E-tile Hard IP User Guide

E-Tile Hard IP for Ethernet and E-Tile CPRI PHY Intel® FPGA IPs

Updated for Intel® Quartus® Prime Design Suite: **21.2**

Online Version
Send Feedback

UG-20160

ID: 683468
Version: 2021.08.04

Contents

1. About E-tile Hard IP User Guide

This user guide consists of information for the following IP cores:

- E-Tile Hard IP for Ethernet Intel® FPGA IP (Intel Stratix® 10)
- E-Tile Ethernet IP for Intel Agilex™ FPGA
- E-Tile CPRI PHY Intel FPGA IP

For more information on specific IP release, refer to the *Release Information* sections.

Related Information

- E-Tile Hard IP for Ethernet Intel FPGA IP Core Release Information on page 20
- E-Tile Ethernet IP for Intel Agilex FPGA Core Release Information on page 20
- E-Tile CPRI PHY IP Core Release Information on page 285
- E-tile Hard IP Intel Stratix 10 Design Examples User Guide
  Describes the Ethernet, CPRI PHY, and Dynamic Reconfiguration design example generation, simulation, compilation, and testing for Intel Stratix 10 devices.
- E-tile Hard IP Intel Agilex Design Examples User Guide
  Describes the Ethernet, CPRI PHY, and Dynamic Reconfiguration design example generation, simulation, compilation, and testing for Intel Agilex devices.
- E-Tile Transceiver PHY User Guide

2. About the E-Tile Hard IP for Ethernet Intel FPGA IP Core

Intel Stratix 10 and Intel Agilex E-tile FPGA production devices include a configurable, hardened protocol stack for Ethernet that is compatible with the *IEEE 802.3 High Speed Ethernet Standard* and the *25G/50G Ethernet Specification, Draft 1.6* from the 25 Gigabit Ethernet Consortium.

Table 1. Ethernet IP Naming Convention

The table shows the Ethernet-based IPs available in IP Catalog.

Supported Device Family	IP Catalog
Intel Stratix 10	E-Tile Hard IP for Ethernet Intel FPGA IP
Intel Agilex	E-Tile Ethernet IP for Intel Agilex FPGA

Note: Unless specified, the E-Tile Hard IP for Ethernet Intel FPGA IP applies to all supported device families.

The E-Tile Hard IP for Ethernet Intel FPGA IP provides access to this hard IP at Ethernet data rates of 10 Gbps, 25 Gbps, and 100 Gbps. The IP core is included in the Intel FPGA IP Library and is available from the Intel Quartus® Prime Pro Edition IP Catalog.

The IP core is available in the following variants, each providing a different combination of Ethernet channels and features:

- Single 10GE/25GE channel
- 1 to 4 10GE/25GE channels with optional Reed-Solomon Forward Error Correction (RS-FEC)
- 100GE channel with optional RS-FEC
- 100GE or 1 to 4 10GE/25GE channels with optional RS-FEC, and optional 1588 Precision Time Protocol (PTP)
- Custom PCS with optional RS-FEC

The 100GE or 1 to 4 10GE/25GE channels with optional RS-FEC, and optional 1588 Precision Time Protocol (PTP) variant contains a 100G Ethernet channel, and up to 4 single-lane channels that can run at 10G or 25G. However, the single-lane channels and the 100GE channel cannot run at the same time.

For any variant except the custom PCS with RS-FEC variant, you can choose a Media Access Control (MAC) + Physical Coding Sublayer(PCS) variation, a PCS-only variation, a custom PCS variation, a Flexible Ethernet (FlexE) variation, or an Optical Transport Network (OTN) variation.

Figure 1. Variant Selection

For any variant, you can choose a MAC + PCS variation, a PCS-only variation, a FlexE variation, an OTN variation or a custom PCS variation.

Ethernet IP Layers	Protocol Layers Included												Variants				
	MAC	Flow Control	IEEE 1588 PTP 1-Step/2-Step	PCS Encoding/Decoding	PCS Scrambling/Descrambling	PCS Striping/Alignment	RS-FEC Transcode/Detranscode	RS-FEC (528,514) Encode/Decode/Correct	RS-FEC (544,514) Encode/Decode/Correct	RS-FEC Stripping/Alignment	PMA	PMD	Single 10GE/25GE Channel	1 to 4 10GE/25GE Channels with optional RSFEC	100G Channels	100GE or 1 to 4 10GE/25GE Channels with optional RS-FEC and PTP	Custom PCS with Optional RSFEC (3)
MAC + PCS	Yes	Yes	—	Yes	Yes	Yes	—	—	—	—	Yes	Yes	Yes	Yes	Yes	Yes	—
MAC + 1588 PTP + PCS	Yes	Yes	Yes	Yes	Yes	Yes	—	—	—	—	Yes	Yes	—	—	—	Yes	—
MAC + PCS + (528,514 RSFEC)	Yes	Yes	—	Yes	Yes	—	Yes	Yes	—	Yes	Yes	Yes	Yes(1)	Yes(1)	Yes	Yes(1)	—
MAC + PCS + (544,514 RSFEC)	Yes	Yes	—	Yes	Yes	—	Yes	—	Yes	Yes	Yes	Yes	—	—	Yes	Yes(2)	—
MAC + 1588 PTP + PCS + (528,514 RSFEC)	Yes	Yes	Yes	Yes	Yes	—	Yes	Yes	—	Yes	Yes	Yes	—	—	—	Yes(1)	—
PCS Only	—	—	—	Yes	Yes	Yes	—	—	—	—	Yes	Yes	Yes	Yes	Yes	Yes	Yes
OTN	—	—	—	—	—	Yes	—	—	—	—	Yes	Yes	Yes	Yes	Yes	Yes	—
FlexE	—	—	—	—	Yes	Yes	—	—	—	—	Yes	Yes	Yes	Yes	Yes	Yes	—
PCS + (528,514 RSFEC)	—	—	—	Yes	Yes	—	Yes	Yes	—	Yes	Yes	Yes	Yes(1)	Yes(1)	Yes	Yes(1)	Yes
PCS + (544,514 RSFEC)	—	—	—	Yes	Yes	—	Yes	—	Yes	Yes	Yes	Yes	—	—	Yes	Yes(2)	—
OTN + (528,514 RSFEC)	—	—	—	—	—	—	Yes	Yes	—	Yes	Yes	Yes	Yes(1)	Yes(1)	Yes	Yes(1)	—
OTN + (544,514 RSFEC)	—	—	—	—	—	—	Yes	—	Yes	Yes	Yes	Yes	—	—	Yes	Yes(2)	—
FlexE + (528,514 RSFEC)	—	—	—	—	Yes	—	Yes	Yes	—	Yes	Yes	Yes	Yes(1)	Yes(1)	Yes	Yes(1)	—
FlexE + (544,514 RSFEC)	—	—	—	—	Yes	—	Yes	—	Yes	Yes	Yes	Yes	—	—	Yes	Yes(2)	—
Custom PCS Only	—	—	—	Yes	Yes	Yes	—	—	—	—	Yes	Yes	—	—	—	—	Yes
Custom PCS + RSFEC	—	—	—	Yes	Yes	—	Yes	Yes	—	Yes	Yes	Yes	—	—	—	—	Yes

(1) 10G data rate does not support RSFEC.
(2) Only 100G data rate supports (544,514) RSFEC.
(3) Customizable data rate PCS from 2.5 to 28 Gbps for protocols other than Ethernet..

Table 2. Client Interfaces for IP Core Variations

IP Core Variation	Client Interface Type
MAC+PCS	Avalon® Streaming (Avalon-ST)(1)
PCS_Only	Media Independent Interface (MII)
Custom PCS	MII
FlexE	PCS66
OTN	PCS66

Note: The E-Tile Hard IP for Ethernet Intel FPGA IP provides support for the OTN feature. For further inquiries, contact your nearest Intel sales representative.

(1) The E-Tile Hard IP for Ethernet Intel FPGA IP MAC interface uses a modified AVST interface. For recommended usage of the TX MAC interface, refer to the Figure 37 on page 124.

E-Tile Hard IP for Ethernet Intel FPGA IP core supports a variety of protocol implementations.

Table 3. Ethernet Protocols

Ethernet Channel	Protocol	Number of Lanes and Line Rate
100GE	100GBASE-KR4	4x25.78125 Gbps Non-Return-to-Zero (NRZ) lanes for Copper Backplane
	100GBASE-CR4	4x25.78125 Gbps NRZ lanes for Direct Attach Copper Cable
	CAUI-4	4x25.78125 Gbps NRZ lanes for Low Loss Links: Chip-to-Chip or Chip-to-Module
	100GBASE-KR2	2x53.125 Gbps PAM4 lanes for Copper Backplane
	100GBASE-CR2	2x53.125 Gbps PAM4 lanes for Direct Attach Copper Cable
	CAUI-2	2x53.125 Gbps PAM4 lanes for Low Loss Links: Chip-to-Chip, Chip-to-Module
25GE	25GBASE-KR	1x25.78125 Gbps NRZ lane for Copper Backplane
	25GBASE-CR	1x25.78125 Gbps NRZ lane for Direct Attach Copper Cable
	25GBASE-R AUI	1x25.78125 Gbps NRZ lane for Low Loss Connections to External PHY Modules
	25GBASE-R Consortium Link	1x25.78125 Gbps NRZ lane based on the *25G/50G Consortium Specification*
10GE	10GBASE-KR	1x10.3125 Gbps NRZ lane for Copper Backplane
	10GBASE-CR	1x10.3125 Gbps NRZ lane for Direct Attach Copper Cable
	XAUI	1x10.3125 Gbps NRZ lane for Low Loss Connections to External PHY Modules

Related Information

- IEEE Website
  The IEEE 802.3-2015 High Speed Ethernet Standard is available on the IEEE website.
- 25G Ethernet Consortium
- Knowledge Data Base (KDB)
  Provides links to applicable articles for a variety of FPGA related issues.
- E-tile Hard IP Intel Stratix 10 Design Examples User Guide
  Describes the Ethernet, CPRI PHY, and Dynamic Reconfiguration design example generation, simulation, compilation, and testing for Intel Stratix 10 devices.
- E-tile Hard IP Intel Agilex Design Examples User Guide
  Describes the Ethernet, CPRI PHY, and Dynamic Reconfiguration design example generation, simulation, compilation, and testing for Intel Agilex devices.
- E-Tile Transceiver PHY User Guide

2.1. E-Tile Hard IP for Ethernet Intel FPGA IP Supported Features

The IP core is designed to the *IEEE 802.3-2015 High Speed Ethernet Standard* available on the IEEE website (www.ieee.org) and the *25G/50G Ethernet Specification, Draft 1.6* available from the 25 Gigabit Ethernet Consortium. The MAC provides cut-

through frame processing to optimize latency, and supports full wire line speed with a 64-byte frame length and back-to-back or mixed length traffic with no dropped packets. All E-Tile Hard IP for Ethernet Intel FPGA IP variations are in full-duplex mode.

Table 4. E-Tile Hard IP for Ethernet Intel FPGA IP Features

Features	Description
PCS	Hard IP logic that interfaces seamlessly to E-tile transceivers.
	CAUI external interface consisting of four transceiver lanes operating at 25.78125 Gbps.
	CAUI-2 external interface with two transceiver lanes operating at 53.125 Gbps with PAM4 encoding
	25G AUI external interface with 1 transceiver lane operating at 25.78125 Gbps
	10G AUI external interface with 1 transceiver lane operating at 10.3125 Gbps
	Supports CAUI-4 links based on 64B/66B encoding with data striping and alignment markers to align data from multiple lanes.
	Supports customizable data rate PCS from 2.5 to 28 Gbps for protocols other than Ethernet.
	Optional RS-FEC(528,514) or RS-FEC(544,514) for 25G and 100G variations.
	Supports 10G, 25G, and 100G variations. • Auto-negotiation (AN) as defined in *IEEE Standard 802.3-2915 Clause 73* and the *25G Ethernet Consortium Schedule Draft 1.6*, and • Link training (LT) as defined in *IEEE Standard 802.3-2915 Clauses 92 and 93* and the *25G Ethernet Consortium Schedule Draft 1.6*
	RX Skew Variation tolerance that exceeds the *IEEE 802.3-2015 High Speed Ethernet Standard* Clause 80.5 requirements.
OTN	Optional 25GE constant bit rate (CBR); with TX and RX PCS66 bit encoding/decoding and scrambling/descrambling disabled. *Note:* The E-Tile Hard IP for Ethernet Intel FPGA IP provides support for the OTN feature. For further inquiries, contact your nearest Intel sales representative.
	Optional RS-FEC(528,514) or RS-FEC(544,514) for 25G and 100G variations.
Flexible Ethernet (FlexE)	Optional CBR; with TX and RX PCS66 bit encoding/decoding disabled and scrambling/descrambling enabled.
	Optional RS-FEC(528,514) or RS-FEC(544,514) for 25G and 100G variations.
PMA Direct Mode	Optional to switch from MAC+PCS to PMA only mode during run-time.
Frame Structure Control	Support for jumbo packets.
	RX CRC pass-through control.
	1000 bits RX PCS lane skew tolerance for 100G links, which exceeds the *IEEE 802.3-2015 High Speed Ethernet Standard* Clause 82.2.12 requirements.
	Optional per-packet TX CRC generation and insertion.
	Optional Deficit Idle Counter (DIC) options to maintain a finely controlled 8-byte, 10-byte, or 12-byte inter-packet gap (IPG) minimum average, or allow the user to drive the IPG from the client interface
	RX and TX preamble pass-through options for applications that require proprietary user management information transfer.
	Optional TX MAC source address insertion.

continued...

Features	Description
	TX automatic frame padding to meet the 64-byte minimum Ethernet frame length on the Ethernet link. Optional per-packet disabling of this feature.
	TX error insertion capability supports client invalidation of in-progress input to TX client interface.
Frame Monitoring and Statistics	RX CRC checking and error reporting.
	Optional RX strict Start Frame Delimiter (SFD) checking per IEEE specification.
	Optional RX strict preamble checking per IEEE specification.
	RX malformed packet checking per IEEE specification.
	Received control frame type indication.
	Statistics counters.
	Snapshot feature for precisely timed capture of statistics counter values.
	Optional fault signaling: detects and reports local fault and generates remote fault, with support for unidirectional link fault as defined in *IEEE 802.3-2015 High Speed Ethernet Standard* Clause 66.
Flow Control	Optional *IEEE 802.3-2015 Ethernet Standard* Clause 31 Ethernet flow control operation using the pause registers or pause interface.
	Optional priority-based flow control that complies with the *IEEE Standard 802.1Q-2014 —Amendment 17: Priority-based Flow Control*.
	Pause frame filtering control.
	Software can dynamically toggle local TX MAC data flow to support selective input flow cut-off.
Precision Time Protocol (PTP)	Optional support for the IEEE Standard 1588-2008 Precision Clock Synchronization Protocol (1588 PTP) (1588v2).
	1-step (1588v1 and 1588v2) and 2-step TX (1588v2) timestamps.
	Support for PTP headers in a variety of frame formats, including Ethernet encapsulated, UDP in IPv4, and UDP in IPv6.
	Support for Checksum Zero and Checksum extension byte calculations.
	Support for Correction field operations.
	Programmable extra latency.
Debug and testability	Optional serial PMA loopback (TX to RX) at the serial transceiver for self-diagnostic testing.
	Optional parallel loopback (TX to RX) at the MAC or at the PCS for self-diagnostic testing.
	Bit-interleaved parity error counters to monitor bit errors per PCS lane.
	RX PCS error block counters to monitor errors during and between frames.
	Malformed and dropped packet counters.
	High BER detection to monitor link bit error rates over all PCS lanes.
	Optional scrambled Idle test pattern generation and checking.
	Snapshot feature for precisely timed capture of statistics counter values.
	TX error insertion capability supports test and debug.
	Support for Ethernet Link Inspector (ELI) tool to monitor an Ethernet link.

continued...

Features	Description
	Support for Ethernet Tool Kit to monitor an Ethernet link.
User System Interface	Avalon memory-mapped interface (Avalon-MM) to access the IP core control and status registers.
	Avalon streaming interface (Avalon-ST) connects the MAC to client logic with the start of frame in the most significant byte (MSB) in MAC+PCS variations. Interface for 100G channel has 512 bits; the 10/25G channels use 64 bits when the MAC layer is enabled.
	MII data path interface connects the PCS to client logic in PCS-only variations. Interface for 100G variants has 256 bits of data and 32 bits of control; interface for 10G/25G variants has 64 bits of data and 8 bits of control.
	Hardware and software reset control.
	Supports Synchronous Ethernet (SyncE) by providing a CDR recovered clock output signal to the device fabric.
	Supports external source clock for EMIB interface for applications that requires switching transceiver line rate.

For a detailed specification of the Ethernet protocol refer to the *IEEE 802.3-2015 High Speed Ethernet Standard*.

Related Information

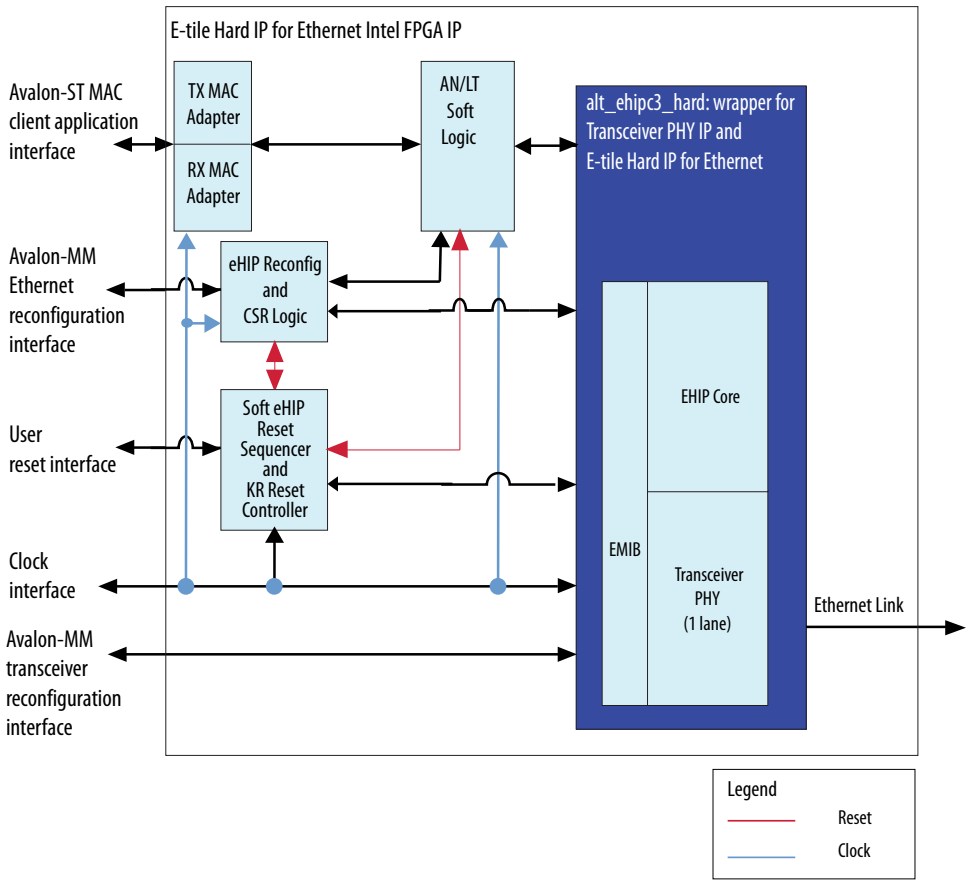
- IEEE Website
  The IEEE 802.3-2015 High Speed Ethernet Standard is available on the IEEE website.
- 25G Ethernet Consortium

2.2. E-Tile Hard IP for Ethernet Intel FPGA IP Overview

The E-Tile Hard IP for Ethernet Intel FPGA IP block diagrams show the main blocks, and internal and external connections for each variant.

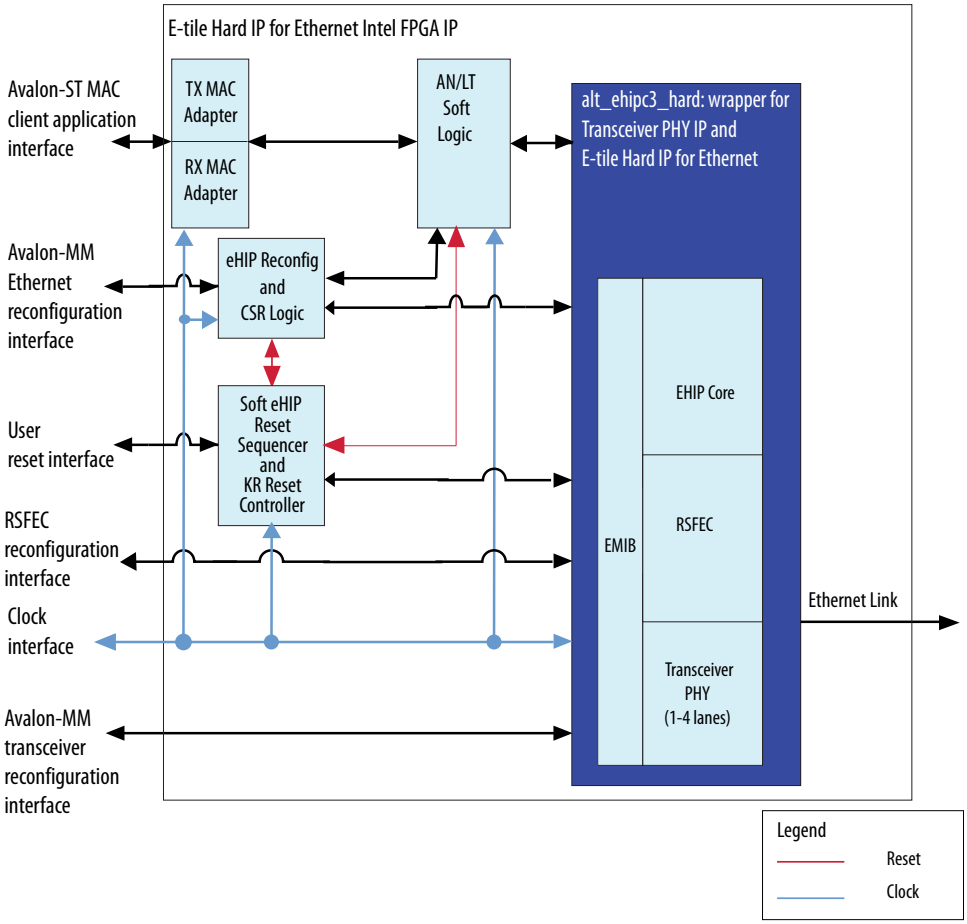
For these block diagrams, the reconfiguration and soft reset sequencer implement the reconfiguration interfaces and resets for the core, respectively. The auto-negotiation and link training (AN/LT) soft logic is only inserted when you select **Enable AN/LT**.

