive buffer		-	6		
	OPS output skew GTOUUP, GTOULO, GTOVUP, GTOVLO, GTOWUP, GTOWLO		t_{GTOSK}	-	5	ns	Figure 60.38
GPT(PWM Delay Generation Circuit)	GTIOCxY_Z output skew (x = 0 to 3, Y = A or B, Z = A)		t_{HRSK}*2	-	2.0	ns	Figure 60.39
AGT	AGTIO, AGTEE input cycle		t_{ACYC}*3	100	-	ns	Figure 60.40
	AGTIO, AGTEE input high width, low width		t_{ACKWH}, t_{ACKWL}	40	-	ns	
	AGTIO, AGTO, AGTOA, AGTOB output cycle		t_{ACYC2}	62.5	-	ns	
ADC12	ADC12 trigger input pulse width		t_{TRGW}	1.5	-	t_{Pcyc}	Figure 60.41
KINT	KRn (n = 00 to 07) pulse width		t_{KR}	250	-	ns	Figure 60.42

Note: t_{Pcyc}: PCLKB cycle, t_{PDcyc}: PCLKD cycle.

Note 1. This skew applies when the same driver I/O is used. If the I/O of the middle and high drivers is mixed, operation is not guaranteed.

Note 2. The load is 30 pF.

Note 3. Constraints on input cycle:
When not switching the source clock: $t_{Pcyc} \times 2 < t_{ACYC}$ should be satisfied.
When switching the source clock: $t_{Pcyc} \times 6 < t_{ACYC}$ should be satisfied.

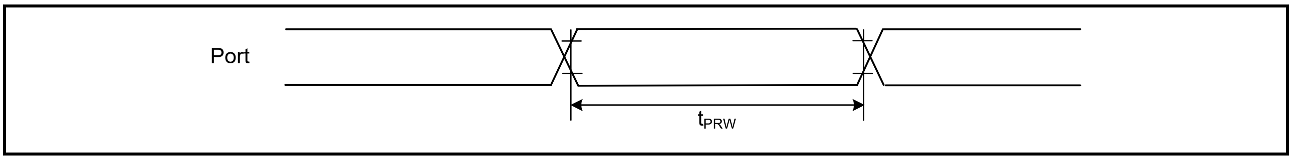


Figure 60.34 I/O ports input timing

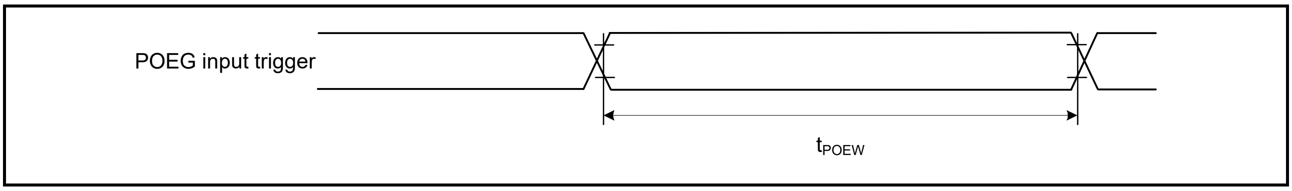


Figure 60.35 POEG input trigger timing

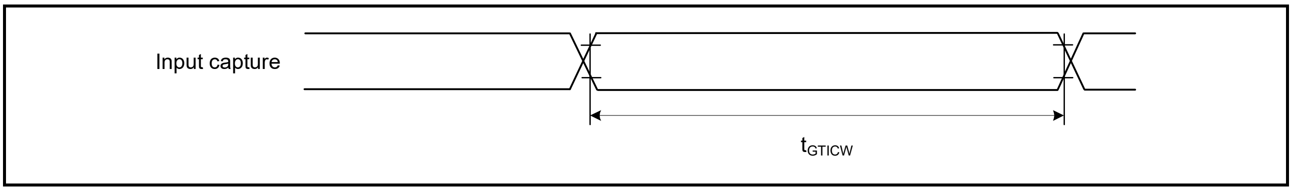


Figure 60.36 GPT32 input capture timing

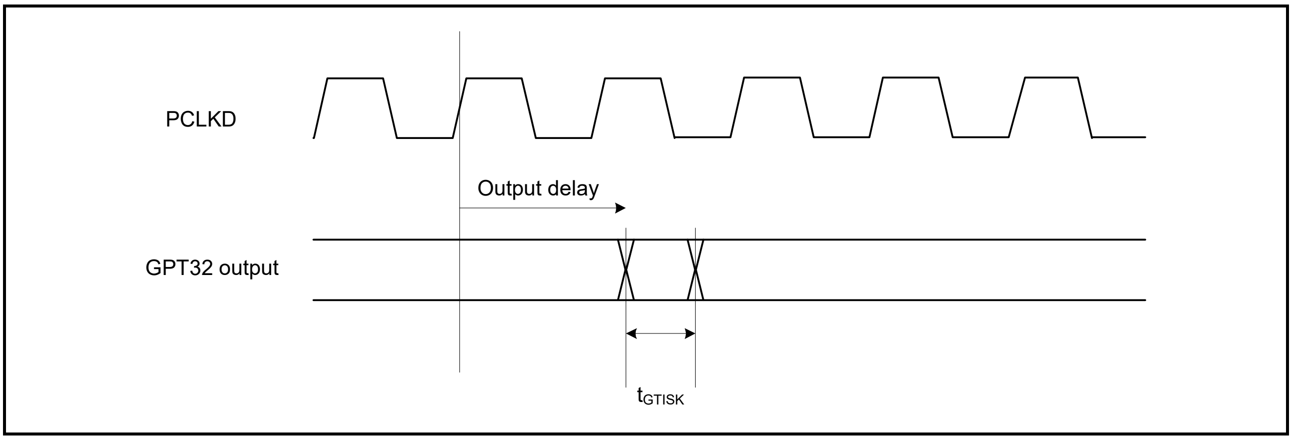


Figure 60.37 GPT32 output delay skew

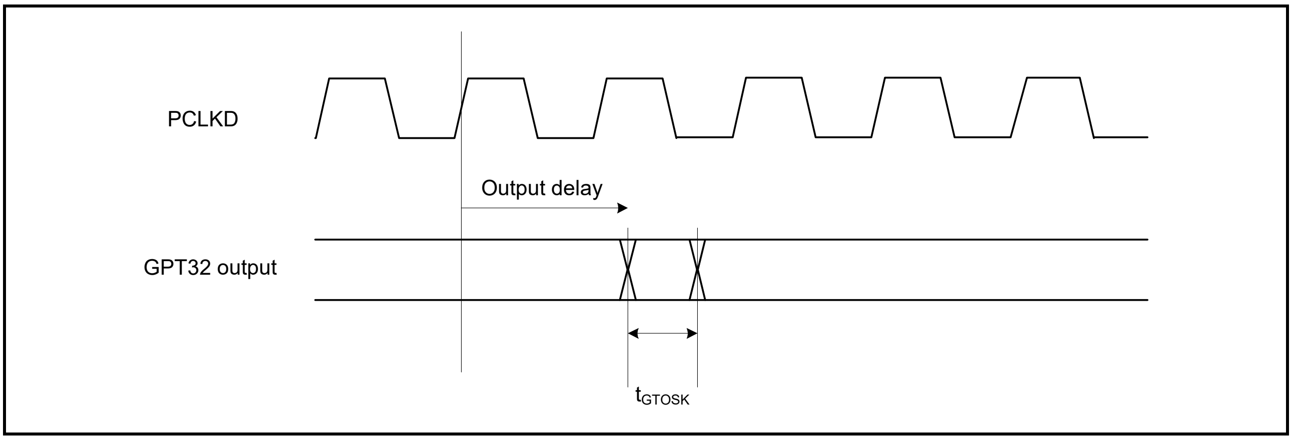


Figure 60.38 GPT32 output delay skew for OPS

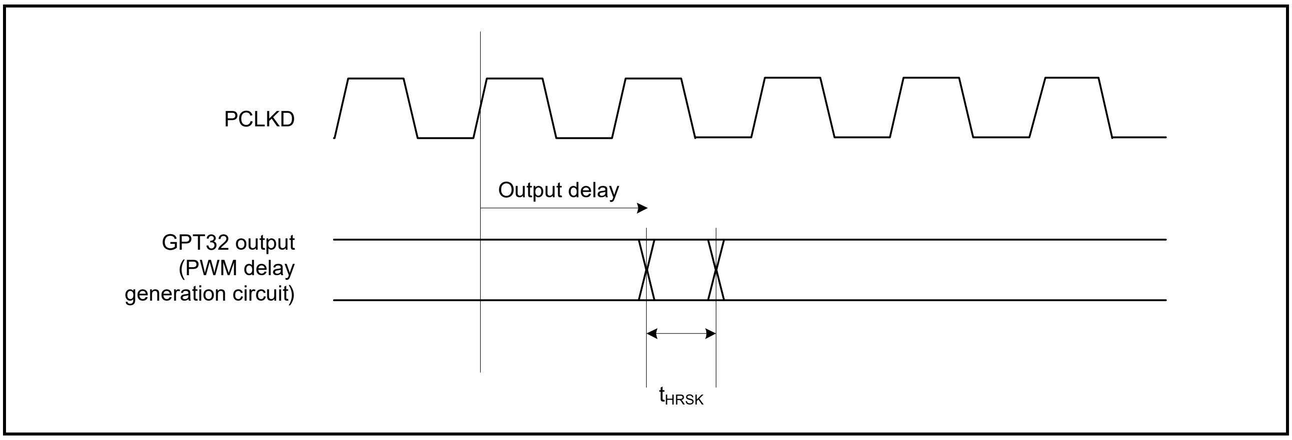


Figure 60.39 GPT32 (PWM Delay Generation Circuit) output delay skew

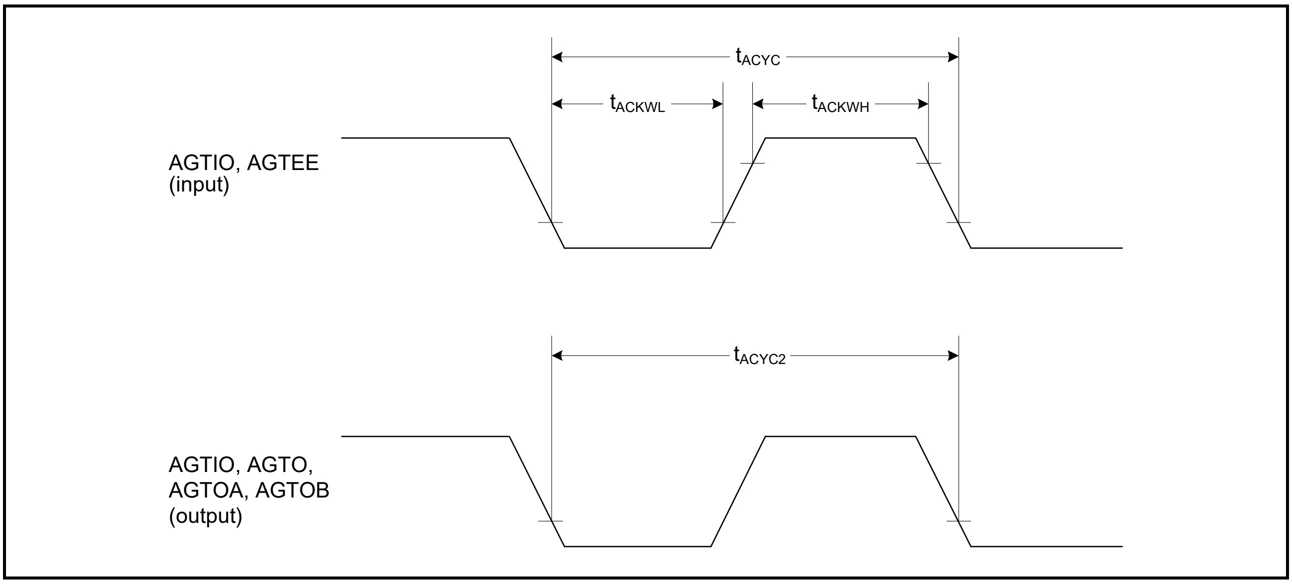


Figure 60.40 AGT input/output timing

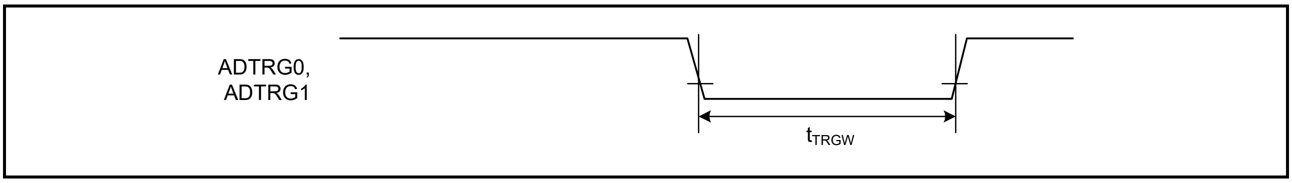


Figure 60.41 ADC12 trigger input timing

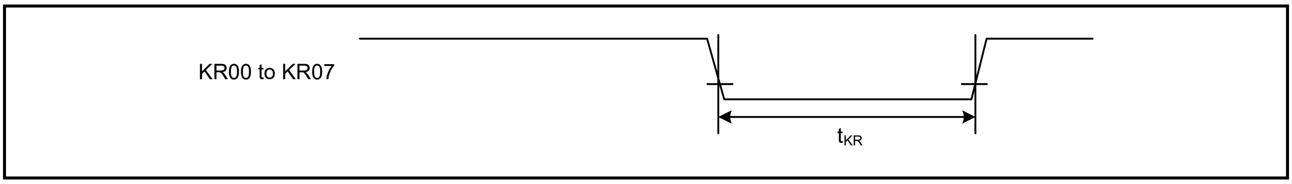


Figure 60.42 Key interrupt input timing

60.3.8 PWM Delay Generation Circuit Timing

Table 60.20 PWM Delay Generation Circuit timing

Parameter	Min	Typ	Max	Unit	Test conditions
Operation frequency	80	-	120	MHz	-
Resolution	-	260	-	ps	PCLKD = 120 MHz
DNL*1	-	±2.0	-	LSB	-

Note 1. This value normalizes the differences between lines in 1-LSB resolution.

60.3.9 CAC Timing

Table 60.21 CAC timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	$t_{PBcyc} \le tcac^{*2}$	t_{CACREF}	$4.5 \times t_{cac} + 3 \times t_{PBcyc}$	-	-	ns	-
		$t_{PBcyc} > tcac^{*2}$		$5 \times t_{cac} + 6.5 \times t_{PBcyc}$	-	-	ns	

Note 1. t_{PBcyc}: PCLKB cycle.

Note 2. t_{cac}: CAC count clock source cycle.

60.3.10 SCI Timing

Table 60.22 SCI timing (1)

Conditions: High drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: SCK0 to SCK9. For other pins, middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit*1	Test conditions
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	-	t_{Pcyc}	Figure 60.43
		Clock synchronous		6	-		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	-	5	ns	
	Input clock fall time		t_{SCKf}	-	5	ns	
	Output clock cycle	Asynchronous	t_{Scyc}	6	-	t_{Pcyc}	
		Clock synchronous		4	-		
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		t_{SCKr}	-	5	ns	
	Output clock fall time		t_{SCKf}	-	5	ns	
	Transmit data delay	Clock synchronous	t_{TXD}	-	25	ns	Figure 60.44
	Receive data setup time	Clock synchronous	t_{RXS}	15	-	ns	
	Receive data hold time	Clock synchronous	t_{RXH}	5	-	ns	

Note 1. t_{Pcyc}: PCLKA cycle.

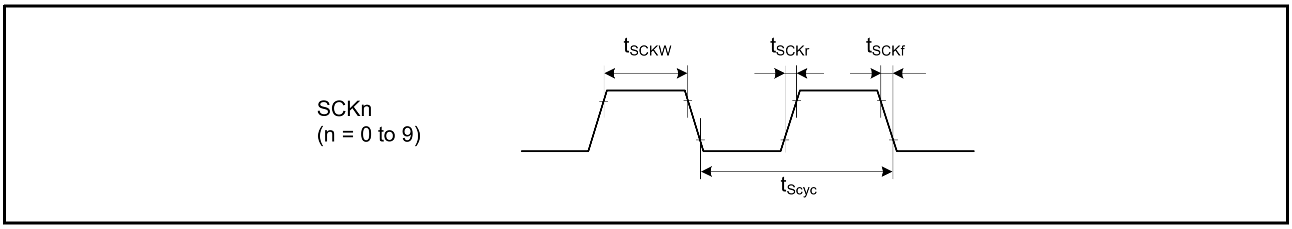


Figure 60.43 SCK clock input/output timing

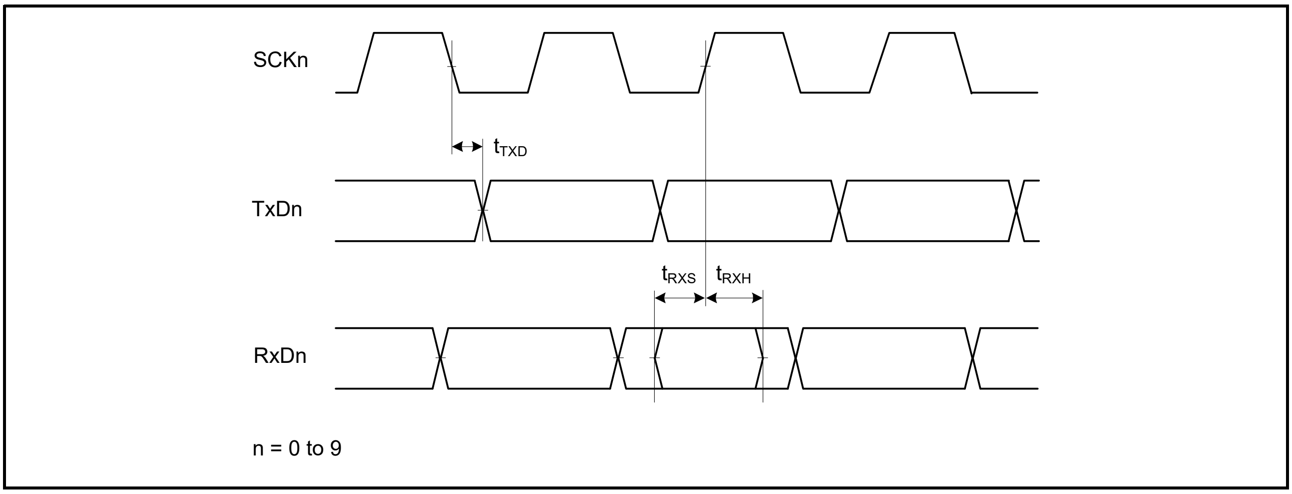


Figure 60.44 SCI input/output timing in clock synchronous mode

Table 60.23 SCI timing (2)

Conditions: High drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: SCK0 to SCK9.
For other pins, middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
Simple SPI	SCK clock cycle output (master)	t_{SPcyc}	4 (PCLKA ≤ 60 MHz) 8 (PCLKA > 60 MHz)	65536	t_{Pcyc}	Figure 60.45
	SCK clock cycle input (slave)	-	6 (PCLKA ≤ 60 MHz) 12 (PCLKA > 60 MHz)	65536		
	SCK clock high pulse width	t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width	t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise and fall time	t_{SPCKr}, t_{SPCKf}	-	20	ns	
	Data input setup time	t_{SU}	33.3	-	ns	Figure 60.46 to Figure 60.49
	Data input hold time	t_H	33.3	-	ns	
	SS input setup time	t_{LEAD}	1	-	t_{SPcyc}	
	SS input hold time	t_{LAG}	1	-	t_{SPcyc}	
	Data output delay	t_{OD}	-	33.3	ns	
	Data output hold time	t_{OH}	-10	-	ns	
	Data rise and fall time	t_{Dr}, t_{Df}	-	16.6	ns	
	SS input rise and fall time	t_{SSLr}, t_{SSLf}	-	16.6	ns	
	Slave access time	t_{SA}	-	4 (PCLKA ≤ 60 MHz) 8 (PCLKA > 60 MHz)	t_{Pcyc}	Figure 60.49
	Slave output release time	t_{REL}	-	5 (PCLKA ≤ 60 MHz) 10 (PCLKA > 60 MHz)	t_{Pcyc}	

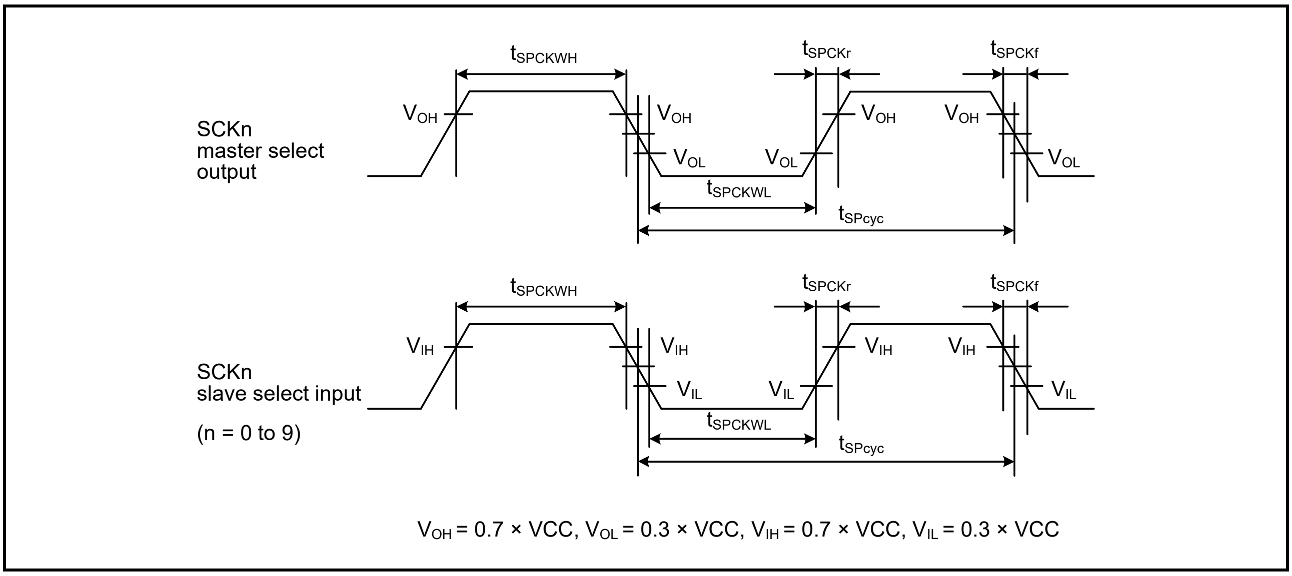


Figure 60.45 SCI simple SPI mode clock timing

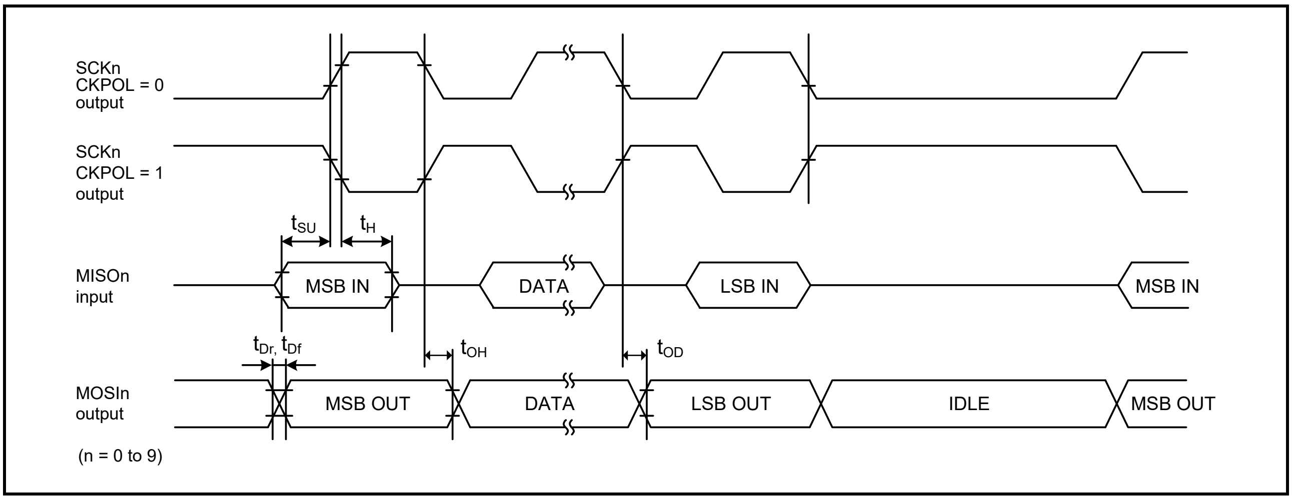


Figure 60.46 SCI simple SPI mode timing for master when CKPH = 1

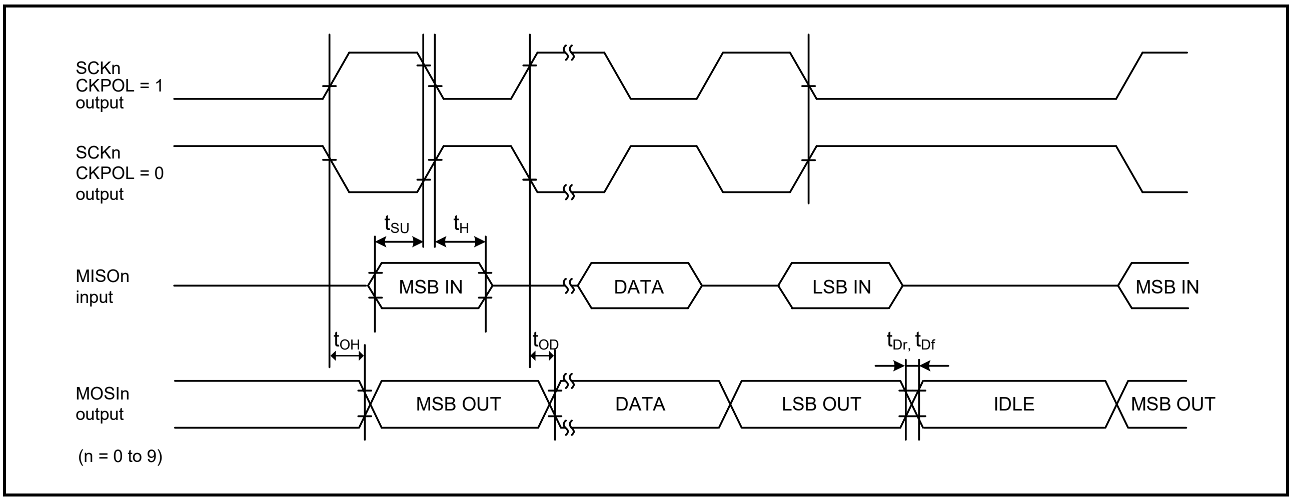


Figure 60.47 SCI simple SPI mode timing for master when CKPH = 0

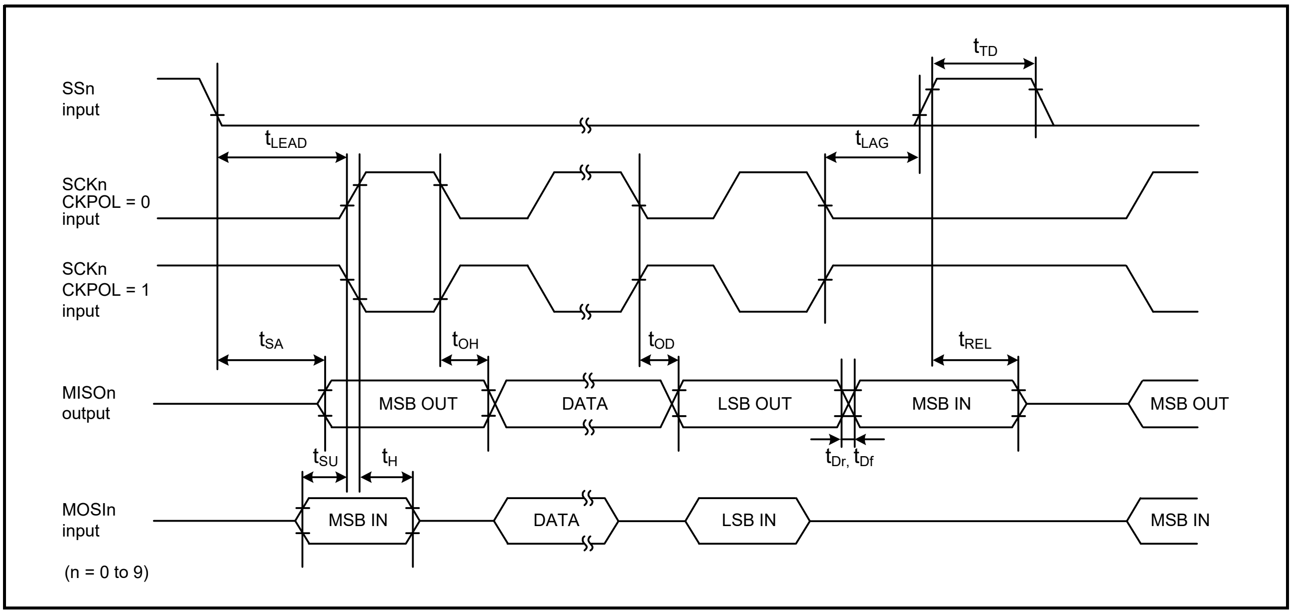


Figure 60.48 SCI simple SPI mode timing for slave when CKPH = 1

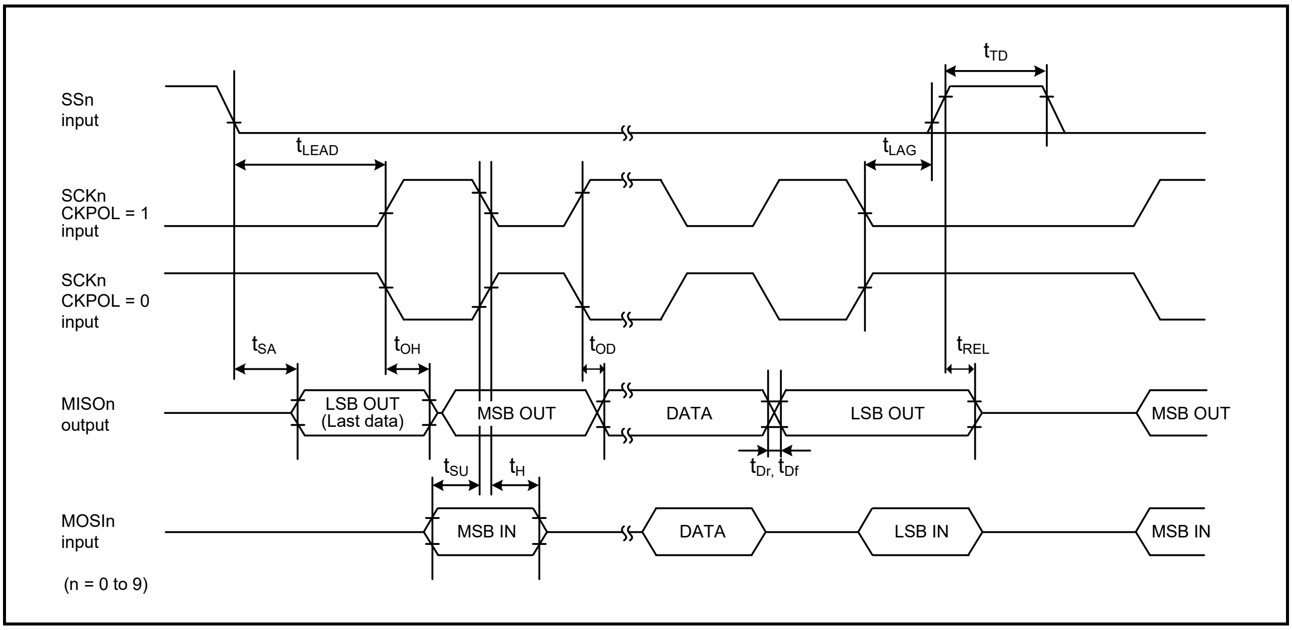


Figure 60.49 SCI simple SPI mode timing for slave when CKPH = 0

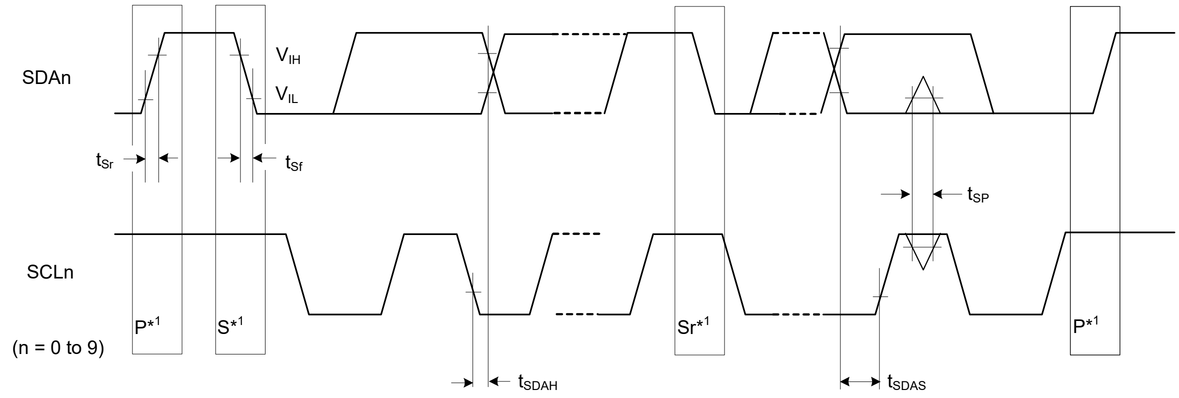
Table 60.24 SCI timing (3) (1 of 2)
Conditions: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
Simple IIC (Standard mode)	SDA input rise time	t_{Sr}	-	1000	ns	Figure 60.50
	SDA input fall time	t_{Sf}	-	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc}	ns	
	Data input setup time	t_{SDAS}	250	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b*1	-	400	pF	

Table 60.24 SCI timing (3) (2 of 2)
Conditions: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions
Simple IIC (Fast mode)	SDA input rise time	t_{Sr}	-	300	ns	Figure 60.50
	SDA input fall time	t_{Sf}	-	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc}	ns	
	Data input setup time	t_{SDAS}	100	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_{b}*[1]	-	400	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle.
Note 1. Cb indicates the total capacity of the bus line.



Note 1. S, P, and Sr indicate the following:
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions:
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA

Figure 60.50 SCI simple IIC mode timing

60.3.11 SPI Timing

Table 60.25 SPI timing

Conditions:
For RSPCKA and RSPCKB pins, high drive output is selected with the port drive capability bit in the PmnPFS register.
For other pins, middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter			Symbol	Min	Max	Unit*1	Test conditions*2
SPI	RSPCK clock cycle	Master	t_{SPcyc}	2 (PCLKA ≤ 60 MHz) 4 (PCLKA > 60 MHz)	4096	t_{Pcyc}	Figure 60.51 C = 30 pF
		Slave		4	4096		
	RSPCK clock high pulse width	Master	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	-	ns	
		Slave		$2 \times t_{Pcyc}$	-		
	RSPCK clock low pulse width	Master	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	-	ns	
		Slave		$2 \times t_{Pcyc}$	-		
	RSPCK clock rise and fall time	Master	t_{SPCKr}, t_{SPCKf}	-	5	ns	
		Slave		-	1	μs	
	Data input setup time	Master	t_{SU}	4	-	ns	Figure 60.52 to Figure 60.57 C = 30 pF
		Slave		5	-		
	Data input hold time	Master (PCLKA division ratio set to 1/2)	t_{HF}	0	-	ns	
		Master (PCLKA division ratio set to a value other than 1/2)	t_{H}	t_{Pcyc}	-		
		Slave	t_{H}	20	-		
	SSL setup time	Master	t_{LEAD}	N × t_{SPcyc} - 10*3	N × t_{SPcyc} + 100*3	ns	
		Slave		6 x t_{Pcyc}	-	ns	
	SSL hold time	Master	t_{LAG}	N × t_{SPcyc} - 10 *4	N × t_{SPcyc} + 100*4	ns	
		Slave		6 x t_{Pcyc}	-	ns	
	Data output delay	Master	t_{OD}	-	6.3	ns	
		Slave		-	20		
	Data output hold time	Master	t_{OH}	0	-	ns	
		Slave		0	-		
	Successive transmission delay	Master	t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns	
		Slave		$6 \times t_{Pcyc}$			
	MOSI and MISO rise and fall time	Output	t_{Dr}, t_{Df}	-	5	ns	
		Input		-	1	μs	
	SSL rise and fall time	Output	t_{SSLr}, t_{SSLf}	-	5	ns	
		Input		-	1	μs	
	Slave access time		t_{SA}	-	2 x t_{Pcyc} + 28	ns	Figure 60.56 and Figure 60.57 C = 30pF
	Slave output release time		t_{REL}	-	2 x t_{Pcyc} + 28		

Note 1. t_{Pcyc}: PCLKA cycle.

Note 2. Must use pins that have a letter ("_A", "_B") to indicate group membership appended to their name as groups. For the SPI interface, the AC portion of the electrical characteristics is measured for each group.
Note 3. N is set to an integer from 1 to 8 by the SPCKD register.
Note 4. N is set to an integer from 1 to 8 by the SSLND register.

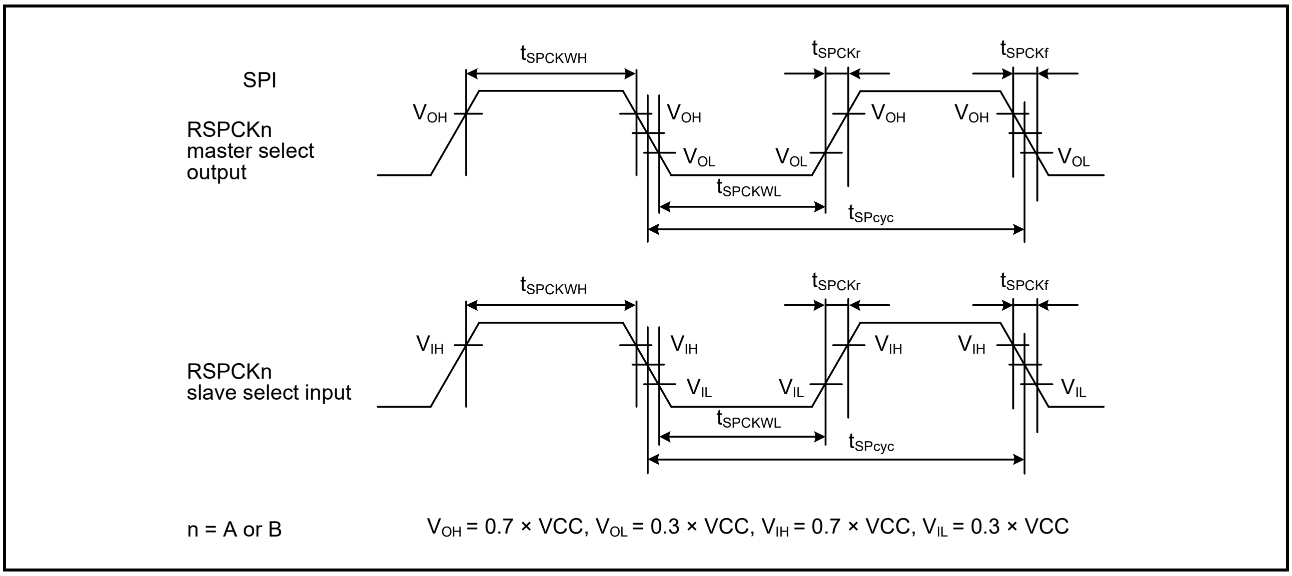


Figure 60.51 SPI clock timing

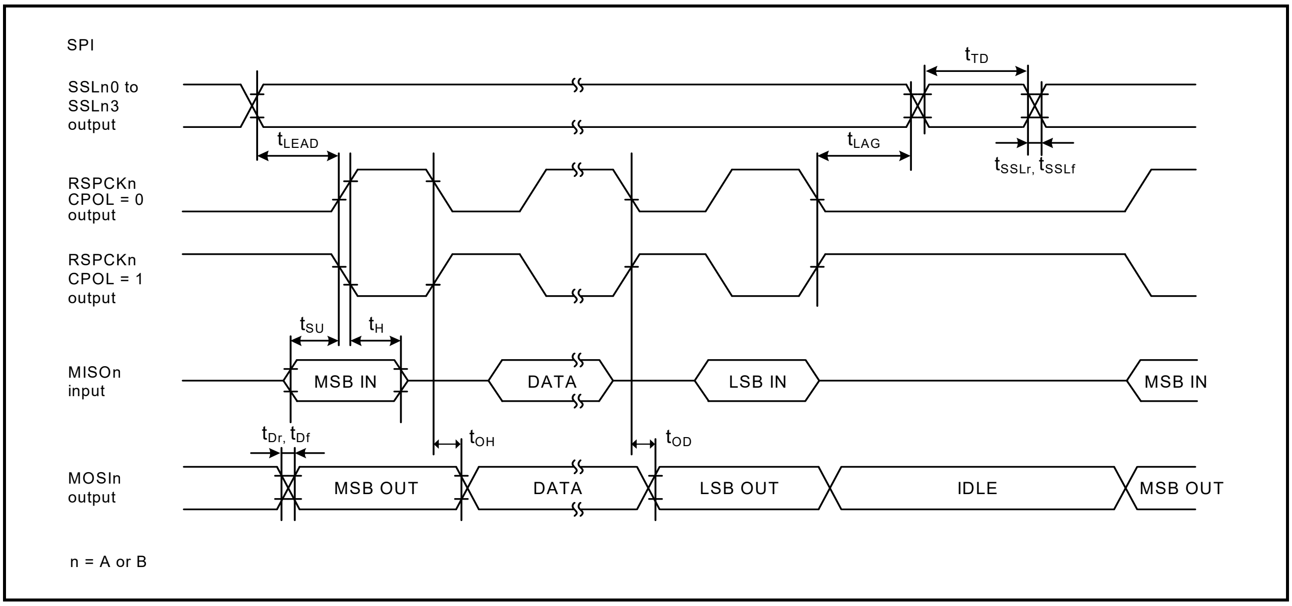


Figure 60.52 SPI timing for master when CPHA = 0

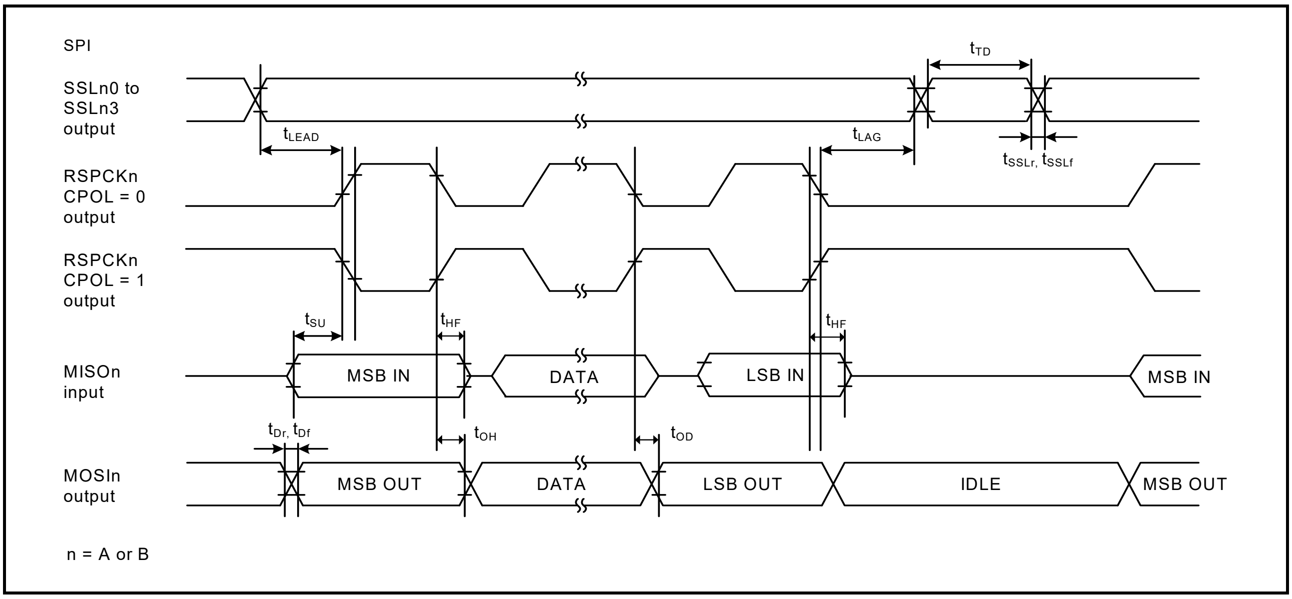


Figure 60.53 SPI timing for master when CPHA = 0 and the bit rate is set to PCLKA/2

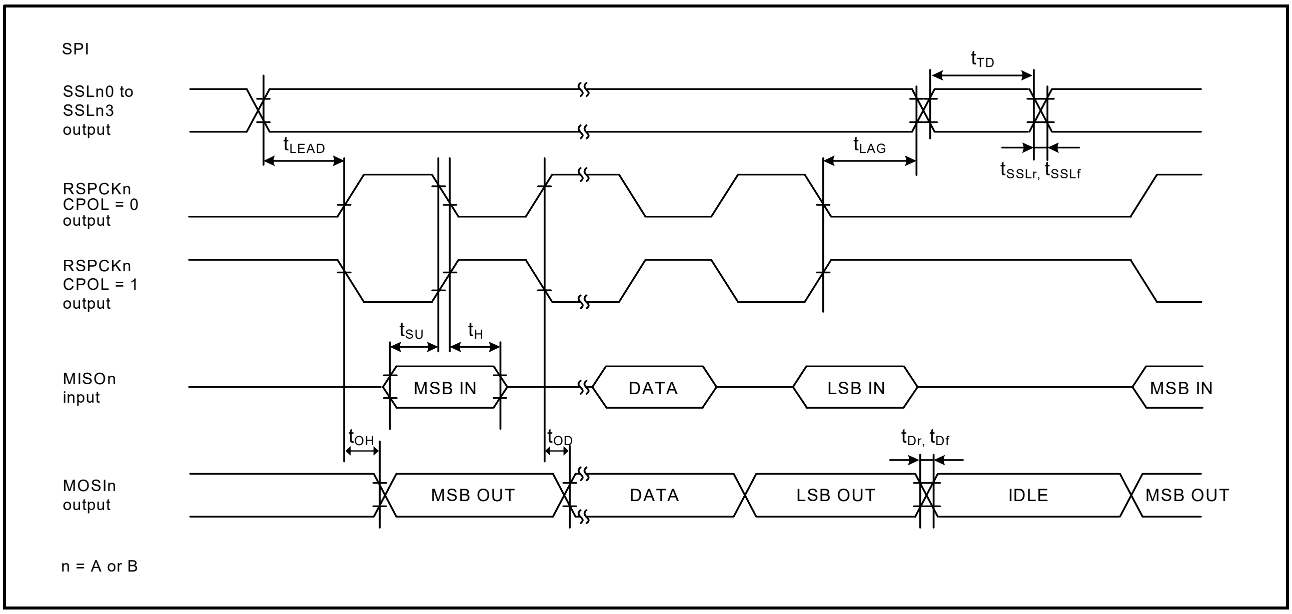


Figure 60.54 SPI timing for master when CPHA = 1

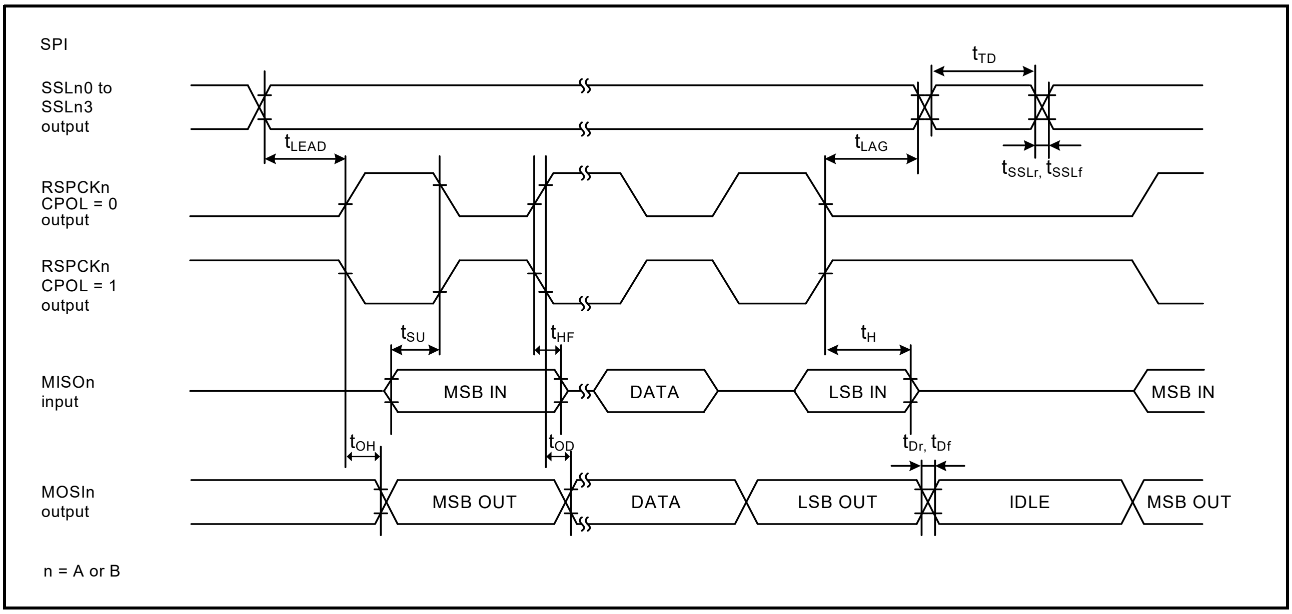


Figure 60.55 RSPI timing for master when CPHA = 1 and the bit rate is set to PCLKA/2

