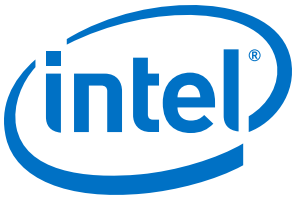


SDI II Intel® FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.1**

Subscribe
Send Feedback

UG-01125 | 2019.08.08
Latest document on the web: **PDF** | **HTML**

Contents

1. SDI II Intel® FPGA IP Core Quick Reference

The Intel® FPGA Serial Digital Interface (SDI) II intellectual property (IP) core is the next generation SDI IP core.

The SDI II Intel FPGA IP core is part of the Intel FPGA IP Library, which is distributed with the Intel Quartus® Prime software and downloadable from www.intel.com.

Note: For system requirements and installation instructions, refer to the *Intel FPGA Software Installation & Licensing* manual.

Table 1. Brief Information About the SDI II Intel FPGA IP Core

Information		Description
Release Information	Version	19.1
	Release Date	April 2019
	Ordering Code	IP-SDI-II
IP Core Information	SDI Data Rate Support	• 270-Mbps SD-SDI, as defined by *SMPTE ST 259* specification • 1.485-Gbps or 1.4835-Gbps HD-SDI, as defined by *SMPTE ST 292* specification • 2.97-Gbps or 2.967-Gbps 3G-SDI, as defined by *SMPTE ST 424* specification • 5.94-Gbps or 5.934-Gbps 6G-SDI, as defined by *SMPTE ST 2081* specification • 11.88-Gbps or 11.868-Gbps 12G-SDI, as defined by *SMPTE ST 2082* specification
	Features	• Automatic detection of SDI standards and video transport formats • Payload identification packet (ST 352) insertion and extraction • Cyclical redundancy check (CRC) encoding and decoding (except SD) • Line number (LN) insertion and extraction (except SD) • Framing and extraction of video timing signals • Dual link HD-SDI data stream synchronization (except SD) • 3G-SDI with data mapped by ST 425-*x* mapping • 6G-SDI with data mapped by ST 2081-*x* mapping • 12G-SDI with data mapped by ST 2082-*x* mapping • 20-bit interface support for SD-SDI • Dynamic TX clock switching to support integer and fractional video frame rates

continued...

Information		Description
	Applications	• Digital video equipment • Mixing and recording equipment
	Device Family Support	Intel Arria® 10, Intel Cyclone® 10 GX, Intel Stratix® 10 (L-tile and H-tile), Arria V, Arria V GZ, Cyclone V, and Stratix V FPGA device families.
	Design Tools	• IP Catalog in the Intel Quartus Prime software for design creation and compilation • ModelSim* - Intel FPGA Edition, ModelSim - Intel FPGA Starter Edition, Riviera-PRO*, VCS*/VCS MX, NCSim, and Xcelium* Parallel simulator software for design simulation or synthesis using Intel Quartus Prime tool

Related Information

- Introduction to Intel® FPGA Software Installation and Licensing
- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- SDI II Intel FPGA IP User Guide Archives on page 82
  Provides a list of user guides for previous versions of the SDI II Intel FPGA IP core.

2. SDI II Intel FPGA IP Core Overview

The SDI II Intel FPGA IP core implements a transmitter, receiver, or full-duplex SDI at standard definition (SD), high definition (HD), or 3 gigabits per second (3G) to 12G rate as defined by the Society of Motion Picture and Television Engineers (SMPTE). The SDI II Intel FPGA IP core supports dual rates (SD-SDI and HD-SDI), triple rates (SD-SDI, HD-SDI, and 3G-SDI) and multi rates (SD-SDI, HD-SDI, 3G-SDI, 6G-SDI, and 12G-SDI). These modes provide automatic receiver rate detection and transceiver dynamic reconfiguration.

The SDI II Intel FPGA IP core supports 28 nm devices and beyond.

Table 2. Intel Device Family Support

Device Family	Support Level
Intel Stratix 10—L-tile (from Intel Quartus Prime Pro Edition version 19.1 onwards)	Final
Intel Stratix 10—H-tile (from Intel Quartus Prime Pro Edition version 17.1 onwards)	Final
Intel Cyclone 10 GX (from Intel Quartus Prime Pro Edition version 17.1.1 onwards)	Final
Intel Arria 10 (from Intel Quartus Prime version 14.0A10 onwards)	Final
Arria V GZ and Cyclone V (from Intel Quartus Prime Standard Edition version 13.0 onwards)	Final
Arria V GX/GT/SX/ST and Stratix V (from Intel Quartus Prime Standard Edition version 12.1 onwards)	Final

The following terms define device support levels for Intel FPGA IP cores:

- Advance support—the IP core is available for simulation and compilation for this device family. Timing models include initial engineering estimates of delays based on early post-layout information. The timing models are subject to change as silicon testing improves the correlation between the actual silicon and the timing models. You can use this IP core for system architecture and resource utilization studies, simulation, pinout, system latency assessments, basic timing assessments (pipeline budgeting), and I/O transfer strategy (data-path width, burst depth, I/O standards tradeoffs).
- Preliminary support—the IP core is verified with preliminary timing models for this device family. The IP core meets all functional requirements, but might still be undergoing timing analysis for the device family. It can be used in production designs with caution.
- Final support—the IP core is verified with final timing models for this device family. The IP core meets all functional and timing requirements for the device family and can be used in production designs.

2.1. General Description

The SMPTE defines a SDI standard that is widely used as an interconnect between equipment in video production facilities. The SDI II Intel FPGA IP core can handle the following SDI data rates:

- 270 megabits per second (Mbps) SD-SDI, as defined by *SMPTE ST 259-1997 10-Bit 4:2:2 Component Serial Digital Interface*
- 1.485 gigabits per second (Gbps) or 1.4835-Gbps HD-SDI, as defined by *SMPTE ST 292-1998 Bit-Serial Digital Interface for High Definition Television Systems*
- 2.97-Gbps or 2.967-Gbps 3G SDI, as defined by *SMPTE ST 424*
- 5.94-Gbps or 5.934-Gbps 6G-SDI, as defined by *SMPTE ST 2081*
- 11.88-Gbps or 11.868-Gbps 12G-SDI, as defined by *SMPTE ST 2082*

Table 3. SDI II Intel FPGA Standard Support

Table below lists the SDI II Intel FPGA IP standard support for various FPGA devices.

Device Family	SDI Video Standard						
	Single Rate				Multiple Rates		
	SD-SDI	HD-SDI	3G-SDI	Dual Link HD-SDI	Dual Rate (up to HD)	Triple Rate (up to 3G)	Multi Rate (up to 12G)
Arria V GX/GT/SX/ST	Yes	Yes	Yes	Yes	Yes	Yes	No
Arria V GZ	Yes	Yes	Yes	Yes	Yes	Yes	No
Stratix V	Yes	Yes	Yes	Yes	Yes	Yes	No
Cyclone V	Yes	Yes	Yes	Yes	Yes	Yes	No
Intel Arria 10	No	Yes	Yes	No	No	Yes	Yes
Intel Stratix 10	No	Yes	Yes	No	No	Yes	Yes
Intel Cyclone 10 GX	No	Yes	Yes	No	No	Yes	Yes

2.2. Performance and Resource Utilization

The tables below list the typical resource utilization data and the recommended speed grades for the SDI II Intel FPGA IP core with the Intel Quartus Prime software, version 19.1.

Note: The resource utilization data was obtained by using the most common configurations for each video standard and from one specific variant of each device family.

Table 4. Resource Utilization for Each Video Standard for Intel Arria 10 and Intel Cyclone 10 GX Devices

Standard	ALMs Needed	Dedicated Logic Registers	Block Memory Bits
HD-SDI TX	100	144	0
HD-SDI RX	532	924	0
3G-SDI TX	372	404	0

continued...

Standard	ALMs Needed	Dedicated Logic Registers	Block Memory Bits
3G-SDI RX	842	1,506	0
Triple Rate TX	462	525	0
Triple Rate RX	1,082	1,807	0
Multi Rate (Up to 12G-SDI) TX	2,567	3,019	0
Multi Rate (Up to 12G-SDI) RX	4,168	5,898	0

Table 5. Resource Utilization for Each Video Standard for Intel Stratix 10 Devices

Standard	ALMs Needed	Dedicated Logic Registers	Block Memory Bits
HD-SDI TX	117	125	0
HD-SDI RX	632	883	0
3G-SDI TX	380	400	0
3G-SDI RX	981	1,378	0
Triple Rate TX	486	515	0
Triple Rate RX	1,269	1,744	0
Multi Rate (Up to 12G-SDI) TX	2,780	3,017	0
Multi Rate (Up to 12G-SDI) RX	5,124	5,951	0

Table 6. Resource Utilization for Each Video Standard for Arria V, Cyclone V, and Stratix V Devices

Standard	ALMs Needed	Dedicated Logic Registers	Block Memory Bits
SD-SDI TX	96	167	0
SD-SDI RX	502	693	60
HD-SDI TX	146	213	0
HD-SDI RX	542	929	0
HD Dual Link TX	452	553	0
HD Dual Link RX	1,249	2,154	4,608
3G-SDI TX	448	468	0
3G-SDI RX	863	1,449	0
Dual Rate TX	252	264	0
Dual Rate RX	930	1,348	0
Triple Rate TX	514	567	0
Triple Rate RX	1,115	1,763	0

Table 7. Recommended Speed Grades

Device Family	FPGA Fabric Speed Grade
Arria V GX/GT/SX/ST	Any supported speed grade
Arria V GZ	Any supported speed grade
Cyclone V	–6, –7
Stratix V	Any supported speed grade
Intel Arria 10	Any supported speed grade
Intel Stratix 10	Any supported speed grade
Intel Cyclone 10 GX	Any supported speed grade

3. SDI II Intel FPGA IP Core Getting Started

3.1. Installing and Licensing Intel FPGA IP Cores

The Intel Quartus Prime software installation includes the Intel FPGA IP library. This library provides many useful IP cores for your production use without the need for an additional license. Some Intel FPGA IP cores require purchase of a separate license for production use. The Intel FPGA IP Evaluation Mode allows you to evaluate these licensed Intel FPGA IP cores in simulation and hardware, before deciding to purchase a full production IP core license. You only need to purchase a full production license for licensed Intel IP cores after you complete hardware testing and are ready to use the IP in production.

The Intel Quartus Prime software installs IP cores in the following locations by default:

Figure 1. IP Core Installation Path

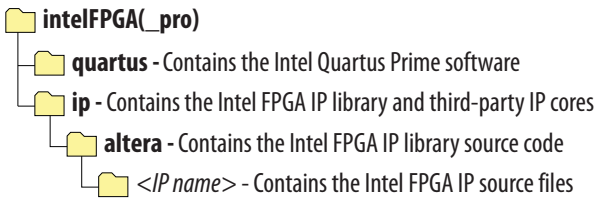


Table 8. IP Core Installation Locations

Location	Software	Platform
`<drive>:\intelFPGA_pro\quartus\ip\altera`	Intel Quartus Prime Pro Edition	Windows*
`<drive>:\intelFPGA\quartus\ip\altera`	Intel Quartus Prime Standard Edition	Windows
`<home directory>:/intelFPGA_pro/quartus/ip/altera`	Intel Quartus Prime Pro Edition	Linux*
`<home directory>:/intelFPGA/quartus/ip/altera`	Intel Quartus Prime Standard Edition	Linux

Note: The Intel Quartus Prime software does not support spaces in the installation path.

3.1.1. Intel FPGA IP Evaluation Mode

The free Intel FPGA IP Evaluation Mode allows you to evaluate licensed Intel FPGA IP cores in simulation and hardware before purchase. Intel FPGA IP Evaluation Mode supports the following evaluations without additional license:

- Simulate the behavior of a licensed Intel FPGA IP core in your system.
- Verify the functionality, size, and speed of the IP core quickly and easily.
- Generate time-limited device programming files for designs that include IP cores.
- Program a device with your IP core and verify your design in hardware.

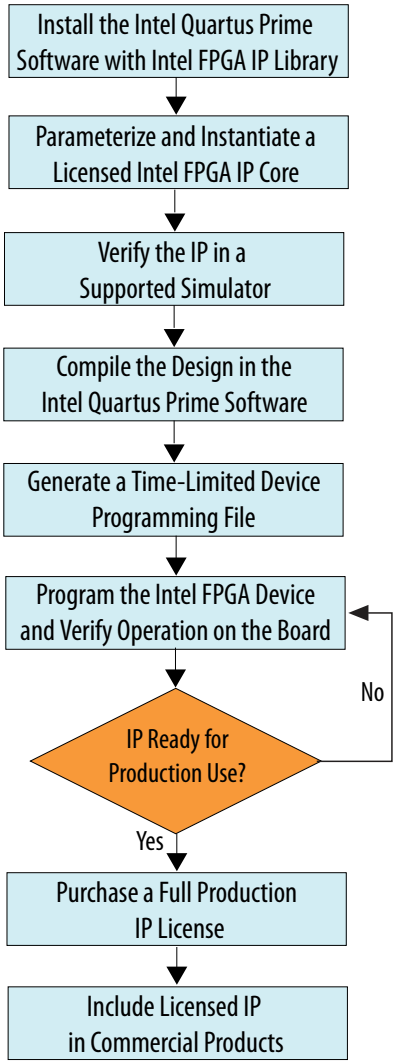
Intel FPGA IP Evaluation Mode supports the following operation modes:

- **Tethered**—Allows running the design containing the licensed Intel FPGA IP indefinitely with a connection between your board and the host computer. Tethered mode requires a serial joint test action group (JTAG) cable connected between the JTAG port on your board and the host computer, which is running the Intel Quartus Prime Programmer for the duration of the hardware evaluation period. The Programmer only requires a minimum installation of the Intel Quartus Prime software, and requires no Intel Quartus Prime license. The host computer controls the evaluation time by sending a periodic signal to the device via the JTAG port. If all licensed IP cores in the design support tethered mode, the evaluation time runs until any IP core evaluation expires. If all of the IP cores support unlimited evaluation time, the device does not time-out.
- **Untethered**—Allows running the design containing the licensed IP for a limited time. The IP core reverts to untethered mode if the device disconnects from the host computer running the Intel Quartus Prime software. The IP core also reverts to untethered mode if any other licensed IP core in the design does not support tethered mode.

When the evaluation time expires for any licensed Intel FPGA IP in the design, the design stops functioning. All IP cores that use the Intel FPGA IP Evaluation Mode time out simultaneously when any IP core in the design times out. When the evaluation time expires, you must reprogram the FPGA device before continuing hardware verification. To extend use of the IP core for production, purchase a full production license for the IP core.

You must purchase the license and generate a full production license key before you can generate an unrestricted device programming file. During Intel FPGA IP Evaluation Mode, the Compiler only generates a time-limited device programming file (*<project name>*`_time_limited.sof`) that expires at the time limit.

Figure 2. Intel FPGA IP Evaluation Mode Flow



Note: Refer to each IP core's user guide for parameterization steps and implementation details.

Intel licenses IP cores on a per-seat, perpetual basis. The license fee includes first-year maintenance and support. You must renew the maintenance contract to receive updates, bug fixes, and technical support beyond the first year. You must purchase a full production license for Intel FPGA IP cores that require a production license, before generating programming files that you may use for an unlimited time. During Intel FPGA IP Evaluation Mode, the Compiler only generates a time-limited device programming file (`<project name>_time_limited.sof`) that expires at the time limit. To obtain your production license keys, visit the Self-Service Licensing Center.

The Intel FPGA Software License Agreements govern the installation and use of licensed IP cores, the Intel Quartus Prime design software, and all unlicensed IP cores.

Related Information

- Intel Quartus Prime Licensing Site
- Introduction to Intel FPGA Software Installation and Licensing

3.2. Design Walkthrough

This walkthrough explains how to create an SDI II Intel FPGA IP core design using the Intel Quartus Prime software and IP Catalog. After you generate a custom variation of the SDI II Intel FPGA IP core, you can incorporate it into your overall project.

This walkthrough includes the following steps:

1. Creating a New Intel Quartus Prime Project on page 13
2. Launching IP Catalog on page 14
3. Parameterizing the IP Core on page 14
4. Generating a Design Example and Simulation Testbench on page 15
5. Simulating the SDI II Intel FPGA IP Core Design on page 80

3.2.1. Creating a New Intel Quartus Prime Project

You need to create a new Intel Quartus Prime project with the **New Project Wizard**, which specifies the working directory for the project, assigns the project name, and designates the name of the top-level design entity.

To create a new project, perform the following the steps.

1. From the Windows Start menu, select **All Programs ➤ Intel FPGA <version number> <edition> ➤ Intel Quartus Prime <edition> <version>**.
2. On the **File** menu, click **New Project Wizard**.
3. In the **New Project Wizard: Directory, Name, Top-Level Entity** page, specify the working directory, project name, and top-level design entity name. Click **Next**.
4. In the **New Project Wizard: Add Files** page, select the existing design files (if any) you want to include in the project.(1) Click **Next**.
5. In the **New Project Wizard: Family & Device Settings** page, select the device family and specific device you want to target for compilation. Click **Next**.
6. In the **EDA Tool Settings** page, select the EDA tools you want to use with the Intel Quartus Prime software to develop your project.
7. The last page in the **New Project Wizard** window shows the summary of your chosen settings. Click **Finish** to complete the Intel Quartus Prime project creation.

(1) To include existing files, you must specify the directory path to where you installed the SDI II Intel FPGA IP core. You must also add the user libraries if you installed the IP Library in a different directory from where you installed the Intel Quartus Prime software.

3.2.2. Launching IP Catalog

To launch the IP Catalog in the Intel Quartus Prime software, follow these steps:

1. On the Tools menu, click **IP Catalog**.
2. Expand the **Interface Protocols> Audio & Video** folder and double-click **SDI II Intel FPGA** to launch the parameter editor.

   The parameter editor prompts you to specify your FPGA IP variation name, optional ports, architecture features, and output file generation options. The parameter editor generates a top-level **.qsys** or **.ip** file representing the FPGA IP core in your project.
3. Click **OK** to display the SDI II Intel FPGA IP core parameter editor.

3.2.3. Parameterizing the IP Core

To parameterize the SDI II Intel FPGA IP core, follow these steps:

1. Select the video standard.
2. Select **Bidirectional**, **Transmitter**, or **Receiver** interface direction.
3. Select **Combined Transceiver and Protocol**, **Separate Transceiver** or **Separate Protocol**, (for Arria V, Cyclone V, and Stratix V devices only).
4. Turn on the necessary transceiver options (for Arria V, Cyclone V, and Stratix V devices only).
5. Turn on the necessary receiver options.

   Some options may be grayed out, because they are not supported in the currently selected configuration.
6. Turn on the necessary transmitter options.

   Some options may be grayed out, because they are not supported in the currently selected configuration.
7. Select the necessary options in the **Design Example** tab, (if you are generating the design example for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices).
8. Click **Finish**.

Related Information

- SDI II Intel FPGA IP Core Parameters on page 17
- Design Examples for Arria V, Cyclone V, and Stratix V Devices on page 70
- SDI II Intel Stratix 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Stratix 10 devices.
- SDI II Intel Arria 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Arria 10 devices.
- SDI II Intel Cyclone 10 GX FPGA IP Design Example User Guide
  Provides the design examples for Intel Cyclone 10 GX devices.

3.2.4. Generating a Design Example and Simulation Testbench

After you have parameterized the SDI II Intel FPGA IP core, click **Generate Example Design** to create the following entities:

- Design example— serves as a common entity for simulation and hardware verification.
- Simulation testbench—consists of the design example entity and other non-synthesizable components. The example testbench and the automated script are located in:
  - Arria V, Cyclone V, and Stratix V: `<variation name>_example_design/sdi_ii/simulation/verilog`

    or `<variation name>_example_design/sdi_ii/simulation/vhdl` directory.
  - Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10: `<your design example folder>/simulation` directory.

Note: Generating a design example can increase processing time.

You can now integrate your custom IP core variation into your design, simulate, and compile.

3.3. SDI II Intel FPGA IP Core Component Files

Table 9. Generated Files

Table below describes the generated files and other files that might be in your project directory. The names and types of files vary depending on whether you create your design with VHDL or Verilog HDL.

Extension	Description
`<variation name>.sv`	An IP core variation file, which defines a Verilog HDL description of the custom IP core. Instantiate the entity defined by this file inside your design.
`<variation name>.v` (Arria V, Cyclone V, and Stratix V devices)	
`<variation name>.qsys` (Intel Arria 10 on Intel Quartus Prime Standard Edition)	
`<variation name>.ip` (Intel Quartus Prime Pro Edition)	
`<variation name>.sdc`	Contains timing constraints for your SDI variation.
`<variation name>.qip`	Contains Intel Quartus Prime project information for your IP core variations. Add this file in your Intel Quartus Prime project before you compile your design in the Intel Quartus Prime software.

3.4. Compiling the SDI II Intel FPGA IP Core Design

To compile your design, click **Processing ➤ Start Compilation** in the Intel Quartus Prime software. Use the generated `.qip` or `.ip` file to include the relevant files into your project.

You can find the design examples of the SDI II Intel FPGA IP core in:

- Arria V, Cyclone V, and Stratix V: `<variation name>_example_design/sdi_ii/example_design/sdi_ii_0001_ed` directory.
- Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10: `<your design example folder>/rtl` directory.

Note: To create a new project using the generated design example, follow the steps in the *Creating a New Intel Quartus Prime Project* section and add the design example `.qip` file in step 4.

Related Information

- Creating a New Intel Quartus Prime Project on page 13
- Compilation
  Provides more information about compiling designs and compiler settings.
- Design Examples for Arria V, Cyclone V, and Stratix V Devices on page 70
  Provides the design examples for Arria V, Cyclone V, and Stratix V devices.
- SDI II Intel Stratix 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Stratix 10 devices.
- SDI II Intel Arria 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Arria 10 devices.
- SDI II Intel Cyclone 10 GX FPGA IP Design Example User Guide
  Provides the design examples for Intel Cyclone 10 GX devices.

3.5. Programming an FPGA

After successfully compiling your design, program the targeted Intel FPGA with the Intel Quartus Prime Programmer and verify the design in hardware.

For instructions on programming the FPGA device, refer to the *Device Programming* section in volume 3 of the Intel Quartus Prime Handbook.

Related Information

Programming Intel FPGA Devices

4. SDI II Intel FPGA IP Core Parameters

Note: For SDI II Intel FPGA design example parameters, refer to the respective SDI II Intel FPGA design example user guides.

Table 10. SDI II Intel FPGA IP Core Parameters

Note: **Transceiver Options** are available only for Arria V, Cyclone V, and Stratix V devices.

Parameter	Value	Description
Configuration Options		
Video standard	SD-SDI, HD-SDI, 3G-SDI, HD-SDI dual link, Dual rate (up to HD-SDI), Triple rate (up to 3G-SDI), Multi rate (up to 12G-SDI)	Sets the video standard. • SD-SDI—disables option for line insertion and extraction, and CRC generation and extraction • HD-SDI—enables option for in line insertion and extraction and CRC generation and extraction • Dual-, triple-, or multi-rate SDI—includes the processing blocks for the respective supported rates. Logics for bypass paths and to automatically switch between the input standards are included. *Note:* **SD-SDI**, **HD-SDI dual link**, and **Dual rate (up to HD-SDI)** options are not available for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices. **Multi rate (up to 12G-SDI)** option is not available for Arria V, Cyclone V, and Stratix V devices.
SD interface bit width	10, 20	Selects the SD interface bit width. Only applicable for dual rate and triple rate.
Direction	Bidirectional, Receiver, Transmitter	Sets the port direction. The selection enables or disables the receiver and transmitter supporting logic appropriately. • Bidirectional—instantiates both the SDI transmitter and receiver. • Receiver—instantiates the SDI receiver • Transmitter—instantiates the SDI transmitter.
Transceiver and/or Protocol	Combined, Transceiver, Protocol	Selects the transceiver or protocol components, or both. • Transceiver—includes tx/rx_phy_mgmt/phy_adapter and Native PHY IP. This option is useful if you want to use the same transceiver component to support both SDI and ASI IP cores. • Protocol—allows each submodule to be removed or reused across different video standards. The transmitter and receiver data paths are independent from each other. *Note:* This option is available only for Arria V, Cyclone V, and Stratix V devices.
Transceiver Options		
Transceiver reference clock frequency	148.5/148.35 MHz, 74.25/74.175 MHz,	Selects the transceiver reference clock frequency. The **74.25/74.175 MHz** option is available only for HD-SDI and HD-SDI dual link video standards, and if you select **CMU** as the TX PLL. *Note:* This option is not available if you select ATX PLL.
TX PLL type	CMU, ATX	Selects the transmitter PLL for TX or bidirectional ports.

continued...

Transceiver Options		
		ATX PLL is useful for bidirectional channels—you can use the ATX PLL as the transmitter PLL instead of the CMU PLL from another channel. *Note:* This option is not available if you select ATX PLL.
Dynamic Tx clock switching	Off, Tx PLL switching, Tx PLL reference clock switching	• Off: Disable dynamic switching • Tx PLL switching: Instantiates two PLLs, each with a reference input clock • Tx PLL reference clock switching: Instantiates a PLL with two reference input clocks. Turn on this option to allow dynamic switching between 1 and 1/1.001 data rates. *Note:* This option is only available for Arria V, Cyclone V, and Stratix V devices using TX or bidirectional ports, and all video standards except SD-SDI.