Figure 2. Single 10G/25G Channel



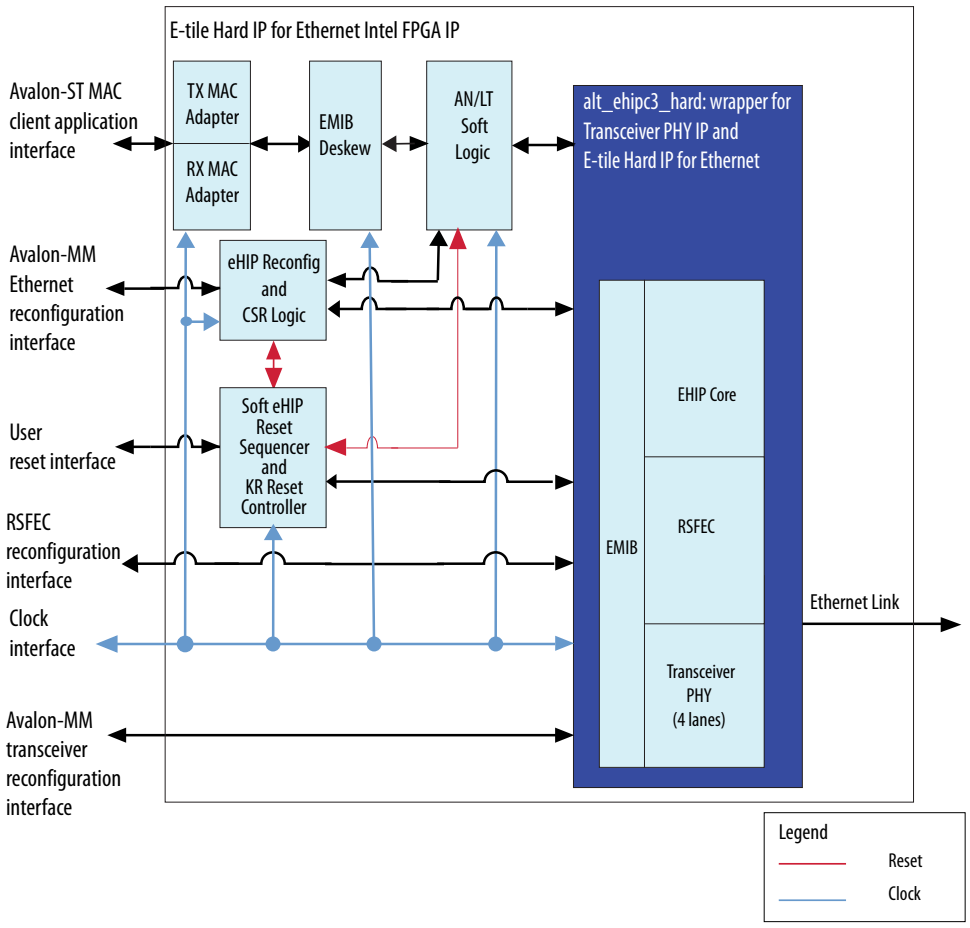
This variant supports only single channel 10G/25G Ethernet without RS-FEC and PTP features.

Figure 3. 1 to 4 10G/25G Channels with Optional RS-FEC



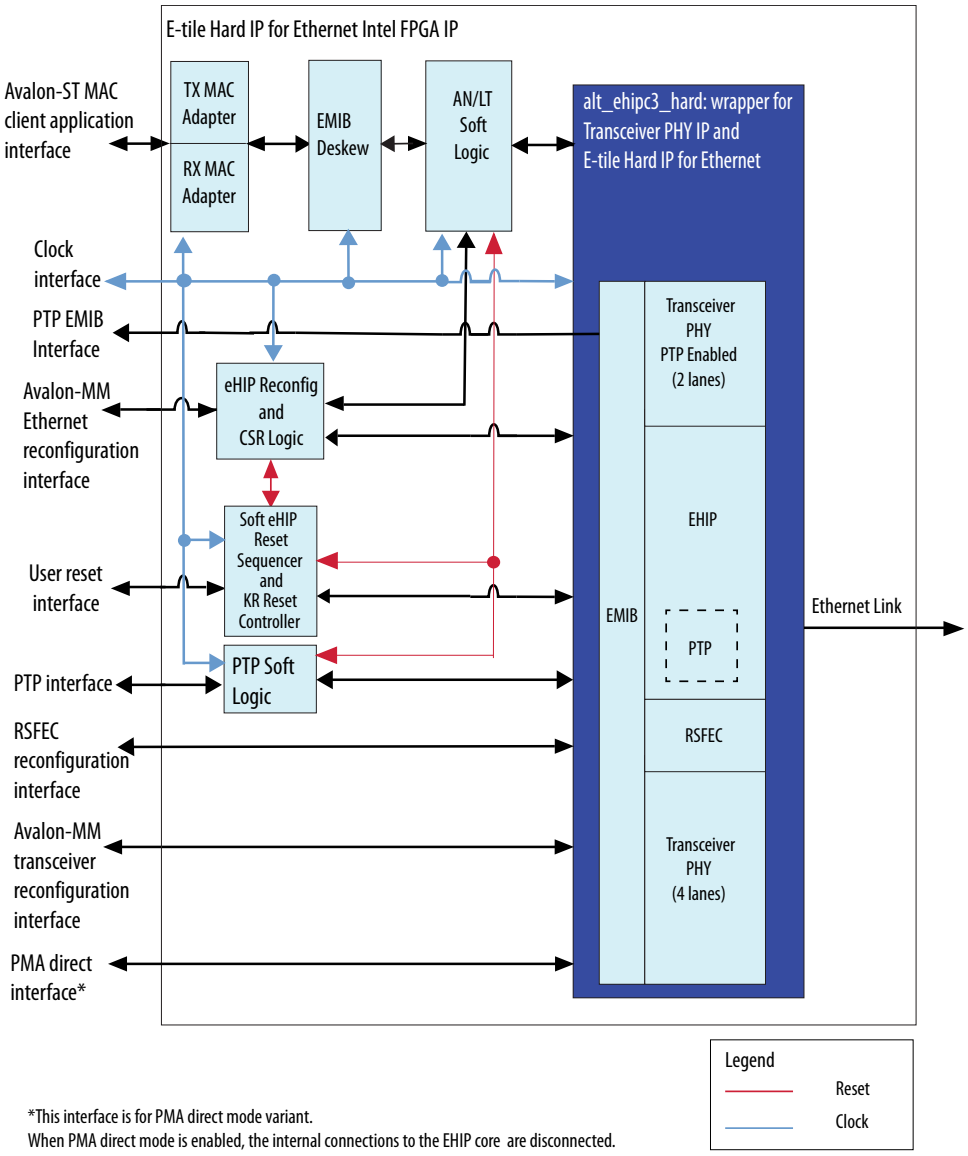
- This variant of the IP core includes up to four channels. Each channel has its own set of adapters, reconfiguration logic, AN/LT and reset sequencer logic.
- RS-FEC is optional for this variant.

Figure 4. 100G Channel with Optional RS-FEC



- This variant of the IP core includes a single 100G channel that has its own set of adapters, reconfiguration logic, AN/LT and reset sequencer logic.
- The deskew logic corrects for possible skew over the EMIB interfaces between the main die and the E-tile.
- RS-FEC is optional for this variant. You can select RS-FEC(528,514) or RS-FEC(544,514) for this variant.

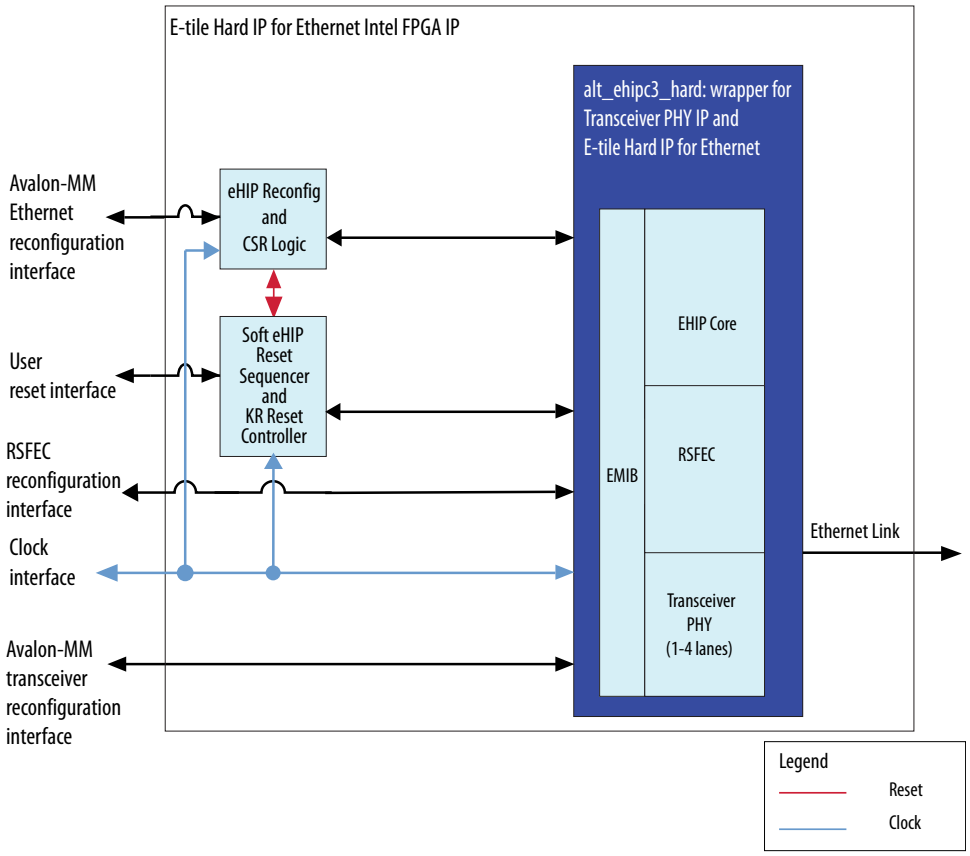
Figure 5. 100G Channel with 1 to 4 10G/25G Channels, RS-FEC, and Precision Time Protocol (PTP)



- This variant of the IP core includes a 100G channel or between 1 to 4 10G/25G channels.
- Because the 100G channel uses the same transceivers as the 10G/25G channels, you cannot use the 100G channel when any of the 10G/25G channels are running. You can switch the reconfiguration interfaces on the core between channels at run time.
- For this variant, each channel has its own set of adapters, reconfiguration logic, AN/LT and reset sequencer logic.

- The deskew logic corrects for possible skew over the EMIB interfaces between the main die and the E-tile. The EMIB deskew logic is always used for 100G channels. For 10G/25G channels, the deskew logic is used when you enable PTP, to ensure the PTP commands to each channel are synchronized to data. The deskew logic is not required for single 10G/25G channel.
- The PTP soft component logic block provides the user PTP interface, performs the soft logic operations required for the E-tile timestamp system, and interacts with the TOD module (the Time-of-Day clock) that you provide.
- RS-FEC and PTP are optional for this variant. This variant only supports RS-FEC(528,514) with PTP enabled.

Figure 6. Custom PCS with Optional RS-FEC



- This variant of IP core supports customizable line rate PCS up to four channels. Each channel has its own set of adapters, reconfiguration logic, and reset sequencer logic. This variant does not include an Ethernet MAC.
- This variant supports transceiver line rates ranges from 2.5 to 28 Gbps used in other protocols.
- RS-FEC is optional for this variant and it supports RS-FEC for Ethernet and Fibre Channel modes.

2.3. IP Core Device Family and Speed Grade Support

The following sections list the device family and device speed grade support offered by the E-Tile Hard IP for Ethernet Intel FPGA:

E-Tile Hard IP for Ethernet Intel FPGA IP Device Family Support on page 17
E-Tile Hard IP for Ethernet Intel FPGA IP Device Speed Grade Support on page 17

2.3.1. E-Tile Hard IP for Ethernet Intel FPGA IP Device Family Support

Table 5. Intel FPGA IP Core Device Support Levels

Device Support Level	Definition
Advance	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 (datapath width, burst depth, I/O standards tradeoffs).
Preliminary	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	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.

Table 6. E-Tile Hard IP for Ethernet Intel FPGA IP Device Family Support

Shows the level of support offered by the E-Tile Hard IP for Ethernet Intel FPGA IP for each Intel FPGA device family.

Device Family	Support
Intel Stratix 10	Final
Intel Agilex	Preliminary

Related Information

Timing and Power Models
Reports the default device support levels in the current version of the Intel Quartus Prime Pro Edition software.

2.3.2. E-Tile Hard IP for Ethernet Intel FPGA IP Device Speed Grade Support

The E-Tile Hard IP for Ethernet Intel FPGA IP supports the following speed grades for Intel Stratix 10 and Intel Agilex E-tile devices:

- Transceiver speed grade: -1, -2 or -3(2)
- Core speed grade: -1 or -2

For information about the applicable device speed grades based on the target data rates, refer to the *Device Data Sheet*.

(2) This speed grade only supports the 10G data rate.

Related Information

- Intel Stratix 10 Device Datasheet
- Intel Agilex Device Data Sheet

2.4. IP Core Verification

To ensure functional correctness of the E-Tile Hard IP for Ethernet Intel FPGA IP, Intel performs extensive validation through both simulation and hardware testing. Before releasing a version of the E-Tile Hard IP for Ethernet Intel FPGA IP, Intel runs comprehensive regression tests in the current version of the Intel Quartus Prime Pro Edition software.

Intel verifies that the current version of the Intel Quartus Prime Pro Edition software compiles the previous version of each IP core. Intel does not verify compilation with IP core versions older than the previous release.

2.4.1. Simulation Environment

Intel performs the following tests on the E-Tile Hard IP for Ethernet Intel FPGA IP in the simulation environment using internal and third party standard bus functional models (BFM):

- Constrained random tests that cover randomized frame size and contents
- Randomized error injection tests that inject Frame Check Sequence (FCS) field errors, runt packets, and corrupt control characters, and then check for the proper response from the IP core
- Assertion based tests to confirm proper behavior of the IP core with respect to the specification
- Extensive coverage of our runtime configuration space and proper behavior in all possible modes of operation

2.4.2. Compilation Checking

Intel performs compilation testing on an extensive set of E-Tile Hard IP for Ethernet Intel FPGA IP variations and designs that target different devices, to ensure the Intel Quartus Prime Pro Edition software places and routes the IP core ports correctly.

2.4.3. Hardware Testing

Intel performs hardware testing of the key functions of the E-Tile Hard IP for Ethernet Intel FPGA IP on available FPGA devices using standard 10/25, and 100 Gbps Ethernet network test equipment and optical modules. The Intel hardware tests of the E-Tile Hard IP for Ethernet Intel FPGA IP also ensure reliable solution coverage for hardware related areas such as performance, link synchronization, and reset recovery.

2.5. Resource Utilization

The resources for the E-Tile Hard IP for Ethernet Intel FPGA IP were obtained from the Intel Quartus Prime Pro Edition software version 19.1

Table 7. Resource Utilization for Selected Variations

Ethernet Rate	IP Core Variation	ALMs	Dedicated Logic Registers	Memory 20K
10G	MAC+PCS	2,100	2,900	6
	MAC+PCS with IEEE 1588/PTP	5,500	11,700	11
	PCS Only	1,918	2,484	4
	OTN	1,936	2,505	4
	FlexE	1,950	5,539	4
25G	MAC+PCS	2,100	2,900	6
	MAC+PCS with RS-FEC	2,400	3,400	6
	MAC+PCS IEEE 1588/PTP	5,500	11,700	11
	MAC+PCS with RS-FEC and IEEE 1588/PTP	5,500	11,700	11
	PCS Only	1,929	2,486	4
	PCS with RS-FEC	2,308	3,073	4
	OTN	1,922	2,537	4
	OTN with RS-FEC	2,292	3,064	4
	FlexE	1,915	2,475	4
	FlexE with RS-FEC	2,281	3,057	4
100G	MAC+PCS	5,777	8,443	6
	MAC + 1588PTP + PCS	14,966	28,687	9
	MAC+PCS with (528,514)RS-FEC	6,016	8,739	6
	MAC+PCS with (528,514)RS-FEC and IEEE 1588/PTP	9,147	14,234	21
	MAC+PCS with (544,514) RS-FEC	6,029	8,827	6
	PCS Only	2,412	2,913	4
	PCS with (528,514) RS-FEC	2,668	3,217	4
	PCS with (544,514) RS-FEC	2,682	3,251	4
	OTN	2,401	2,905	4
	OTN with (528,514) RS-FEC	2,647	3,178	4
	OTN with (544,514) RS-FEC	2,648	3,200	4
	FlexE	2,400	2,929	4

continued...

Ethernet Rate	IP Core Variation	ALMs	Dedicated Logic Registers	Memory 20K
	FlexE with (528,514)RS-FEC	2,645	3,178	4
	FlexE with (544,514) RS-FEC	2,649	3,232	4
	KR with AN/LT	10,362	15,843	28
	KR FEC with AN/LT	10,386	15,542	28

Note: The E-Tile Hard IP for Ethernet Intel FPGA IP provides support for the OTN feature. For further inquiries, contact your nearest Intel sales representative.

Table 8. Resource Utilization for Custom PCS Variations

Data Rate	Variant	ALMs	Dedicated Logic Registers	Memory 20K
10G	RS-FEC disabled	853	1,228	0
24G	RS-FEC enabled	856	1,114	0
	RS-FEC disabled	824	1,168	0

2.6. Release Information

IP versions are the same as the Intel Quartus Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme. If an IP core version is not listed, the user guide for the previous IP core version applies.

The IP versioning scheme (X.Y.Z) number changes from one software version to another. A change in:

- X indicates a major revision of the IP. If you update your Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 9. E-Tile Hard IP for Ethernet Intel FPGA IP Core Release Information

Item	Description
IP Version	20.2.1
Intel Quartus Prime Version	21.2
Release Date	2021.06.23
Ordering Code	IP-ETH-ETILEHIP

Table 10. E-Tile Ethernet IP for Intel Agilex FPGA Core Release Information

Item	Description
IP Version	20.2.0
Intel Quartus Prime Version	21.2
Release Date	2021.06.23
Ordering Code	IP-ETH-ETILEHIP

Related Information

E-Tile Hard IP Release Notes
The IP Release Notes list IP changes in a particular release.

2.7. Getting Started

The following sections explain how to install, parameterize, simulate, and initialize the IP:

Installing and Licensing Intel FPGA IP Cores on page 21

Specifying the IP Core Parameters and Options on page 24

Generated File Structure on page 25

Integrating Your IP Core in Your Design on page 27

IP Core Testbenches on page 52

Compiling the Full Design on page 53

Related Information

- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Qsys Simulation Scripts
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.

2.7.1. Installing and Licensing Intel FPGA IP Cores

The Intel Quartus Prime Pro Edition 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 7. IP Core Installation Path

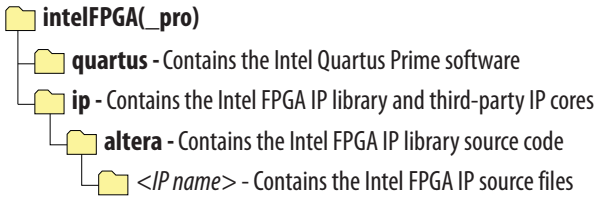


Table 11. IP Core Installation Locations

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

2.7.1.1. 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.

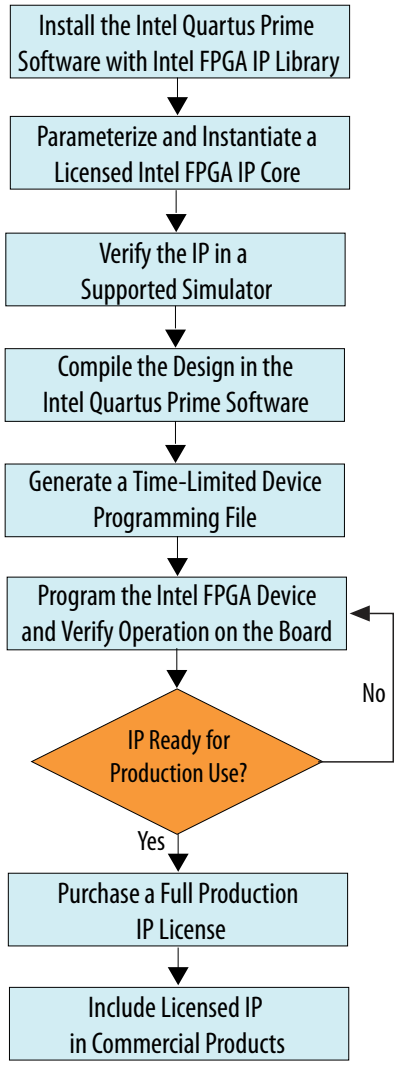
Intel FPGA IP Evaluation Mode supports the following operation modes:

- **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 8. 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 FPGA Licensing Support Center
- Introduction to Intel FPGA Software Installation and Licensing

2.7.2. Specifying the IP Core Parameters and Options

The E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor allows you to quickly configure your custom IP variation. Use the following steps to specify IP core options and parameters in the Intel Quartus Prime Pro Edition software.

1. If you do not already have an Intel Quartus Prime Pro Edition project in which to integrate your E-Tile Hard IP for Ethernet Intel FPGA IP, you must create one.
   a. In the Intel Quartus Prime Pro Edition, click **File ➤ New Project Wizard** to create a new Quartus Prime project, or **File ➤ Open Project** to open an existing Quartus Prime project. The wizard prompts you to specify a device.
   b. Specify the device family and select a production E-tile device that meets the speed grade requirements for the IP core.
   c. Click **Finish**.
2. In the IP Catalog, locate and select **E-tile Hard IP for Ethernet Intel FPGA IP**. The **New IP Variation** window appears.
3. Specify a top-level name for your new custom IP variation. The parameter editor saves the IP variation settings in a file named *<your_ip>*`.ip`.
4. Click **OK**. The parameter editor appears.
5. Specify the parameters for your IP core variation. Refer to Parameter Editor Parameters on page 53 for information about specific IP core parameters.
6. Optionally, to generate a simulation testbench or compilation and hardware design example, follow the instructions in the *E-tile Hard IP for Ethernet Design Example User Guide*.
7. Click **Generate HDL**. The **Generation** dialog box appears.
8. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
9. Click **Close**. The parameter editor adds the top-level `.ip` file to the current project automatically. If you are prompted to manually add the `.ip` file to the project, click **Project ➤ Add/Remove Files in Project** to add the file.
10. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports and set any appropriate per-instance RTL parameters.