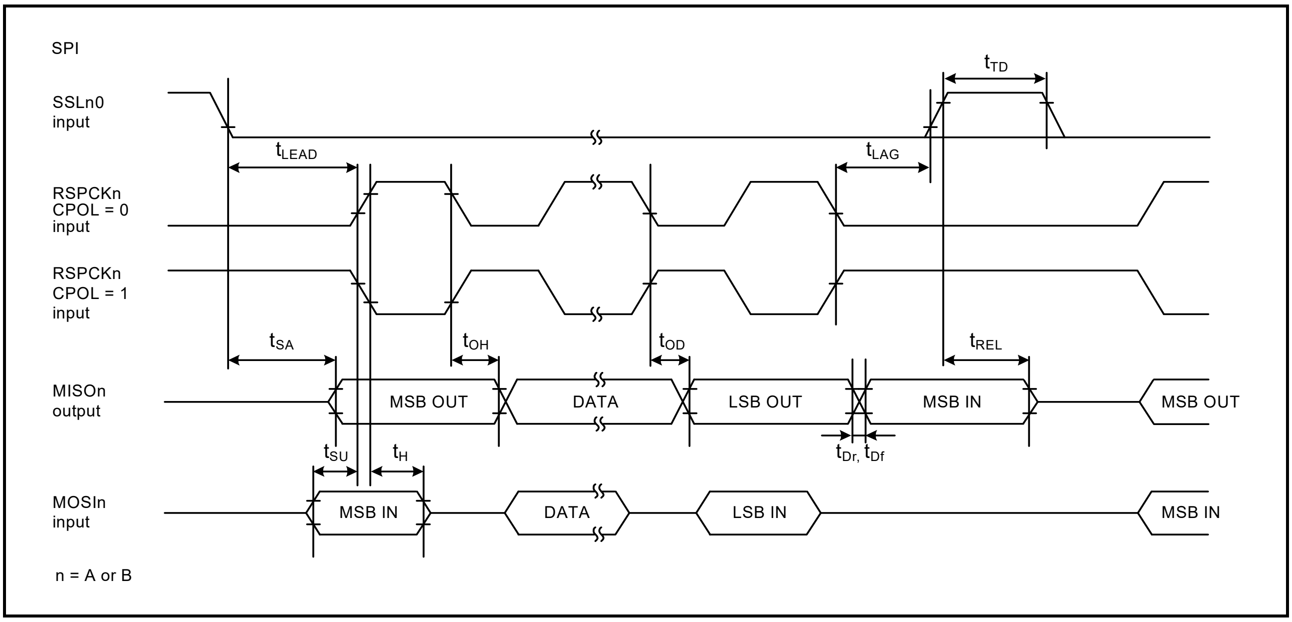


Figure 60.56 SPI timing for slave when CPHA = 0

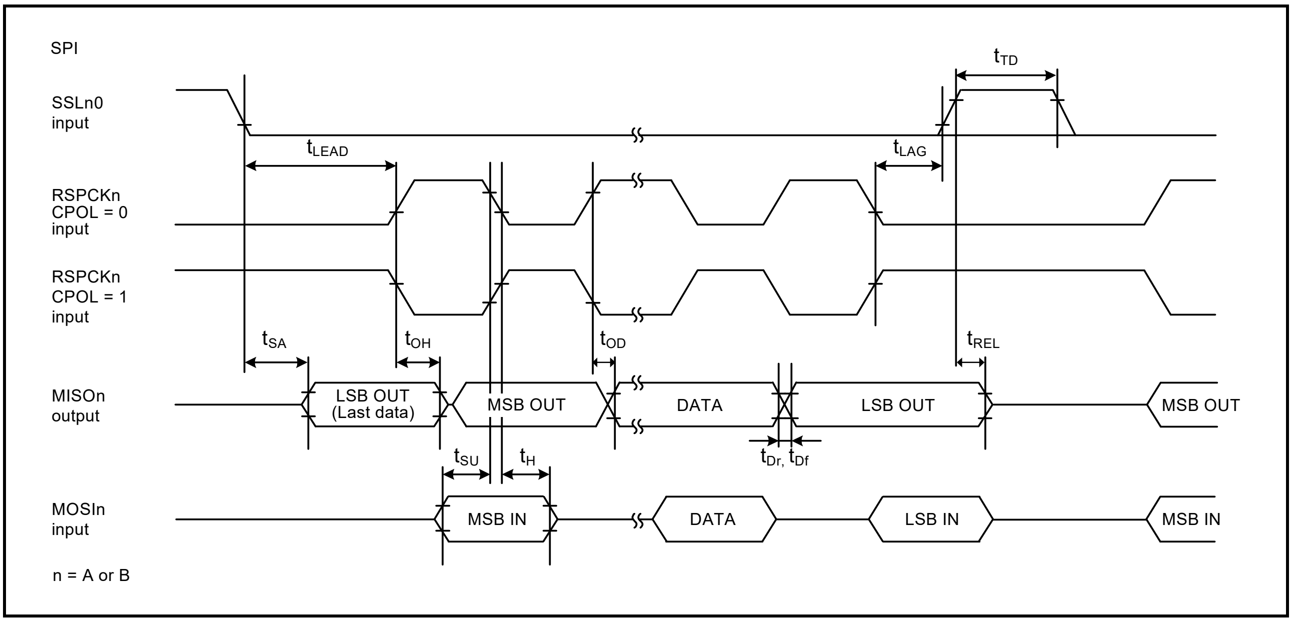


Figure 60.57 SPI timing for slave when CPHA = 1

60.3.12 QSPI Timing

Table 60.26 QSPI timing

Conditions: High drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit*1	Test conditions
QSPI	QSPCK clock cycle	t_{QScyc}	2	48	t_{Pcyc}	Figure 60.58
	QSPCK clock high pulse width	t_{QSWH}	$t_{QScyc} \times 0.4$	-	ns	
	QSPCK clock low pulse width	t_{QSWL}	$t_{QScyc} \times 0.4$	-	ns	
	Data input setup time	t_{Su}	8	-	ns	Figure 60.59
	Data input hold time	t_{IH}	0	-	ns	
	QSSL setup time	t_{LEAD}	(N+0.5) x t_{Qscyc} - 5 *2	(N+0.5) x t_{Qscyc} +100 *2	ns	
	QSSL hold time	t_{LAG}	(N+0.5) x t_{Qscyc} - 5 *3	(N+0.5) x t_{Qscyc} +100 *3	ns	
	Data output delay	t_{OD}	-	4	ns	
	Data output hold time	t_{OH}	-3.3	-	ns	
	Successive transmission delay	t_{TD}	1	16	t_{QScyc}	

Note 1. t_{Pcyc}: PCLKA cycle.

Note 2. N is set to 0 or 1 in SFMSLD.

Note 3. N is set to 0 or 1 in SFMSHD.

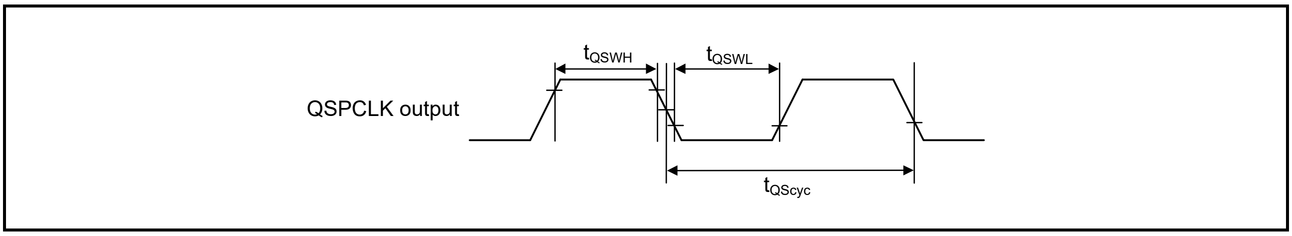


Figure 60.58 QSPI clock timing

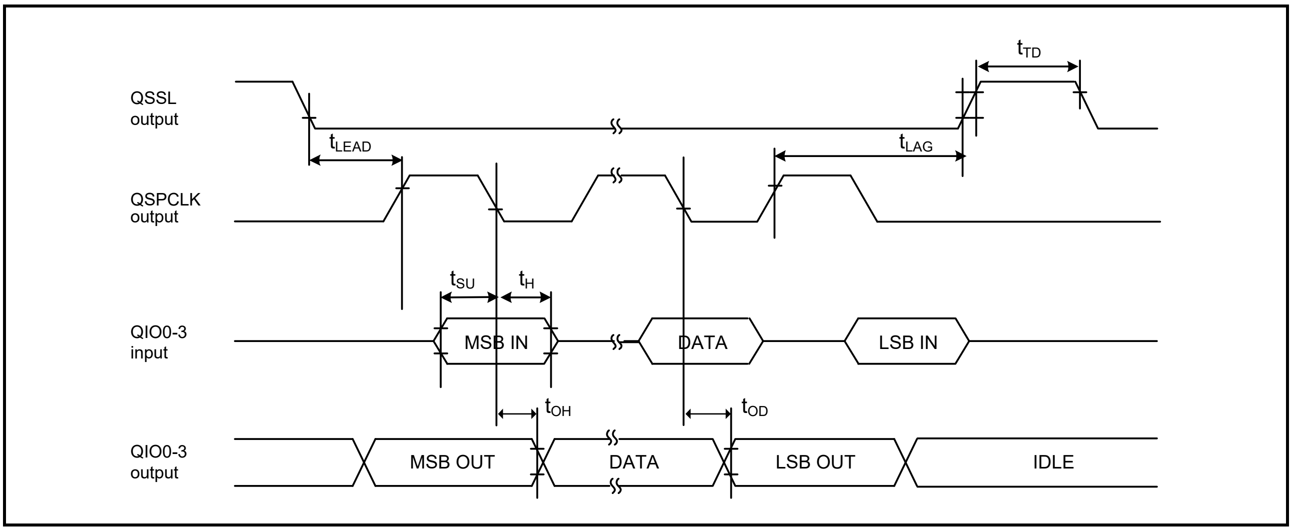


Figure 60.59 Transmit and receive timing

60.3.13 IIC Timing

Table 60.27 IIC timing (1) (1 of 2)

(1) Conditions: Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_A, SCL1_A, SDA1_B, SCL1_B.

(2) The following pins do not require setting: SCL0_A, SDA0_A, SCL2, SDA2.

(3) Use pins that have a letter appended to their names, for instance “_A” or “_B”, to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	Min*1	Max	Unit	Test conditions*3
IIC (Standard mode, SMBus) ICFER.FMPE = 0	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	-	ns	Figure 60.60
	SCL input high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	-	ns	
	SCL input low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	-	ns	
	SCL, SDA input rise time	t_{Sr}	-	1000	ns	
	SCL, SDA input fall time	t_{Sf}	-	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time when wakeup function is disabled	t_{BUF}	3 (6) × t_{IICcyc} + 300	-	ns	
	SDA input bus free time when wakeup function is enabled	t_{BUF}	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	-	ns	
	START condition input hold time when wakeup function is disabled	t_{STAH}	t_{IICcyc} + 300	-	ns	
	START condition input hold time when wakeup function is enabled	t_{STAH}	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	-	ns	
	Repeated START condition input setup time	t_{STAS}	1000	-	ns	
	STOP condition input setup time	t_{STOS}	1000	-	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b	-	400	pF	

Table 60.27 IIC timing (1) (2 of 2)

(1) Conditions: Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: SDA0_B, SCL0_B, SDA1_A, SCL1_A, SDA1_B, SCL1_B.
(2) The following pins do not require setting: SCL0_A, SDA0_A, SCL2, SDA2.
(3) Use pins that have a letter appended to their names, for instance "_A" or "_B", to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter		Symbol	Min*1	Max	Unit	Test conditions*3
IIC (Fast mode)	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	-	ns	Figure 60.60
	SCL input high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	-	ns	
	SCL input low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	-	ns	
	SCL, SDA input rise time	t_{Sr}	20 × (external pullup voltage/5.5V)*2	300	ns	
	SCL, SDA input fall time	t_{Sf}	20 × (external pullup voltage/5.5V)*2	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time when wakeup function is disabled	t_{BUF}	3 (6) × t_{IICcyc} + 300	-	ns	
	SDA input bus free time when wakeup function is enabled	t_{BUF}	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	-	ns	
	START condition input hold time when wakeup function is disabled	t_{STAH}	t_{IICcyc} + 300	-	ns	
	START condition input hold time when wakeup function is enabled	t_{STAH}	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	-	ns	
	Repeated START condition input setup time	t_{STAS}	300	-	ns	
	STOP condition input setup time	t_{STOS}	300	-	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b	-	400	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, t_{Pcyc}: PCLKB cycle.

Note 1. Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.

Note 2. Only supported for SCL0_A, SDA0_A, SCL2, and SDA2.

Note 3. Must use pins that have a letter ("_A", "_B") to indicate group membership appended to their name as groups. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Table 60.28 IIC timing (2)

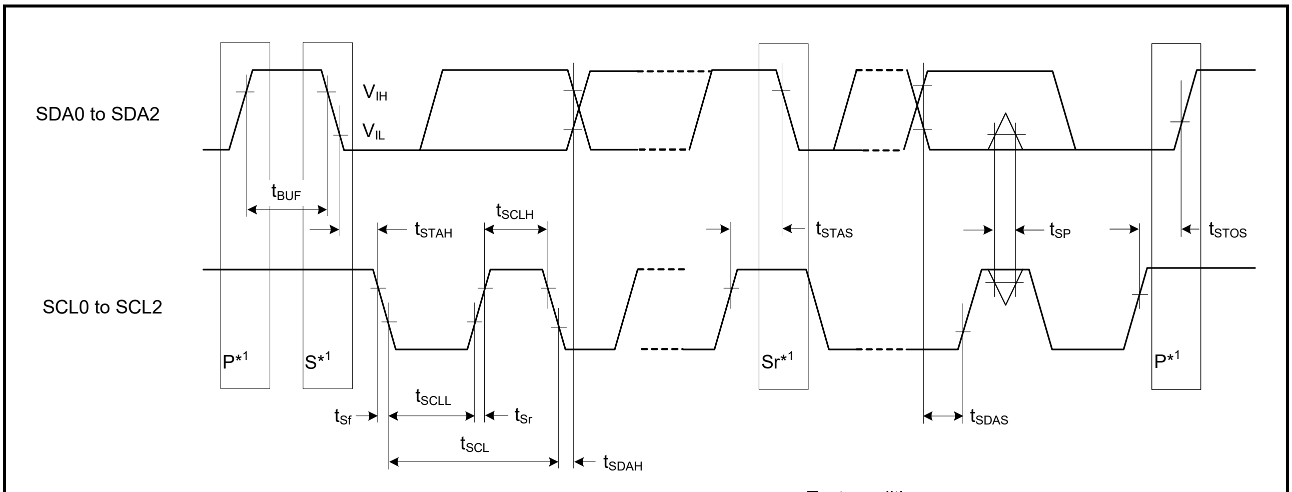
Setting of the SCL0_A, SDA0_A pins is not required with the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min*1,*2	Max	Unit	Test conditions
IIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	t_{SCL}	$6\ (12) \times t_{IICcyc} + 240$	-	ns	Figure 60.60
	SCL input high pulse width	t_{SCLH}	$3\ (6) \times t_{IICcyc} + 120$	-	ns	
	SCL input low pulse width	t_{SCLL}	$3\ (6) \times t_{IICcyc} + 120$	-	ns	
	SCL, SDA input rise time	t_{Sr}	-	120	ns	
	SCL, SDA input fall time	t_{Sf}	-	120	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	$1\ (4) \times t_{IICcyc}$	ns	
	SDA input bus free time when wakeup function is disabled	t_{BUF}	$3\ (6) \times t_{IICcyc} + 120$	-	ns	
	SDA input bus free time when wakeup function is enabled	t_{BUF}	$3\ (6) \times t_{IICcyc} + 4 \times t_{Pcyc} + 120$	-	ns	
	Start condition input hold time when wakeup function is disabled	t_{STAH}	$t_{IICcyc} + 120$	-	ns	
	START condition input hold time when wakeup function is enabled	t_{STAH}	$1\ (5) \times t_{IICcyc} + t_{Pcyc} + 120$	-	ns	
	Restart condition input setup time	t_{STAS}	120	-	ns	
	Stop condition input setup time	t_{STOS}	120	-	ns	
	Data input setup time	t_{SDAS}	$t_{IICcyc} + 30$	-	ns	
	Data input hold time	t_{SDAH}	0	-	ns	
	SCL, SDA capacitive load	C_b	-	550	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, t_{Pcyc}: PCLKB cycle.

Note 1. Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.

Note 2. Cb indicates the total capacity of the bus line.



Note 1. S, P, and Sr indicate the following:
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions:
$V_{IH} = VCC \times 0.7$, $V_{IL} = VCC \times 0.3$
$V_{OL} = 0.6$ V, $I_{OL} = 6$ mA (ICFER.FMPE = 0)
$V_{OL} = 0.4$ V, $I_{OL} = 15$ mA (ICFER.FMPE = 1)

Figure 60.60 I²C bus interface input/output timing

60.3.14 SSIE Timing

Table 60.29 SSIE timing

(1) High drive output is selected with the port drive capability bit in the PmnPFS register.

(2) Use pins that have a letter appended to their names, for instance "_A" or "_B" to indicate group membership. For the SSIE interface, the AC portion of the electrical characteristics is measured for each group.

Parameter			Symbol	Target specification		Unit	Comments
				Min.	Max.		
SSIBCK	Cycle	Master	t_O	80	-	ns	Figure 60.61
		Slave	t_I	80	-	ns	
	High level/ low level	Master	t_{HC}/t_{LC}	0.35	-	t_O	
		Slave		0.35	-	t_I	
	Rising time/falling time	Master	t_{RC}/t_{FC}	-	0.15	t_O / t_I	
		Slave		-	0.15	t_O / t_I	
SSILRCK/SSIFS, SSITXD0, SSIRXD0, SSIDATA1	Input set up time	Master	t_{SR}	12	-	ns	Figure 60.63, Figure 60.64
		Slave		12	-	ns	
	Input hold time	Master	t_{HR}	8	-	ns	
		Slave		15	-	ns	
	Output delay time	Master	t_{DTR}	-10	5	ns	
		Slave		0	20	ns	Figure 60.63, Figure 60.64
	Output delay time from SSILRCK/SSIFS change	Slave	t_{DTRW}	-	20	ns	Figure 60.65*1
GTIOC1A, AUDIO_CLK	Cycle		t_{EXcyc}	20	-	ns	Figure 60.62
	High level/ low level		$t_{EXL}/$ t_{EXH}	0.4	0.6	t_{EXcyc}	

Note 1. For slave-mode transmission, SSIE has a path, through which the signal input from the SSILRCK/SSIFS pin is used to generate transmit data, and the transmit data is logically output to the SSITXD0 or SSIDATA1 pin.

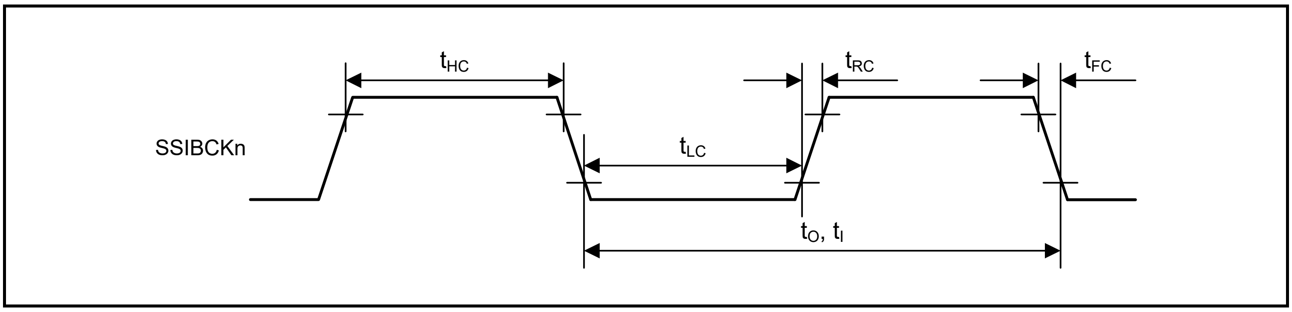


Figure 60.61 SSIE clock input/output timing

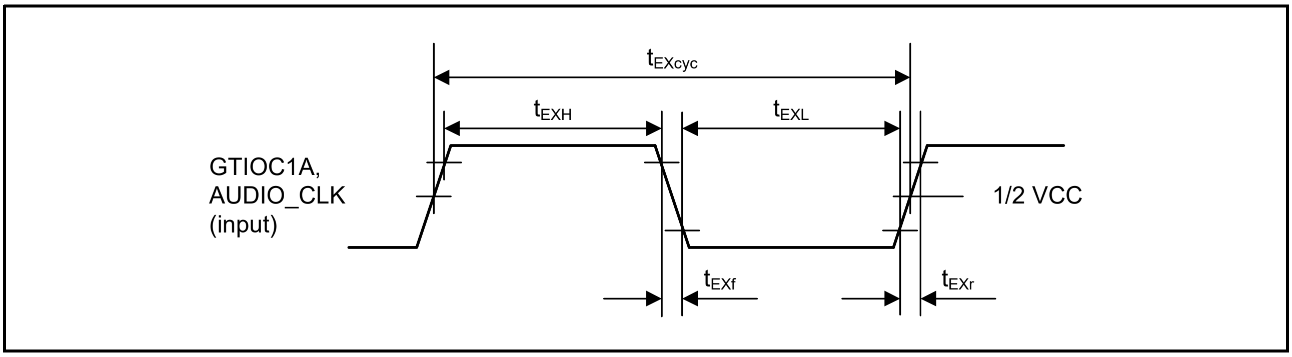


Figure 60.62 Clock input timing

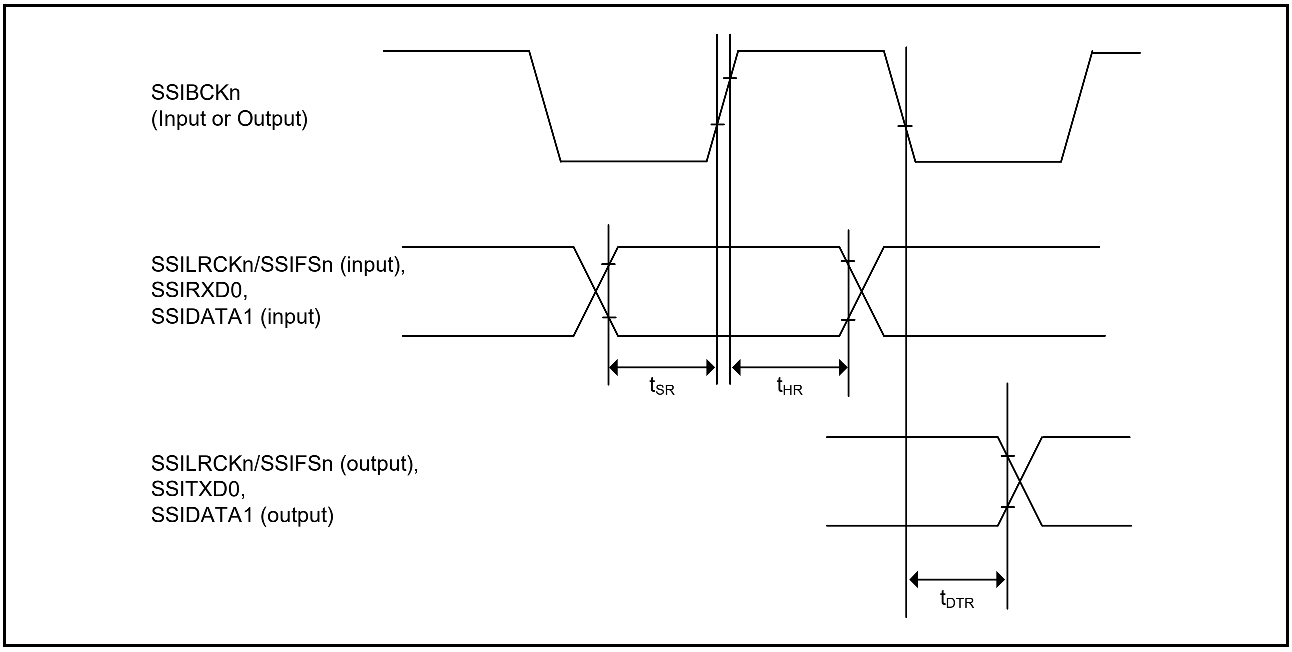


Figure 60.63 SSIE data transmit and receive timing when SSICR.BCKP = 0

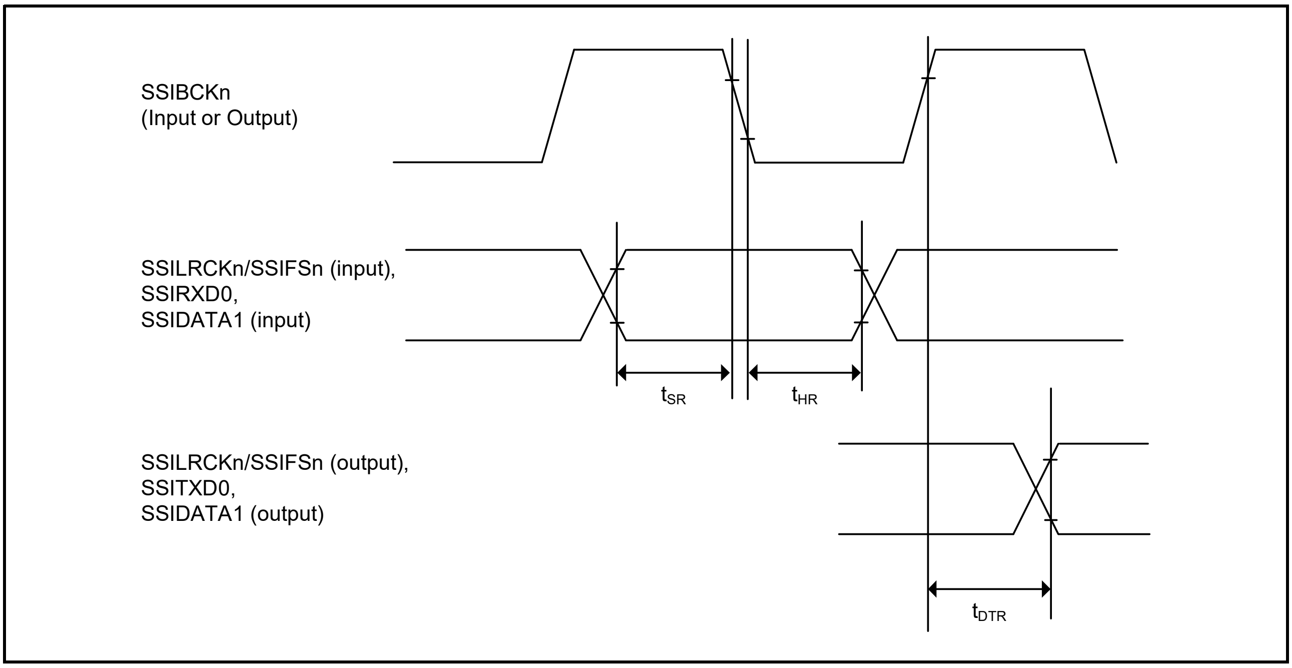


Figure 60.64 SSIE data transmit and receive timing when SSICR.BCKP = 1

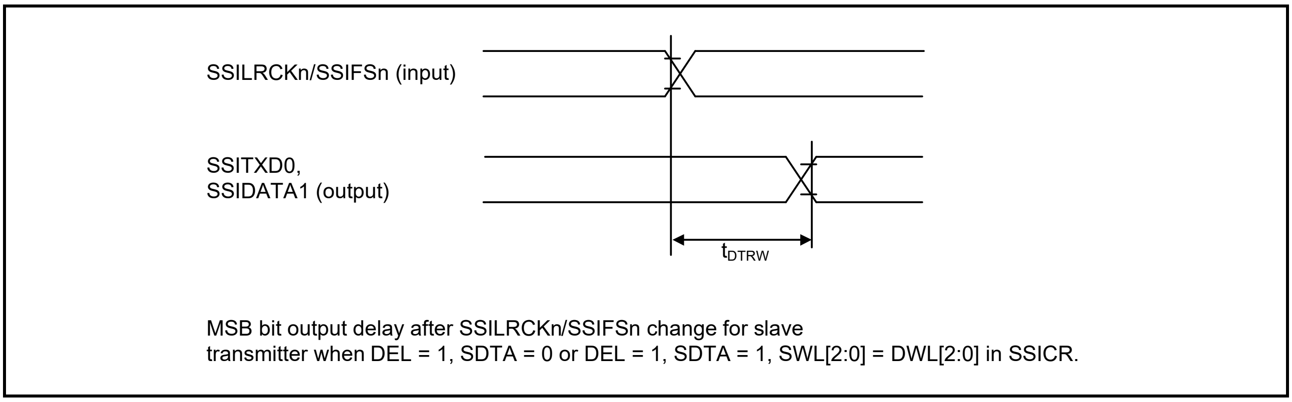


Figure 60.65 SSIE data output delay after SSILRCKn/SSIFSn change

60.3.15 SD/MMC Host Interface Timing

Table 60.30 SD/MMC Host Interface signal timing
Conditions: High drive output is selected in the port drive capability bit in the PmnPFS register.
Clock duty ratio is 50%.

Parameter	Symbol	Min	Max	Unit	Test conditions*1
SDCLK clock cycle	T_{SDCYC}	20	-	ns	Figure 60.66
SDCLK clock high pulse width	T_{SDWH}	6.5	-	ns	
SDCLK clock low pulse width	T_{SDWL}	6.5	-	ns	
SDCLK clock rise time	T_{SDLH}	-	3	ns	
SDCLK clock fall time	T_{SDHL}	-	3	ns	
SDCMD/SDDAT output data delay	T_{SDODLY}	-6	5	ns	
SDCMD/SDDAT input data setup	T_{SDIS}	4	-	ns	
SDCMD/SDDAT input data hold	T_{SDIH}	2	-	ns	

Note 1. Must use pins that have a letter ("_A", "_B") to indicate group membership appended to their name as groups. For

the SD/MMC Host interface, the AC portion of the electrical characteristics is measured for each group.

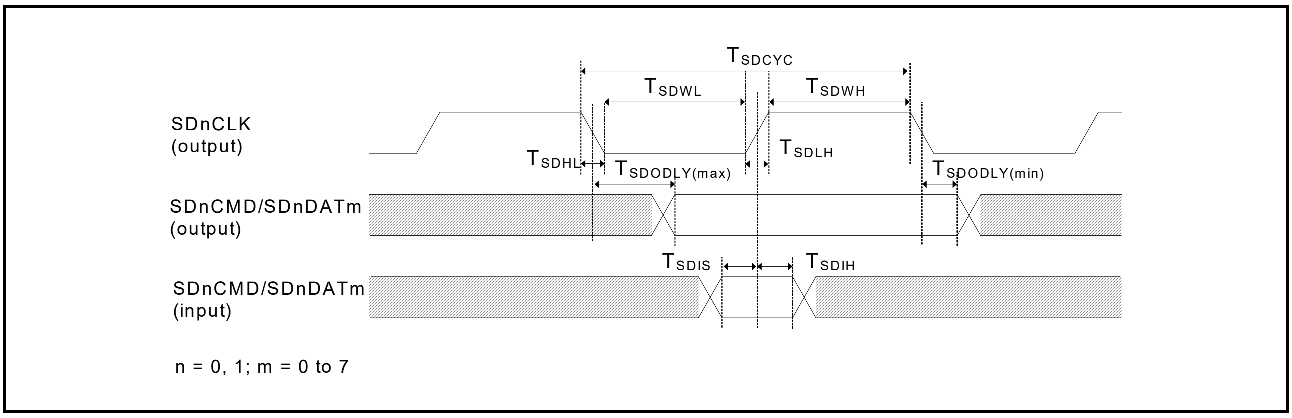


Figure 60.66 SD/MMC Host Interface signal timing

60.3.16 ETHERC Timing

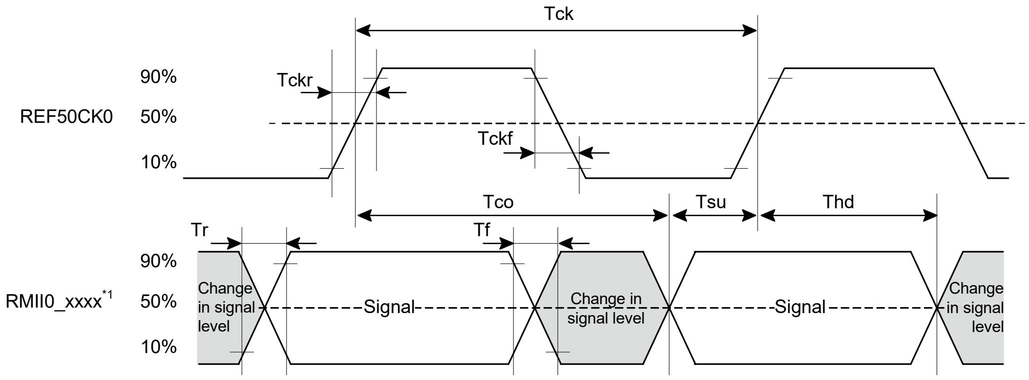
Table 60.31 ETHERC timing

Conditions: ETHERC (RMII): Middle drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: ET0_MDC, ET0_MDIO.
For other pins, high drive output is selected in the port drive capability bit in the PmnPFS register.
ETHERC (MII): Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions*3
ETHERC (RMII)	REF50CK cycle time	T_{ck}	20	-	ns	Figure 60.67 to Figure 60.70
	REF50CK frequency, typical 50 MHz	-	-	50 + 100 ppm	MHz	
	REF50CK duty	-	35	65	%	
	REF50CK rise/fall time	$T_{ckr/ckf}$	0.5	3.5	ns	
	RMII0_xxxx*1 output delay	T_{co}	2.5	12.0	ns	
	RMII0_xxxx*2 setup time	T_{su}	3	-	ns	
	RMII0_xxxx*2 hold time	T_{hd}	1	-	ns	
	RMII0_xxxx*1, *2 rise/fall time	T_r/T_f	0.5	4	ns	
	ET0_WOL output delay	t_{WOLd}	1	23.5	ns	Figure 60.71
ETHERC (MII)	ET0_TX_CLK cycle time	t_{Tcyc}	40	-	ns	-
	ET0_TX_EN output delay	t_{TENd}	1	20	ns	Figure 60.72
	ET0_ETXD0 to ET0_ETXD3 output delay	t_{MTDd}	1	20	ns	
	ET0_CRS setup time	t_{CRSs}	10	-	ns	
	ET0_CRS hold time	t_{CRSh}	10	-	ns	
	ET0_COL setup time	t_{COLs}	10	-	ns	Figure 60.73
	ET0_COL hold time	t_{COLh}	10	-	ns	
	ET0_RX_CLK cycle time	t_{TRcyc}	40	-	ns	-
	ET0_RX_DV setup time	t_{RDVs}	10	-	ns	Figure 60.74
	ET0_RX_DV hold time	t_{RDVh}	10	-	ns	
	ET0_ERXD0 to ET0_ERXD3 setup time	t_{MRDs}	10	-	ns	
	ET0_ERXD0 to ET0_ERXD3 hold time	t_{MRDh}	10	-	ns	
	ET0_RX_ER setup time	t_{RERs}	10	-	ns	Figure 60.75
	ET0_RX_ER hold time	t_{RESh}	10	-	ns	
	ET0_WOL output delay	t_{WOLd}	1	23.5	ns	Figure 60.76

Note 1. RMII0_TXD_EN, RMII0_TXD1, RMII0_TXD0.
Note 2. RMII0_CRS_DV, RMII0_RXD1, RMII0_RXD0, RMII0_RX_ER.

Note 3. The following pins, must use pins that have a letter ("_A", "_B") to indicate group membership appended to their name as groups. For the ETHERC (RMII) Host interface, the AC portion of the electrical characteristics is measured for each group. REF50CK0_A, REF50CK0_B, RMII0_xxxx_A, RMII0_xxxx_B



Note 1. RMII0_TXD_EN, RMII0_TXD1, RMII0_TXD0, RMII0_CRS_DV, RMII0_RXD1, RMII0_RXD0, RMII0_RX_ER

Figure 60.67 REF50CK0 and RMII signal timing

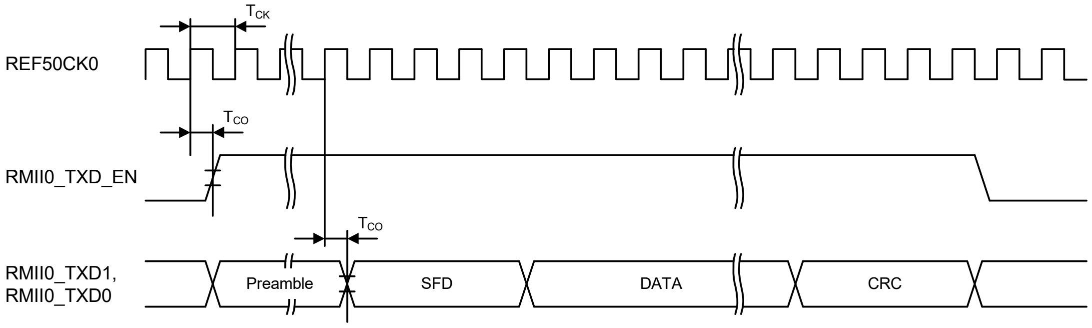


Figure 60.68 RMII transmission timing

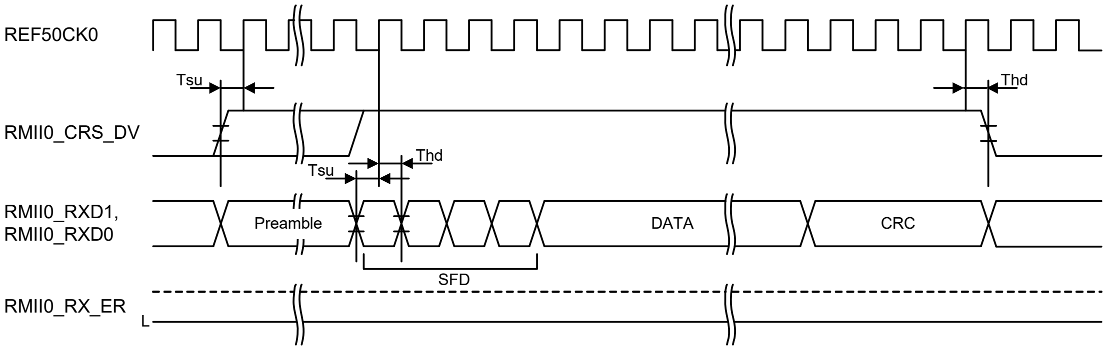


Figure 60.69 RMII reception timing in normal operation

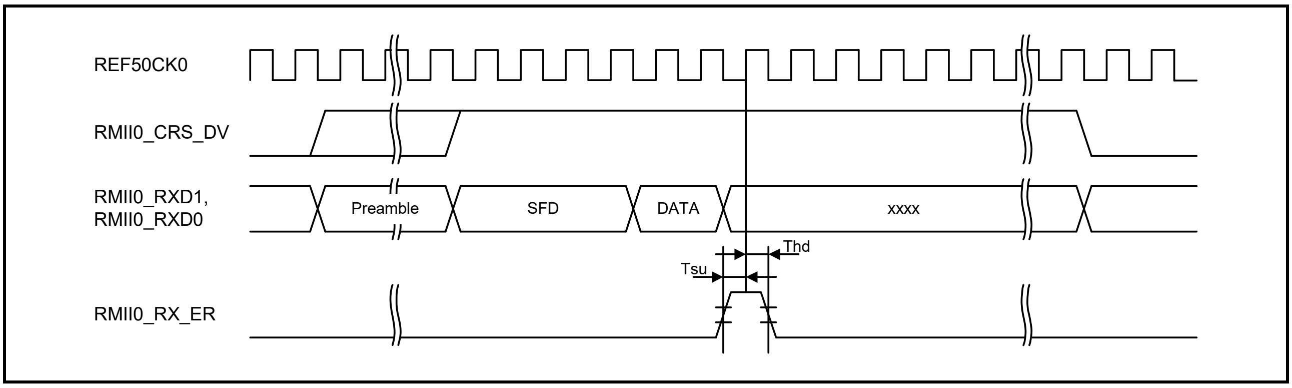


Figure 60.70 RMII reception timing when an error occurs

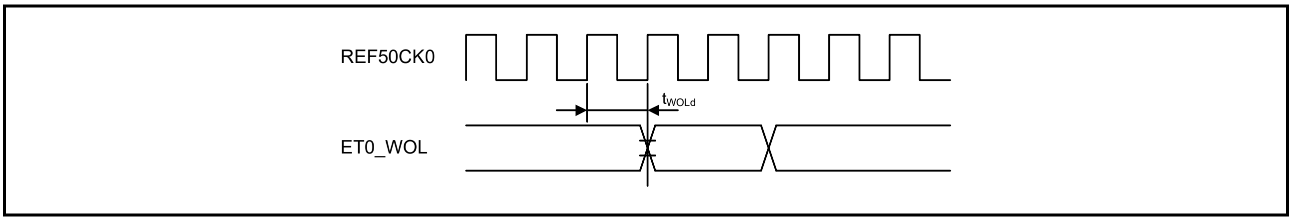


Figure 60.71 WOL output timing for RMII

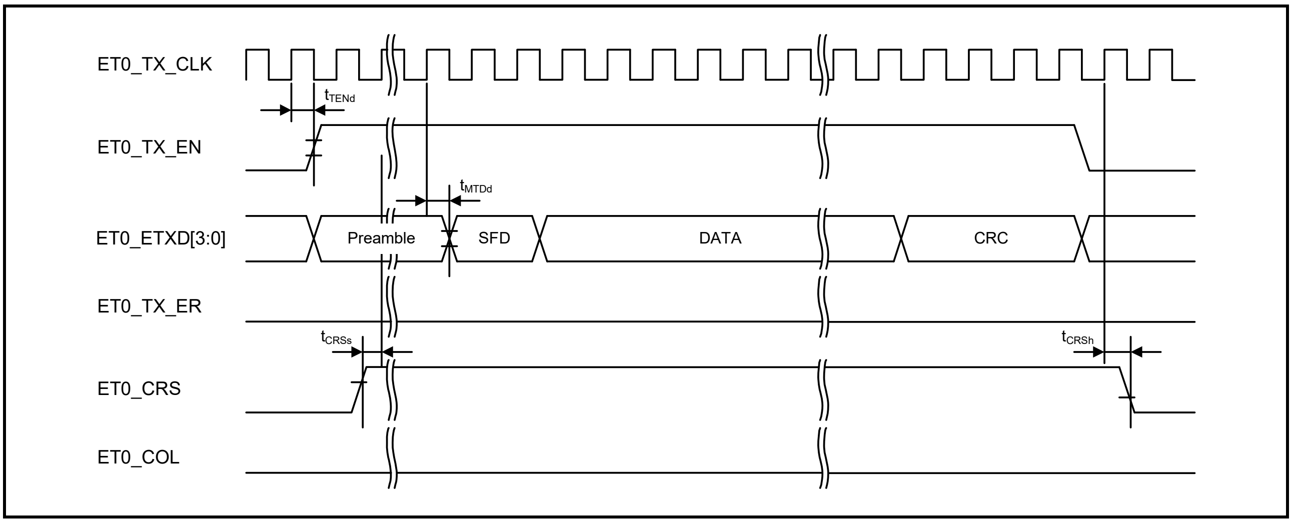


Figure 60.72 MII transmission timing in normal operation

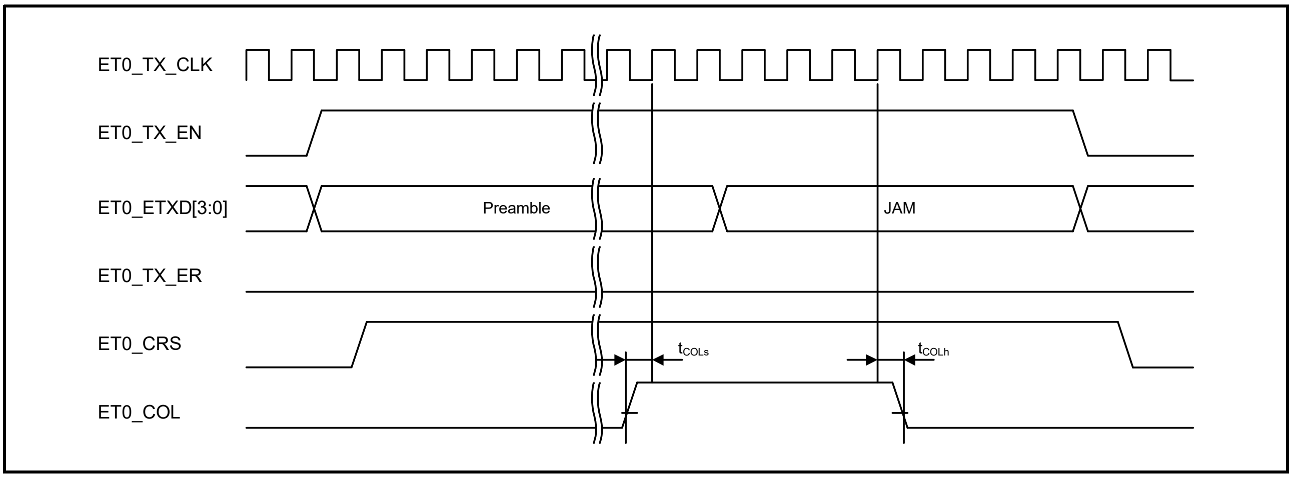


Figure 60.73 MII transmission timing when a conflict occurs

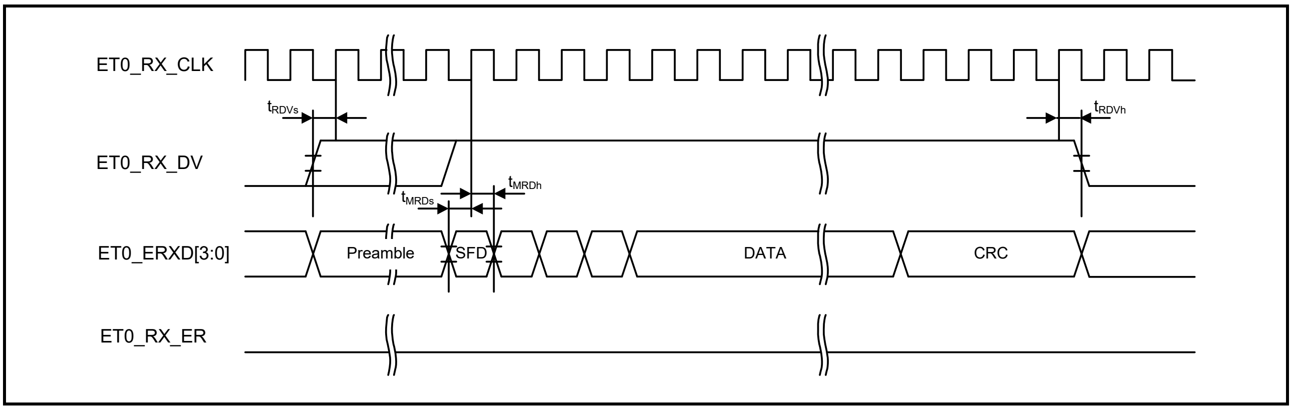


Figure 60.74 MII reception timing in normal operation

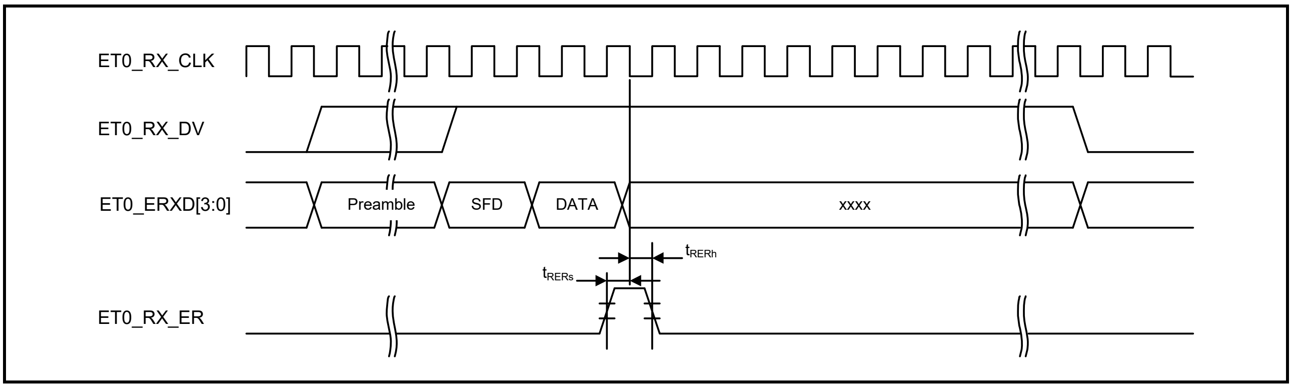


Figure 60.75 MII reception timing when an error occurs

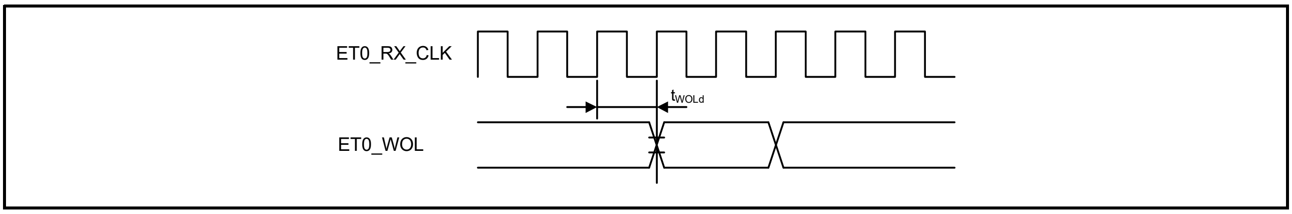


Figure 60.76 WOL output timing for MII

60.3.17 PDC Timing

Table 60.32 PDC timing

Conditions: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF

Parameter		Symbol	Min	Max	Unit	Test conditions
PDC	PIXCLK input cycle time	t_{PIXcyc}	37	-	ns	Figure 60.77
	PIXCLK input high pulse width	t_{PIXH}	10	-	ns	
	PIXCLK input low pulse width	t_{PIXL}	10	-	ns	
	PIXCLK rise time	t_{PIXr}	-	5	ns	
	PIXCLK fall time	t_{PIXf}	-	5	ns	
	PCKO output cycle time	t_{PCKcyc}	$2 \times t_{PBcyc}$	-	ns	Figure 60.78
	PCKO output high pulse width	t_{PCKH}	$(t_{PCKcyc} - t_{PCKr} - t_{PCKf})/2 - 3$	-	ns	
	PCKO output low pulse width	t_{PCKL}	$(t_{PCKcyc} - t_{PCKr} - t_{PCKf})/2 - 3$	-	ns	
	PCKO rise time	t_{PCKr}	-	5	ns	
	PCKO fall time	t_{PCKf}	-	5	ns	
	VSYNV/HSYNC input setup time	t_{SYNCS}	10	-	ns	Figure 60.79
	VSYNV/HSYNC input hold time	t_{SYNCH}	5	-	ns	
	PIXD input setup time	t_{PIXDS}	10	-	ns	
	PIXD input hold time	t_{PIXDH}	5	-	ns	

Note 1. t_{PBcyc}: PCLKB cycle.