Receiver Options		
Increase error tolerance level	On, Off	• On: Error tolerance level = 15 • Off: Error tolerance level = 4 Turn on this option to increase the tolerance level for consecutive missed end of active videos (EAVs), start of active videos (SAVs), or erroneous frames.
CRC error output	On, Off	• On: CRC monitoring (Not applicable for SD-SDI mode) • Off: No CRC monitoring (saves logic)
Extract Payload ID (SMPTE ST 352)	On, Off	• On: Extract payload ID • Off: No payload ID extraction (saves logic) You must turn on this option for 3G-SDI, HD SDI dual link, triple-rate, and multi-rate modes. The extracted payload ID is required for consistent detection of the 1080p format. It is compulsory to turn on this option for design example demonstration when you turn on **Convert HD-SDI dual link to 3G-SDI (level B)** or **Convert 3G-SDI (level B) to HD-SDI dual link**.
Rx core clock (rx_coreclk) frequency	148.5/148.35 MHz, 297.0/296.70 MHz	Selects the supported clock frequency for the `rx_coreclk` signal. This option is only available when you select **Multi rate (up to 12G-SDI)** in **Receiver** or **Bidirectional** mode. For other standards, the default frequency is 148.5/148.35 MHz. *Note:* This option is only available for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices in the Intel Quartus Prime Pro Edition software.
Convert HD-SDI dual link to 3G-SDI (level B)	On, Off	• On: Converts to level B (2 × SMPTE ST 292 HD-SDI mapping, including SMPTE ST 372 dual link mapping) for HD-SDI dual link receiver output. • Off: No conversion *Note:* This option is only available for Arria V, Cyclone V, and Stratix V devices using HD-SDI dual link receiver.
Convert 3G-SDI (level B) to HD-SDI dual link	On, Off	• On: Converts to HD-SDI dual link (direct image format mapping) for 3G-SDI receiver output. • Off: No conversion *Note:* This option is only available for Arria V, Cyclone V, and Stratix V devices using 3G-SDI receiver.

Transmitter Options		
Insert payload ID (SMPTE ST 352)	On, Off	• On: Insert payload ID • Off: No payload ID insertion (saves logic)

5. SDI II Intel FPGA IP Core Functional Description

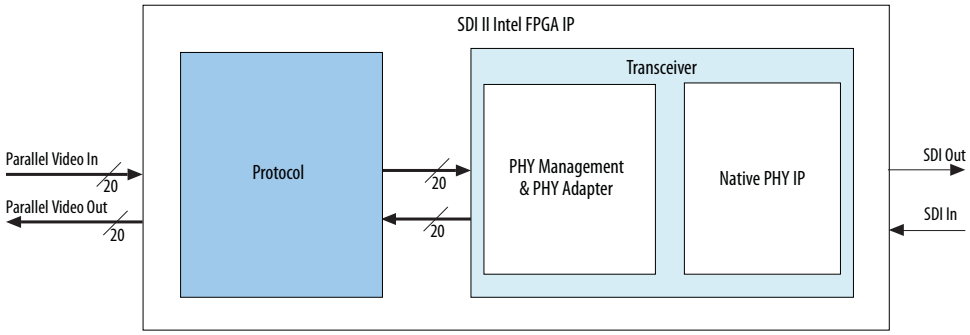
The SDI II Intel FPGA IP core implements a transmitter, receiver, or full-duplex interface.

The SDI II Intel FPGA IP core consists of the following components:

- Protocol block—transmitter or receiver
- Transceiver blocks—PHY management & adapter and Native PHY IP

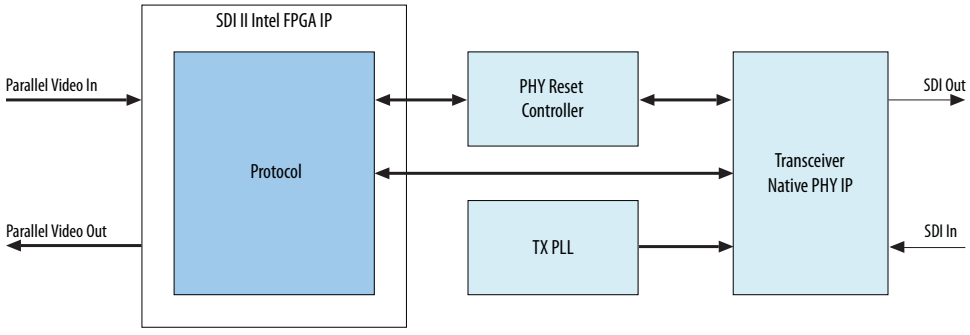
In the parameter editor, you can specify either protocol, transceiver, or combined blocks for your design. For example, if you have multiple protocol blocks in a design, you can multiplex them into one transceiver.

Figure 3. SDI II Intel FPGA IP Core Block Diagram for Arria V, Cyclone V, and Stratix V Devices



For the Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, the SDI II Intel FPGA IP core no longer provides the transceiver, and the TX PLL is no longer wrapped in the transceiver PHY. You must generate the transceiver and the TX PLL separately.

Figure 4. SDI II Intel FPGA IP Core Block Diagram for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 Devices



5.1. Protocol

The protocol block handles the SDI-specific parts of the core and generally operates on a parallel domain data.

5.1.1. Transmitter

The transmitter performs the following functions:

- HD-SDI LN insertion
- Sync bit insertion
- HD-SDI CRC generation and insertion
- Payload ID insertion
- Matching timing reference signal (TRS) word
- Clock enable signal generation
- Scrambling and non-return-zero inverted (NRZI) coding

The block diagrams below illustrate the SDI II Intel FPGA IP core transmitter (simplex) data path for each supported video standard.

For more information about the function of each submodule, refer to the *Submodules* section.

Figure 5. SD-SDI Transmitter Data Path Block Diagram

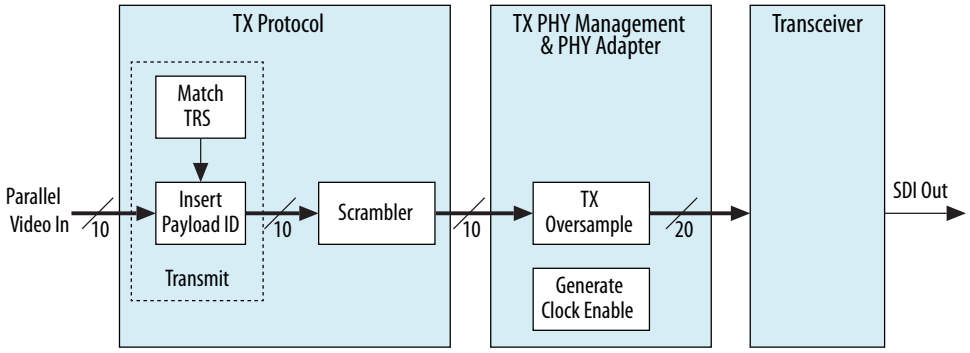


Figure 6. HD/3G-SDI Transmitter Data Path Block Diagram

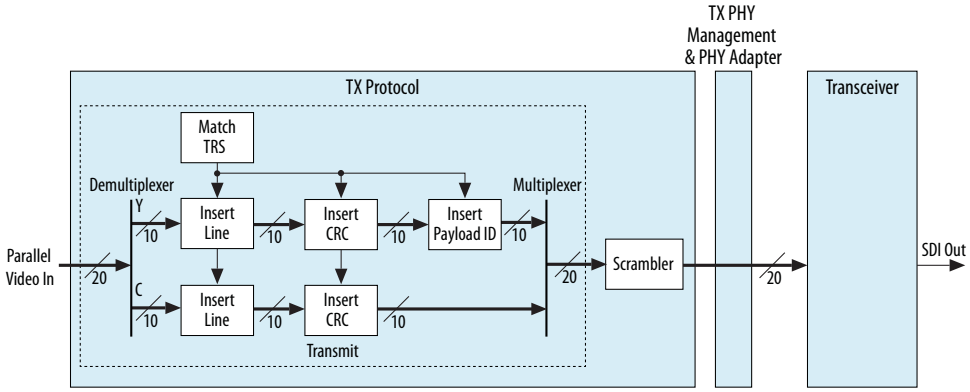


Figure 7. Dual Rate SDI Transmitter Data Path Block Diagram

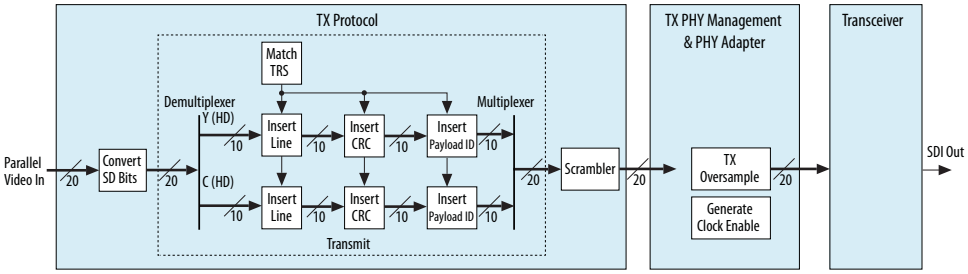


Figure 8. Dual Link HD-SDI Transmitter Data Path Block Diagram

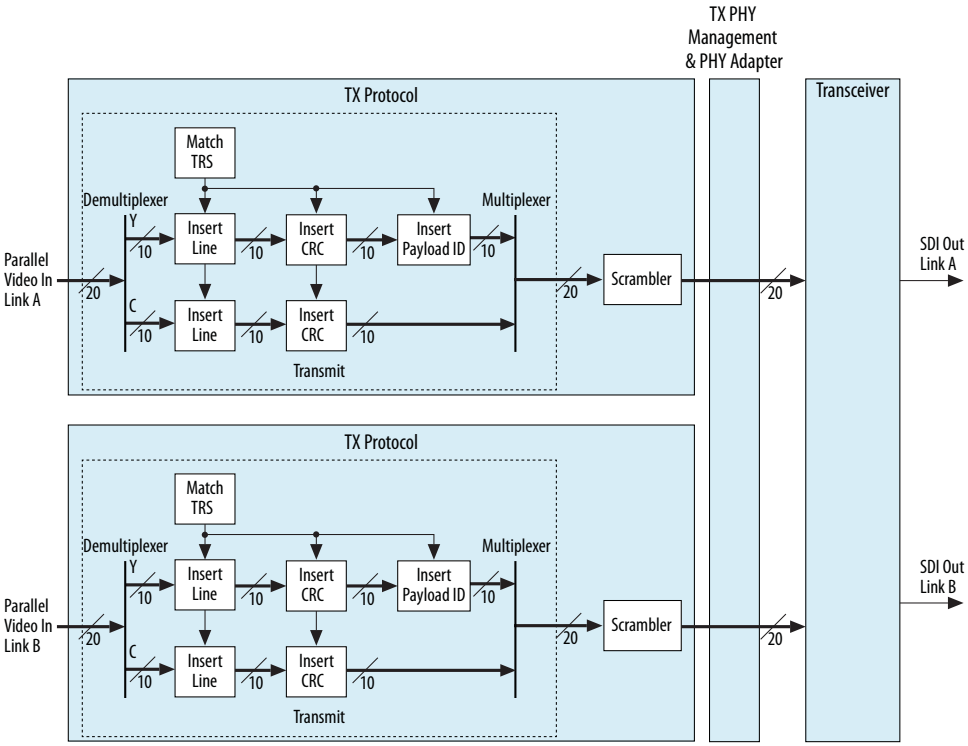


Figure 9. Triple Rate SDI Transmitter Data Path Block Diagram

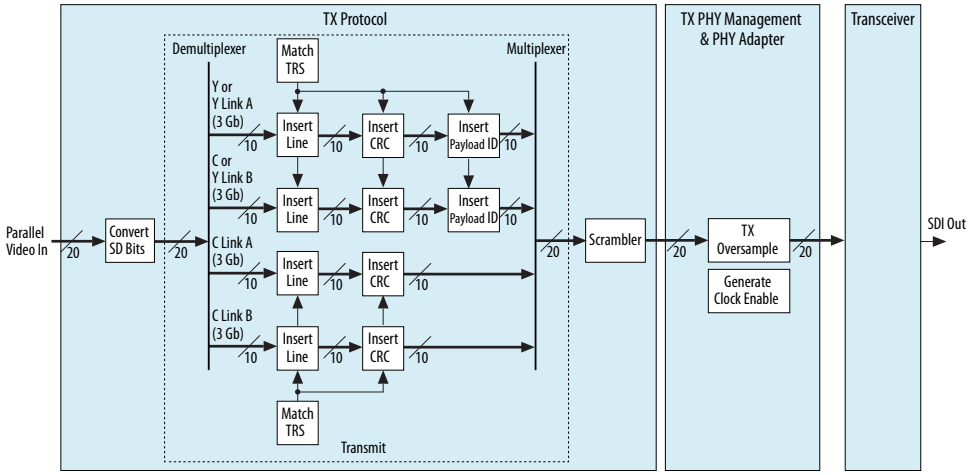
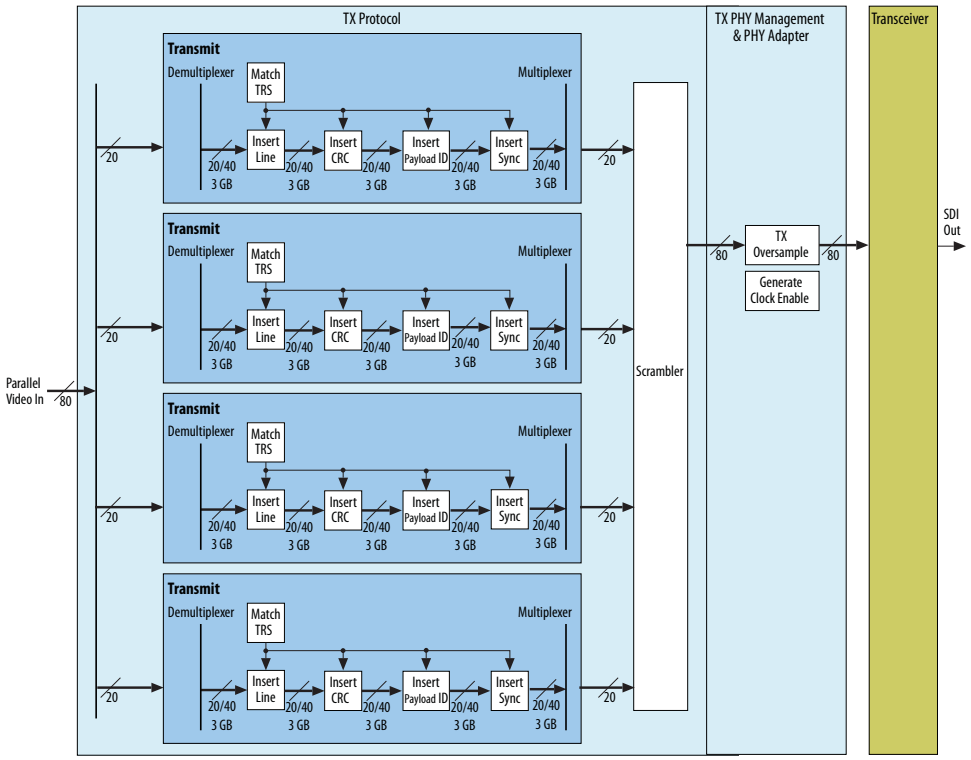


Figure 10. Multi Rate (up to 12G-SDI) Transmitter Data Path Block Diagram

Note: The transmit block shown in the diagram is the simplified version of the transmit block in the *Triple Rate SDI Transmitter Data Path Block Diagram*.



Related Information

Submodules on page 28

5.1.2. Receiver

The receiver performs the following functions:

- Video standard detection
- Video rate detection
- NRZI decoding and descrambling
- Word alignment
- Demultiplex data links
- Video timing flags extraction
- HD-SDI LN extraction
- HD-SDI CRC
- Payload ID extraction
- Synchronizing data streams
- Accessing transceiver
- Identifying and tracking of ancillary data
- Sync bit removal

The block diagrams below illustrate the SDI II Intel FPGA IP core receiver (simplex) data path for each supported video standard.

Figure 11. SD-SDI Receiver Data Path Block Diagram

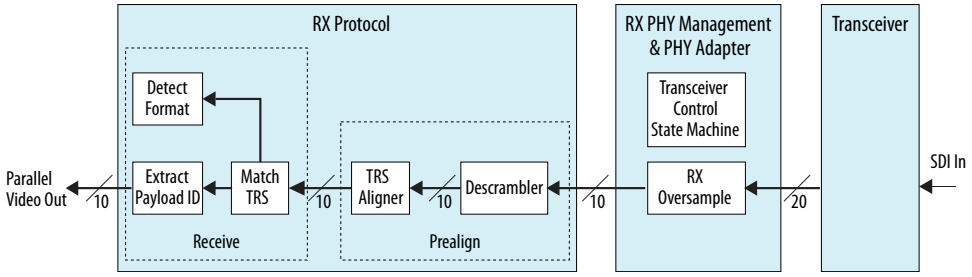


Figure 12. HD-SDI Receiver Data Path Block Diagram

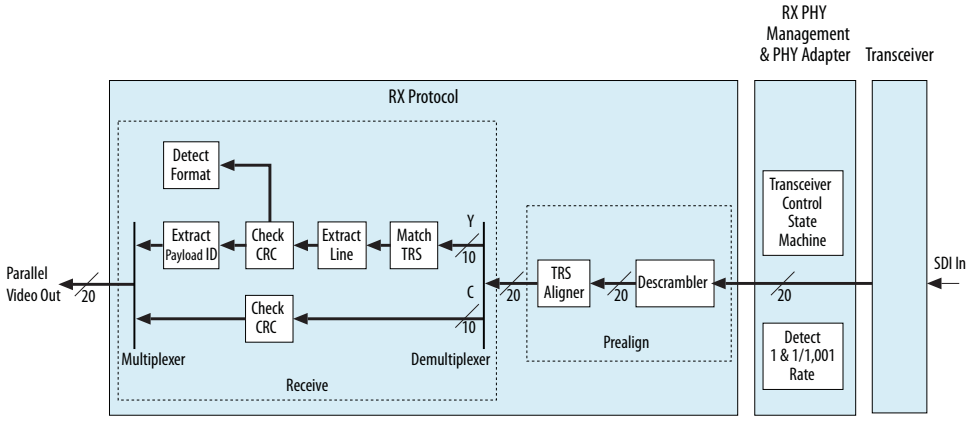


Figure 13. 3G-SDI Receiver Data Path Block Diagram

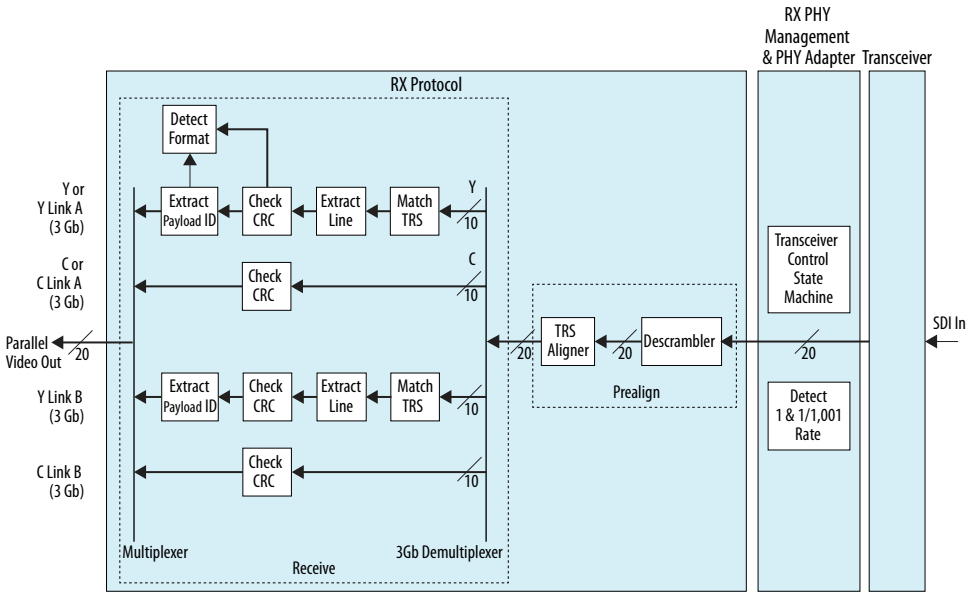


Figure 14. Dual Rate SDI Receiver Data Path Block Diagram

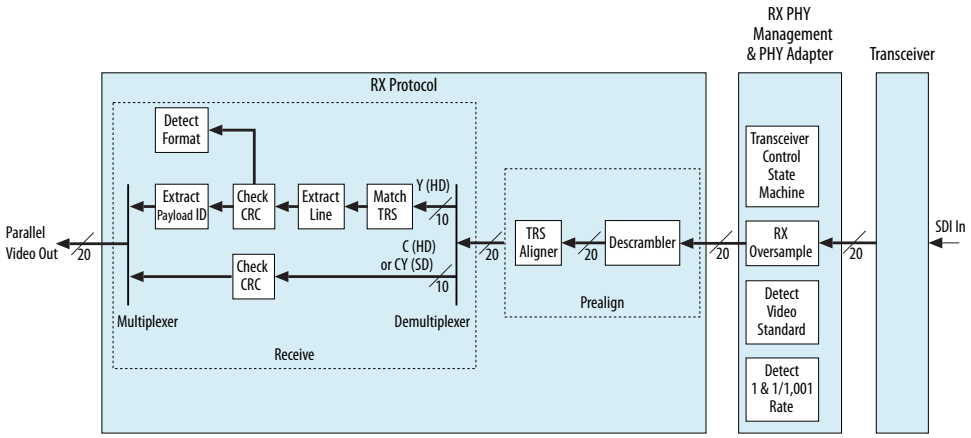


Figure 15. Dual Link HD-SDI Receiver Data Path Block Diagram

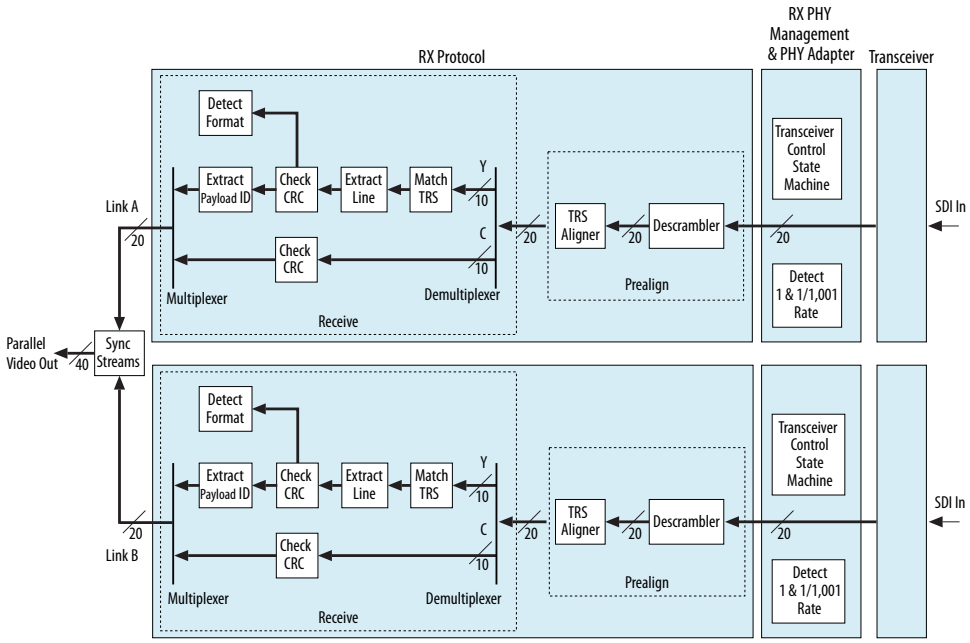


Figure 16. Triple Rate SDI Receiver Data Path Block Diagram

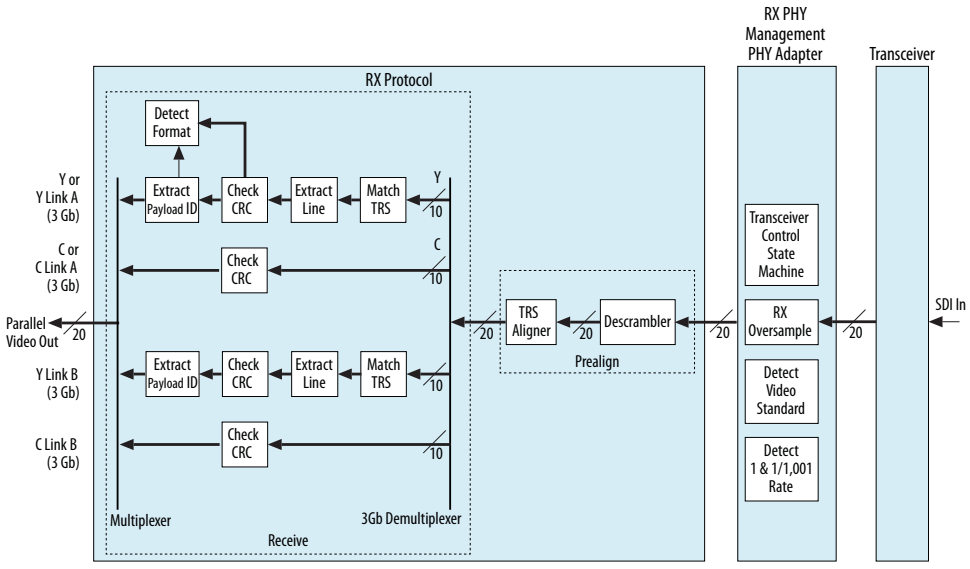
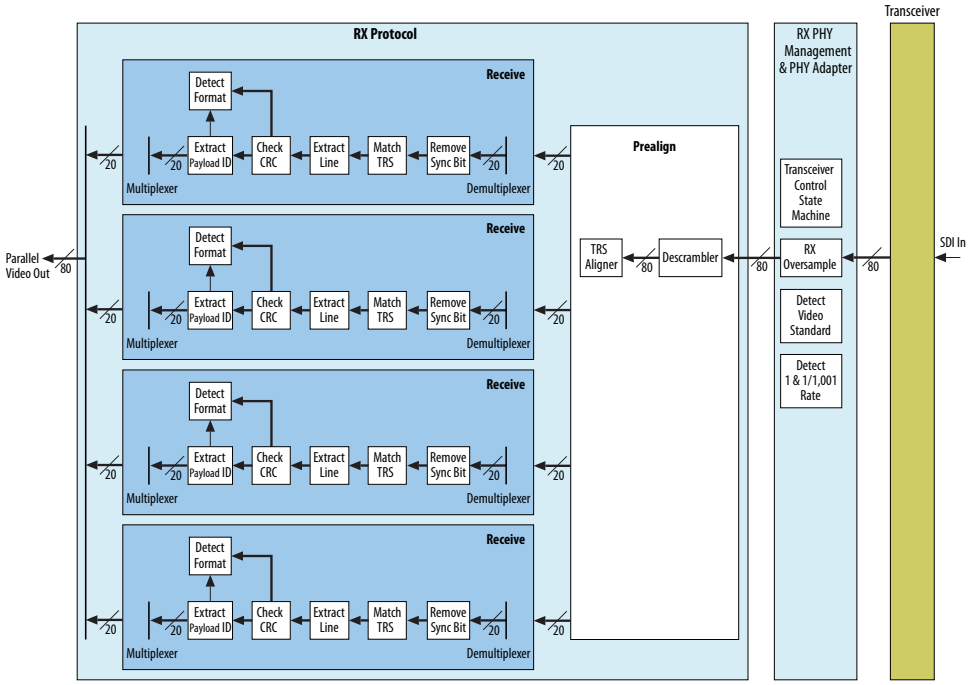