Related Information

- E-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
- E-Tile Hard IP for Ethernet Intel Agilex FPGA IP Design Example User Guide

2.7.3. Generated File Structure

The Intel Quartus Prime Pro Edition software generates the following IP core output file structure.

For information about the file structure of the design example, refer to the *E-Tile Hard IP for Ethernet Intel FPGA IP Design Example User Guide*.

Figure 9. E-Tile Hard IP for Ethernet Intel FPGA IP Generated Files

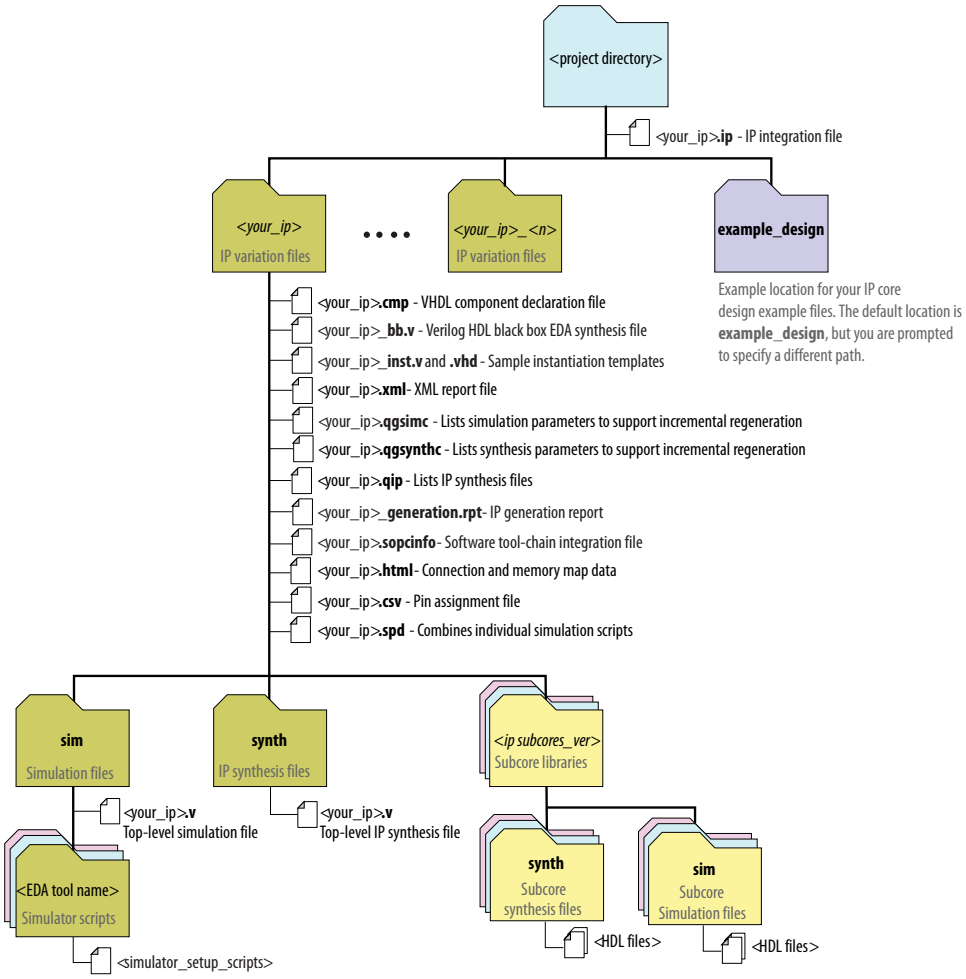


Table 12. IP Core Generated Files

File Name	Description
<*your_ip*>.ip	The Platform Designer system or top-level IP variation file. <*your_ip*> is the name that you give your IP variation.
<*your_ip*>.cmp	The VHDL Component Declaration (.cmp) file is a text file that contains local generic and port definitions that you can use in VHDL design files.
<*your_ip*>.html	A report that contains connection information, a memory map showing the address of each slave with respect to each master to which it is connected, and parameter assignments.

continued...

File Name	Description
<*your_ip*>_generation.rpt	IP or Platform Designer generation log file. A summary of the messages during IP generation.
<*your_ip*>.qgsimc	Lists simulation parameters to support incremental regeneration.
<*your_ip*>.qgsynthc	Lists synthesis parameters to support incremental regeneration.
<*your_ip*>.qip	Contains all the required information about the IP component to integrate and compile the IP component in the Intel Quartus Prime software.
<*your_ip*>.sopcinfo	Describes the connections and IP component parameterizations in your Platform Designer system. You can parse its contents to get requirements when you develop software drivers for IP components. Downstream tools such as the Nios® II tool chain use this file. The .sopcinfo file and the system.h file generated for the Nios II tool chain include address map information for each slave relative to each master that accesses the slave. Different masters may have a different address map to access a particular slave component.
<*your_ip*>.csv	Contains information about the upgrade status of the IP component.
<*your_ip*>.spd	Required input file for ip-make-simscript to generate simulation scripts for supported simulators. The .spd file contains a list of files generated for simulation, along with information about memories that you can initialize.
<*your_ip*>_bb.v	You can use the Verilog black-box (_bb.v) file as an empty module declaration for use as a black box.
<*your_ip*>_inst.v or _inst.vhd	HDL example instantiation template. You can copy and paste the contents of this file into your HDL file to instantiate the IP variation.
<*your_ip*>.svd	Allows hard processor system (HPS) System Debug tools to view the register maps of peripherals connected to HPS in a Platform Designer system. During synthesis, the .svd files for slave interfaces visible to System Console masters are stored in the .sof file in the debug section. System Console reads this section, which Platform Designer can query for register map information. For system slaves, Platform Designer can access the registers by name.
<*your_ip*>.v or <*your_ip*>.vhd	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim* script msim_setup.tcl to set up and run a simulation.
synopsys/vcs/ synopsys/vcsmx/	Contains a shell script vcs_setup.sh to set up and run a VCS* simulation. Contains a shell script vcsmx_setup.sh and synopsys_ sim.setup file to set up and run a VCS MX* simulation.
cadence/	Contains a shell script ncsim_setup.sh and other setup files to set up and run an NCSIM* simulation.
submodules/	Contains HDL files for the IP core submodules.
<*child IP cores*>/	For each generated child IP core directory, Platform Designer generates synth/ andsim/ sub-directories.

Related Information

- E-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
- E-Tile Hard IP for Ethernet Intel Agilex FPGA IP Design Example User Guide

2.7.4. Integrating Your IP Core in Your Design

When you integrate your IP core instance in your design, you must pay attention to the following items:

Channel Placement on page 27
Pin Assignments on page 49
Clock Requirements on page 50
External Time-of-Day Module for Variations with 1588 PTP Feature on page 50
SDC for Multiple E-Tile Instances on page 52

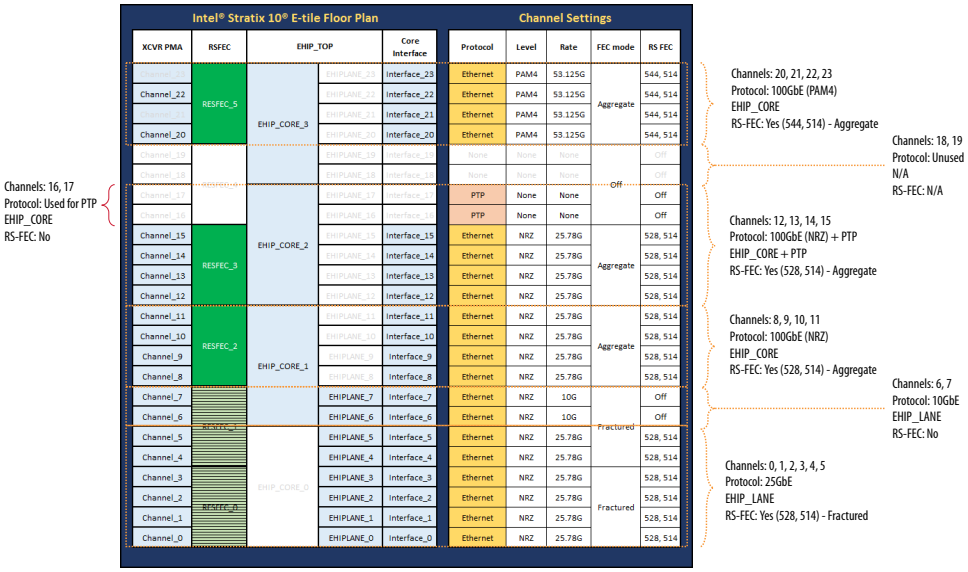
2.7.4.1. Channel Placement

Depending on the number of bonded PMA channels, the Channel Placement tool supports two E-tile variants, 24-channel placement and 16-channel placement variant.

In Intel Stratix 10 devices, each E-tile provides Hard IP for up to 4 100G Ethernet channels, and up to 24 10G/25G Ethernet channels. In Intel Agilex devices, each E-tile provides Hard IP for up to 4 100G Ethernet channels, and either up to 24 10G/25G Ethernet channels in 24-channel bonding configuration, or up to 16 10G/24G Ethernet channels in 16-channel bonding configuration.

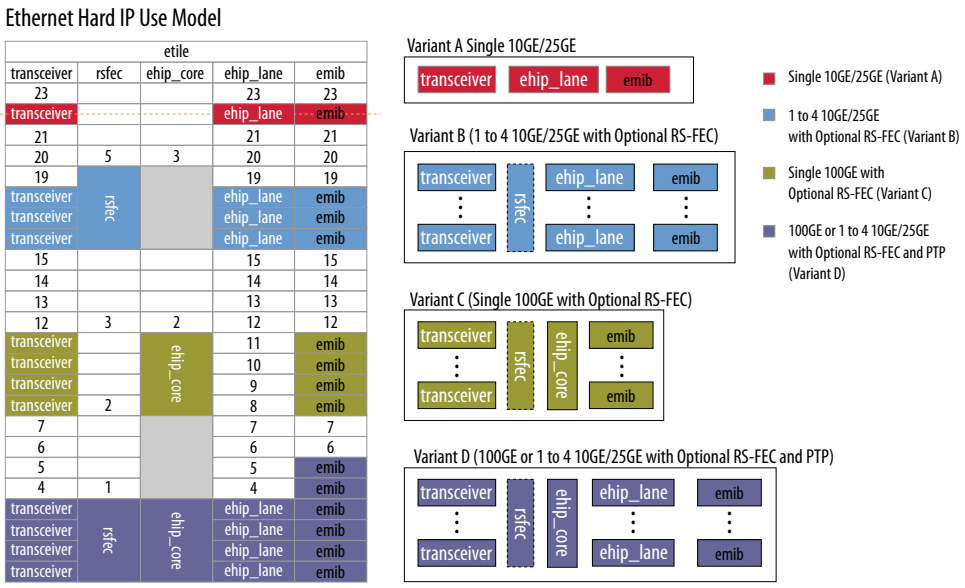
Figure 10. Ethernet Hard IP Overview

The figure shows an example placement of the channels using the E-tile Channel Placement Tool.



RS-FEC is configurable for single-lane 10G/25G and multi-lane 100G Ethernet interfaces.

Figure 11. Ethernet Cores Position on an E-tile Device



You can place the core by constraining a serial pin from the core to one of the transceiver pins on the selected E-tile device. For example, if you constrain the serial pins from a Variant A core to transceiver pin 10, the core will be placed in Variant A position 10.

Related Information

- E-Tile Transceiver PHY User Guide
  For information about constraints on transceiver configuration for Hard IP for Ethernet in E-Tile devices.
- E-Tile Channel Placement Tool

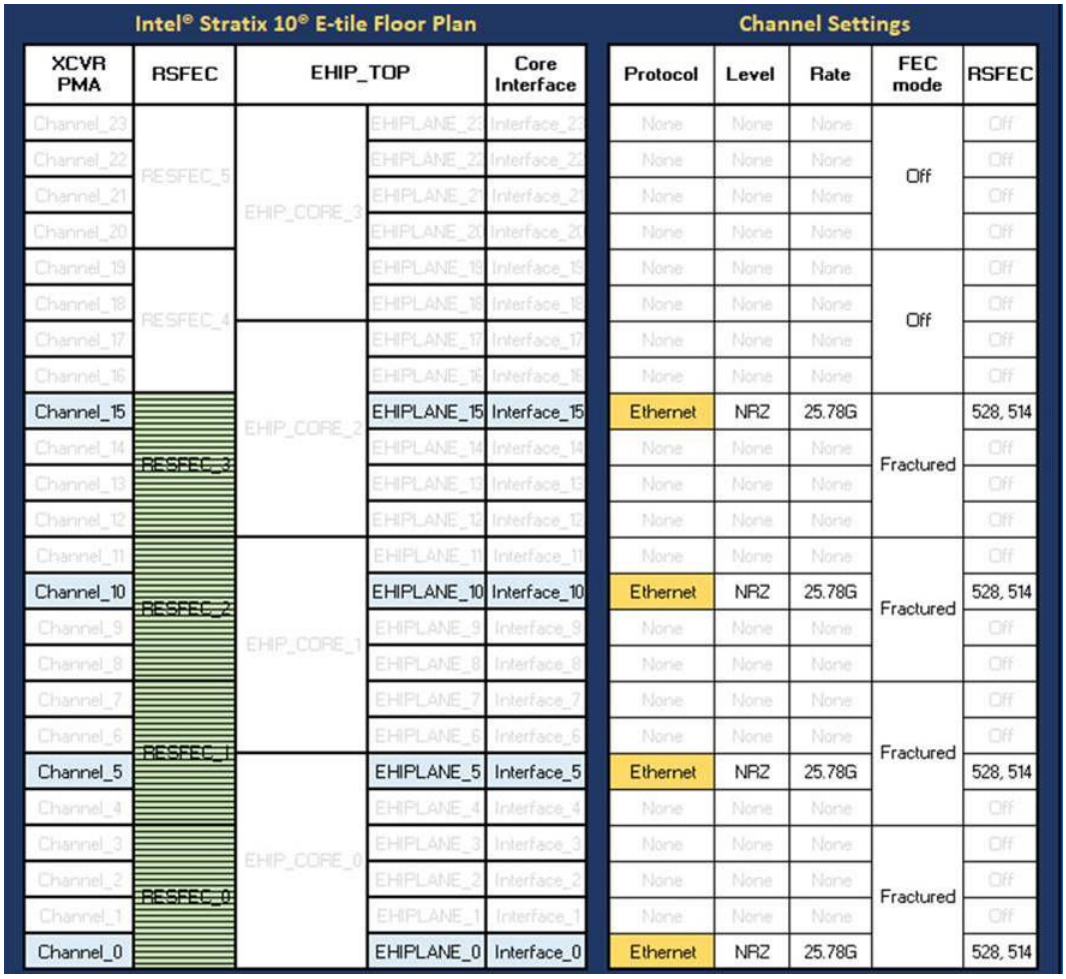
2.7.4.1.1. Guidelines and Restrictions for 24-bonded Channels Variant

The E-tile transceiver requires the channel placement to be contiguous when RS-FEC is enabled. The following are the allowed channel placements for different number of channels.

Single Channel 10GE/25GE with RS-FEC Variant

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **Single Channel**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0**(channel_0), **first_lane1**(channel_5), **first_lane2**(channel_10), or **first_lane3**(channel_15)
- The FEC mode is set to fractured mode.



Intel® Stratix 10® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

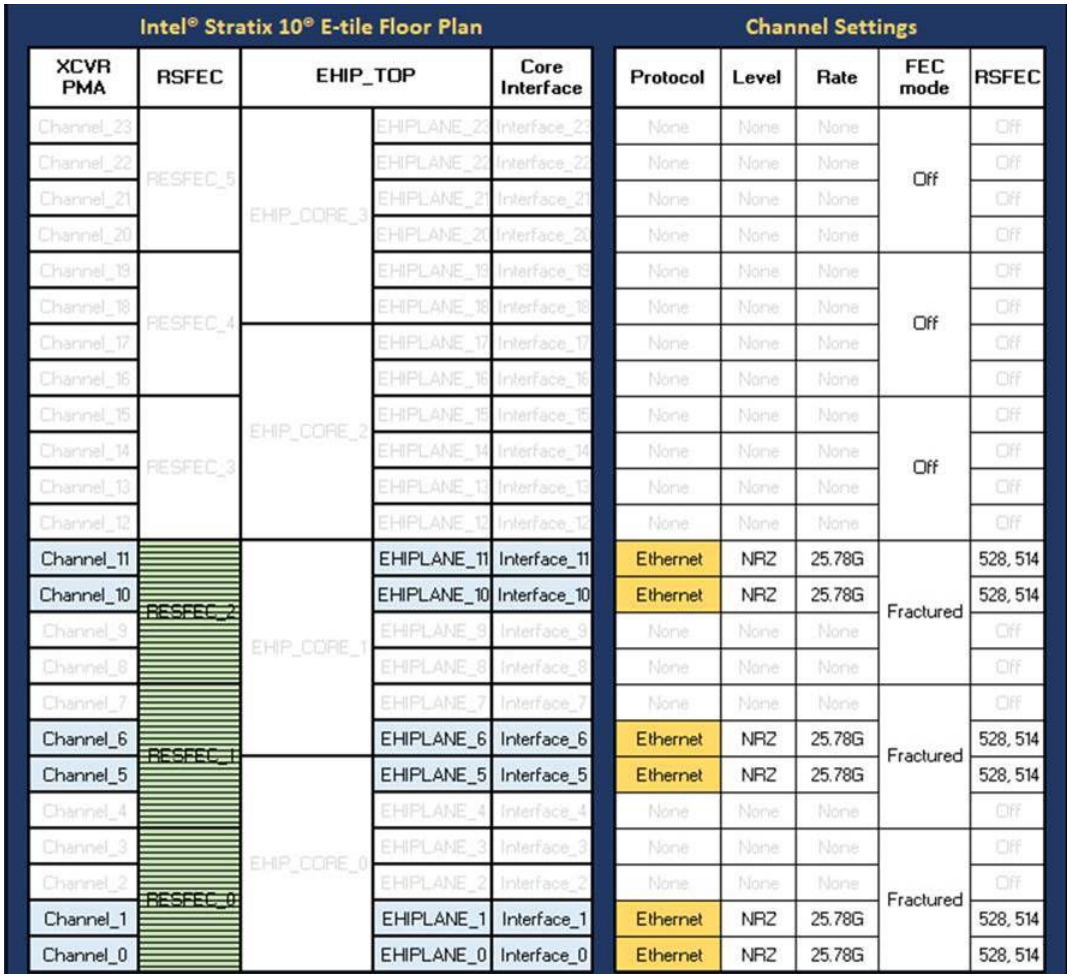
Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Fractured	Off
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None		Off
None	None	None	Fractured	Off
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None	Fractured	Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514

Two Channels 10GE/25GE with RS-FEC Variant

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **2 Channels**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0**(channel_0), **first_lane1**(channel_5), or **first_lane2**(channel_10)
- The FEC mode is set to fractured mode.