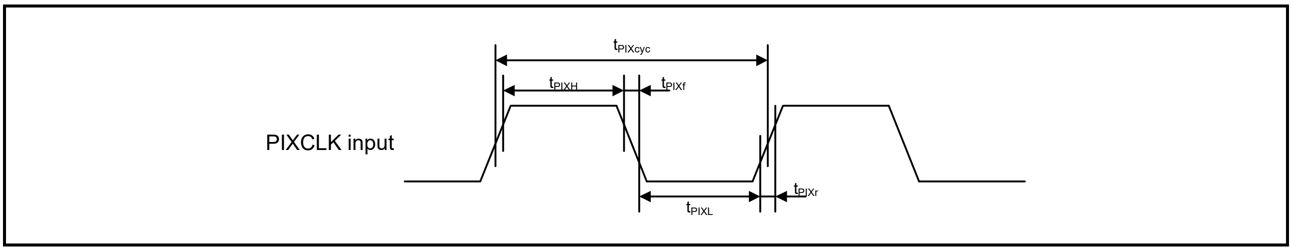


Figure 60.77 PDC input clock timing

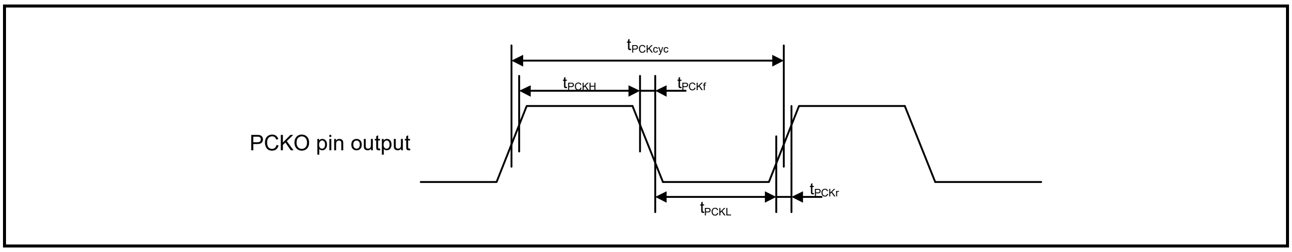


Figure 60.78 PDC output clock timing

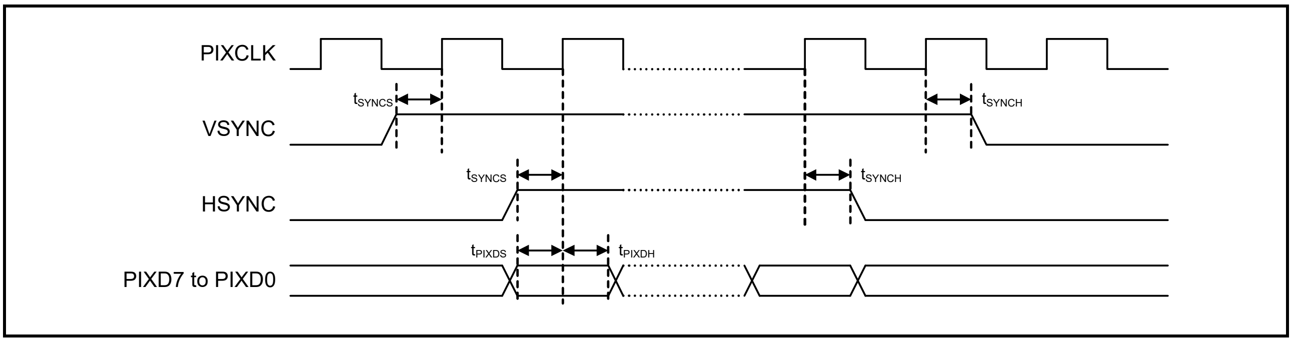


Figure 60.79 PDC AC timing

60.3.18 GLCDC Timing

Table 60.33 GLCDC timing

Conditions:
LCD_CLK: High drive output is selected in the port drive capability bit in the PmnPFS register.
LCD_DATA: Middle drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
LCD_EXTCLK input clock frequency		t_{Ecyc}	-	-	60*1	MHz	Figure 60.80
LCD_EXTCLK input clock low pulse width		t_{WL}	0.45	-	0.55	t_{Ecyc}	
LCD_EXTCLK input clock high pulse width		t_{WH}	0.45	-	0.55		
LCD_CLK output clock frequency		t_{Lcyc}	-	-	60*1	MHz	Figure 60.81
LCD_CLK output clock low pulse width		t_{LOL}	0.4	-	0.6	t_{Lcyc}	Figure 60.81
LCD_CLK output clock high pulse width		t_{LOH}	0.4	-	0.6	t_{Lcyc}	Figure 60.81
LCD data output delay timing	_A or _B combinations*2	t_{DD}	-3.5	-	4	ns	Figure 60.82
	_A and _B combinations*3		-5.0	-	5.5		

Note 1. Parallel RGB888, 666,565: Maximum 54 MHz
Serial RGB888: Maximum 60 MHz (4x speed)

Note 2. Use pins that have a letter appended to their names, for instance, “_A” or “_B”, to indicate

Note 3. Pins of group “_A” and “_B” combinations are used.

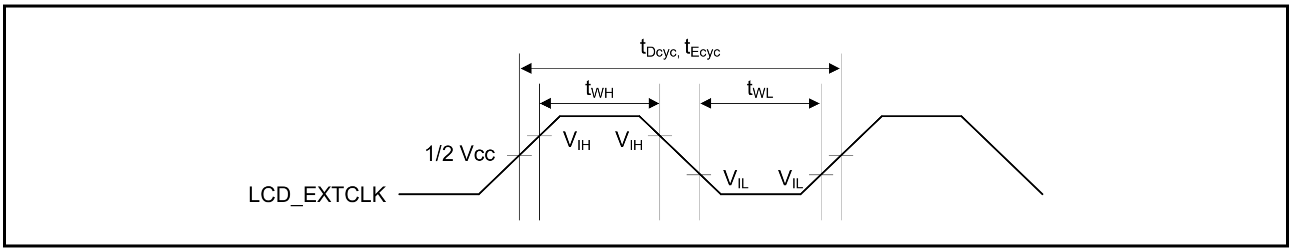


Figure 60.80 LCD_EXTCLK clock input timing

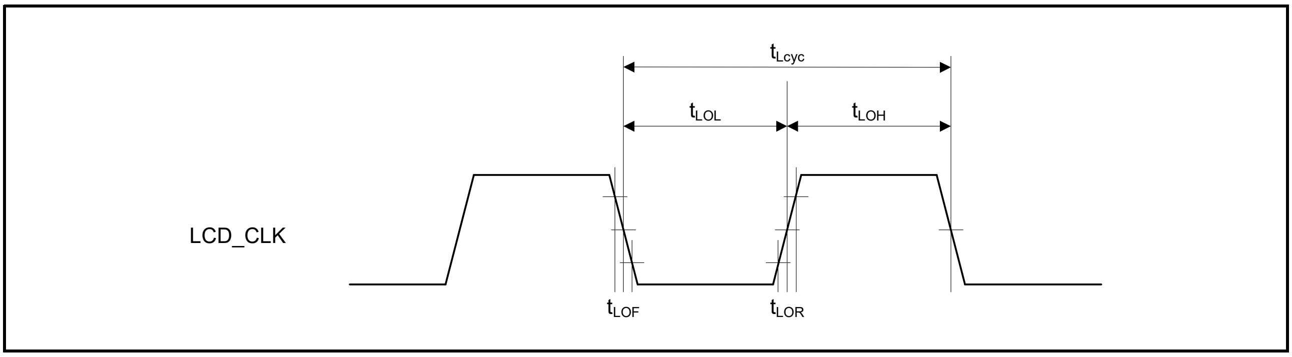


Figure 60.81 LCD_CLK clock output timing

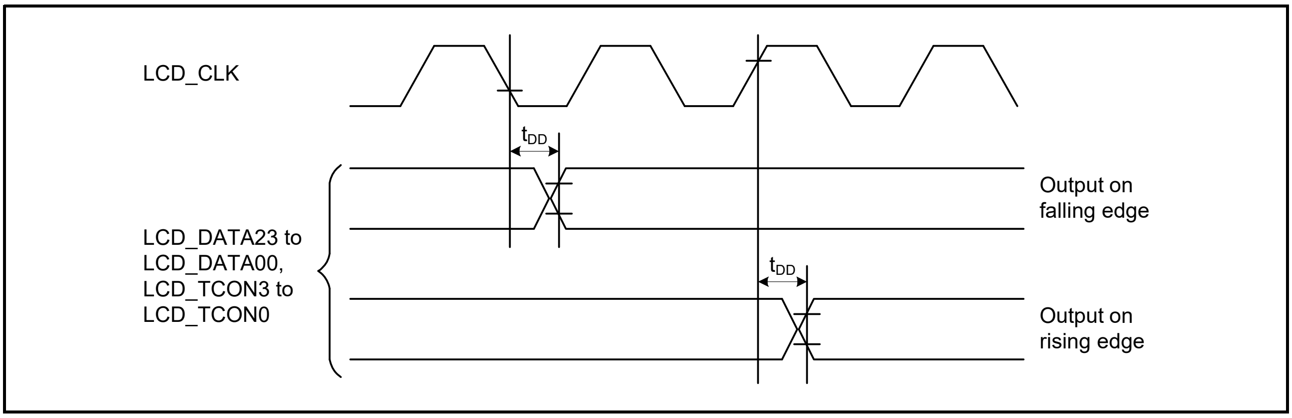


Figure 60.82 Display output timing

60.4 USB Characteristics

60.4.1 USBHS Timing

Table 60.34 USBHS low-speed characteristics for host only (USBHS_DP and USBHS_DM pin characteristics)
Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24 MHz, UCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions	
Input characteristics	Input high voltage	V_{IH}	2.0	-	-	V	-	-
	Input low voltage	V_{IL}	-	-	0.8	V	-	-
	Differential input sensitivity	V_{DI}	0.2	-	-	V	| USBHS_DP - USBHS_DM |	-
	Differential common-mode range	V_{CM}	0.8	-	2.5	V	-	-
Output characteristics	Output high voltage	V_{OH}	2.8	-	3.6	V	I_{OH} = -200 μA	-
	Output low voltage	V_{OL}	0.0	-	0.3	V	I_{OL}= 2 mA	-
	Cross-over voltage	V_{CRS}	1.3	-	2.0	V	-	Figure 60.83, Figure 60.84
	Rise time	t_{LR}	75	-	300	ns	-	
	Fall time	t_{LF}	75	-	300	ns	-	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	-	125	%	t_{LR} / t_{LF}	-
Pull-up, Pull-down characteristics	USBHS_DP and USBHS_DM pull-down resistors (Host)	R_{pd}	14.25	-	24.80	kΩ	-	

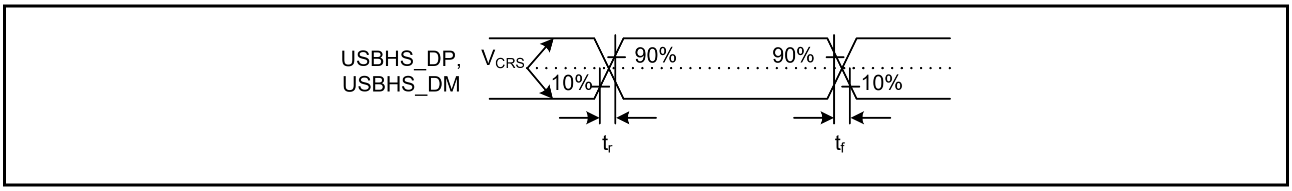


Figure 60.83 USBHS_DP and USBHS_DM output timing in low-speed mode

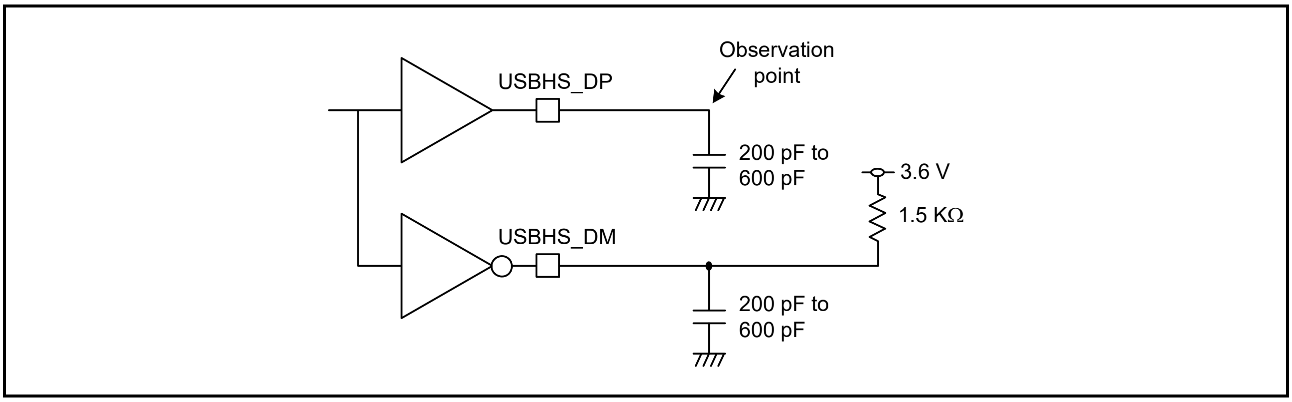


Figure 60.84 Test circuit in low-speed mode

Table 60.35 USBHS full-speed characteristics (USBHS_DP and USBHS_DM pin characteristics)
Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24 MHz, UCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions	
Input characteristics	Input high voltage	V_{IH}	2.0	-	-	V	-	-
	Input low voltage	V_{IL}	-	-	0.8	V	-	-
	Differential input sensitivity	V_{DI}	0.2	-	-	V	| USBHS_DP - USBHS_DM |	-
	Differential common-mode range	V_{CM}	0.8	-	2.5	V	-	-
Output characteristics	Output high voltage	V_{OH}	2.8	-	3.6	V	I_{OH} = -200 μA	-
	Output low voltage	V_{OL}	0.0	-	0.3	V	I_{OL}= 2 mA	-
	Cross-over voltage	V_{CRS}	1.3	-	2.0	V	-	Figure 60.85, Figure 60.86
	Rise time	t_{LR}	4	-	20	ns	-	
	Fall time	t_{LF}	4	-	20	ns	-	
	Rise/fall time ratio	t_{LR} / t_{LF}	90	-	111.11	%	t_{FR} / t_{FF}	-
	Output resistance	Z_{DRV}	40.5	-	49.5	Ω	Rs Not used (PHYSET.REPSEL[1:0] = 01b and PHYSET. HSEB = 0)	
DC characteristics	USBHS_DM pull-up resistor (device)	R_{pu}	0.900	-	1.575	kΩ	During idle state	
			1.425	-	3.090	kΩ	During transmission and reception	
	USBHS_DP/USBHS_DM pull-down resistor (host)	R_{pd}	14.25	-	24.80	kΩ	-	

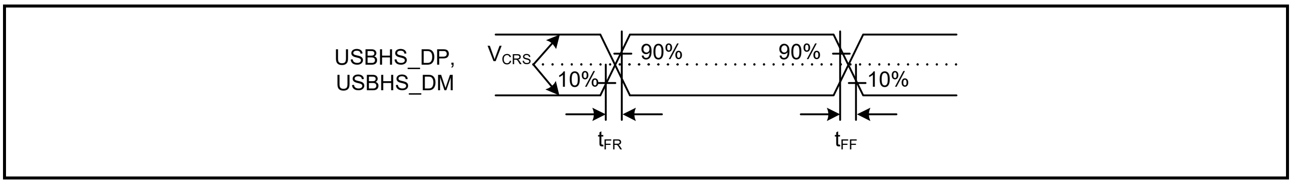


Figure 60.85 USBHS_DP and USBHS_DM output timing in full-speed mode

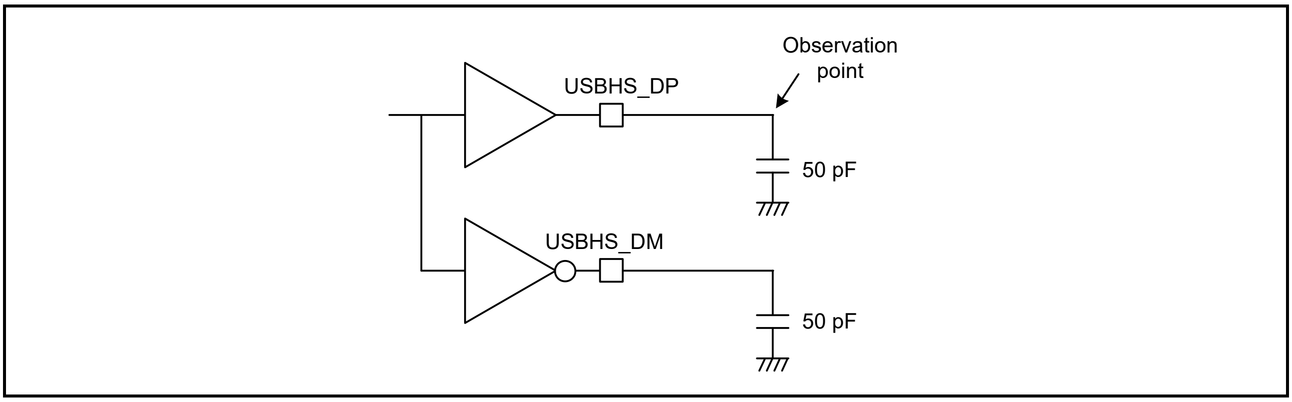


Figure 60.86 Test circuit in full-speed mode

Table 60.36 **USBHS high-speed characteristics (USBHS_DP and USBHS_DM pin characteristics)**
Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Squelch detect sensitivity	V_{HSSQ}	100	-	150	mV	Figure 60.87
	Disconnect detect sensitivity	V_{HSDSC}	525	-	625	mV	Figure 60.88
	Common-mode voltage	V_{HSCM}	-50	-	500	mV	-
Output characteristics	Idle state	V_{HSOI}	-10.0	-	10	mV	-
	Output high voltage	V_{HSOH}	360	-	440	mV	
	Output low voltage	V_{HSOL}	-10.0	-	10	mV	
	Chirp J output voltage (difference)	V_{CHIRPJ}	700	-	1100	mV	
	Chirp K output voltage (difference)	V_{CHIRPK}	-900	-	-500	mV	
AC characteristics	Rise time	t_{HSR}	500	-	-	ps	Figure 60.89
	Fall time	t_{HSF}	500	-	-	ps	
	Output resistance	Z_{HSDRV}	40.5	-	49.5	Ω	-

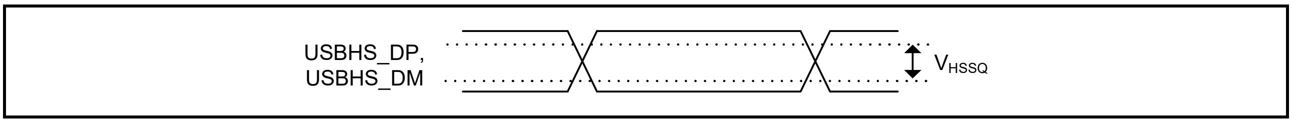


Figure 60.87 USBHS_DP and USBHS_DM squelch detect sensitivity in high-speed mode

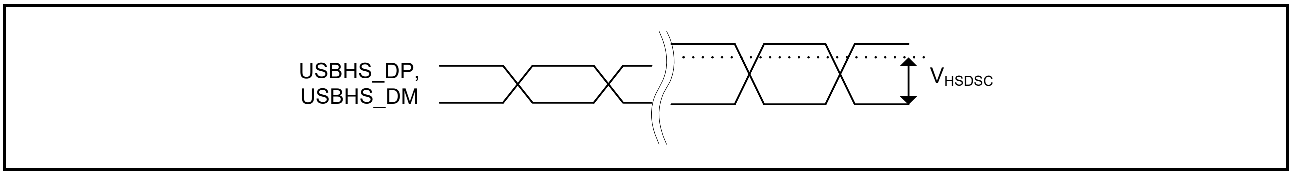


Figure 60.88 USBHS_DP and USBHS_DM disconnect detect sensitivity in high-speed mode

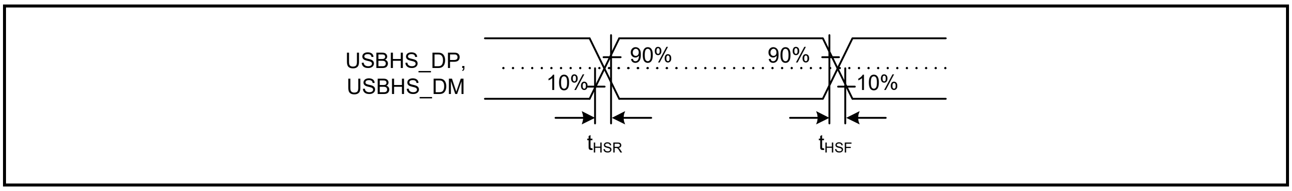


Figure 60.89 USBHS_DP and USBHS_DM output timing in high-speed mode

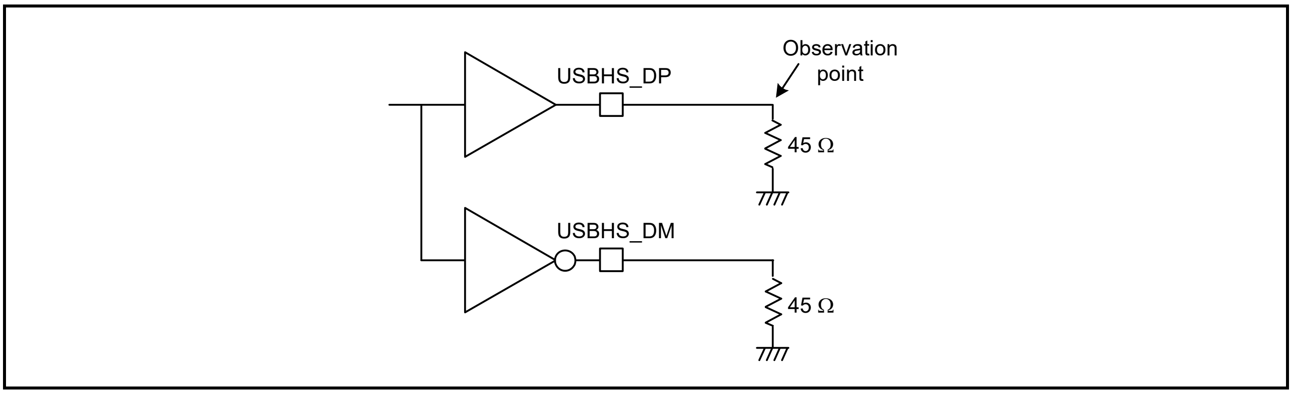


Figure 60.90 Test circuit in high-speed mode

Table 60.37 USBHS high-speed characteristics (USBHS_DP and USBHS_DM pin characteristics)
Conditions: USBHS_RREF = 2.2 kΩ ± 1%, USBMCLK = 12/20/24 MHz

Parameter		Symbol	Min	Max	Unit	Test conditions
Battery Charging Specification	D+ sink current	I_{DP_SINK}	25	175	µA	-
	D- sink current	I_{DM_SINK}	25	175	µA	-
	DCD source current	I_{DP_SRC}	7	13	µA	-
	Data detection voltage	V_{DAT_REF}	0.25	0.4	V	-
	D+ source voltage	V_{DP_SRC}	0.5	0.7	V	Output current = 250 µA
	D- source voltage	V_{DM_SRC}	0.5	0.7	V	Output current = 250 µA

60.4.2 USBFS Timing

Table 60.38 USBFS low-speed characteristics for host only (USB_DP and USB_DM pin characteristics) (1 of 2)
Conditions: VCC = AVCC0 = VCC_USB = VBATT = 3.0 to 3.6V, 2.7 ≤ VREFH0/VREFH ≤ AVCC0, VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V, UCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	-	-	V	-
	Input low voltage	V_{IL}	-	-	0.8	V	-
	Differential input sensitivity	V_{DI}	0.2	-	-	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	-	2.5	V	-
Output characteristics	Output high voltage	V_{OH}	2.8	-	3.6	V	I_{OH} = -200 µA
	Output low voltage	V_{OL}	0.0	-	0.3	V	I_{OL}= 2 mA
	Cross-over voltage	V_{CRS}	1.3	-	2.0	V	Figure 60.91
	Rise time	t_{LR}	75	-	300	ns	
	Fall time	t_{LF}	75	-	300	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	-	125	%	t_{LR}/ t_{LF}

Table 60.38 USBFS low-speed characteristics for host only (USB_DP and USB_DM pin characteristics) (2 of 2)
Conditions: VCC = AVCC0 = VCC_USB = VBATT = 3.0 to 3.6V, 2.7 ≤ VREFH0/VREFH ≤ AVCC0, VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V, UCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Pull-up and pull-down characteristics	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	-	24.80	kΩ	-

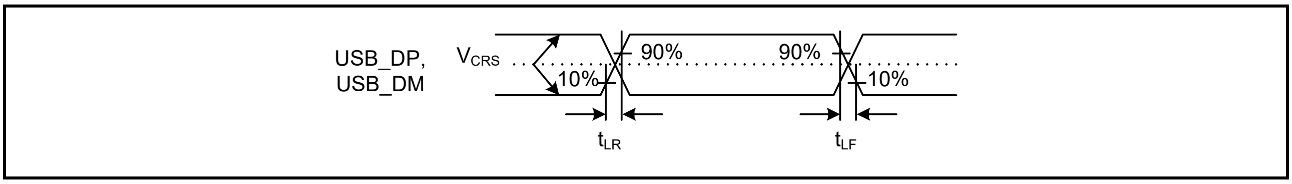


Figure 60.91 USB_DP and USB_DM output timing in low-speed mode

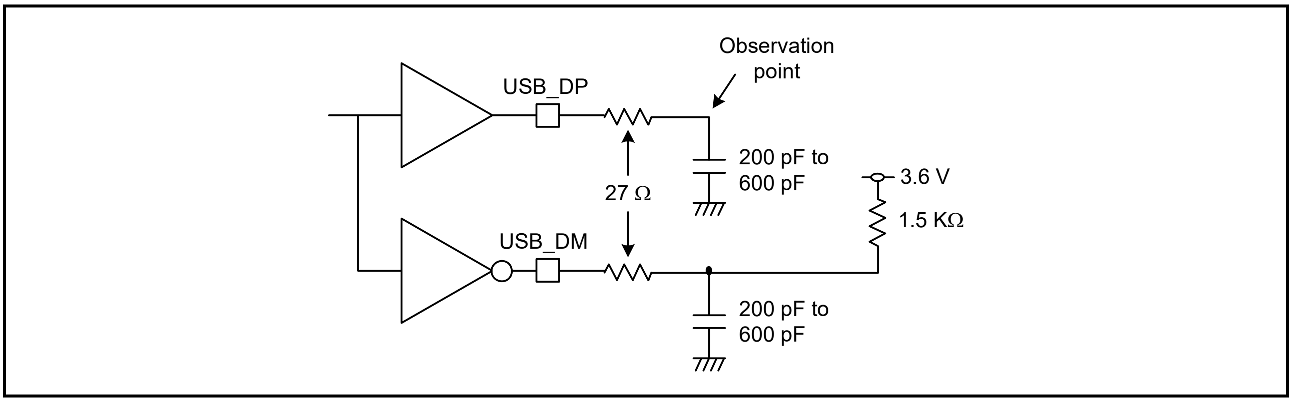


Figure 60.92 Test circuit in low-speed mode

Table 60.39 USBFS full-speed characteristics (USB_DP and USB_DM pin characteristics)
Conditions: VCC = AVCC0 = VCC_USB = VBATT = 3.0 to 3.6 V, 2.7 ≤ VREFH0/VREFH ≤ AVCC0, VCC_USBHS = AVCC_USBHS = 3.0 to 3.6 V, UCLK = 48 MHz

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input characteristics	Input high voltage	V_{IH}	2.0	-	-	V	-
	Input low voltage	V_{IL}	-	-	0.8	V	-
	Differential input sensitivity	V_{DI}	0.2	-	-	V	| USB_DP - USB_DM |
	Differential common-mode range	V_{CM}	0.8	-	2.5	V	-
Output characteristics	Output high voltage	V_{OH}	2.8	-	3.6	V	I_{OH} = -200 μA
	Output low voltage	V_{OL}	0.0	-	0.3	V	I_{OL}= 2 mA
	Cross-over voltage	V_{CRS}	1.3	-	2.0	V	Figure 60.93
	Rise time	t_{LR}	4	-	20	ns	
	Fall time	t_{LF}	4	-	20	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	90	-	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	28	-	44	Ω	USBFS: Rs = 27 Ω included
Pull-up and pull-down characteristics	DM pull-up resistance in device controller mode	R_{pu}	0.900	-	1.575	kΩ	During idle state
			1.425	-	3.090	kΩ	During transmission and reception
	USB_DP and USB_DM pull-down resistance in host controller mode	R_{pd}	14.25	-	24.80	kΩ	-

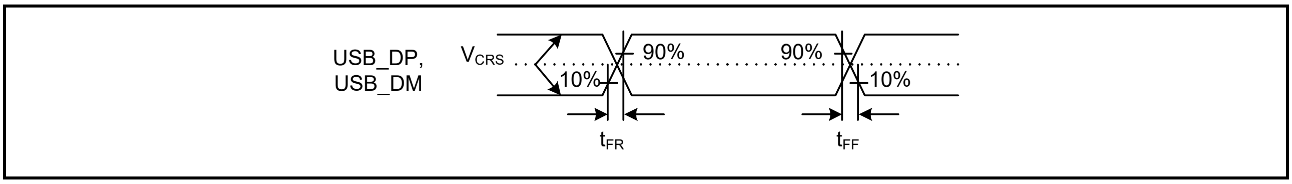


Figure 60.93 USB_DP and USB_DM output timing in full-speed mode

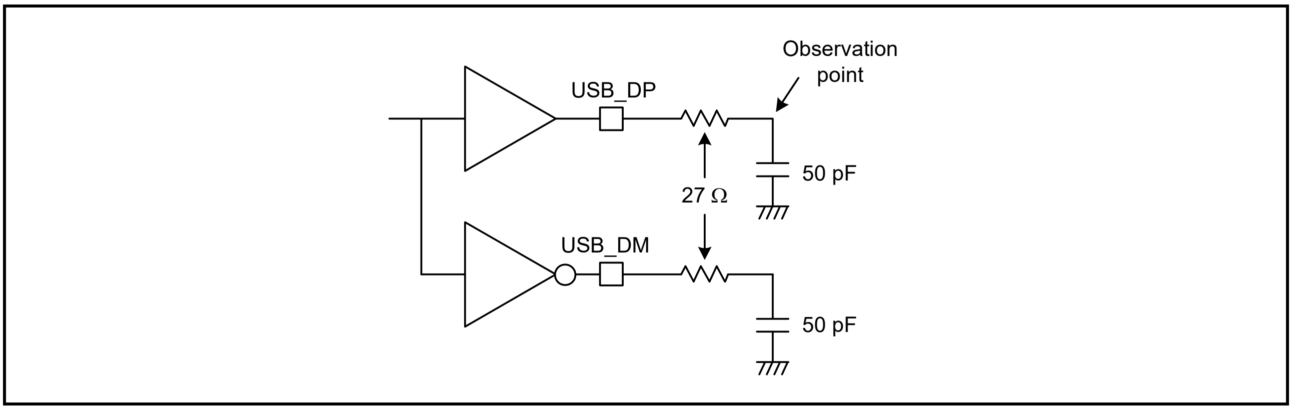


Figure 60.94 Test circuit in full-speed mode

60.5 ADC12 Characteristics

Table 60.40 A/D conversion characteristics for unit 0 (1 of 2)
Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	-	60	MHz	-
Analog input capacitance			-	-	30	pF	-
Quantization error			-	±0.5	-	LSB	-
Resolution			-	-	12	Bits	-
Channel-dedicated sample-and-hold circuits in use (AN000 to AN002)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	1.06 (0.4 + 0.25)*2	-	-	μs	• Sampling of channel-dedicated sample-and-hold circuits in 24 states • Sampling in 15 states
	Offset error		-	±1.5	±3.5	LSB	AN000 to AN002 = 0.25 V
	Full-scale error		-	±1.5	±3.5	LSB	AN000 to AN002 = VREFH0- 0.25 V
	Absolute accuracy		-	±2.5	±5.5	LSB	-
	DNL differential nonlinearity error		-	±1.0	±2.0	LSB	-
	INL integral nonlinearity error		-	±1.5	±3.0	LSB	-
	Holding characteristics of sample-and hold circuits		-	-	20	μs	-
	Dynamic range		0.25	-	VREFH 0 - 0.25	V	-
Channel-dedicated sample-and-hold circuits not in use (AN000 to AN002)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	-	-	μs	Sampling in 16 states
	Offset error		-	±1.0	±2.5	LSB	-
	Full-scale error		-	±1.0	±2.5	LSB	-
	Absolute accuracy		-	±2.0	±4.5	LSB	-
	DNL differential nonlinearity error		-	±0.5	±1.5	LSB	-
	INL integral nonlinearity error		-	±1.0	±2.5	LSB	-

Table 60.40 A/D conversion characteristics for unit 0 (2 of 2)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
High-precision channels (AN003 to AN007)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	-	-	μs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	-	-	μs	Sampling in 11 states VCC = AVCC0 = 3.0 to 3.6 V 3.0 V ≤ VREFH0 ≤ AVCC0
	Offset error		-	±1.0	±2.5	LSB	-
	Full-scale error		-	±1.0	±2.5	LSB	-
	Absolute accuracy		-	±2.0	±4.5	LSB	-
	DNL differential nonlinearity error		-	±0.5	±1.5	LSB	-
	INL integral nonlinearity error		-	±1.0	±2.5	LSB	-
Normal-precision channels (AN016 to AN020)	Conversion time*1 (Operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.88 (0.667)*2	-	-	μs	Sampling in 40 states
	Offset error		-	±1.0	±5.5	LSB	-
	Full-scale error		-	±1.0	±5.5	LSB	-
	Absolute accuracy		-	±2.0	±7.5	LSB	-
	DNL differential nonlinearity error		-	±0.5	±4.5	LSB	-
	INL integral nonlinearity error		-	±1.0	±5.5	LSB	-

Note: These specification values apply when there is no access to the external bus during A/D conversion. If access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage is stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

Table 60.41 A/D conversion characteristics for unit 1 (1 of 2)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	-	60	MHz	-
Analog input capacitance			-	-	30	pF	-
Quantization error			-	±0.5	-	LSB	-
Resolution			-	-	12	Bits	-
Channel-dedicated sample-and-hold circuits in use (AN100 to AN102)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	1.06 (0.4 + 0.25)*2	-	-	μs	• Sampling of channel-dedicated sample-and-hold circuits in 24 states • Sampling in 15 states
	Offset error		-	±1.5	±3.5	LSB	AN100 to AN102 = 0.25 V
	Full-scale error		-	±1.5	±3.5	LSB	AN100 to AN102 = VREFH - 0.25 V
	Absolute accuracy		-	±2.5	±5.5	LSB	-
	DNL differential nonlinearity error		-	±1.0	±2.0	LSB	-
	INL integral nonlinearity error		-	±1.5	±3.0	LSB	-
	Holding characteristics of sample-and hold circuits		-	-	20	μs	-
	Dynamic range		0.25	-	VREFH - 0.25	V	-

Table 60.41 A/D conversion characteristics for unit 1 (2 of 2)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Channel-dedicated sample-and-hold circuits not in use (AN100 to AN102)	Conversion time*1 (Operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	-	-	μs	Sampling in 16 states
	Offset error		-	±1.0	±2.5	LSB	-
	Full-scale error		-	±1.0	±2.5	LSB	-
	Absolute accuracy		-	±2.0	±4.5	LSB	-
	DNL differential nonlinearity error		-	±0.5	±1.5	LSB	-
	INL integral nonlinearity error		-	±1.0	±2.5	LSB	-
High-precision channels (AN103, AN105 to AN107)	Conversion time*1 (Operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	-	-	μs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	-	-	μs	Sampling in 11 states VCC = AVCC0 = 3.0 to 3.6 V 3.0 V ≤ VREFH ≤ AVCC0
	Offset error		-	±1.0	±2.5	LSB	-
	Full-scale error		-	±1.0	±2.5	LSB	-
	Absolute accuracy		-	±2.0	±4.5	LSB	-
	DNL differential nonlinearity error		-	±0.5	±1.5	LSB	-
	INL integral nonlinearity error		-	±1.0	±2.5	LSB	-
Normal-precision channels (AN116 to AN119)	Conversion time*1 (Operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.88 (0.667)*2	-	-	μs	Sampling in 40 states
	Offset error		-	±1.0	±5.5	LSB	-
	Full-scale error		-	±1.0	±5.5	LSB	-
	Absolute accuracy		-	±2.0	±7.5	LSB	-
	DNL differential nonlinearity error		-	±0.5	±4.5	LSB	-
	INL integral nonlinearity error		-	±1.0	±5.5	LSB	-

Note: These specification values apply when there is no access to the external bus during A/D conversion. If access occurs during A/D conversion, values might not fall within the indicated ranges.
The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.
The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage is stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

Table 60.42 A/D conversion characteristics for simultaneous using of channel-dedicated sample-and-hold circuits in unit0 and unit1

Conditions: PCLKC = 30/60 MHz

Parameter		Min	Typ	Max	Test conditions
Channel-dedicated sample-and-hold circuits in use with continious sampling function enabled (AN000 to AN002)	Offset error	-	±1.5	±5.0	• PCLKC = 60 MHz • Sampling in 15 states
	Full-scale error	-	±2.5	±5.0	
	Absolute accuracy	-	±4.0	±8.0	
Channel-dedicated sample-and-hold circuits in use with continious sampling function enabled (AN100 to AN102)	Offset error	-	±1.5	±5.0	
	Full-scale error	-	±2.5	±5.0	
	Absolute accuracy	-	±4.0	±8.0	
Channel-dedicated sample-and-hold circuits in use with continious sampling function enabled (AN000 to AN002)	Offset error	-	±1.5	±3.5	• PCLKC = 30 MHz • Sampling in 7 states
	Full-scale error	-	±1.5	±3.5	
	Absolute accuracy	-	±3.0	±5.5	
Channel-dedicated sample-and-hold circuits in use with continious sampling function enabled (AN100 to AN102)	Offset error	-	±1.5	±3.5	
	Full-scale error	-	±1.5	±3.5	
	Absolute accuracy	-	±3.0	±5.5	

Note: When simultaneously using channel-dedicated sample-and-hold circuits in unit0 and unit1, setting the ADSHMSR.SHMD bit to 1 is recommended.

Table 60.43 A/D internal reference voltage characteristics

Parameter	Min	Typ	Max	Unit	Test conditions
A/D internal reference voltage	1.13	1.18	1.23	V	-
Sampling time	4.15	-	-	μs	-

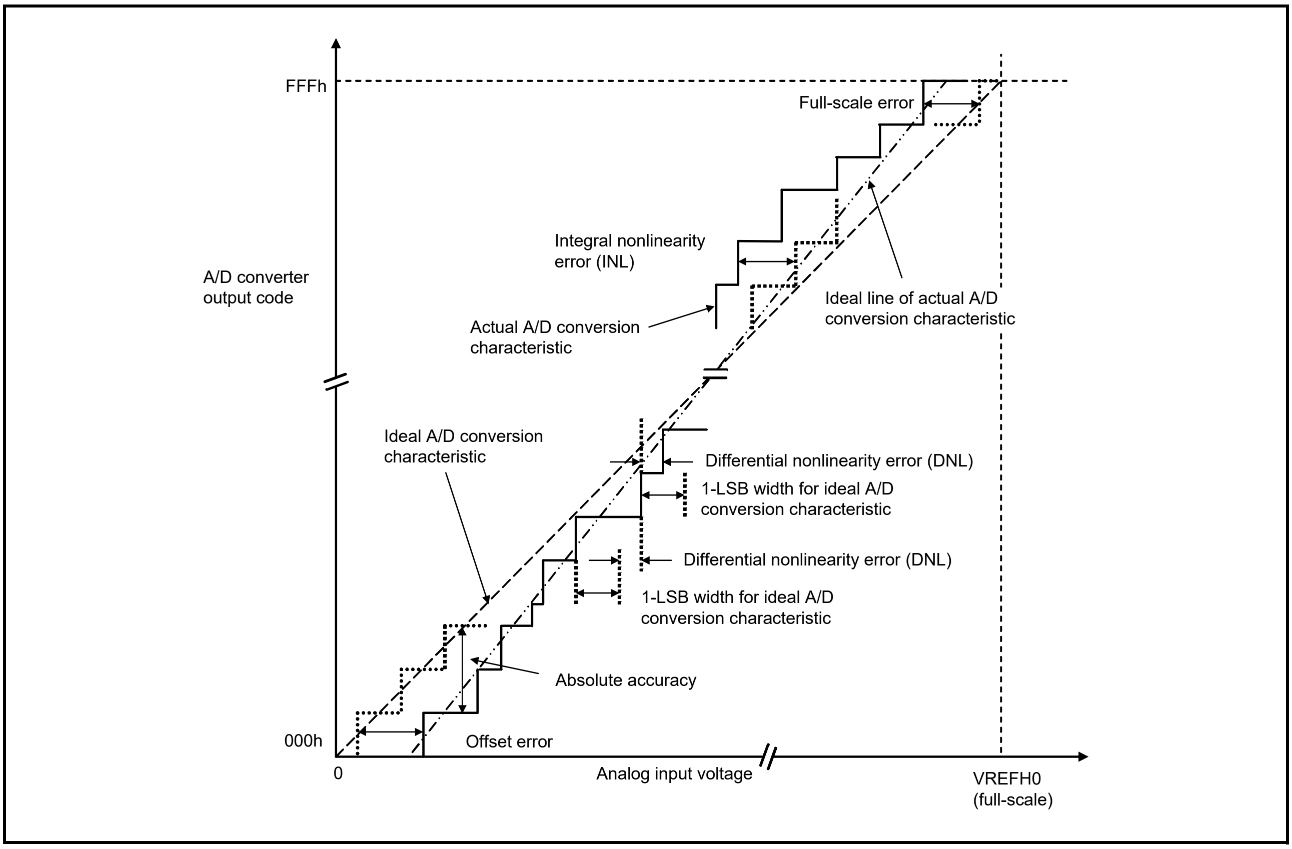


Figure 60.95 Illustration of ADC12 characteristic terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of the analog input voltage (1-LSB width), which can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and the reference voltage VREFH0 = 3.072 V, then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, and 1.5 mV are used as the analog input voltages. If the analog input voltage is 6 mV, an absolute accuracy of ±5 LSB means that the actual A/D conversion result is in the range of 003h to 00Dh, though an output code of 008h can be expected from the theoretical A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between the 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

Offset error

Offset error is the difference between the transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between the transition point of the ideal last output code and the actual last output code.