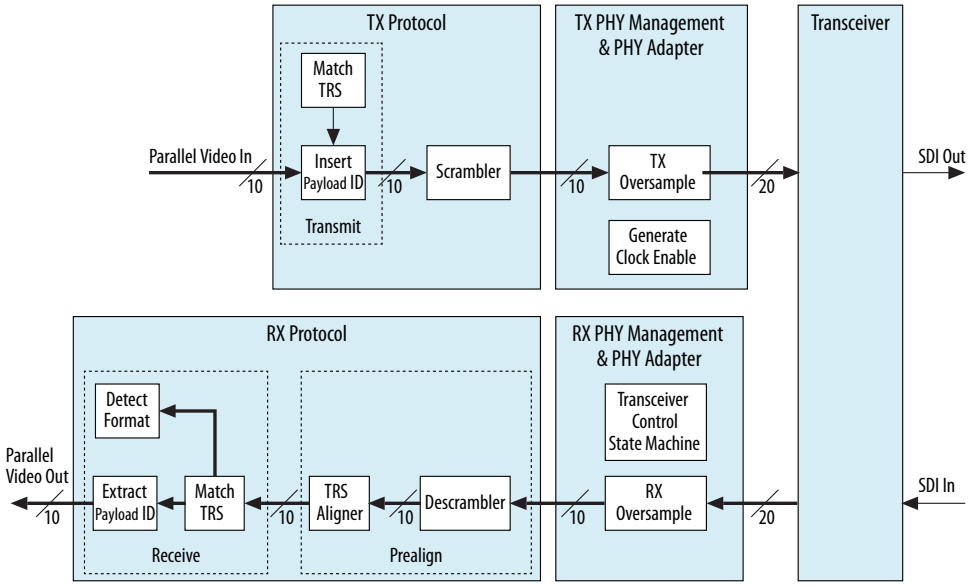
Figure 17. Multi Rate (up to 12G-SDI) Receiver Data Path Block Diagram

Note: The receive block shown in the diagram is the simplified version of the transmit block in the *Triple Rate SDI Receiver Data Path Block Diagram*.



For bidirectional or duplex mode, the protocol and PHY management & adapter blocks remain the same for each direction, except the Native PHY IP core, which is configured in duplex mode. The figure below illustrates the data path of a SD-SDI duplex mode.

Figure 18. SD-SDI Duplex Mode Block Diagram



5.2. Transceiver

The transceiver block consists of two components:

- PHY management and adapter
- Native PHY IP

These two components handle the serial transport aspects of the SDI II Intel FPGA IP core.

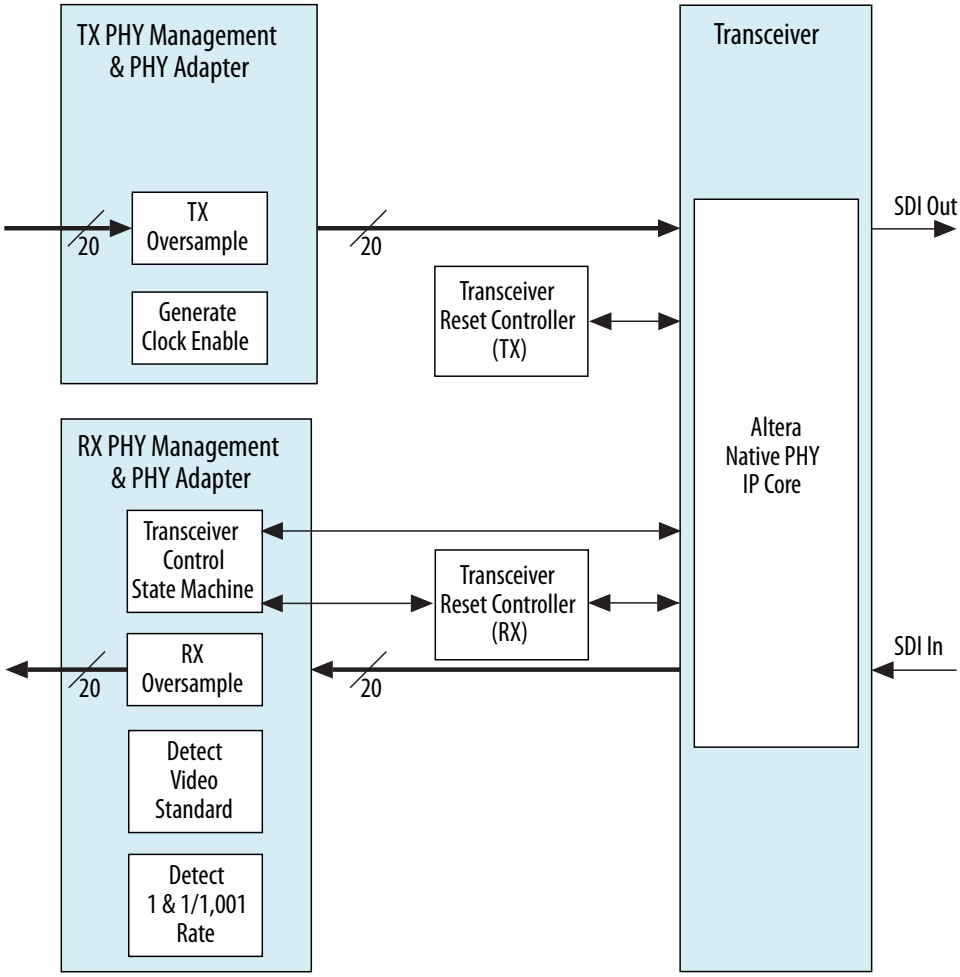
Note: The transceiver block is only available for Arria V, Cyclone V, and Stratix V devices. For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, you must generate the transceiver and the TX PLL instances separately; refer to Table 24 on page 64.

For Arria V, Cyclone V, and Stratix V devices, the SDI II Intel FPGA IP core instantiates the Native PHY IP core using the Tcl file associated with each device.

The block diagram below illustrates the Native PHY IP core setup in the SDI II Intel FPGA IP core (duplex) data path.

Figure 19. Native PHY IP Core Setup in Duplex Mode

The Native PHY IP core does not include an embedded reset controller and an Avalon® Memory-Mapped (Avalon-MM) interface. This PHY IP core exposes all signals directly as ports. To implement reset functionality for a new IP core, the transceiver reset controller is required to handle all the transceiver reset sequencing. The transceiver reset controller controls the embedded reset controller and also manages additional control options such as automatic or manual reset recovery mode.



Related Information

V-Series Transceiver PHY FPGA IP User Guide
Provides more information about the Native PHY IP core.

5.3. Submodules

5.3.1. Insert Line

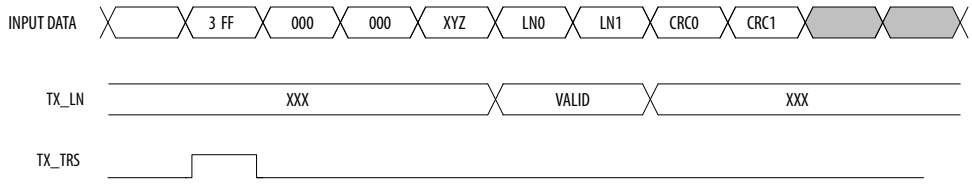
The insert line submodule provides HD-SDI and higher standards the option to include line numbers along with the video data.

This information is at the end of active video (EAV) extension words of the data stream, as defined in the *SMPTE ST 292* specification. The line number is 11 bits wide and spreads over two SDI words to use the SDI legal data space.

This submodule takes the 11-bit line number data value, correctly encodes them, and inserts them into the 10-bit stream. The line number value is user-defined. The top level port signal is `tx_ln[10:0]` and `tx_ln_b[10:0]` for link B in 3G-SDI (level B) and HD dual link modes. You also have the option to enable or disable this feature using the `tx_enable_ln` signal at the top level port. The SDI II Intel FPGA IP core inserts the same line number value into both video channels. The Y and C channels require two of these submodules.

Figure 20. Line Number Insertion and Signal Requirements

This figure illustrates the line number insertion and signal requirements. For a correct line insertion, assert the `tx_trs` signal for the first word of both EAV and start of active video (SAV) TRS.



5.3.2. Insert/Check CRC

The HD-SDI can optionally include a line-based CRC code, which makes up two of the EAV extension words as defined in the *SMPTE ST 292* specification.

This submodule calculates the CRC based on the LFSR approach in the SMPTE specification. Note that you can configure this submodule to either insert or check the CRC.

For the transmitter, the core formats and inserts the CRC into two CRC EAV extension words—CRC0 and CRC1. For correct CRC generation and insertion, assert the `tx_trs` signal for the first word of both EAV and SAV TRS as shown in the *Line Number Insertion* timing diagram. Perform CRC insertion only when the top level port, `tx_enable_crc`, is set to logic 1.

For the receiver, the core checks the CRC against the value of CRC0 and CRC1 that appear in the incoming stream. If there is a mismatch between the locally calculated value and the value in the stream, this submodule indicates an error.

Related Information

Insert Line on page 28

5.3.3. Insert Payload ID

The *SMPTE ST 352* specification defines an ancillary packet type that provides specific information about the video payload carried by a digital interface. These payload ID packets carry information such as the interface type, sampling structure, component bit depth, and picture update rate.

Recent SMPTE interfaces such as dual link HD-SDI and 3G-SDI require the payload ID packets because it is very difficult to properly interpret the video data without the packet information from the payload ID packets.

The payload ID packet must be on specific video line locations at the beginning of the horizontal ancillary (HANC) space in one of these two conditions:

- Right after the EAV.
- Right after the CRC words that follow the EAV (for interfaces using CRC words).

Table 11. Recommended Payload ID Packet Location

The table below lists the payload ID packet location recommended by SMPTE specification. You may observe SDI data having payload ID packets located in different line numbers.

Video Format	Field	Line Number
525i	1	13
	2	276
625i	1	9
	2	322
1080i	1	10
	2	572
525p	—	13
625p	—	9
720p	—	10
1080p	—	10

For dual link HD-SDI interface, the payload ID packets are placed only in the Y data stream of both links. This submodule in the transmitter data path modifies the Y data stream that passes through.

Note: This submodule introduces a latency of a few clock cycles. The C data stream is delayed by a few clock cycles to keep it synchronized with the Y data stream.

The following rules apply for inserting and overwriting payload ID packets:

- Rule 1: If there is no ancillary packet at the beginning of the HANC space on a line where the payload ID packet is supposed to occur, the submodule inserts the payload ID packet at the beginning of the HANC space.
- Rule 2: If there is an existing payload ID packet at the beginning of the HANC space on a line specified by `tx_line_f0` or `tx_line_f1`, the submodule overwrites the packet with the new payload ID information if the `tx_vpid_overwrite` signal is high. If the `tx_vpid_overwrite` signal is low, the submodule does not overwrite.
- Rule 3: If there is a different type of ancillary packet(s) at the beginning of the HANC space on a line where the payload ID packet is supposed to occur, the submodule does not overwrite the existing ancillary packet(s). Instead, the submodule looks for empty space in the HANC space to insert the payload ID packet after the existing ancillary packet(s). If the submodule finds a payload ID packet later in the HANC space before finding an empty space, it overwrites the existing payload ID packet with the new data if the `tx_vpid_overwrite` signal is high. If the `tx_vpid_overwrite` signal is low, the submodule will not overwrite.

For correct payload ID insertion, assert the `tx_trs` signal for the first word of both EAV and SAV TRS as shown in the *Line Number Insertion* timing diagram.

Related Information

Insert Line on page 28

5.3.4. Match TRS

This submodule indicates that the current word is a particular TRS word in both the transmitter and receiver.

5.3.5. Scrambler

The *SMPTE ST 259* and *SMPTE ST 292* specifications define a common channel coding for both SD-SDI and HD-SDI. This channel coding consists of a scrambling function ($G_1(X) = X^9 + X^4 + 1$), followed by NRZI encoding ($G_2(X) = X + 1$).

The scrambling submodule implements the channel coding by iteratively applying the scrambling and NRZI encoding algorithm to each bit of the output data, processing the LSB first. The code handles all transmit data: SD (10 bits wide), HD/3G (20 bits wide), 6G (40 bits wide), and 12G (80 bits wide).

5.3.6. TX Sample

The TX sample submodule is a transmit oversampling block. It repeats each bit of the input word a given number of times and constructs the output words.

This submodule relies on the fact that the input data is only valid on 1/*x* of the clock cycles, where *x* is the oversampling factor. Both the input and output words are clocked from the same clock domain.

Table 12. Oversampling Requirement

The table below lists the number of times oversampling is required for the different video standards.

Real Video Rate vs. IP Mode	SD-SDI	HD-SDI	Dual Rate	Triple Rate	Multi Rate
SD-SDI	11	Not applicable	11	11	44
HD-SDI	Not applicable	—	2	2	8
3G-SDI	Not applicable	Not applicable	Not applicable	Not applicable	4
6G-SDI	Not applicable	Not applicable	Not applicable	Not applicable	2
12G-SDI	Not applicable	Not applicable	Not applicable	Not applicable	—

5.3.7. Clock Enable Generator

The clock enable generator is a simple logic that generates a clock enable signal.

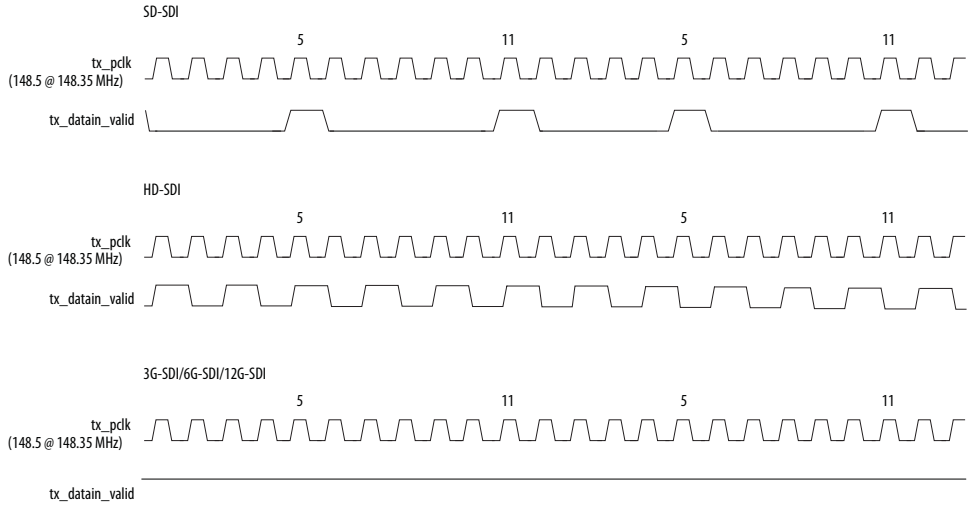
The clock enable signal serves as a data valid signal, `tx_datain_valid` for the incoming video data signal, `tx_datain`. The video data signal is based on the incoming video standard signal, `tx_std`. The transmit parallel clock, `tx_pclk`, can be a single frequency of either 148.5 MHz or 148.35 MHz.

The clock enable generator generates a clock signal in the following conditions:

- If the `tx_datain` signal is SD—generate a `tx_datain_valid` pulse every 5th and 11th clock cycle of the tx_pclk domain.
- If the `tx_datain` signal is HD—generate a `tx_datain_valid` pulse every other clock cycle of the tx_pclk domain.
- If the `tx_datain` signal is neither SD nor HD—the `tx_datain_valid` pulse remains high for 3G, 6G, or 12G.

Figure 21. Triple Rate Transmit Clocking Scheme

This figure illustrates the behavior of the `tx_datain_valid` pulse in each video standard.



5.3.8. RX Sample

This submodule extracts data from the oversampled incoming data stream. In oversampling schemes, each bit is repeated many times. For example, a stream of 0 1 0 1 may look like 000111000111 at the oversample clock or data rate.

5.3.9. Detect Video Standard

The detect video standard submodule performs coarse rate detection on the incoming video stream for dual-, triple-, or multi-rate SDI.

This scheme is required for the SDI II Intel FPGA IP core to reprogram the transceivers to the correct settings for the video standard present at the input.

Related Information

Transceiver Controller on page 33

5.3.10. Detect 1 and 1/1.001 Rates

This submodule indicates if the incoming video stream is running at PAL (1) or NTSC (1/1.001) rate. The output port signal, `rx_clkout_is_ntsc_paln` is set to 0 if the submodule detects the incoming stream as PAL (148.5 MHz or 74.25 MHz recovered clock) and set to 1 if the incoming stream is detected as NTSC (148.35 MHz or 74.175 MHz recovered clock).

For correct video rate detection, you must set the top level port signal, `rx_coreclk_is_ntsc_paln`, to the following bit:

- 0 if the `rx_coreclk` signal is 297 MHz, 148.5 MHz or the `rx_coreclk_hd` signal is 74.25 MHz
- 1 if the `rx_coreclk` signal is 296.7 MHz, 148.35 MHz or the `rx_coreclk_hd` signal is 74.175 MHz

5.3.11. Transceiver Controller

The transceiver controller controls the transceiver and performs dynamic reconfiguration (if necessary) to achieve the desired receiver functionality for the SDI.

When the interface receives SD-SDI, the receiver transceiver sets to lock-to-refclk (LTR) mode and when the interface receives HD-SDI or higher SDI data rate, the receiver transceiver sets to lock-to-data (LTD) mode.

In dual-rate, triple-rate, or multi-rate mode, the IP core first sets to the highest data-rate mode (transceiver running at 2.97 Gbps for dual/triple rate and 11.88 Gbps for multi rate) in LTR mode.

The detect video standard submodule starts running for a period of time. The output of this submodule determines if the transceiver requires dynamic reconfiguration to a new mode. The dual-rate and triple-rate modes use 11× oversampling to receive SD-SDI. This means that you require only two transceiver setups because the rates for 3G-SDI and 11× SD-SDI are the same. For multi-rate (up to 12G) modes, you require two more setups to accommodate 6G-SDI and 12G-SDI.

Related Information

Detect Video Standard on page 32

5.3.12. Descrambler

This submodule implements data descrambling as defined in the *SMPTE ST 259* and *SMPTE ST 292* specifications. This submodule is similar to the scrambler submodule, where it implements the reverse of the scrambling applied to the data. This submodule uses an LFSR and also implements NRZI.

5.3.13. TRS Aligner

The TRS aligner word aligns the descrambled receiver data until the bit order of the output data and the original video data are the same. The EAV and SAV sequences determine the correct word alignment.

Table 13. EAV and SAV Sequences

This table lists the sequence pattern for each video standard.

Video Standard	EAV and SAV Sequences
SD-SDI	3FF 000 000
HD-SDI	3FF 3FF 000 000 000 000
3G-SDI Level A	3FF 3FF 000 000 000 000
3G-SDI Level B	3FF 3FF 3FF 3FF 000 000 000 000 000 000 000 000
6G-SDI with 4 Streams Interleaved	3FF 3FF 3FF 3FF 000 000 000 000 000 000 000 000
6G-SDI with 8 Streams Interleaved	3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
12G-SDI with 8 Streams Interleaved	3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000
12G-SDI with 16 Streams Interleaved	3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 3FF 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000 000

The TRS aligner determines the correct word alignment for the data. The aligner looks for three consecutive TRSs with the same alignment and then stores that alignment. If the aligner subsequently detects two consecutive TRSs with a different alignment, then it stores this new alignment.

5.3.14. 3Gb Demux

The 3Gb Demux submodule demultiplexes the Y link A, C link A, Y link B, and C link B from the received 20-bit data for further processing. This submodule is mainly for 3G-SDI (level B) operation and it is required in 3G-SDI and triple rate SDI modes.

Related Information

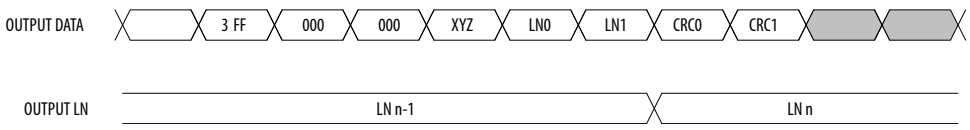
3G-SDI (Level B) to HD-SDI Dual Link Conversion on page 38

5.3.15. Extract Line

The HD-SDI and higher standards include the current video line number as part of the EAV extension words. The insert line submodule encodes the 11-bit line number in two of these extension words as defined in the *SMPTE ST 292* specification.

This submodule decodes the data words and registers them when the Match TRS submodule indicates that the current words are LN0 and LN1 extension words.

Figure 22. Line Number Extraction



5.3.16. Extract Payload ID

This submodule detects one 10-bit Y data stream from an interface and extracts the payload ID packet present in that data stream.

This submodule produces a valid signal, which indicates that a valid payload ID packet data is present on the submodule's payload output port. The submodule updates this payload each time it detects an error-free *SMPTE ST 352* packet. The submodule discards erroneous packets like checksum error and the payload port retains the information from the last good packet. The valid output signal goes high immediately upon receiving a good packet. If the submodule detects erroneous packets or the packets are no longer present, the valid output signal remains high for a number of frames or fields after the last good packet is received.

This submodule provides all four bytes of the payload ID data on its payload output port.

5.3.17. Detect Format

The detect format submodule monitors the line and frame timing of an incoming SDI stream. It generates various flags to indicate whether the receive stream is locked, and reports matching known video formats as `rx_format`.

A word counter monitors the EAV and SAV positions in the incoming video. The word counter increments on each valid word and stores the count value when an EAV or SAV is seen. If the count values are the same as a predefined value, the core determines the incoming video to be TRS locked. The predefined value is set to 6, therefore after six consecutive lines of the same EAV and SAV timing, the `rx_trs_locked` signal is active.

A line counter increments at the start of each video line. When the core finds the first active line of a field or frame, the line counter starts incrementing until the last active line of the same field or frame.

To determine the video format, a comparison logic compares the word and line count values in the video stream against the known values predefined for various video formats. The logic searches sequentially from one known value to another.

- If the logic finds a match, the core is determined to be frame locked and the `rx_frame_locked` signal is active. The core reports the matched known value as `rx_format`.
- If the logic does not find any match and the count is consistent over two video frames, the `rx_frame_locked` signal remains active but the `rx_format` stays asserted.

5.3.18. Sync Streams

This submodule is required in the HD-SDI dual link receiver as it synchronizes and deskews both data streams received by two separate transceivers of link A and link B. When the TRS word on both streams are aligned to each other, the core is considered locked and the `rx_dl_locked` signal asserts.