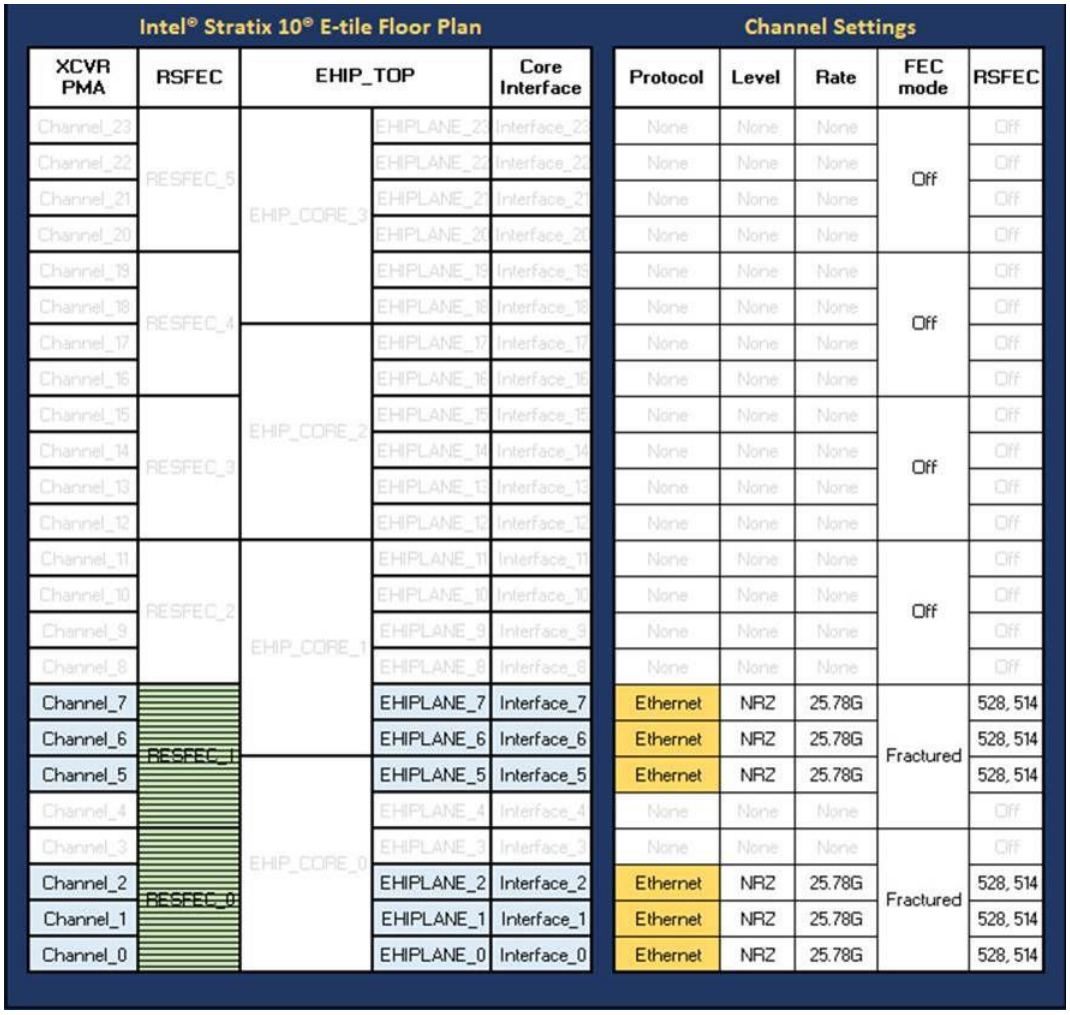


Intel® Stratix 10® E-tile Floor Plan					Channel Settings				
XCVR PMA	RSFEC	EHIP_TOP		Core Interface	Protocol	Level	Rate	FEC mode	RSFEC
Channel_23	RSFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23	None	None	None	Off	Off
Channel_22			EHIPLANE_22	Interface_22	None	None	None		Off
Channel_21			EHIPLANE_21	Interface_21	None	None	None		Off
Channel_20			EHIPLANE_20	Interface_20	None	None	None		Off
Channel_19	RSFEC_4		EHIPLANE_19	Interface_19	None	None	None	Off	Off
Channel_18			EHIPLANE_18	Interface_18	None	None	None		Off
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17	None	None	None		Off
Channel_16			EHIPLANE_16	Interface_16	None	None	None		Off
Channel_15	RSFEC_3		EHIPLANE_15	Interface_15	None	None	None	Off	Off
Channel_14			EHIPLANE_14	Interface_14	None	None	None		Off
Channel_13			EHIPLANE_13	Interface_13	None	None	None		Off
Channel_12			EHIPLANE_12	Interface_12	None	None	None		Off
Channel_11	RSFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11	Ethernet	NRZ	25.78G	Fractured	528, 514
Channel_10			EHIPLANE_10	Interface_10	Ethernet	NRZ	25.78G		528, 514
Channel_9			EHIPLANE_9	Interface_9	None	None	None		Off
Channel_8			EHIPLANE_8	Interface_8	None	None	None		Off
Channel_7	RSFEC_1		EHIPLANE_7	Interface_7	None	None	None	Fractured	Off
Channel_6			EHIPLANE_6	Interface_6	Ethernet	NRZ	25.78G		528, 514
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5	Ethernet	NRZ	25.78G		528, 514
Channel_4			EHIPLANE_4	Interface_4	None	None	None		Off
Channel_3	RSFEC_0		EHIPLANE_3	Interface_3	None	None	None	Fractured	Off
Channel_2			EHIPLANE_2	Interface_2	None	None	None		Off
Channel_1			EHIPLANE_1	Interface_1	Ethernet	NRZ	25.78G		528, 514
Channel_0			EHIPLANE_0	Interface_0	Ethernet	NRZ	25.78G		528, 514

Three Channels 10GE/25GE with RS-FEC

Following are the parameter settings for these channel placements:

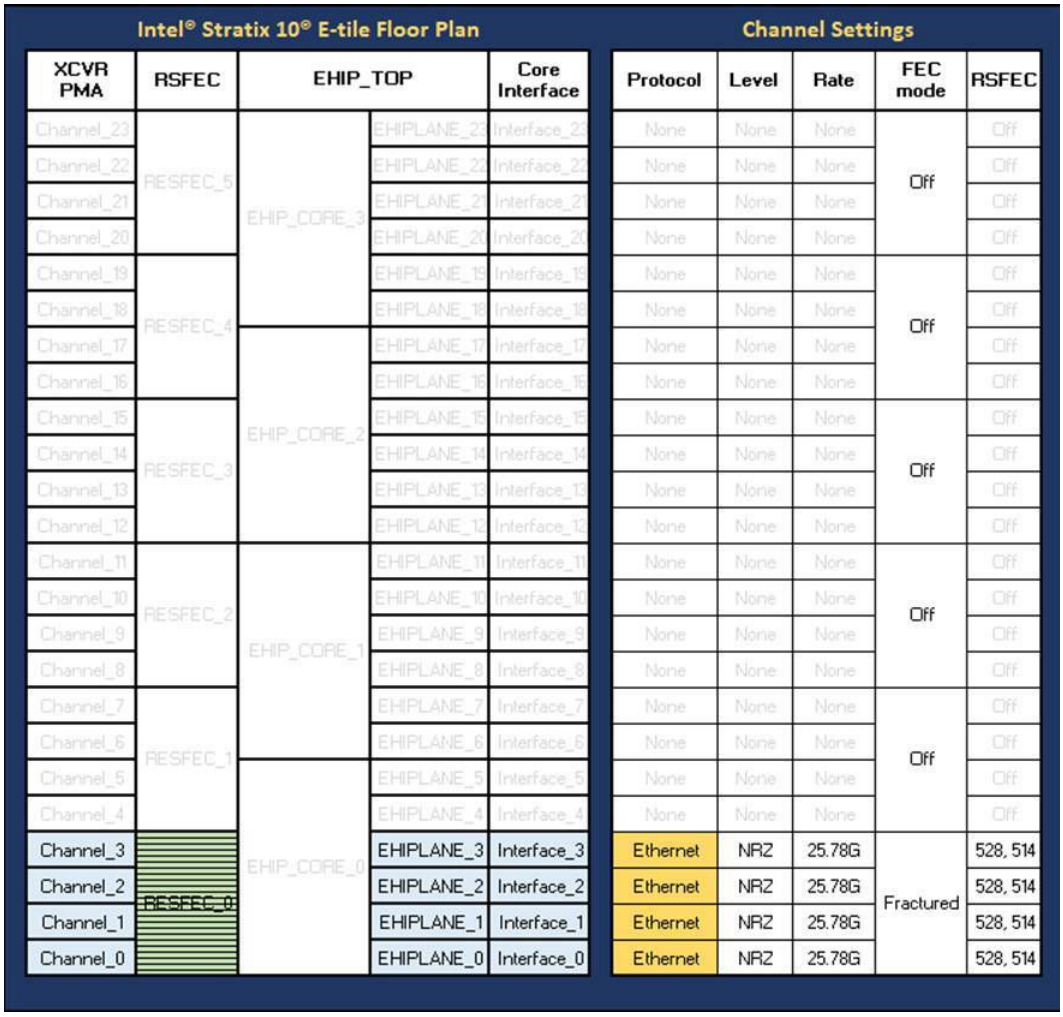
- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **3 Channels**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0**(channel_0) or **first_lane1**(channel_5)
- The FEC mode is set to fractured mode.



Four Channels 10GE/25GE with RS-FEC Variant

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **4 Channels**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0**(channel_0)
- The FEC mode is set to fractured mode.



Intel® Stratix 10® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RSFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RSFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RSFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RSFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RSFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RSFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514

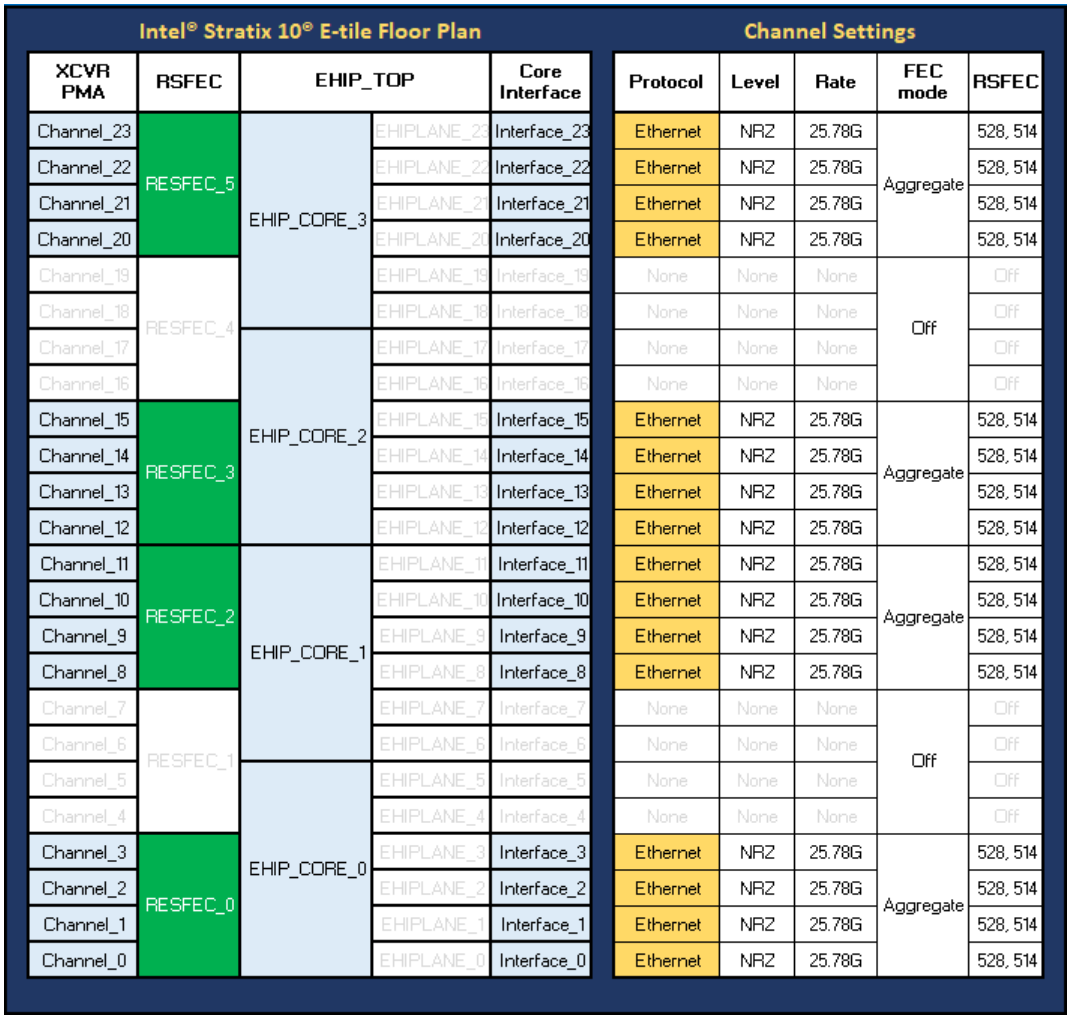
100GE with RS-FEC Variant

You can use the following channel placements for a four-channel Native PHY with single RS-FEC block without PTP variant.

In **100GE** tab, **Select Ethernet IP Layers**: MAC+PCS+RSFEC.

Following are the **IP** tab parameter settings for these channel placements:

- **Select Core Variant**: **Single 100GE with optional RSFEC**
- **Enable RSFEC**: Selected
- The FEC mode is set to aggregate mode.



Intel® Stratix 10® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
Ethernet	NRZ	25.78G	Aggregate	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G	Aggregate	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G	Aggregate	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G	Aggregate	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514

Single Channel 10GE/25GE with RS-FEC and PTP Variant

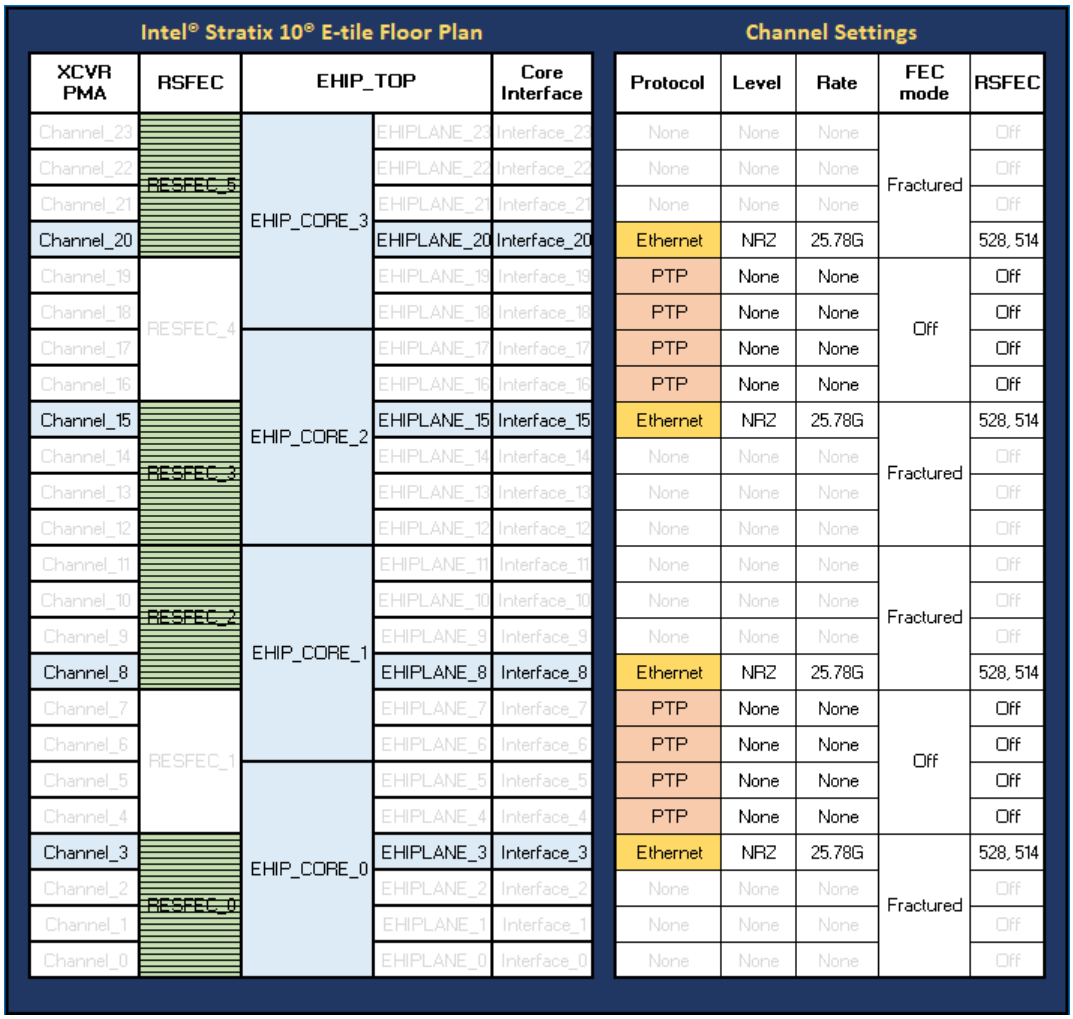
When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **Single Channel**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**
- **Enable IEEE 1588 PTP**: Selected

- **Enable RSFEC**: Selected
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP0/2**(EHIP_CORE_0/EHIP_CORE2), the **First RSFEC Lane** should set to **first_lane3**(channel_3 or channel_15).
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3), the **First RSFEC Lane** should set to **first_lane0**(channel_8 or channel_20).
- The FEC mode is set to fractured mode.



Two Channels 10GE/25GE with RS-FEC and PTP Variant

When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **2 Channels**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**
- **Enable IEEE 1588 PTP**: Selected
- **Enable RSFEC**: Selected
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP0/2**(EHIP_CORE_0/EHIP_CORE2), the **First RSFEC Lane** should set to **first_lane2**.
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3), the **First RSFEC Lane** should set to **first_lane0**(channel_8 or channel_20).
- The FEC mode is set to fractured mode.

Intel® Stratix 10® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RSFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RSFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RSFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RSFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RSFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RSFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
None	None	None	Fractured	Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None		Off
None	None	None	Fractured	Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None		Off

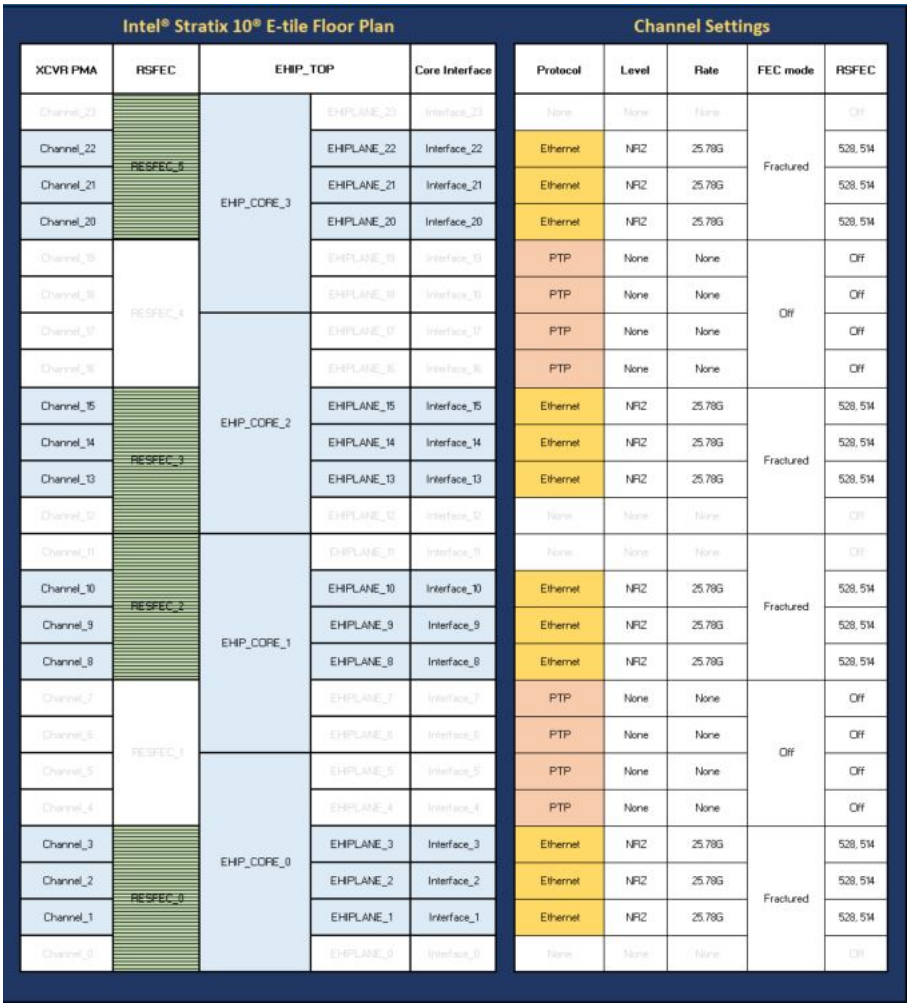
Three Channels 10GE/25GE with RS-FEC and PTP Variant

When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **3 Channels**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**
- **Enable IEEE 1588 PTP**: Selected
- **Enable RSFEC**: Selected
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP0/2**(EHIP_CORE_0/EHIP_CORE2), the **First RSFEC Lane** should set to **first_lane1**(channel_3 or channel_15).
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3), the **First RSFEC Lane** should set to **first_lane0**(channel_8 or channel_20).
- The FEC mode is set to fractured mode.



Four Channels 10GE/25GE with RS-FEC and PTP Variant

When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **4 Channels**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**
- **Enable IEEE 1588 PTP**: Selected
- **Enable RSFEC**: Selected

- **IEEE1588/PTP channel placement restriction**: **EHIP0/2**(EHIP_CORE_0/EHIP_CORE2) or **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3)
- **First RSFEC Lane**: **first_lane0**
- The FEC mode is set to fractured mode.

Intel® Stratix 10® E-tile Floor Plan					Channel Settings				
XCVR PMA	RSFEC	EHIP_TOP		Core Interface	Protocol	Level	Rate	FEC mode	RSFEC
Channel_23	RSFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23	Ethernet	NRZ	25.78G	Fractured	528, 514
Channel_22			EHIPLANE_22	Interface_22	Ethernet	NRZ	25.78G		528, 514
Channel_21			EHIPLANE_21	Interface_21	Ethernet	NRZ	25.78G		528, 514
Channel_20			EHIPLANE_20	Interface_20	Ethernet	NRZ	25.78G		528, 514
Channel_19	RSFEC_4		EHIPLANE_19	Interface_19	PTP	None	None	Off	Off
Channel_18			EHIPLANE_18	Interface_18	PTP	None	None		Off
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17	PTP	None	None		Off
Channel_16			EHIPLANE_16	Interface_16	PTP	None	None		Off
Channel_15	RSFEC_3		EHIPLANE_15	Interface_15	Ethernet	NRZ	25.78G	Fractured	528, 514
Channel_14			EHIPLANE_14	Interface_14	Ethernet	NRZ	25.78G		528, 514
Channel_13			EHIPLANE_13	Interface_13	Ethernet	NRZ	25.78G		528, 514
Channel_12			EHIPLANE_12	Interface_12	Ethernet	NRZ	25.78G		528, 514
Channel_11	RSFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11	Ethernet	NRZ	25.78G	Fractured	528, 514
Channel_10			EHIPLANE_10	Interface_10	Ethernet	NRZ	25.78G		528, 514
Channel_9			EHIPLANE_9	Interface_9	Ethernet	NRZ	25.78G		528, 514
Channel_8			EHIPLANE_8	Interface_8	Ethernet	NRZ	25.78G		528, 514
Channel_7	RSFEC_1		EHIPLANE_7	Interface_7	PTP	None	None	Off	Off
Channel_6			EHIPLANE_6	Interface_6	PTP	None	None		Off
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5	PTP	None	None		Off
Channel_4			EHIPLANE_4	Interface_4	PTP	None	None		Off
Channel_3	RSFEC_0		EHIPLANE_3	Interface_3	Ethernet	NRZ	25.78G	Fractured	528, 514
Channel_2			EHIPLANE_2	Interface_2	Ethernet	NRZ	25.78G		528, 514
Channel_1			EHIPLANE_1	Interface_1	Ethernet	NRZ	25.78G		528, 514
Channel_0			EHIPLANE_0	Interface_0	Ethernet	NRZ	25.78G		528, 514

2.7.4.1.2. Guidelines and Restrictions for 16-bonded Channels Variant

The E-tile transceiver requires the channel placement to be contiguous when RS-FEC is enabled. The following are the allowed channel placements for different number of channels.

Single Channel 10GE/25GE with RS-FEC Variant in Intel Agilex Devices

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **Single Channel**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0**(channel_0), **first_lane1**(channel_9), **first_lane2**(channel_14), or **first_lane3**(channel_23)
- The FEC mode is set to fractured mode.

Two Channels 10GE/25GE with RS-FEC Variant in Agilex Devices

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **2 Channels**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0**(channel_0), **first_lane1**(channel_9), or **first_lane2**(channel_14)
- The FEC mode is set to fractured mode.