60.6 DAC12 Characteristics

Table 60.44 D/A conversion characteristics

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	-	-	12	Bits	-
Without output amplifier					
Absolute accuracy	-	-	±24	LSB	Resistive load 2 MΩ
INL	-	±2.0	±8.0	LSB	Resistive load 2 MΩ
DNL	-	±1.0	±2.0	LSB	-
Output impedance	-	8.5	-	kΩ	-
Conversion time	-	-	3.0	µs	Resistive load 2 MΩ, Capacitive load 20 pF
Output voltage range	0	-	VREFH	V	-
With output amplifier					
INL	-	±2.0	±4.0	LSB	-
DNL	-	±1.0	±2.0	LSB	-
Conversion time	-	-	4.0	µs	-
Resistive load	5	-	-	kΩ	-
Capacitive load	-	-	50	pF	-
Output voltage range	0.2	-	VREFH - 0.2	V	-

60.7 TSN Characteristics

Table 60.45 TSN characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Relative accuracy	-	-	±1.0	-	°C	-
Temperature slope	-	-	4.0	-	mV/°C	-
Output voltage (at 25°C)	-	-	1.24	-	V	-
Temperature sensor start time	t_{START}	-	-	30	µs	-
Sampling time	-	4.15	-	-	µs	-

60.8 OSC Stop Detect Characteristics

Table 60.46 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	-	-	1	ms	Figure 60.96

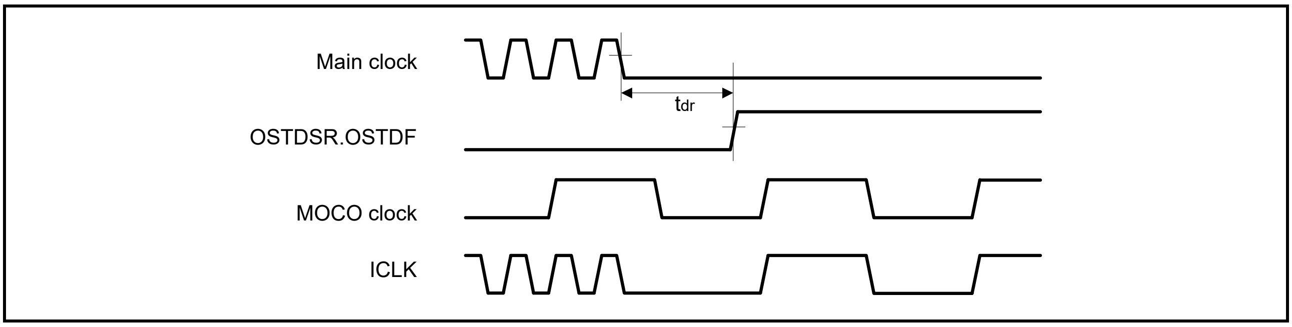


Figure 60.96 Oscillation stop detection timing

60.9 POR and LVD Characteristics

Table 60.47 Power-on reset circuit and voltage detection circuit characteristics

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Voltage detection level	Power-on reset (POR)	DPSBYCR.DEEPCUT[1:0] = 00b or 01b	V_{POR}	2.5	2.6	2.7	V	Figure 60.97
		DPSBYCR.DEEPCUT[1:0] = 11b		1.8	2.25	2.7		
	Voltage detection circuit (LVD0)		V_{det0_1}	2.84	2.94	3.04		Figure 60.98
			V_{det0_2}	2.77	2.87	2.97		
			V_{det0_3}	2.70	2.80	2.90		
	Voltage detection circuit (LVD1)		V_{det1_1}	2.89	2.99	3.09		Figure 60.99
			V_{det1_2}	2.82	2.92	3.02		
			V_{det1_3}	2.75	2.85	2.95		
	Voltage detection circuit (LVD2)		V_{det2_1}	2.89	2.99	3.09		Figure 60.100
			V_{det2_2}	2.82	2.92	3.02		
			V_{det2_3}	2.75	2.85	2.95		
Internal reset time	Power-on reset time		t_{POR}	-	4.5	-	ms	Figure 60.97
	LVD0 reset time		t_{LVD0}	-	0.51	-		Figure 60.98
	LVD1 reset time		t_{LVD1}	-	0.38	-		Figure 60.99
	LVD2 reset time		t_{LVD2}	-	0.38	-		Figure 60.100
Minimum VCC down time*1			t_{VOFF}	200	-	-	μs	Figure 60.97, Figure 60.98
Response delay			t_{det}	-	-	200	μs	Figure 60.97 to Figure 60.100
LVD operation stabilization time (after LVD is enabled)			$t_{d(E-A)}$	-	-	10	μs	Figure 60.99, Figure 60.100
Hysteresis width (LVD1 and LVD2)			V_{LVH}	-	70	-	mV	

Note 1. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det1}, and V_{det2} for POR and LVD.

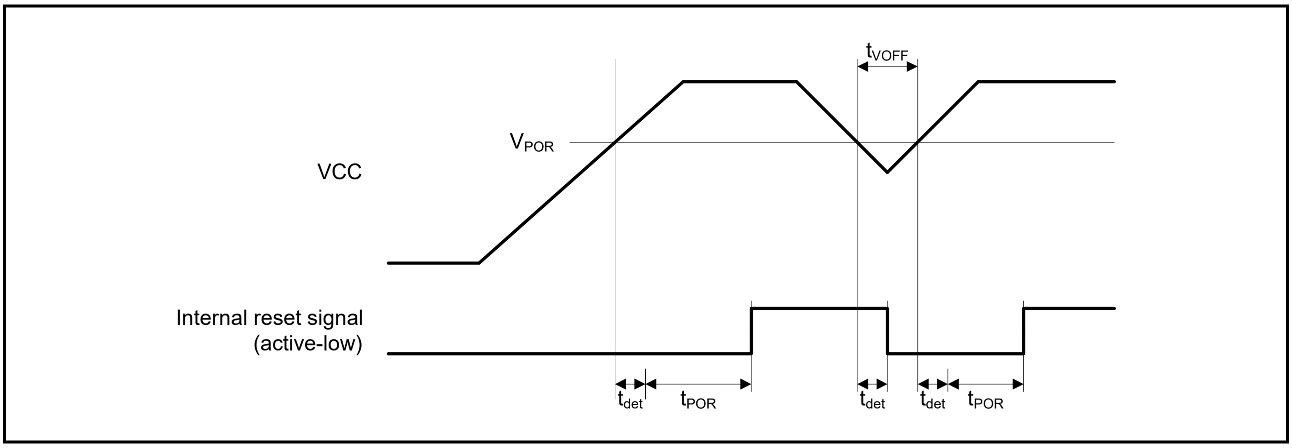


Figure 60.97 Power-on reset timing

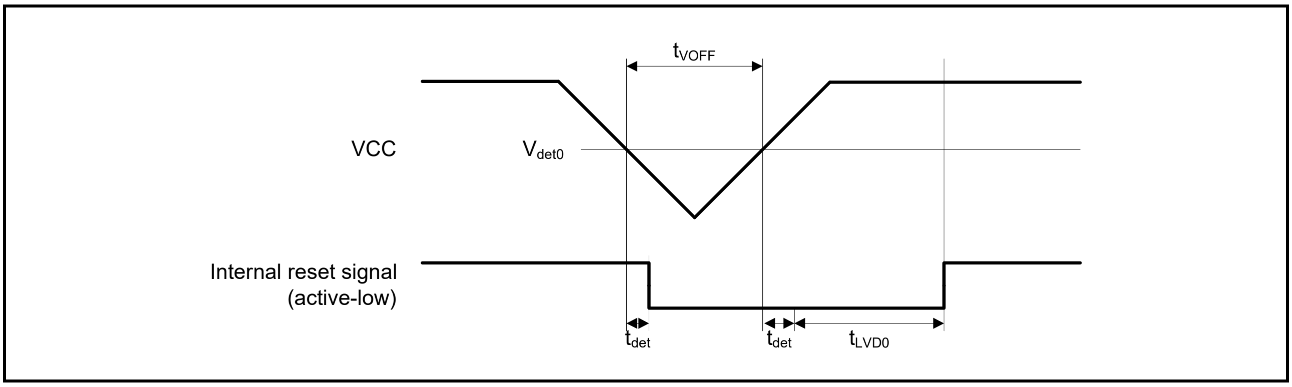


Figure 60.98 Voltage detection circuit timing (V_{det0})

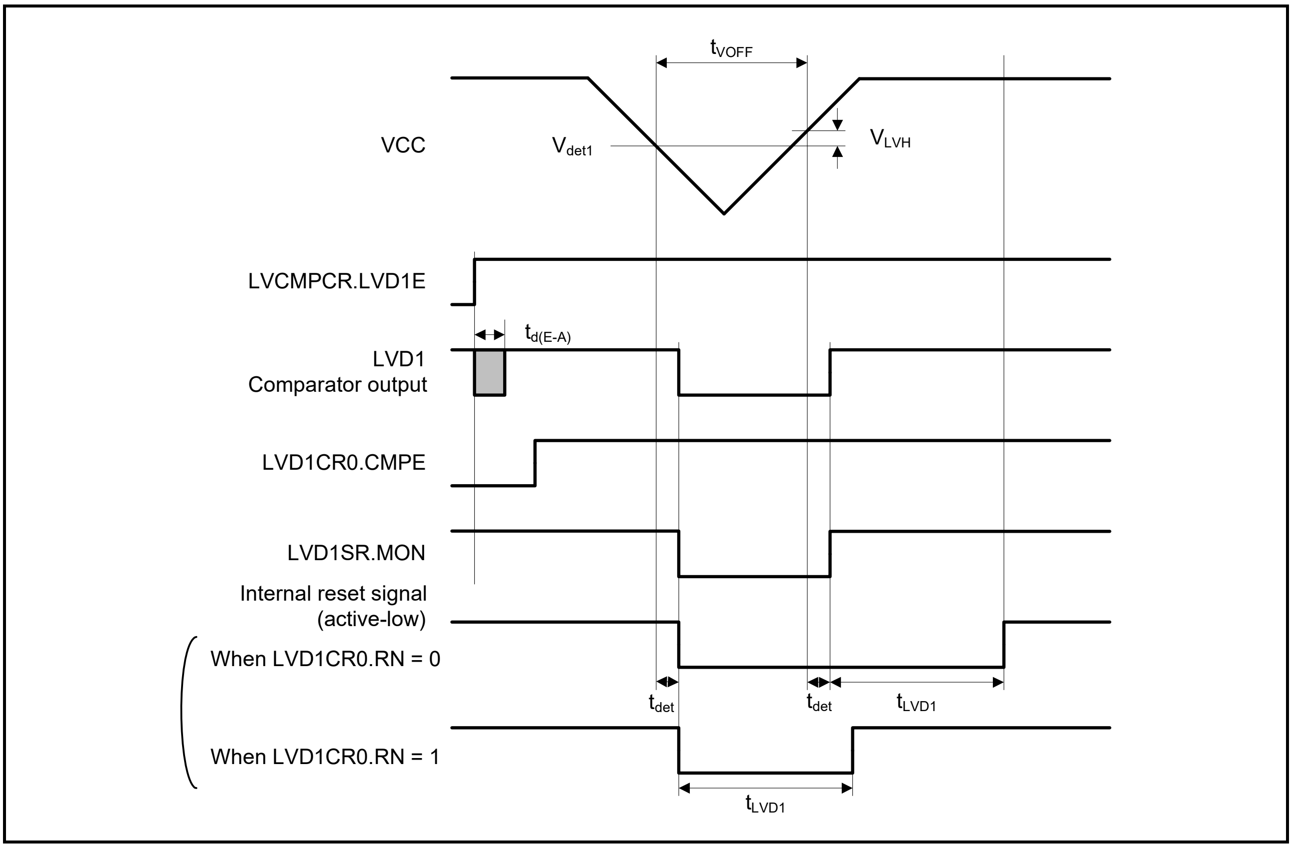


Figure 60.99 Voltage detection circuit timing (V_{det1})

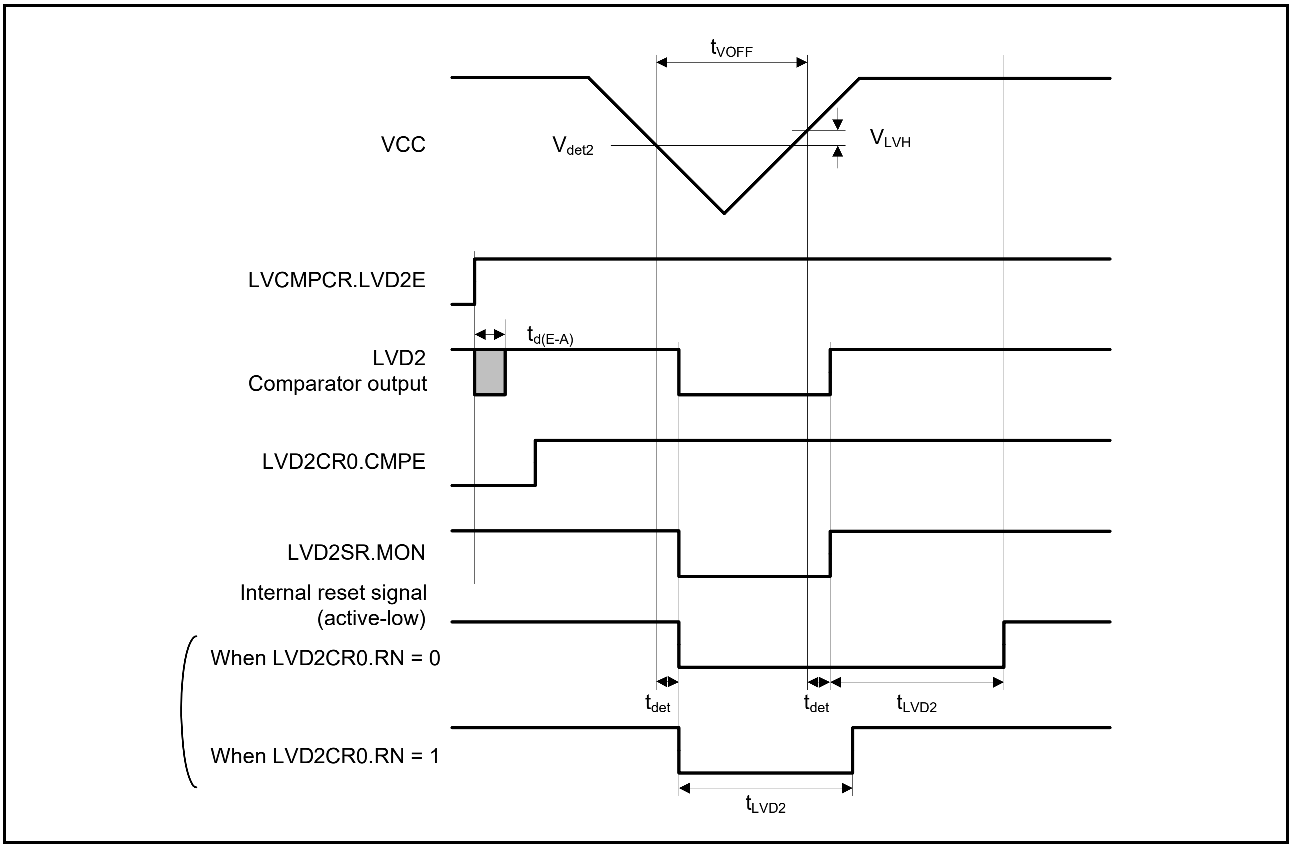


Figure 60.100 Voltage detection circuit timing (V_{det2})

60.10 VBATT Characteristics

Table 60.48 Battery backup function characteristics

Conditions: VCC = AVCC0 = VCC_USB = 2.7 to 3.6 V, 2.7 ≤ VREFH0/VREFH ≤ AVCC0, VBATT = 1.8 to 3.6 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Voltage level for switching to battery backup	$V_{DETBATT}$	2.50	2.60	2.70	V	Figure 60.101
Lower-limit VBATT voltage for power supply switching caused by VCC voltage drop	V_{BATTSW}	2.70	-	-	V	
VCC-off period for starting power supply switching	$t_{VOFFBATT}$	200	-	-	µs	

Note: The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup ($V_{DETBATT}$).

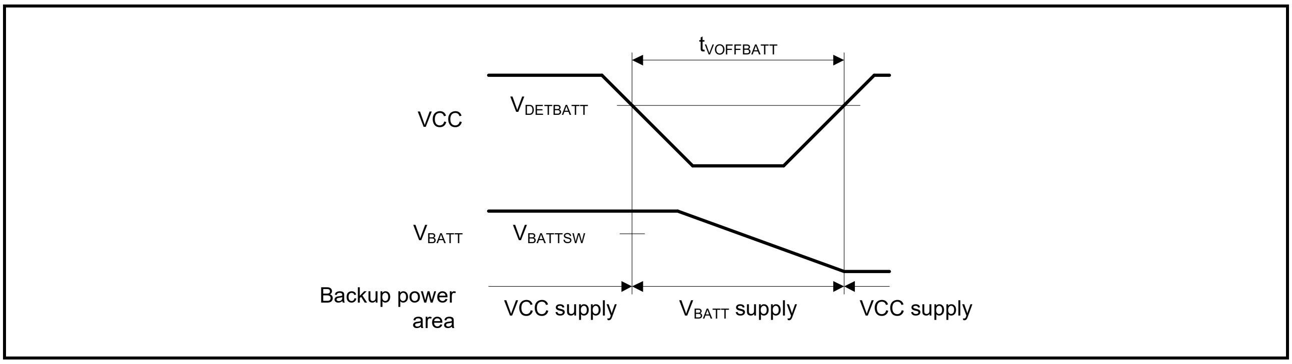


Figure 60.101 Battery backup function characteristics

60.11 CTSU Characteristics

Table 60.49 CTSU characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
External capacitance connected to TSCAP pin	C_{tscap}	9	10	11	nF	-
TS pin capacitive load	C_{base}	-	-	50	pF	-
Permissible output high current	Σ_{IoH}	-	-	-40	mA	When the mutual capacitance method is applied

60.12 ACMPHS Characteristics

Table 60.50 ACMPHS characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Reference voltage range	VREF	0	-	AVCC0	V	-
Input voltage range	VI	0	-	AVCC0	V	-
Output delay*1	Td	-	50	100	ns	VI = VREF ± 100 mV
Internal reference voltage	Vref	1.13	1.18	1.23	V	-

Note 1. This value is the internal propagation delay.

60.13 PGA Characteristics

Table 60.51 PGA characteristics in single mode

Parameter	Symbol	Min	Typ	Max	Unit
PGAVSS input voltage range	PGAVSS	0	-	0	V
	AIN0 (G = 2.000)	0.050 × AVCC0	-	0.45 × AVCC0	V
	AIN1 (G = 2.500)	0.047 × AVCC0	-	0.360 × AVCC0	V
	AIN2 (G = 2.667)	0.046 × AVCC0	-	0.337 × AVCC0	V
	AIN3 (G = 2.857)	0.046 × AVCC0	-	0.32 × AVCC0	V
	AIN4 (G = 3.077)	0.045 × AVCC0	-	0.292 × AVCC0	V
	AIN5 (G = 3.333)	0.044 × AVCC0	-	0.265 × AVCC0	V
	AIN6 (G = 3.636)	0.042 × AVCC0	-	0.247 × AVCC0	V
	AIN7 (G = 4.000)	0.040 × AVCC0	-	0.212 × AVCC0	V
	AIN8 (G = 4.444)	0.036 × AVCC0	-	0.191 × AVCC0	V
	AIN9 (G = 5.000)	0.033 × AVCC0	-	0.17 × AVCC0	V
	AIN10 (G = 5.714)	0.031 × AVCC0	-	0.148 × AVCC0	V
	AIN11 (G = 6.667)	0.029 × AVCC0	-	0.127 × AVCC0	V
	AIN12 (G = 8.000)	0.027 × AVCC0	-	0.09 × AVCC0	V
	AIN13 (G = 10.000)	0.025 × AVCC0	-	0.08 × AVCC0	V
	AIN14 (G = 13.333)	0.023 × AVCC0	-	0.06 × AVCC0	V
Gain error	Gerr0 (G = 2.000)	-1.0	-	1.0	%
	Gerr1 (G = 2.500)	-1.0	-	1.0	%
	Gerr2 (G = 2.667)	-1.0	-	1.0	%
	Gerr3 (G = 2.857)	-1.0	-	1.0	%
	Gerr4 (G = 3.077)	-1.0	-	1.0	%
	Gerr5 (G = 3.333)	-1.5	-	1.5	%
	Gerr6 (G = 3.636)	-1.5	-	1.5	%
	Gerr7 (G = 4.000)	-1.5	-	1.5	%
	Gerr8 (G = 4.444)	-2.0	-	2.0	%
	Gerr9 (G = 5.000)	-2.0	-	2.0	%
	Gerr10 (G = 5.714)	-2.0	-	2.0	%
	Gerr11 (G = 6.667)	-2.0	-	2.0	%
	Gerr12 (G = 8.000)	-2.0	-	2.0	%
	Gerr13 (G = 10.000)	-2.0	-	2.0	%
	Gerr14 (G = 13.333)	-2.0	-	2.0	%
Offset error	Voff	-8	-	8	mV

Table 60.52 PGA characteristics in differential mode (1 of 2)

Parameter		Symbol	Min	Typ	Max	Unit
PGAVSS input voltage range		PGAVSS	-0.5	-	0.3	V
Differential input voltage range	G = 1.500	AIN-PGAVSS	-0.5	-	0.5	V
	G = 2.333		-0.4	-	0.4	V
	G = 4.000		-0.2	-	0.2	V
	G = 5.667		-0.15	-	0.15	V

Table 60.52 PGA characteristics in differential mode (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit
Gain error	G = 1.500	Gerr	-1.0	-	1.0	%
	G = 2.333		-1.0	-	1.0	
	G = 4.000		-1.0	-	1.0	
	G = 5.667		-1.0	-	1.0	

60.14 Flash Memory Characteristics

60.14.1 Code Flash Memory Characteristics

Table 60.53 Code flash memory characteristics

Conditions: Program or erase: FCLK = 4 to 60 MHz

Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ	Max	Min	Typ	Max		
Programming time N_{PEC} ≤ 100 times	128-byte	t_{P128}	-	0.75	13.2	-	0.34	6.0	ms	
	8-KB	t_{P8K}	-	49	176	-	22	80	ms	
	32-KB	t_{P32K}	-	194	704	-	88	320	ms	
Programming time N_{PEC} > 100 times	128-byte	t_{P128}	-	0.91	15.8	-	0.41	7.2	ms	
	8-KB	t_{P8K}	-	60	212	-	27	96	ms	
	32-KB	t_{P32K}	-	234	848	-	106	384	ms	
Erasure time N_{PEC} ≤ 100 times	8-KB	t_{E8K}	-	78	216	-	43	120	ms	
	32-KB	t_{E32K}	-	283	864	-	157	480	ms	
Erasure time N_{PEC} > 100 times	8-KB	t_{E8K}	-	94	260	-	52	144	ms	
	32-KB	t_{E32K}	-	341	1040	-	189	576	ms	
Reprogramming/erasure cycle*Note:		N_{PEC}	10000*1	-	-	10000*1	-	-	Times	
Suspend delay during programming		t_{SPD}	-	-	264	-	-	120	μs	
First suspend delay during erasure in suspend priority mode		t_{SESD1}	-	-	216	-	-	120	μs	
Second suspend delay during erasure in suspend priority mode		t_{SESD2}	-	-	1.7	-	-	1.7	ms	
Suspend delay during erasure in erasure priority mode		t_{SEED}	-	-	1.7	-	-	1.7	ms	
Forced stop command		t_{FD}	-	-	32	-	-	20	μs	
Data hold time*2		t_{DRP}	10*2, *3	-	-	10*2, *3	-	-	Years	
			30*2, *3	-	-	30*2, *3	-	-		Ta = +85°C

Note: The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 10,000), erasing can be performed n times for each block. For example, when 128-byte programming is performed 64 times for different addresses in 8-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. (Overwriting is prohibited.)

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.

Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.

Note 3. This result is obtained from reliability testing.

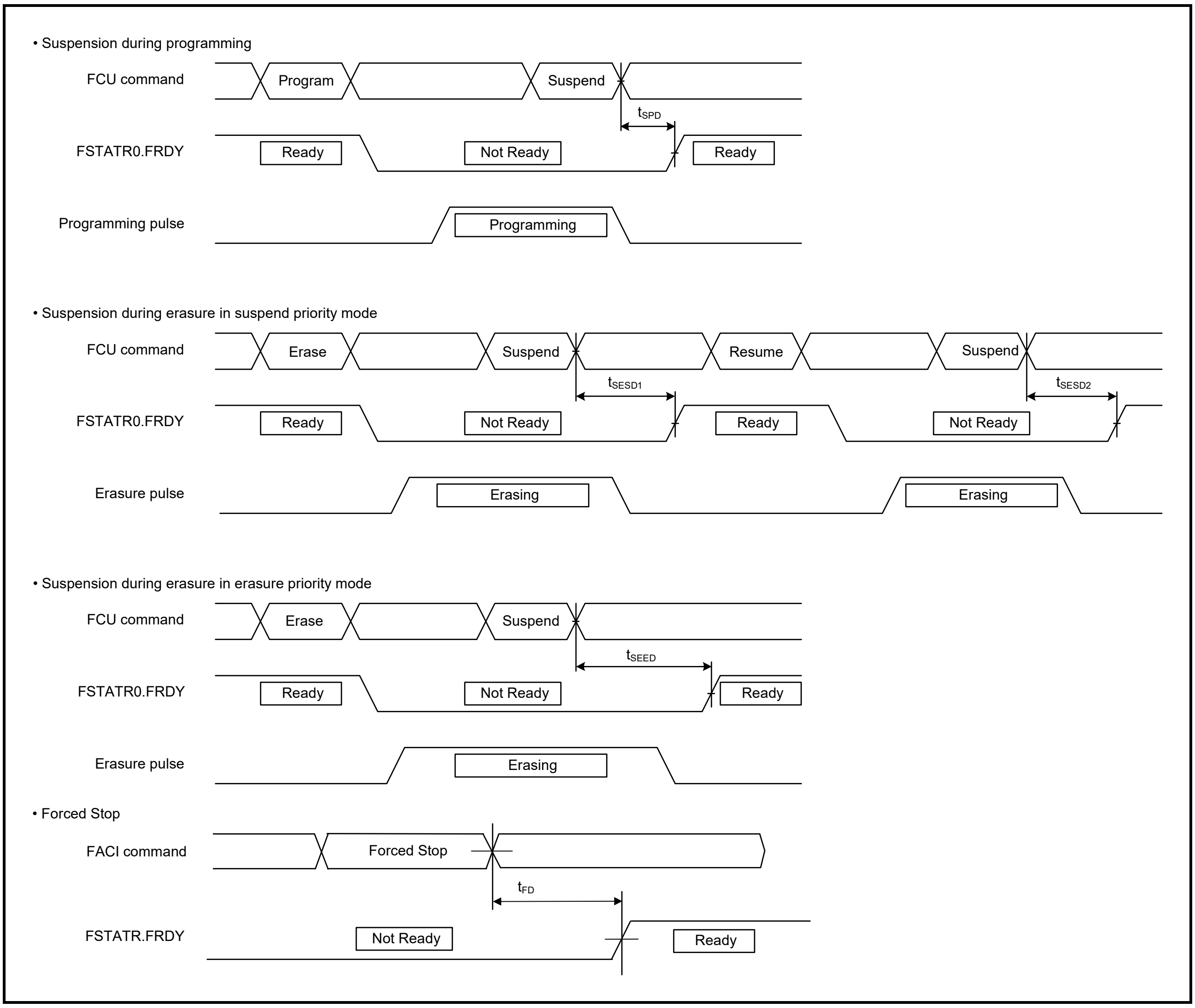


Figure 60.102 Suspension and forced stop timing for flash memory programming and erasure

60.14.2 Data Flash Memory Characteristics

Table 60.54 Data flash memory characteristics (1 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ	Max	Min	Typ	Max		
Programming time	4-byte	t_{DP4}	-	0.36	3.8	-	0.16	1.7	ms	
	8-byte	t_{DP8}	-	0.38	4.0	-	0.17	1.8		
	16-byte	t_{DP16}	-	0.42	4.5	-	0.19	2.0		
Erasure time	64-byte	t_{DE64}	-	3.1	18	-	1.7	10	ms	
	128-byte	t_{DE128}	-	4.7	27	-	2.6	15		
	256-byte	t_{DE256}	-	8.9	50	-	4.9	28		
Blank check time	4-byte	t_{DBC4}	-	-	84	-	-	30	μs	
Reprogramming/erasure cycle*1		N_{DPEC}	125000 *2	-	-	125000 *2	-	-	-	

Table 60.54 Data flash memory characteristics (2 of 2)
Conditions: Program or erase: FCLK = 4 to 60 MHz
Read: FCLK ≤ 60 MHz

Parameter		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
			Min	Typ	Max	Min	Typ	Max		
Suspend delay during programming	4-byte	t_{DSPD}	-	-	264	-	-	120	µs	
	8-byte		-	-	264	-	-	120		
	16-byte		-	-	264	-	-	120		
First suspend delay during erasure in suspend priority mode	64-byte	t_{DSESD1}	-	-	216	-	-	120	µs	
	128-byte		-	-	216	-	-	120		
	256-byte		-	-	216	-	-	120		
Second suspend delay during erasure in suspend priority mode	64-byte	t_{DSESD2}	-	-	300	-	-	300	µs	
	128-byte		-	-	390	-	-	390		
	256-byte		-	-	570	-	-	570		
Suspend delay during erasing in erasure priority mode	64-byte	t_{DSEED}	-	-	300	-	-	300	µs	
	128-byte		-	-	390	-	-	390		
	256-byte		-	-	570	-	-	570		
Forced stop command		t_{FD}	-	-	32	-	-	20	µs	
Data hold time*3		t_{DRP}	10*3,*4	-	-	10*3,*4	-	-	Year	
			30*3,*4	-	-	30*3,*4	-	-		Ta = +85°C

Note 1. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 125,000), erasing can be performed n times for each block. For example, when 4-byte programming is performed 16 times for different addresses in 64-byte blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. (Overwriting is prohibited.)

Note 2. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.

Note 3. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.

Note 4. This result is obtained from reliability testing.

60.15 Boundary Scan

Table 60.55 Boundary scan characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	100	-	-	ns	Figure 60.103
TCK clock high pulse width	t_{TCKH}	45	-	-	ns	
TCK clock low pulse width	t_{TCKL}	45	-	-	ns	
TCK clock rise time	t_{TCKr}	-	-	5	ns	
TCK clock fall time	t_{TCKf}	-	-	5	ns	
TMS setup time	t_{TMSS}	20	-	-	ns	Figure 60.104
TMS hold time	t_{TMSH}	20	-	-	ns	
TDI setup time	t_{TDIS}	20	-	-	ns	
TDI hold time	t_{TDIH}	20	-	-	ns	
TDO data delay	t_{TDOD}	-	-	40	ns	
Boundary scan circuit startup time*1	T_{BSSTUP}	t_{RESWP}	-	-	-	Figure 60.105

Note 1. Boundary scan does not function until the power-on reset becomes negative.

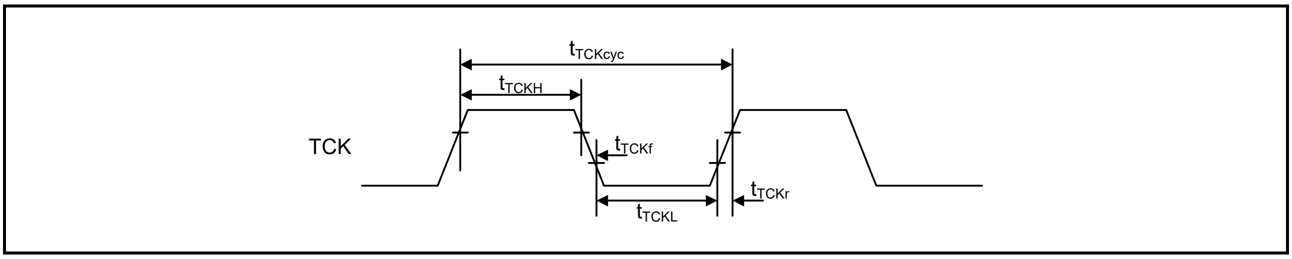


Figure 60.103 Boundary scan TCK timing

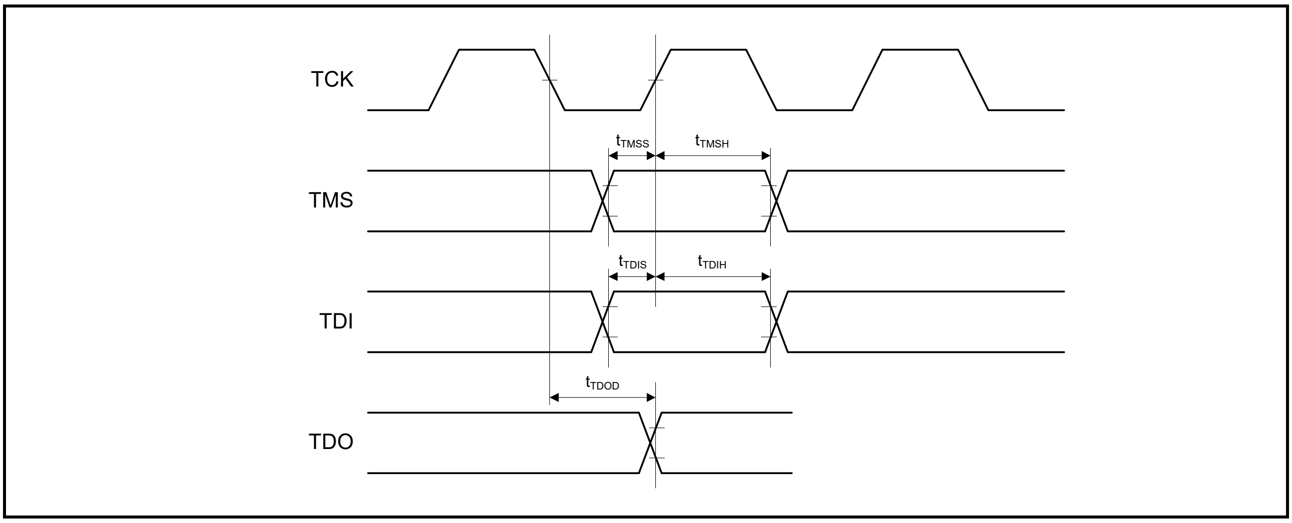


Figure 60.104 Boundary scan input/output timing

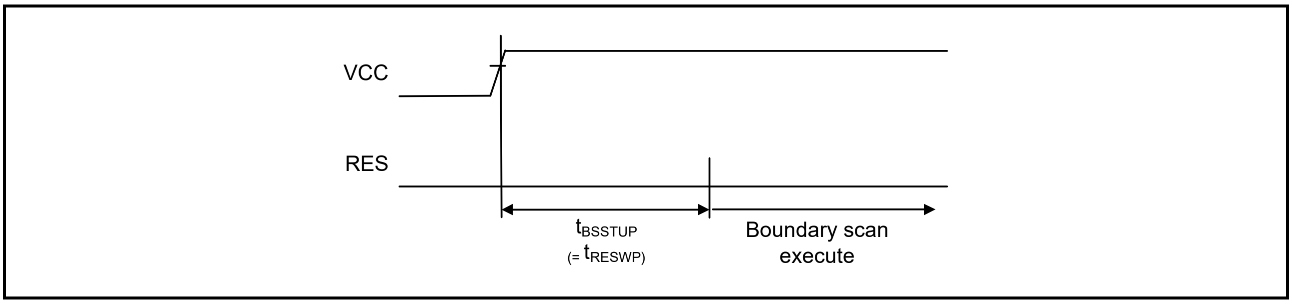


Figure 60.105 Boundary scan circuit startup timing

60.16 Joint Test Action Group (JTAG)

Table 60.56 JTAG

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	40	-	-	ns	Figure 60.103
TCK clock high pulse width	t_{TCKH}	15	-	-	ns	
TCK clock low pulse width	t_{TCKL}	15	-	-	ns	
TCK clock rise time	t_{TCKr}	-	-	5	ns	
TCK clock fall time	t_{TCKf}	-	-	5	ns	

Table 60.56 JTAG

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TMS setup time	t_{TMSS}	8	-	-	ns	Figure 60.104
TMS hold time	t_{TMSH}	8	-	-	ns	
TDI setup time	t_{TDIS}	8	-	-	ns	
TDI hold time	t_{TDIH}	8	-	-	ns	
TDO data delay time	t_{TDOD}	-	-	20	ns	

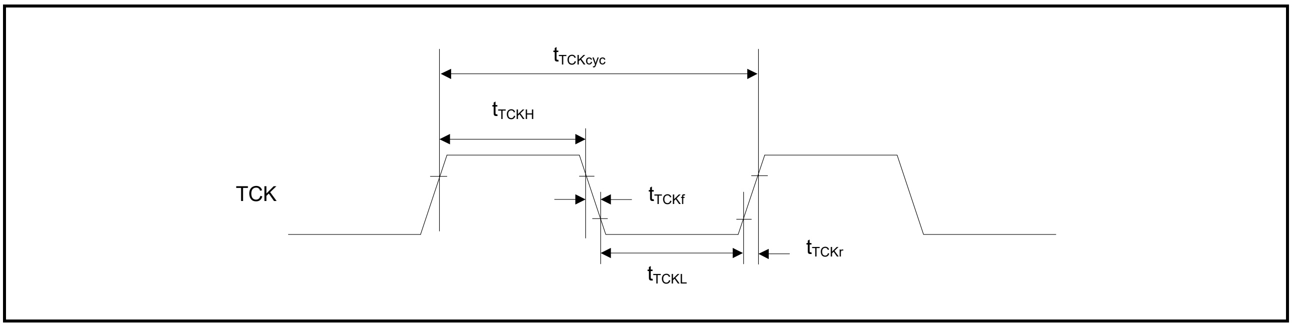


Figure 60.106 JTAG TCK timing

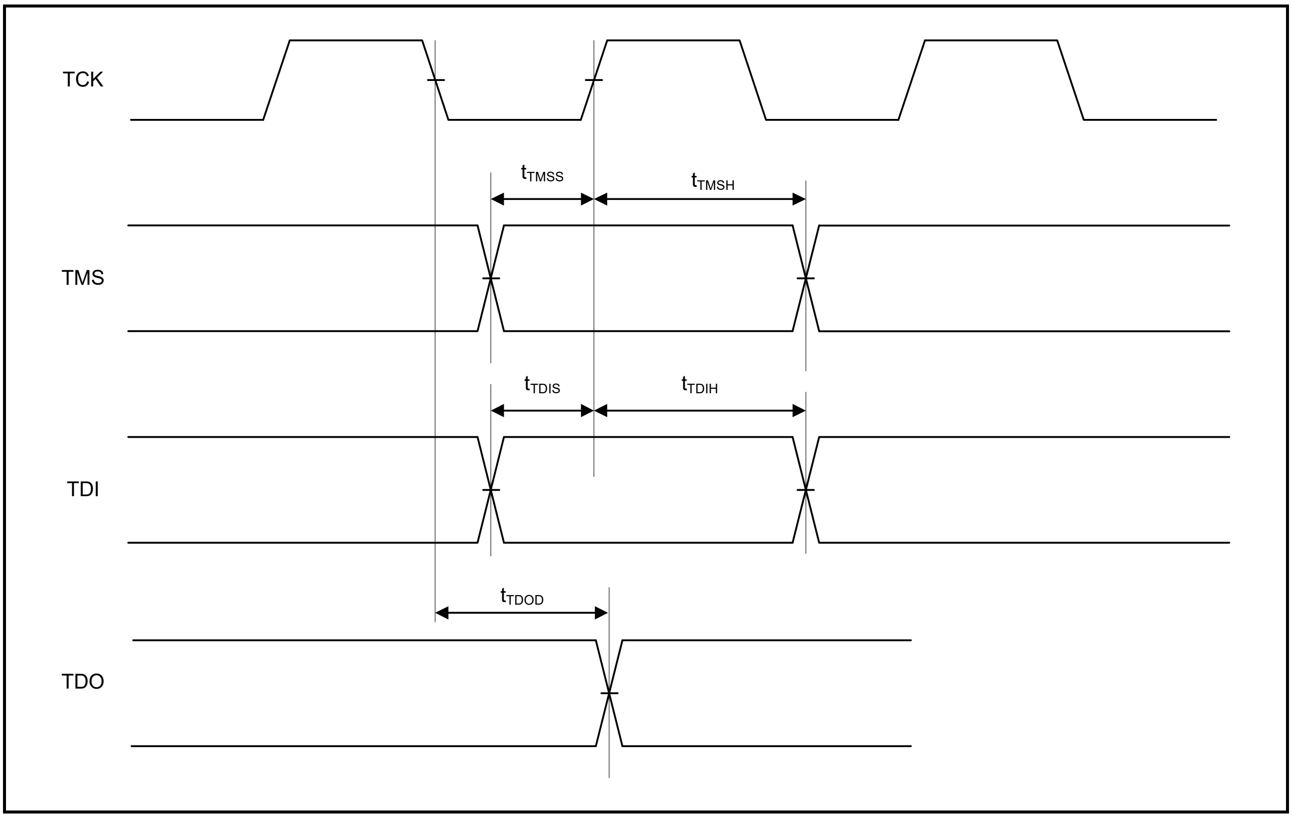


Figure 60.107 JTAG input/output timing

60.17 Serial Wire Debug (SWD)

Table 60.57 SWD

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	40	-	-	ns	Figure 60.108
SWCLK clock high pulse width	t_{SWCKH}	15	-	-	ns	
SWCLK clock low pulse width	t_{SWCKL}	15	-	-	ns	
SWCLK clock rise time	t_{SWCKr}	-	-	5	ns	
SWCLK clock fall time	t_{SWCKf}	-	-	5	ns	
SWDIO setup time	t_{SWDS}	8	-	-	ns	Figure 60.109
SWDIO hold time	t_{SWDH}	8	-	-	ns	
SWDIO data delay time	t_{SWDD}	2	-	28	ns	

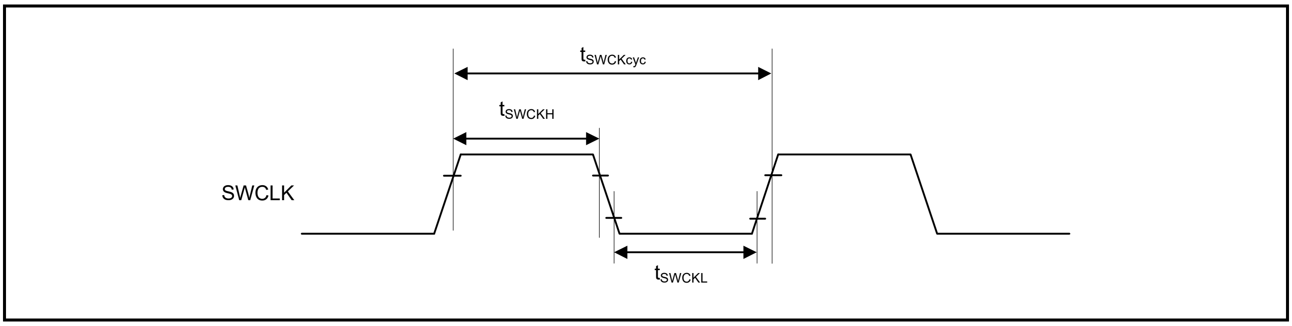


Figure 60.108 SWD SWCLK timing

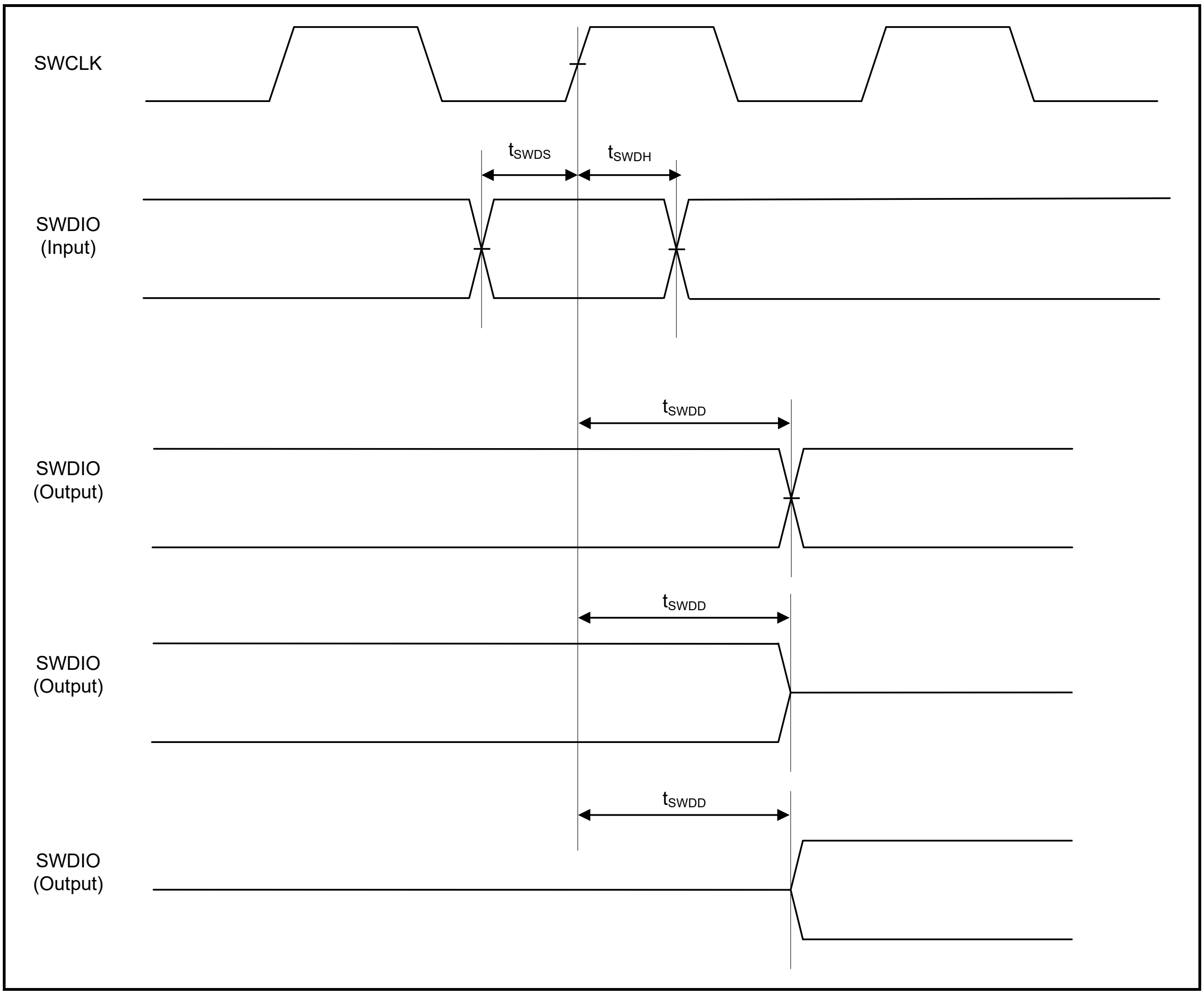


Figure 60.109 SWD input/output timing

60.18 Embedded Trace Macro Interface (ETM)

Table 60.58 ETM

Conditions: High drive output is selected in the port drive capability bit in the PmnPFS register.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCLK clock cycle time	$t_{TCLKcyc}$	33.3	-	-	ns	Figure 60.110
TCLK clock high pulse width	t_{TCLKH}	13.6	-	-	ns	
TCLK clock low pulse width	t_{TCLKL}	13.6	-	-	ns	
TCLK clock rise time	t_{TCLKr}	-	-	3	ns	
TCLK clock fall time	t_{TCLKf}	-	-	3	ns	
TDATA[3:0] output setup time	t_{TRDS}	3.5	-	-	ns	Figure 60.111
TDATA[3:0] output hold time	t_{TRDH}	2.5	-	-	ns	