5.3.19. Convert SD Bits

This submodule is enabled when you set the **SD Interface Bit Width** parameter option to 20. This submodule converts the SD parallel data in 20 bits back to 10 bits as per the requirement for further processing.

This submodule contains a clock enable generator to generate two data valid pulses at every 11th clock cycle of the `tx_pclk` domain. Each time the data valid signal is asserted, this block alternately transmits the lower 10 bits and upper 10 bits of the SD 20-bit interface data to the downstream logic.

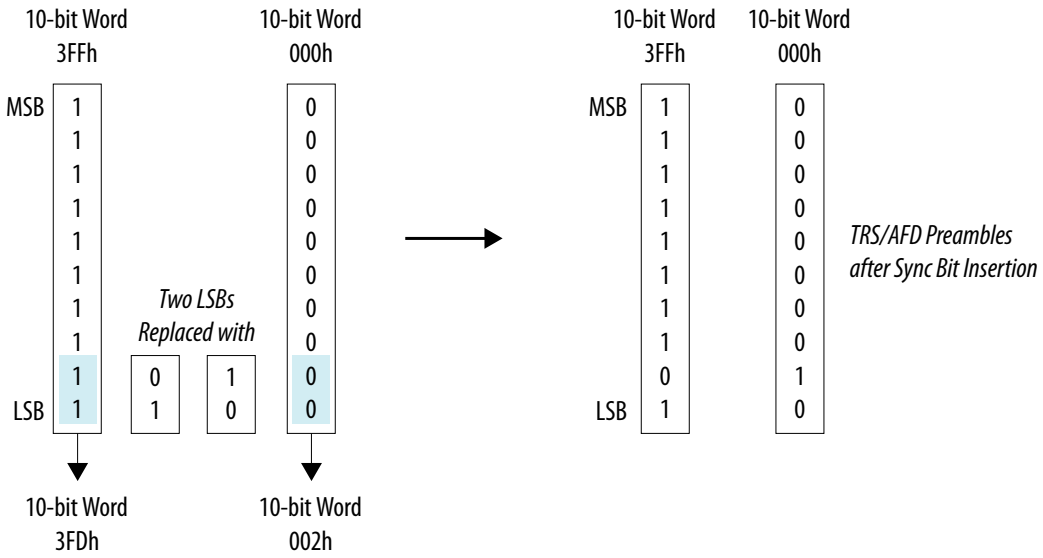
5.3.20. Insert Sync Bits

Inserting sync bits prevents long runs of 0s.

Repeating patterns of 3FF or 000h for 6G-SDI and 12G-SDI video standards in the 10-bit parallel interface may result in a long run of zeros feeding the scrambling polynomial. A long run of zeros goes up to a length of 160 "1"s and 339 "0"s, which may cause the generation of the *pothole pathological* condition.

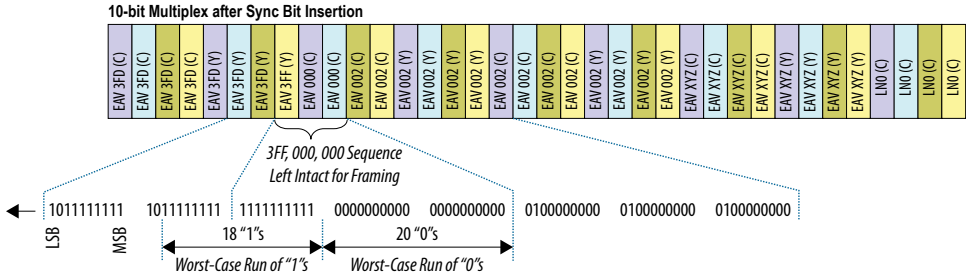
To prevent long runs, this feature modifies the 10-bit parallel interface data stream. It replaces the two LSBs of repeated 3FF or 000 code words with sync-bit values of 10b for 000h words and 01b for 3FFh words.

Figure 23. Sync Bits



However, to ensure the words are synchronized and aligned in the receiver, this feature retains one complete sequence of preambles (3FFh 000h 000h) without modification.

Figure 24. Sync Bits Insertion Process



5.3.21. Remove Sync Bits

The sync bit inserted in 6G-SDI or 12G-SDI data from the source must be removed to allow other receiver submodules to function correctly.

This submodule detects the sync bit presented in the data stream and restores back the correct words, for example TRS words.

5.4. Optional Features

The SDI II Intel FPGA IP core also provides some optional features.

5.4.1. HD-SDI Dual Link to 3G-SDI (Level B) Conversion

To interface between a HD-SDI dual link receiver and 3G-SDI single link transmitter equipment, perform a HD-SDI dual link to 3G-SDI (level B) conversion. Level B is defined as 2× *SMPTE ST 292* HD-SDI mapping, including *SMPTE ST 372* dual link mapping.

Note: This feature is only available for Arria V, Cyclone V, and Stratix V devices. You can enable this feature through the SDI II Intel FPGA parameter editor.

This conversion takes either two 1.485 Gbps dual link signals or two separate co-timed HD signals and combines them into a single 3G-SDI stream.

Figure 25. Example of HD-SDI Dual Link to 3G-SDI (Level B) Conversion

The figure shows the conversion of two HD-SDI data streams to 3G-SDI (level B) data streams.

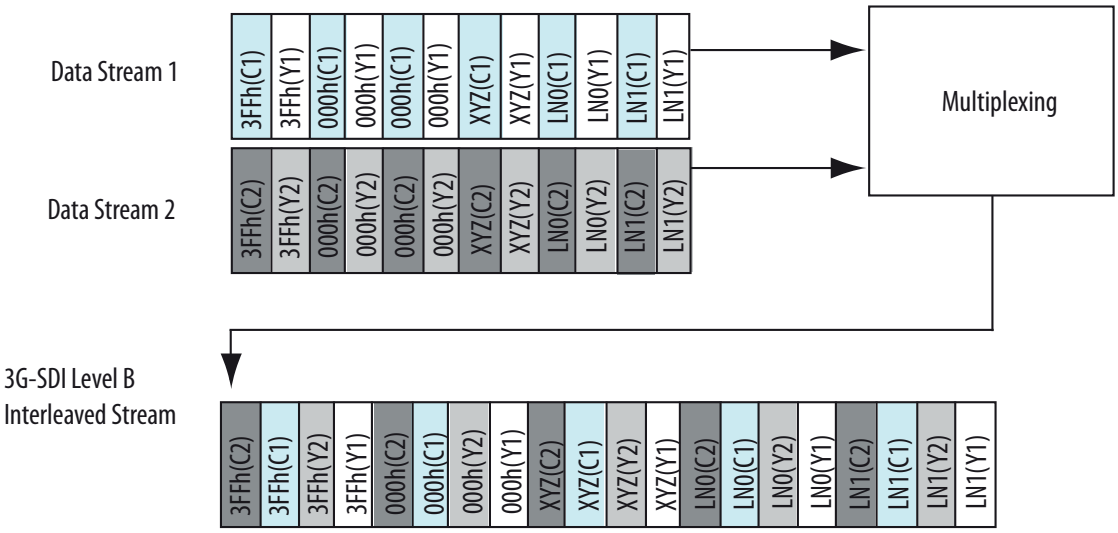
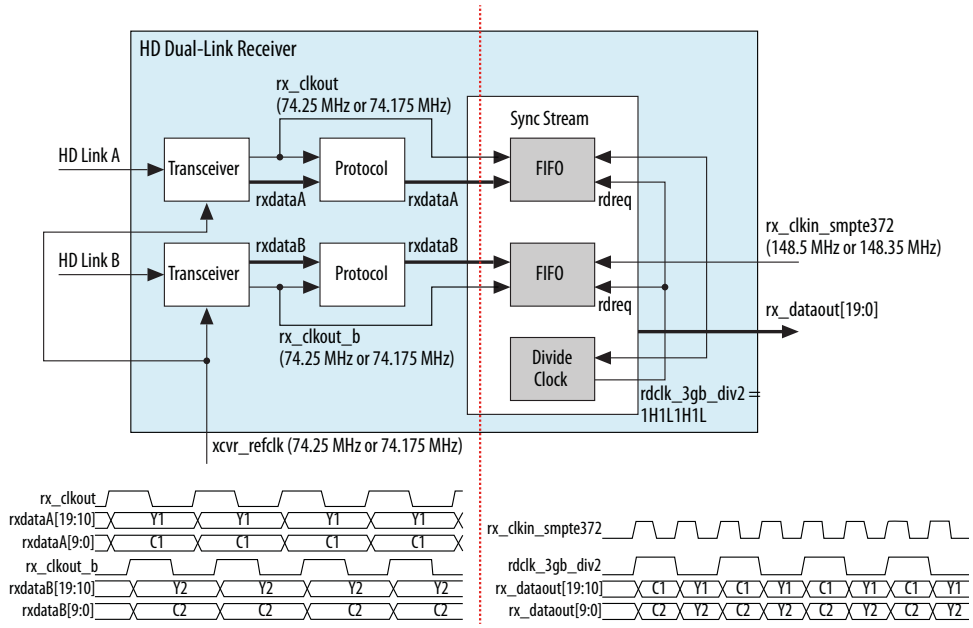


Figure 26. Implementation of HD-SDI Dual Link to 3G-SDI (Level B) Conversion

The figure shows a block diagram of HD-SDI dual link to 3G-SDI (level B) conversion.



5.4.2. 3G-SDI (Level B) to HD-SDI Dual Link Conversion

To interface between 3-Gbps single link receiver and HD-SDI dual link transmitter equipment, perform a 3G-SDI (level B) to HD-SDI dual link conversion.

Note: This feature is only available for Arria V, Cyclone V, and Stratix V devices. You can enable this feature through the SDI II Intel FPGA parameter editor.

This conversion takes a single 3G-SDI signal and separates the signal into two 1.485 Gbps signals, which can either be a dual link 1080p signal or two separate co-timed HD data streams.

Figure 27. Example of 3G-SDI (Level B) to HD-SDI Dual Link Conversion

The figure shows the conversion of 3G-SDI (level B) data to two HD-SDI data streams.

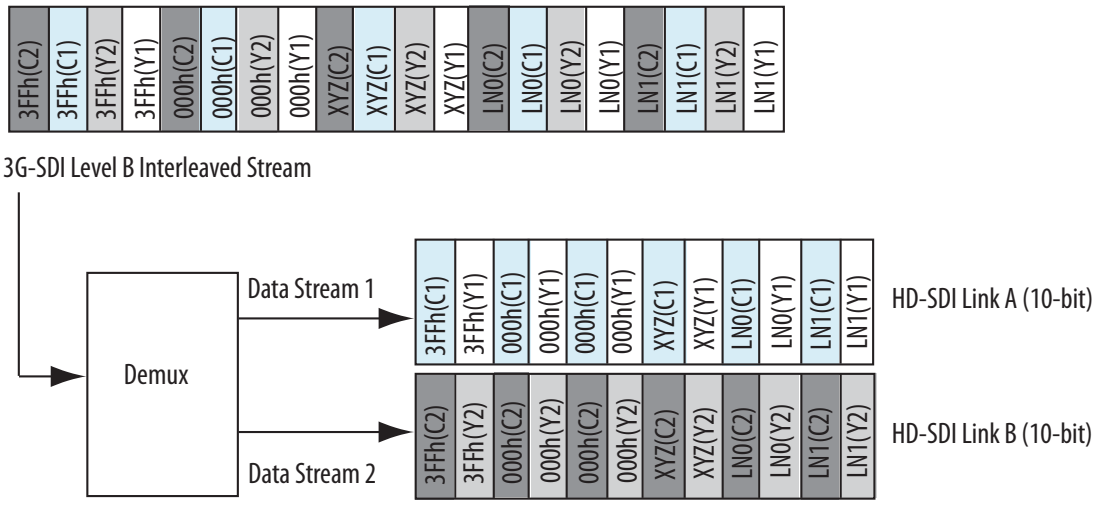
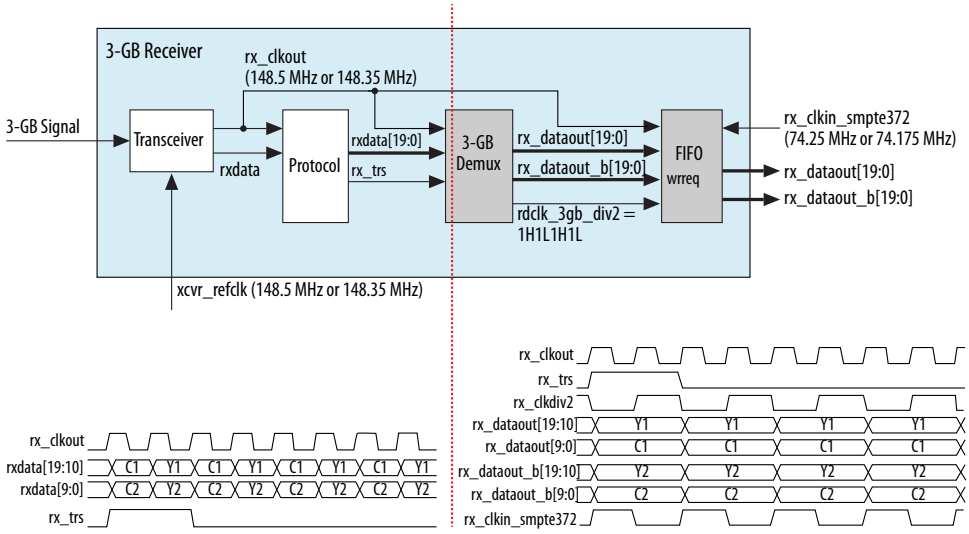


Figure 28. Implementation of 3G-SDI (Level B) to HD-SDI Dual Link Conversion

The figure shows a block diagram of 3G-SDI (level B) to HD-SDI dual link conversion.



5.4.3. SMPTE RP168 Switching Support

The SMPTE RP168 standard defines the requirements for synchronous switching between two video sources to take place with minimal interference to the receiver. The RP168 standard has restrictions for which lines the source switching can occur.

The SDI II Intel FPGA IP core has flexibility and does not restrict you to switch at only a particular line defined in the RP168 standard. You can perform switching at any time between different video sources if the source has similar standard and format. After switching, all the status output signals, including the `rx_trs_locked`, `rx_frame_locked`, and `rx_align_locked` signals, remain unchanged. You should not see any interrupts at downstream.

5.4.4. SD 20-Bit Interface for Dual/Triple Rate

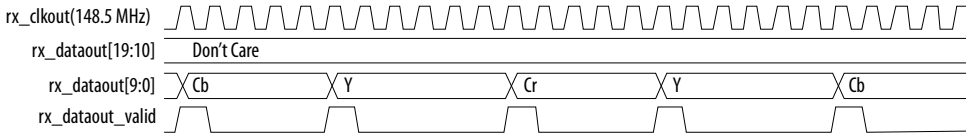
For a common SD interface, the serial data format is 10 bits wide, whereas for HD or 3G, the data format is 20 bits wide, divided into two parallel 10-bit datastreams (known as Y and C).

To make the interface bit width common for all standards in the dual-rate or triple-rate SDI mode:

- The receiver can extract the data and align them in 20-bit width
- The transmitter can accept SD data in 20-bit width and retransmit them successfully

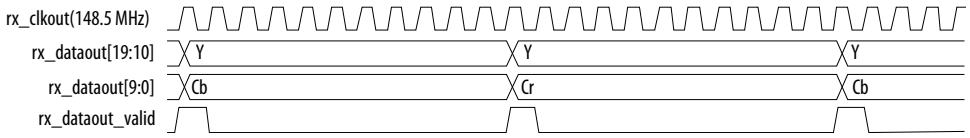
The timing diagrams below show a comparison of data arrangement between 10-bit and 20-bit interface.

Figure 29. SD 10-Bit Interface



- The upper 10 bits of `rx_dataout` are insignificant data.
- The lower 10 bits of `rx_dataout` are Luma (Y) and chroma (Cb, Cr) channels (interleaved).
- The 1H 4L 1H 5L cadence of `rx_dataout_valid` repeats indefinitely (ideal).

Figure 30. SD 20-Bit Interface



- The upper 10 bits of `rx_dataout` are Luma (Y) channel and the lower 10 bits are Chroma (Cb, Cr) channel.
- The 1H 10L cadence of `rx_dataout_valid` repeats indefinitely (ideal).

5.4.5. Dynamic TX Clock Switching for Arria V, Cyclone V, and Stratix V Devices

The dynamic TX clock switching feature allows you to dynamically switch between NTSC and PAL transceiver data rates for all video standards except SD-SDI.

Note: For information about dynamic TX clock switching for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, refer to the respective design example user guides.

The dynamic TX clock switching enables an SDI video equipment to operate on NTSC or PAL. You can choose to switch the TX clock through one of these two methods:

- Instantiate an alternate TX PLL and supply two different clocks to the two PLLs. Switch between the primary PLL and the alternate PLL for transmission.
- Use the primary PLL with two reference input clocks. The PLL switches between these two clocks for transmission.

To implement this feature, you are required to provide two reference clocks (`xcvr_refclk` and `xcvr_refclk_alt`) to the SDI II Intel FPGA IP core. The frequency of the reference clocks must be assigned to 148.5 MHz and 148.35 MHz in any assignment order.

The TX PLL select signal (`ch1_{tx/du}_tx_pll_sel`) is an input control signal that you provide to the core and the transceiver reconfiguration controller to select the desired clock input for the Native PHY IP core.

- Set `ch1_{tx/du}_tx_pll_sel` to 0 to select `xcvr_refclk`
- Set `ch1_{tx/du}_tx_pll_sel` to 1 to select `xcvr_refclk_alt`

To dynamically switch between the two reference clocks, you need to implement a simple handshaking mechanism. The handshake is initiated when the reconfiguration request signal (`ch1_{tx/du}_tx_start_reconfig`) is asserted high. This signal must remain asserted until the reconfiguration process completes. The reconfiguration process completes when the reconfiguration done signal (`ch1_{tx/du}_tx_reconfig_done`) is asserted high. The TX PLL select signal (`ch1_{tx/du}_tx_pll_sel`) needs to be stable throughout the reconfiguration process.

To complete the handshaking process, you must deassert the reconfiguration request signal (`ch1_{tx/du}_tx_start_reconfig`) upon assertion of the reconfiguration done signal (`ch1_{tx/du}_tx_reconfig_done`). The dynamic TX clock switching only takes effect after the `tx_rst` is asserted high and deasserted low accordingly.

Figure 31. Hardware Implementation of the Dynamic TX Clock Switching Feature

This figure shows the TX clock switching feature with two TX PLLs.

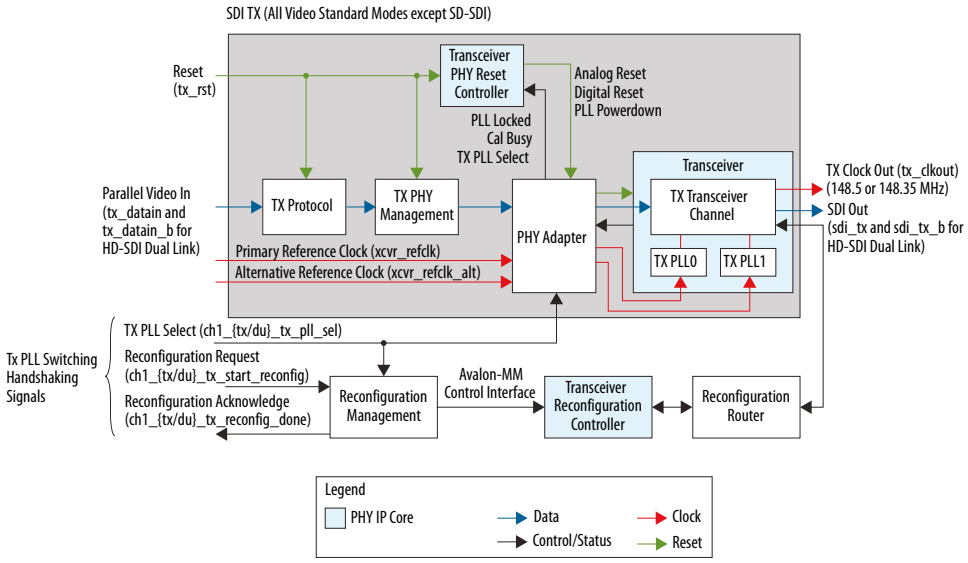
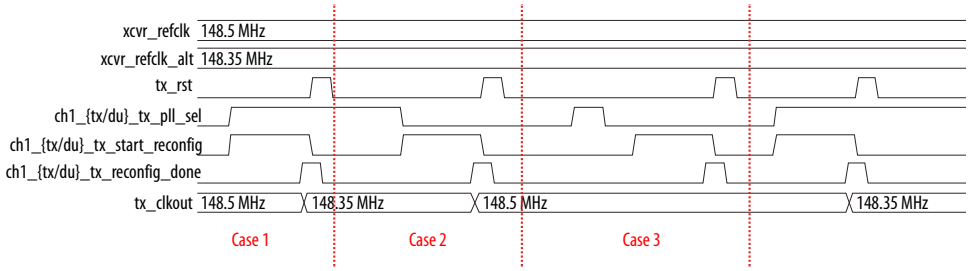


Figure 32. Dynamic TX Clock Switching Timing Diagram



The table below describes the behavior of the dynamic switching feature when you initiate a handshaking process (with reference to the timing diagram).

Table 14. Dynamic Switching Behavior During a Handshaking Process

Case	Description
1	The handshaking process attempts to switch to select `xcvr_refclk_alt`. `tx_clkout` successfully locks to `xcvr_refclk_alt` (148.35 MHz).
2	The handshaking process attempts to switch to select `xcvr_refclk`. `tx_clkout` successfully locks to `xcvr_refclk` (148.5 MHz).
3	The handshaking process attempts to switch to select `xcvr_refclk_alt`. The switching fails because `ch1_{tx/du}_tx_pll_sel` changes from 1 to 0 before the assertion of `ch1_{tx/du}_tx_start_reconfig`. Therefore, `tx_clkout` remains locked to `xcvr_refclk` (148.5MHz).

Implementing TX PLL and Reference Clock Switching

To implement the TX PLL and reference clock switching, follow these steps:

1. Trigger the `tx_pll_sel` signal to the desired reference clock: 0 for 148.5 or 1 for 148.35 MHz.
2. Assert the `tx_start_reconfig` signal at the same clock cycle. You may assert the signal at the next clock cycle as long as you do not toggle back the `tx_pll_sel` signal.
3. Keep the `tx_start_reconfig` signal asserted until the `tx_reconfig_done` signal asserts.
4. Deassert the `tx_start_reconfig` signal and assert the `tx_rst` signal at the next cycle.
5. The TX clock (`tx_clk`) should run at the new frequency now.

6. SDI II Intel FPGA IP Core Signals

The following tables list the SDI II Intel FPGA IP core signals by components.

- Protocol blocks—transmitter, receiver
- Transceiver blocks—PHY management, PHY adapter, Native PHY IP

Note: These signals are applicable for all supported Intel FPGA devices unless specified otherwise.

6.1. SDI II Intel FPGA IP Core Resets and Clocks

Table 15. Resets and Clock Signals

Signal	Width	Direction	Description
tx_rst	1	Input	Reset signal for the transmitter. This signal is active high and level sensitive. This signal must be synchronous to tx_pclk clock domain (for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices) or tx_coreclk (for Arria V, Cyclone V, and Stratix V devices).
pll_powerdown_in	1*N*	Input	When asserted, this signal resets TX PLL. You must connect this signal to pll_powerdown_out. You can connect this signal from multiple SDI instances to pll_powerdown_out of one of the SDI instances to merge the PLL in these instances. For TX PLL merging, pll_powerdown_in and xcvr_refclk from multiple instances must share the same source. *N* = Number of PLLs in the core—1 (default) or 2 (when TX PLL switching enabled) *Note:* Not applicable for these settings: • In protocol only mode. • For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices. If you enabled the **Dynamic Tx clock switching** parameter, your design requires XCVR_TX_PLL_RECONFIG_GROUP QSF assignment. Refer to the *Transceiver PHY IP Core User Guide* for more information.
pll_powerdown_out	1*N*	Output	When asserted, this signal resets the selected TX PLL. *N* = Number of PLLs in the core—1 (default) or 2 (when TX PLL switching enabled) *Note:* Not applicable for these settings: • In protocol only mode. • For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_rst	1	Input	Reset signal for the receiver. This signal is active high and level sensitive. This reset signal must be synchronous to the rx_coreclk or rx_coreclk_hd clock domain.

continued...