Intel® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None		Off
None	None	None	Fractured	Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None	Off	Off
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Fractured	Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514

Three Channels 10GE/25GE with RS-FEC in Intel Agilex Devices

Following are the parameter settings for these channel placements:

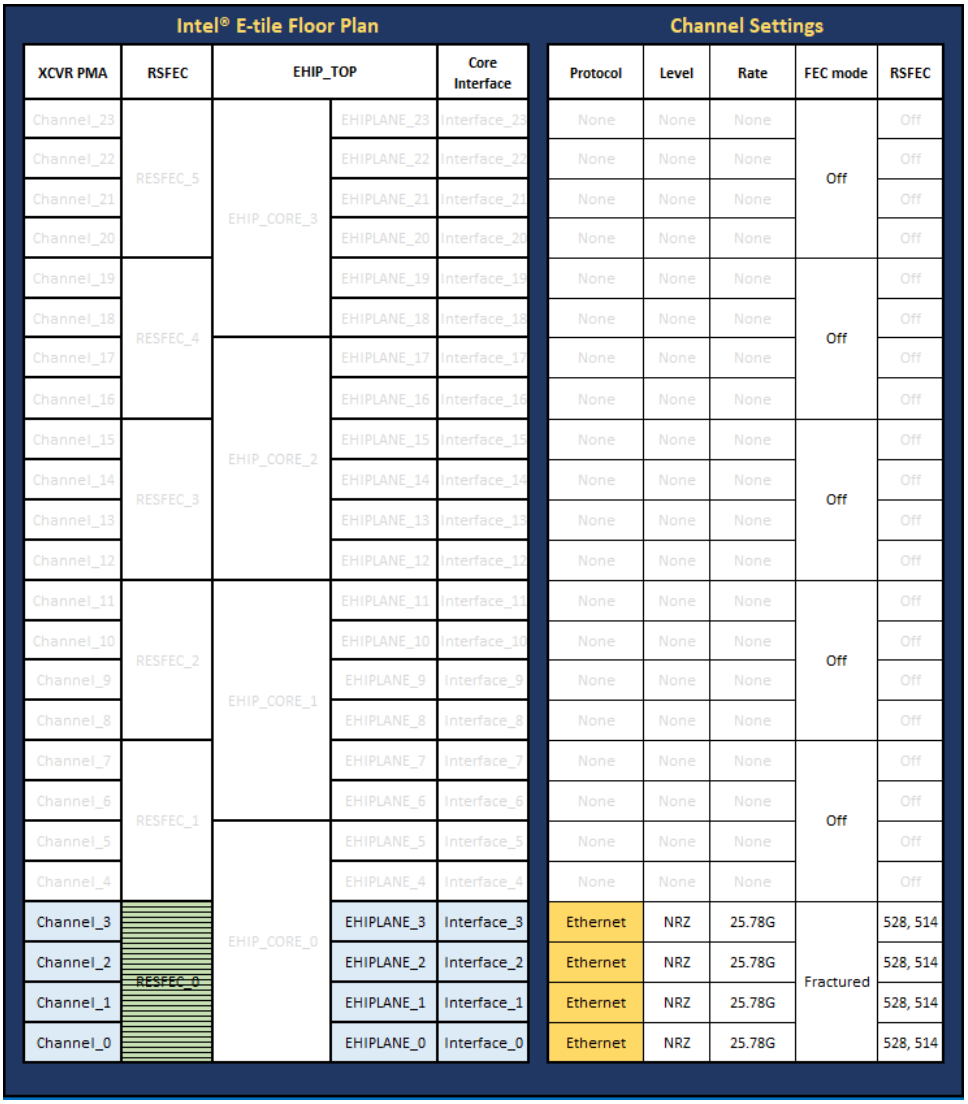
- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **3 Channels**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0**(channel_0) or **first_lane1**(channel_9)
- The FEC mode is set to fractured mode.

Intel® E-tile Floor Plan					Channel Settings				
XCVR PMA	RSFEC	EHIP_TOP		Core Interface	Protocol	Level	Rate	FEC mode	RSFEC
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23	None	None	None	Off	Off
Channel_22			EHIPLANE_22	Interface_22	None	None	None		Off
Channel_21			EHIPLANE_21	Interface_21	None	None	None		Off
Channel_20			EHIPLANE_20	Interface_20	None	None	None		Off
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19	None	None	None	Off	Off
Channel_18			EHIPLANE_18	Interface_18	None	None	None		Off
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17	None	None	None		Off
Channel_16			EHIPLANE_16	Interface_16	None	None	None		Off
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15	None	None	None	Off	Off
Channel_14			EHIPLANE_14	Interface_14	None	None	None		Off
Channel_13			EHIPLANE_13	Interface_13	None	None	None		Off
Channel_12			EHIPLANE_12	Interface_12	None	None	None		Off
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11	Ethernet	NRZ	25.78G	Fractured	528, 514
Channel_10			EHIPLANE_10	Interface_10	Ethernet	NRZ	25.78G		528, 514
Channel_9			EHIPLANE_9	Interface_9	Ethernet	NRZ	25.78G		528, 514
Channel_8			EHIPLANE_8	Interface_8	None	None	None		Off
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7	None	None	None	Off	Off
Channel_6			EHIPLANE_6	Interface_6	None	None	None		Off
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5	None	None	None		Off
Channel_4			EHIPLANE_4	Interface_4	None	None	None		Off
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3	None	None	None	Fractured	Off
Channel_2			EHIPLANE_2	Interface_2	Ethernet	NRZ	25.78G		528, 514
Channel_1			EHIPLANE_1	Interface_1	Ethernet	NRZ	25.78G		528, 514
Channel_0			EHIPLANE_0	Interface_0	Ethernet	NRZ	25.78G		528, 514

Four Channels 10GE/25GE with RS-FEC Variant in Intel Agilex Devices

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **1 to 4 10GE/25GE with optional RSFEC**
- **Number of Channels of 10GE/25GE**: **4 Channels**
- **Enable RSFEC**: Selected
- **First RSFEC Lane**: **first_lane0** (channel_0)
- The FEC mode is set to fractured mode.



100GE with RS-FEC Variant in Intel Agilex Devices

You can use the following channel placements for a four-channel Native PHY with single RS-FEC block without PTP variant.

In **100GE** tab, **Select Ethernet IP Layers**: MAC+PCS+RSFEC.

Following are the **IP** tab parameter settings for these channel placements:

- **Select Core Variant**: **Single 100GE with optional RSFEC**
- **Enable RSFEC**: Selected
- The FEC mode is set to aggregate mode.

Intel® E-tile Floor Plan					Channel Settings				
XCVR PMA	RSFEC	EHIP_TOP		Core Interface	Protocol	Level	Rate	FEC mode	RSFEC
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23	Ethernet	NRZ	25.78G	Aggregate	528, 514
Channel_22			EHIPLANE_22	Interface_22	Ethernet	NRZ	25.78G		528, 514
Channel_21			EHIPLANE_21	Interface_21	Ethernet	NRZ	25.78G		528, 514
Channel_20			EHIPLANE_20	Interface_20	Ethernet	NRZ	25.78G		528, 514
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19	None	None	None	Off	Off
Channel_18			EHIPLANE_18	Interface_18	None	None	None		Off
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17	None	None	None		Off
Channel_16			EHIPLANE_16	Interface_16	None	None	None		Off
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15	Ethernet	NRZ	25.78G	Aggregate	528, 514
Channel_14			EHIPLANE_14	Interface_14	Ethernet	NRZ	25.78G		528, 514
Channel_13			EHIPLANE_13	Interface_13	Ethernet	NRZ	25.78G		528, 514
Channel_12			EHIPLANE_12	Interface_12	Ethernet	NRZ	25.78G		528, 514
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11	Ethernet	NRZ	25.78G	Aggregate	528, 514
Channel_10			EHIPLANE_10	Interface_10	Ethernet	NRZ	25.78G		528, 514
Channel_9			EHIPLANE_9	Interface_9	Ethernet	NRZ	25.78G		528, 514
Channel_8			EHIPLANE_8	Interface_8	Ethernet	NRZ	25.78G		528, 514
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7	None	None	None	Off	Off
Channel_6			EHIPLANE_6	Interface_6	None	None	None		Off
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5	None	None	None		Off
Channel_4			EHIPLANE_4	Interface_4	None	None	None		Off
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3	Ethernet	NRZ	25.78G	Aggregate	528, 514
Channel_2			EHIPLANE_2	Interface_2	Ethernet	NRZ	25.78G		528, 514
Channel_1			EHIPLANE_1	Interface_1	Ethernet	NRZ	25.78G		528, 514
Channel_0			EHIPLANE_0	Interface_0	Ethernet	NRZ	25.78G		528, 514

Single Channel 10GE/25GE with RS-FEC and PTP Variant in Intel Agilex Devices

When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **Single Channel**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**
- **Enable IEEE 1588 PTP**: Selected
- **Enable RSFEC**: Selected
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP0/2**(EHIP_CORE_0/EHIP_CORE2), the **First RSFEC Lane** should set to **first_lane3**(channel_3 or channel_15).
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3), the **First RSFEC Lane** should set to **first_lane0**(channel_8 or channel_20).
- The FEC mode is set to fractured mode.

Intel® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
None	None	None	Fractured	Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
None	None	None		Off
None	None	None		Off
None	None	None		Off
None	None	None	Fractured	Off
None	None	None		Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
None	None	None		Off
None	None	None		Off
None	None	None		Off

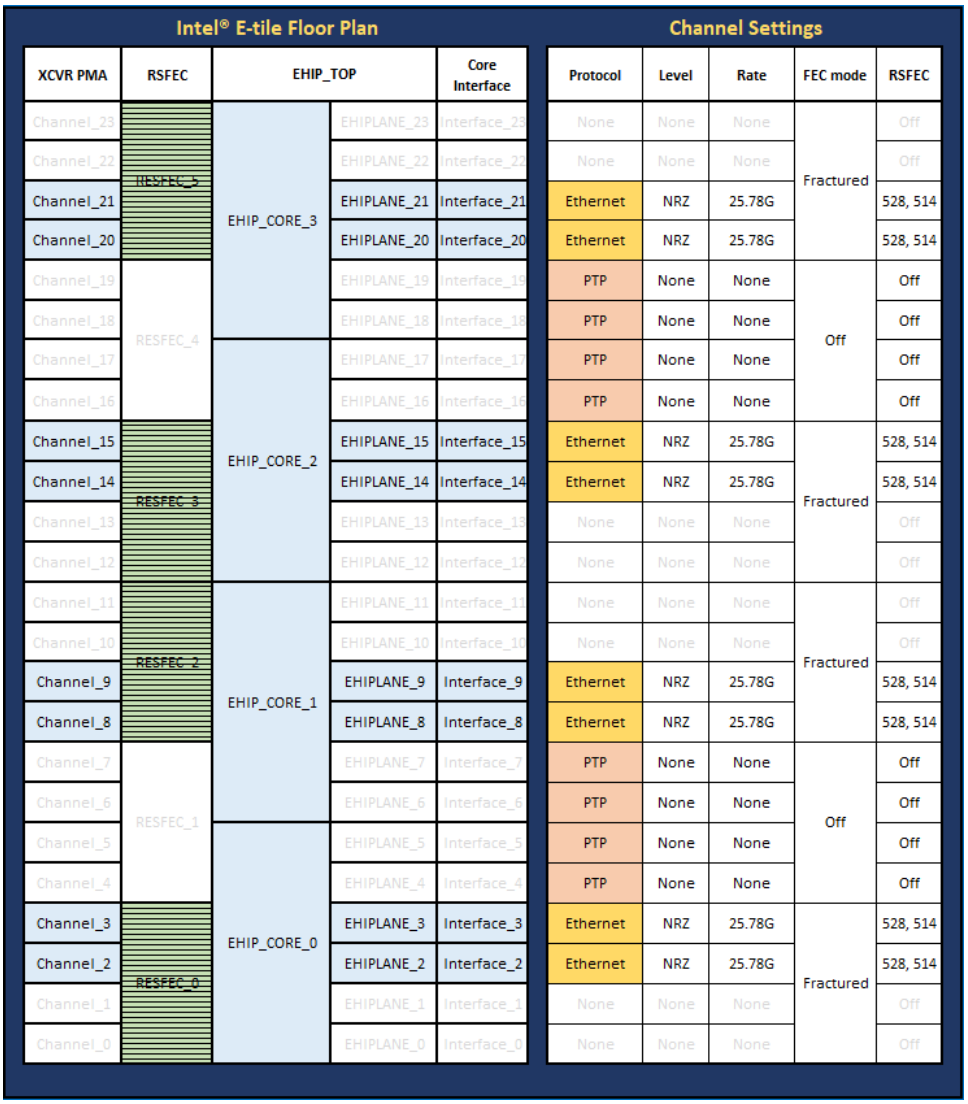
Two Channels 10GE/25GE with RS-FEC and PTP Variant in Intel Agilex Devices

When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **2 Channels**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**
- **Enable IEEE 1588 PTP**: Selected
- **Enable RSFEC**: Selected
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP0/2**(EHIP_CORE_0/EHIP_CORE2), the **First RSFEC Lane** should set to **first_lane2**.
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3), the **First RSFEC Lane** should set to **first_lane0**(channel_8 or channel_20).
- The FEC mode is set to fractured mode.



Intel® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
None	None	None	Fractured	Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None		Off
None	None	None	Fractured	Off
None	None	None		Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None		Off

Three Channels 10GE/25GE with RS-FEC and PTP Variant in Intel Agilex Devices

When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **3 Channels**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**
- **Enable IEEE 1588 PTP**: Selected
- **Enable RSFEC**: Selected
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP0/2**(EHIP_CORE_0/EHIP_CORE2), the **First RSFEC Lane** should set to **first_lane1**(channel_3 or channel_15).
- When **IEEE1588/PTP channel placement restriction** is set to **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3), the **First RSFEC Lane** should set to **first_lane0**(channel_8 or channel_20).
- The FEC mode is set to fractured mode.

Intel® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
None	None	None	Fractured	Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off
None	None	None	Fractured	Off
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
None	None	None		Off

Four Channels 10GE/25GE with RS-FEC and PTP Variant

When RS-FEC and PTP are enabled, place the RS-FEC channel next to the PTP channels. The supported PTP channel locations for this variant are:

- Channel_4/5 when using EHIP_CORE0
- Channel 6/7 when using EHIP_CORE1
- Channel 16/17 when using EHIP_CORE2
- Channel 18/19 when using EHIP_CORE3

Following are the parameter settings for these channel placements:

- **Select Core Variant**: **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP**
- **Number of Channels of 10GE/25GE**: **4 Channels**
- **Active Channel(s) at startup**: **10G/25GE Channel(s)**

- **Enable IEEE 1588 PTP**: Selected
- **Enable RSFEC**: Selected
- **IEEE1588/PTP channel placement restriction**: **EHIP0/2**(EHIP_CORE_0/ EHIP_CORE2) or **EHIP1/3**(EHIP_CORE_1/EHIP_CORE3)
- **First RSFEC Lane**: **first_lane0**
- The FEC mode is set to fractured mode.

Intel® E-tile Floor Plan

XCVR PMA	RSFEC	EHIP_TOP		Core Interface
Channel_23	RESFEC_5	EHIP_CORE_3	EHIPLANE_23	Interface_23
Channel_22			EHIPLANE_22	Interface_22
Channel_21			EHIPLANE_21	Interface_21
Channel_20			EHIPLANE_20	Interface_20
Channel_19	RESFEC_4		EHIPLANE_19	Interface_19
Channel_18			EHIPLANE_18	Interface_18
Channel_17		EHIP_CORE_2	EHIPLANE_17	Interface_17
Channel_16			EHIPLANE_16	Interface_16
Channel_15	RESFEC_3		EHIPLANE_15	Interface_15
Channel_14			EHIPLANE_14	Interface_14
Channel_13			EHIPLANE_13	Interface_13
Channel_12			EHIPLANE_12	Interface_12
Channel_11	RESFEC_2	EHIP_CORE_1	EHIPLANE_11	Interface_11
Channel_10			EHIPLANE_10	Interface_10
Channel_9			EHIPLANE_9	Interface_9
Channel_8			EHIPLANE_8	Interface_8
Channel_7	RESFEC_1		EHIPLANE_7	Interface_7
Channel_6			EHIPLANE_6	Interface_6
Channel_5		EHIP_CORE_0	EHIPLANE_5	Interface_5
Channel_4			EHIPLANE_4	Interface_4
Channel_3	RESFEC_0		EHIPLANE_3	Interface_3
Channel_2			EHIPLANE_2	Interface_2
Channel_1			EHIPLANE_1	Interface_1
Channel_0			EHIPLANE_0	Interface_0

Channel Settings

Protocol	Level	Rate	FEC mode	RSFEC
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
PTP	None	None	Off	Off
PTP	None	None		Off
PTP	None	None		Off
PTP	None	None		Off
Ethernet	NRZ	25.78G	Fractured	528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514
Ethernet	NRZ	25.78G		528, 514

2.7.4.2. Pin Assignments

When you integrate your E-Tile Hard IP for Ethernet Intel FPGA IP core instance in your design, you must make appropriate pin assignments. You can create a virtual pin to avoid making specific pin assignments for top-level signals until you are ready to map the design to hardware.

Intel Stratix 10 E-tile devices offer four instances of the hard IP on each E-tile. Each instance offers one 100G channel and six 10G/25G channels. Your design must not include pin assignments that conflict with its location. In devices with multiple E-tiles, you can specify the E-tile to which the Ethernet link serial pins should map.

Related Information

Quartus Prime Help
For information about the Quartus Prime software, including virtual pins and the IP Catalog.

2.7.4.3. Clock Requirements

The E-Tile Hard IP for Ethernet Intel FPGA IP provides locally generated PLL clocks used for RX and TX datapath, and recovered clocks to enable Synchronous Ethernet (SyncE).

For synchronized-mode operation without external AIB clocking enabled, ensure that the output clock, `o_clk_pll_div64`, drives both `i_sl_clk_rx` and the `i_sl_clk_tx` input clocks. If multiple IP core instances are instantiated, connect `o_clk_pll_div64` output clock of respective channel to its input clocks. When external AIB clocking is enabled, `i_sl_clk_tx` and `i_sl_clk_rx` input clocks must be driven by the same source as `i_aib_clk`. Another important aspect for Synchronous Ethernet operation is to connect filtered and divided version of RX recovered clock (`o_clk_rec_div64` or `o_clk_rec_div66`) to input reference clock (`i_clk_ref`).

For any PTP variants, Intel recommends using the clock output from PTP channel adjacent to the data channel. For example, in a three-channel 25G design for `EHIP_CORE0/2`, your design must use the clock output from the PTP channel adjacent to channel 3. Similarly, for `EHIP_CORE1/3`, use clock output from the PTP channel next to channel 0.

Related Information

Clocks on page 158

2.7.4.4. External Time-of-Day Module for Variations with 1588 PTP Feature

E-Tile Hard IP for Ethernet Intel FPGA IP configurations that include the 1588 PTP module require an external time-of-day (TOD) module to provide a continuous flow of current time-of-day information. The TOD module must provide the entire 96 bits of time-of-day, in the V2 format, to avoid the reduced accuracy. You can instantiate the TOD module from the IP Catalog[3] and add it to your design. The TOD module updates the time-of-day output value on every clock cycle in the V2 format (96 bits).

When you generate a design example for your IP core with PTP variation, it includes the TOD module.

TOD Module Required Connections in 10G/25G Basic Mode or 100G PTP

When you select **Basic Mode** in the **PTP Accuracy Mode** under PTP Options, each 10G/25G Ethernet IP instance includes only one `i_ptp_tod[95:0]` port. For 1 to 4 10G/25G channels in the same E-Tile Hard IP for Ethernet IP core or 100G E-Tile Hard

[3] The TOD module is located in the Ethernet IEEE 1588 TOD Synchronizer Intel FPGA IP.

IP for Ethernet IP core, you need a single TOD IP and a single TOD Synchronizer IP. You cannot share the TOD IP with different E-Tile Hard IP for Ethernet IP core channels.

Table 13. TOD Module Required Connections - 10G/25G Basic Mode or 100G PTP

Required connections for TOD module, listed using signal names of TOD modules. If you create your own TOD module it must have the output signals required by the E-Tile Hard IP for Ethernet Intel FPGA IP. However, its signal names could be different than the TOD module signal names in the table.

TOD Module Signals	Basic Mode Recommended Connections
`clk`	Avalon memory-mapped interface clock
`rst_n`	Avalon memory-mapped interface reset
`period_clk`	`o_clk_pll_div64` (PTP clock) For more information, refer to *Clocks* section.
`period_rst_n`	Global/system reset
`time_of_day_96b[95:0]`	`i_ptp_tod[95:0]`

TOD Module Required Connections in 10G/25G Advanced Mode

When you select **Advanced Mode** in the **PTP Accuracy Mode** under PTP Options, each channel in a 10G/25G Ethernet IP instance contains two TOD IP interface ports, the `i_sl_ptp_tx_tod[95:0]` for TX port and the `i_sl_ptp_rx_tod[95:0]` for RX port. Each channel in the same 10G/25G E-Tile Hard IP for Ethernet IP core requires two TOD IPs and two TOD Synchronizer IPs.

Note: When reconfiguring between 10G and 25G data rate, each channel may require two additional TOD synchronizers to synchronize two different slaves TOD frequencies, 156.25 MHz frequency for 10G data rate and 390.625 MHz frequency for 25G data rate. Refer to the *Clocks* section for more details on clock connections and required frequencies.

Table 14. TOD Module Required Connections - 10G/25G Advanced Mode

Required connections for TOD module, listed using signal names of TOD modules. If you create your own TOD module it must have the output signals required by the E-Tile Hard IP for Ethernet Intel FPGA IP. However, its signal names could be different than the TOD module signal names in the table.

TOD Module Signals	Advanced Mode Recommended Connections
TX TOD Module Signals	
clk	Avalon memory-mapped interface clock
rst_n	Avalon memory-mapped interface reset
period_clk	o_clk_pll_div66
period_rst_n	Global/system reset
time_of_day_96b[95:0]	i_sl_ptp_tx_tod[95:0]
RX TOD Module Signals	
clk	Avalon memory-mapped interface clock
rst_n	Avalon memory-mapped interface reset
period_clk	o_clk_rec_div66
period_rst_n	Global/system reset
time_of_day_96b[95:0]	i_sl_ptp_rx_tod[95:0]

For TOD Synchronizer IP connections, refer to the *Ethernet Design Example Components User Guide*.

Related Information

- E-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
- E-Tile Hard IP for Ethernet Intel Agilex FPGA IP Design Example User Guide
- PTP Timestamp Accuracy on page 83
  Information about the Basic and Advanced Modes
- Clock Connection Guidelines for 10/25GE with PTP and without External AIB Clocking on page 173
- Ethernet Design Example Components User Guide
  Information about the Ethernet IEEE 1588 Time of Day Clock Intel FPGA IP.

2.7.4.5. SDC for Multiple E-Tile Instances

Refer to the Hold Timing Violation section in the *E-Tile Transceiver PHY User Guide*.