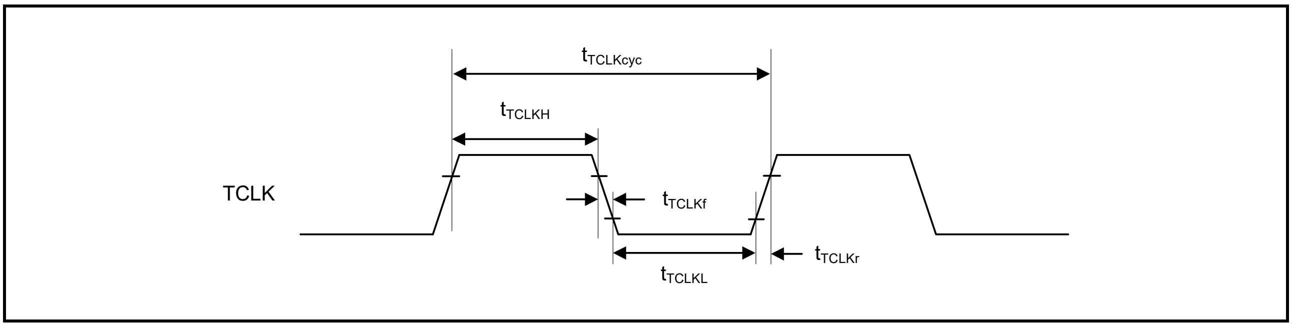


Figure 60.110 ETM TCLK timing

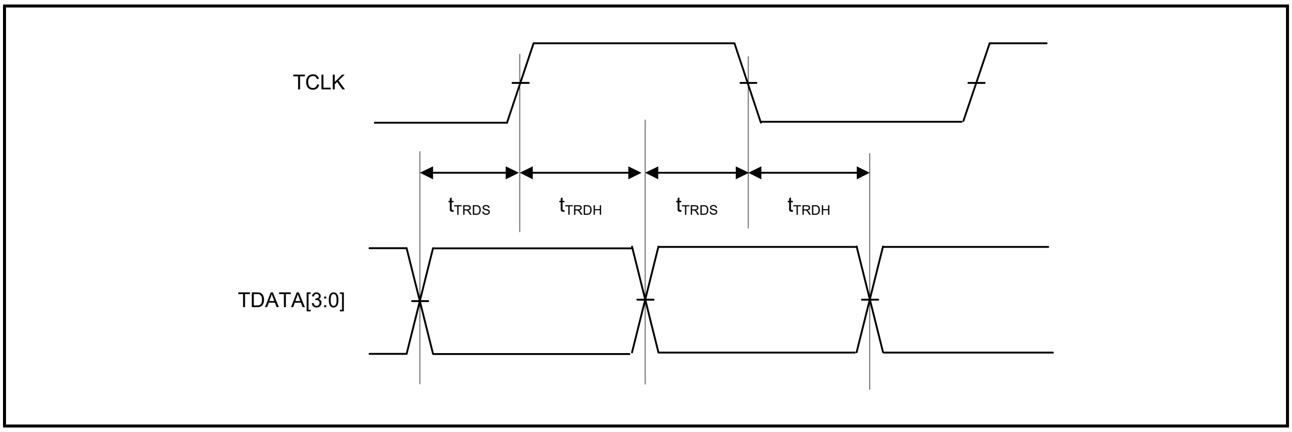


Figure 60.111 ETM output timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port states in each processing state (1 of 6)

Port name	Reset	Software Standby mode		Deep Software Standby mode	After Deep Software Standby mode is canceled (return to startup mode)	
		OPE = 0	OPE = 1		IOKEEP = 0	IOKEEP = 1*1
P000/IRQ6-DS, P001/IRQ7-DS, P002/IRQ8-DS	Hi-Z	Hi-Z*2		Keep-O*3	Hi-Z	Keep
P003	Hi-Z	Hi-Z		Keep	Hi-Z	Keep
P004/IRQ9-DS, P005/IRQ10-DS, P006/IRQ11-DS	Hi-Z	Hi-Z*2		Keep-O*3	Hi-Z	Keep
P007	Hi-Z	Hi-Z		Keep	Hi-Z	Keep
P008/IRQ12-DS, P009/IRQ13-DS, P010/IRQ14-DS	Hi-Z	Keep-O*2		Keep-O*3	Hi-Z	Keep
P014/DA0	Hi-Z	[DA0 output (DAOE0 = 1)] D/A output retained [All other (DAOE0 = 0)] Keep-O		Keep	Hi-Z	Keep
P015/IRQ13/DA1	Hi-Z	[DA1 output (DAOE1 = 1)] D/A output retained [All other (DAOE1 = 0)] Keep-O*2		Keep	Hi-Z	Keep
P100/D00[A00/D00]/ DQ00/KR00/AGTIO0/ RXD0/IRQ2	Hi-Z	[D00 output] Hi-Z [DQ00 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P101/D01[A01/D01]/ DQ01/KR01/IRQ1	Hi-Z	[D01 output] Hi-Z [DQ01 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P102/D02[A02/D02]/ DQ02/KR02	Hi-Z	[D02 output] Hi-Z [DQ02 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P103/D03[A03/D03]/ DQ03/KR03	Hi-Z	[D03 output] Hi-Z [DQ03 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P104/D04[A04/D04]/ DQ04/KR04/IRQ1	Hi-Z	[D04 output] Hi-Z [DQ04 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P105/D05[A05/D05]/ DQ05/KR05/IRQ0	Hi-Z	[D05 output] Hi-Z [DQ05 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P106/D06[A06/D06]/ DQ06/KR06	Hi-Z	[D06 output] Hi-Z [DQ06 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P107/D07[A07/D07]/ DQ07/KR07	Hi-Z	[D07 output] Hi-Z [DQ07 output] Hi-Z [All other] Keep-O*2		Keep	Hi-Z	Keep
P108/TMS	Pull-up	Keep-O		Keep	Pull-up	Keep

Table 1.1 Port states in each processing state (2 of 6)

Port name	Reset	Software Standby mode		Deep Software Standby mode	After Deep Software Standby mode is canceled (return to startup mode)	
		OPE = 0	OPE = 1		IOKEEP = 0	IOKEEP = 1[*1]
P109/TDO/CLKOUT	TDO output	[CLKOUT selected] CLKOUT output [All other] Keep-O		[TDO output] TDO output retained [All other] Keep	[TDO output] TDO output retained [All other] Hi-Z	[TDO output] TDO output retained [All other] Keep
P110/IRQ3/TDI/VCOUT	Pull-up	[ACMPHS selected] VCOUT output [All other] Keep-O[*2]		Keep	Pull-up	Keep
P111/A05/IRQ4	Hi-Z	[A05 output] Hi-Z [All other] Keep-O[*2]	[A05 output] Address output retained [All other] Keep-O[*2]	Keep	Hi-Z	Keep
P112/A04	Hi-Z	[A04 output] Hi-Z [All other] Keep-O	[A04 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P113/A03	Hi-Z	[A03 output] Hi-Z [All other] Keep-O	[A03 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P114/A02	Hi-Z	[A02 output] Hi-Z [All other] Keep-O	[A02 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P115/A01	Hi-Z	[A01 output] Hi-Z [All other] Keep-O	[A01 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P200/NMI	Hi-Z	Hi-Z		Keep	Hi-Z	Keep
P201	Pull-up	Keep-O		Keep	Pull-up	Keep
P202/WR1/BC1/IRQ3-DS	Hi-Z	[WR1/BC1 output] Hi-Z [All other] Keep-O[*2]	[WR1/BC1 output] H [All other] Keep-O[*2]	Keep-O[*3]	Hi-Z	Keep
P203/A19/IRQ2-DS	Hi-Z	[A19 output] Hi-Z [All other] Keep-O[*2]	[A19 output] Address output retained [All other] Keep-O[*2]	Keep-O[*3]	Hi-Z	Keep
P204/A18/AGTIO1/SCL0_B/USB_OVRCURB-DS	Hi-Z	[A18 output] Hi-Z [All other] Keep-O[*2]	[A18 output] Address output retained [All other] Keep-O[*2]	Keep-O[*3]	Hi-Z	Keep
P205/A16/USB_OVRCURA-DS/CLKOUT/IRQ1-DS	Hi-Z	[A16 output] Hi-Z [CLKOUT selected] CLKOUT output [All other] Keep-O[*2]	[A16 output] Address output retained [CLKOUT selected] CLKOUT output [All other] Keep-O[*2]	Keep-O[*3]	Hi-Z	Keep
P206/WAIT/IRQ0-DS	Hi-Z	Keep-O[*2]		Keep-O[*3]	Hi-Z	Keep
P207/A17	Hi-Z	[A17 output] Hi-Z [All other] Keep-O	[A17 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P208 to P211	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P212/IRQ3/EXTAL, P213/IRQ2/XTAL	Hi-Z	Keep-O[*2]		Keep	Hi-Z	Keep
P214	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P300/TCK	Pull-up	Keep-O		Keep	Pull-up	Keep
P301/A06/AGTIO0/IRQ6	Hi-Z	[A06 output] Hi-Z [All other] Keep-O[*2]	[A06 output] Address output retained [All other] Keep-O[*2]	Keep	Hi-Z	Keep
P302/A07/IRQ5	Hi-Z	[A07 output] Hi-Z [All other] Keep-O[*2]	[A07 output] Address output retained [All other] Keep-O[*2]	Keep	Hi-Z	Keep
P303/A08	Hi-Z	[A08 output] Hi-Z [All other] Keep-O	[A08 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P304/A09/IRQ9	Hi-Z	[A09 output] Hi-Z [All other] Keep-O[*2]	[A09 output] Address output retained [All other] Keep-O[*2]	Keep	Hi-Z	Keep

Table 1.1 Port states in each processing state (3 of 6)

Port name	Reset	Software Standby mode		Deep Software Standby mode	After Deep Software Standby mode is canceled (return to startup mode)	
		OPE = 0	OPE = 1		IOKEEP = 0	IOKEEP = 1*1
P305/A10/IRQ8	Hi-Z	[A10 output] Hi-Z [All other] Keep-O*2	[A10 output] Address output retained [All other] Keep-O*2	Keep	Hi-Z	Keep
P306/A11	Hi-Z	[A11 output] Hi-Z [All other] Keep-O	[A11 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P307/A12	Hi-Z	[A12 output] Hi-Z [All other] Keep-O	[A12 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P308/A13	Hi-Z	[A13 output] Hi-Z [All other] Keep-O	[A13 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P309/A14	Hi-Z	[A14 output] Hi-Z [All other] Keep-O	[A14 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P310/A15	Hi-Z	[A15 output] Hi-Z [All other] Keep-O	[A15 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P311/CS2/RAS	Hi-Z	[CS2 output] Hi-Z [RAS output] Hi-Z [All other] Keep-O	[CS2 output] H [RAS output] SDSELF.SFEN = 0: H SDSELF.SFEN = 1: L [All other] Keep-O	Keep	Hi-Z	Keep
P312/CS3/CAS	Hi-Z	[CS3 output] Hi-Z [CAS output] Hi-Z [All other] Keep-O	[CS3 output] H [CAS output] SDSELF.SFEN = 0: H SDSELF.SFEN = 1: L [All other] Keep-O	Keep	Hi-Z	Keep
P313/A20	Hi-Z	[A20 output] Hi-Z [All other] Keep-O	[A20 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P314/A21	Hi-Z	[A21 output] Hi-Z [All other] Keep-O	[A21 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P315/A22	Hi-Z	[A22 output] Hi-Z [All other] Keep-O	[A22 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P400/AGTIO1/ SCL0_A/IRQ0	Hi-Z	Keep-O*2		Keep	Hi-Z	Keep
P401/SDA0_A/ IRQ5-DS, P402/IRQ4-DS/ RTCIC0/ AGTIO0/AGTIO1, P403/RTCIC1/ AGTIO0/AGTIO1, P404/RTCIC2	Hi-Z	Keep-O*2		Keep-O*3	Hi-Z	Keep
P405, P406	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P407/AGTIO0/ SDA0_B/USB_VBUS/ RTCOUT	Hi-Z	[RTCOUT selected] RTCOUT output [All other] Keep-O*2		Keep-O*3	Hi-Z	Keep
P408/SCL0_C/IRQ7, P409/IRQ6, P410/RXD0/IRQ5, P411/IRQ4	Hi-Z	Keep-O*2		Keep	Hi-Z	Keep
P412, P413	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P414/IRQ9, P415/IRQ8	Hi-Z	Keep-O*2		Keep	Hi-Z	Keep
P500	Hi-Z	Keep-O		Keep	Hi-Z	Keep

Table 1.1 Port states in each processing state (4 of 6)

Port name	Reset	Software Standby mode		Deep Software Standby mode	After Deep Software Standby mode is canceled (return to startup mode)	
		OPE = 0	OPE = 1		IOKEEP = 0	IOKEEP = 1[*1]
P501/ USB_OVRCURA/ IRQ11, P502/ USB_OVRCURB/ IRQ12	Hi-Z	Keep-O[*2]		Keep	Hi-Z	Keep
P503	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P504/ALE	Hi-Z	[ALE output] Hi-Z [All other] Keep-O	[ALE output] L [All other] Keep-O	Keep	Hi-Z	Keep
P505/IRQ14, P506/IRQ15	Hi-Z	Keep-O[*2]		Keep	Hi-Z	Keep
P507, P508	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P511/IRQ15, P512/IRQ14	Hi-Z	Keep-O[*2]		Keep	Hi-Z	Keep
P513	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P600/RD/CLKOUT	Hi-Z	[RD output] Hi-Z [CLKOUT selected] CLKOUT output [All other] Keep-O	[RD output] H [CLKOUT selected] CLKOUT output [All other] Keep-O	Keep	Hi-Z	Keep
P601/WR0/WR/DQM0	Hi-Z	[WR0/WR output] Hi-Z [DQM0 output] Hi-Z [All other] Keep-O	[WR0/WR output] H [DQM0 output] DQM0 output retained [All other] Keep-O	Keep	Hi-Z	Keep
P602/EBCLK/SDCLK	Hi-Z	[EBCLK output] H [SDCLK output] H [All other] Keep-O		Keep	Hi-Z	Keep
P603/D13[A13/D13]/ DQ13	Hi-Z	[D13 output] Hi-Z [DQ13 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P604/D12[A12/D12]/ DQ12	Hi-Z	[D12 output] Hi-Z [DQ12 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P605/D11[A11/D11]/ DQ11	Hi-Z	[D11 output] Hi-Z [DQ11 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P606, P607	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P608/A00/BC0/DQM1	Hi-Z	[A00 output] Hi-Z [BC0 output] Hi-Z [DQM1 output] Hi-Z [All other] Keep-O	[A00 output] Address output retained [BC0 output] H [DQM1 output] DQM1 output retained [All other] Keep-O	Keep	Hi-Z	Keep
P609/CS1/CKE	Hi-Z	[CS1 output] Hi-Z [CKE output] Hi-Z [All other] Keep-O	[CS1 output] H [CKE output] SDSELF.SFEN = 0: H SDSELF.SFEN = 1: L [All other] Keep-O	Keep	Hi-Z	Keep
P610/CS0/WE	Hi-Z	[CS0 output] Hi-Z [WE output] Hi-Z [All other] Keep-O	[CS0 output] H [WE output] H [All other] Keep-O	Keep	Hi-Z	Keep

Table 1.1 Port states in each processing state (5 of 6)

Port name	Reset	Software Standby mode		Deep Software Standby mode	After Deep Software Standby mode is canceled (return to startup mode)	
		OPE = 0	OPE = 1		IOKEEP = 0	IOKEEP = 1[*1]
P611/SDCS/CLKOUT	Hi-Z	[SDCS output] Hi-Z [CLKOUT selected] CLKOUT output [All Other] Keep-O	[SDCS output] SDSELF.SFEN = 0: H SDSELF.SFEN = 1: L [CLKOUT selected] CLKOUT output [All other] Keep-O	Keep	Hi-Z	Keep
P612/D08[A08/D08]/DQ08	Hi-Z	[D08 output] Hi-Z [DQ08 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P613/D09[A09/D09]/DQ09	Hi-Z	[D09 output] Hi-Z [DQ09 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P614/D10[A10/D10]/DQ10	Hi-Z	[D10 output] Hi-Z [DQ10 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P615	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P700 to P702	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P703/VCOUT	Hi-Z	[ACMPHS selected] VCOUT output [All other] Keep-O		Keep	Hi-Z	Keep
P704	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P705/AGTIO0	Hi-Z	Keep-O[*2]		Keep	Hi-Z	Keep
P706/USBHS_OVRCURB/IRQ7, P707/USBHS_OVRCURA/IRQ8	Hi-Z	Keep-O[*2]		Keep-O[*3]	Hi-Z	Keep
P708/IRQ11, P709/IRQ10	Hi-Z	Keep-O[*2]		Keep	Hi-Z	Keep
P710 to P713	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P800/D14[A14/D14]/DQ14	Hi-Z	[D14 output] Hi-Z [DQ14 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P801/D15[A15/D15]/DQ15	Hi-Z	[D15 output] Hi-Z [DQ15 output] Hi-Z [All other] Keep-O		Keep	Hi-Z	Keep
P802 to P806	Hi-Z	Keep-O		Keep	Hi-Z	Keep
P900/A23	Hi-Z	[A23 output] Hi-Z [All other] Keep-O	[A23 output] Address output retained [All other] Keep-O	Keep	Hi-Z	Keep
P901/AGTIO1	Hi-Z	Keep-O[*2]		Keep	Hi-Z	Keep
P905/CS4	Hi-Z	[CS4 output] Hi-Z [All other] Keep-O	[CS4 output] H [All other] Keep-O	Keep	Hi-Z	Keep
P906/CS5	Hi-Z	[CS5 output] Hi-Z [All other] Keep-O	[CS5 output] H [All other] Keep-O	Keep	Hi-Z	Keep
P907/CS6	Hi-Z	[CS6 output] Hi-Z [All other] Keep-O	[CS6 output] H [All other] Keep-O	Keep	Hi-Z	Keep

Table 1.1 Port states in each processing state (6 of 6)

Port name	Reset	Software Standby mode		Deep Software Standby mode	After Deep Software Standby mode is canceled (return to startup mode)	
		OPE = 0	OPE = 1		IOKEEP = 0	IOKEEP = 1*1
P908/CS7	Hi-Z	[CS7 output] Hi-Z [All other] Keep-O	[CS7 output] H [All other] Keep-O	Keep	Hi-Z	Keep
PA00, PA01	Hi-Z	Keep-O		Keep	Hi-Z	Keep
PA08 to PA10	Hi-Z	Keep-O		Keep	Hi-Z	Keep
PB00	Hi-Z	Keep-O		Keep	Hi-Z	Keep
PB01/USBHS_VBUS	Hi-Z	Keep-O*2		Keep-O*3	Hi-Z	Keep
USB_DP	Hi-Z	Keep-O*4		Hi-Z*3	Hi-Z	
USB_DM	Hi-Z	Keep-O*4		Hi-Z*3	Hi-Z	
USBHS_DP	Hi-Z	Keep-O*4		Hi-Z*5	Hi-Z	
USBHS_DM	Hi-Z	Keep-O*4		Hi-Z*5	Hi-Z	

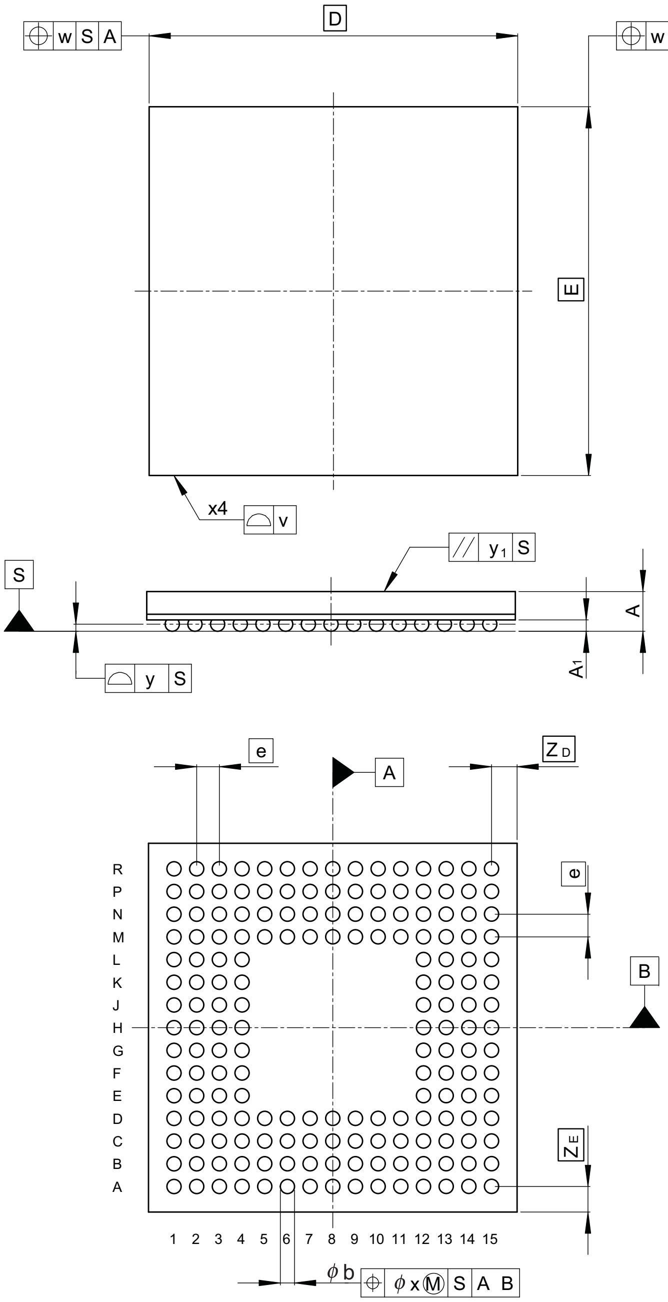
H: High-level
L: Low-level
Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins go to high-impedance.
Keep: Pin states are retained during periods in Software Standby mode.

Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.
Note 2. Input is enabled if the pin is specified as the Software Standby canceling source while it is used as an external interrupt pin.
Note 3. Input is enabled if the pin is specified as the Deep Software Standby canceling source.
Note 4. Input is enabled while the pin is used as an input pin.
Note 5. For host operation, set the USBHS.SYSCFG.DRPD bit to 1 to enable the USBHS_DP and USBHS_DM pull-down resistors. For device operation, set the USBHS.SYSCFG.DPRPU bit to 1 to enable the DP pull-up resistor.

Appendix 2.Package Dimensions

For information on the latest version of the package dimensions or mountings, go to "Packages" on the Renesas Electronics Corporation website.

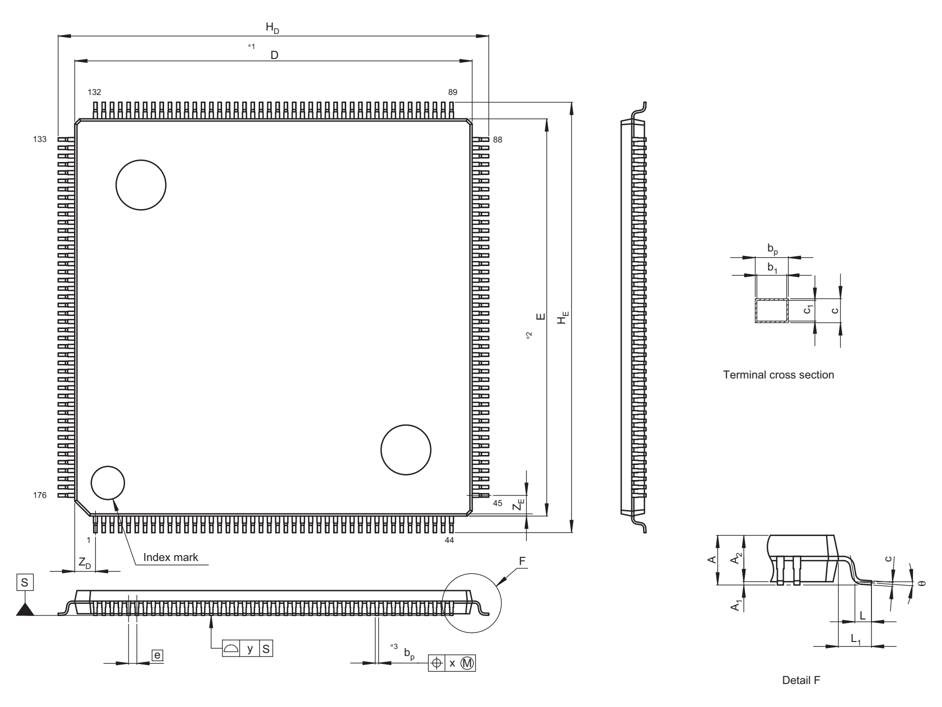
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.)
P-LFBGA176-13x13-0.80	PLBG0176GE-A	176FHS-A	0.45 g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	13.0	—
E	—	13.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.40
A_1	0.35	0.40	0.45
e	—	0.80	—
b	0.45	0.50	0.55
x	—	—	0.08
y	—	—	0.10
y_1	—	—	0.2
S_D	—	—	—
S_E	—	—	—
Z_D	—	0.90	—
Z_E	—	0.90	—

Figure 2.1 176-pin BGA

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP176-24x24-0.50	PLQP0176KB-A	176P6Q-A/FP-176E/FP-176EV	1.8g

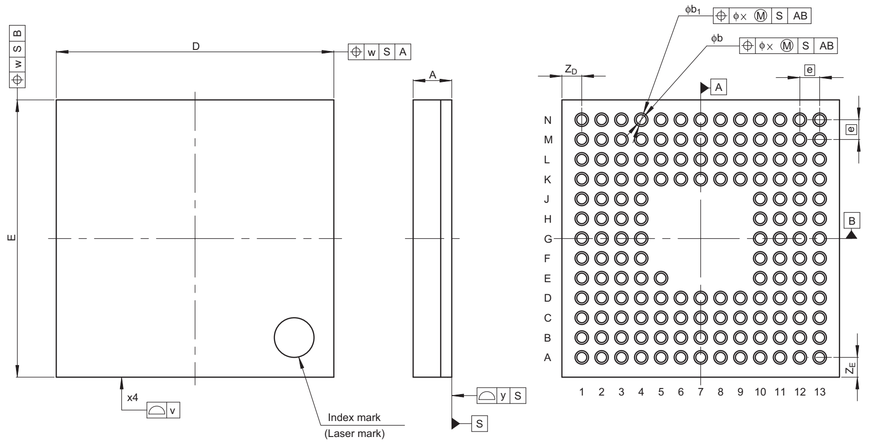


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	23.9	24.0	24.1
E	23.9	24.0	24.1
A_2	—	1.4	—
H_D	25.8	26.0	26.2
H_E	25.8	26.0	26.2
A	—	—	1.7
A_1	0.05	0.1	0.15
b_p	0.15	0.20	0.25
b_1	—	0.18	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	1.25	—
Z_E	—	1.25	—
L	0.35	0.5	0.65
L_1	—	1.0	—

Figure 2.2 176-pin LQFP

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA145-7x7-0.50	PTLG0145KA-A	145F0G	0.1g

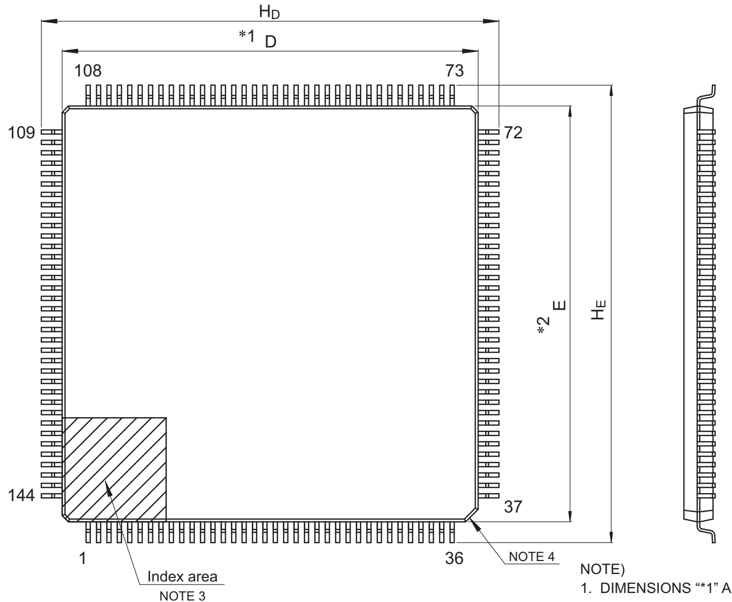


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	7.0	—
E	—	7.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.5	—
b	0.21	0.25	0.29
b_1	0.29	0.34	0.39
x	—	—	0.08
y	—	—	0.08
Z_D	—	0.5	—
Z_E	—	0.5	—

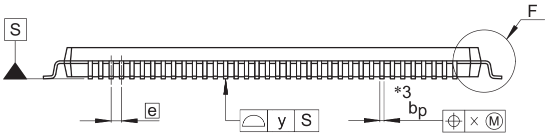
Figure 2.3 145-pin LGA

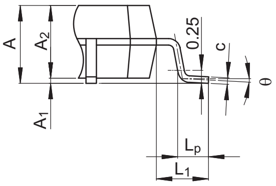
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP144-20x20-0.50	PLQP0144KA-B	—	1.2

Unit: mm



NOTE)
1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.





Detail F

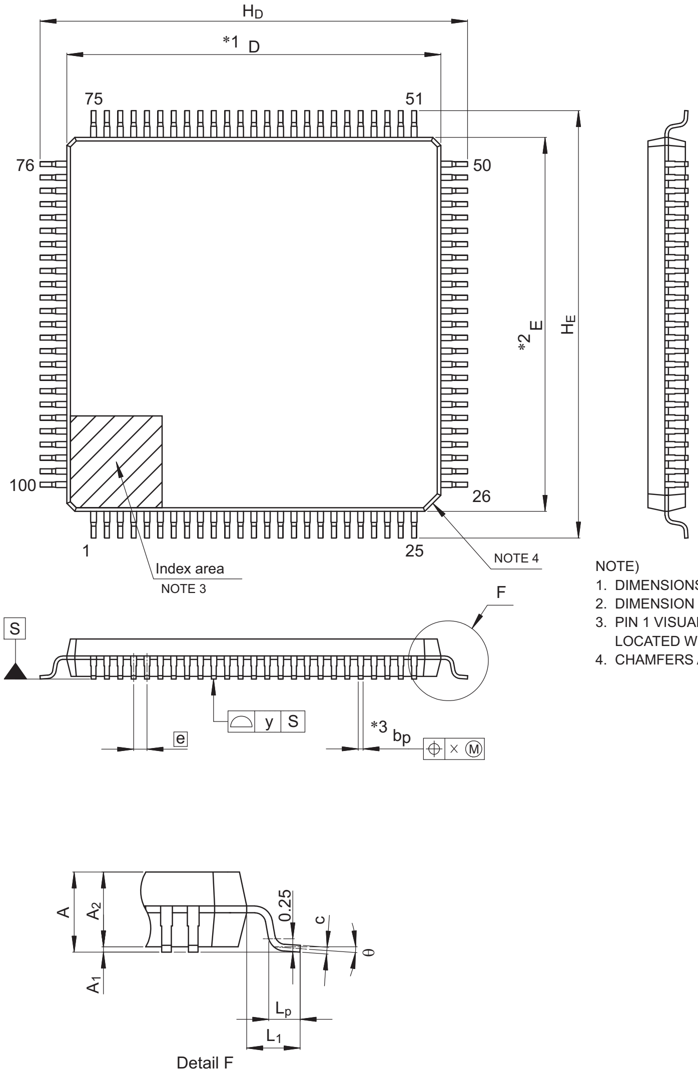
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
$\boxed{e}$	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

© 2016 Renesas Electronics Corporation. All rights reserved.

Figure 2.4 144-pin LQFP

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm



NOTE)

1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

© 2015 Renesas Electronics Corporation. All rights reserved.

Figure 2.5 100-pin LQFP

Appendix 3. I/O Registers

This appendix describes I/O register addresses, access cycles, and reset values by function.

3.1 Peripheral Base Addresses

This section provides the base addresses for peripherals described in this manual.

Table 3.1 shows the name, description, and the base address of each peripheral.

Table 3.1 Peripheral base address (1 of 3)

Name	Description	Base address
MMPU	Bus Master MPU	0x40000000
SMPU	Bus Slave MPU	0x40000C00
SPMON	CPU Stack Pointer Monitor	0x40000D00
MMF	Memory Mirror Function	0x40001000
SRAM	SRAM Control	0x40002000
BUS	Bus Control	0x40003000
DMAC0	Direct Memory Access Controller 0	0x40005000
DMAC1	Direct Memory Access Controller 1	0x40005040
DMAC2	Direct Memory Access Controller 2	0x40005080
DMAC3	Direct Memory Access Controller 3	0x400050C0
DMAC4	Direct Memory Access Controller 4	0x40005100
DMAC5	Direct Memory Access Controller 5	0x40005140
DMAC6	Direct Memory Access Controller 6	0x40005180
DMAC7	Direct Memory Access Controller 7	0x400051C0
DMA	DMAC Module Activation	0x40005200
DTC	Data Transfer Controller	0x40005400
ICU	Interrupt Controller	0x40006000
DBG	Debug Function	0x4001B000
FCACHE	Flash Cache	0x4001C000
SYSTEM	System Control	0x4001E000
PORT0	Port 0 Control Registers	0x40040000
PORT1	Port 1 Control Registers	0x40040020
PORT2	Port 2 Control Registers	0x40040040
PORT3	Port 3 Control Registers	0x40040060
PORT4	Port 4 Control Registers	0x40040080
PORT5	Port 5 Control Registers	0x400400A0
PORT6	Port 6 Control Registers	0x400400C0
PORT7	Port 7 Control Registers	0x400400E0
PORT8	Port 8 Control Registers	0x40040100
PORT9	Port 9 Control Registers	0x40040120
PORTA	Port A Control Registers	0x40040140
PORTB	Port B Control Registers	0x40040160
PFS	Pmn Pin Function Control Register	0x40040800
PMISC	Miscellaneous Port Control Register	0x40040D00
ELC	Event Link Controller	0x40041000
POEG	Port Output Enable Module for GPT	0x40042000
RTC	Realtime Clock	0x40044000
WDT	Watchdog Timer	0x40044200

Table 3.1 Peripheral base address (2 of 3)

Name	Description	Base address
IWDT	Independent Watchdog Timer	0x40044400
CAC	Clock Frequency Accuracy Measurement Circuit	0x40044600
MSTP	Module Stop Control B,C,D	0x40047000
SRCRAM	Sampling Rate Converter RAM	0x40048000
SRC	Sampling Rate Converter	0x4004DFF0
SSIE0	Serial Sound Interface Enhanced (SSIE)	0x4004E000
SSIE1	Serial Sound Interface Enhanced (SSIE)	0x4004E100
CAN0	CAN0 Module	0x40050000
CAN1	CAN1 Module	0x40051000
IIC0	Inter-Integrated Circuit 0	0x40053000
IIC1	Inter-Integrated Circuit 1	0x40053100
IIC2	Inter-Integrated Circuit 2	0x40053200
DOC	Data Operation Circuit	0x40054100
ADC120	12bit A/D Converter 0	0x4005C000
ADC121	12bit A/D Converter 1	0x4005C200
TSN	Temperature Sensor	0x4005D000
DAC12	12-bit D/A converter	0x4005E000
USBHS	USB 2.0 High-Speed Module	0x40060000
SDHI0	SD Host Interface 0	0x40062000
SDHI1	SD Host Interface 1	0x40062400
EDMAC0	DMA Controller for the Ethernet Controller Channel 0	0x40064000
ETHERC0	Ethernet Controller Channel 0	0x40064100
PTPEDMAC	DMA Controller for EPTPC	0x40064400
EPTPC_CFG	EPTPC Configuration	0x40064500
EPTPC	PTP Module for the Ethernet Controller	0x40065000
EPTPC0	PTP Module 0 for the Ethernet Controller	0x40065800
SCI0	Serial Communication Interface 0	0x40070000
SCI1	Serial Communication Interface 1	0x40070020
SCI2	Serial Communication Interface 2	0x40070040
SCI3	Serial Communication Interface 3	0x40070060
SCI4	Serial Communication Interface 4	0x40070080
SCI5	Serial Communication Interface 5	0x400700A0
SCI6	Serial Communication Interface 6	0x400700C0
SCI7	Serial Communication Interface 7	0x400700E0
SCI8	Serial Communication Interface 8	0x40070100
SCI9	Serial Communication Interface 9	0x40070120
IRDA	Infrared Data Association	0x40070F00
SPI0	Serial Peripheral Interface 0	0x40072000
SPI1	Serial Peripheral Interface 1	0x40072100
CRC	CRC Calculator	0x40074000
GPT32EH0	General PWM Timer 0 (32-bit Enhanced High Resolution)	0x40078000
GPT32EH1	General PWM Timer 1 (32-bit Enhanced High Resolution)	0x40078100
GPT32EH2	General PWM Timer 2 (32-bit Enhanced High Resolution)	0x40078200
GPT32EH3	General PWM Timer 3 (32-bit Enhanced High Resolution)	0x40078300
GPT32E4	General PWM Timer 4 (32-bit Enhanced)	0x40078400

Table 3.1 Peripheral base address (3 of 3)

Name	Description	Base address
GPT32E5	General PWM Timer 5 (32-bit Enhanced)	0x40078500
GPT32E6	General PWM Timer 6 (32-bit Enhanced)	0x40078600
GPT32E7	General PWM Timer 7 (32-bit Enhanced)	0x40078700
GPT328	General PWM Timer 8 (32-bit)	0x40078800
GPT329	General PWM Timer 9 (32-bit)	0x40078900
GPT3210	General PWM Timer 10 (32-bit)	0x40078A00
GPT3211	General PWM Timer 11 (32-bit)	0x40078B00
GPT3212	General PWM Timer 12 (32-bit)	0x40078C00
GPT3213	General PWM Timer 13 (32-bit)	0x40078D00
GPT_OPS	Output Phase Switching Controller	0x40078FF0
GPT_ODC	PWM Delay Generation Circuit	0x4007B000
KINT	Key Interrupt Function	0x40080000
CTSU	Capacitive Touch Sensing Unit	0x40081000
AGT0	Asynchronous General purpose Timer 0	0x40084000
AGT1	Asynchronous General purpose Timer 1	0x40084100
ACMPHS0	High-Speed Analog Comparator 0	0x40085000
ACMPHS1	High-Speed Analog Comparator 1	0x40085100
ACMPHS2	High-Speed Analog Comparator 2	0x40085200
ACMPHS3	High-Speed Analog Comparator 3	0x40085300
ACMPHS4	High-Speed Analog Comparator 4	0x40085400
ACMPHS5	High-Speed Analog Comparator 5	0x40085500
USBFS	USB 2.0 FS Module	0x40090000
PDC	Parallel Data Capture Unit	0x40094000
GLCDC	Graphics LCD Controller	0x400E0000
DRW	2D Drawing Engine	0x400E4000
JPEG	JPEG Codec	0x400E6000
QSPI	Quad-SPI	0x64000000

Name = Peripheral name
Description = Peripheral functionality
Base address = Lowest reserved address or address used by the peripheral

3.2 Access Cycles

This section provides access cycle information for the I/O registers described in this manual.

The following information applies to Table 3.2 and Table 3.3:

- Registers are grouped by associated module
- The number of access cycles indicates the number of cycles based on the specified reference clock
- In the internal I/O area, reserved addresses that are not allocated to registers must not be accessed, otherwise operations cannot be guaranteed
- The number of I/O access cycles depends on bus cycles of the internal peripheral bus, divided clock synchronization cycles, and wait cycles of each module. Divided clock synchronization cycles differ depending on the frequency ratio between ICLK and PCLK.
- When the frequency of ICLK is equal to that of PCLK, the number of divided clock synchronization cycles is always constant.
- When the frequency of ICLK is greater than that of PCLK, at least 1 PCLK cycle is added to the number of divided clock synchronization cycles.

Note: This applies to the number of cycles when access from the CPU does not conflict with the instruction fetching to the external memory or bus access from other bus masters such as DTC or DMAC.

Table 3.2 Access cycles (1 of 2)

	Address		Number of access cycles					
			ICLK = PCLK		ICLK > PCLK^{*1}		Cycle Unit	
Peripherals	**From**	**To**	**Read**	**Write**	**Read**	**Write**		**Related function**
MMPU, SMPU, SPMON, MMF, SRAM, BUS, DMACn, DMA, DTC, ICU, DBG, FCACHE	4000 0000h	4001 CFFFh	4				ICLK	Memory Protection Unit, Memory Mirror Function, SRAM, Buses, DMA Controller, Data Transfer Controller, Interrupt Controller, CPU, Flash Memory
SYSTEM	4001 E000h	4001 E3FFh	5				ICLK	Low Power Modes, Resets, Low Voltage Detection, Clock Generation Circuit, Register Write Protection
SYSTEM	4001 E400h	4001 E6FFh	9		5 to 8		PCLKB	Low Power Modes, Resets, Low Voltage Detection, Battery Backup Function
PORTn, PFS, PMISC, ELC, POEG, RTC, WDT, IWDT, CAC, MSTP	4004 0000h	4004 7FFFh	3		2 to 3		PCLKB	I/O Ports, Event Link Controller, Port Output Enable for GPT, Realtime Clock, Watchdog Timer, Independent Watchdog Timer, Clock Frequency Accuracy Measurement Circuit, Module Stop Control
SRCRAM	4004 8000h	4004 DFEFh	4	3	3 to 4	2 to 3	PCLKB	Sampling Rate Converter
SRC	4004 DFF0h	4004 DFF7h	5		4 to 5		PCLKB	
SRC	4004 DFF8h	4004 DFFFh	3		2 to 3		PCLKB	
SSIEn, CANn, IICn, DOC, ADC12n, TSN, DAC12	4004 E000h	4005 FFFFh	3		2 to 3		PCLKB	Serial Sound Interface Enhanced, Controller Area Network Module, I2C Bus Interface, Data Operation Circuit, 12-Bit A/D Converter, Temperature Sensor, 12-Bit D/A Converter
USBHS	4006 0000h	4006 0FFFh	(3+BWAIT)^{*2}		(2+BWAIT) to (3+BWAIT)^{*2}		PCLKA	USB 2.0 High-Speed Module
SDHIn	4006 2000h	4006 2FFFh	3		2 to 3		PCLKA	SD/MMC Host Interface
EDMAC0	4006 4000h	4006 40FFh	4		-		PCLKA	Ethernet DMA Controller
ETHERC0	4006 4100h	4006 41FFh	13		-		PCLKA	Ethernet MAC Controller
PTPEDMAC	4006 4400h	4006 44FFh	4		-		PCLKA	Ethernet DMA Controller
EPTPC_CFG, EPTPC, EPTPC0	4006 4500h	4006 5BFFh	(1+wait cycle)^{*3}		-		PCLKA	Ethernet PTP Controller
SCI0 to SCI9	4007 0000h	4007 0EFFh	3^{*4}		2 to 3^{*4}		PCLKA	Serial Communications Interface
IRDA	4007 0F00h	4007 0FFFh	3		2 to 3		PCLKA	IrDA Interface
SPI0, SPI1	4007 2000h	4007 2FFFh	3^{*5}		2 to 3^{*5}		PCLKA	Serial Peripheral Interface
CRC	4007 4000h	4007 4FFFh	3		2 to 3		PCLKA	CRC Calculator
GPT32EHi, GPT32Ej, GPT32k, GPT_OPS	4007 8000h	4007 8FFFh	5	3	4 to 5	2 to 3	PCLKA	General PWM Timer

Table 3.2 Access cycles (2 of 2)

Peripherals	Address		Number of access cycles					Related function
			ICLK = PCLK		ICLK > PCLK*1		Cycle Unit	
	From	To	Read	Write	Read	Write		
GPT_ODC	4007 B000h	4007 BFFFh	2		1 to 2		PCLKA	PWM Delay Generation Circuit
KINT, CTSU	4008 0000h	4008 1FFFh	2		1 to 2		PCLKB	Key interrupt Function, Capacitive Touch Sensing Unit
AGTn	4008 4000h	4008 4FFFh	5	3	4 to 5	2 to 3	PCLKB	Asynchronous General Purpose Timer
ACMPHSn	4008 5000h	4008 5FFFh	2		1 to 2		PCLKB	High-Speed Analog Comparator
USBFS	4009 0000h	4009 03FFh	4		3 to 4		PCLKB	USB 2.0 Full-Speed Module
USBFS	4009 0400h	4009 04FFh	2		1 to 2		PCLKB	USB 2.0 Full-Speed Module
PDC	4009 4000h	4009 4FFFh	3		2 to 3		PCLKB	Parallel Data Capture Unit
GLCDC, DRW	400E 0000h	400E 4FFFh	3		-		PCLKA	Graphics LCD Controller 2D Drawing Engine
JPEG	400E 6000h	400E 603Fh	13	5	-		PCLKA	JPEG Codec
JPEG	400E 6040h	400E 6FFFh	5	4	-		PCLKA	JPEG Codec
QSPI	6400 0000h	6400 000Fh	3	13 to *6	2 to 3	12 to *6	PCLKA	Quad Serial Peripheral Interface
QSPI	6400 0010h	6400 0013h	24 to *6	5 to *6	23 to *6	4 to *6	PCLKA	Quad Serial Peripheral Interface
QSPI	6400 0014h	6400 0037h	3	13 to *6	2 to 3	12 to *6	PCLKA	Quad Serial Peripheral Interface
QSPI	6400 0804h	6400 0807h	2	2	1 to 2	1 to 2	PCLKA	Quad Serial Peripheral Interface

Note 1. If the number of PCLK cycles is non-integer (for example 1.5), the minimum value is without the decimal point, and the maximum value is rounded up to the decimal point. For example, 1.5 to 2. 5 is 1 to 3.