Signal	Width	Direction	Description
rx_rst_proto_in	1	Input	Receiver protocol reset signal. This signal must be driven by the rx_rst_proto_out reset signal from the transceiver block. *Note:* Applicable for receiver protocol configuration only (Arria V, Cyclone V, and Stratix V devices).
rx_rst_proto_in_b	1	Input	Receiver protocol reset signal for link B. This signal must be driven by the rx_rst_proto_out_b reset signal from the transceiver block. *Note:* For HD-SDI dual link receiver protocol configuration only.
rx_rst_proto_out	1	Output	Reset the receiver protocol downstream logic. This generated signal is synchronous to rx_clkout clock domain and must be used to drive the rx_rst_proto_in signal of the receiver protocol block.
rx_rst_proto_out_b	1	Output	Reset the receiver protocol downstream logic. *Note:* For HD-SDI dual link receiver transceiver configuration only.
trig_rst_ctrl	1	Output	Reset output signal to the transceiver reset controller to reset the transceiver. This signal is synchronous to the rx_coreclk or rx_coreclk_hd clock domain. *Note:* Applicable only for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
tx_pclk	1	Input	Transmitter core parallel clock signal. This clock signal must be driven by the by parallel output clock from TX transceiver. • SD-SDI = 148.5 MHz • HD-SDI = 74.25 MHz or 74.175 MHz, depending on video frame rate • 3G-SDI = 148.5 MHz or 148.35 MHz, depending on video frame rate • HD-SDI Dual Link = 74.25 MHz or 74.175 MHz, depending on video frame rate • Dual Rate = 148.5 MHz or 148.35 MHz, depending on video frame rate • Triple Rate = 148.5 MHz or 148.35 MHz, depending on video frame rate • Multi Rate (up to 12G-SDI) = 148.5 MHz or 148.35 MHz, depending on video frame rate
tx_coreclk	1	Input	148.5-MHz or 148.35-MHz transmitter core clock signal. This clock source must be always stable and can be shared with xcvr_refclk. *Note:* Not applicable for these settings: • In protocol only mode. • If the selected transceiver reference clock frequency is 74.25 MHz/74.175 MHz. • For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
tx_coreclk_hd	1	Input	74.25-MHz or 74.175-MHz transmitter core clock signal. This clock source must be always stable and can be shared with xcvr_refclk. *Note:* Applicable for HD-SDI and HD-SDI dual link modes only if the selected transceiver reference clock frequency is 74.25 MHz/ 74.175 MHz. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_coreclk	1	Input	Receiver core clock signal. You can set the following frequencies: • 148.5-MHz or 148.35-MHz: Applicable for all configurations. • 297.0 MHz or 296.70 MHz: Applicable only for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices in multi-rate modes. This clock source must be stable and there are no required relationships with any other clocks. The clock source can be asynchronous or synchronous to any transceiver's clock.

continued...

Signal	Width	Direction	Description
			Note: Not applicable if the selected transceiver reference clock frequency is 74.25 MHz/74.175 MHz.
rx_coreclk_hd	1	Input	74.25-MHz or 74.175-MHz receiver core clock signal. This clock source must be always stable and can be shared with xcvr_refclk. This clock source must be stable and there are no required relationships with any other clocks. The clock source can be asynchronous or synchronous to any transceiver's clock. *Note:* Applicable for HD-SDI and HD-SDI dual link modes only if the selected transceiver reference clock frequency is 74.25 MHz/ 74.175 MHz. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_clkin	1	Input	Receiver protocol clock input. This signal must be driven by the rx_clkout clock signal from the transceiver block. • SD-SDI = 148.5 MHz • HD-SDI = 74.25 MHz or 74.175 MHz, depending on video frame rate • 3G-SDI = 148.5 MHz or 148.35 MHz, depending on video frame rate *Note:* For receiver protocol configuration only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_clkin_b	1	Input	Receiver protocol clock input for link B. This signal must be driven by the rx_clkout_b clock signal from the transceiver block ((74.25 MHz or 74.125 MHz, depending on video frame rate). *Note:* For HD-SDI dual link receiver protocol configuration only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_clkin_smpte372	1	Input	Clock input for HD-SDI dual link to 3G-SDI (level B) and 3G-SDI (level B) to HD-SDI dual link operations. • HD-SDI dual link to 3G-SDI (level B) = 148.5 MHz or 148.35 MHz • 3G-SDI (level B) to HD-SDI dual link = 74.25 MHz or 74.175 MHz
xcvr_rxclk	1	Input	Receiver parallel clock input. Driven by rx_pma_div_clkout (for multi-rate modes) or rx_clkout (for other modes) from the transceiver. • SD-SDI = 148.5 MHz • HD-SDI = 74.25 MHz or 74.175 MHz, depending on video frame rate • 3G-SDI = 148.5 MHz or 148.35 MHz, depending on video frame rate • 6G-SDI = 148.5 MHz or 148.35 MHz, depending on video frame rate • 12G-SDI = 148.5 MHz or 148.35 MHz, depending on video frame rate *Note:* Applicable only for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
xcvr_refclk	1	Input	Reference clock signal for the transceiver. Only a single reference clock frequency is required to support both integer and fractional frame rates for RX CDR. The clock source must be stable. It must be a free running clock connected to the transceiver clock pin. • SD-SDI = 148.5 MHz • HD-SDI = 74.25 MHz, 74.175 MHz, 148.5 MHz, or 148.35 MHz • 3G-SDI = 148.5 MHz or 148.35 MHz • HD-SDI Dual Link: 74.25 MHz, 74.175 MHz, 148.5 MHz, or 148.35 MHz • Dual Rate: 148.5 MHz or 148.35 MHz • Triple Rate: 148.5 MHz or 148.35 MHz

continued...

Signal	Width	Direction	Description
			Note: Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
xcvr_refclk_alt	1	Input	Alternative clock input for the Native PHY IP core. The frequency of this signal must be the alternate frequency value of the xcvr_refclk signal. • HD-SDI = 74.25 MHz, 74.175 MHz, 148.5 MHz, or 148.35 MHz • 3G-SDI = 148.5 MHz or 148.35 MHz • HD-SDI Dual Link: 74.25 MHz, 74.175 MHz, 148.5 MHz, or 148.35 MHz • Dual Rate: 148.5 MHz or 148.35 MHz • Triple Rate: 148.5 MHz or 148.35 MHz *Note:* Applicable only when you turn on the **Tx PLL Dynamic Switching** option. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
tx_clkout	1	Output	TX transceiver parallel output clock. This frequency for this clock should be the same as the user-provided xcvr_refclk. *Note:* Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_clkout	1	Output	RX transceiver parallel output clock. • SD-SDI = 148.5 MHz • HD-SDI = 74.25 MHz or 74.175 MHz, depending on video frame rate • 3G-SDI = 148.5 MHz or 148.35 MHz, depending on video frame rate *Note:* Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_clkout_b	1	Output	RX transceiver parallel output clock for link B. The output clock frequency must be 74.25 or 74.175 MHz, depending on video frame rate. *Note:* For HD-SDI dual link only.

Figure 33. TX Clocking Diagram for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 Devices

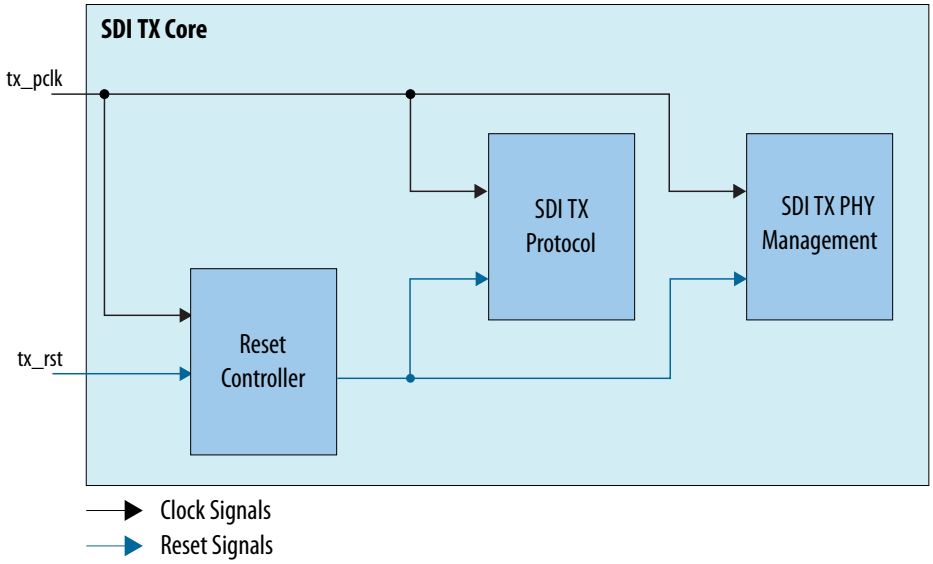
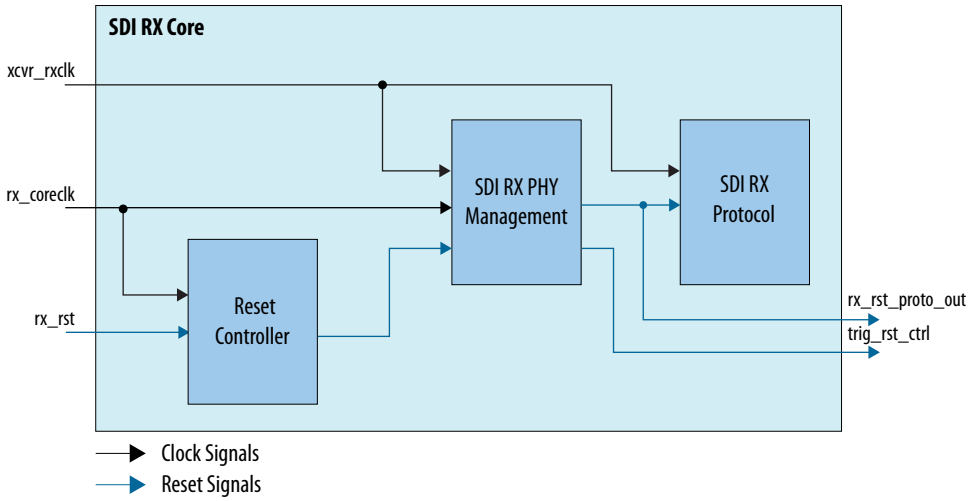


Figure 34. RX Clocking Diagram for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 Devices



Note: For a more comprehensive TX and RX Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 clocking diagrams with transceivers, refer to the respective design example user guides.

Figure 35. TX Clocking Diagram for Arria V, Cyclone V, and Stratix V Devices

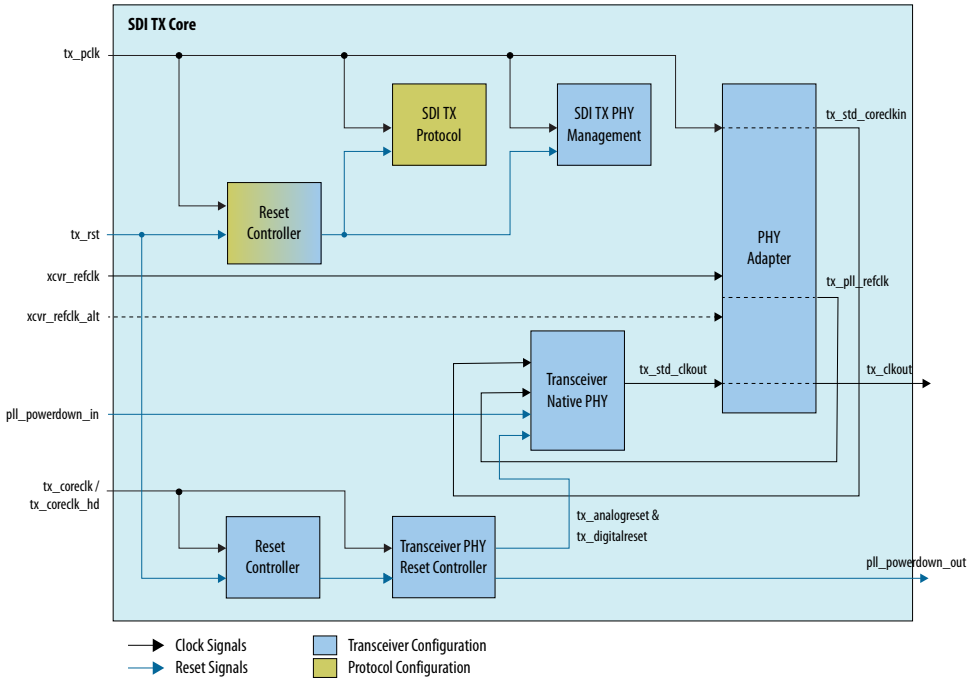
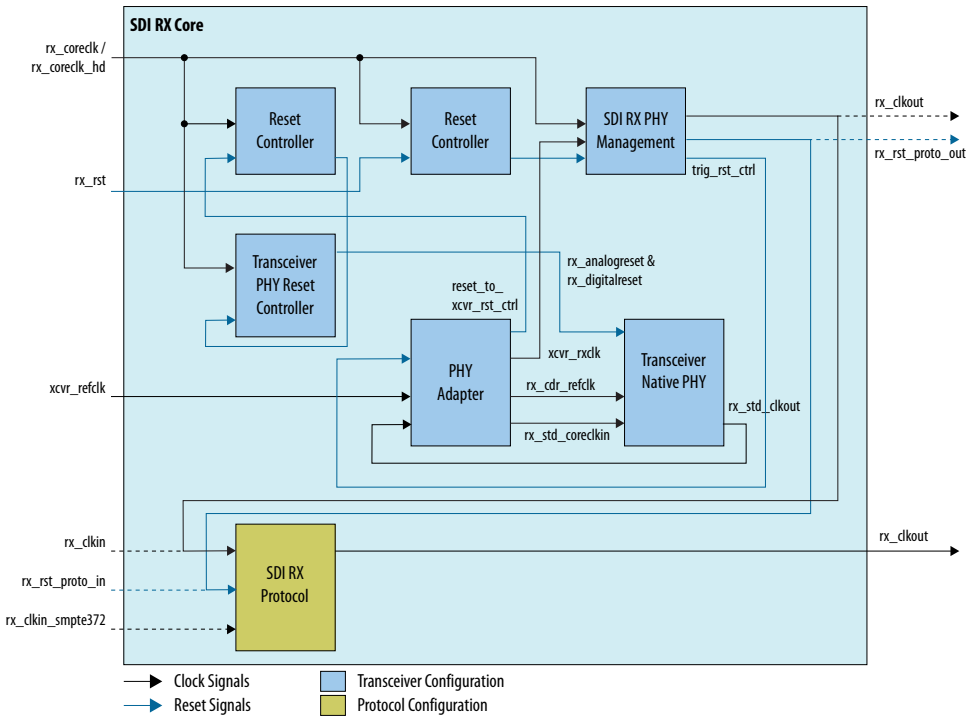


Figure 36. RX Clocking Diagram for Arria V, Cyclone V, and Stratix V Devices



Note: For the Arria V, Cyclone V, and Stratix V devices, the source or destination for dual link signals with "_b suffix are the same as the original signals. For example, the destination for `rx_clkin_b` is the same as `rx_clkin`. which is directed to the SDI RX protocol block. Similarly, the source of the output signal `rx_rst_proto_out_b` is the same as `rx_rst_proto_out`, which comes from the SDI RX PHY management block.

6.2. Transmitter Protocol Signals

Table 16. Transmitter Protocol Signals—Synchronous to `tx_pclk`

Note: *S* = Indicates the number of 20-bit interfaces; 4 for multi-rate (up to 12G) mode and 1 for other modes.

Signal	Width	Direction	Description
`tx_enable_crc`	1	Input	Enables CRC insertion for all modes except SD-SDI. *Note:* Not applicable for transceiver only configurations.
`tx_enable_ln`	1	Input	Enables LN insertion for all modes except SD-SDI. *Note:* Not applicable for transceiver only configurations.
`tx_std`	3	Input	Transmitter video standard. • SD-SDI = 000 • HD-SDI = 001 • 3G-SDI Level A = 011 • 3G-SDI Level B = 010

continued...

Signal	Width	Direction	Description
			• 6G-SDI 4 Streams Interleaved = 101 • 6G-SDI 8 Streams Interleaved = 100 • 12G-SDI 8 Streams Interleaved = 111 • 12G-SDI16 Streams Interleaved= 110 *Note:* Applicable for 3G-SDI, and dual-rate, triple-rate, and multi-rate modes.
tx_datain	20*S*	Input	User-supplied transmitter parallel data. • SD-SDI = bits 19:10 unused; bits 9:0 C, Y multiplex • HD-SDI = bits 19:10 Y; bits 9:0 C • HD-SDI dual link = bits 19:10 Y link A, bits 9:0 C link A • 3G-SDI Level A = bits 19:10 Y; bits 9:0 C • 3G-SDI Level B = bits 19:10 C, Y multiplex (link A); bits 9:0 C, Y multiplex (link B) • 6G-SDI: bits 79:40 unused; bits 39:30 data stream 1; bits 29:20 data stream 2; bit 19:10 data stream 3; bits 9:0 data stream 4. • 12G-SDI: bits 79:70 data stream 1; bits 69:60 data stream 2; bit 59:50 data stream 3; bits 49:40 data stream 4; bits 39:30 data stream 5; bits 29:20 stream 6; bits 19:10 stream 7; bits 9:0 data stream 8 Refer to Image Mapping on page 51 for more information about the 6G-SDI and 12G-SDI image mapping. For transceiver only configurations, the transmitter does not scramble these data before sending to the Native PHY IP core.
tx_datain_b	20	Input	User-supplied transmitter parallel data for link B. HD-SDI dual link = bits 19:10 Y link B, bits 9:0 C link B For transceiver only configurations, the transmitter does not scramble these data before sending to the Native PHY IP core. *Note:* For HD-SDI dual link mode only.
tx_datain_valid	1	Input	Transmitter parallel data valid. The timing (H: High, L: Low) must be synchronous to tx_pclk clock domain and has the following settings: • SD-SDI = 1H 4L 1H 5L • HD-SDI = H • 3G-SDI = H • HD-SDI Dual Link = H • Dual rate = SD (1H 4L 1H 5L); HD (1H 1L) • Triple rate = SD (1H 4L 1H 5L); HD (1H 1L); 3G (H) • Multi rate (up to 12G) = SD (1H 4L 1H 5L); HD (1H 1L); 3G/6G/12G (H) This signal can be driven by user logic or by the tx_dataout_valid signal for SD-SDI, and dual-rate, triple-rate, and multi-rate modes.
tx_datain_valid_b	1	Input	Transmitter parallel data valid for link B. Applicable for HD-SDI dual link mode only. HD-SDI dual link = H This signal can be driven by user logic or by the tx_dataout_valid_b signal.
tx_trs	1	Input	Transmitter TRS input. Assert this signal on the first word of both EAV and SAV TRSs. • For 3G level B, 6G 8 streams interleaved, and 12G 16 streams interleaved, first word means two tx_pclk cycles. • For the other modes, first word means one tx_pclk cycle. *Note:* Not applicable for transceiver configurations.
tx_trs_b	1	Input	Transmitter TRS input for link B. *Note:* For HD-SDI dual link combined or protocol only configurations.

continued...

Signal	Width	Direction	Description
tx_ln	11*S*	Input	Transmitter line number. For Payload ID insertion, drive this signal with valid values. Not applicable when you disable the **Insert Video Payload ID (SMPTE ST 352)** option in SD-SDI.
tx_ln_b	11*S*	Input	Transmitter line number for link B. For Payload ID insertion, drive this signal with valid values. For use in 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) line number insertion.
tx_dataout	20*S*	Output	Transmitter parallel data out. • Arria V, Cyclone V, and Stratix V devices: Available for transmitter protocol configuration only. • Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices: Available whenever TX core is included.
tx_dataout_b	20	Output	Transmitter parallel data out for link B. *Note:* Applicable for HD-SDI dual link transmitter protocol configuration only.
tx_dataout_valid	1	Output	Data valid generated by the core. This signal can be used to drive tx_datain_valid. The timing (H: High, L: Low) must be synchronous to tx_pclk clock domain and have the following settings: • SD-SDI = 1H 4L 1H 5L • HD-SDI = H • 3G-SDI = H • HD-SDI Dual Link = H • Dual rate = SD (1H 4L 1H 5L); HD (1H 1L) • Triple rate = SD (1H 4L 1H 5L); HD (1H 1L); 3G (H) • Multi rate (up to 12G) = SD (1H 4L 1H 5L); HD (1H 1L); 3G/6G/12G (H)
tx_dataout_valid_b	1	Output	Data valid generated by the core for link B. The timing (H: High, L: Low) is identical to the tx_dataout_valid signal and is synchronous to tx_pclk clock domain. *Note:* Applicable for HD-SDI dual link mode only.
tx_std_out	3	Output	Indicates the transmitted video standard. This signal connects to tx_std in the transceiver only configuration. *Note:* Applicable for 3G-SDI, dual-rate, and triple-rate transmitter protocol only configuration. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
tx_vpid_overwrite	1	Input	When a payload ID is embedded in the video stream, the core enables this signal to overwrite the existing payload ID. No effect when disabled. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte1	8*S*	Input	The core inserts payload ID byte 1. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte2	8*S*	Input	The core inserts payload ID byte 2. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte3	8*S*	Input	The core inserts payload ID byte 3. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte4	8*S*	Input	The core inserts payload ID byte 4.

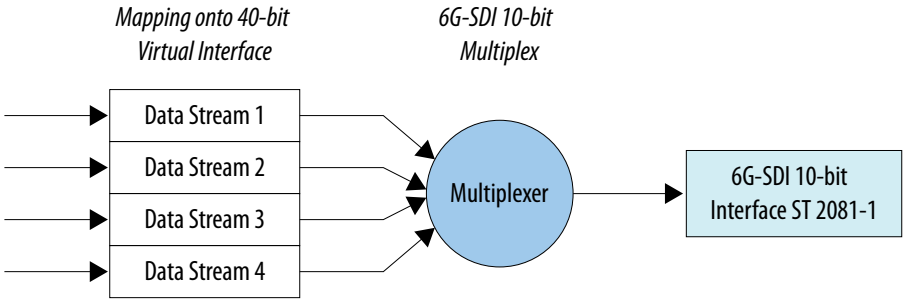
continued...

Signal	Width	Direction	Description
			Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte1_b	8*S*	Input	The core inserts payload ID byte 1 for link B. For 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte2_b	8*S*	Input	The core inserts payload ID byte 2 for link B. For 3G-SDI, HD-SDI dual link triple-rate, and multi-rate (up to 12G) modes only. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte3_b	8*S*	Input	The core inserts payload ID byte 3 for link B. For 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_vpid_byte4_b	8*S*	Input	The core inserts payload ID byte 4 for link B. For 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_line_f0	11*S*	Input	Line number of field 0 (F0) of inserted payload ID. The line number must be valid and cannot be set to 0. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.
tx_line_f1	11*S*	Input	Line number of field 1 (F1) of inserted payload ID. The line number must be valid and cannot be set to 0. Applicable only when you enable the **Insert Payload ID (SMPTE ST 352)** option.

6.2.1. Image Mapping

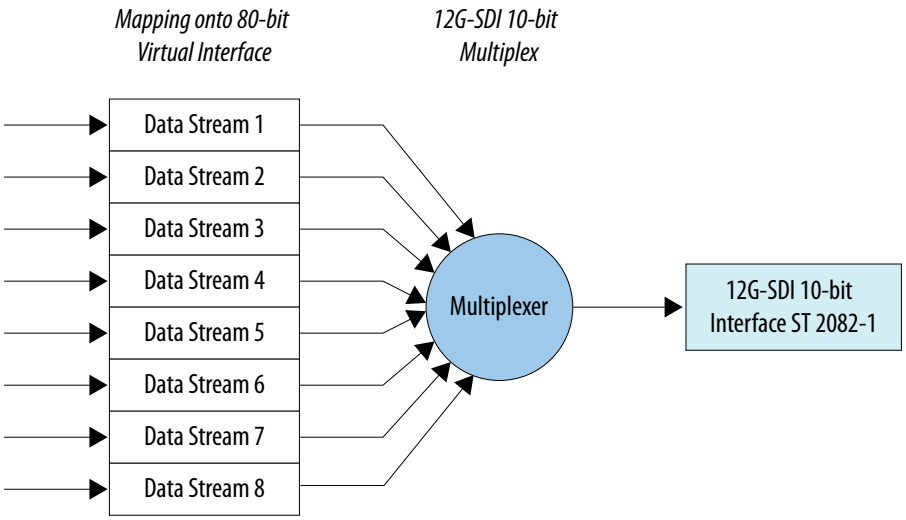
Image mapping differs for 6G-SDI and 12-SDI interfaces.

Figure 37. Transmitting 6G-SDI Image



For instance, if you are transmitting image per ST 2081-10 Mode 1 mapping, each data stream should be C, Y multiplex of each sub image.

Figure 38. Transmitting 12G-SDI Image



For instance, if you are transmitting image per ST 2082-10 mode 1 mapping, each odd data stream should be Y samples of each sub image, while the even data stream should be C samples of each sub image.