2.7.5. IP Core Testbenches

Intel provides a compilation-only design example and a testbench that you can generate for the E-Tile Hard IP for Ethernet Intel FPGA IP.

To generate the testbench, in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor, you must first set the parameter values for the IP core variation you intend to generate in your end product. If you do not set the parameter values for your DUT to match the parameter values in your end product, the testbench you generate does not exercise the IP core variation you intend.

The testbench demonstrates a basic test of the IP core. It is not intended to be a substitute for a full verification environment.

Related Information

- E-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
- E-Tile Hard IP for Ethernet Intel Agilex FPGA IP Design Example User Guide

2.7.6. Compiling the Full Design

You can use the **Start Compilation** command on the Processing menu in the Intel Quartus Prime Pro Edition software to compile your design.

Note: Intel does not guarantee timing closure, you may need to add extra timing constraint for your Ethernet Design. For additional timing constraint, refer to the *E-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide* or *E-Tile Hard IP for Ethernet Intel Agilex FPGA IP Design Example User Guide*.

Related Information

- Block-Based Design Flows
- Programming Intel FPGA Devices

2.8. E-Tile Hard IP for Ethernet Intel FPGA IP Parameters

Related Information

- E-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
- E-Tile Hard IP for Ethernet Intel Agilex FPGA IP Design Example User Guide

2.8.1. Parameter Editor Parameters

The E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor provides the parameters you can set to configure your E-Tile Hard IP for Ethernet Intel FPGA IP variation and simulation and hardware design examples.

The E-Tile Hard IP for Ethernet Intel FPGA IP parameter has three tabs, an **IP** tab, 10GE/25GE and/or 100GE Tab, and an **Example Design** tab. For information about the **Example Design** tab, refer to the *E-Tile Hard IP for Ethernet Intel FPGA IP Design Example User Guide*.

Table 15. E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: IP Tab

This table does not provide information about invalid parameter value combinations. If you make selections that create a conflict, the parameter editor generates error messages in the **System Messages** pane.

Parameter	Range	Default Setting	Parameter Description
Core Options			
Select Core Variant	• Single 10GE/25GE • 1 to 4 10GE/25GE with optional RSFEC • Single 100GE with optional RSFEC • 100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP • Custom PCS with optional RSFEC	Single 10GE/25GE	Select a variant of the E-tile Ethernet core with the types of channels required. If you choose the **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP** variant, you can select to use a 100GE channel or 1 to 4 10GE/25GE channels with or without RS-FEC and/or IEEE 1588 timestamps. These options are switchable at compile time or run time.
Number of Channels of 10GE/25GE	• Single Channel • 2 Channels • 3 Channels • 4 Channels	Single Channel	Set the number of channels when you select variants that allow 1 to 4 channels. Resources such as RS-FEC and PTP are more efficient if shared. If your design requires multiple 10GE/25GE channels, consider increasing the number of channels in the core to share resources more efficiently.
Active Channel(s) at startup	• 10GE/25GE Channel(s) • 100GE Channel	10GE/25GE Channel(s)	If you choose the **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP**, select the channel (100GE or 10GE/25GE) to be connected to the transceivers at start-up.
Enable IEEE 1588 PTP	• On • Off	Off	Turn on this parameter to add IEEE 1588 PTP Timestamp offload functions to the core. The core can generate 1-step or 2-step TX timestamps and RX timestamps. This option supports 10/25G and 100G with RS-FEC(528,514) variants. *Note:* For 10/25G variants, user transmitted PTP packets must be more than 32 bytes, or else it may break the functionality of the IP core. *Note:* 25G with PTP supports only single channel variants if you enable auto-negotiation and link training feature. *Note:* Timestamps require some soft logic. To enable the soft logic. Connect the core to a PTP Time-of-Day module that produces TOD values using the 96b IEEE 1588v2 time format.
IEEE1588/PTP channel placement restriction	• EHIP0/2 • EHIP1/3	EHIP0/2	Selects the Ethernet Hard IP core to be used when PTP is enabled. This selection determines the PTP channel placement within an E-tile transceiver: • EHIP0/2 uses PTP channel 4,5,16, and 17 • EHIP1/3 uses PTP channel 6,7,18, and 19
Enable SyncE	• On • Off	Off	Turn on this parameter to enable Synchronous Ethernet (SyncE) operation. This parameter is only available for 10G and 25G variant(4). *Note:* This parameter is not available when **Enable IEEE 1588 PTP** is turned on.

continued...

Parameter	Range	Default Setting	Parameter Description
Enable RS-FEC	• On • Off	Off	Turn on this parameter to include additional hard logic to perform Reed-Solomon Forward Error Correction (RS-FEC). *Note:* 25G with RS-FEC supports only single channel variants when you enable auto-negotiation and link training feature unless **Enable external AIB clocking** is enabled. This feature is not supported for 10G variants.
First RSFEC Lane	• first_lane0 • first_lane1 • first_lane2 • first_lane3	first_lane0	Selects the first RS-FEC lane to be used. There are four lanes in the RS-FEC block. When the RS-FEC block is in fractured mode, any of the four lanes may be selected as the first lane. For multiple channel Native PHY IP core instances with the RS-FEC block enabled, the RS-FEC lanes used must be contiguous and must fit within a single four-channel RS-FEC block.
RSFEC Clocking Mode	• fec_dir_adp_clk_0 • fec_dir_adp_clk_1 • fec_dir_adp_clk_2 • fec_dir_adp_clk_3	fec_dir_adp_clk_0	Sets the clocking mode for the RS-FEC block. For RS-FEC with PTP enabled topologies, the clock selection is fixed. In all other cases, this control selects the TX adapter clock used to clock the RS-FEC block.
Enable AN/LT	• On • Off	Off	Turn on this parameter to enable the IP core to support auto-negotiation as defined in *IEEE Standard 802.3-2015* Clause 73 and the *25G/50G Ethernet Consortium Schedule Draft 1-6*, and link training as defined in *IEEE Standard 802.3-2015* Clauses 92 and 93 and the *25G/50G Ethernet Consortium Schedule Draft 1-6*. *Note:* Multi-channel 25GE with RS-FEC with **Enable AN/LT** is supported only if **Enable external AIB clocking** is enabled. To use AN in loopback mode, set the **Ignore Nonce Field** to 1 (*Auto Negotiation Config Register 1*, offset 0xC0, bit [7]), which disables the Nonce bit checking.
Enable external AIB clocking	• On • Off	Off	Turn on this parameter to enable additional `i_clk_aib` and `i_clk_aib_2x` signals to allow external clock sources to drive the datapath in the EHIP core and the EMIB block. Enables this option for the multi-channel 25G with RS-FEC when **Enable AN/LT** is set. *Important:* When you enable this parameter in multi channels 25G variants, triggering a reset to the master channel's EMIB interface impacts the slave channels operation.
PTP Options			
PTP Accuracy Mode	• Basic Mode • Advanced Mode	Basic Mode	When selected, specifies the PTP accuracy for selected Ethernet variant. In 10G/25G Ethernet variant:

continued...

(4) Currently, the SyncE feature is not available in the Intel Stratix 10 E-tile Hardware Design Example.

Parameter	Range	Default Setting	Parameter Description
			• Basic Mode: PTP accuracy is ± 3 ns • Advanced Mode: PTP accuracy is ±1.5 ns In 100G Ethernet variant: • Basic Mode: PTP accuracy is ± 8 ns • Advanced Mode: PTP accuracy is not supported.
AN/LT Options			
Auto-Negotiation			
Enable Auto-Negotiation on Reset	• On • Off	On	If this parameter is turned on, the IP core is configured after reset to implement auto-negotiation as defined in Clause 73 of *IEEE Std 802.3–2015*. If this parameter is turned off, the IP core does not perform the auto-negotiation after reset. Instead, the auto-negotiation can be re-enabled by control and status register (CSR) setting.
Link Fail Inhibit Time	100–4000 ms	504 ms	Specifies the time before link status is set to FAIL or OK. A link fails if the time duration specified by this parameter expires before link status is set to OK. For more information, refer to *Clause 73 Auto-Negotiation for Backplane Ethernet* in *IEEE Standard 802.3–2015*. The IP core asserts the `o_rx_pcs_ready` signal to indicate link status is OK. In simulation, the default value is 504 corresponds to 1.6 ms. In hardware, the default value for variants with RS-FEC(544,514) is set to 3 seconds.
Advertise CR Technology Ability	• On • Off	On	If this parameter is turned on, the IP core advertises CR capability by default. If this parameter is turned off, but auto-negotiation is turned on, the IP core advertises KR capability by default.
Request RSFEC	• On • Off	On	Turn on this parameter to request RS-FEC from remote link partner during auto-negotiation. This parameter must be turned on when **Enable RSFEC** is on in order to use KR functionality.
Enable Dynamic RSFEC for KR	• On • Off	Off	When selected, IP enables the ability to switch from RS-FEC Enabled Mode to RS-FEC Disabled Mode. *Note:* Only available if **Enable RSFEC** is selected. In 25G variants, **First RSFEC Lane** must be set to first_lane0.
Auto-Negotiation Master	• Lane 0 • Lane 1 • Lane 2 • Lane 3	Lane 0	Selects the master channel for auto-negotiation. The IP core allows you to change the master channel dynamically by configuring the CSR setting. Available in 100G modes. In 100G PAM4 mode, the valid selections are Lane0 and Lane2.
Advertise both 10G and 25G during AN	• On • Off	Off	Turn on this parameter to advertise 10 and 25 Gbps data rate during auto-negotiation. When this parameter is turned off, the IP core advertises only the data rate specified in the **Select Ethernet Rate** parameter in the 10GE/25GE tab.

continued...

Parameter	Range	Default Setting	Parameter Description
			This parameter is not available with **Single 100GE with optional RSFEC** variant or when **100GE Channel** is selected as **Active Channel(s) at startup**. This feature is not compatible with the PTP, RS-FEC, or external AIB clocking.
Advertise Pause ability	• On • Off	On	If this parameter is turned on, the IP core indicates on the Ethernet link that it supports symmetric pauses as defined in *Annex 28B* of Section 2 of *IEEE Std 802.3–2015*.
Advertise Pause ASM_DIR ability	• On • Off	On	If this parameter is turned on, the IP core indicates on the Ethernet link that it supports asymmetric pauses as defined in *Annex 28B* of Section 2 of *IEEE Std 802.3–2015*.
Link Training: General			
Enable Link Training on Reset	• On • Off	On	If this parameter is turned on, the IP core is configured after reset to perform link training, which configures the remote link partner TX PMD for the lowest Bit Error Rate (BER). LT is defined in Clause 92 of *IEEE Std 802.3–2015*.
Configuration, Debug and Extension Options			
Enable Native PHY Debug Master Endpoint	• On • Off	On	If this parameter is turned on, the Transceiver Native PHY IP includes an embedded Native PHY Debug Master Endpoint that connects internally to the Avalon memory-mapped slave interface for dynamic reconfiguration. The Native PHY Debug Master Endpoint can access the reconfiguration space of the transceiver. It can perform certain test and debug functions via JTAG using the System Console.
Enable JTAG to Avalon Master Bridge	• On • Off	Off	Turn on this parameter to enable an internal JTAG connection to the Avalon memory-mapped Master Bridge for register reconfigurations. This connection allows the System Console to run the Ethernet Link Inspector.

Table 16. E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: 10GE/25GE Tab

This table does not provide information about invalid parameter value combinations. If you make selections that create a conflict, the parameter editor generates error messages in the **System Messages** pane.

Parameter	Range	Default Setting	Parameter Description
General Options 10GE/25GE			
Select Ethernet Rate	• 10G • 25G	25G	Selects the IP core Ethernet data rate.
Select Ethernet IP Layers	When **Enable RSFEC** and **Enable IEEE 1588 PTP** are off: • MAC+PCS • PCS Only • OTN • FlexE	MAC+PCS	Selects the Ethernet Protocol layers provided by the channel. *Note:* RS-FEC is not supported in the 10G data rate.

continued...

Parameter	Range	Default Setting	Parameter Description
	When **Enable RSFEC** is ON and **Enable IEEE 1588 PTP** is OFF: • MAC+PCS+RS-FEC • PCS+RS-FEC • OTN+RS-FEC • FlexE+RS-FEC	MAC+PCS+RS-FEC	
	When **Enable RSFEC** and **Enable IEEE 1588 PTP** are on: • MAC+PTP+PCS +RS-FEC	MAC+PTP+PCS +RSFEC	
Include alternate ports	• On • Off	Off	This is an advanced option for applications that need to change the active Ethernet IP layers at run-time. When you turn on this option, all possible datapath interfaces are included in the core, and the active interface depends on the control and status register (CSR) setting.
MAC Options: Basic 10GE/25GE *Note:* In PCS Only, OTN, and FlexE variations, these parameters have no effect.			
TX Maximum Frame Size	65–65535	1518	Maximum packet size (in bytes) the IP core can transmit on the Ethernet link without reporting an oversized packet in the TX statistics counters. In variations without MAC, this parameter has no effect and remains at the default value of 1518.
RX Maximum Frame Size	65–65535	1518	Maximum packet size (in bytes) the IP core can receive on the Ethernet link without reporting an oversized packet in the RX statistics counters. If you turn on the **Enforce Maximum Frame Size** parameter, the IP core truncates incoming Ethernet packets that exceed this size. In variations without MAC, this parameter has no effect and remains at the default value of 1518.
Enforce Maximum Frame Size	• On • Off	Off	Specifies whether the IP core is able to receive an oversized packet or truncates these packets.
Choose Link Fault Generation Mode	• OFF • Unidirectional • Bidirectional	Bidirectional	Specifies the IP core response to link fault events. Bidirectional link fault handling complies with the Ethernet specification, specifically IEEE 802.3 Figure 81-11. Unidirectional link fault handling implements IEEE 802.3 Clause 66: in response to local faults, the IP core transmits Remote Fault ordered sets in interpacket gaps but does not respond to incoming Remote Fault ordered sets. The **OFF** option is provided for backward compatibility.
Stop TX traffic when link partner sends pause	• Yes • No • Disable Flow Control	No	Selects whether the IP core responds to PAUSE frames from the Ethernet link by stopping TX traffic, or not. This parameter has no effect if flow control is disabled. If you disable flow control, the IP core neither responds to incoming PAUSE and PFC frames nor generates outgoing PAUSE and PFC frames.

continued...

Parameter	Range	Default Setting	Parameter Description
			If this parameter has the value of **No**, you can use the `i_tx_pause` signal on the TX client interface to force the TX MAC to stop TX traffic. Bytes to remove from RX frames
Bytes to remove from RX frames	• None • Remove CRC bytes • Remove CRC and PAD bytes	Remove CRC bytes	You can set for the RX MAC to remove CRC and/or PAD bytes from incoming RX frames before passing the bytes to the RX MAC Client. If the PAD and CRC bytes are not needed downstream, the remove option can reduce the need for downstream packet processing logic.
Forward RX Pause Requests	• On • Off	Off	Selects whether the RX MAC forwards incoming PAUSE and PFC frames on the RX client interface, or drops them after internal processing. *Note:* If flow control is turned off, the IP core forwards all incoming PAUSE and PFC frames directly to the RX client interface and performs no internal processing. In that case this parameter has no effect.
Use Source Address Insertion	• On • Off	Off	Selects whether the IP core supports overwriting the source address in an outgoing Ethernet packet with the value in the TXMAC_SADDR registers at offsets 0x40C and 0x40D. If the parameter is turned on, the IP core overwrites the packet source address from the register if `i_tx_skip_crc` has the value of 0. If the parameter is turned off, the IP core does not overwrite the source address. Source address insertion applies to PAUSE and PFC packets provided on the TX MAC client interface, but does not apply to PAUSE and PFC packets the IP core transmits in response to the assertion of `i_tx_pause` or `i_tx_pfc[n]` on the TX MAC client interface.
Enable TX VLAN Detection	• On • Off	On	Specifies whether the IP core TX statistics block treats TX VLAN and Stacked VLAN Ethernet frames as regular control frames, or performs Length/Type field decoding, includes these frame in VLAN statistics, and counts the payload bytes instead of the full Ethernet frame in the `TxFrameOctetsOK` counter at offsets 0x862 and 0x863. If turned on, the IP core identifies these frames in TX statistics as VLAN or Stacked VLAN frames. If turned off, the IP core treats these frames as regular control frames.
Enable RX VLAN Detection	• On • Off	On	Specifies whether the IP core RX statistics block treats RX VLAN and Stacked VLAN Ethernet frames as regular control frames, or performs Length/Type field decoding, includes these frame in VLAN statistics, and counts the payload bytes instead of the full Ethernet frame in the `RxFrameOctetsOK` counter at offsets 0x962 and 0x963. If turned on, the IP core identifies these frames in RX statistics as VLAN or Stacked VLAN frames. If turned off, the IP core treats these frames as regular control frames.
Ready latency	0-3	0	Selects the readyLatency value on the TX client interface. readyLatency is an Avalon streaming interface property that defines the number of

continued...

Parameter	Range	Default Setting	Parameter Description
			clock cycles of delay from when the IP core asserts the `o_sl_tx_ready` signal to the clock cycle in which the IP core can accept data on the TX client interface. Refer to the Avalon Interface Specifications. In PCS Only, OTN, and FlexE variations, this parameter has no effect. Selecting a longer latency (higher number) eases timing closure at the expense of increased latency for the TX datapath in MAC +PCS variations.
Enable asynchronous adapter clocks(5)	• On • Off	Off	Turn on if you want to drive TX/RX interface using clock source different from `i_sl_clk_rx`/`i_sl_clk_tx`. The different clock source is driven through another clock port (`i_sl_async_clk_rx`/ `i_sl_async_clk_tx`). Supported clock frequency range is between 390.625 MHz and 402.83203125 MHz.
MAC Options: Specialized 10GE/25GE *Note:* In PCS Only, OTN, and FlexE variations, these parameters have no effect.			
Enable preamble passthrough	• On • Off	Off	If turned on, the IP core is in RX and TX preamble pass-through mode. In RX preamble pass-through mode, the IP core passes the preamble and SFD to the client instead of stripping them out of the Ethernet packet. In TX preamble pass-through mode, the client specifies the preamble to be sent in the Ethernet frame.
Enable strict preamble check	• On • Off	Off	If turned on, the IP core rejects RX packets whose preamble is not the standard Ethernet preamble (0x55_55_55_55_55_55). This option provides an additional layer of protection against spurious Start frames that can occur at startup or when bit errors occur.
Enable strict SFD check	• On • Off	Off	If turned on, the IP core rejects RX packets whose SFD byte is not the standard Ethernet SFD (0xD5). This option provides an additional layer of protection against spurious Start frames that can occur at startup or when bit errors occur.
Average Inter-packet Gap	• 1 • 8 • 10 • 12	12	Specifies the average minimum inter-packet gap (IPG) the IP core maintains on the TX Ethernet link.Specifies the average minimum inter-packet gap (IPG) the IP core maintains on the TX Ethernet link. The default value of 12 complies with the Ethernet standard. The remaining values support increased throughput. The value of 1 specifies that the IP core transmits Ethernet packets as soon as the data is available, with the minimum possible gap. The IPG depends on the space you leave between frame data as you write it to the core. The IP core no longer complies with the

continued...

(5) In 10G variant, the asynchronous adapter clock is available only when PTP is enabled.

Parameter	Range	Default Setting	Parameter Description
			Ethernet standard but the application has control over the average gap and maximizing the throughput. For more information, refer to the *Inter-Packet Gap Generation and Insertion* section.
Additional IPG removed per AM period	Integer	0	Specifies the number of inter-packet gaps the IP core removes per alignment marker period, in addition to the default number required for protocol compliance. Each increment of 1 in the value of **Additional IPG removed per AM period** increases throughput by 3ppm in 100G variations. To specify larger throughput increases, use the **Average Inter-packet Gap** parameter.
PMA Options 10GE/25E			
PHY Reference Frequency	• 156.25 MHz • 322.265625 MHz • 312.5 MHz(6) • 644.53125 MHz (6)	322.265625 MHz	Sets the expected incoming PHY `i_clk_ref` reference frequency. The input clock frequency must match the frequency you specify for this parameter (±100 ppm). *Note:* If you turn on **Enable AN/LT**, the required input clock frequency are 156.25 or 312.5 MHz.
Enable custom rate	• On • Off	Off	Turn on to enable custom rate. *Note:* This parameter is not available when **Enable SyncE** is turned on.
Include refclk_mux	• On • Off	Off	Turn on to increase the number of allowed reference clocks. The reference clocks connect to the channel's input. By default, all channels are connected to the `refclk[0]` reference clock. To reassign a channel to a different reference clock, program the refclk_mux_sel register.
Include deterministic latency measurement interface	• On • Off	Off	This feature is available for internal use only.

Table 17. E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: 100GE Tab

This table does not provide information about invalid parameter value combinations. If you make selections that create a conflict, the parameter editor generates error messages in the **System Messages** pane.

Parameter	Range	Default Setting	Parameter Description
General Options			
Select Ethernet Rate	100G	100G	Selects the IP core Ethernet data rate.
Select Ethernet IP Layers	• MAC+PCS • MAC+PTP+PCS • MAC+PCS +(528,514) RS-FEC	MAC+PCS	Selects the Ethernet Protocol layers provided by the channel. *Note:* The E-Tile Hard IP for Ethernet Intel FPGA IP provides support for the OTN feature. For further inquiries, contact your nearest Intel sales representative.

continued...

(6) Not supported for 10G/25G mode. For more information refer to this knowledge base.

Parameter	Range	Default Setting	Parameter Description
	• MAC+PCS +(544,514) RS-FEC • MAC+PTP+PCS +(528,514) RS-FEC • PCS Only • PCS+(528,514) RS-FEC • PCS+(544,514) RS-FEC • OTN • OTN+(528,514) RS-FEC • OTN+(544,514) RS-FEC • FlexE • FlexE+(528,514) RS-FEC • FlexE+(544,514) RS-FEC		
MAC Options: Basic 100GE *Note:* In PCS Only, OTN, and FlexE variations, these parameters have no effect.			
TX Maximum Frame Size	65–65535	1518	Maximum packet size (in bytes) the IP core can transmit on the Ethernet link without reporting an oversized packet in the TX statistics counters. In variations without MAC, this parameter has no effect and remains at the default value of 1518.
RX Maximum Frame Size	65–65535	1518	Maximum packet size (in bytes) the IP core can receive on the Ethernet link without reporting an oversized packet in the RX statistics counters. If you turn on the **Enforce Maximum Frame Size** parameter, the IP core truncates incoming Ethernet packets that exceed this size. In variations without MAC, this parameter has no effect and remains at the default value of 1518.
Enforce Maximum Frame Size	• On • Off	Off	Specifies whether the IP core is able to receive an oversized packet or truncates these packets.
Choose Link Fault Generation Mode	• OFF • Unidirectional • Bidirectional	Bidirectional	Specifies the IP core response to link fault events. Bidirectional link fault handling complies with the Ethernet specification, specifically IEEE 802.3 Figure 81-11. Unidirectional link fault handling implements IEEE 802.3 Clause 66: in response to local faults, the IP core transmits Remote Fault ordered sets in interpacket gaps but does not respond to incoming Remote Fault ordered sets. The **OFF** option is provided for backward compatibility.
Stop TX traffic when link partner sends pause	• Yes • No • Disable Flow Control	No	Selects whether the IP core responds to PAUSE frames from the Ethernet link by stopping TX traffic, or not. This parameter has no effect if flow control is disabled. If you disable flow

continued...