Note 2. BWAIT is the number of waits (not cycles) described in the USBHS.BUSWAIT register.

Note 3. The wait cycle refers to the EPTPC chapter (30.6.2 Wait Cycles for Register Access).

Note 4. When accessing a 16-bit register (FTDRHL, FRDRHL, FCR, FDR, LSR, and CDR), access is 2 cycles more than the value shown in Table 3.2. When accessing an 8-bit register (including FTDRH, FTDRL, FRDRH, and FRDRL), the access cycles are as shown in Table 3.2.

Note 5. When accessing the 32-bit register (SPDR), access is 2 cycles more than the value in Table 3.2. When accessing an 8-bit or 16-bit register (SPDR_HA), the access cycles are as shown in Table 3.2.

Note 6. The access cycles depend on the QSPI bus cycles.

3.3 Register Descriptions

This section provides information associated with registers described in this manual.

Table 3.3 shows a list of registers including address offsets, address sizes, access rights, and reset values.

Table 3.3 Register description (1 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
MMPU	3	0x400	A,B,C	MMPUCTL%s	Bus Master MPU Control Register	0x000	16	read/ write	0x0000	0xFFFF
MMPU	-	-	-	MMPUPTA	Group A Protection of Register	0x102	16	read/ write	0x0000	0xFFFF
MMPU	32	0x010	0-31	MMPUACA%s	Group A Region %s Access Control Register	0x200	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (2 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
MMPU	32	0x010	0-31	MMPUSA%s	Group A Region %s Start Address Register	0x204	32	read/ write	0x00000000	0x00000003
MMPU	32	0x010	0-31	MMPUEA%s	Group A Region %s End Address Register	0x208	32	read/ write	0x00000003	0x00000003
MMPU	-	-	-	MMPUPTB	Group B Protection of Register	0x502	16	read/ write	0x0000	0xFFFF
MMPU	8	0x010	0-7	MMPUACB%s	Group B Region %s Access Control Register	0x600	32	read/ write	0x00000000	0xFFFFFFFF
MMPU	8	0x010	0-7	MMPUSB%s	Group B Region %s Start Address Register	0x604	32	read/ write	0x00000000	0x00000003
MMPU	8	0x010	0-7	MMPUEB%s	Group B Region %s End Address Register	0x608	32	read/ write	0x00000003	0x00000003
MMPU	-	-	-	MMPUPTC	Group C protection of register	0x902	16	read/ write	0x0000	0xFFFF
MMPU	8	0x010	0-7	MMPUACC%s	Group C Region %s Access Control Register	0xA00	32	read/ write	0x00000000	0xFFFFFFFF
MMPU	8	0x010	0-7	MMPUSC%s	Group C Region %s Start Address Register	0xA04	32	read/ write	0x00000000	0x00000003
MMPU	8	0x010	0-7	MMPUEC%s	Group C Region %s Start Address Register	0xA08	32	read/ write	0x00000003	0x00000003
SMPU	-	-	-	SMPUCTL	Slave MPU Control Register	0x00	16	read/ write	0x0000	0xFFFF
SMPU	-	-	-	SMPUMBIU	Access Control Register for MBIU	0x10	16	read/ write	0x2000	0xFFFF
SMPU	-	-	-	SMPUFBIU	Access Control Register for FBIU	0x14	16	read/ write	0x00C0	0xFFFF
SMPU	2	0x4	0,1	SMPUSRAM%s	Access Control Register for SRAM%s	0x18	16	read/ write	0x0000	0xFFFF
SMPU	4	0x4	0,2,6, 7	SMPUP%sBIU	Access Control Register for P%sBIU	0x20	16	read/ write	0x00F0	0xFFFF
SMPU	-	-	-	SMPUEXBIU	Access Control Register for EXBIU	0x30	16	read/ write	0x0000	0xFFFF
SMPU	-	-	-	SMPUEXBIU2	Access Control Register for EXBIU2	0x34	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	MSPMPUOAD	Stack Pointer Monitor Operation After Detection Register	0x00	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	MSPMPUCTL	Stack Pointer Monitor Access Control Register	0x04	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	MSPMPUPT	Stack Pointer Monitor Protection Register	0x06	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	MSPMPUSA	Main Stack Pointer Monitor Start Address Register	0x08	32	read/ write	0x00000000	0x00000003
SPMON	-	-	-	MSPMPUEA	Main Stack Pointer Monitor End Address Register	0x0C	32	read/ write	0x00000003	0x00000003
SPMON	-	-	-	PSPMPUOAD	Stack Pointer Monitor Operation After Detection Register	0x10	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	PSPMPUCTL	Stack Pointer Monitor Access Control Register	0x14	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	PSPMPUPT	Stack Pointer Monitor Protection Register	0x16	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	PSPMPUSA	Process Stack Pointer Monitor Start Address Register	0x18	32	read/ write	0x00000000	0x00000003
SPMON	-	-	-	PSPMPUEA	Process Stack Pointer Monitor End Address Register	0x1C	32	read/ write	0x00000003	0x00000003
MMF	-	-	-	MMSFR	MemMirror Special Function Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (3 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
MMF	-	-	-	MMEN	MemMirror Enable Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
SRAM	-	-	-	PARIOAD	SRAM Parity Error Operation After Detection Register	0x00	8	read/ write	0x00	0xFF
SRAM	-	-	-	SRAMPRCR	SRAM Protection Register	0x04	8	read/ write	0x00	0xFF
SRAM	-	-	-	SRAMWTSC	RAM Wait State Control Register	0x08	8	read/ write	0x0E	0xFF
SRAM	-	-	-	ECCMODE	ECCRAM Operating Mode Control Register	0xC0	8	read/ write	0x00	0xFF
SRAM	-	-	-	ECC2STS	ECCRAM 2-Bit Error Status Register	0xC1	8	read/ write	0x00	0xFF
SRAM	-	-	-	ECC1STSEN	ECCRAM 1-Bit Error Information Update Enable Register	0xC2	8	read/ write	0x00	0xFF
SRAM	-	-	-	ECC1STS	ECCRAM 1-Bit Error Status Register	0xC3	8	read/ write	0x00	0xFF
SRAM	-	-	-	ECCPRCR	ECCRAM Protection Register	0xC4	8	read/ write	0x00	0xFF
SRAM	-	-	-	ECCPRCR2	ECCRAM Protection Register 2	0xD0	8	read/ write	0x00	0xFF
SRAM	-	-	-	ECCRAMETST	ECCRAM Test Control Register	0xD4	8	read/ write	0x00	0xFF
SRAM	-	-	-	ECCOAD	RAM ECC Error Operation After Detection Register	0xD8	8	read/ write	0x00	0xFF
BUS	8	0x10	0-7	CS%sMOD	CS%s Mode Register	0x0002	16	read/ write	0x0000	0xFFFF
BUS	8	0x10	0-7	CS%sWCR1	CS%s Wait Control Register 1	0x0004	32	read/ write	0x07070707	0xFFFFFFFF
BUS	8	0x10	0-7	CS%sWCR2	CS%s Wait Control Register 2	0x0008	32	read/ write	0x00000007	0xFFFFFFFF
BUS	-	-	-	CS0CR	CS0 Control Register	0x0802	16	read/ write	0x0021	0xFFFF
BUS	8	0x10	0-7	CS%sREC	CS%s Recovery Cycle Register	0x080A	16	read/ write	0x0000	0xFFFF
BUS	7	0x10	1-7	CS%sCR	CS%s Control Register	0x0812	16	read/ write	0x0000	0xFFFF
BUS	-	-	-	CSRECEN	CS Recovery Cycle Insertion Enable Register	0x0880	16	read/ write	0x3E3E	0xFFFF
BUS	-	-	-	SDCCR	SDC Control Register	0x0C00	8	read/ write	0x00	0xFF
BUS	-	-	-	SDCMOD	SDC Mode Register	0x0C01	8	read/ write	0x00	0xFF
BUS	-	-	-	SDAMOD	SDRAM Access Mode Register	0x0C02	8	read/ write	0x00	0xFF
BUS	-	-	-	SDSELF	SDRAM Self-Refresh Control Register	0x0C10	8	read/ write	0x00	0xFF
BUS	-	-	-	SDRFCR	SDRAM Refresh Control Register	0x0C14	16	read/ write	0x0001	0xFFFF
BUS	-	-	-	SDRFEN	SDRAM Auto-Refresh Control Register	0x0C16	8	read/ write	0x00	0xFF
BUS	-	-	-	SDICR	SDRAM Initialization Sequence Control Register	0x0C20	8	read/ write	0x00	0xFF
BUS	-	-	-	SDIR	SDRAM Initialization Register	0x0C24	16	read/ write	0x0010	0xFFFF
BUS	-	-	-	SDADR	SDRAM Address Register	0x0C40	8	read/ write	0x00	0xFF
BUS	-	-	-	SDTR	SDRAM Timing Register	0x0C44	32	read/ write	0x00000002	0xFFFFFFFF

Table 3.3 Register description (4 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
BUS	-	-	-	SDMOD	SDRAM Mode Register	0x0C48	16	read/ write	0x0000	0xFFFF
BUS	-	-	-	SDSR	SDRAM Status Register	0x0C50	8	read-only	0x00	0xFF
BUS	2	0x4	M4I, M4D	BUSMCNT%s	Master Bus Control Register %s	0x1000	16	read/ write	0x0000	0xFFFF
BUS	-	-	-	BUSMCNTSYS	Master Bus Control Register SYS	0x1008	16	read/ write	0x0000	0xFFFF
BUS	-	-	-	BUSMCNTDMA	Master Bus Control Register DMA	0x100C	16	read/ write	0x0000	0xFFFF
BUS	2	0x4	EDM, GPX	BUSMCNT%s	Master Bus Control Register %s	0x1010	16	read/ write	0x0000	0xFFFF
BUS	2	0x4	FLI,RAMH	BUSSCNT%s	Slave Bus Control Register %s	0x1100	16	read/ write	0x0000	0xFFFF
BUS	-	-	-	BUSSCNTMBIU	Slave Bus Control Register MBIU	0x1108	16	read/ write	0x0000	0xFFFF
BUS	2	0x4	RAM0,RAM1	BUSSCNT%s	Slave Bus Control Register %s	0x110C	16	read/ write	0x0000	0xFFFF
BUS	4	0x4	P0B, P2B, P3B, P4B	BUSSCNT%s	Slave Bus Control Register %s	0x1114	16	read/ write	0x0000	0xFFFF
BUS	2	0x4	P6B, P7B	BUSSCNT%s	Slave Bus Control Register %s	0x1128	16	read/ write	0x0000	0xFFFF
BUS	4	0x4	FBU, EXT, EXT2,GPX	BUSSCNT%s	Slave Bus Control Register %s	0x1130	16	read/ write	0x0000	0xFFFF
BUS	11	0x10	1-11	BUS%sERRADD	Bus Error Address Register %s	0x1800	32	read-only	0x00000000	0x00000000
BUS	11	0x10	1-11	BUS%sERRSTAT	Bus Error Status Register %s	0x1804	8	read-only	0x00	0xFE
DMAC0-7	-	-	-	DMSAR	DMA Source Address Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
DMAC0-7	-	-	-	DMDAR	DMA Destination Address Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
DMAC0-7	-	-	-	DMCRA	DMA Transfer Count Register	0x08	32	read/ write	0x00000000	0xFFFFFFFF
DMAC0-7	-	-	-	DMCRB	DMA Block Transfer Count Register	0x0C	16	read/ write	0x0000	0xFFFF
DMAC0-7	-	-	-	DMTMD	DMA Transfer Mode Register	0x10	16	read/ write	0x0000	0xFFFF
DMAC0-7	-	-	-	DMINT	DMA Interrupt Setting Register	0x13	8	read/ write	0x00	0xFF
DMAC0-7	-	-	-	DMAMD	DMA Address Mode Register	0x14	16	read/ write	0x0000	0xFFFF
DMAC0-7	-	-	-	DMOFR	DMA Offset Register	0x18	32	read/ write	0x00000000	0xFFFFFFFF
DMAC0-7	-	-	-	DMCNT	DMA Transfer Enable Register	0x1C	8	read/ write	0x00	0xFF
DMAC0-7	-	-	-	DMREQ	DMA Software Start Register	0x1D	8	read/ write	0x00	0xFF
DMAC0-7	-	-	-	DMSTS	DMAC Module Activation Register	0x1E	8	read/ write	0x00	0xFF
DMA	-	-	-	DMAST	DMA Module Activation Register	0x00	8	read/ write	0x00	0xFF
DTC	-	-	-	DTCCR	DTC Control Register	0x00	8	read/ write	0x08	0xFF

Table 3.3 Register description (5 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
DTC	-	-	-	DTCVBR	DTC Vector Base Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
DTC	-	-	-	DTCST	DTC Module Start Register	0x0C	8	read/ write	0x00	0xFF
DTC	-	-	-	DTCSTS	DTC Status Register	0x0E	16	read-only	0x0000	0xFFFF
ICU	16	0x1	0-15	IRQCR%s	IRQ Control Register %s	0x000	8	read/ write	0x00	0xFF
ICU	-	-	-	NMICR	NMI Pin Interrupt Control Register	0x100	8	read/ write	0x00	0xFF
ICU	-	-	-	NMIER	Non-Maskable Interrupt Enable Register	0x120	16	read/ write	0x0000	0xFFFF
ICU	-	-	-	NMICLR	Non-Maskable Interrupt Status Clear Register	0x130	16	write-only	0x0000	0xFFFF
ICU	-	-	-	NMISR	Non-Maskable Interrupt Status Register	0x140	16	read-only	0x0000	0xFFFF
ICU	-	-	-	WUPEN	Wake Up interrupt enable register	0x1A0	32	read/ write	0x00000000	0xFFFFFFFF
ICU	-	-	-	SELSR0	Event Selection to Cancel Snooze Mode	0x200	16	read/ write	0x0000	0xFFFF
ICU	8	0x4	0-7	DELSR%s	DMAC Event Link Setting Register %s	0x280	32	read/ write	0x00000000	0xFFFFFFFF
ICU	96	0x4	0-95	IELSR%s	INT Event Link Setting Register %s	0x300	32	read/ write	0x00000000	0xFFFFFFFF
DBG	-	-	-	DBGSTR	Debug Status Register	0x000	32	read-only	0x00000000	0xFFFFFFFF
DBG	-	-	-	DBGSTOPCR	Debug Stop Control Register	0x010	32	read/ write	0x00000003	0xFFFFFFFF
DBG	-	-	-	TRACECTR	Trace Control Register	0x020	32	read/ write	0x00000000	0xFFFFFFFF
FCACHE	-	-	-	FCACHEE	Flash Cache Enable Register	0x100	16	read/ write	0x0000	0xFFFF
FCACHE	-	-	-	FCACHEIV	Flash Cache Invalidate Register	0x104	16	read/ write	0x0000	0xFFFF
FCACHE	-	-	-	FLWT	Flash Wait Cycle Register	0x11C	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	SBYCR	Standby Control Register	0x00C	16	read/ write	0x4000	0xFFFF
SYSTEM	-	-	-	MSTPCRA	Module Stop Control Register A	0x01C	32	read/ write	0xFFBFFF1C	0xFFFFFFFF
SYSTEM	-	-	-	SCKDIVCR	System Clock Division Control Register	0x020	32	read/ write	0x22022222	0xFFFFFFFF
SYSTEM	-	-	-	SCKDIVCR2	System Clock Division Control Register 2	0x024	8	read/ write	0x40	0xFF
SYSTEM	-	-	-	SCKSCR	System Clock Source Control Register	0x026	8	read/ write	0x01	0xFF
SYSTEM	-	-	-	PLLCCR	PLL Clock Control Register	0x028	16	read/ write	0x1300	0xFFFF
SYSTEM	-	-	-	PLLCR	PLL Control Register	0x02A	8	read/ write	0x01	0xFF
SYSTEM	-	-	-	BCKCR	External Bus Clock Control Register	0x030	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	MOSCCR	Main Clock Oscillator Control Register	0x032	8	read/ write	0x01	0xFF
SYSTEM	-	-	-	HOCOCR	High-Speed On-Chip Oscillator Control Register	0x036	8	read/ write	0x00	0xFE
SYSTEM	-	-	-	MOCOCR	Middle-Speed On-Chip Oscillator Control Register	0x038	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	FLLCR1	FLL Control Register 1	0x039	8	read/ write	0x00	0xFF

Table 3.3 Register description (6 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	FLLCR2	FLL Control Register 2	0x03A	16	read/ write	0x0000	0xFFFF
SYSTEM	-	-	-	OSCSF	Oscillation Stabilization Flag Register	0x03C	8	read-only	0x00	0xFE
SYSTEM	-	-	-	CKOCR	Clock Out Control Register	0x03E	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	TRCKCR	Trace Clock Control Register	0x03F	8	read/ write	0x01	0xFF
SYSTEM	-	-	-	OSTDCR	Oscillation Stop Detection Control Register	0x040	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	OSTDSR	Oscillation Stop Detection Status Register	0x041	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	EBCKOCR	External Bus Clock Output Control Register	0x052	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	SDCKOCR	SDRAM Clock Output Control Register	0x053	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	MOCOUTCR	MOCO User Trimming Control Register	0x061	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	HOCOUTCR	HOCO User Trimming Control Register	0x062	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	SNZCR	Snooze Control Register	0x092	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	SNZEDCR	Snooze End Control Register	0x094	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	SNZREQCR	Snooze Request Control Register	0x098	32	read/ write	0x00000000	0xFFFFFFFF
SYSTEM	-	-	-	OPCCR	Operating Power Control Register	0x0A0	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	MOSCWTCR	Main Clock Oscillator Wait Control Register	0x0A2	8	read/ write	0x05	0xFF
SYSTEM	-	-	-	HOCOWTCR	High-speed on-chip oscillator wait control register	0x0A5	8	read/ write	0x02	0xFF
SYSTEM	-	-	-	SOPCCR	Sub Operating Power Control Register	0x0AA	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	RSTSR1	Reset Status Register 1	0x0C0	16	read/ write	0x0000	0xE0F8
SYSTEM	2	0x2	1,2	LVD%sCR1	Voltage Monitor %s Circuit Control Register 1	0x0E0	8	read/ write	0x01	0xFF
SYSTEM	2	0x2	1,2	LVD%sSR	Voltage Monitor %s Circuit Status Register	0x0E1	8	read/ write	0x02	0xFF
SYSTEM	-	-	-	PRCR	Protect Register	0x3FE	16	read/ write	0x0000	0xFFFF
SYSTEM	-	-	-	DPSBYCR	Deep Standby Control Register	0x400	8	read/ write	0x01	0xFF
SYSTEM	-	-	-	DPSIER0	Deep Standby Interrupt Enable Register 0	0x402	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIER1	Deep Standby Interrupt Enable Register 1	0x403	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIER2	Deep Standby Interrupt Enable Register 2	0x404	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIER3	Deep Standby Interrupt Enable Register 3	0x405	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIFR0	Deep Standby Interrupt Flag Register 0	0x406	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIFR1	Deep Standby Interrupt Flag Register 1	0x407	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIFR2	Deep Standby Interrupt Flag Register 2	0x408	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIFR3	Deep Standby Interrupt Flag Register 3	0x409	8	read/ write	0x00	0xFF

Table 3.3 Register description (7 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	DPSIEGR0	Deep Standby Interrupt Edge Register 0	0x40A	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIEGR1	Deep Standby Interrupt Edge Register 1	0x40B	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	DPSIEGR2	Deep Standby Interrupt Edge Register 2	0x40C	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	SYOCDCR	System Control OCD Control Register	0x40E	8	read/ write	0x00	0xFE
SYSTEM	-	-	-	STCONR	Standby Condition Register	0x40F	8	read/ write	0xC3	0xFF
SYSTEM	-	-	-	RSTSR0	Reset Status Register 0	0x410	8	read/ write	0x00	0x70
SYSTEM	-	-	-	RSTSR2	Reset Status Register 2	0x411	8	read/ write	0x00	0xFE
SYSTEM	-	-	-	MOMCR	Main Clock Oscillator Mode Oscillation Control Register	0x413	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	FWEPROR	Flash P/E Protect Register	0x416	8	read/ write	0x02	0xFF
SYSTEM	-	-	-	LVCMPCR	Voltage Monitor Circuit Control Register	0x417	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	LVDLVLR	Voltage Detection Level Select Register	0x418	8	read/ write	0xF3	0xFF
SYSTEM	2	0x1	1,2	LVD%sCR0	Voltage Monitor %s Circuit Control Register 0	0x41A	8	read/ write	0x8A	0xF7
SYSTEM	-	-	-	SOSCCR	Sub-clock oscillator control register	0x480	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	SOMCR	Sub Clock Oscillator Mode Control Register	0x481	8	read/ write	0x00	0xFD
SYSTEM	-	-	-	LOCOCR	Low-Speed On-Chip Oscillator Control Register	0x490	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	LOCOUTCR	LOCO User Trimming Control Register	0x492	8	read/ write	0x00	0xFF
SYSTEM	-	-	-	VBTICTLR	VBATT Input Control Register	0x4BB	8	read/ write	0x00	0xF8
SYSTEM	512	0x1	0-511	VBTBKR[%s]	VBATT Backup Register [%s]	0x500	8	read/ write	0x00	0x00
PORT0,5-9,A,B	-	-	-	PCNTR1	Port Control Register 1	0x00	32	read/ write	0x00000000	0xFFFFFFFF
PORT0,5-9,A,B	-	-	-	PODR	Output data register	0x00	16	read/ write	0x0000	0xFFFF
PORT0,5-9,A,B	-	-	-	PDR	Data direction register	0x02	16	read/ write	0x0000	0xFFFF
PORT0,5-9,A,B	-	-	-	PCNTR2	Port Control Register 2	0x04	32	read-only	0x00000000	0xFFFF0000
PORT0,5-9,A,B	-	-	-	PIDR	Input data register	0x06	16	read-only	0x0000	0x0000
PORT0,5-9,A,B	-	-	-	PCNTR3	Port Control Register 3	0x08	32	write-only	0x00000000	0xFFFFFFFF
PORT0,5-9,A,B	-	-	-	PORR	Output reset register	0x08	16	write-only	0x0000	0xFFFF
PORT0,5-9,A,B	-	-	-	POSR	Output set register	0x0A	16	write-only	0x0000	0xFFFF
PORT1-4	-	-	-	PCNTR1	Port Control Register 1	0x00	32	read/ write	0x00000000	0xFFFFFFFF
PORT1-4	-	-	-	PODR	Output data register	0x00	16	read/ write	0x0000	0xFFFF
PORT1-4	-	-	-	PDR	Data direction register	0x02	16	read/ write	0x0000	0xFFFF
PORT1-4	-	-	-	PCNTR2	Port Control Register 2	0x04	32	read-only	0x00000000	0xFFFF0000

Table 3.3 Register description (8 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PORT1-4	-	-	-	EIDR	Event input data register	0x04	16	read-only	0x0000	0x0000
PORT1-4	-	-	-	PIDR	Input data register	0x06	16	read-only	0x0000	0xFFFF
PORT1-4	-	-	-	PCNTR3	Port Control Register 3	0x08	32	write-only	0x00000000	0xFFFFFFFF
PORT1-4	-	-	-	PORR	Output set register	0x08	16	write-only	0x0000	0xFFFF
PORT1-4	-	-	-	POSR	Output reset register	0x0A	16	write-only	0x0000	0xFFFF
PORT1-4	-	-	-	PCNTR4	Port Control Register 4	0x0C	32	read/write	0x00000000	0xFFFFFFFF
PORT1-4	-	-	-	EORR	Event output set register	0x0C	16	read/write	0x0000	0xFFFF
PORT1-4	-	-	-	EOSR	Event output reset register	0x0E	16	read/write	0x0000	0xFFFF
PFS	-	-	-	P000PFS	P000 Pin Function Control Register	0x000	32	read/write	0x00008000	0xFFFFFFFF
PFS	-	-	-	P000PFS_HA	P000 Pin Function Control Register	0x002	16	read/write	0x8000	0xFFFF
PFS	-	-	-	P000PFS_BY	P000 Pin Function Control Register	0x003	8	read/write	0x00	0xFF
PFS	7	0x4	1-7	P00%sPFS	P00%s Pin Function Control Register	0x004	32	read/write	0x00008000	0xFFFFFFFF
PFS	7	0x4	1-7	P00%sPFS_HA	P00%s Pin Function Control Register	0x006	16	read/write	0x8000	0xFFFF
PFS	7	0x4	1-7	P00%sPFS_BY	P00%s Pin Function Control Register	0x007	8	read/write	0x00	0xFF
PFS	2	0x4	8-9	P00%sPFS	P00%s Pin Function Control Register	0x020	32	read/write	0x00000000	0xFFFFFFFF
PFS	2	0x4	8-9	P00%sPFS_HA	P00%s Pin Function Control Register	0x022	16	read/write	0x0000	0xFFFF
PFS	2	0x4	8-9	P00%sPFS_BY	P00%s Pin Function Control Register	0x023	8	read/write	0x00	0xFF
PFS	6	0x4	10-15	P0%sPFS	P0%s Pin Function Control Register	0x028	32	read/write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P0%sPFS_HA	P0%s Pin Function Control Register	0x02A	16	read/write	0x0000	0xFFFF
PFS	6	0x4	10-15	P0%sPFS_BY	P0%s Pin Function Control Register	0x02B	8	read/write	0x00	0xFF
PFS	-	-	-	P100PFS	P100 Pin Function Control Register	0x040	32	read/write	0x00000000	0xFFFFFFFF
PFS	-	-	-	P100PFS_HA	P100 Pin Function Control Register	0x042	16	read/write	0x0000	0xFFFF
PFS	-	-	-	P100PFS_BY	P100 Pin Function Control Register	0x043	8	read/write	0x00	0xFF
PFS	7	0x4	1-7	P10%sPFS	P10%s Pin Function Control Register	0x044	32	read/write	0x00000000	0xFFFFFFFF
PFS	7	0x4	1-7	P10%sPFS_HA	P10%s Pin Function Control Register	0x046	16	read/write	0x0000	0xFFFF
PFS	7	0x4	1-7	P10%sPFS_BY	P10%s Pin Function Control Register	0x047	8	read/write	0x00	0xFF
PFS	-	-	-	P108PFS	P108 Pin Function Control Register	0x060	32	read/write	0x00010410	0xFFFFFFFF
PFS	-	-	-	P108PFS_HA	P108 Pin Function Control Register	0x062	16	read/write	0x0410	0xFFFF
PFS	-	-	-	P108PFS_BY	P108 Pin Function Control Register	0x063	8	read/write	0x10	0xFF
PFS	-	-	-	P109PFS	P109 Pin Function Control Register	0x064	32	read/write	0x00010410	0xFFFFFFFF

Table 3.3 Register description (9 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	-	-	-	P109PFS_HA	P109 Pin Function Control Register	0x066	16	read/ write	0x0410	0xFFFF
PFS	-	-	-	P109PFS_BY	P109 Pin Function Control Register	0x067	8	read/ write	0x10	0xFF
PFS	-	-	-	P110PFS	P110 Pin Function Control Register	0x068	32	read/ write	0x00010010	0xFFFFFFFF
PFS	-	-	-	P110PFS_HA	P110 Pin Function Control Register	0x06A	16	read/ write	0x0010	0xFFFF
PFS	-	-	-	P110PFS_BY	P110 Pin Function Control Register	0x06B	8	read/ write	0x10	0xFF
PFS	5	0x4	11-15	P1%sPFS	P1%s Pin Function Control Register	0x06C	32	read/ write	0x00000000	0xFFFFFFFF
PFS	5	0x4	11-15	P1%sPFS_HA	P1%s Pin Function Control Register	0x06E	16	read/ write	0x0000	0xFFFF
PFS	5	0x4	11-15	P1%sPFS_BY	P1%s Pin Function Control Register	0x06F	8	read/ write	0x00	0xFF
PFS	-	-	-	P200PFS	P200 Pin Function Control Register	0x080	32	read/ write	0x00000000	0xFFFFFFFF
PFS	-	-	-	P200PFS_HA	P200 Pin Function Control Register	0x082	16	read/ write	0x0000	0xFFFF
PFS	-	-	-	P200PFS_BY	P200 Pin Function Control Register	0x083	8	read/ write	0x00	0xFF
PFS	-	-	-	P201PFS	P201 Pin Function Control Register	0x084	32	read/ write	0x00000010	0xFFFFFFFF
PFS	-	-	-	P201PFS_HA	P201 Pin Function Control Register	0x086	16	read/ write	0x0010	0xFFFF
PFS	-	-	-	P201PFS_BY	P201 Pin Function Control Register	0x087	8	read/ write	0x10	0xFF
PFS	8	0x4	2-9	P20%sPFS	P20%s Pin Function Control Register	0x088	32	read/ write	0x00000000	0xFFFFFFFF
PFS	8	0x4	2-9	P20%sPFS_HA	P20%s Pin Function Control Register	0x08A	16	read/ write	0x0000	0xFFFF
PFS	8	0x4	2-9	P20%sPFS_BY	P20%s Pin Function Control Register	0x08B	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P2%sPFS	P2%s Pin Function Control Register	0x0A8	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P2%sPFS_HA	P2%s Pin Function Control Register	0x0AA	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P2%sPFS_BY	P2%s Pin Function Control Register	0x0AB	8	read/ write	0x00	0xFF
PFS	-	-	-	P300PFS	P300 Pin Function Control Register	0x0C0	32	read/ write	0x00010010	0xFFFFFFFF
PFS	-	-	-	P300PFS_HA	P300 Pin Function Control Register	0x0C2	16	read/ write	0x0010	0xFFFF
PFS	-	-	-	P300PFS_BY	P300 Pin Function Control Register	0x0C3	8	read/ write	0x10	0xFF
PFS	9	0x4	1-9	P30%sPFS	P30%s Pin Function Control Register	0x0C4	32	read/ write	0x00000000	0xFFFFFFFF
PFS	9	0x4	1-9	P30%sPFS_HA	P30%s Pin Function Control Register	0x0C6	16	read/ write	0x0000	0xFFFF
PFS	9	0x4	1-9	P30%sPFS_BY	P30%s Pin Function Control Register	0x0C7	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P3%sPFS	P3%s Pin Function Control Register	0x0E8	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P3%sPFS_HA	P30%s Pin Function Control Register	0x0EA	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P3%sPFS_BY	P30%s Pin Function Control Register	0x0EB	8	read/ write	0x00	0xFF
PFS	10	0x4	0-9	P40%sPFS	P40%s Pin Function Control Register	0x100	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (10 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	10	0x4	0-9	P40%sPFS_HA	P40%s Pin Function Control Register	0x102	16	read/ write	0x0000	0xFFFF
PFS	10	0x4	0-9	P40%sPFS_BY	P40%s Pin Function Control Register	0x103	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P4%sPFS	P4%s Pin Function Control Register	0x128	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P4%sPFS_HA	P4%s Pin Function Control Register	0x12A	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P4%sPFS_BY	P4%s Pin Function Control Register	0x12B	8	read/ write	0x00	0xFF
PFS	10	0x4	0-9	P50%sPFS	P50%s Pin Function Control Register	0x140	32	read/ write	0x00000000	0xFFFFFFFF
PFS	10	0x4	0-9	P50%sPFS_HA	P50%s Pin Function Control Register	0x142	16	read/ write	0x0000	0xFFFF
PFS	10	0x4	0-9	P50%sPFS_BY	P50%s Pin Function Control Register	0x143	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P5%sPFS	P5%s Pin Function Control Register	0x168	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P5%sPFS_HA	P5%s Pin Function Control Register	0x16A	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P5%sPFS_BY	P5%s Pin Function Control Register	0x16B	8	read/ write	0x00	0xFF
PFS	10	0x4	0-9	P60%sPFS	P60%s Pin Function Control Register	0x180	32	read/ write	0x00000000	0xFFFFFFFF
PFS	10	0x4	0-9	P60%sPFS_HA	P60%s Pin Function Control Register	0x182	16	read/ write	0x0000	0xFFFF
PFS	10	0x4	0-9	P60%sPFS_BY	P60%s Pin Function Control Register	0x183	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P6%sPFS	P6%s Pin Function Control Register	0x1A8	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P6%sPFS_HA	P6%s Pin Function Control Register	0x1AA	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P6%sPFS_BY	P6%s Pin Function Control Register	0x1AB	8	read/ write	0x00	0xFF
PFS	10	0x4	0-9	P70%sPFS	P70%s Pin Function Control Register	0x1C0	32	read/ write	0x00000000	0xFFFFFFFF
PFS	10	0x4	0-9	P70%sPFS_HA	P70%s Pin Function Control Register	0x1C2	16	read/ write	0x0000	0xFFFF
PFS	10	0x4	0-9	P70%sPFS_BY	P70%s Pin Function Control Register	0x1C3	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P7%sPFS	P7%s Pin Function Control Register	0x1E8	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P7%sPFS_HA	P7%s Pin Function Control Register	0x1EA	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P7%sPFS_BY	P7%s Pin Function Control Register	0x1EB	8	read/ write	0x00	0xFF
PFS	10	0x4	0-9	P80%sPFS	P80%s Pin Function Control Register	0x200	32	read/ write	0x00000000	0xFFFFFFFF
PFS	10	0x4	0-9	P80%sPFS_HA	P80%s Pin Function Control Register	0x202	16	read/ write	0x0000	0xFFFF
PFS	10	0x4	0-9	P80%sPFS_BY	P80%s Pin Function Control Register	0x203	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P8%sPFS	P8%s Pin Function Control Register	0x228	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P8%sPFS_HA	P8%s Pin Function Control Register	0x22A	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P8%sPFS_BY	P8%s Pin Function Control Register	0x22B	8	read/ write	0x00	0xFF
PFS	10	0x4	0-9	P90%sPFS	P90%s Pin Function Control Register	0x240	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (11 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	10	0x4	0-9	P90%sPFS_HA	P90%s Pin Function Control Register	0x242	16	read/ write	0x0000	0xFFFF
PFS	10	0x4	0-9	P90%sPFS_BY	P90%s Pin Function Control Register	0x243	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	P9%sPFS	P9%s Pin Function Control Register	0x268	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	P9%sPFS_HA	P9%s Pin Function Control Register	0x26A	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	P9%sPFS_BY	P9%s Pin Function Control Register	0x26B	8	read/ write	0x00	0xFF
PFS	10	0x4	0-9	PA0%sPFS	PA0%s Pin Function Control Register	0x280	32	read/ write	0x00000000	0xFFFFFFFF
PFS	10	0x4	0-9	PA0%sPFS_HA	PA0%s Pin Function Control Register	0x282	16	read/ write	0x0000	0xFFFF
PFS	10	0x4	0-9	PA0%sPFS_BY	PA0%s Pin Function Control Register	0x283	8	read/ write	0x00	0xFF
PFS	6	0x4	10-15	PA%sPFS	PA%s Pin Function Control Register	0x2A8	32	read/ write	0x00000000	0xFFFFFFFF
PFS	6	0x4	10-15	PA%sPFS_HA	PA%s Pin Function Control Register	0x2AA	16	read/ write	0x0000	0xFFFF
PFS	6	0x4	10-15	PA%sPFS_BY	PA%s Pin Function Control Register	0x2AB	8	read/ write	0x00	0xFF
PFS	8	0x4	0-7	PB0%sPFS	PB0%s Pin Function Control Register	0x2C0	32	read/ write	0x00000000	0xFFFFFFFF
PFS	8	0x4	0-7	PB0%sPFS_HA	PB0%s Pin Function Control Register	0x2C2	16	read/ write	0x0000	0xFFFF
PFS	8	0x4	0-7	PB0%sPFS_BY	PB0%s Pin Function Control Register	0x2C3	8	read/ write	0x00	0xFF
PMISC	-	-	-	PFENET	Ethernet Control Register	0x00	8	read/ write	0x00	0xFF
PMISC	-	-	-	PWPR	Write-Protect Register	0x03	8	read/ write	0x80	0xFF
ELC	-	-	-	ELCR	Event Link Controller Register	0x00	8	read/ write	0x00	0xFF
ELC	2	0x2	0,1	ELSEGR%s	Event Link Software Event Generation Register %s	0x02	8	read/ write	0x80	0xFF
ELC	19	0x4	0-18	ELSR%s	Event Link Setting Register %s	0x10	16	read/ write	0x0000	0xFFFF
POEG	4	0x100	A,B,C ,D	POEGG%s	POEG Group %s Setting Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
RTC	-	-	-	R64CNT	64-Hz Counter	0x00	8	read- only	0x00	0x80
RTC	-	-	-	RSECCNT	Second Counter	0x02	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT0	Binary Counter 0	0x02	8	read/ write	0x00	0x00
RTC	-	-	-	RMINCNT	Minute Counter	0x04	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT1	Binary Counter 1	0x04	8	read/ write	0x00	0x00
RTC	-	-	-	RHRCNT	Hour Counter	0x06	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT2	Binary Counter 2	0x06	8	read/ write	0x00	0x00
RTC	-	-	-	RWKCNT	Day-of-Week Counter	0x08	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT3	Binary Counter 3	0x08	8	read/ write	0x00	0x00
RTC	-	-	-	RDAYCNT	Day Counter	0x0A	8	read/ write	0x00	0xC0

Table 3.3 **Register description (12 of 44)**

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
RTC	-	-	-	RMONCNT	Month Counter	0x0C	8	read/ write	0x00	0xE0
RTC	-	-	-	RYRCNT	Year Counter	0x0E	16	read/ write	0x0000	0xFF00
RTC	-	-	-	RSECAR	Second Alarm Register	0x10	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT0AR	Binary Counter 0 Alarm Register	0x10	8	read/ write	0x00	0x00
RTC	-	-	-	RMINAR	Minute Alarm Register	0x12	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT1AR	Binary Counter 1 Alarm Register	0x12	8	read/ write	0x00	0x00
RTC	-	-	-	RHRAR	Hour Alarm Register	0x14	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT2AR	Binary Counter 2 Alarm Register	0x14	8	read/ write	0x00	0x00
RTC	-	-	-	RWKAR	Day-of-Week Alarm Register	0x16	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT3AR	Binary Counter 3 Alarm Register	0x16	8	read/ write	0x00	0x00
RTC	-	-	-	RDAYAR	Date Alarm Register	0x18	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT0AER	Binary Counter 0 Alarm Enable Register	0x18	8	read/ write	0x00	0x00
RTC	-	-	-	RMONAR	Month Alarm Register	0x1A	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT1AER	Binary Counter 1 Alarm Enable Register	0x1A	8	read/ write	0x00	0x00
RTC	-	-	-	RYRAR	Year Alarm Register	0x1C	16	read/ write	0x0000	0xFF00
RTC	-	-	-	BCNT2AER	Binary Counter 2 Alarm Enable Register	0x1C	16	read/ write	0x0000	0xFF00
RTC	-	-	-	RYRAREN	Year Alarm Enable Register	0x1E	8	read/ write	0x00	0x00
RTC	-	-	-	BCNT3AER	Binary Counter 3 Alarm Enable Register	0x1E	8	read/ write	0x00	0x00
RTC	-	-	-	RCR1	RTC Control Register 1	0x22	8	read/ write	0x00	0x0A
RTC	-	-	-	RCR2	RTC Control Register 2	0x24	8	read/ write	0x00	0x0E
RTC	-	-	-	RCR4	RTC Control Register 4	0x28	8	read/ write	0x00	0xFE
RTC	-	-	-	RFRH	Frequency Register H	0x2A	16	read/ write	0x0000	0xFFFE
RTC	-	-	-	RFRL	Frequency Register L	0x2C	16	read/ write	0x0000	0x0000
RTC	-	-	-	RADJ	Time Error Adjustment Register	0x2E	8	read/ write	0x00	0x00
RTC	3	0x2	0-2	RTCCR%s	Time Capture Control Register %s	0x40	8	read/ write	0x00	0x00
RTC	3	0x10	0-2	RSECCP%s	Second Capture Register %s	0x52	8	read-only	0x00	0x00
RTC	3	0x10	0-2	BCNT0CP%s	BCNT0 Capture Register %s	0x52	8	read-only	0x00	0x00
RTC	3	0x10	0-2	RMINCP%s	Minute Capture Register %s	0x54	8	read-only	0x00	0x00
RTC	3	0x10	0-2	BCNT1CP%s	BCNT1 Capture Register %s	0x54	8	read-only	0x00	0x00
RTC	3	0x10	0-2	RHRCP%s	Hour Capture Register %s	0x56	8	read-only	0x00	0x00

Table 3.3 Register description (13 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
RTC	3	0x10	0-2	BCNT2CP%s	BCNT2 Capture Register %s	0x56	8	read-only	0x00	0x00
RTC	3	0x10	0-2	RDAYCP%s	Date Capture Register %s	0x5A	8	read-only	0x00	0x00
RTC	3	0x10	0-2	BCNT3CP%s	BCNT3 Capture Register %s	0x5A	8	read-only	0x00	0x00
RTC	3	0x10	0-2	RMONCP%s	Month Capture Register %s	0x5C	8	read-only	0x00	0x00
WDT	-	-	-	WDTRR	WDT Refresh Register	0x00	8	read/ write	0xFF	0xFF
WDT	-	-	-	WDTCR	WDT Control Register	0x02	16	read/ write	0x33F3	0xFFFF
WDT	-	-	-	WDTSR	WDT Status Register	0x04	16	read/ write	0x0000	0xFFFF
WDT	-	-	-	WDTRCR	WDT Reset Control Register	0x06	8	read/ write	0x80	0xFF
WDT	-	-	-	WDTCSTPR	WDT Count Stop Control Register	0x08	8	read/ write	0x80	0xFF
IWDT	-	-	-	IWDTRR	IWDT Refresh Register	0x00	8	read/ write	0xFF	0xFF
IWDT	-	-	-	IWDTSR	IWDT Status Register	0x04	16	read/ write	0x0000	0xFFFF
CAC	-	-	-	CACR0	CAC Control Register 0	0x00	8	read/ write	0x00	0xFF
CAC	-	-	-	CACR1	CAC Control Register 1	0x01	8	read/ write	0x00	0xFF
CAC	-	-	-	CACR2	CAC Control Register 2	0x02	8	read/ write	0x00	0xFF
CAC	-	-	-	CAICR	CAC Interrupt Control Register	0x03	8	read/ write	0x00	0xFF
CAC	-	-	-	CASTR	CAC Status Register	0x04	8	read-only	0x00	0xFF
CAC	-	-	-	CAULVR	CAC Upper-Limit Value Setting Register	0x06	16	read/ write	0x0000	0xFFFF
CAC	-	-	-	CALLVR	CAC Lower-Limit Value Setting Register	0x08	16	read/ write	0x0000	0xFFFF
CAC	-	-	-	CACNTBR	CAC Counter Buffer Register	0x0A	16	read-only	0x0000	0xFFFF
MSTP	-	-	-	MSTPCRB	Module Stop Control Register B	0x00	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
MSTP	-	-	-	MSTPCRC	Module Stop Control Register C	0x04	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
MSTP	-	-	-	MSTPCRD	Module Stop Control Register D	0x08	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
SRCRAM	5552	0x4	0-5551	SRCFCTR[%s]	Filter Coefficient Table [%s]	0x00	32	read/ write	0x00000000	0xFFC00000
SRC	-	-	-	SRCID	Input Data Register	0x00	32	write-only	0x00000000	0xFFFFFFFF
SRC	-	-	-	SRCOD	Output Data Register	0x04	32	read-only	0x00000000	0xFFFFFFFF
SRC	-	-	-	SRCIDCTRL	Input Data Control Register	0x08	16	read/ write	0x0000	0xFFFF
SRC	-	-	-	SRCODCTRL	Output Data Control Register	0x0A	16	read/ write	0x0000	0xFFFF
SRC	-	-	-	SRCCTRL	Control Register	0x0C	16	read/ write	0x0000	0xFFFF
SRC	-	-	-	SRCSTAT	Status Register	0x0E	16	read/ write	0x0002	0xFFFF
SSIE0,1	-	-	-	SSICR	Control Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (14 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SSIE0,1	-	-	-	SSISR	Status Register	0x04	32	read/ write	0x02000013	0x3E00007F
SSIE0,1	-	-	-	SSIFCR	FIFO Control Register	0x10	32	read/ write	0x00000000	0xFFFFFFFF
SSIE0,1	-	-	-	SSIFSR	FIFO Status Register	0x14	32	read/ write	0x00010000	0xFFFFFFFF
SSIE0,1	-	-	-	SSIFTDR	Transmit FIFO Data Register	0x18	32	write-only	0x00000000	0x00000000
SSIE0,1	-	-	-	SSIFRDR	Receive FIFO Data Register	0x1C	32	read-only	0x00000000	0x00000000
SSIE0,1	-	-	-	SSIOFR	Audio Format Register	0x20	32	read/ write	0x00000000	0xFFFFFFFF
SSIE0,1	-	-	-	SSISCR	Status Control Register	0x24	32	read/ write	0x00000000	0xFFFFFFFF
CAN0,1	32	0x10	0-31	MB%s_ID	Mailbox Register	0x200	32	read/ write	0x00000000	0x00000000
CAN0,1	32	0x10	0-31	MB%s_DL	Mailbox Register	0x204	16	read/ write	0x0000	0x0000
CAN0,1	32	0x10	0-31	MB%s_D0	Mailbox Register	0x206	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_D1	Mailbox Register	0x207	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_D2	Mailbox Register	0x208	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_D3	Mailbox Register	0x209	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_D4	Mailbox Register	0x20A	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_D5	Mailbox Register	0x20B	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_D6	Mailbox Register	0x20C	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_D7	Mailbox Register	0x20D	8	read/ write	0x00	0x00
CAN0,1	32	0x10	0-31	MB%s_TS	Mailbox Register	0x20E	16	read/ write	0x0000	0x0000
CAN0,1	8	0x4	0-7	MKR[%s]	Mask Register	0x400	32	read/ write	0x00000000	0x00000000
CAN0,1	2	0x4	0,1	FIDCR%s	FIFO Received ID Compare Registers	0x420	32	read/ write	0x00000000	0x00000000
CAN0,1	-	-	-	MKIVLR	Mask Invalid Register	0x428	32	read/ write	0x00000000	0x00000000
CAN0,1	-	-	-	MIER	Mailbox Interrupt Enable Register	0x42C	32	read/ write	0x00000000	0x00000000
CAN0,1	-	-	-	MIER_FIFO	Mailbox Interrupt Enable Register for FIFO Mailbox Mode	0x42C	32	read/ write	0x00000000	0x00000000
CAN0,1	32	0x1	0-31	MCTL_TX[%s]	Message Control Register for Transmit	0x820	8	read/ write	0x00	0xFF
CAN0,1	32	0x1	0-31	MCTL_RX[%s]	Message Control Register for Receive	0x820	8	read/ write	0x00	0xFF
CAN0,1	-	-	-	CTLR	Control Register	0x840	16	read/ write	0x0500	0xFFFF
CAN0,1	-	-	-	STR	Status Register	0x842	16	read-only	0x0500	0xFFFF
CAN0,1	-	-	-	BCR	Bit Configuration Register	0x844	32	read/ write	0x00000000	0xFFFFFFFF
CAN0,1	-	-	-	RFCR	Receive FIFO Control Register	0x848	8	read/ write	0x80	0xFF