6.3. Receiver Protocol Signals

Table 17. Receiver Protocol Signals—Synchronous to `rx_coreclk`

Signal	Width	Direction	Description
`rx_coreclk_is_ntsc_paln`	1	Input	Indicates to the receiver core if `rx_coreclk` or `rx_coreclk_hd` is at NSTC (1/1.001) or PAL (1) rate. This signal is required for the receiver core to detect the incoming video rate as NTSC or PAL. • 0 = PAL rate (when `rx_coreclk` = 297 / 148.5 MHz or `rx_coreclk_hd` = 74.25 MHz) • 1 = NTSC rate (when `rx_coreclk` = 296.70 / 148.35 MHz or `rx_coreclk_hd` = 74.175 MHz) *Note:* Not applicable for SD-SDI and protocol only configurations.
`rx_std_in`	3	Input	Indicates to the receiver core protocol block the video standard received by the transceiver block. *Note:* Applicable for 3G-SDI, dual-rate, and triple-rate receiver protocol only configurations. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
`rx_clkout_is_ntsc_paln`	1	Output	Indicates that the receiver core is receiving video rate at NSTC (1/1.001) or PAL (1). • 0 = PAL rate (`rx_clkout` = 148.5 MHz or 74.25 MHz) • 1 = NTSC rate (when `rx_clkout` = 148.35 MHz or 74.175 MHz) *Note:* Not applicable for SD-SDI and protocol only modes.
`rx_std` (for transceiver only configurations)	3	Output	Receiver video standard. • 3'b000: SD-SDI • 3'b001: HD-SDI • 3'b011: 3G-SDI *Note:* Applicable for 3G-SDI, dual-rate, and triple-rate configurations only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

Table 18. Receiver Protocol Signals—Synchronous to `rx_clkout` or `xcvr_rxclk`

Note: *S* = Indicates the number of 20-bit interfaces; 4 for multi-rate (up to 12G) mode and 1 for other modes.

Signal	Width	Direction	Description
`rx_datain`	20*S*	Input	Receiver parallel data from the transceiver. For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, this signal is directly connected to the `rx_parallel_data` signal from the transceiver. *Note:* If you are not enabling the simplified data interface, refer to the Transceiver parameter editor or the *Transceiver PHY IP Core User Guide* for proper data bit mapping. For older supported devices, this signal is directly connected to the `rx_dataout` signal from the SDI receiver in transceiver mode. *Note:* Available only in protocol mode.
`rx_datain_b`	20	Input	Receiver parallel data from the transceiver for link B. This signal is directly connected to the `rx_dataout_b` signal from the SDI receiver in transceiver mode. *Note:* Applicable for HD-SDI dual link protocol only configuration. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
`rx_datain_valid`	1	Input	Data valid from the oversampling logic. Assertion of this signal indicates the current data on `rx_datain` is valid. The timing (H: High, L: Low) for each video standard has the following settings: • SD-SDI = 1H 4L 1H 5L • HD-SDI = H • 3G-SDI = H • HD-SDI Dual Link = H • Dual rate = SD (1H 4L 1H 5L); HD (H) • Triple rate = SD (1H 4L 1H 5L); HD (H); 3G (H) • Multi rate (up to 12G) = SD (1H 4L 1H 5L); HD (H); 3G/6G/12G (H) This signal is directly connected to the `rx_dataout_valid` signal from the SDI receiver in transceiver mode. *Note:* Applicable for protocol only configuration. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
`rx_datain_valid_b`	1	Input	Data valid from the oversampling logic. Assertion of this signal indicates the current data on `rx_datain_b` is valid. This signal is directly connected to the `rx_dataout_valid_b` signal from the SDI receiver in transceiver mode. *Note:* Applicable for HD-SDI dual link receiver protocol only configuration. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
`rx_trs_loose_lock_in`	1	Input	Indicates that the receiver protocol block detects a single and valid TRS locking signal. This signal must be driven by `rx_trs_loose_lock_out` of the receiver protocol block. *Note:* Applicable for receiver transceiver configuration only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
`rx_trs_loose_lock_in_b`	1	Input	Indicates that the receiver protocol block for link B detects a single and valid TRS locking signal. This signal must be driven by `rx_trs_loose_lock_out_b` of the receiver protocol block. *Note:* Applicable for HD-SDI dual link receiver transceiver configuration only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

continued...

Signal	Width	Direction	Description
rx_trs_in	1	Input	The signal driven by rx_trs to indicate to the PHY management block that the receiver protocol block detected a valid TRS. *Note:* Applicable for receiver transceiver configuration only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_dataout	20*S*	Output	Receiver parallel data out. In dual-rate or triple-rate mode: • Only lower 10 bits are valid for SD-SDI when **SD Interface Bit Width** = 10. In multi-rate mode: • HD/3G-SDI: Only lower 20 bits are valid • 6G-SDI: Only lower 40 bits are valid For bit ordering, refer to tx_datain signal description.
rx_dataout_b	20	Output	Parallel data out signal for the receiver (link B). Applicable only for HD-SDI dual link configuration. *Note:* Applicable for HD-SDI dual link configuration only.
rx_dataout_valid	1	Output	Data valid from the oversampling logic. The receiver asserts this signal to indicate current data on rx_dataout is valid. The timing (H: High, L: Low) for each video standard has the following settings: • SD-SDI = 1H 4L 1H 5L • HD-SDI = H • 3G-SDI = H • HD-SDI Dual Link = H • Dual rate = SD (1H 4L 1H 5L); HD (H) • Triple rate = SD (1H 4L 1H 5L); HD (H); 3G (H) • Multi rate (up to 12G) = SD (1H 4L 1H 5L); HD (H); 3G/6G/12G (H) The 1H4L 1H5L cadence for SD-SDI repeats indefinitely in an ideal case where the video source clock matches the CDR reference clock source. In a typical scenario, you may observe the cadence being shifted periodically (for instance, 1H4L 1H5L 1H5L 1H4L).
rx_dataout_valid_b	1	Output	Data valid from the oversampling logic. The receiver asserts this signal to indicate current data on rx_dataout_b is valid. The timing (H: High, L: Low) for each video standard is identical to the rx_dataout_valid signal. *Note:* Applicable for HD-SDI dual link configuration only.
rx_f	1*S*	Output	Field bit timing signal. This signal indicates which video field is currently active. For interlaced frame, 0 means first field (F0) while 1 means second field (F1). For progressive frame, the value is always 0.
rx_v	1*S*	Output	Vertical blanking interval timing signal. The receiver asserts this signal when the vertical blanking interval is active.
rx_h	1*S*	Output	Horizontal blanking interval timing signal. The receiver asserts this signal when the horizontal blanking interval is active.
rx_ap	1*S*	Output	Active picture interval timing signal. The receiver asserts this signal when the active picture interval is active.
rx_std	3	Output	Receiver video standard. • 3'b000: SD-SDI • 3'b001: HD-SDI • 3'b011: 3G-SDI Level A • 3'b010 3G-SDI Level B • 3'b101: 6G-SDI 4 Streams Interleaved

continued...

Signal	Width	Direction	Description
			• 3'b100: 6G-SDI 8 Streams Interleaved • 3'b111: 12G-SDI 8 Streams Interleaved • 3'b110: 12G-SDI16 Streams Interleaved *Note:* Applicable for 3G-SDI, dual-rate, triple-rate, and multi-rate configurations.
rx_format	4*S*	Output	Indicates the format for the received video transport. Refer to rx_format on page 57 for more information about the video format values.
rx_eav	1*S*	Output	Receiver output that indicates current TRS is EAV. This signal is asserted at the fourth word of TRS, which is the XYZ word.
rx_trs	1*S*	Output	Receiver output that indicates current word is TRS. This signal is asserted at the first word of 3FF 000 000 TRS.
rx_ln	11*S*	Output	Receiver line number output. *Note:* Applicable for all modes except SD-SDI.
rx_ln_b	11*S*	Output	Receiver line number output for link B. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only.
rx_align_locked	1	Output	Alignment locked, indicating that a TRS has been spotted and word alignment is performed.
rx_align_locked_b	1	Output	Alignment locked for link B, indicating that a TRS has been spotted and word alignment is performed. *Note:* Applicable for HD-SDI dual link configuration only.
rx_trs_locked	1*S*	Output	TRS locked, indicating that six consecutive TRSs with same timing has been spotted.
rx_trs_locked_b	1	Output	TRS locked for link B, indicating that six consecutive TRSs with same timing has been spotted. *Note:* Applicable for HD-SDI dual link configuration only.
rx_frame_locked	1	Output	Frame locked, indicating that multiple frames with same timing has been spotted.
rx_frame_locked_b	1	Output	Frame locked for link B, indicating that multiple frames with same timing has been spotted. *Note:* Applicable for HD-SDI dual link configuration only.
rx_dl_locked	1	Output	Dual link locked, indicating that both ports are aligned. *Note:* Applicable for HD-SDI dual link configuration only.
rx_trs_loose_lock_out	1	Output	Indicates that the receiver protocol block detects a single and valid TRS locking signal. This signal must be used to drive rx_trs_loose_lock_in of the receiver transceiver block. *Note:* Applicable for protocol only configuration. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_trs_loose_lock_out_b	1	Output	Indicates that the receiver protocol block for link B detects a single and valid TRS locking signal. This signal must be used to drive rx_trs_loose_lock_in_b of the receiver transceiver block. *Note:* Applicable for HD-SDI dual link protocol only configuration. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_crc_error_c	1*S*	Output	CRC error on chroma channel. Applicable only when you enable CRC checking. *Note:* Applicable for all modes except SD-SDI.

continued...

Signal	Width	Direction	Description
rx_crc_error_y	1*S*	Output	CRC error on luma channel. *Note:* Applicable only when you enable CRC checking. Applicable for all modes except SD-SDI.
rx_crc_error_c_b	1*S*	Output	CRC error on chroma channel for link B. *Note:* Applicable only when you enable CRC checking. Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate modes only.
rx_crc_error_y_b	1*S*	Output	CRC error on luma channel for link B. Applicable only when you enable CRC checking. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate modes only.
rx_vpid_byte1	8*S*	Output	The core extracts payload ID byte 1. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option.
rx_vpid_byte2	8*S*	Output	The core extracts payload ID byte 2. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option.
rx_vpid_byte3	8*S*	Output	The core extracts payload ID byte 3. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option.
rx_vpid_byte4	8*S*	Output	The core extracts payload ID byte 4. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option.
rx_vpid_valid	1*S*	Output	Indicates that the extracted payload ID is valid. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option.
rx_vpid_checksum_error	1*S*	Output	Indicates that the extracted payload ID has a checksum error. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option.
rx_vpid_byte1_b	8*S*	Output	The core extracts payload ID byte 1 for link B. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only.
rx_vpid_byte2_b	8*S*	Output	The core extracts payload ID byte 2 for link B. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only.
rx_vpid_byte3_b	8*S*	Output	The core extracts payload ID byte 3 for link B. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only.
rx_vpid_byte4_b	8*S*	Output	The core extracts payload ID byte 4 for link B. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only.
rx_vpid_valid_b	1*S*	Output	Indicates that the extracted payload ID for link B is valid.

continued...

Signal	Width	Direction	Description
			Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only.
`rx_vpid_checksum_error_b`	1*S*	Output	Indicates that the extracted payload ID for link B has a checksum error. Applicable only when you enable the **Extract Payload ID (SMPTE ST 352)** option. *Note:* Applicable for 3G-SDI, HD-SDI dual link, triple-rate, and multi-rate (up to 12G) modes only.
`rx_line_f0`	11*S*	Output	Line number of field 0 (F0) of the payload ID location. Requires two complete frames to update this signal. Applicable only when you enable the **Extract Video Payload ID (SMPTE ST 352)** option.
`rx_line_f1`	11*S*	Output	Line number of field 1 (F1) of the payload ID location. Requires two complete frames to update this signal. Applicable only when you enable the **Extract Video Payload ID (SMPTE ST 352)** option.

6.3.1. rx_format

The format represents only the video transport format; not the picture format.For example, when the core transports 1080p50 video on HD-SDI dual link, the video transport format is 1080i50.

Table 19. Video Format Values

Encoding Value	SMPTE Standard	Active Lines Per Frame	Transport Format	Frame Rate
0000	SMPTE ST 259	486	I	29.97
0001	SMPTE ST 259	576	I	25
0100	SMPTE ST 274	1080	I	30/29.97/60/59.94(2)
0101	SMPTE ST 274	1080	I	25/50(3)
0110	SMPTE ST 274	1080	P	24/23.98
0111	SMPTE ST 296	720	P	60/59.94
1000	SMPTE ST 296	720	P	50
1001	SMPTE ST 296	720	P	30/29.97
1010	SMPTE ST 296	720	P	25
1011	SMPTE ST 296	720	P	24/23.98
1100	SMPTE ST 274	1080	P	30/29.97/60/59.94
1101	SMPTE ST 274	1080	P	25/50

continued...

(2) Frame rates 60 and 59.94 are meant for 3G Level B/HD Dual Link when receiving 1080p60/59.94 format.

(3) Frame rate 50 is meant for 3G Level B/HD Dual Link when receiving 1080p50 format.

Encoding Value	SMPTE Standard	Active Lines Per Frame	Transport Format	Frame Rate
1110	SMPTE ST 274	1080	I	24
1111	Undetectable format, revert to default value			
Others	Reserved			

To differentiate video format with 1 and 1/1.001 rate, refer to the `rx_clkout_is_ntsc_paln` output signal. For example, if `rx_format` = 0100, `rx_clkout_is_ntsc_paln` = 1, then the format for the received video is 1080i59.94. Otherwise, it is 1080i60.

To differentiate between video format across HD-SDI and 3G-SDI interfaces, also refer to the `rx_std` output signal. For example, if `rx_format` = 1100 and `rx_clkout_is_ntsc_paln` = 0, `rx_std` = 01, then the received video format is 1080p30. If the `rx_std` = 11 or 10, then the received video format is 1080p60.

Note: Intel recommends that you refer to the Payload ID to get the most accurate video format details.

For 6G-SDI or 12G-SDI interfaces, each of the 20-bit interface reports its own detected format. For example, `rx_format` of all four 20-bit interfaces report 1100 (1080p60) when receiving 2160p60 in 12G-SDI, whereas only the lower two interfaces report valid `rx_format` in 6G-SDI.

Table 20. Example of 16-bit rx_format for 6G-SDI and 12G-SDI Interfaces

SDI Interface	rx_format			
	[15:11]	[11:8]	[7:4]	[3:0]
12G-SDI	1100	1100	1100	1100
6G-SDI	Not valid	Not valid	1100	1100

6.4. Transceiver Signals

Table 21. Transceiver Serial Data Pins (for Arria V, Cyclone V, and Stratix V Devices)

Signal	Direction	Description
`sdi_tx`	Output	Transmitter serial out.
`sdi_tx_b`	Output	Transmitter serial out for link B. *Note:* Applicable for HD-SDI dual link configuration only.
`sdi_rx`	Input	Receiver serial in.
`sdi_rx_b`	Input	Receiver serial in for link B. *Note:* Applicable for HD-SDI dual link configuration only.

Table 22. Transceiver Signals

Signal	Width	Clock Domain	Direction	Description
xcvr_refclk_sel	1	tx_coreclk	Input	Transceiver reference clock select signal that selects which clock to be used. • 0 = xcvr_refclk • 1 = xcvr_refclk_alt Applicable only when you enable the **Tx PLL Dynamic Switching** option. *Note:* Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
tx_pll_locked	1	–	Output	PLL locked signal (TX PLL0) for the Native PHY IP core. *Note:* Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
tx_pll_locked_alt	1	–	Output	PLL locked signal (TX PLL1) for the Native PHY IP core. Applicable only when you enable the **Tx PLL Dynamic Switching** option. *Note:* Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_to_xcvr	70*N*	–	Input	Dynamic reconfiguration input for the Native PHY IP core, where N is the reconfiguration interface. • N = 1 for receiver • N = 2 for transmitter and bidirectional *Note:* Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_to_xcvr_b	70*N*	–	Input	Dynamic reconfiguration input for the Native PHY IP core, where N is the reconfiguration interface. • N = 1 for receiver • N = 2 for transmitter and bidirectional *Note:* For HD-SDI dual link configuration only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
reconfig_from_xcvr	46*N*	–	Output	Dynamic reconfiguration output for the Native PHY IP core, where N is the reconfiguration interface. • N = 1 for receiver • N = 2 for transmitter and bidirectional *Note:* Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

continued...

Signal	Width	Clock Domain	Direction	Description
reconfig_from_xcvr_b	46*N*	–	Output	Dynamic reconfiguration output for the Native PHY IP core, where N is the reconfiguration interface. • N = 1 for receiver • N = 2 for transmitter and bidirectional *Note:* For HD-SDI dual link configuration only. Not applicable for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
rx_sdi_start_reconfig	1	rx_coreclk	Output	Request to start dynamic reconfiguration. This signal stays asserted until rx_sdi_reconfig_done indicates that the reconfiguration process is complete. *Note:* Applicable for dual rate, triple-rate, and multi-rate modes only.
rx_sdi_reconfig_done	1	–	Input	Indicates that dynamic reconfiguration has completed. This signal should connect to the reconfiguration status signal of the external transceiver reconfiguration management. • For Arria V, Cyclone V, and Stratix V devices, assertion of this signal indicates to the receiver that the process is done. • For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices, deassertion of this signal indicates to the receiver that the process is done. *Note:* Applicable for dual rate, triple-rate, and multi-rate modes only.
rx_ready	1	–	Input	Status signal from the transceiver reset controller to indicate when Rx PHY sequence is complete. *Note:* Applicable only for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
gxb_ltr	1	rx_coreclk	Output	Control signal to the transceiver rx_set_locktoref input signal. Assertion of this signal programs the Rx CDR to lock manually to reference mode. *Note:* Applicable only for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.
gxb_ltd	1	rx_coreclk	Output	Control signal to the transceiver rx_set_locktodata input signal. *Note:* Applicable only for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices.

7. SDI II Intel FPGA IP Core Design Considerations

There are several considerations that require your attention to ensure the success of your designs.

7.1. Transceiver Handling Guidelines

7.1.1. Handling Transceiver in Arria V, Cyclone V, and Stratix V Devices

In the Arria V, Cyclone V, and Stratix V design example, you can expand the transceiver to multiple channels.

The generated design example consists of two SDI channels, where the SDI duplex instance always occupy Channel 0 (Ch0), while the SDI instance at Channel 1 (Ch1) depends on your selection from the parameter editor. To expand and accommodate more channels, you must perform some modifications to the source files.

For example, when Ch0 is duplex, Ch1 is RX and TX, if you want to instantiate an additional SDI duplex instance at Channel 2 (Ch2), you need to make some modifications to the following components.

7.1.1.1. Modifying the Transceiver Reconfiguration Controller

Perform the following changes to modify the transceiver reconfiguration controller:

- Edit the **Number_of_reconfig_interfaces** parameter. This parameter specifies the total number of reconfiguration interfaces that connect to this block.
- Each channel or TX PLL needs one reconfiguration interface. Therefore, an SDI duplex or SDI TX mode requires two interfaces while an SDI RX mode requires only one interface. If you enable the dynamic TX clock switching feature, the SDI duplex or SDI TX mode requires three interfaces. The additional interface is for the additional TX PLL. For simplicity, assume this option is disabled.
- Determine the total number of reconfiguration interfaces required in your design and assign the parameter value accordingly. In this design example, the total number of reconfiguration interfaces is 7 (Ch0=2, Ch1=3 and Ch2=2).
- Link the `reconfig_to_xcvr` and `reconfig_from_xcvr` signals from the additional SDI duplex instance at Ch2. You must link the signals in the order of the logical channel number (`rx_log_ch_num` and `tx_log_ch_num`) in the reconfiguration logic source file (`sdi_ii_reconfig_logic.v`).
- In the design example that instantiates the transceiver reconfiguration controller, add the wire connection between the additional SDI duplex instance at Ch2 and the transceiver reconfiguration controller as shown below:

```
wire [ 139:0] reconfig_to_xcvr_du_ch2;
wire [  91:0] reconfig_from_xcvr_du_ch2;
wire [ 139:0] reconfig_to_xcvr_tx_ch1;
wire [  69:0] reconfig_to_xcvr_rx_ch1;
wire [  91:0] reconfig_from_xcvr_tx_ch1;
wire [  45:0] reconfig_from_xcvr_rx_ch1;
wire [ 139:0] reconfig_to_xcvr_du_ch0;
wire [  91:0] reconfig_from_xcvr_du_ch0;

alt_xcvr_reconfig #(
   .number_of_reconfig_interfaces (7),
   ….
) u_reconfig (
   .reconfig_to_xcvr    ({reconfig_to_xcvr_du_ch2,
                          reconfig_to_xcvr_tx_ch1,
                          reconfig_to_xcvr_rx_ch1,
                          reconfig_to_xcvr_du_ch0}),
   .reconfig_from_xcvr  ({reconfig_from_xcvr_du_ch2,
                          reconfig_from_xcvr_tx_ch1,
                          reconfig_from_xcvr_rx_ch1,
                          reconfig_from_xcvr_du_ch0}),
);
```

7.1.1.2. Modifying the Reconfiguration Management

Perform the following changes to modify the reconfiguration management:

- Edit the **Number_of_channels** parameter in `sdi_ii_ed_reconfig_mgmt.v`. This parameter value should be the total number of the SDI RX channels declared in the design. In this example, the NUM_CHS is 3.
- Link the interface signals—`sdi_rx_start_reconfig`, `sdi_rx_reconfig_done`, and `sdi_rx_std`—between multiple SDI instances and reconfiguration management block. Link the interface signals—`sdi_tx_start_reconfig`, `sdi_tx_reconfig_done`, and `sdi_tx_pll_sel`—between user and reconfiguration management block. You must link the signals in the order of the logical channel number (`rx_log_ch_num` and `tx_log_ch_num`) in the reconfiguration logic source file (`sdi_ii_reconfig_logic.v`). For example:

```
wire tx_start_reconfig_ch2,tx_start_reconfig_ch1,tx_start_reconfig_ch0;
wire tx_pll_sel_ch2,tx_pll_sel_ch1,tx_pll_sel_ch0;
wire tx_reconfig_done_ch2,tx_reconfig_done _ch1,tx_reconfig_done_ch0;
wire rx_start_reconfig_ch2,rx_start_reconfig_ch1,rx_start_reconfig_ch0;
wire [1:0] rx_std_ch2, rx_std_ch1,rx_std_ch0;
wire rx_reconfig_done_ch2,rx_reconfig_done _ch1,rx_reconfig_done_ch0;
```

```
sdi_ii_ed_reconfig_mgmt #(
   . NUM_CHS (3),
) u_reconfig_mgmt (
   .sdi_tx_start_reconfig (tx_start_reconfig_ch2,
tx_start_reconfig_ch1,tx_start_reconfig_ch0),
   .sdi_tx_pll_sel  (tx_pll_sel_ch2,tx_pll_sel_ch1,tx_pll_sel_ch0),
   .sdi_tx_reconfig_done (tx_reconfig_done_ch2,
tx_reconfig_done_ch1,tx_reconfig_done_ch0),
   .sdi_rx_start_reconfig (rx_start_reconfig_ch2,
rx_start_reconfig_ch1,rx_start_reconfig_ch0),
   .sdi_rx_std   (rx_std_ch2,rx_std_ch1,rx_std_ch0),
   .sdi_rx_reconfig_done (rx_reconfig_done_ch2,
rx_reconfig_done_ch1,rx_reconfig_done_ch0)
)
```

- In the reconfiguration logic source file, the default setting for the wire `rx_log_ch_num` is 0 and 2 for channel 0 and channel 1, respectively. The default setting for the wire `tx_log_ch_num` is 0 and 2 (duplex) or 3 (TX) for channel 0 and channel 1, respectively. These numbers are referring to the **Number_of_channels** parameter value that was set in the transceiver reconfiguration controller. The logical channel number for each SDI channel is as listed in the table below.

Table 23. Logical Channel Number for Each SDI Channel

SDI Channel	Direction	Number of Reconfiguration Interfaces	Logical Channel Number
0	Duplex	2	• 0: RX/TX channel • 1: Tx PLL
1	RX and TX	3 (1 for RX and 2 for TX)	• 2: RX channel • 3: TX channel • 4: TX PLL
2	Duplex	2	• 5: RX/TX channel • 6: TX PLL

- Edit the reconfiguration logic source file to assign the logical channel number for the additional SDI duplex instance, which occupies the SDI Ch2. The logical channel number specified in the source file is the reconfiguration interface that is intended for dynamic reconfiguration. For example, if TX channel is intended for dynamic reconfiguration, `tx_log_ch_num[2]` should be 5.

```
wire [7:0] rx_log_ch_num [0:NUM_CHS-1];
   assign rx_log_ch_num[0] = 8'd0; // Duplex Rx channel share same
                                      logical channel number with Tx
   assign rx_log_ch_num[1] = 8'd2; // Rx channel
   assign rx_log_ch_num[2] = 8'd5; // Duplex Rx channel

wire [7:0] tx_log_ch_num [0:NUM_CHS-1];
   assign tx_log_ch_num[0] = 8'd0; // Duplex Tx channel share same
                                      logical channel number with Rx
   assign tx_log_ch_num[1] = 8'd3; // Tx channel
   assign tx_log_ch_num[2] = 8'd5; // Duplex Tx channel
```

Related Information

Transceiver PHY IP Core User Guide
More information about the transceiver reconfiguration controller logical channel numbering.

7.1.1.3. Modifying the Reconfiguration Router

For ease of implementation, you can bypass this block by connecting the interface signals—`reconfig_to_xcvr`, `reconfig_from_xcvr`, `sdi_rx_start_reconfig`, `sdi_rx_reconfig_done`, `sdi_rx_std`, `sdi_tx_start_reconfig`, `sdi_tx_reconfig_done`, and `sdi_tx_pll_sel`—directly between the SDI instance and the transceiver reconfiguration controller or the reconfiguration management.