Parameter	Range	Default Setting	Parameter Description
			control, the IP core neither responds to incoming PAUSE and PFC frames nor generates outgoing PAUSE and PFC frames. If this parameter has the value of **No**, you can use the `i_tx_pause` signal on the TX client interface to force the TX MAC to stop TX traffic.
Bytes to remove from RX frames	• None • Remove CRC bytes • Remove CRC and PAD bytes	Remove CRC bytes	You can set for the RX MAC to remove CRC and/or PAD bytes from incoming RX frames before passing the bytes to the RX MAC Client. If the PAD and CRC bytes are not needed downstream, the remove option can reduce the need for downstream packet processing logic.
Forward RX Pause Requests	• On • Off	Off	Selects whether the RX MAC forwards incoming PAUSE and PFC frames on the RX client interface, or drops them after internal processing. *Note:* If flow control is turned off, the IP core forwards all incoming PAUSE and PFC frames directly to the RX client interface and performs no internal processing. In that case this parameter has no effect.
Use Source Address Insertion	• On • Off	Off	Selects whether the IP core supports overwriting the source address in an outgoing Ethernet packet with the value in the `TXMAC_SADDR` registers at offsets 0x40C and 0x40D. If the parameter is turned on, the IP core overwrites the packet source address from the register if `i_tx_skip_crc` has the value of 0. If the parameter is turned off, the IP core does not overwrite the source address. Source address insertion applies to PAUSE and PFC packets provided on the TX MAC client interface, but does not apply to PAUSE and PFC packets the IP core transmits in response to the assertion of `i_tx_pause` or `i_tx_pfc[n]` on the TX MAC client interface.
Enable TX VLAN Detection	• On • Off	On	Specifies whether the IP core TX statistics block treats TX VLAN and Stacked VLAN Ethernet frames as regular control frames, or performs Length/Type field decoding, includes these frame in VLAN statistics, and counts the payload bytes instead of the full Ethernet frame in the `TxFrameOctetsOK` counter at offsets 0x862 and 0x863. If turned on, the IP core identifies these frames in TX statistics as VLAN or Stacked VLAN frames. If turned off, the IP core treats these frames as regular control frames.
Enable RX VLAN Detection	• On • Off	On	Specifies whether the IP core RX statistics block treats RX VLAN and Stacked VLAN Ethernet frames as regular control frames, or performs Length/Type field decoding, includes these frame in VLAN statistics, and counts the payload bytes instead of the full Ethernet frame in the `RxFrameOctetsOK` counter at offsets 0x962 and 0x963. If turned on, the IP core identifies these frames in RX statistics as VLAN or Stacked VLAN frames. If turned off, the IP core treats these frames as regular control frames.

continued...

Parameter	Range	Default Setting	Parameter Description
Enable asynchronous adapter clocks	• On • Off	Off	Turn on if you want to drive `i_clk_rx` and `i_clk_tx` clocks from different clock sources. *Note:* For 100GbE, the asynchronous adapter clocks are only available when PTP is disabled.
Ready latency	0-3	0	Selects the readyLatency value on the TX client interface. readyLatency is an Avalon streaming interface property that defines the number of clock cycles of delay from when the IP core asserts the `o_tx_ready` signal to the clock cycle in which the IP core can accept data on the TX client interface. Refer to the Avalon Interface Specifications. This feature only supports MAC+PCS variant. In PCS Only, OTN, and FlexE variations, this parameter has no effect. Selecting other variations produces an error. Selecting a longer latency (higher number) eases timing closure at the expense of increased latency for the TX datapath in MAC +PCS variations.
MAC Options: Specialized 100GE *Note:* In PCS Only, OTN, and FlexE variations, these parameters have no effect.			
Enable preamble passthrough	• On • Off	Off	If turned on, the IP core is in RX and TX preamble pass-through mode. In RX preamble pass-through mode, the IP core passes the preamble and SFD to the client instead of stripping them out of the Ethernet packet. In TX preamble pass-through mode, the client specifies the preamble to be sent in the Ethernet frame.
Enable strict preamble check	• On • Off	Off	If turned on, the IP core rejects RX packets whose preamble is not the standard Ethernet preamble (0x55_55_55_55_55_55). This option provides an additional layer of protection against spurious Start frames that can occur at startup or when bit errors occur.
Enable strict SFD check	• On • Off	Off	If turned on, the IP core rejects RX packets whose SFD byte is not the standard Ethernet SFD (0xD5). This option provides an additional layer of protection against spurious Start frames that can occur at startup or when bit errors occur.
Average Inter-packet Gap	• 1 • 8 • 10 • 12	12	Specifies the average minimum inter-packet gap (IPG) the IP core maintains on the TX Ethernet link.Specifies the average minimum inter-packet gap (IPG) the IP core maintains on the TX Ethernet link. The default value of 12 complies with the Ethernet standard. The remaining values support increased throughput. The value of 1 specifies that the IP core transmits Ethernet packets as soon as the data is available, with the minimum possible gap. The IPG depends on the space you leave between frame data as you write it to the core. The IP core no longer complies with the Ethernet standard but the application has control over the average gap and maximizing

continued...

Parameter	Range	Default Setting	Parameter Description
			the throughput. For more information, refer to the *Inter-Packet Gap Generation and Insertion* section.
Additional IPG removed per AM period	Integer	0	Specifies the number of inter-packet gaps the IP core removes per alignment marker period, in addition to the default number required for protocol compliance. In 100G variations, the default number is 20. Each increment of 1 in the value of **Additional IPG removed per AM period** increases throughput by 3ppm in 100G variations. To specify larger throughput increases, use the **Average Inter-packet Gap** parameter.
PMA Options 100GE			
Preserve Unused Transceiver Channels(7)	• On • Off	Off	Preserves the unused PAM4 slave channel when you select variant with (544,514) RS-FEC option.
Reference Clock Frequency for Preserved Channels	125 MHz to 500 MHz	125 MHz	When the **Preserve Unused Transceiver Channels** is enabled, the IP core adds additional reference clock to the preserve unused PAM4 channel. Set the reference clock value within the available frequency range.
PHY Reference Frequency	• 156.25 MHz • 322.265625 MHz • 312.5 MHz • 644.53125 MHz	156.25 MHz	Sets the expected incoming PHY `i_clk_ref` reference frequency. The input clock frequency must match the frequency you specify for this parameter (±100 ppm). Variants with (544,514) RS-FEC option only support 156.25 MHz and 312.5 MHz PHY `i_clk_ref` reference frequency. *Note:* If you turn on **Enable AN/LT**, the required input clock frequency are 156.25 or 312.5 MHz.

Table 18. E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: Custom PCS Channel(s) Tab

This table does not provide information about invalid parameter value combinations. If you make selections that create a conflict, the parameter editor generates error messages in the **System Messages** pane.

Parameter	Range	Default Setting	Parameter Description
PCS Core Options			
Number of PCS Channels in core	• 1 • 2 • 3 • 4	1	Set the number of PCS channels you want to implement. Resources such as RS-FEC and PTP are more efficient if shared. If your design requires multiple channels, consider increasing the number of channels in the core to share resources more efficiently.
PCS General Options			

continued...

(7) For more information about preserving unused transceiver channels in high-speed PAM4 mode, refer to the *E-Tile Transceiver PHY User Guide.*

Parameter	Range	Default Setting	Parameter Description
Custom PCS mode	• PCS_Only • PCS+RS-FEC	PCS_Only	Selects the Ethernet Protocol layers provided by the channel.
RSFEC Fibre Channel(s) mode	• Disable • Enable	Disable	To enable or disable RS-FEC Fibre Channel mode for custom PCS.
Custom PCS Rate	2500 to 28000 Mbps	2500 Mbps	Specifies the transceiver TX data rate in megabits per second (Mbps) unit.
PMA Options			
PMA modulation type	NRZ	NRZ	Specifies the type of modulation for TX serial data.
PMA reference clock frequency	• 500.000000 • 312.500000 • 277.777777 • 250.000000 • 227.272727 • 208.333333 • 192.307692 • 178.571428 • 166.666666 • 156.250000 • 147.058823 • 138.888888 • 131.578947 • 125.000000	250.000000	Sets the custom PCS reference clock frequency.
Enable custom rate regulation	• On • Off		Turn on this option to add the custom rate ports to your design. You are required to drive the ports with an appropriate flow regulation signal.

For parameters in the **PMA Adaptation** tab, refer to the *PMA Adaptation* topic in the *E-Tile Transceiver PHY User Guide.*

Related Information

- E-Tile Transceiver PHY User Guide: PMA Parameters
  Information about PMA Adaptation parameters.
- E-Tile Transceiver PHY User Guide: Dynamic Reconfiguration Examples
  Information about configuring PMA parameters.

2.8.2. RTL Parameters

The E-Tile Hard IP for Ethernet Intel FPGA IP provides parameters in the generated RTL that you can modify for your IP core instance. Generating an IP core variation from the parameter editor creates an RTL module. Your design might instantiate multiple instances of this module. You can specify RTL parameter values for each instance. Each RTL parameter determines the initial and reset value of one or more register fields in the IP core.

RTL parameters allow you to customize your IP core instance to vary from the defaults you selected for your IP core variation and from other instances of the same IP core variation. This capability allows you to fine-tune your design without regenerating and without reading and writing registers following power-up. In addition, you can specify parameter values that should not be identical for multiple instances. For example, you can specify a different TX source address for each instance, without having to write to the relevant registers.

To access the RTL parameters, refer to the IP configuration and test files. The simulation-based RTL parameters are located in `<your_project_directory>\<your_IP_name>\sim\<your_IP_name>.v` The synthesis-based RTL parameters are located in `<your_project_directory>\<your_IP_name>\synth\<your_IP_name>.v`.

Table 19. E-Tile Hard IP for Ethernet Intel FPGA IP RTL Parameters

Parameter	Parameter Description
Parameters Available for all IP Core Variations	
`sim_mode`	Specifies whether the IP core is in simulation mode, in which alignment marker periods are shortened to decrease the time to RX PCS alignment. • Value `disable` (default value): The IP core MAC implements standard alignment marker periods as specified in the *IEEE Standard 802.3–2015*. Before compiling for synthesis, ensure this parameter has this value. • Value `enable`: The IP core implements shorter alignment marker periods to accelerate RX PCS alignment in simulation. The simulation link partner must have the same alignment marker periods. This mode is intended for simulation only. The value of this parameter determines the initial and reset values of these register fields: • `am_interval[13:0]` field (bits [13:0]) of the RXPCS_CONF register at 0ffset 0x360. • `am_period[15:0]` field (bits [31:16]) of the TXMAC_EHIP_CFG register at 0ffset 0x40B.
Parameters Available for MAC+PCS IP Core Variations Only	
`rx_pause_daddr`	Sets the destination addresses for PAUSE and PFC frames. The RX MAC uses this address to filter whether incoming PAUSE and PFC frames apply to the current IP core. • Default value is 0x01_80_C2_00_00_01, the Ethernet standard multicast address for PAUSE and PFC. • Range is 0 through 2^{48}–1. • Value can be a unicast or multicast address. • The RX MAC processes PAUSE and PFC frames only if their destination address matches this address (actually, the address in the RX_PAUSE_DADDR registers). The value of this parameter determines the initial and reset values of the RX_PAUSE_DADDR registers at offsets 0x707 and 0x708.
`source_address_insertion`	Selects whether the IP core supports overwriting the source address in an outgoing packet it receives on the TX MAC interface, with the value in the TXMAC_SADDR registers at offsets 0x40C and 0x40D. • The default value is the value of the parameter editor **Use Source Address Insertion** parameter. • Value `enable`: If `i_tx_skip_crc` has the value of 0, in packets the IP core receives on the TX MAC client interface, the TX MAC overwrites the source address field with the value in the TXMAC_SADDR registers at offsets 0x40C and 0x40D. *Note:* The IP core does not overwrite the source address in Ethernet PAUSE and PFC packets it generates on the Ethernet link in response to assertion of the `i_tx_pause` signal or an `i_tx_pfc[n]` signal on the TX MAC client interface. • Value `disable`: The TX MAC does not overwrite the source address field in packets it receives on the TX MAC client interface. The value of this parameter determines the initial and reset values of the `en_saddr_insert` field (bit [3]) of the TXMAC_CONTROL register at 0ffset 0x40A.

continued...

Parameter	Parameter Description
tx_pause_daddr	Sets the destination addresses that the TX MAC inserts in PAUSE and PFC frames that the IP core transmits on the Ethernet link in response to assertion of the i_tx_pause signal or an i_tx_pfc[n] signal on the TX MAC client interface. • Default value is 0x01_80_C2_00_00_01, the Ethernet standard multicast address for PAUSE and PFC. • Range is 0 through 2^{48}–1. • Value can be a unicast or multicast address. The value of this parameter determines the initial and reset values of the TX_PFC_DADDR registers at offsets 0x60D and 0x60E.
tx_pause_saddr	Sets the source addresses that the TX MAC inserts in PAUSE and PFC frames that the IP core transmits on the Ethernet link in response to assertion of the i_tx_pause signal or an i_tx_pfc[n] signal on the TX MAC client interface. • Default value is the value of the RTL parameter txmac_saddr, which is the initial source address the IP core inserts in all TX packets written to the TX MAC client interface when source MAC address insertion is enabled. • Range is 0 through 2^{48}–1. • Value should be a unicast address. The value of this parameter determines the initial and reset values of the TX_PFC_SADDR registers at offsets 0x60F and 0x610.
txmac_saddr	Sets the source addresses that the TX MAC inserts in packets written to the TX MAC client interface when source MAC address insertion is enabled. • Default value is the value you specify for the parameter editor **TX MAC Source Address** parameter. • Range is 0 through 2^{48}–1. • The Intel FPGA team recommends you program each IP core instance with a unique unicast MAC address. The value of this parameter determines the initial and reset values of the TXMAC_SADDR registers at offsets 0x40C and 0x40D.

2.9. Functional Description

The E-Tile Hard IP for Ethernet Intel FPGA IP MAC+PCS variations implement an Ethernet MAC in accordance with the *IEEE 802.3 Ethernet Standard*. The IP core handles the frame encapsulation and flow of data between client logic and an Ethernet network through a 10-Gbps, 25-Gbps, and 100-Gbps Ethernet PHY implemented in hard IP, with optional Reed Solomon Forward Error Correction (RS-FEC).

In the transmit direction, the MAC accepts client frames, and inserts inter-packet gap (IPG), preamble, start of frame delimiter (SFD), padding, and CRC bits before passing them to the PHY. You can configure the MAC to accept some of the additions with the client frame. The MAC also updates the TX statistics counters. The PHY encodes the MAC frame as required for reliable transmission over the media to the remote end.

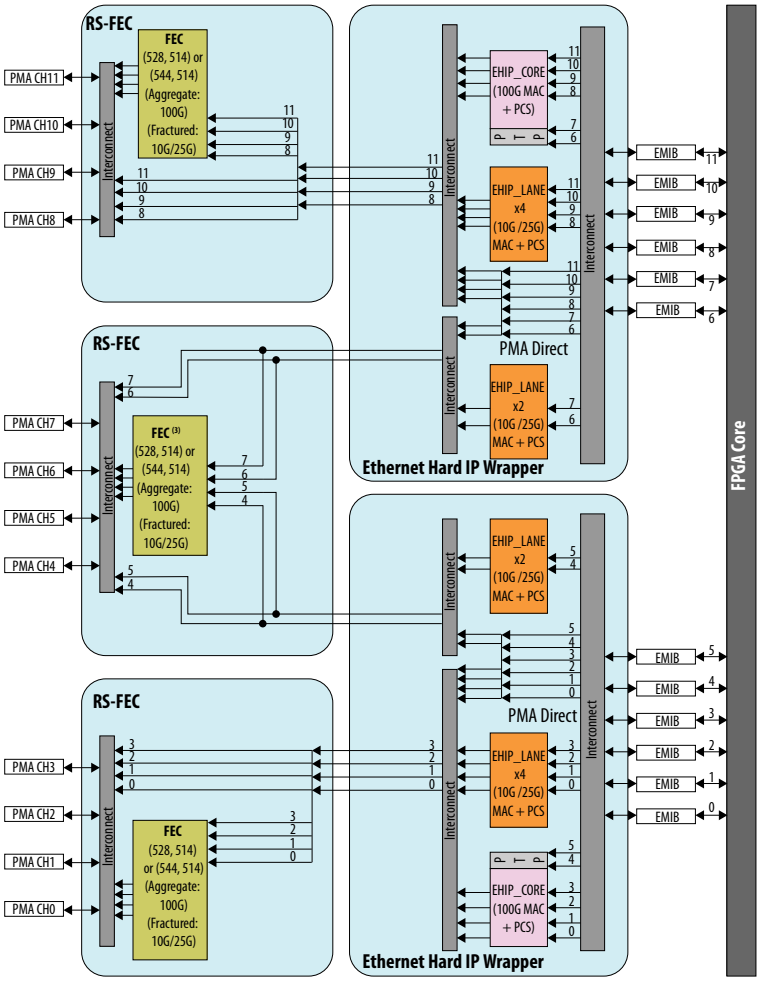
In the receive direction, the PHY passes frames to the MAC. The MAC accepts frames from the PHY, performs checks, updates statistics counters, strips out the CRC, preamble, and SFD, and passes the rest of the frame to the client. In RX preamble pass-through mode, the MAC passes on the preamble and SFD to the client instead of stripping them out. You can configure the MAC to provide the full RX frame at the client interface, the frame with CRC bytes removed, or the frame with CRC and RX PAD bytes removed.

The E-Tile Hard IP for Ethernet Intel FPGA IP also supports PCS Only, FlexE, and OTN variations. The PCS Only variations provide an MII interface to the client and transmit and receive Ethernet packets through a 10-Gbps, 25-Gbps, and 100-Gbps Ethernet PHY implemented in hard IP. The FlexE and OTN variations use PCS66 interface for

transmitting and receiving 66b blocks, bypassing the MAC. The PCS Only, OTN , and FlexE variations support optional KR-FEC(528,514) or KP-FEC(544,514) for 25G and 100G Ethernet rate.

Figure 12. E-tile Architecture and Datapath Overview

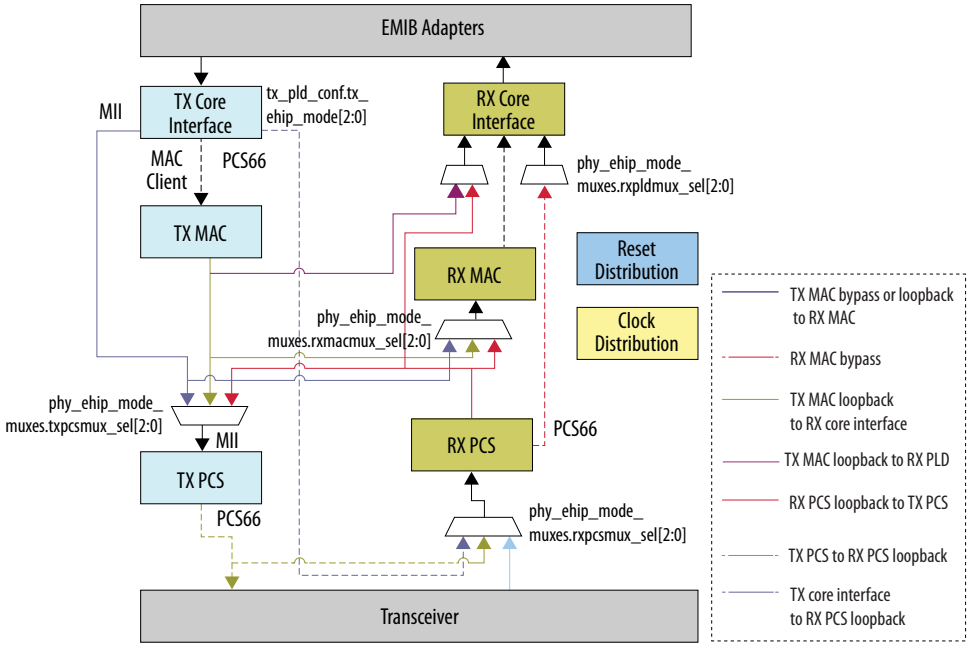
Showing 12 out of 24 channels per tile.



Notes:

1. Not all datapath combinations are available.
2. Datapath enablement depends on the configuration you are implementing. Refer to the E-Tile Channel Placement tool for possible configurations.
3. This FEC block can only be used in aggregate mode with FEC direct application (e.g. 128GFC Fibre-Channel). This FEC block cannot be used in in aggregate mode with EHIP_CORE because there is no EHIP_CORE in this location. Refer to Intel Stratix 10 E-Tile Transceiver User Guide for more information on FEC direct application.

Figure 13. E-Tile Hard IP for Ethernet Intel FPGA IP Instance and Bypass Modes



Note: The E-Tile Hard IP for Ethernet Intel FPGA IP provides support for the OTN feature. For further inquiries, contact your nearest Intel sales representative.

2.9.1. E-Tile Hard IP for Ethernet Intel FPGA IP MAC

2.9.1.1. MAC TX Datapath

When the TX MAC module in a channel is enabled, it receives the client payload data with the destination and source addresses and then adds, appends, or updates various header fields in accordance with the configuration specified. The MAC does not modify the destination address or the payload received from the client. However, the TX MAC module adds a preamble (if the IP core is not configured to receive the preamble from user logic), pads the payload of frames greater than eight bytes to satisfy the minimum Ethernet frame payload of 46 bytes, and if you enable source address insertion, replaces the bytes in the source address field position of your data with a stored source address you provide as a parameter.

Note: The TX MAC interface does not support non-contiguous transfer. The `i_sl_tx_valid/i_tx_valid` must be continuously asserted between the assertions of the start of packet and end of packet signals for the same packet. You must implement store and forward packet mechanism when transferring non-contiguous packets.