Table 3.3 Register description (15 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CAN0,1	-	-	-	RFPCR	Receive FIFO Pointer Control Register	0x849	8	write-only	0x00	0x00
CAN0,1	-	-	-	TFCR	Transmit FIFO Control Register	0x84A	8	read/ write	0x80	0xFF
CAN0,1	-	-	-	TFPCR	Transmit FIFO Pointer Control Register	0x84B	8	write-only	0x00	0x00
CAN0,1	-	-	-	EIER	Error Interrupt Enable Register	0x84C	8	read/ write	0x00	0xFF
CAN0,1	-	-	-	EIFR	Error Interrupt Factor Judge Register	0x84D	8	read/ write	0x00	0xFF
CAN0,1	-	-	-	RECR	Receive Error Count Register	0x84E	8	read-only	0x00	0xFF
CAN0,1	-	-	-	TECR	Transmit Error Count Register	0x84F	8	read-only	0x00	0xFF
CAN0,1	-	-	-	ECSR	Error Code Store Register	0x850	8	read/ write	0x00	0xFF
CAN0,1	-	-	-	CSSR	Channel Search Support Register	0x851	8	read/ write	0x00	0x00
CAN0,1	-	-	-	MSSR	Mailbox Search Status Register	0x852	8	read-only	0x80	0xFF
CAN0,1	-	-	-	MSMR	Mailbox Search Mode Register	0x853	8	read/ write	0x00	0xFF
CAN0,1	-	-	-	TSR	Time Stamp Register	0x854	16	read-only	0x0000	0xFFFF
CAN0,1	-	-	-	AFSR	Acceptance Filter Support Register	0x856	16	read/ write	0x0000	0x0000
CAN0,1	-	-	-	TCR	Test Control Register	0x858	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICCR1	I2C Bus Control Register 1	0x00	8	read/ write	0x1F	0xFF
IIC0	-	-	-	ICCR2	I2C Bus Control Register 2	0x01	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICMR1	I2C Bus Mode Register 1	0x02	8	read/ write	0x08	0xFF
IIC0	-	-	-	ICMR2	I2C Bus Mode Register 2	0x03	8	read/ write	0x06	0xFF
IIC0	-	-	-	ICMR3	I2C Bus Mode Register 3	0x04	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICFER	I2C Bus Function Enable Register	0x05	8	read/ write	0x72	0xFF
IIC0	-	-	-	ICSER	I2C Bus Status Enable Register	0x06	8	read/ write	0x09	0xFF
IIC0	-	-	-	ICIER	I2C Bus Interrupt Enable Register	0x07	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICSR1	I2C Bus Status Register 1	0x08	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICSR2	I2C Bus Status Register 2	0x09	8	read/ write	0x00	0xFF
IIC0	3	0x2	0-2	SARL%s	Slave Address Register L%s	0x0A	8	read/ write	0x00	0xFF
IIC0	3	0x2	0-2	SARU%s	Slave Address Register U%s	0x0B	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICBRL	I2C Bus Bit Rate Low-Level Register	0x10	8	read/ write	0xFF	0xFF
IIC0	-	-	-	ICBRH	I2C Bus Bit Rate High-Level Register	0x11	8	read/ write	0xFF	0xFF
IIC0	-	-	-	ICDRT	I2C Bus Transmit Data Register	0x12	8	read/ write	0xFF	0xFF
IIC0	-	-	-	ICDRR	I2C Bus Receive Data Register	0x13	8	read-only	0x00	0xFF

Table 3.3 Register description (16 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
IIC0	-	-	-	ICWUR	I2C Bus Wake Up Unit Register	0x16	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICWUR2	I2C Bus Wake Up Unit Register 2	0x17	8	read-only	0x03	0xFF
IIC1,2	-	-	-	ICCR1	I2C Bus Control Register 1	0x00	8	read/ write	0x1F	0xFF
IIC1,2	-	-	-	ICCR2	I2C Bus Control Register 2	0x01	8	read/ write	0x00	0xFF
IIC1,2	-	-	-	ICMR1	I2C Bus Mode Register 1	0x02	8	read/ write	0x08	0xFF
IIC1,2	-	-	-	ICMR2	I2C Bus Mode Register 2	0x03	8	read/ write	0x06	0xFF
IIC1,2	-	-	-	ICMR3	I2C Bus Mode Register 3	0x04	8	read/ write	0x00	0xFF
IIC1,2	-	-	-	ICFER	I2C Bus Function Enable Register	0x05	8	read/ write	0x72	0xFF
IIC1,2	-	-	-	ICSER	I2C Bus Status Enable Register	0x06	8	read/ write	0x09	0xFF
IIC1,2	-	-	-	ICIER	I2C Bus Interrupt Enable Register	0x07	8	read/ write	0x00	0xFF
IIC1,2	-	-	-	ICSR1	I2C Bus Status Register 1	0x08	8	read/ write	0x00	0xFF
IIC1,2	-	-	-	ICSR2	I2C Bus Status Register 2	0x09	8	read/ write	0x00	0xFF
IIC1,2	3	0x2	0-2	SARL%s	Slave Address Register L%s	0x0A	8	read/ write	0x00	0xFF
IIC1,2	3	0x2	0-2	SARU%s	Slave Address Register U%s	0x0B	8	read/ write	0x00	0xFF
IIC1,2	-	-	-	ICBRL	I2C Bus Bit Rate Low-Level Register	0x10	8	read/ write	0xFF	0xFF
IIC1,2	-	-	-	ICBRH	I2C Bus Bit Rate High-Level Register	0x11	8	read/ write	0xFF	0xFF
IIC1,2	-	-	-	ICDRT	I2C Bus Transmit Data Register	0x12	8	read/ write	0xFF	0xFF
IIC1,2	-	-	-	ICDRR	I2C Bus Receive Data Register	0x13	8	read-only	0x00	0xFF
DOC	-	-	-	DOCR	DOC Control Register	0x00	8	read/ write	0x00	0xFF
DOC	-	-	-	DODIR	DOC Data Input Register	0x02	16	read/ write	0x0000	0xFFFF
DOC	-	-	-	DODSR	DOC Data Setting Register	0x04	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCSR	A/D Control Register	0x000	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADANSA0	A/D Channel Select Register A0	0x004	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADANSA1	A/D Channel Select Register A1	0x006	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADADS0	A/D-Converted Value Addition/Average Channel Select Register 0	0x008	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADADS1	A/D-Converted Value Addition/Average Channel Select Register 1	0x00A	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADADC	A/D-Converted Value Addition/Average Count Select Register	0x00C	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADCER	A/D Control Extended Register	0x00E	16	read/ write	0x0000	0xFFFF

Table 3.3 Register description (17 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC120	-	-	-	ADSTRGR	A/D Conversion Start Trigger Select Register	0x010	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADEXICR	A/D Conversion Extended Input Control Register	0x012	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADANSB0	A/D Channel Select Register B0	0x014	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADANSB1	A/D Channel Select Register B1	0x016	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADDBLDR	A/D Data Duplication Register	0x018	16	read-only	0x0000	0xFFFF
ADC120	-	-	-	ADTSDR	A/D Temperature Sensor Data Register	0x01A	16	read-only	0x0000	0xFFFF
ADC120	-	-	-	ADOCDR	A/D Internal Reference Voltage Data Register	0x01C	16	read-only	0x0000	0xFFFF
ADC120	-	-	-	ADRD	A/D Self-Diagnosis Data Register	0x01E	16	read-only	0x0000	0xFFFF
ADC120	8	0x2	0-7	ADDR%s	A/D Data Register %s	0x020	16	read-only	0x0000	0xFFFF
ADC120	5	0x2	16-20	ADDR%s	A/D Data Register %s	0x040	16	read-only	0x0000	0xFFFF
ADC120	-	-	-	ADSHCR	A/D Sample and Hold Circuit Control Register	0x066	16	read/ write	0x0018	0xFFFF
ADC120	-	-	-	ADDISCR	A/D Disconnection Detection Control Register	0x07A	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADSHMSR	A/D Sample and Hold Operation Mode Select Register	0x07C	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADGSPCR	A/D Group Scan Priority Control Register	0x080	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADDBLDRA	A/D Data Duplication Register A	0x084	16	read-only	0x0000	0xFFFF
ADC120	-	-	-	ADDBLDRB	A/D Data Duplication Register B	0x086	16	read-only	0x0000	0xFFFF
ADC120	-	-	-	ADWINMON	A/D Compare Function Window A/B Status Monitor Register	0x08C	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADCMPCR	A/D Compare Function Control Register	0x090	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPANSER	A/D Compare Function Window A Extended Input Select Register	0x092	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADCMPLER	A/D Compare Function Window A Extended Input Comparison Condition Setting Register	0x093	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADCMPANSR0	A/D Compare Function Window A Channel Select Register 0	0x094	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPANSR1	A/D Compare Function Window A Channel Select Register 1	0x096	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPLR0	A/D Compare Function Window A Comparison Condition Setting Register 0	0x098	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPLR1	A/D Compare Function Window A Comparison Condition Setting Register 1	0x09A	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPDR0	A/D Compare Function Window A Lower-Side Level Setting Register	0x09C	16	read/ write	0x0000	0xFFFF

Table 3.3 Register description (18 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC120	-	-	-	ADCMPDR1	A/D Compare Function Window A Upper-Side Level Setting Register	0x09E	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPSR0	A/D Compare Function Window A Channel Status Register 0	0x0A0	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPSR1	A/D Compare Function Window A Channel Status Register 1	0x0A2	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPSER	A/D Compare Function Window A Extended Input Channel Status Register	0x0A4	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADCMPBNSR	A/D Compare Function Window B Channel Selection Register	0x0A6	8	read/ write	0x00	0xFF
ADC120	-	-	-	ADWINLLB	A/D Compare Function Window B Lower-Side Level Setting Register	0x0A8	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADWINULB	A/D Compare Function Window B Upper-Side Level Setting Register	0x0AA	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADCMPBSR	A/D Compare Function Window B Status Register	0x0AC	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADSSTRL	A/D Sampling State Register L	0x0DD	8	read/ write	0x0B	0xFF
ADC120	-	-	-	ADSSTRT	A/D Sampling State Register T	0x0DE	8	read/ write	0x0B	0xFF
ADC120	-	-	-	ADSSTRO	A/D Sampling State Register O	0x0DF	8	read/ write	0x0B	0xFF
ADC120	8	0x1	0-7	ADSSTR0%s	A/D Sampling State Register %s (Corresponding Channel is AN00%s)	0x0E0	8	read/ write	0x0B	0xFF
ADC120	-	-	-	ADPGACR	A/D Programmable Gain Amplifier Control Register	0x1A0	16	read/ write	0x9999	0xFFFF
ADC120	-	-	-	ADPGAGS0	A/D Programmable Gain Amplifier Gain Setting Register 0	0x1A2	16	read/ write	0x0000	0xFFFF
ADC120	-	-	-	ADPGADCR0	A/D Programmable Gain Amplifier Differential Input Control Register	0x1B0	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCSR	A/D Control Register	0x000	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADANSA0	A/D Channel Select Register A0	0x004	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADANSA1	A/D Channel Select Register A1	0x006	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADADS0	A/D-Converted Value Addition/Average Channel Select Register 0	0x008	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADADS1	A/D-Converted Value Addition/Average Channel Select Register 1	0x00A	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADADC	A/D-Converted Value Addition/Average Count Select Register	0x00C	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADCER	A/D Control Extended Register	0x00E	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADSTRGR	A/D Conversion Start Trigger Select Register	0x010	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADEXICR	A/D Conversion Extended Input Control Register	0x012	16	read/ write	0x0000	0xFFFF

Table 3.3 Register description (19 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC121	-	-	-	ADANSB0	A/D Channel Select Register B0	0x014	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADANSB1	A/D Channel Select Register B1	0x016	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADDBLDR	A/D Data Duplication Register	0x018	16	read-only	0x0000	0xFFFF
ADC121	-	-	-	ADTSDR	A/D Temperature Sensor Data Register	0x01A	16	read-only	0x0000	0xFFFF
ADC121	-	-	-	ADOCDR	A/D Internal Reference Voltage Data Register	0x01C	16	read-only	0x0000	0xFFFF
ADC121	-	-	-	ADRD	A/D Self-Diagnosis Data Register	0x01E	16	read-only	0x0000	0xFFFF
ADC121	4	0x2	0-3	ADDR%s	A/D Data Register %s	0x020	16	read-only	0x0000	0xFFFF
ADC121	3	0x2	5-7	ADDR%s	A/D Data Register %s	0x02A	16	read-only	0x0000	0xFFFF
ADC121	4	0x2	16-19	ADDR%s	A/D Data Register %s	0x040	16	read-only	0x0000	0xFFFF
ADC121	-	-	-	ADSHCR	A/D Sample and Hold Circuit Control Register	0x066	16	read/ write	0x0018	0xFFFF
ADC121	-	-	-	ADDISCR	A/D Disconnection Detection Control Register	0x07A	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADSHMSR	A/D Sample and Hold Operation Mode Select Register	0x07C	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADGSPCR	A/D Group Scan Priority Control Register	0x080	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADDBLDRA	A/D Data Duplication Register A	0x084	16	read-only	0x0000	0xFFFF
ADC121	-	-	-	ADDBLDRB	A/D Data Duplication Register B	0x086	16	read-only	0x0000	0xFFFF
ADC121	-	-	-	ADWINMON	A/D Compare Function Window A/B Status Monitor Register	0x08C	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADCMPCR	A/D Compare Function Control Register	0x090	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPANSER	A/D Compare Function Window A Extended Input Select Register	0x092	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADCMPLER	A/D Compare Function Window A Extended Input Comparison Condition Setting Register	0x093	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADCMPANSR0	A/D Compare Function Window A Channel Select Register 0	0x094	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPANSR1	A/D Compare Function Window A Channel Select Register 1	0x096	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPLR0	A/D Compare Function Window A Comparison Condition Setting Register 0	0x098	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPLR1	A/D Compare Function Window A Comparison Condition Setting Register 1	0x09A	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPDR0	A/D Compare Function Window A Lower-Side Level Setting Register	0x09C	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPDR1	A/D Compare Function Window A Upper-Side Level Setting Register	0x09E	16	read/ write	0x0000	0xFFFF

Table 3.3 Register description (20 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC121	-	-	-	ADCMPSR0	A/D Compare Function Window A Channel Status Register 0	0x0A0	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPSR1	A/D Compare Function Window A Channel Status Register 1	0x0A2	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPSER	A/D Compare Function Window A Extended Input Channel Status Register	0x0A4	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADCMPBNSR	A/D Compare Function Window B Channel Selection Register	0x0A6	8	read/ write	0x00	0xFF
ADC121	-	-	-	ADWINLLB	A/D Compare Function Window B Lower-Side Level Setting Register	0x0A8	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADWINULB	A/D Compare Function Window B Upper-Side Level Setting Register	0x0AA	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADCMPBSR	A/D Compare Function Window B Status Register	0x0AC	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADSSTRL	A/D Sampling State Register L	0x0DD	8	read/ write	0x0B	0xFF
ADC121	-	-	-	ADSSTRT	A/D Sampling State Register T	0x0DE	8	read/ write	0x0B	0xFF
ADC121	-	-	-	ADSSTRO	A/D Sampling State Register O	0x0DF	8	read/ write	0x0B	0xFF
ADC121	4	0x1	0-3	ADSSTR0%s	A/D Sampling State Register %s (Corresponding Channel is AN10%s)	0x0E0	8	read/ write	0x0B	0xFF
ADC121	3	0x1	5-7	ADSSTR0%s	A/D Sampling State Register %s (Corresponding Channel is AN10%s)	0x0E5	8	read/ write	0x0B	0xFF
ADC121	-	-	-	ADPGACR	A/D Programmable Gain Amplifier Control Register	0x1A0	16	read/ write	0x9999	0xFFFF
ADC121	-	-	-	ADPGAGS0	A/D Programmable Gain Amplifier Gain Setting Register 0	0x1A2	16	read/ write	0x0000	0xFFFF
ADC121	-	-	-	ADPGADCR0	A/D Programmable Gain Amplifier Differential Input Control Register	0x1B0	16	read/ write	0x0000	0xFFFF
TSN	-	-	-	TSCR	Temperature Sensor Control Register	0x00	8	read/ write	0x00	0xFF
DAC12	2	0x2	0,1	DADR%s	D/A Data Register %s	0x00	16	read/ write	0x0000	0xFFFF
DAC12	-	-	-	DACR	D/A Control Register	0x0004	8	read/ write	0x1F	0xFF
DAC12	-	-	-	DADPR	DADRm Format Select Register	0x0005	8	read/ write	0x00	0xFF
DAC12	-	-	-	DAADSCR	D/A-A/D Synchronous Start Control Register	0x0006	8	read/ write	0x00	0xFF
DAC12	-	-	-	DAAMPCR	D/A Output Amplifier Control Register	0x0008	8	read/ write	0x00	0xFF
DAC12	-	-	-	DAASWCR	D/A Amplifier Stabilization Wait Control Register	0x101C	8	read/ write	0x00	0xFF
DAC12	-	-	-	DAADUSR	D/A A/D Synchronous Unit Select Register	0xC0	8	read/ write	0x00	0xFF
USBHS	-	-	-	SYSCFG	System Configuration Control Register	0x000	16	read/ write	0x0020	0xFFFF
USBHS	-	-	-	BUSWAIT	CPU Bus Wait Register	0x002	16	read/ write	0x000F	0x3F3F

Table 3.3 Register description (21 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBHS	-	-	-	SYSSTS0	System Configuration Status Register	0x004	16	read-only	0x0000	0x0000
USBHS	-	-	-	PLLSTA	PLL Status Register	0x006	16	read-only	0x0000	0x0001
USBHS	-	-	-	DVSTCTR0	Device State Control Register 0	0x008	16	read/write	0x0000	0x07F7
USBHS	-	-	-	TESTMODE	USB Test Mode Register	0x00C	16	read/write	0x0000	0x000F
USBHS	-	-	-	CFIFO	CFIFO Port Register	0x014	32	read/write	0x00000000	0xFFFFFFFF
USBHS	-	-	-	CFIFOL	CFIFO Port Register L	0x014	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	CFIFOLL	CFIFO Port Register LL	0x014	8	read/write	0x00	0xFF
USBHS	-	-	-	CFIFOH	CFIFO Port Register H	0x016	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	CFIFOHH	CFIFO Port Register HH	0x017	8	read/write	0x00	0xFF
USBHS	-	-	-	D0FIFO	D0FIFO Port Register	0x018	32	read/write	0x00000000	0xFFFFFFFF
USBHS	-	-	-	D0FIFOL	D0FIFO Port Register L	0x018	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	D0FIFOLL	D0FIFO Port Register LL	0x018	8	read/write	0x00	0xFF
USBHS	-	-	-	D0FIFOH	D0FIFO Port Register H	0x01A	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	D0FIFOHH	D0FIFO Port Register HH	0x01B	8	read/write	0x00	0xFF
USBHS	-	-	-	D1FIFO	D1FIFO Port Register	0x01C	32	read/write	0x00000000	0xFFFFFFFF
USBHS	-	-	-	D1FIFOL	D1FIFO Port Register L	0x01C	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	D1FIFOLL	D1FIFO Port Register LL	0x01C	8	read/write	0x00	0xFF
USBHS	-	-	-	D1FIFOH	D1FIFO Port Register H	0x01E	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	D1FIFOHH	D1FIFO Port Register HH	0x01F	8	read/write	0x00	0xFF
USBHS	-	-	-	CFIFOSEL	CFIFO Port Select Register	0x020	16	read/write	0x0000	0xCD27
USBHS	-	-	-	CFIFOCTR	CFIFO Port Control Register	0x022	16	read/write	0x0000	0xEFFE
USBHS	-	-	-	D0FIFOSEL	D0FIFO Port Select Register	0x028	16	read/write	0x0000	0xFD07
USBHS	-	-	-	D0FIFOCTR	D0FIFO Port Control Register	0x02A	16	read/write	0x0000	0xEFFE
USBHS	-	-	-	D1FIFOSEL	D1FIFO Port Select Register	0x02C	16	read/write	0x0000	0xFD07
USBHS	-	-	-	D1FIFOCTR	D1FIFO Port Control Register	0x02E	16	read/write	0x0000	0xEFFE
USBHS	-	-	-	INTENB0	Interrupt Enable Register 0	0x030	16	read/write	0x0000	0xFF00
USBHS	-	-	-	INTENB1	Interrupt Enable Register 1	0x032	16	read/write	0x0000	0xDB71
USBHS	-	-	-	BRDYENB	BRDY Interrupt Enable Register	0x036	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	NRDYENB	NRDY Interrupt Enable Register	0x038	16	read/write	0x0000	0xFFFF
USBHS	-	-	-	BEMPENB	BEMP Interrupt Enable Register	0x03A	16	read/write	0x0000	0xFFFF

Table 3.3 Register description (22 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBHS	-	-	-	SOFCFG	SOF Pin Configuration Register	0x03C	16	read/ write	0x0000	0x0170
USBHS	-	-	-	PHYSET	PHY Setting Register	0x03E	16	read/ write	0x0033	0x8B3B
USBHS	-	-	-	INTSTS0	Interrupt Status Register 0	0x040	16	read/ write	0x0000	0xFF7F
USBHS	-	-	-	INTSTS1	Interrupt Status Register 1	0x042	16	read/ write	0x0000	0xDB71
USBHS	-	-	-	BRDYSTS	BRDY Interrupt Status Register	0x046	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	NRDYSTS	NRDY Interrupt Status Register	0x048	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	BEMPSTS	BEMP Interrupt Status Register	0x04A	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	FRMNUM	Frame Number Register	0x04C	16	read/ write	0x0000	0xC7FF
USBHS	-	-	-	UFRMNUM	uFrame Number Register	0x04E	16	read/ write	0x0000	0x0007
USBHS	-	-	-	USBADDR	USB Address Register	0x050	16	read/ write	0x0000	0x007F
USBHS	-	-	-	USBREQ	USB Request Type Register	0x054	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	USBVAL	USB Request Value Register	0x056	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	USBINDX	USB Request Index Register	0x058	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	USBLENG	USB Request Length Register	0x05A	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	DCPCFG	DCP Configuration Register	0x05C	16	read/ write	0x0000	0x0190
USBHS	-	-	-	DCPMAXP	DCP Maximum Packet Size Register	0x05E	16	read/ write	0x0040	0xF07F
USBHS	-	-	-	DCPCTR	DCP Control Register	0x060	16	read/ write	0x0000	0xF1F7
USBHS	-	-	-	PIPESEL	Pipe Window Select Register	0x064	16	read/ write	0x0000	0x000F
USBHS	-	-	-	PIPECFG	Pipe Configuration Register	0x068	16	read/ write	0x0000	0xC79F
USBHS	-	-	-	PIPEBUF	Pipe Buffer Register	0x06A	16	read/ write	0x0000	0x7CFF
USBHS	-	-	-	PIPEMAXP	Pipe Maximum Packet Size Register	0x06C	16	read/ write	0x0000	0xF7FF
USBHS	-	-	-	PIPEPERI	Pipe Cycle Control Register	0x06E	16	read/ write	0x0000	0x1007
USBHS	9	0x002	1-9	PIPE%sCTR	PIPE Control Register	0x070	16	read/ write	0x0000	0xF7E3
USBHS	5	0x004	1-5	PIPE%sTRE	PIPE Transaction Counter Enable Register	0x090	16	read/ write	0x0000	0x0300
USBHS	5	0x004	1-5	PIPE%sTRN	PIPE Transaction Counter Register	0x092	16	read/ write	0x0000	0xFFFF
USBHS	10	0x002	0-9	DEVADD%s	Device Address Configuration Register	0x0D0	16	read/ write	0x0000	0x7FC0
USBHS	-	-	-	DEVADDA	Device Address Configuration Register A	0x0E4	16	read/ write	0x0000	0x7FC0
USBHS	-	-	-	LPCTRL	Low Power Control Register	0x100	16	read/ write	0x0000	0x0081
USBHS	-	-	-	LPSTS	Low Power Status Register	0x102	16	read/ write	0x0000	0x4000
USBHS	-	-	-	BCCTRL	Battery Charging Control Register	0x140	16	read/ write	0x0000	0x033F

Table 3.3 Register description (23 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBHS	-	-	-	PL1CTRL1	Function L1 Control Register 1	0x144	16	read/ write	0x0000	0x4FFF
USBHS	-	-	-	PL1CTRL2	Function L1 Control Register 2	0x146	16	read/ write	0x0000	0x1F00
USBHS	-	-	-	HL1CTRL1	Host L1 Control Register 1	0x148	16	read/ write	0x0000	0x0007
USBHS	-	-	-	HL1CTRL2	Host L1 Control Register 2	0x14A	16	read/ write	0x0000	0x9F0F
USBHS	-	-	-	PHYTRIM1	PHY Timing Register 1	0x150	16	read/ write	0x0605	0x7F8F
USBHS	-	-	-	PHYTRIM2	PHY Timing Register 2	0x152	16	read/ write	0x1106	0x738F
USBHS	-	-	-	DPUSR0R	Deep Standby USB Transceiver Control/Pin Monitor Register	0x160	32	read/ write	0x00000000	0xFF4FFFFF
USBHS	-	-	-	DPUSR1R	Deep Standby USB Suspend/Resume Interrupt Register	0x164	32	read/ write	0x00000000	0xFFFFFFFF
USBHS	-	-	-	DPUSR2R	Deep Standby USB Suspend/Resume Interrupt Register	0x168	16	read/ write	0x0000	0xFFFF
USBHS	-	-	-	DPUSRCR	Deep Standby USB Suspend/Resume Command Register	0x16A	16	read/ write	0x0000	0xFFFF
SDHI0,1	-	-	-	SD_CMD	Command Type Register	0x000	32	read/ write	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_ARG	SD Command Argument Register	0x008	32	read/ write	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_ARG1	SD Command Argument Register 1	0x00C	32	read/ write	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_STOP	Data Stop Register	0x010	32	read/ write	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_SECCNT	Block Count Register	0x014	32	read/ write	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP10	SD Card Response Register 10	0x018	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP1	SD Card Response Register 1	0x01C	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP32	SD Card Response Register 32	0x020	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP3	SD Card Response Register 3	0x024	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP54	SD Card Response Register 54	0x028	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP5	SD Card Response Register 5	0x02C	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP76	SD Card Response Register 76	0x030	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_RSP7	SD Card Response Register 7	0x034	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_INFO1	SD Card Interrupt Flag Register 1	0x038	32	read/ write	0x00000000	0xFFFFFB5F
SDHI0,1	-	-	-	SD_INFO2	SD Card Interrupt Flag Register 2	0x03C	32	read/ write	0x00002000	0xFFFFFF7F
SDHI0,1	-	-	-	SD_INFO1_MASK	SD_INFO1 Interrupt Mask Register	0x040	32	read/ write	0x0000031D	0xFFFFFFFF
SDHI0,1	-	-	-	SD_INFO2_MASK	SD_INFO2 Interrupt Mask Register	0x044	32	read/ write	0x00008B7F	0xFFFFFFFF
SDHI0,1	-	-	-	SD_CLK_CTRL	SD Clock Control Register	0x048	32	read/ write	0x00000020	0xFFFFFFFF

Table 3.3 Register description (24 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SDHI0,1	-	-	-	SD_SIZE	Transfer Data Length Register	0x04C	32	read/ write	0x00000200	0xFFFFFFFF
SDHI0,1	-	-	-	SD_OPTION	SD Card Access Control Option Register	0x050	32	read/ write	0x000040EE	0xFFFFFFFF
SDHI0,1	-	-	-	SD_ERR_STS1	SD Error Status Register 1	0x058	32	read-only	0x00002000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_ERR_STS2	SD Error Status Register 2	0x05C	32	read-only	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SD_BUF0	SD Buffer Register	0x060	32	read/ write	0x00000000	0x00000000
SDHI0,1	-	-	-	SDIO_MODE	SDIO Mode Control Register	0x068	32	read/ write	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	SDIO_INFO1	SDIO Interrupt Flag Register 1	0x06C	32	read/ write	0x00000000	0xFFFFFFF9
SDHI0,1	-	-	-	SDIO_INFO1_MASK	SDIO_INFO1 Interrupt Mask Register	0x070	32	read/ write	0x0000C007	0xFFFFFFFF
SDHI0,1	-	-	-	SD_DMAEN	DMA Mode Enable Register	0x1B0	32	read/ write	0x00001010	0xFFFFFFFF
SDHI0,1	-	-	-	SOFT_RST	Software Reset Register	0x1C0	32	read/ write	0x00000007	0xFFFFFFFF
SDHI0,1	-	-	-	SDIF_MODE	SD Interface Mode Setting Register	0x1CC	32	read/ write	0x00000000	0xFFFFFFFF
SDHI0,1	-	-	-	EXT_SWAP	Swap Control Register	0x1E0	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	EDMR	EDMAC Mode Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	TRIMD	Transmit Interrupt Setting Register	0x07C	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	EDTRR	EDMAC Transmit Request Register	0x08	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	TBRAR	Transmit Buffer Read Address Register	0x0D4	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	EDRRR	EDMAC Receive Request Register	0x10	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	TDLAR	Transmit Descriptor List Start Address Register	0x18	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	RDLAR	Receive Descriptor List Start Address Register	0x20	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	EESR	ETHERC/EDMAC Status Register	0x28	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	EESIPR	ETHERC/EDMAC Status Interrupt Enable Register	0x30	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	TRSCER	ETHERC/EDMAC Transmit/ Receive Status Copy Enable Register	0x38	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	RMFCR	Missed-Frame Counter Register	0x40	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	TFTR	Transmit FIFO Threshold Register	0x48	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	FDR	Transmit FIFO Threshold Register	0x50	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	RMCR	Receive Method Control Register	0x58	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	TFUCR	Transmit FIFO Underflow Counter	0x64	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	RFOCR	Receive FIFO Overflow Counter	0x68	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	IOSR	Independent Output Signal Setting Register	0x6C	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (25 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
EDMAC0	-	-	-	FCFTR	Flow Control Start FIFO Threshold Setting Register	0x70	32	read/ write	0x00070007	0xFFFFFFFF
EDMAC0	-	-	-	RPADIR	Receive Data Padding Insert Register	0x78	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	RBWAR	Receive Buffer Write Address Register	0xC8	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	RDFAR	Receive Descriptor Fetch Address Register	0xCC	32	read/ write	0x00000000	0xFFFFFFFF
EDMAC0	-	-	-	TDFAR	Transmit Descriptor Fetch Address Register	0xD8	32	read-only	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	ECMR	ETHERC Mode Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	RFLR	Receive Frame Maximum Length Register	0x08	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	ECSR	ETHERC Status Register	0x10	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	ECSIPR	ETHERC Interrupt Enable Register	0x18	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	PIR	PHY Interface Register	0x20	32	read/ write	0x00000000	0xFFFFFFF7
ETHERC0	-	-	-	PSR	PHY Status Register	0x28	32	read-only	0x00000000	0xFFFFFFFE
ETHERC0	-	-	-	RDMLR	Random Number Generation Counter Upper Limit Setting Register	0x40	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	IPGR	IPG Register	0x50	32	read/ write	0x00000014	0xFFFFFFFF
ETHERC0	-	-	-	APR	Automatic PAUSE Frame Register	0x54	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	MPR	Manual PAUSE Frame Register	0x58	32	write-only	0x00000000	0xFFFF0000
ETHERC0	-	-	-	RFCF	Received PAUSE Frame Counter	0x60	32	read-only	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	TPAUSER	PAUSE Frame Retransmit Count Setting Register	0x64	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	TPAUSECR	PAUSE Frame Retransmit Counter	0x68	32	read-only	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	BCFRR	Broadcast Frame Receive Count Setting Register	0x6C	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	MAHR	MAC Address Upper Bit Register	0xC0	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	MALR	MAC Address Lower Bit Register	0xC8	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	TROCR	Transmit Retry Over Counter Register	0xD0	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	CDCR	Late Collision Detect Counter Register	0xD4	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	LCCR	Lost Carrier Counter Register	0xD8	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	CNDCR	Carrier Not Detect Counter Register	0xDC	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	CEFCR	CRC Error Frame Receive Counter Register	0xE4	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	FRECR	Frame Receive Error Counter Register	0xE8	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	TSFRCR	Too-Short Frame Receive Counter Register	0xEC	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	TLFRCR	Too-Long Frame Receive Counter Register	0xF0	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 **Register description (26 of 44)**

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ETHERC0	-	-	-	RFCR	Received Alignment Error Frame Counter Register	0xF4	32	read/ write	0x00000000	0xFFFFFFFF
ETHERC0	-	-	-	MAFCR	Multicast Address Frame Receive Counter Register	0xF8	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	EDMR	PTPEDMAC Mode Register	0x000	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	EDTRR	EDMAC Transmit Request Register	0x008	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	EDRRR	EDMAC Receive Request Register	0x010	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	TDLAR	Transmit Descriptor List Start Address Register	0x018	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	RDLAR	Receive Descriptor List Start Address Register	0x020	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	EESR	PTP/EDMAC Status Register	0x028	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	EESIPR	PTP/EDMAC Status Interrupt Enable Register	0x030	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	RMFCR	Missed-Frame Counter Register	0x040	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	TFTR	Transmit FIFO Threshold Register	0x048	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	FDR	Transmit FIFO Threshold Register	0x050	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	RMCR	Receive Method Control Register	0x058	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	TFUCR	Transmit FIFO Underflow Counter	0x064	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	RFOCR	Receive FIFO Overflow Counter	0x068	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	IOSR	Independent Output Signal Setting Register	0x06C	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	FCFTR	Flow Control Start FIFO Threshold Setting Register	0x070	32	read/ write	0x00070007	0xFFFFFFFF
PTPEDMAC	-	-	-	RPADIR	Receive Data Padding Insert Register	0x078	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	TRIMD	Transmit Interrupt Setting Register	0x07C	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	RBWAR	Receive Buffer Write Address Register	0x0C8	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	RDFAR	Receive Descriptor Fetch Address Register	0x0CC	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	TBRAR	Transmit Buffer Read Address Register	0x0D4	32	read/ write	0x00000000	0xFFFFFFFF
PTPEDMAC	-	-	-	TDFAR	Transmit Descriptor Fetch Address Register	0x0D8	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC_CFG	-	-	-	PTRSTR	EPTPC Reset Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC_CFG	-	-	-	STCSELR	STCA Clock Select Register	0x04	32	read/ write	0x00000006	0xFFFFFFFF
EPTPC_CFG	-	-	-	BYPASS	Bypass 1588 module Register	0x08	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MIESR	MINT Interrupt Source Status Register	0x000	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MIEIPR	MINT Interrupt Request Permission Register	0x004	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	ELIPPR	ELC Output/IPLS Interrupt Request Permission Register	0x010	32	read/ write	0x00003F3F	0xFFFFFFFF

Table 3.3 Register description (27 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
EPTPC	-	-	-	ELIPACR	ELC Output/IPLS Interrupt Permission Automatic Clearing Register	0x014	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	STSR	STCA Status Register	0x040	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	STIPR	STCA Status Notification Permission Register	0x044	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	STCFR	STCA Clock Frequency Setting Register	0x050	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	STMR	STCA Operating Mode Register	0x054	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	SYNTOR	Sync Message Reception Timeout Register	0x058	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	IPTSELR	IPLS Interrupt Request Timer Select Register	0x060	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MITSELR	MINT Interrupt Request Timer Select Register	0x064	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	ELTSELR	ELC Output Timer Select Register	0x068	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	STCHSELR	Time Synchronization Channel Select Register	0x06C	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	SYNSTARTR	Slave Time Synchronization Start Register	0x080	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	LCIVLDR	Local Time Counter Initial Value Load Directive Register	0x084	32	write-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	SYNTDARU	Synchronization Loss Detection Threshold Registers	0x090	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	SYNTDARL	Synchronization Loss Detection Threshold Registers	0x094	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	SYNTDBRU	Synchronization Detection Threshold Registers	0x098	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	SYNTDBRL	Synchronization Detection Threshold Registers	0x09C	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	LCIVRU	Local Time Counter Initial Value Registers	0x0B0	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	LCIVRM	Local Time Counter Initial Value Registers	0x0B4	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	LCIVRL	Local Time Counter Initial Value Registers	0x0B8	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	GETW10R	Worst 10 Acquisition Directive Register	0x124	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	PLIMITRU	Positive Gradient Limit Registers	0x128	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	PLIMITRM	Positive Gradient Limit Registers	0x12C	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	PLIMITRL	Positive Gradient Limit Registers	0x130	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MLIMITRU	Negative Gradient Limit Registers	0x134	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MLIMITRM	Negative Gradient Limit Registers	0x138	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MLIMITRL	Negative Gradient Limit Registers	0x13C	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	GETINFOR	Statistical Information Retention Control Register	0x140	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	LCCVRU	Local Time Counters	0x170	32	read-only	0x00000000	0xFFFFFFFF

Table 3.3 Register description (28 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
EPTPC	-	-	-	LCCVRM	Local Time Counters	0x174	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	LCCVRL	Local Time Counters	0x178	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	PW10VRU	Positive Gradient Worst 10 Value Registers	0x210	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	PW10VRM	Positive Gradient Worst 10 Value Registers	0x214	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	PW10VRL	Positive Gradient Worst 10 Value Registers	0x218	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MW10RU	Negative Gradient Worst 10 Value Registers	0x2D0	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MW10RM	Negative Gradient Worst 10 Value Registers	0x2D4	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	MW10RL	Negative Gradient Worst 10 Value Registers	0x2D8	32	read-only	0x00000000	0xFFFFFFFF
EPTPC	6	0x10	0-5	TMSTTRU%s	Timer Start Time Setting Register %s	0x300	32	read/write	0x00000000	0xFFFFFFFF
EPTPC	6	0x10	0-5	TMSTTRL%s	Timer Start Time Setting Register %s	0x304	32	read/write	0x00000000	0xFFFFFFFF
EPTPC	6	0x10	0-5	TMCYCR%s	Timer Cycle Setting Registers %s	0x308	32	read/write	0x00000000	0xFFFFFFFF
EPTPC	6	0x10	0-5	TMPLSR%s	Timer Pulse Width Setting Register %s	0x30C	32	read/write	0x00000000	0xFFFFFFFF
EPTPC	-	-	-	TMSTARTR	Timer Start Register	0x37C	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYSR	SYNFP Status Register	0x000	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYIPR	SYNFP Status Notification Permission Register	0x004	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYMACRU	SYNFP MAC Address Registers	0x010	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYMACRL	SYNFP MAC Address Registers	0x014	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYLLCCTLR	SYNFP LLC-CTL Value Register	0x018	32	read/write	0x00000003	0xFFFFFFFF
EPTPC0	-	-	-	SYIPADDRR	SYNFP Local IP Address Register	0x01C	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYSPVRR	SYNFP Specification Version Setting Register	0x040	32	read/write	0x00000002	0xFFFFFFFF
EPTPC0	-	-	-	SYDOMR	SYNFP Domain Number Setting Register	0x044	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	ANFR	Announce Message Flag Field Setting Register	0x050	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYNFR	Sync Message Flag Field Setting Register	0x054	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	DYRQFR	Delay_Req Message Flag Field Setting Register	0x058	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	DYRPFR	Delay_Resp Message Flag Field Setting Register	0x05C	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYCIDRU	SYNFP Local Clock ID Registers	0x060	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYCIDRL	SYNFP Local Clock ID Registers	0x064	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYPNUMR	SYNFP Local Port Number Register	0x068	32	read/write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYRVLDR	SYNFP Register Value Load Directive Register	0x080	32	write-only	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYRFL1R	SYNFP Reception Filter Register 1	0x090	32	read/write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (29 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
EPTPC0	-	-	-	SYRFL2R	SYNFP Reception Filter Register 2	0x094	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYTRENR	SYNFP Transmission Enable Register	0x098	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	MTCIDU	Master Clock ID Registers	0x0A0	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	MTCIDL	Master Clock ID Registers	0x0A4	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	MTPID	Master clock port number register	0x0A8	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYTLIR	SYNFP Transmission Interval Setting Register	0x0C0	32	read/ write	0x00000001	0xFFFFFFFF
EPTPC0	-	-	-	SYRLIR	SYNFP Received logMessageInterval Value Indication Register	0x0C4	32	read-only	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	OFMRU	offsetFromMaster Value Register	0x0C8	32	read-only	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	OFMRL	offsetFromMaster Value Register	0x0CC	32	read-only	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	MPDRU	meanPathDelay Value Register	0x0D0	32	read-only	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	MPDRL	meanPathDelay Value Register	0x0D4	32	read-only	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	GMPR	grandmasterPriority Field Setting Register	0x0E0	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	GMCQR	grandmasterClockQuality Field Setting Register	0x0E4	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	GMIDRU	grandmasterIdentity Field Setting Register	0x0E8	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	GMIDRL	grandmasterIdentity Field Setting Register	0x0EC	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	CUOTSR	currentUtcOffset/timeSource Field Setting Register	0x0F0	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SRR	stepsRemoved Field Setting Register	0x0F4	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	PPMACRU	PTP-primary Message Destination MAC Address Setting Register	0x100	32	read/ write	0x00011B19	0xFFFFFFFF
EPTPC0	-	-	-	PPMACRL	PTP-primary Message Destination MAC Address Setting Register	0x104	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	PDMACRU	PTP-pdelay Message MAC Address Setting Register	0x108	32	read/ write	0x000180C2	0xFFFFFFFF
EPTPC0	-	-	-	PDMACRL	PTP-pdelay Message MAC Address Setting Register	0x10C	32	read/ write	0x0000000E	0xFFFFFFFF
EPTPC0	-	-	-	PETYPER	PTP Message EtherType Setting Register	0x110	32	read/ write	0x000088F7	0xFFFFFFFF
EPTPC0	-	-	-	PPIPR	PTP-primary Message Destination IP Address Setting Register	0x120	32	read/ write	0xE0000181	0xFFFFFFFF
EPTPC0	-	-	-	PDIPR	PTP-pdelay Message Destination IP Address Setting Register	0x124	32	read/ write	0xE000006B	0xFFFFFFFF
EPTPC0	-	-	-	PETOSR	PTP Event Message TOS Setting Register	0x128	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	PGTOSR	PTP general Message TOS Setting Register	0x12C	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	PPTTLR	PTP-primary Message TTL Setting Register	0x130	32	read/ write	0x00000080	0xFFFFFFFF

Table 3.3 Register description (30 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
EPTPC0	-	-	-	PDTTLR	PTP-pdelay Message TTL Setting Register	0x134	32	read/ write	0x00000001	0xFFFFFFFF
EPTPC0	-	-	-	PEUDPR	PTP Event Message UDP Destination Port Number Setting Register	0x138	32	read/ write	0x0000013F	0xFFFFFFFF
EPTPC0	-	-	-	PGUDPR	PTP general Message UDP Destination Port Number Setting Register	0x13C	32	read/ write	0x00000140	0xFFFFFFFF
EPTPC0	-	-	-	FFLTR	Frame Reception Filter Setting Register	0x140	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	2	0x8	0-1	FMAC%sRU	Frame Reception Filter MAC Address %s Setting Registers	0x160	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	2	0x8	0-1	FMAC%sRL	Frame Reception Filter MAC Address %s Setting Registers	0x164	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	DASYMRU	Asymmetric Delay Setting Registers	0x1C0	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	DASYMRL	Asymmetric Delay Setting Registers	0x1C4	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	TSLATR	Timestamp Latency Setting Register	0x1C8	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYCONFR	SYNFP Operation Setting Register	0x1CC	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	SYFORMR	SYNFP Frame Format Setting Register	0x1D0	32	read/ write	0x00000000	0xFFFFFFFF
EPTPC0	-	-	-	RSTOUTR	Response Message Reception Timeout Register	0x1D4	32	read/ write	0x00000000	0xFFFFFFFF
SCI0-9	-	-	-	SMR	Serial Mode Register (SCMR.SMIF = 0)	0x00	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	SMR_SMCI	Serial mode register (SCMR.SMIF = 1)	0x00	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	BRR	Bit Rate Register	0x01	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	SCR	Serial Control Register (SCMR.SMIF = 0)	0x02	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	SCR_SMCI	Serial Control Register (SCMR.SMIF =1)	0x02	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	TDR	Transmit Data Register	0x03	8	read/ write	0xFF	0xFF
SCI0-9	-	-	-	SSR	Serial Status Register(SCMR.SMIF = 0 and FCR.FM=0)	0x04	8	read/ write	0x84	0xFF
SCI0-9	-	-	-	SSR_FIFO	Serial Status Register(SCMR.SMIF = 0 and FCR.FM=1)	0x04	8	read/ write	0x80	0xFF
SCI0-9	-	-	-	SSR_SMCI	Serial Status Register(SCMR.SMIF = 1)	0x04	8	read/ write	0x84	0xFF
SCI0-9	-	-	-	RDR	Receive Data Register	0x05	8	read-only	0x00	0xFF
SCI0-9	-	-	-	SCMR	Smart Card Mode Register	0x06	8	read/ write	0xF2	0xFF
SCI0-9	-	-	-	SEMR	Serial Extended Mode Register	0x07	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	SNFR	Noise Filter Setting Register	0x08	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	SIMR1	I2C Mode Register 1	0x09	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	SIMR2	I2C Mode Register 2	0x0A	8	read/ write	0x00	0xFF

Table 3.3 Register description (31 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SCI0-9	-	-	-	SIMR3	I2C Mode Register 3	0x0B	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	SISR	I2C Status Register	0x0C	8	read-only	0x00	0xCB
SCI0-9	-	-	-	SPMR	SPI Mode Register	0x0D	8	read/ write	0x00	0xFF
SCI0-9	-	-	-	TDRHL	Transmit 9-bit Data Register	0x0E	16	read/ write	0xFFFF	0xFFFF
SCI0-9	-	-	-	FTDRHL	Transmit FIFO Data Register HL	0x0E	16	write-only	0xFFFF	0xFFFF
SCI0-9	-	-	-	FTDRH	Transmit FIFO Data Register H	0x0E	8	write-only	0xFF	0xFF
SCI0-9	-	-	-	FTDRL	Transmit FIFO Data Register L	0x0F	8	write-only	0xFF	0xFF
SCI0-9	-	-	-	RDRHL	Receive 9-bit Data Register	0x10	16	read-only	0x0000	0xFFFF
SCI0-9	-	-	-	FRDRHL	Receive FIFO Data Register HL	0x10	16	read-only	0x0000	0xFFFF
SCI0-9	-	-	-	FRDRH	Receive FIFO Data Register H	0x10	8	read-only	0x00	0xFF
SCI0-9	-	-	-	FRDRL	Receive FIFO Data Register L	0x11	8	read-only	0x00	0xFF
SCI0-9	-	-	-	MDDR	Modulation Duty Register	0x12	8	read/ write	0xFF	0xFF
SCI0-9	-	-	-	DCCR	Data Compare Match Control Register	0x13	8	read/ write	0x40	0xFF
SCI0-9	-	-	-	FCR	FIFO Control Register	0x14	16	read/ write	0xF800	0xFFFF
SCI0-9	-	-	-	FDR	FIFO Data Count Register	0x16	16	read-only	0x0000	0xFFFF
SCI0-9	-	-	-	LSR	Line Status Register	0x18	16	read-only	0x0000	0xFFFF
SCI0-9	-	-	-	CDR	Compare Match Data Register	0x1A	16	read/ write	0x0000	0xFFFF
SCI0-9	-	-	-	SPTR	Serial Port Register	0x1C	8	read/ write	0x03	0xFF
IRDA	-	-	-	IRCR	IrDA Control Register	0x00	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPCR	SPI Control Register	0x00	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SSLP	SPI Slave Select Polarity Register	0x01	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPPCR	RSPI Pin Control Register	0x02	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPSR	SPI Status Register	0x03	8	read/ write	0x20	0xFF
SPI0,1	-	-	-	SPDR	SPI Data Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
SPI0,1	-	-	-	SPDR_HA	SPI Data Register (halfword access)	0x04	16	read/ write	0x0000	0xFFFF
SPI0,1	-	-	-	SPSCR	SPI Sequence Control Register	0x08	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPSSR	SPI Sequence Status Register	0x09	8	read-only	0x00	0xFF
SPI0,1	-	-	-	SPBR	SPI Bit Rate Register	0x0A	8	read/ write	0xFF	0xFF
SPI0,1	-	-	-	SPDCR	SPI Data Control Register	0x0B	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPCKD	SPI Clock Delay Register	0x0C	8	read/ write	0x00	0xFF