7.1.2. Handling Transceiver in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 Devices

For Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 design examples, duplicate another transceiver reconfiguration management generated from the design for additional channels.

The respective Transceiver Native PHY IP cores provide the following SDI presets that you can apply to your design. If you do not use the presets, the Intel Quartus Prime software generates your transceiver configurations together with the design example.

Table 24. SDI Presets in the Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP and L-Tile/H-Tile Transceiver Native PHY Intel Stratix 10 IP Cores

Presets	Description
SDI 3G NTSC	• Preset for 3G-SDI single rate (TX and RX) and triple rate TX. • Set for SDI data rate factor of 1/1.001. • Configured in Duplex mode. You may change the direction based on your design needs.
SDI 3G PAL	• Preset for 3G-SDI single rate (TX and RX) and triple rate TX. • Set for SDI data rate factor of 1/1. • Configured in Duplex mode. You may change the direction based on your design needs.

continued...

Presets	Description
SDI HD NTSC	• Preset for HD-SDI single rate and HD-SDI dual link (TX and RX). • Set for SDI data rate factor of 1/1.001. • Configured in Duplex mode. You may change the direction based on your design needs.
SDI HD PAL	• Preset for HD-SDI single rate and HD-SDI dual link (TX and RX). • Set for SDI data rate factor of 1/1. • Configured in Duplex mode. You may change the direction based on your design needs.
SDI Multi rate (up to 12G) Rx	• Preset for multi rate up to 12G-SDI (RX). • Contains multiple profiles for HD-SDI, 3G-SDI, 6G-SDI, and 12G-SDI for dynamic reconfiguration. If you want to use duplex mode, combine the Tx settings from the SDI Multi rate Tx preset and only profile 0 of the Rx preset. Disable the multiple profiles option. However, if you are using the reconfig files from the design example as the reconfiguration management block, you may need to instantiate the PHY IP core with this preset to generate all 4 `reconfig_parameмter_CFG` files and add into your design.
SDI Multi rate (up to 12G) Tx	• Preset for multi rate up to 12G-SDI (TX). • Configured in data rate of 11,880 Mbps. • Change the data rate to 11,868 Mbps to transmit with data rate factor of 1/1.001. If you want to use duplex mode, combine the Tx settings from this preset and only profile 0 of the SDI Multi rate Rx preset. Disable the multiple profiles option. However, if you are using the reconfig files from the design example as the reconfiguration management block, you may need to instantiate the PHY IP core with the SDI Multi rate Rx preset to generate all 4 `reconfig_parameмter_CFG` files and add into your design.
SDI Triple rate Rx	• Preset for triple rate up to 3G-SDI (RX). • Contains multiple profiles for HD-SDI and 3G-SDI for dynamic reconfiguration. If you want to use duplex mode, combine the Tx settings from the SDI 3G NTSC or SDI 3G PAL preset and only profile 0 of this preset. Disable the multiple profiles option. However, if you are using the reconfig files from the design example as the reconfiguration management block, you may need to instantiate the PHY IP core with this preset to generate all 2 `reconfig_parameмter_CFG` files and add into your design.

7.1.2.1. Changing RX CDR Reference Clock in Transceiver Native PHY IP Core

For triple-rate or multi-rate modes, you must modify the reference clock value for every profile if you are going to change the CDR reference clock value.

To change the CDR frequency, make the following settings in the respective Transceiver Native PHY parameter editor:

1. On the **RX PMA** tab, for the **Selected CDR reference clock frequency** parameter, select the desired clock frequency, e.g. 297 MHz.
2. Then, on the **Dynamic Reconfiguration** tab, click **Store configuration to selected profile**. The default profile (e.g. **0**) is now configured.
3. If there are more than one profile, select the subsequent profile (e.g. **1**) at the **Selected reconfiguration profile** parameter.
4. Click **Load configuration from selected profile** to load profile 1.
5. Then on the **RX PMA** tab, select **297** MHz.
6. Repeat until all the profiles are configured.

7.1.2.2. Merging Simplex Mode Transceiver in the Same Channel

To merge simplex mode transceiver in the same channel, add the following commands in the Quartus Settings File (`.qsf`) in your project directory:

- `set_instance_assignment -name XCVR_RECONFIG_GROUP 1 -to <tx_serial_pin>`
- `set_instance_assignment -name XCVR_RECONFIG_GROUP 1 -to <rx_serial_pin>`

For more details about merging transceivers, refer to the *Dynamic Reconfiguration Interface Merging Across Multiple IP Blocks* section in the respective FPGA Transceiver PHY user guides.

Related Information

- Transceiver PHY Intel Arria 10 FPGA IP User Guide
- Transceiver PHY Intel Cyclone 10 FPGA IP User Guide
- L-Tile/H-Tile Transceiver PHY Intel Stratix 10 FPGA IP User Guide

7.1.2.3. Using Generated Reconfiguration Management for Triple and Multi Rates

You may encounter the following errors when you use the generated reconfiguration management block from the Intel Quartus Prime Standard Edition software:

- `Error (10161): Verilog HDL error at rcfg_sdi_cdr.sv: object "altera_xcvr_native_a10_reconfig_parameters_CFG0" is not declared. Verify the object name is correct. If the name is correct, declare the object.`
- `Error (10161): Verilog HDL error at rcfg_sdi_cdr.sv: object "altera_xcvr_native_a10_reconfig_parameters_CFG1" is not declared. Verify the object name is correct. If the name is correct, declare the object.`

The reconfiguration management block requires the *CFG* files that are generated from the transceiver to determine which registers to be reconfigured for data rate changes. However, the Intel Quartus Prime software cannot recognize these files outside of the transceiver library files.

To resolve this issue, add the library switch to the `rcfg_sdi_cdr.sv` file in your project's `.qsf`.

```
set_global_assignment -name SYSTEMVERILOG_FILE <file hierarchy
before the file>/rcfg_sdi_cdr.sv -library
<phy_name_quartus_version>
```

1. Find the exact *library* name that you should assign in the transceiver `.qip` file.
2. Open the transceiver `.qip` file and search for the string: `parameter_CFG0`.

   You should see: `set_global_assignment –library <phy_name_quartus_version> -name SYSTEMVERILOG_FILE ….CFG0.sv`.

7.1.2.4. Ensuring Independent RX and TX Operations in the Same Channel

The `rx_cal_busy` and `tx_cal_busy` signals from the transceiver are from the same internal node and change state concurrently during calibration. Because these signals are from the same internal node, the RX and TX transceivers in the same channel are affected by each other when one transceiver is in calibration. Problems may occur when the RX and TX transceivers in the same channel are required to work independently, because the TX is held in reset when the RX recalibrates or vice versa.

A possible workaround for this problem is to use the transceiver arbiter from the generated design example. For more details about the arbiter's signal interface, refer to the respective design example user guides.

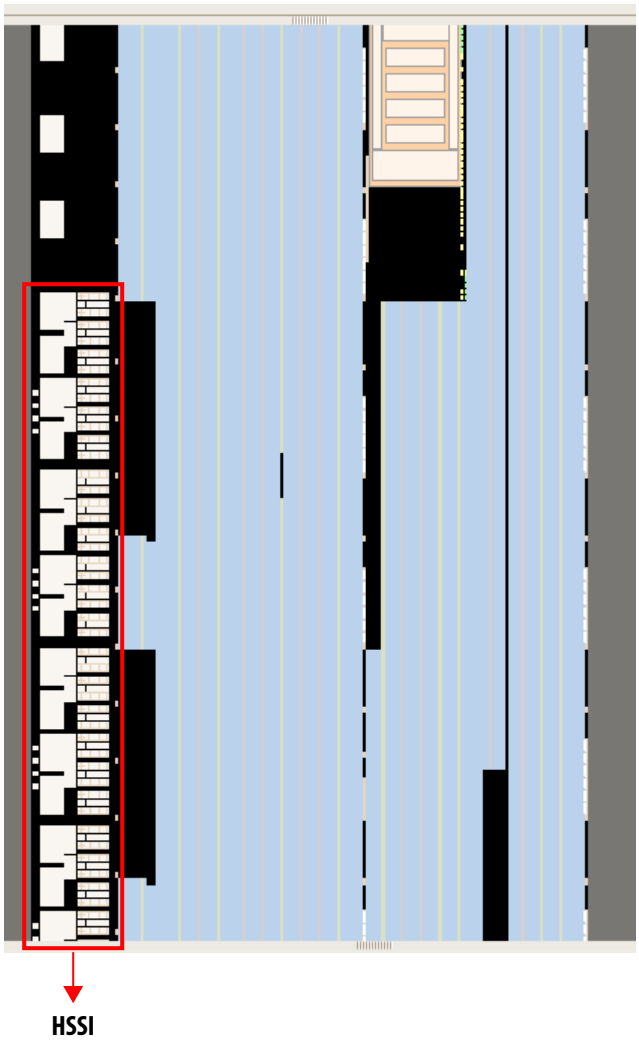
Related Information

- Capability Registers
  Provides more information about capability registers.
- SDI II Intel Stratix 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Stratix 10 devices.
- SDI II Intel Arria 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Arria 10 devices.
- SDI II Intel Cyclone 10 GX FPGA IP Design Example User Guide
  Provides the design examples for Intel Cyclone 10 GX devices.

7.1.2.5. Potential Routing Problem During Fitter Stage in Intel Arria 10 and Intel Cyclone 10 GX Devices

The SDI II Intel FPGA IP core must to be paired with HSSI channels. For certain Intel Arria 10 and Intel Cyclone 10 GX device parts, all the HSSI channels reside at one side of the chip. Multiple instantiations of the SDI II Intel FPGA IP core in a design (especially for multi-rate mode) may cause that side of the chip to be congested with the ALMs and core logic.

Figure 39. Chip Planner View of HSSI Channels Placement on an Intel Arria 10 Device



The architecture for Intel Arria 10 and Intel Cyclone 10 GX devices is designed to place most HSSI clocks on the peripheral clocks (PCLKs). The logic of the IP core may not fit efficiently into the available regions covered by the PCLKs, and moving the logic farther away is not ideal because the logic needs to interact with the HSSI channels. These circumstances may cause routing challenge and Fitter failure.

To overcome this issue, check the placement of the HSSI channels on the chip and consider the availability of the resources on that side before starting your design.

7.1.2.6. Unconstrained Clocks in SDI Multi-Rate RX Using Intel Arria 10 and Intel Cyclone 10 GX Devices

You will observe some unconstrained clocks from the Transceiver Native PHY Intel Arria 10/Cyclone 10 FPGA IP core if you are using the SDI Multi rate (up to 12G) RX preset with Intel Arria 10 or Intel Cyclone 10 GX devices.

For example, you may observe the following report in an RX PHY with simplex configuration:

```
<Rx PHY path …|…
gen_twentynm_hssi_8g_rx_pcs.inst_twentynm_hssi_8g_rx_pcs~byte_deserializer_pcs_c
lk_div_by_2_reg.reg>
```

To resolve this violation, apply the following constraints in your `.sdc` file.

1. Create generated clock name (name this clock).

```
create_generated_clock -name (Clock Name, e.g. <Rx PHY path>||rx_clk}
-source {<Rx PHY path>|g_xcvr_native_insts[0].twentynm_xcvr_native_inst|
twentynm_xcvr_native_inst|
inst_twentynm_pcs|gen_twentynm_hssi_8g_rx_pcs.inst_twentynm_hssi_8g_rx_pcs|
byte_deserializer_pcs_clk_div_by_2_reg} \\

-divide_by 2 -multiply_by 1 -duty_cycle 50.00 \\

{<Rx PHY path>|g_xcvr_native_insts[0].twentynm_xcvr_native_inst|
twentynm_xcvr_native_inst|
inst_twentynm_pcs|
gen_twentynm_hssi_8g_rx_pcs.inst_twentynm_hssi_8g_rx_pcs~byte_deserializer_p
cs_clk_div_by_2_reg.reg}
```

2. Set false path.

```
set_false_path [get_clocks {<Clock name given in (1), e.g. <Rx PHY path>||
rx_clk}]
```

Note: Refer to the generated `.sdc` file for the design example provided in the parameter editor: `sdi_ii_a10_demo.sdc` (Intel Arria 10) or `sdi_ii_c10_demo.sdc` (Intel Cyclone 10 GX).

7.2. Timing Violation

You may avoid some timing violation for Arria V, Cyclone V, and Stratix V designs by editing `.qsf`.

After you create a new project, the Intel Quartus Prime software generates a `.qsf`. Add the following assignments to the `.qsf` file to avoid timing violation from the synchronizers.

```
set_instance_assignment -name GLOBAL_SIGNAL OFF -to *|
altera_reset_synchronizer:alt_rst_sync_uq1|
altera_reset_synchronizer_int_chain_out
```

8. SDI II Intel FPGA IP Core Testbench and Design Examples

Intel offers design examples that you can simulate, compile, and test in hardware.

The implementation of the SDI II Intel FPGA IP on hardware requires additional components specific to the targeted device.

8.1. Design Examples for Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10

The SDI II Intel FPGA IP core offers design examples that you can generate through the IP catalog in the Intel Quartus Prime Pro Edition software.

For detailed information about the SDI II Intel FPGA IP design examples, refer to following user guides:

Related Information

- SDI II Intel Stratix 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Stratix 10 devices.
- SDI II Intel Arria 10 FPGA IP Design Example User Guide
  Provides the design examples for Intel Arria 10 devices.
- SDI II Intel Cyclone 10 GX FPGA IP Design Example User Guide
  Provides the design examples for Intel Cyclone 10 GX devices.

8.2. Design Examples for Arria V, Cyclone V, and Stratix V Devices

The SDI II Intel FPGA design example for Arria V, Cyclone V, or Stratix V devices are synthesizable.

Figure below illustrates the generated design example entity and simulation testbench for Arria V, Cyclone V, and Stratix V devices. This design example consists of a video pattern generator, transceiver reconfiguration controller, reconfiguration management, loopback path, and various SDI blocks occupying two transceiver channels.

Figure 40. Design Example Entity and Simulation Testbench

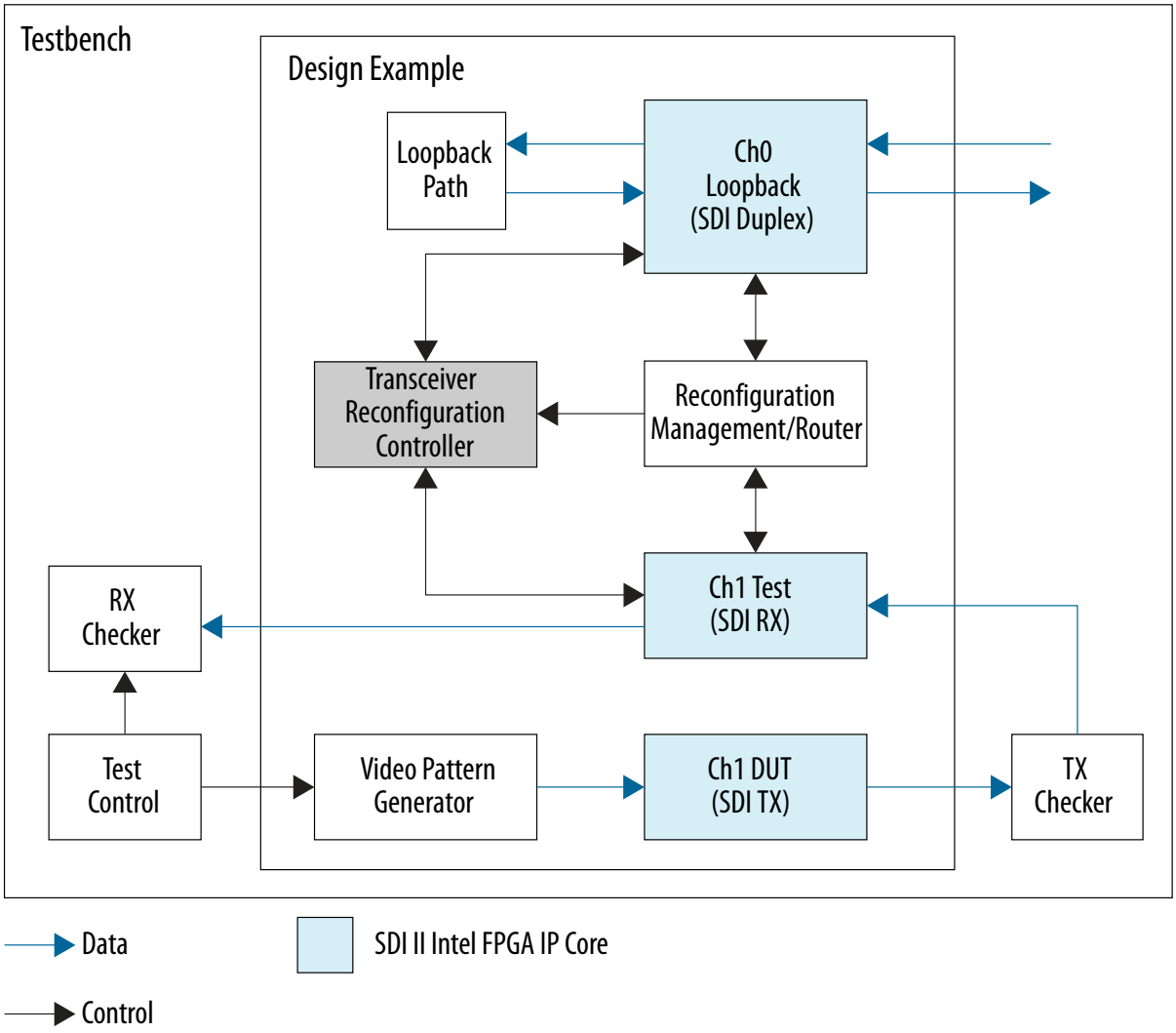


Figure 41. Design Example Entity and Simulation Testbench for HD-SDI Dual Link to 3G-SDI (Level B) Conversion

The figure below illustrates the generated design example entity and simulation testbench when you generate HD-SDI dual link receiver with **Convert HD-SDI dual link to 3G-SDI (level B)** option enabled.

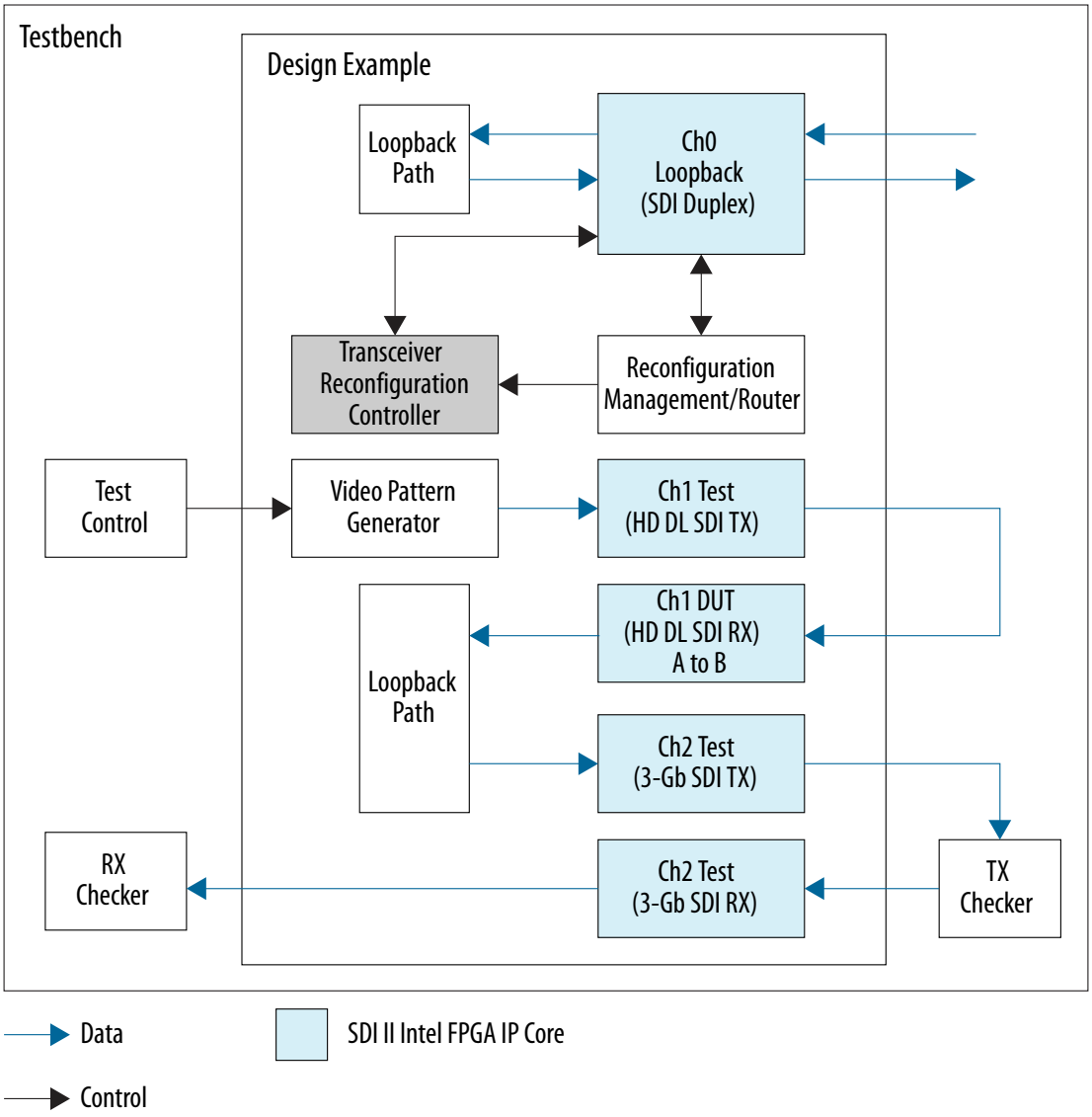
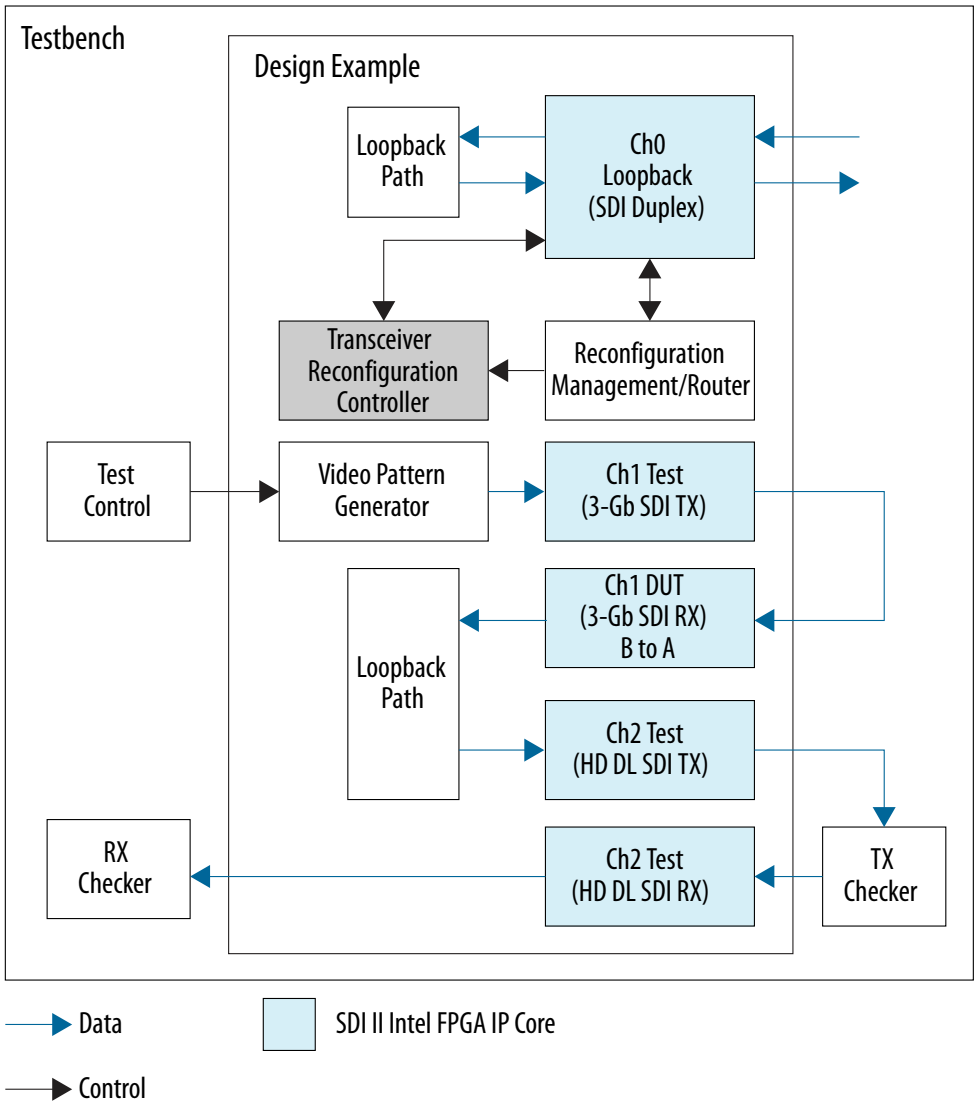


Figure 42. Design Example Entity and Simulation Testbench for 3G-SDI (Level B) to HD-SDI Dual Link Conversion

The figure below illustrates the generated design example entity and simulation testbench when you generate 3G-SDI or triple rate SDI receiver with **Convert 3G-SDI (level B) to HD-SDI dual link** option enabled.



8.2.1. Design Example Components

The Arria V, Cyclone V, and Stratix V design examples for the SDI II Intel FPGA IP core consist of the following components:

- Video pattern generator
- Transceiver reconfiguration controller
- Reconfiguration management
- Reconfiguration router

8.2.1.1. Video Pattern Generator

The video pattern generator generates a colorbar or pathological pattern. The colorbar is preferable for image generation while the pathological pattern can stress the PLL and cable equalizer of the attached video equipment. You can configure the video pattern generator to generate various video formats.

Table 25. Configuring the Video Pattern Generator to Generate Different Video Formats

Table below lists the examples of how to configure the video pattern generator signals to generate a video format that you desire.

Example	Video Format	Interface	Signal		
			pattgen_tx_std	pattgen_tx_format	pattgen_dl_mapping
Example 1: Generate 1080i video format	1080i60	HD-SDI	3'b001	4'b0100	1'b0
	1080i60×2	HD-SDI dual link	3'b001	4'b0100	1'b0
		3Gb	3'b010	4'b0100	1'b0
Example 2: Generate 1080p video format	1080p30	HD-SDI	3'b001	4'b1100	1'b0
	1080p30×2	HD-SDI dual link	3'b001	4'b1100	1'b0
	1080p60	HD-SDI dual link	3'b001	4'b1100	1'b1
		3Ga	3'b011	4'b1100	1'b0
		3Gb	3'b010	4'b1100	1'b1

Related Information

Video Pattern Generator Signals on page 77

8.2.1.2. Transceiver Reconfiguration Controller

For Arria V, Cyclone V, and Stratix V design examples, the transceiver reconfiguration controller allows you to change the device transceiver settings at any time.

Any portion of the transceiver can be selectively reconfigured. Each portion of the reconfiguration requires a read-modify-write operation (read first, then write), in such a way by modifying only the appropriate bits in a register and not changing other bits. Prior to this operation, you must define the logical channel number and the streamer module mode.

Note: The transceiver reconfiguration controller only reconfigures the TX transceiver if you are performing TX clock switching.

You can perform a transceiver dynamic reconfiguration in these two modes:

- Streamer module mode 1 (manual mode)—execute a series of Avalon-MM write operation to change the transceiver settings. In this mode, you can execute a write operation directly from the reconfiguration management/router interface to the device transceiver registers.
- Streamer module mode 0—use the `.mif` files to change the transceiver settings.

For read operation, after defining the logical channel number and the streamer module mode, the following sequence of events occur:

1. Define the transceiver register offset in the offset register.
2. Read the data register. Toggle the read process by setting bit 1 of the control and status register (CSR) to logic 1.
3. Once the busy bit in the CSR is cleared to logic 0, it indicates that the read operation is complete and the required data should be available for reading.

For write operation, after setting the logical channel number and the streamer module mode, the following sequence of events occur:

1. Define the transceiver register offset (in which the data is written to) in the offset register.
2. Write the data to the data register. Toggle the write process by setting bit 0 of the CSR to logic 1.
3. When the busy bit in the CSR is cleared to logic 0, it indicates that the transceiver register offset modification is successful.

Related Information

- Transceiver Reconfiguration Controller Signals on page 78
- Modifying the Transceiver Reconfiguration Controller on page 62
- V-Series Transceiver PHY FPGA IP User Guide
  Provides more information about the transceiver reconfiguration controller streamer module.

8.2.1.3. Reconfiguration Management

The reconfiguration management block (`sdi_ii_ed_reconfig_mgmt.v` and `sdi_ii_reconfig_logic.v`) contains the reconfiguration user logic (a finite state machine) to determine the bits that needs to be modified, and selects the correct data to be written to the appropriate transceiver register through streamer module mode 1. It also provides handshaking between the SDI receiver and the transceiver reconfiguration controller. In this design, each reconfiguration block must interface with only one transceiver reconfiguration controller.

During the reconfiguration process, the logic first reads the data from the transceiver register that needs to be reconfigured and stores the data temporarily in a local register. Then, the logic overwrites only the appropriate bits of the data with predefined values and write the modified data to the transceiver register. Since only one transceiver register can be accessed at a time, the whole process repeats when reconfiguring other registers.

For multiple SDI channels reconfiguration, the logical channel number needs to be set appropriately for each channel and reconfiguration interface. For example, in the design example and simulation testbench figure, there are one SDI duplex, one SDI RX, and one SDI TX block. The number of reconfiguration interface for SDI duplex is 2 (one for channel and one for TX PLL), for SDI RX is 1 (for channel), for SDI TX is 2 (one for channel and one for TX PLL). The total number of reconfiguration interface required in the transceiver reconfiguration controller is 5.

The table below lists the channel and transceiver reconfiguration controller interface numbers.

The logical channel number for the receiver in SDI duplex is 0 and the logical channel number for SDI RX is 2. The generated example design entity demonstrates this interface connection.

Table 26. Channel Numbers Setting for Multiple SDI Channels Reconfiguration

SDI Block	SDI Channel Number	Transceiver Reconfiguration Controller Interface Number
SDI Duplex	0	0 and 1
SDI RX	1	2
SDI TX	1	3 and 4

Related Information

- Reconfiguration Management Parameters on page 79
- Modifying the Reconfiguration Management on page 63
- V-Series Transceiver PHY IP Core User Guide
  Provides more information about the logical channel number.

8.2.1.4. Reconfiguration Router

The reconfiguration router (`sdi_ii_ed_reconfig_router.v`) connects multiple SDI instances to the reconfiguration management and transceiver reconfiguration controller blocks. The reconfiguration router receives all the interface signals between the transceiver reconfiguration controller and reconfiguration management, as well as SDI instances, and transmits the signals to their respective destinations.

The reconfiguration router converts reconfiguration related interface signals of multiple SDI instances and user interface to a single-wide data bus for the reconfiguration management and transceiver reconfiguration controller blocks. You can bypass this component if you want to implement designs that expands to more channels.

Related Information

- Reconfiguration Router Signals on page 80
- Modifying the Reconfiguration Router on page 64

8.2.1.5. Avalon-MM Translators

The Avalon-MM Master Translator and Avalon-MM Slave Translator are Avalon-MM interface blocks that access the Transceiver Reconfiguration Controller registers. The translators are not SDI-specific and are automatically instantiated when the core interfaces with an Avalon-MM master or slave component.

If you want to bypass the Avalon MM translator in your design, connect `reconfig_mgmt_address[8:2]` from the reconfiguration management block to `reconfig_mgmt_address` from the Transceiver Reconfiguration Controller.

You can connect the other signals from the reconfiguration management block directly to the Transceiver Reconfiguration Controller.

- reconfig_mgmt_waitrequest
- reconfig_mgmt_read
- reconfig_mgmt_readdata
- reconfig_mgmt_write
- reconfig_mgmt_writedata

Related Information

- Platform Designer Interconnect
  Provides more information about the Avalon-MM Translator functions.
- Avalon Interface Specifications

8.2.2. Design Reference

This section includes detailed description about the SDI II Intel FPGA IP core design examples.

8.2.2.1. Video Pattern Generator Signals

Table 27. Video Pattern Generator Top Level Signals

Table below lists the input signals for the video pattern generator. The listed signals are exported at the top level of the design example. Other signals—that are not exported—connect within the design example entity.

Signal	Width	Direction	Description
pattgen_tx_std	3	Input	Transmit video standard. • 000: SD-SDI • 001: HD-SDI or HD-SDI dual link • 010: 3G-SDI level B • 011: 3G-SDI level A
pattgen_tx_format	4	Input	Transmit video format. • 0000: ST 259 525i • 0001: ST 259 625i • 0100: ST 274 1080i60/ST 274 1080sF30 • 0101: ST 274 1080i50/ST 274 1080sF25 • 0110: ST 274 1080p24 • 0111: ST 296 720p60 • 1000: ST 296 720p50 • 1001: ST 296 720p30 • 1010: ST 296 720p25 • 1011: ST 296 720p24 • 1100: ST 274 1080p30/ST 274 1080p60 • 1101: ST 274 1080p25/ST 274 1080p505 • 1110: ST 274 1080sF24 • Others: Reserved for future use
pattgen_dl_mapping	1	Input	Dual link mapping. Set to 1'b1 for HD-SDI dual link and 3Gb transmit video standard only.
pattgen_ntsc_paln	1	Input	Transmit rate. • 0: PAL (1) rate. For example, 1080p30 • 1: NTSC (1/1.001) rate. For example, 1080p29.97. This input ignores all SD video formats (525i, 625i) and certain HD video formats that do not support NTSC rate (1080i50, 720p50, 720p25, 1080p25).

continued...

Signal	Width	Direction	Description
pattgen_bar_100_75n	1	Input	Generate color bars. • 0: 75% color bars • 1: 100% color bars
pattgen_patho	1	Input	Set to 1'b1 to generate pathological pattern.
pattgen_blank	1	Input	Set to 1'b1 to generate black signal.
pattgen_no_color	1	Input	Set to 1'b1 to generate bars with no color.
pattgen_sgmt_frame	1	Input	Set to 1'b1 to generate segmented frame picture for tx_format: • 0100: ST 274 1080sF30 • 0101: ST 274 1080sF25

Related Information

Video Pattern Generator on page 74

8.2.2.2. Transceiver Reconfiguration Controller Signals

Table 28. Transceiver Reconfiguration Controller Signals for Arria V, Cyclone V, and Stratix V Devices

Table below lists the input signals for the transceiver reconfiguration controller. The listed signals are exported at the top level of the design example. Other signals—that are not exported—connects within the design example entity.

Signal	Width	Direction	Description
reconfig_clk	1	Input	Clock signal for the transceiver reconfiguration controller and reconfiguration management/router. Refer to the transceiver reconfiguration controller section in the *V-Series Transceiver PHY IP Core User Guide* for information about the frequency range.
reconfig_rst	1	Input	Reset signal for the transceiver reconfiguration controller and reconfiguration management/router. This signal is active high and level sensitive.

Related Information

- Transceiver Reconfiguration Controller on page 74
- Modifying the Transceiver Reconfiguration Controller on page 62
- V-Series Transceiver PHY FPGA IP User Guide
  Provides more information about the transceiver reconfiguration controller frequency range.

8.2.2.3. Reconfiguration Management Parameters

Tables below list the parameters for reconfiguration management.

Table 29. Reconfiguration Management Parameters for Arria V, Cyclone V, and Stratix V Devices

Parameter	Value	Description
NUM_CHS	1 (minimum)	Number of channels required to do reconfiguration.
FAMILY	• Arria V • Arria V GZ • Cyclone V • Stratix V	Supported device family.
DIRECTION	• tx • rx • du	Direction of the core selected in the parameter editor. This parameter affects the logical channel number assigned in the generated example design. If you are making any changes to the design, please ignore this parameter and assign the logical channel number correctly. Refer to *Expanding to Multiple Channels* section to know how to assign the logical channel number.
VIDEO_STANDARD	• tr • dl	Current video standard. Specify *dl* for HD dual-link or *tr* for other standards.
XCVR_TX_PLL_SEL	• 1 • 2	The selected method to perform TX PLL reconfiguration for dynamic clock switching. Specify 1 to switch TX PLL or 2 to switch TX PLL reference clock. The specified value must match the parameter value you select when you instantiate the IP core. Refer to *Dynamic TX Clock Switching* section to know more about clock switching.

Related Information

- Reconfiguration Management on page 75
- Modifying the Reconfiguration Management on page 63

8.2.2.4. Reconfiguration Router Signals

Table below lists the signals for the reconfiguration router.

Table 30. Reconfiguration Router Top Level Signals

The listed signals are exported at the top level of the design example. Other signals—that are not exported—connect within the design example entity.

Note: These signals are available only when you use the Dynamic TX clock switching feature.

Refer to Dynamic TX Clock Switching for usage requirements.

Signal	Width	Direction	Description
ch1_<direction>_tx_start_reconfig	1	Input	Dynamic reconfiguration request signal for TX PLL dynamic switching at transmitter or duplex instance at channel 1.
ch1_<direction>_tx_pll_sel	1	Input	TX PLL select signal for TX PLL dynamic switching at transmitter or duplex instance at channel 1. This signal is also connected to xcvr_refclk_sel signal of the SDI instance.
ch1_<direction>_tx_reconfig_done	1	Output	Dynamic reconfiguration acknowledge signal for TX PLL dynamic switching at transmitter or duplex instance at channel 1.

Related Information

- Reconfiguration Router on page 76
- Modifying the Reconfiguration Router on page 64

8.2.3. Simulating the SDI II Intel FPGA IP Core Design

After design generation, the files located in the simulation testbench directory are available for you to simulate your design.

The SDI II Intel FPGA IP core supports the following EDA simulators listed in the table below.

Table 31. Supported EDA Simulators

Simulator	Supported Platform	Supported Language
ModelSim - Intel FPGA Starter Edition	Windows/Linux	VHDL and Verilog HDL
ModelSim - Intel FPGA Edition	Windows/Linux	Verilog HDL
Synopsys VCS/VCS MX	Windows/Linux	Verilog HDL
Aldec Riviera-PRO	Linux	Verilog HDL

To simulate the design using the ModelSim - Intel FPGA Starter Edition or ModelSim - Intel FPGA Edition simulator, follow these steps:

1. Start the simulator.
2. On the File menu, click **Change Directory > Select** <*simulation folder*>/<*preferred HDL*>/**mentor**.
3. Run the provided `run_sim.tcl` script. This file compiles the design and runs the simulation automatically. It provides a pass/fail indication on completion.

To simulate the design using the VCS/VCS MX simulator (in Linux), follow these steps:

1. Start the VCS/VCS MX simulator.
2. On the File menu, click **Change Directory > Select** <*simulation folder*>/<*preferred HDL*>/**synopsys**.
3. Run the provided `run_vcs.sh` (in VCS) or `run_vcsmx.sh` (in VCSMX) script. This file compiles the design and runs the simulation automatically. It provides a pass/fail indication on completion.

To simulate the design using the Aldec Riviera-PRO simulator, follow these steps:

1. Start the Aldec Riviera-PRO simulator.
2. On the File menu, click **Change Directory > Select** <*simulation folder*>/<*preferred HDL*>/**aldec**.
3. Run the provided `run_riviera.tcl` script. This file compiles the design and runs the simulation automatically. It provides a pass/fail indication on completion.

8.2.3.1. Simulation Run Time

Table 32. Estimated Simulation Run Time

The table lists the default estimated run-time settings for each video standard.

Video Standard	Estimated Run Time (ms)
SD-SDI	3.26
HD-SDI	9.51
3G-SDI	7.62
HD-SDI Dual Link	6.15
Dual Rate (up to HD-SDI)	13.44
Triple Rate (up to 3G-SDI)	55.83

9. SDI II Intel FPGA IP User Guide Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

IP Core Version	User Guide
18.1	SDI II Intel FPGA IP User Guide
18.0	SDI II Intel FPGA IP User Guide
17.1	Intel FPGA SDI II IP Core User Guide
17.0	SDI II IP Core User Guide
16.1	SDI II IP Core User Guide
16.0	SDI II IP Core User Guide
15.1	SDI II IP Core User Guide
15.0	SDI II IP Core User Guide
14.1	SDI II IP Core User Guide

10. Document Revision History for the SDI II Intel FPGA IP User Guide

Document Version	Intel Quartus Prime Version	Changes
2019.08.08	19,1	Edited a bad character in the *Merging Simplex Mode Transceiver in the Same Channel* section.
2019.04.01	19.1	• Added support for Intel Stratix 10 L-tile devices. Support for both Intel Stratix 10 L-tile and H-tile devices are final. • Edited the description for the `tx_trs` signal in the *Transmitter Protocol Signals* section. Removed the line "For use in LN, CRC, or payload ID insertion". This signal is always required for all 6G-SDI and 12G-SDI designs. • Edited the description for the `tx_datain_valid` and `tx_datain_valid_b` signals in the *Transmitter Protocol Signals* section. This signal can be driven by user logic or by the `tx_dataout_valid_b` signal. • Edited the description for the `rx_coreclk` signal in the *Core Resets and Clocks* section. Added information that this clock source must be stable and there are no required relationships with any other clocks. The clock source can be asynchronous or synchronous to any transceiver's clock. • Added steps for implementing TX PLL and reference clock switching in the *Dynamic TX Clock Switching for Arria V, Cyclone V, and Stratix V Devices* section.
2018.09.24	18.1	• Revised the resource utilization data information for version 18.1. • Added guidelines about how to use duplex mode with the multi-rate and triple-rate presets for Intel Arria 10 devices in the *Handling Transceiver in Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 Devices* section. • Added the *Unconstrained Clocks in SDI Multi-Rate RX Preset Using Intel Arria 10 and Intel Cyclone 10 GX Devices* section under *SDI II IP Core Design Considerations* chapter. • Renamed *Potential Routability Problem During Fitter Stage in Intel Arria 10 and Intel Cyclone 10 GX Devices* to *Potential Routing Problem During Fitter Stage in Intel Arria 10 and Intel Cyclone 10 GX Devices*.
2018.05.07	18.0	• Renamed Intel FPGA SDI II IP core to SDI II Intel FPGA IP core as part of standardizing and rebranding exercise. • Renamed hard transceiver to Native PHY IP for better clarity. • Added support for Intel Cyclone 10 GX device. • Added support for Xcelium Parallel simulator. • Revised the resource utilization data information for version 18.0. • Added `<variation name>.qsys` and `<variation name>.ip` in the generated files list in the *SDI II Intel FPGA IP Core Component Files* section. • Added new parameter, **Rx core clock (rx_coreclk) frequency**. This parameter is available only when you select **Multi rate (up to 12G)** and **Receiver** or **Bidirectional** direction in the Intel Quartus Prime Pro Edition software.

continued...

Document Version	Intel Quartus Prime Version	Changes
		• Updated the description for the `rx_coreclk` signal. You can select either 148.5/148.35 MHz or 297.0/296.70 MHz for **Multi rate (up to 12G)** mode using Intel Arria 10, Intel Cyclone 10 GX, and Intel Stratix 10 devices. • Updated the description for the `rx_coreclk_is_ntsc_paln` signal to add 297.0 MHz and 296.70 MHz options. • Updated the description for the `tx_ln` and `tx_ln_b` signals to include that for Payload ID insertion, these signals must be driven with valid values. • Updated the description for the `tx_line_f0` and `tx_line_f1` signals to include that the line number must be valid and cannot be set to 0. • Edited the *Triple Rate Transmit Clocking Scheme* timing diagram in the *Clock Enable Generator* section. The valid signal for the SD-SDI standard should deassert at the rising edge of the second clock cycle and not at the falling edge of the first clock cycle. • Renamed *Potential Routability Issue During Fitter Stage in Intel Arria 10 Devices* topic to *Potential Routability Problem During Fitter Stage in Intel Arria 10 and Intel Cyclone 10 GX Devices*. Potential routability problem affects Intel Cyclone 10 GX devices too.

Date	Version	Changes
November 2017	2017.11.06	• Renamed SDI II IP core to Intel FPGA SDI II as per Intel rebranding. • Changed the term Qsys to Platform Designer • Added preliminary support for Intel Stratix 10 (H-Tile) devices. • Revised the resource utilization data information for version 17.1. • Added guidelines on how to change the RX CDR reference clock value for higher clock frequencies. • Added information about Intel Stratix 10 in the *Intel FPGA SDI II IP Core Parameters* and *Intel FPGA SDI II IP Core Signals* sections. • Moved information about the Intel FPGA SDI II design example parameters to the respective design example user guides.
May 2017	2017.05.08	• Rebranded as Intel. • Revised the resource utilization data and added recommended speed grades information for version 17.0. • Clarified the description for the `tx_trs` signal. The first word of both EAV and SAV TRSs could mean two `tx_pclk` cycles or one `tx_pclk` cycle depending on the mode selected. • Added an example of 16-bit `rx_format` for 6G-SDI and 12G-SDI interfaces. • Added additional information about the overwrite Payload ID feature. • Edited the multi-rate (up to 12G-SDI) transmitter and receiver data path block diagrams to include the sync bit insertion and removal blocks. • Updated the SMPTE standards to the latest naming convention. • Added a note in the *Transceiver Reconfiguration Controller* section that the transceiver reconfiguration controller only reconfigures the TX transceiver if you are performing TX clock switching.

continued...

Date	Version	Changes
December 2016	2016.12.20	• Added detailed description for `tx_datain` and `rx_dataout` signals about 6G-SDI and 12G-SDI interfaces. • Added information about image mapping for 6G-SDI and 12G-SDI interfaces. • Added information for `rx_dataout_valid` signal that the 1H4L 1H5L cadence for SD-SDI repeats indefinitely in an ideal case but in a typical scenario the cadence shift periodically (for instance, 1H4L 1H5L 1H5L 1H4L). • Updated `rx_format` information to include that for 6G-SDI or 12G-SDI interfaces, each of the 20-bit interface reports its own detected format. • Added information for `pll_powerdown_in` signal that sharing Tx PLLs for designs that also implement dynamic reconfiguration require `XCVR_TX_PLL_RECONFIG_GROUP` QSF assignment.
October 2016	2016.10.31	• Restructured the chapters. • Added information for the new **Design Example** parameters. • Removed all Arria 10 design example related information. For more information about Arria 10 design examples, refer to the *SDI II IP Core Design Example User Guide*. • Added clocking diagrams for Arria 10 devices and the V series devices—Arria V, Cyclone V, and Stratix V. • Added guideline to overcome potential routability issue during Fitter stage.
May 2016	2016.05.02	• Added new option, fPLL, for the Arria 10 TX PLL parameter and removed the ATX PLL option. • Added estimated run-time settings for the different SDI II video standards. • Added guideline for transceiver handling. The transceiver handling guidelines differ for Arria 10 devices and the V series devices—Arria V, Cyclone V, and Stratix V. • Added new transceiver signals: — `rx_analogreset_ack` — `tx_analogreset_ack` — `rx_cal_busy` — `pll_powerdown` — `xcvr_rxclk` — `xcvr_rxclk_b` — `rst_tx_phy` • Added a new receiver signals: `rx_datain` and `rx_datain_valid`. • Removed these signals: `rx_pll_locked` and `rx_pll_locked_b`. These signals are redundant and no longer required after the switch to Native PHY. • Updated the design example directory. • Added links to archived versions of the *SDI II IP Core User Guide*.
November 2015	2015.11.02	• Added information that the `rx_format` signal for each stream reports its own detected format for 6G-SDI and 12G-SDI interfaces. • Added information about 3 new interface signals for Arria V, Cyclone V, Stratix V devices: `rx_trs_in`, `pll_powerdown_in`, and `pll_powerdown_out` • Added reconfiguration management parameters for Arria 10 devices: `VIDEO_STANDARD`, `ED_TXPLL_SWITCH`, and `XCVR_RCFG_IF_TYPE`. • Added descriptions for the SDI presets available in the Arria 10 Transceiver Native PHY IP core.
May 2015	2015.05.04	• Changed the resource utilization table to include data for each SDI standard and updated the data for version 15.0. • Added new multi-rate data path block diagrams for transmitter and receiver. • Added new information about inserting sync bits.

continued...

Date	Version	Changes
		• Renamed the term *video payload ID (VPID)* to *payload ID* as per SMPTE specification. • Renamed *Level A* to *HD-SDI dual link* and *Level B* to *3G-SDI (level B)*. • Updated the following new parameter options: — Added new video standard **Multi rate (up to 12G)** for Arria 10 devices. — Added TX PLL reference clock switching option for **Dynamic Tx clock switching** parameter. • Added a note for the interface signals to indicate that multi-rate (up to 12G) mode requires 4 streams and the rest require one stream. • Added a new parameter for Reconfiguration Management: XCVR_TX_PLL_SEL. • Added information for multi standard support including 6G-SDI and 12G-SDI. • Added the multi standard (including 6G-SDI and 12G-SDI) information for the following signals: — `tx_enable_ln` — `tx_std` — `tx_datain` — `tx_datain_valid` — `tx_ln_b` — `tx_dataout` — `tx_dataout_valid` — `tx_vpid_byte(1-4)_b` — `rx_std` — `rx_dataout_valid` — `rx_format` — `rx_ln_b` — `rx_vpid_byte(1-4)_b` — `rx_vpid_checksum_error_b` • Added information that the following signals are not applicable for Arria 10 devices: — `rx_coreclk_hd` — `rx_clkin` — `rx_clkin_b` — `rx_rst_proto_in` — `rx_rst_proto_in_b`
January 2015	2015.01.23	• Updated the resource utilization table for version 14.1. • Changed the names of the following parameters for receiver options: — **Convert Level A to Level B (SMPTE 372M)** changed to **Convert HD-SDI dual link to 3G-SDI (level B)**. — **Convert Level B to Level A (SMPTE 372M)** changed to **Convert 3G-SDI (level B) to HD-SDI dual link**. • Edited information about `rx_format` signal, which now reports video transport format instead of picture format. The signal reports 3G Level A RGB or YCbCr 4:4:4 format.

continued...