Table 3.3 Register description (32 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SPI0,1	-	-	-	SSLND	SPI Slave Select Negation Delay Register	0x0D	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPND	SPI Next-Access Delay Register	0x0E	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPCR2	SPI Control Register 2	0x0F	8	read/ write	0x00	0xFF
SPI0,1	8	0x2	0-7	SPCMD%s	SPI Command Register %s	0x10	16	read/ write	0x070D	0xFFFF
SPI0,1	-	-	-	SPDCR2	SPI Data Control Register 2	0x20	8	read/ write	0x00	0xFF
CRC	-	-	-	CRCCR0	CRC Control Register0	0x00	8	read/ write	0x00	0xFF
CRC	-	-	-	CRCCR1	CRC Control Register1	0x01	8	read/ write	0x00	0xFF
CRC	-	-	-	CRCDIR	CRC Data Input Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
CRC	-	-	-	CRCDIR_BY	CRC Data Input Register (byte access)	0x04	8	read/ write	0x00	0xFF
CRC	-	-	-	CRCDOR	CRC Data Output Register	0x08	32	read/ write	0x00000000	0xFFFFFFFF
CRC	-	-	-	CRCDOR_HA	CRC Data Output Register (halfword access)	0x08	16	read/ write	0x0000	0xFFFF
CRC	-	-	-	CRCDOR_BY	CRC Data Output Register (byte access)	0x08	8	read/ write	0x00	0xFF
CRC	-	-	-	CRCSAR	Snoop Address Register	0x0C	16	read/ write	0x0000	0xFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTWP	General PWM Timer Write-Protection Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTSTR	General PWM Timer Software Start Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTSTP	General PWM Timer Software Stop Register	0x08	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCLR	General PWM Timer Software Clear Register	0x0C	32	write-only	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTSSR	General PWM Timer Start Source Select Register	0x10	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTPSR	General PWM Timer Stop Source Select Register	0x14	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCSR	General PWM Timer Clear Source Select Register	0x18	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTUPSR	General PWM Timer Up Count Source Select Register	0x1C	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (33 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTDNSR	General PWM Timer Down Count Source Select Register	0x20	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCR	General PWM Timer Control Register	0x2C	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	read/ write	0x00000001	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTIOR	General PWM Timer I/O Control Register	0x34	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTST	General PWM Timer Status Register	0x3C	32	read/ write	0x00008000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTBER	General PWM Timer Buffer Enable Register	0x40	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTITC	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	0x44	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCNT	General PWM Timer Counter	0x48	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCCRA	General PWM Timer Compare Capture Register A	0x4C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCCRB	General PWM Timer Compare Capture Register B	0x50	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCCRC	General PWM Timer Compare Capture Register C	0x54	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCCRE	General PWM Timer Compare Capture Register E	0x58	32	read/ write	0xFFFFFFFF	0xFFFFFFFF

Table 3.3 Register description (34 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCCRD	General PWM Timer Compare Capture Register D	0x5C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTCCRF	General PWM Timer Compare Capture Register F	0x60	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTPR	General PWM Timer Cycle Setting Register	0x64	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTPDBR	General PWM Timer Cycle Setting Double-Buffer Register	0x6C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTADTRA	A/D Converter Start Request Timing Register A	0x70	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTADTBRA	A/D Converter Start Request Timing Buffer Register A	0x74	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTADTDBRA	A/D Converter Start Request Timing Double-Buffer Register A	0x78	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTADTRB	A/D Converter Start Request Timing Register B	0x7C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTADTBRB	A/D Converter Start Request Timing Buffer Register B	0x80	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTADTDBRB	A/D Converter Start Request Timing Double-Buffer Register B	0x84	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	read/ write	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTDVU	General PWM Timer Dead Time Value Register U	0x8C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTDVD	General PWM Timer Dead Time Value Register D	0x90	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTDBU	General PWM Timer Dead Time Buffer Register U	0x94	32	read/ write	0xFFFFFFFF	0xFFFFFFFF

Table 3.3 Register description (35 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTDBD	General PWM Timer Dead Time Buffer Register D	0x98	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTSOS	General PWM Timer Output Protection Function Status Register	0x9C	32	read-only	0x00000000	0xFFFFFFFF
GPT32EH 0-3,GPT32E 4-7	-	-	-	GTSOTR	General PWM Timer Output Protection Function Temporary Release Register	0xA0	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTWP	General PWM Timer Write-Protection Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTSTR	General PWM Timer Software Start Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTSTP	General PWM Timer Software Stop Register	0x08	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTCLR	General PWM Timer Software Clear Register	0x0C	32	write-only	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTSSR	General PWM Timer Start Source Select Register	0x10	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTPSR	General PWM Timer Stop Source Select Register	0x14	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTCSR	General PWM Timer Clear Source Select Register	0x18	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTUPSR	General PWM Timer Up Count Source Select Register	0x1C	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTDNSR	General PWM Timer Down Count Source Select Register	0x20	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTCR	General PWM Timer Control Register	0x2C	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	read/ write	0x00000001	0xFFFFFFFF
GPT328-13	-	-	-	GTIOR	General PWM Timer I/O Control Register	0x34	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTST	General PWM Timer Status Register	0x3C	32	read/ write	0x00008000	0xFFFFFFFF
GPT328-13	-	-	-	GTBER	General PWM Timer Buffer Enable Register	0x40	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTITC	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	0x44	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTCNT	General PWM Timer Counter	0x48	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTCCRA	General PWM Timer Compare Capture Register A	0x4C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF

Table 3.3 Register description (36 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT328-13	-	-	-	GTCCRB	General PWM Timer Compare Capture Register B	0x50	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTCCRC	General PWM Timer Compare Capture Register C	0x54	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTCCRE	General PWM Timer Compare Capture Register E	0x58	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTCCRD	General PWM Timer Compare Capture Register D	0x5C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTCCRF	General PWM Timer Compare Capture Register F	0x60	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTPR	General PWM Timer Cycle Setting Register	0x64	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT328-13	-	-	-	GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	read/ write	0x00000000	0xFFFFFFFF
GPT328-13	-	-	-	GTDVU	General PWM Timer Dead Time Value Register U	0x8C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT_OPS	-	-	-	OPSCR	Output Phase Switching Control Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
GPT_ODC	-	-	-	GTDLYCR	PWM Output Delay Control Register	0x00	16	read/ write	0x0000	0xFFFF
GPT_ODC	-	-	-	GTDLYCR2	PWM Output Delay Control Register2	0x02	16	read/ write	0x0000	0xFFFF
GPT_ODC	4	0x4	0-3	GTDLYR%sA	GTIOC%sA Rising Output Delay Register	0x18	16	read/ write	0x0000	0xFFFF
GPT_ODC	4	0x4	0-3	GTDLYR%sB	GTIOC%sB Rising Output Delay Register	0x1A	16	read/ write	0x0000	0xFFFF
GPT_ODC	4	0x4	0-3	GTDLYF%sA	GTIOC%sA Falling Output Delay Register	0x28	16	read/ write	0x0000	0xFFFF
GPT_ODC	4	0x4	0-3	GTDLYF%sB	GTIOC%sB Falling Output Delay Register	0x2A	16	read/ write	0x0000	0xFFFF
KINT	-	-	-	KRCTL	KEY Return Control Register	0x00	8	read/ write	0x00	0xFF
KINT	-	-	-	KRF	KEY Return Flag Register	0x04	8	read/ write	0x00	0xFF
KINT	-	-	-	KRM	KEY Return Mode Register	0x08	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUCR0	CTSU Control Register 0	0x00	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUCR1	CTSU Control Register 1	0x01	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUSDPRS	CTSU Synchronous Noise Reduction Setting Register	0x02	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUSST	CTSU Sensor Stabilization Wait Control Register	0x03	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUMCH0	CTSU Measurement Channel Register 0	0x04	8	read/ write	0x1F	0xFF
CTSU	-	-	-	CTSUMCH1	CTSU Measurement Channel Register 1	0x05	8	read/ write	0x1F	0xFF
CTSU	-	-	-	CTSUCHAC0	CTSU Channel Enable Control Register 0	0x06	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUCHAC1	CTSU Channel Enable Control Register 1	0x07	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUCHAC2	CTSU Channel Enable Control Register 2	0x08	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUCHTRC0	CTSU Channel Transmit/ Receive Control Register 0	0x0B	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUCHTRC1	CTSU Channel Transmit/ Receive Control Register 1	0x0C	8	read/ write	0x00	0xFF

Table 3.3 Register description (37 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CTSU	-	-	-	CTSUCHTRC2	CTSU Channel Transmit/ Receive Control Register 2	0x0D	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUDCLKC	CTSU High-Pass Noise Reduction Control Register	0x10	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUST	CTSU Status Register	0x11	8	read/ write	0x00	0xFF
CTSU	-	-	-	CTSUSSC	CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register	0x12	16	read/ write	0x0000	0xFFFF
CTSU	-	-	-	CTSUSO0	CTSU Sensor Offset Register 0	0x14	16	read/ write	0x0000	0xFFFF
CTSU	-	-	-	CTSUSO1	CTSU Sensor Offset Register 1	0x16	16	read/ write	0x0000	0xFFFF
CTSU	-	-	-	CTSUSC	CTSU Sensor Counter	0x18	16	read-only	0x0000	0xFFFF
CTSU	-	-	-	CTSURC	CTSU Reference Counter	0x1A	16	read-only	0x0000	0xFFFF
CTSU	-	-	-	CTSUERRS	CTSU Error Status Register	0x1C	16	read-only	0x0000	0x7FFF
AGT0,1	-	-	-	AGT	AGT Counter Register	0x00	16	read/ write	0xFFFF	0xFFFF
AGT0,1	-	-	-	AGTCMA	AGT Compare Match A Register	0x02	16	read/ write	0xFFFF	0xFFFF
AGT0,1	-	-	-	AGTCMB	AGT Compare Match B Register	0x04	16	read/ write	0xFFFF	0xFFFF
AGT0,1	-	-	-	AGTCR	AGT Control Register	0x08	8	read/ write	0x00	0xFF
AGT0,1	-	-	-	AGTMR1	AGT Mode Register 1	0x09	8	read/ write	0x00	0xFF
AGT0,1	-	-	-	AGTMR2	AGT Mode Register 2	0x0A	8	read/ write	0x00	0xFF
AGT0,1	-	-	-	AGTIOC	AGT I/O Control Register	0x0C	8	read/ write	0x00	0xFF
AGT0,1	-	-	-	AGTISR	AGT Event Pin Select Register	0x0D	8	read/ write	0x00	0xFF
AGT0,1	-	-	-	AGTCMSR	AGT Compare Match Function Select Register	0x0E	8	read/ write	0x00	0xFF
AGT0,1	-	-	-	AGTIOSEL	AGT Pin Select Register	0x0F	8	read/ write	0x00	0xFF
ACMPHS0	-	-	-	CMPCTL	Comparator Control Register	0x000	8	read/ write	0x00	0xFF
ACMPHS0	-	-	-	CMPSEL0	Comparator Input Select Register	0x004	8	read/ write	0x00	0xFF
ACMPHS0	-	-	-	CMPSEL1	Comparator Reference Voltage Select Register	0x008	8	read/ write	0x00	0xFF
ACMPHS0	-	-	-	CMPMON	Comparator Output Monitor Register	0x00C	8	read-only	0x00	0xFF
ACMPHS0	-	-	-	CPIOC	Comparator Output Control Register	0x010	8	read/ write	0x00	0xFF
ACMPHS1 -5	-	-	-	CMPCTL	Comparator Control Register	0x000	8	read/ write	0x00	0xFF
ACMPHS1 -5	-	-	-	CMPSEL0	Comparator Input Select Register	0x004	8	read/ write	0x00	0xFF
ACMPHS1 -5	-	-	-	CMPSEL1	Comparator Reference Voltage Select Register	0x008	8	read/ write	0x00	0xFF
ACMPHS1 -5	-	-	-	CMPMON	Comparator Output Monitor Register	0x00C	8	read-only	0x00	0xFF
ACMPHS1 -5	-	-	-	CPIOC	Comparator Output Control Register	0x010	8	read/ write	0x00	0xFF

Table 3.3 Register description (38 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBFS	-	-	-	SYSCFG	System Configuration Control Register	0x000	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	SYSSTS0	System Configuration Status Register 0	0x004	16	read- only	0x0000	0x0000
USBFS	-	-	-	DVSTCTR0	Device State Control Register 0	0x008	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	CFIFO	CFIFO Port Register	0x014	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	CFIFOL	CFIFO Port Register L	0x014	8	read/ write	0x00	0xFF
USBFS	-	-	-	D0FIFO	D0FIFO Port Register	0x018	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	D0FIFOL	D0FIFO Port Register L	0x018	8	read/ write	0x00	0xFF
USBFS	-	-	-	D1FIFO	D1FIFO Port Register	0x01C	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	D1FIFOL	D1FIFO Port Register L	0x01C	8	read/ write	0x00	0xFF
USBFS	-	-	-	CFIFOSEL	CFIFO Port Select Register	0x020	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	CFIFOCTR	CFIFO Port Control Register	0x022	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	D0FIFOSEL	D0FIFO Port Select Register	0x028	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	D0FIFOCTR	D0FIFO Port Control Register	0x02A	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	D1FIFOSEL	D1FIFO Port Select Register	0x02C	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	D1FIFOCTR	D1FIFO Port Control Register	0x02E	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	INTENB0	Interrupt Enable Register 0	0x030	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	INTENB1	Interrupt Enable Register 1	0x032	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	BRDYENB	BRDY Interrupt Enable Register	0x036	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	NRDYENB	NRDY Interrupt Enable Register	0x038	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	BEMPENB	BEMP Interrupt Enable Register	0x03A	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	SOFCFG	SOF Output Configuration Register	0x03C	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	INTSTS0	Interrupt Status Register 0	0x040	16	read/ write	0x0000	0xFF7F
USBFS	-	-	-	INTSTS1	Interrupt Status Register 1	0x042	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	BRDYSTS	BRDY Interrupt Status Register	0x046	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	NRDYSTS	NRDY Interrupt Status Register	0x048	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	BEMPSTS	BEMP Interrupt Status Register	0x04A	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	FRMNUM	Frame Number Register	0x04C	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	DVCHGR	Device State Change Register	0x04E	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	USBADDR	USB Address Register	0x050	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	USBREQ	USB Request Type Register	0x054	16	read/ write	0x0000	0xFFFF

Table 3.3　Register description (39 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBFS	-	-	-	USBVAL	USB Request Value Register	0x056	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	USBINDX	USB Request Index Register	0x058	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	USBLENG	USB Request Length Register	0x05A	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	DCPCFG	DCP Configuration Register	0x05C	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	DCPMAXP	DCP Maximum Packet Size Register	0x05E	16	read/ write	0x0040	0xFFFF
USBFS	-	-	-	DCPCTR	DCP Control Register	0x060	16	read/ write	0x0040	0xFFFF
USBFS	-	-	-	PIPESEL	Pipe Window Select Register	0x064	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	PIPECFG	Pipe Configuration Register	0x068	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	PIPEMAXP	Pipe Maximum Packet Size Register	0x06C	16	read/ write	0x0000	0xFFBF
USBFS	-	-	-	PIPEPERI	Pipe Cycle Control Register	0x06E	16	read/ write	0x0000	0xFFFF
USBFS	5	0x002	1-5	PIPE%sCTR	Pipe %s Control Register	0x070	16	read/ write	0x0000	0xFFFF
USBFS	4	0x002	6-9	PIPE%sCTR	Pipe %s Control Register	0x07A	16	read/ write	0x0000	0xFFFF
USBFS	5	0x004	1-5	PIPE%sTRE	Pipe %s Transaction Counter Enable Register	0x090	16	read/ write	0x0000	0xFFFF
USBFS	5	0x004	1-5	PIPE%sTRN	Pipe %s Transaction Counter Register	0x092	16	read/ write	0x0000	0xFFFF
USBFS	6	0x002	0-5	DEVADD%s	Device Address %s Configuration Register	0x0D0	16	read/ write	0x0000	0xFFFF
USBFS	-	-	-	PHYSLEW	PHY Cross Point Adjustment Register	0x0F0	32	read/ write	0x0000000E	0xFF4CFFFF
USBFS	-	-	-	DPUSR0R	Deep Software Standby USB Transceiver Control/Pin Monitor Register	0x400	32	read/ write	0x00000000	0xFF4CFFFF
USBFS	-	-	-	DPUSR1R	Deep Software Standby USB Suspend/Resume Interrupt Register	0x404	32	read/ write	0x00000000	0xFFFFFFFF
PDC	-	-	-	PCCR0	PDC Control Register 0	0x000	32	read/ write	0x00000000	0xFFFFFFFF
PDC	-	-	-	PCCR1	PDC Control Register 1	0x004	32	read/ write	0x00000000	0xFFFFFFFF
PDC	-	-	-	PCSR	PDC Status Register	0x008	32	read/ write	0x00000002	0xFFFFFFFF
PDC	-	-	-	PCMONR	PDC Pin Monitor Register	0x00C	32	read-only	0x00000000	0xFFFFFFFF
PDC	-	-	-	PCDR	PDC Receive Data Register	0x010	32	read-only	0x00000000	0xFFFFFFFF
PDC	-	-	-	VCR	Vertical Capture Register	0x014	32	read/ write	0x00000000	0xFFFFFFFF
PDC	-	-	-	HCR	Horizontal Capture Register	0x018	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	256	0x4	0-255	GR1_CLUT0[%s]	Color Palette 0 Plane for Graphics 1 Plane	0x0000	32	read/ write	0x00000000	0x00000000
GLCDC	256	0x4	0-255	GR1_CLUT1[%s]	Color Palette 1 Plane for Graphics 1 Plane	0x0400	32	read/ write	0x00000000	0x00000000
GLCDC	256	0x4	0-255	GR2_CLUT0[%s]	Color Palette 0 Plane for Graphics 2 Plane	0x0800	32	read/ write	0x00000000	0x00000000
GLCDC	256	0x4	0-255	GR2_CLUT1[%s]	Color Palette 1 Plane for Graphics 2 Plane	0x0C00	32	read/ write	0x00000000	0x00000000

Table 3.3 Register description (40 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GLCDC	-	-	-	BG_EN	Background Plane Setting Operation Control Register	0x1000	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	BG_PERI	Background Plane Setting Free-Running Period Register	0x1004	32	read/ write	0x00170017	0xFFFFFFFF
GLCDC	-	-	-	BG_SYNC	Background Plane Setting Synchronization Position Register	0x1008	32	read/ write	0x00010001	0xFFFFFFFF
GLCDC	-	-	-	BG_VSIZE	Background Plane Setting Full Image Vertical Size Register	0x100C	32	read/ write	0x00070010	0xFFFFFFFF
GLCDC	-	-	-	BG_HSIZE	Background Plane Setting Full Image Horizontal Size Register	0x1010	32	read/ write	0x00060010	0xFFFFFFFF
GLCDC	-	-	-	BG_BGC	Background Plane Setting Background Color Register	0x1014	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	BG_MON	Background Plane Setting Status Monitor Register	0x1018	32	read- only	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_VEN	Graphics %s Register Update Control Register	0x1100	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_FLMRD	Graphics %s Frame Buffer Read Control Register	0x1104	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_FLM1	Graphics %s Frame Buffer Control Register 1	0x1108	32	read/ write	0x00000003	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_FLM2	Graphics %s Frame Buffer Control Register 2	0x110C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_FLM3	Graphics %s Frame Buffer Control Register 3	0x1110	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_FLM5	Graphics %s Frame Buffer Control Register 5	0x1118	32	read/ write	0x000F0000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_FLM6	Graphics %s Frame Buffer Control Register 6	0x111C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB1	Graphics %s Alpha Blending Control Register 1	0x1120	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB2	Graphics %s Alpha Blending Control Register 2	0x1124	32	read/ write	0x00060010	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB3	Graphics %s Alpha Blending Control Register 3	0x1128	32	read/ write	0x00050010	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB4	Graphics %s Alpha Blending Control Register 4	0x112C	32	read/ write	0x00060010	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB5	Graphics %s Alpha Blending Control Register 5	0x1130	32	read/ write	0x00050010	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB6	Graphics %s Alpha Blending Control Register 6	0x1134	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB7	Graphics %s Alpha Blending Control Register 7	0x1138	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB8	Graphics %s Alpha Blending Control Register 8	0x113C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_AB9	Graphics %s Alpha Blending Control Register 9	0x1140	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_BASE	Graphics %s Background Color Control Register	0x114C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_CLUTINT	Graphics %s CLUT Table Interrupt Control Register	0x1150	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x100	1,2	GR%s_MON	Graphics %s Status Monitor Register	0x1154	32	read- only	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LATCH	Gamma %s Register Update Control Register	0x1300	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	GAM_SW	Gamma Correction Block Function Switch Register	0x1304	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (41 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GLCDC	3	0x40	G,B,R	GAM%s_LUT1	Gamma %s Correction Block Table Setting Register 1	0x1308	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LUT2	Gamma %s Correction Block Table Setting Register 2	0x130C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LUT3	Gamma %s Correction Block Table Setting Register 3	0x1310	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LUT4	Gamma %s Correction Block Table Setting Register 4	0x1314	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LUT5	Gamma %s Correction Block Table Setting Register 5	0x1318	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LUT6	Gamma %s Correction Block Table Setting Register 6	0x131C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LUT7	Gamma %s Correction Block Table Setting Register 7	0x1320	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_LUT8	Gamma %s Correction Block Table Setting Register 8	0x1324	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_AREA1	Gamma %s Correction Block Area Setting Register 1	0x1328	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_AREA2	Gamma %s Correction Block Area Setting Register 2	0x132C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_AREA3	Gamma %s Correction Block Area Setting Register 3	0x1330	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_AREA4	Gamma %s Correction Block Area Setting Register 4	0x1334	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	3	0x40	G,B,R	GAM%s_AREA5	Gamma %s Correction Block Area Setting Register 5	0x1338	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	OUT_VLATCH	Output Control Block Register Update Control Register	0x13C0	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	OUT_SET	Output Control Block Output Interface Register	0x13C4	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	OUT_BRIGHT1	Output Control Block Brightness Correction Register 1	0x13C8	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	OUT_BRIGHT2	Output Control Block Brightness Correction Register 2	0x13CC	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	OUT_CONTRAST	Output Control Block Contrast Correction Register	0x13D0	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	OUT_PDTHA	Output Control Block Panel Dither Correction Register	0x13D4	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	OUT_CLKPHASE	Output Control Block Output Phase Control Register	0x13E4	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	TCON_TIM	TCON Reference Timing Setting Register	0x1404	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x8	A,B	TCON_STV%s1	TCON Vertical Timing Setting Register %s1	0x1408	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x8	A,B	TCON_STV%s2	TCON Vertical Timing Setting Register %s2	0x140C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x8	A,B	TCON_STH%s1	TCON Horizontal Timing Setting Register STH%s1	0x1418	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	2	0x8	A,B	TCON_STH%s2	TCON Horizontal Timing Setting Register STH%s2	0x141C	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	TCON_DE	TCON Data Enable Polarity Setting Register	0x1428	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	SYSCNT_DTCTEN	System Control Block State Detection Control Register	0x1440	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	SYSCNT_INTEN	System Control Block Interrupt Request Enable Control Register	0x1444	32	read/ write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (42 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GLCDC	-	-	-	SYSCNT_STCLR	System Control Block Status Clear Register	0x1448	32	read/ write	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	SYSCNT_STMON	System Control Block Status Monitor Register	0x144C	32	read-only	0x00000000	0xFFFFFFFF
GLCDC	-	-	-	SYSCNT_PANEL_CLK	System Control Block Version and Panel Clock Control Register	0x1450	32	read/ write	0x01100000	0xFFFFFFFF
DRW	-	-	-	CONTROL	Geometry Control Register	0x00	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	STATUS	Status Control Register	0x00	32	read-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	CONTROL2	Surface Control Register	0x04	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	HWREVISION	Hardware Version and Feature Set ID Register	0x04	32	read-only	0x0FBE0107	0xFFFFF000
DRW	6	0x4	1-6	L%sSTART	Limiter %s Start Value Register	0x10	32	write-only	0x00000000	0xFFFFFFFF
DRW	6	0x4	1-6	L%sXADD	Limiter %s X-Axis Increment Register	0x28	32	write-only	0x00000000	0xFFFFFFFF
DRW	6	0x4	1-6	L%sYADD	Limiter %s Y-Axis Increment Register	0x40	32	write-only	0x00000000	0xFFFFFFFF
DRW	2	0x4	1,2	L%sBAND	Limiter %s Band Width Parameter Register	0x58	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	COLOR1	Base Color Register	0x64	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	COLOR2	Secondary Color Register	0x68	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	PATTERN	Pattern Register	0x74	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	SIZE	Bounding Box Dimension Register	0x78	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	PITCH	Framebuffer Pitch And Spanstore Delay Register	0x7C	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	ORIGIN	Framebuffer Base Address Register	0x80	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LUSTART	U Limiter Start Value Register	0x90	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LUXADD	U Limiter X-Axis Increment Register	0x94	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LUYADD	U Limiter Y-Axis Increment Register	0x98	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LVSTARTI	V Limiter Start Value Integer Part Register	0x9C	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LVSTARTF	V Limiter Start Value Fractional Part Register	0xA0	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LVXADDI	V Limiter X-Axis Increment Integer Part Register	0xA4	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LVYADDI	V Limiter Y-Axis Increment Integer Part Register	0xA8	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	LVYXADDF	V Limiter Increment Fractional Parts Register	0xAC	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	TEXPITCH	Texels Per Texture Line Register	0xB4	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	TEXMASK	Texture Size or Texture Address Mask Register	0xB8	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	TEXORIGIN	Texture Base Address Register	0xBC	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	IRQCTL	Interrupt Control Register	0xC0	32	write-only	0x00000000	0xFFFFFFFF

Table 3.3 Register description (43 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
DRW	-	-	-	CACHECTL	Cache Control Register	0xC4	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	DLISTSTART	Display List Start Address Register	0xC8	32	write-only	0x00000000	0xFFFFFFFF
DRW	2	0x4	1,2	PERFCOUNT%s	Performance Counter %s	0xCC	32	read/write	0x00000000	0xFFFFFFFF
DRW	-	-	-	PERFTRIGGER	Performance Counters Control Register	0xD4	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	TEXCLADDR	CLUT Start Address Register	0xDC	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	TEXCLDATA	CLUT Data Register	0xE0	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	TEXCLOFFSET	CLUT Offset Register	0xE4	32	write-only	0x00000000	0xFFFFFFFF
DRW	-	-	-	COLKEY	Color Key Register	0xE8	32	write-only	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JCMOD	JPEG Code Mode Register	0x000	8	read/write	0x00	0xFF
JPEG	-	-	-	JCCMD	JPEG Code Command Register	0x001	8	write-only	0x00	0x00
JPEG	-	-	-	JCQTN	JPEG Code Quantization Table Number Register	0x003	8	read/write	0x00	0xFF
JPEG	-	-	-	JCHTN	JPEG Code Huffman Table Number Register	0x004	8	read/write	0x00	0xFF
JPEG	-	-	-	JCDRIU	JPEG Code DRI Upper Register	0x005	8	read/write	0x00	0xFF
JPEG	-	-	-	JCDRID	JPEG Code DRI Lower Register	0x006	8	read/write	0x00	0xFF
JPEG	-	-	-	JCVSZU	JPEG Code Vertical Size Upper Register	0x007	8	read/write	0x00	0xFF
JPEG	-	-	-	JCVSZD	JPEG Code Vertical Size Lower Register	0x008	8	read/write	0x00	0xFF
JPEG	-	-	-	JCHSZU	JPEG Code Horizontal Size Upper Register	0x009	8	read/write	0x00	0xFF
JPEG	-	-	-	JCHSZD	JPEG Coded Horizontal Size Lower Register	0x00A	8	read/write	0x00	0xFF
JPEG	-	-	-	JCDTCU	JPEG Code Data Count Upper Register	0x00B	8	read-only	0x00	0xFF
JPEG	-	-	-	JCDTCM	JPEG Code Data Count Middle Register	0x00C	8	read-only	0x00	0xFF
JPEG	-	-	-	JCDTCD	JPEG Code Data Count Lower Register	0x00D	8	read-only	0x00	0xFF
JPEG	-	-	-	JINTE0	JPEG Interrupt Enable Register 0	0x00E	8	read/write	0x00	0xFF
JPEG	-	-	-	JINTS0	JPEG Interrupt Status Register 0	0x00F	8	read/write	0x00	0xFF
JPEG	-	-	-	JCDERR	JPEG Code Decode Error Register	0x010	8	read/write	0x0A	0xFF
JPEG	-	-	-	JCRST	JPEG Code Reset Register	0x011	8	read-only	0x00	0xFF
JPEG	-	-	-	JIFECNT	JPEG Interface Compression Control Register	0x040	32	read/write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JIFESA	JPEG Interface Compression Source Address Register	0x044	32	read/write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JIFESOFST	JPEG Interface Compression Line Offset Register	0x048	32	read/write	0x00000000	0xFFFFFFFF

Table 3.3 Register description (44 of 44)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
JPEG	-	-	-	JIFEDA	JPEG Interface Compression Destination Address Register	0x04C	32	read/ write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JIFESLC	JPEG Interface Compression Source Line Count Register	0x050	32	read/ write	0xFFF8FFF8	0xFFFFFFFF
JPEG	-	-	-	JIFDCNT	JPEG Interface Decompression Control Register	0x058	32	read/ write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JIFDSA	JPEG Interface Decompression Source Address Register	0x05C	32	read/ write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JIFDDOFST	JPEG Interface Decompression Line Offset Register	0x060	32	read/ write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JIFDDA	JPEG Interface Decompression Destination Address Register	0x064	32	read/ write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JIFDSDC	JPEG Interface Decompression Source Data Count Register	0x068	32	read/ write	0xFFF8FFF8	0xFFFFFFFF
JPEG	-	-	-	JIFDDLC	JPEG Interface Decompression Destination Line Count Register	0x06C	32	read/ write	0xFFF8FFF8	0xFFFFFFFF
JPEG	-	-	-	JIFDADT	JPEG Interface Decompression alpha Set Register	0x070	32	read/ write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JINTE1	JPEG Interrupt Enable Register 1	0x08C	32	read/ write	0x00000000	0xFFFFFFFF
JPEG	-	-	-	JINTS1	JPEG Interrupt Status Register 1	0x090	32	read/ write	0x00000000	0xFFFFFFFF
QSPI	-	-	-	SFMSMD	Transfer Mode Control Register	0x000	32	read/ write	0x00000000	0xFFFFFFFF
QSPI	-	-	-	SFMSSC	Chip Selection Control Register	0x004	32	read/ write	0x00000037	0xFFFFFFFF
QSPI	-	-	-	SFMSKC	Clock Control Register	0x008	32	read/ write	0x00000008	0xFFFFFFFF
QSPI	-	-	-	SFMSST	Status Register	0x00C	32	read-only	0x00000080	0xFFFFFFFF
QSPI	-	-	-	SFMCOM	Communication Port Register	0x010	32	read/ write	0x00000000	0xFFFFFF00
QSPI	-	-	-	SFMCMD	Communication Mode Control Register	0x014	32	read/ write	0x00000000	0xFFFFFFFF
QSPI	-	-	-	SFMCST	Communication Status Register	0x018	32	read/ write	0x00000000	0xFFFFFFFF
QSPI	-	-	-	SFMSIC	Instruction Code Register	0x020	32	read/ write	0x00000000	0xFFFFFFFF
QSPI	-	-	-	SFMSAC	Address Mode Control Register	0x024	32	read/ write	0x00000002	0xFFFFFFFF
QSPI	-	-	-	SFMSDC	Dummy Cycle Control Register	0x028	32	read/ write	0x0000FF00	0xFFFFFFFF
QSPI	-	-	-	SFMSPC	SPI Protocol Control Register	0x030	32	read/ write	0x00000010	0xFFFFFFFF
QSPI	-	-	-	SFMPMD	Port Control Register	0x034	32	read/ write	0x00000000	0xFFFFFFFF
QSPI	-	-	-	SFMCNT1	External QSPI Address Register 1	0x804	32	read/ write	0x00000000	0xFFFFFFFF

Peripheral name = Name of peripheral
Dim = Number of elements in an array of registers
Dim inc = Address increment between two simultaneous registers of a register array in the address map

Dim index = Sub string that replaces the %s placeholder within the register name
Register name = Name of register
Description = Register description
Address offset = Address of the register relative to the base address defined by the peripheral of the register
Size = Bit width of the register
Access = Register access rights:
Read-only: Read access is permitted. Write operations have undefined results.
Write-only: Write access is permitted. Read operations have undefined results.
Read/write: Both read and write accesses are permitted. Writes affect the state of the register and reads return a value related to the register.
Reset value = Default reset value of a register
Reset mask = Identifies which register bits have a defined reset value

Revision History	S5D9 Microcontroller Group User’s Manual

Rev.	Date	Chapter	Summary
1.00	Nov 3, 2016	-	First release
1.10	Mar 23, 2018	-	Second release
		Features	Updated description for 64-KB data flash memory from 100,000 erase/write cycles to 125,000 erase/write cycles
		section 1, Overview	Updated Table 1.1 and Table 1.2
			Updated Figure 1.2, Part numbering scheme
			Updated Table 1.16, Pin functions
		section 2, CPU	Updated section 2.1.1 and section 2.5.1
			Updated description for the EDBGRQ bit and added a note in section 2.6.5.3, MCU Control Register (MCUCTRL)
			Added a note in section 2.9, SysTick System Timer
			Added section (4), When OSIS[127:126] = 2’b11 in 2.11.3.4
		section 4, Address Space	Updated Figure 4.1 and Figure 4.2
		section 5, Memory Mirror Function (MMF)	Added a new paragraph after Figure 5.2
			Modified description in section 5.3.2, Setting Example
		section 7, Option-Setting Memory	Updated Table 7.1, Specifications for ID code protection
		section 8, Low Voltage Detection (LVD)	Updated Figure 8.4
		section 9, Clock Generation Circuit	Updated Table 9.2
			Updated Figure 9.1
			Updated description of section 9.2.13, FLL Control Register 1 (FLLCR1)
			Updated section 9.5.2
		section 11, Low Power Modes	Updated Table 11.2
			Updated names of bit [7] and bit [8] in section 11.2.4
			Updated symbol name of bit [2] in section 11.2.15 and section 11.2.19
			Updated Table 11.10
		section 12, Battery Backup Function	Updated Figure 12.1
		section 14, Interrupt Controller Unit (ICU)	Updated description for the IRQCRi register in section 14.2.1
		section 15, Buses	Updated the values after reset in section 15.3.21 and section 15.3.22
		section 16, Memory Protection Unit (MPU)	Updated Figure 16.4
			Updated section 16.4.1.2, and section 16.6.1.1 through section 16.6.1.10
		section 18, Data Transfer Controller (DTC)	Updated Figure 18.2
			Updated section 18.6.3
		section 19, Event Link Controller (ELC)	Added a new section 19.4.4, ELC Delay Time
		section 20, I/O Ports	Updated section 20.2.1 through section 20.2.5
			Updated Table 20.3, Table 20.10, Table 20.14, and Table 20.15

Rev.	Date	Chapter	Summary
1.10	Mar 23, 2018	section 21, Key Interrupt Function (KINT)	Updated Figure 21.1 and added a new paragraph after the figure
			Updated the note in section 21.2.2
		section 22, Port Output Enable for GPT (POEG)	Updated description in section 22.3.2
		section 23, General PWM Timer (GPT)	Updated Table 23.2
			Updated Figure 23.1
			Updated section 23.2.15
			Updated description of the DTEF flag in section 23.2.16
			Updated section 23.2.18 and section 23.2.32
			Updated Figure 23.81 through Figure 23.85
			Updated Table 23.20 and Table 23.21
		section 25, Asynchronous General-Purpose Timer (AGT)	Updated Figure 25.1
		section 26, Realtime Clock (RTC)	Updated section 26.2.22, section 26.2.26, and section 26.2.27
			Updated Figure 26.7
		section 27, Watchdog Timer (WDT)	Updated Figure 27.3
			Updated section 27.5.1
		section 28, Independent Watchdog Timer (IWDT)	Updated Figure 28.1
		section 29, Ethernet MAC Controller (ETHERC)	Updated the value after reset for bits [15:0] in section 29.2.10
			Updated section 29.2.15 through section 29.2.26
		section 30, Ethernet PTP Controller (EPTPC)	Updated description for the SYTH[3:0] bits in section 30.2.8
			Updated register symbol name in section 30.2.66
		section 32, USB 2.0 Full-Speed Module (USBFS)	Updated section 32.2.4, section 32.2.5, section 32.2.8, section 32.2.13, section 32.2.14, and section 32.2.31
			Updated Table 32.16
			Updated Figure 32.12
			Updated 32.3.4.7
		section 33, USB 2.0 High-Speed Module (USBHS)	Updated description for MBW[1:0] bits in section 33.2.8 and section 33.2.9
			Updated the value after reset for bit [3] and bit [2] in section 33.2.16
		section 34, Serial Communications Interface (SCI)	Updated section 34.2.10
			Updated description of bit [3] and bit [6] in section 34.2.11
			Added a note to the description of the RDRF and TDRE flags in section 34.2.13 through section 34.2.15
			Updated Table 34.13, Table 34.15, Table 34.17, and Table 34.18
			Updated section 34.3.5
			Updated Figure 34.5, Figure 34.6, and Figure 34.30
			Updated section 34.5.2
			Updated Figure 34.38, Figure 34.45, and Figure 34.50. Added Figure 34.51
			Updated Figure 34.52, Figure 34.53, and Figure 34.57
			Updated Table 34.25, added Note 1 to Table 34.26 and Table 34.27
			Updated Figure 34.74 and Figure 34.75

Rev.	Date	Chapter	Summary
1.10	Mar 23, 2018	section 36, I^{2}C Bus Interface (IIC)	Updated section 36.2.1
			Updated Table 36.3
			Updated Figure 36.26, Figure 36.31, Figure 36.36, and Figure 36.45
			Updated section 36.12, Bus Hanging
			Updated Table 36.10
		section 37, Controller Area Network (CAN) Module	Updated the note in section 37.2.6
		section 38, Serial Peripheral Interface (SPI)	Updated Figure 38.29, Figure 38.30, and Figure 38.37 through Figure 38.39
			Updated Table 38.13 and Table 38.15
		section 39, Quad Serial Peripheral Interface (QSPI)	Updated Figure 39.2 and Figure 39.12
		section 41, Serial Sound Interface Enhanced (SSIE)	Updated Figure 41.26
		section 43, SD/MMC Host Interface (SDHI)	Updated Table 43.1
			Updated Figure 43.6
		section 45, Boundary Scan	Updated the value after reset for bits [20], [18], and [17] in section 45.2.2
			Updated the constraints for the boundary scan function in section 45.4, Usage Notes
		section 47, 12-Bit A/D Converter (ADC12)	Updated Note 1 and Note 2 in section 47.2.1
			Updated description of ADST bit in section 47.2.3
			Updated Figure 47.3
			Updated section 47.2.17
			Updated description for the CMPSTB flag in section 47.2.31
			Updated step 2 and step 5 in section 47.3.2.9
			Updated Figure 47.15, Figure 47.20
			Updated section 47.3.4.3
			Updated Figure 47.26 through Figure 47.30, and Figure 47.32
			Updated Table 47.10 and Table 47.13
			Updated section 47.3.9
			Updated formula for calculating the error in absolute accuracy of the ADC12 in section 47.6.7
			Updated section 47.6.8 and section 47.6.12
		section 48, 12-Bit D/A Converter (DAC12)	Updated Figure 48.4
		section 49, Temperature Sensor (TSN)	Updated Table 49.1
			Added section 49.2.2, Temperature Sensor Calibration Data Register(TSCDR)
			Updated section 49.3.1, Preparation for Using the Temperature Sensor
			Updated Figure 49.2
		section 50, High-Speed Analog Comparator (ACMPHS)	Updated Note 1. Note 3., Note 4. and Note 6. in Table 50.2
			Updated description of bit [6] and bit [5] in section 50.2.1
			Updated the name of the register in Step 9 of Table 50.3
		section 51, Capacitive Touch Sensing Unit (CTSU)	Updated Figure 51.14 and Figure 51.16
		section 53, SRAM	Added Note 2. to Table 53.3
			Added section 53.4.3, Store Buffer of SRAM
1.10	Mar 23, 2018	section 55, Flash Memory	Updated Table 55.5
		section 56, 2D Drawing Engine (DRW)	Updated section 56.2.2, section 56.2.6, and section 56.2.10 through section 56.2.13
		section 58, Graphics LCD Controller (GLCDC)	Updated section 58.2.56
		section 60, Electrical Characteristics	Updated Table 60.1, Table 60.6, Table 60.7, Table 60.24
			Updated Figure 60.51 through Figure 60.57, Figure 60.63, Figure 60.64, and Figure 60.66 through Figure 60.76
			Updated Table 60.31, Table 60.40, Table 60.41, Table 60.53, and Table 60.54
		section 3, I/O Registers	Updated Table 3.1 and Table 3.2

Rev.	Date	Chapter	Summary
1.20	Aug 10, 2018	-	Third release
		Features	Updated Features
		section 1, Overview	Updated Table 1.3 and Table 1.13, Security
		section 2, CPU	Updated section 2.1.2, Debug
			Updated Table 2.1
		section 4, Address Space	Updated Figure 4.1 and Figure 4.2
		section 7, Option-Setting Memory	Added a Note: in Changing the option-setting memory by self-programming
		section 15, Buses	Updated Table 15.1 and Table 15.2
		section 16, Memory Protection Unit (MPU)	Updated Table 16.7
		section 20, I/O Ports	Updated description for the PDR and PODR bits in section 20.2.1, Port Control Register 1 (PCNTR1/PODR/PDR)
			Updated description for the PIDR bit in section 20.2.2, Port Control Register 2 (PCNTR2/EIDR/PIDR)
			Updated Figure 20.2 and Figure 20.4
		section 22, Port Output Enable for GPT (POEG)	Updated Figure 22.4
		section 27, Watchdog Timer (WDT)	Updated Figure 27.4
		section 30, Ethernet PTP Controller (EPTPC)	Updated section 30.3.1, Transmission and Reception of Non-PTP Messages
		section 33, USB 2.0 High-Speed Module (USBHS)	Updated Table 33.17
		section 43, SD/MMC Host Interface (SDHI)	Updated value after reset in section 43.2.17
		section 46, Secure Cryptographic Engine (SCE7)	Updated section 46, Secure Cryptographic Engine (SCE7)
		section 47, 12-Bit A/D Converter (ADC12)	Updated Table 47.10, Table 47.14
		section 49, Temperature Sensor (TSN)	Updated Figure 49.1
		section 60, Electrical Characteristics	Updated Table 60.14
			Updated Figure 60.12
1.30	Aug 30, 2019	-	Fourth release
		section 2, CPU	Updated Table 2.3, JTAG/SWD pins
			Updated section 2.6.5.2, MCU Status Register (MCUSTAT)
		section 6, Resets	Updated Table 6.3, Module-related registers initialized by each reset source
		section 8, Low Voltage Detection (LVD)	Updated description for the DET flag in section 8.2.2, Voltage Monitor 1 Circuit Status Register (LVD1SR) and section 8.2.4, Voltage Monitor 2 Circuit Status Register (LVD2SR)
		section 9, Clock Generation Circuit	Updated section 9.2.3, System Clock Source Control Register (SCKSCR)
			Updated section 9.7.1, System Clock (ICLK), added Figure 9.13 and Figure 9.14
			Updated section 9.7.3, Flash Interface Clock (FCLK)
		section 14, Interrupt Controller Unit (ICU)	Updated Table 14.4, Event table
		section 15, Buses	Updated Table 15.14, Conditions for register modification
		section 17, DMA Controller (DMAC)	Updated Figure 17.1, DMAC block diagram
		section 18, Data Transfer Controller (DTC)	Updated section 18.6.3
		section 20, I/O Ports	Updated section 20.2.1, section 20.2.3, and section 20.2.4
			Updated Table 20.12
		section 22, Port Output Enable for GPT (POEG)	Updated Table 22.1 and changed “GTETRGA to GTETRGD” to “GTETRGn” throughout the chapter

Rev.	Date	Chapter	Summary
1.30	Aug 30, 2019	section 23, General PWM Timer (GPT)	Updated section 23.2.8, section 23.2.9
			Updated description for the OmDTYR bit in section 23.2.13
			Updated section 23.2.18
			Changed GTADTRm to GTADTRn in section 23.2.24
			Changed GTADTBRm to GTADTBRn in section 23.2.25
			Changed GTADTDBRm to GTADTDBRn in section 23.2.26
			Updated section 23.2.28
			Changed GTDBm to GTDBn, and changed GTDVm to GTDVn in section 23.2.29
			Updated b31 to b1 in section 23.2.31
			Updated section 23.3.4, Automatic Dead Time Setting Function
			Updated Figure 23.93
			Updated section 23.10.4
		section 25, Asynchronous General-Purpose Timer (AGT)	Updated the note for the AGTIO Pin Select bit name in section 25.2.10, AGT Pin Select Register (AGTIOSEL)
			Added section 25.4.11, When Switching Source Clock
		section 26, Realtime Clock (RTC)	Added section 26.6.8, When Switching Source Clock
		section 29, Ethernet MAC Controller (ETHERC)	Updated the R/W permission for bits in section 29.2.10, Manual PAUSE Frame Register (MPR)
			Updated Figure 29.13 and Figure 29.14
			Added section 29.5.3, Processing when Erroneous Frame Is Detected and section 29.5.4, Collision Occurrence in Half-Duplex Mode
		section 30, Ethernet PTP Controller (EPTPC)	Updated description for the INFABT flag in section 30.2.30, SYNFP Status Register (SYSR)
		section 33, USB 2.0 High-Speed Module (USBHS)	Updated description for bits[15:12] in section 33.2.31, DCP Maximum Packet Size Register (DCPMAXP)
			Updated Figure 33.2
		section 34, Serial Communications Interface (SCI)	Updated Table 34.13
		section 39, Quad Serial Peripheral Interface (QSPI)	Updated Figure 39.3
		section 47, 12-Bit A/D Converter (ADC12)	Updated Step 6 of section 47.3.2.8, A/D conversion in double-trigger mode
			Added Table 47.12, PGA output voltage
		section 48, 12-Bit D/A Converter (DAC12)	Updated the address of section 48.2.6, D/A Amplifier Stabilization Wait Control Register (DAASWCR)
		section 51, Capacitive Touch Sensing Unit (CTSU)	Updated description for b0 and b4 in section 51.2.1, CTSU Control Register 0 (CTSUCR0)
		section 57, JPEG Codec (JPEG)	Added section 57.6, Usage Notes
		section 58, Graphics LCD Controller (GLCDC)	Updated Table 58.2
			Updated Figure 58.22
		section 60, Electrical Characteristics	Updated description for voltage detection level in Table 60.47
			Added a Note to Table 60.17, NMI and IRQ noise filter
			Updated the Note in Table 60.19
		section 1, Port States in Each Processing Mode	Updated Note 5. in Table 1.1, Port states in each processing state

S5D9 Microcontroller Group User’s Manual

Publication Date: Rev.1.30 Aug 30, 2019

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

© 2019 Renesas Electronics Corporation. All rights reserved.

Colophon 6.0