The client interface includes a port named `i_skip_crc`, which when asserted during a frame, makes the MAC skip the insertion of source address, padding, and CRC.

- When CRC insertion is skipped, the client must provide a CRC for the frame data it writes in the last 4 bytes of the frame.
- When padding is skipped, the frame data must be large enough to include a fully formed frame header (at least 14 bytes long) or the MAC will automatically mark it as an error frame.

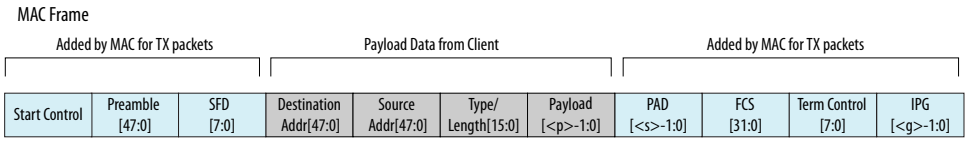
The TX MAC module always inserts IDLE bytes to maintain an average IPG.

The E-Tile Hard IP for Ethernet Intel FPGA IP drops incoming frames of less than nine bytes.

Figure 14. Typical Client Frame at the Transmit Interface

The figure illustrates the changes that the TX MAC makes to the client frame when **Enable preamble passthrough** is turned off. This figure uses the following notational conventions:

- <*p*> = payload size, which is arbitrarily large.
- <*s*> = number of padding bits (0–46 bytes)
- <*g*> = number of IPG bits (full bytes)



The following sections describe the functions performed by the TX MAC:

TX Preamble, Start, and SFD Insertion on page 71
Source Address Insertion on page 72
Length/Type Field Processing on page 72
Frame Padding on page 72
Frame Check Sequence (CRC-32) Insertion on page 72
Inter-Packet Gap Generation and Insertion on page 72

2.9.1.1.1. TX Preamble, Start, and SFD Insertion

In the TX datapath the MAC appends an eight-byte preamble that begins with a Start byte (0xFB) to the client frame.

The source of the preamble depends on whether you enable the preamble pass-through feature by turning on **Enable preamble passthrough** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor.

If the preamble pass-through feature is turned on, the client must provide 8 preamble bytes (including an SFD byte) on the data bus. The MAC will automatically replace the `Start Control` byte.

2.9.1.1.2. Source Address Insertion

If you configure the IP core to use source address insertion, the MAC replaces the bytes in the `Source Addr` field provided by the client interface with the source address given by the `txmac_saddr` parameter.

To enable source address insertion, turn on **Use Source Address Insertion** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor.

2.9.1.1.3. Length/Type Field Processing

This two-byte header field represents either the length of the payload or the type of MAC frame. When the value of this field is equal to or greater than 1536 (0x600) it indicates a type field. Otherwise, this field provides the length of the payload data that ranges from 0–1500 bytes. The TX MAC does not modify this field before forwarding it to the network; it uses this field to generate TX Statistics.

2.9.1.1.4. Frame Padding

When the length of client frame is less than 64 bytes and greater than eight bytes, the TX MAC module inserts pad bytes after the payload to create a frame length equal to the minimum size of 64 bytes. If the `i_skip_crc` signal is asserted while writing frame data, the core does not insert PAD bytes even if the frame is shorter than 64 bytes long.

Caution: The E-Tile Hard IP for Ethernet Intel FPGA IP drops client frames of less than nine bytes because it cannot transfer the frames to the E-tile. You must ensure such frames do not reach the TX client interface.

2.9.1.1.5. Frame Check Sequence (CRC-32) Insertion

As long as the `i_skip_crc` signal on the TX client interface is not asserted,the TX MAC computes and inserts a frame check sequence (FCS) in the transmitted MAC frame. The FCS field contains a 32-bit Cyclic Redundancy Check (CRC32) value. The MAC computes the CRC32 over the frame bytes that include the source address, destination address, length/type field, data, and pad (if applicable). The FCS computation excludes the preamble and SFD. The encoding is defined by the following generating polynomial:

```
FCS(X) = X32 +X26 +X23 +X22 +X16 +X12 +X11 +X10 +X8 +X7 +X5 +X4 +X2 +X1 +1
```

CRC bits are transmitted with MSB (X32) first.

If `i_skip_crc` is asserted while writing frame data, the TX MAC will not append an FCS to the end of the frame. This will cause the resulting packet to be invalid unless the last 4 bytes of frame data are a correctly computed FCS value.

Related Information

Order of Ethernet Transmission on page 80

2.9.1.1.6. Inter-Packet Gap Generation and Insertion

If you set **Average Inter-packet Gap** to **12** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor, the TX MAC maintains the minimum inter-packet gap (IPG) between transmitted frames required by the IEEE 802.3 Ethernet standard. The

standard requires an average minimum IPG of 96 bit times (or 12 byte times). The MAC uses a deficit idle counter to allow the actual gap between frames to vary as needed to meet the maximum throughput requirements of the link.

If you set **Average Inter-packet Gap** to **10** or **8**, the TX MAC maintains a minimum average IPG of 10 or 8 bytes accordingly. This option is provided as an intermediate option to allow you to enforce an IPG that does not conform to the Ethernet standard, but which increases the throughput of your IP core.

If you set **Average Inter-packet Gap** to **1**, the IP core transmits Ethernet packets as soon as the data is available, with the minimum possible gap. The IPG depends on the space you leave between frame data as you write it to the core. If you select this parameter value, the core will no longer comply with the Ethernet standard, but your application will have control over the average gap and throughput can be maximized. For a packet of size (P) bytes, the number of idles bytes (G) inserted after is specified by the following formula G=8-(P%8). A few examples are depicted below:

Packet Size (P)	Gap Idle Bytes (G)
64	8
65	7
66	6
67	5
68	4
69	3
70	2
71	1
72	8

Note: Even when you set the Average Inter-packet Gap to 1, the 10G/25G channels will still enforce an effective IPG of 5. This is because the protocol specifically prohibits IPG lower than 5 for 10G/25G links to prevent MACs from producing packets that cannot be encoded using 64B/66B encoders.

Related Information

Parameter Editor Parameters on page 53

2.9.1.2. MAC RX Datapath

When the RX MAC in the channel is enable, it receives Ethernet frames from the PHY and forwards it to the client with framing information together with the results of header and error checking functions.

You can configure whether to include or remove the PAD bytes and FCS using the **Bytes to remove from RX frames** parameter.

Figure 15. Flow of Frame Through the MAC RX Without Preamble Pass-Through

The figure illustrates the typical flow of frame through the MAC RX when the preamble pass-through feature is turned off. In this figure, <*p*> is payload size, and <*s*> is the number of pad bytes (0–46 bytes).

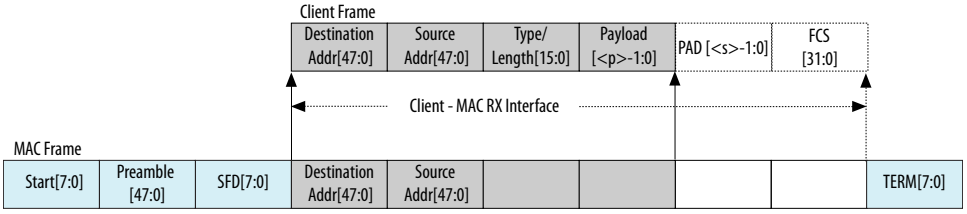
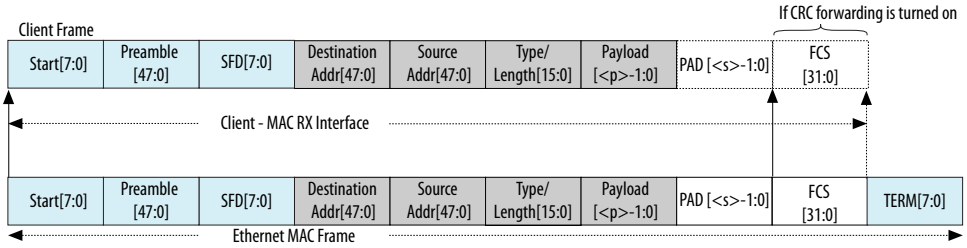


Figure 16. Flow of Frame Through the MAC RX With Preamble Pass-Through

The figure illustrates the typical flow of frame through the MAC RX when the preamble pass-through feature is turned on. In this figure, <*p*> is payload size, and <*s*> is the number of pad bytes (0–46 bytes)..



The following sections describe the functions performed by the RX MAC:

RX Preamble Processing on page 74
RX Strict SFD Checking on page 74
RX FCS Checking on page 75
RX Malformed Packet Handling on page 75
Removing PAD Bytes and FCS Bytes from RX Frames on page 75
RX Undersized Frames, Oversized Frames, and Frames with Length Errors on page 75
Inter-Packet Gap on page 76

2.9.1.2.1. RX Preamble Processing

The preamble sequence is Start, six preamble bytes, and SFD. The Start byte must be on receive lane 0 of the MII, which means byte [7:0] of the data decoded from a 66b block. The IP core uses the Start Control byte (0xFB, with the corresponding MII control bit set to 1) to identify the start of the Ethernet packet, and the location of the preamble.

By default, the MAC RX removes all Start, SFD, preamble, and IPG bytes from accepted frames. However, if you turn on **Enable preamble passthrough** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor, the MAC RX does not remove the eight-byte preamble sequence.

2.9.1.2.2. RX Strict SFD Checking

The E-Tile Hard IP for Ethernet Intel FPGA IP RX MAC checks all incoming packets for a correct Start byte (0xFB).

If you turn on **Enable strict preamble check** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor, the RX MAC requires all RX packets to have an Ethernet standard preamble (0x55_55_55_55_55_55). If you turn on **Enable strict SFD check**, the RX MAC requires all RX packets to have an Ethernet standard Start Frame Delimiter (0xD5).

Strict checking reduces the incidence of runt packets caused by bit errors on the line. However, do not use strict checking in applications where custom preamble values or SFD values are needed.

Table 20. Strict SFD Checking Configuration

Enable Strict SFD Check	0x50A[4]: Preamble Check	0x50A[3]: SFD Check	Fields Checked	Behavior if Check Fails
Off	Don't Care	Don't Care	Start byte	IP core does not recognize a malformed Start byte as a Start byte
On	0	0	Start byte	
	0	1	Start byte and SFD	IP Core drops the packet
	1	0	Start byte and preamble	
	1	1	Start byte and preamble and SFD	

2.9.1.2.3. RX FCS Checking

The RX MAC checks the FCS of all incoming packets that are minimum sized or larger. If the RX MAC detects an FCS error, it marks the frame invalid by asserting `o_rx_error[1]`. FCS errors are also indicated for arriving packets containing an Error control block.

2.9.1.2.4. RX Malformed Packet Handling

While receiving an incoming packet from the Ethernet link, the RX MAC expects packets to end with a Terminate Control byte. Packets that contain Error bytes or a control byte other than Terminate are malformed packets. The RX MAC asserts `o_rx_error[0]` when the frame ends to indicate that it was a malformed packet.

2.9.1.2.5. Removing PAD Bytes and FCS Bytes from RX Frames

The **Bytes to remove from RX frames** parameter in the parameter editor offers the option of removing bytes from the end of RX frames. You can program the RX MAC to present all the bytes that arrive at the end of an RX frame, remove the RX FCS bytes only, or remove the RX FCS bytes and any RX PAD bytes that were added to the frame.

2.9.1.2.6. RX Undersized Frames, Oversized Frames, and Frames with Length Errors

The RX MAC flags RX frames that arrive with fewer than 64 bytes as undersized, and are not checked for FCS. The RX MAC marks undersized frames by asserting `o_rx_error[2]` when the frame ends.

The RX MAC marks RX frames that arrive with more bytes than the **RX Maximum Frame Size** value you specify in the parameter editor as oversized. The RX MAC marks oversized frames by asserting `o_rx_error[3]` when the frame ends.

If you turn on **Enforce Maximum Frame Size** in the parameter editor, oversized frames are not allowed through the RX client interface. When the frame reaches the maximum size, it is ended, and the RX MAC asserts both`o_rx_error[3]` and `o_rx_error[1]` to indicate the frame was truncated.

RX Frames that arrive with a valid Length field (Length/Type ≤ 1500) are checked for length errors. If the length of the packet advertised in the Length/Type field is larger than the length of the frame that actually arrived, the RX MAC asserts `o_rx_error[4]` to indicate that there was a length error.

2.9.1.2.7. Inter-Packet Gap

The MAC RX removes all IPG octets received, and does not forward them to the client interface. It can tolerate a sustained stream of packets with an IPG of 1.

2.9.1.3. Congestion and Flow Control Using PAUSE or Priority Flow Control (PFC)

If you do not select **Disable Flow Control** in the **Stop TX traffic when link partner sends pause** parameter, the E-Tile Hard IP for Ethernet Intel FPGA IP provides flow control to reduce congestion at the local or remote link partner. When either link partner experiences congestion, the respective TX MAC can be instructed to send PAUSE or PFC frames to regulate the flow of data from the other side of the link.

- PAUSE frames instruct the remote transmitter to stop sending data for the duration that the congested receiver specified in an incoming XOFF frame.
- PFC frames instruct the receiver to halt the flow of packets assigned to a specific Priority Queue for a specified duration.

2.9.1.3.1. Conditions Triggering XOFF Frame Transmission

The E-Tile Hard IP for Ethernet Intel FPGA IP supports retransmission. In retransmission, the IP core retransmits a XOFF frame periodically, extending the pause time, based on signal values.

The TX MAC transmits PAUSE XOFF frames when one of the following conditions occurs:

- Client requests XOFF transmission—A client can explicitly request that XOFF frames be sent using the `i_tx_pause` and `i_tx_pfc[7:0]` signals. When `i_tx_pause` is asserted, a PAUSE XOFF frame is sent to the Ethernet network when the current frame transmission completes. When `i_tx_pfc` is asserted, a PFC XOFF packet is transmitted with XOFF requests for each of the Queues that has a bit high in the signal. For example, setting `i_tx_pfc` to 0x03 sends XOFF requests for Queues 0 and 1.
- Host (software) requests PAUSE XOFF transmission—Setting the pause request register triggers a request that a PAUSE XOFF frame be sent. Similarly, setting the PFC request register triggers PFC XOFF frame requests for the selected Priority Queues.
- Retransmission mode—If the retransmit hold-off enable bit has the value of 1, and the `i_tx_pause` signal remains asserted or the pause request register value remains high, when the time duration specified in the hold-off quanta register has lapsed after the previous PAUSE XOFF transmission, the TX MAC sends another PAUSE XOFF frame to the Ethernet network. The same mechanism applies to PFC. While the IP core is paused in retransmission mode, you cannot use either of the other two methods to trigger a new XOFF frame: the signal or register value is already high.

Note: Intel recommends that you use the flow control ports to backpressure the remote Ethernet node.

2.9.1.3.2. Conditions Triggering XON Frame Transmission

The TX MAC transmits PAUSE or PFC XON frames when one of the following conditions occurs:

- Client requests XON transmission—A client can explicitly request that XON frames be sent using the pause control interface signal. When `i_tx_pause` is deasserted, a PAUSE XON frame is sent to the Ethernet network when the current frame transmission completes. Similarly, when `i_tx_pfc[n]` is deasserted, a PFC frame is sent with a PFC XON message for queue *n*. If multiple PFC queues are deasserted, the TX MAC will pack the requests into the same PFC packet if possible.
- Host (software) requests XON transmission—Resetting the pause request register triggers a request that an XON frame be sent.

2.9.1.4. Pause Control and Generation Interface

The flow control interface implements PAUSE as specified by the *IEEE 802.3ba 2010 High Speed Ethernet Standard*, PFC as specified by the *IEEE Standard 802.1Qbb*.

You can configure the PAUSE logic to automatically stop local packet transmission when the link partner sends a PAUSE XOFF packet. The PAUSE logic can pass the PAUSE packets through as normal packets or drop the packets before they reach the RX client.

As for PFC frames, you can configure the PFC logic to pass the PFC packets through as normal packets or drop them before they reach the RX client. However, you don't have an option to stop traffic automatically when a PFC XOFF frame arrives.

Table 21. Pause Control and Generation Signals

Describes the signals that implement pause control. These signals are available only if you turn on flow control in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor.

Note: The signal names may have slight variance depending on the variant you select.

Signal Name	Direction	Description
i_tx_pause (PAUSE) i_tx_pfc (PFC)	Input	Level signal which directs the IP core to insert a PAUSE or PFC frame for priority traffic class [n] on the Ethernet link. If bit [n] of the TX_PAUSE_EN register has the value of 1, the IP core transmits an XOFF frame when this signal is first asserted. If you enable retransmission, the IP core continues to transmit XOFF frames periodically until the signal is de-asserted. When the signal is deasserted, the IP core inserts an XON frame.
o_rx_pause (PAUSE) o_rx_pfc (PFC)	Output	Asserted to indicate an RX a PAUSE or PFC signal match. The IP core asserts bit [n] of this signal when it receives a pause request with an address match, to signal the TX MAC to throttle its transmissions from priority queue [n] on the Ethernet link.

2.9.1.5. Pause Control Frame Filtering

The E-Tile Hard IP for Ethernet Intel FPGA IP supports options to enable or disable the following features for incoming pause control frames. These options are available as long as you do not set the **Stop TX traffic when link partner sends pause** parameter to **Disable Flow Control**.

For filtering, the PAUSE and PFC packets are only processed if their destination address matches the address given by the `rx_pause_daddr` parameter.

- If you turn on **Forward RX Pause Requests** in the parameter editor, the RX PAUSE and PFC frames are always passed along the RX client, even if they are processed.
- If you turn off **Forward RX Pause Requests** in the parameter editor, the RX PAUSE and PFC packets are processed internally, and not presented to the RX client as valid packets.

A PAUSE or PFC packet must have a destination address that matches the `rx_pause_daddr` parameter, a Length/Type field that is set to 0x8808, and the first 2 bytes of the packet set to 0x0001 or 0x0101.

To actually trigger PAUSE or PFC, you must also ensure that the packets are of the correct length and have no FCS error. Because these conditions are not known until the whole packet has arrived, if you turn off **Forward RX Pause Requests**, you may have packets that are filtered because they look like PAUSE or PFC packets, but not processed because they are of the wrong size or have an error.

2.9.1.6. Link Fault Signaling

If you enable **Choose Link Fault Generation Mode** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor, the IP core provides link fault signaling as defined in the *IEEE 802.3ba-2010 High Speed Ethernet Standard* and Clause 66 of the *IEEE 802.3-2012 Ethernet Standard*, based on the `LINK_FAULT_CONFIG` register settings.

The Ethernet MAC includes a Reconciliation Sublayer (RS) located between the MAC and the MII to manage local and remote faults. Link fault signaling on the Ethernet link is disabled by default but can be enabled by bit [0] of the `link_fault_config` register. When the `link_fault_config` register bits [1:0] have the value of 2'b01,

link fault signaling is enabled in normal bidirectional mode. In this mode, the local RS TX logic transmits remote fault sequences in case of a local fault and transmits IDLE control words in case of a remote fault.

If you turn on bit [1] of the `link_fault_config` register, the IP core conforms to Clause 66 of the *IEEE 802.3-2012 Ethernet Standard*. When `link_fault_config[1:0]` has the value of 2'b11, the IP core transmits the fault sequence ordered sets in the interpacket gaps according to the clause requirements.

The RS RX logic sets `remote_fault_status` or `local_fault_status` to 1 when the RS RX block receives remote fault or local fault sequence ordered sets. When valid data is received in more than 127 columns, the RS RX logic resets the relevant fault status (`remote_fault_status` or `local_fault_status`) to 0.

The IEEE standard specifies RS monitoring of RXC<7:0> and RXD<63:0> for Sequence `ordered_sets`. For more information, refer to *Figure 81–9—Link Fault Signaling state diagram* and *Table 81-5—Sequence* `ordered_sets` in the *IEEE 802.3ba 2010 High Speed Ethernet Standard*. The variable `link_fault` is set to indicate the value of an RX Sequence `ordered_set` when four `fault_sequences` containing the same fault value are received with fault sequences separated by less than 128 columns and with no intervening `fault_sequences` of different fault values. The variable `link_fault` is set to OK following any interval of 128 columns not containing a remote fault or local fault Sequence `ordered_set`.

Related Information

- IEEE Website
  The IEEE 802.3ba –2010 High Speed Ethernet Standard and the IEEE 802.3 – 2012 Ethernet Standard are available on the IEEE website.
- Link Fault Configuration on page 223

2.9.1.6.1. Determining Link Fault Condition

In Intel Quartus Prime, the E-Tile Hard IP for Ethernet Intel FPGA IP provides the `o_sl_rx_pcs_fully aligned/o_rx_pcs_fully aligned` signal to determine link fault condition. Implement the following pseudo-code on the RX MII port:

```
If (o_sl_rx_pcs_fully aligned/o_rx_pcs_fully_aligned  == 0)  (

 •local fault pattern received on o_sl_rx_mii_d/o_rx_mii_d

 •remote fault is expected on the TX serial data
)
else if (o_sl_rx_pcs_fully aligned/o_rx_pcs_fully_aligned   == 1 &&
o_sl_rx_mii_valid/o_rx_mii_valid==1)

  •o_sl_rx_mii_d/o_rx_mii_d is a valid XGMII block

else if (o_sl_rx_pcs_fully aligned/o_rx_pcs_fully_aligned   ==1 &&
o_sl_rx_mii_valid/o_rx_mii_valid==0)

 •Ignore o_sl_rx_mii_d/o_rx_mii_d as it is not valid XGMII data

endif
```

Related Information

How do I use the "o_rx_pcs_fully_aligned" signal to tell the difference between a local fault condition and valid RX data when using the Intel® Stratix® 10 E-tile Hard IP for Ethernet Intel® FPGA IP configured in PCS+FEC status without the MAC?

2.9.1.7. Order of Ethernet Transmission

The TX MAC transmits bytes on the Ethernet link starting with the preamble and ending with the FCS in accordance with the IEEE 802.3 standard. On the transmit client interface, the IP core expects the client to send the most significant bytes of the frame first, and to send each byte in big-endian format. Similarly, on the receive client interface, the IP core sends the client the most significant bytes of the frame first, and orders each byte in big-endian format.

Figure 17. Byte Order on the Client Interface Lanes Without Preamble Pass-Through

The figure describes the byte order on the Avalon streaming interface when the preamble pass-through feature is turned off. Destination Address[40] is the broadcast/multicast bit (a type bit), and Destination Address[41] is a locally administered address bit.

	Destination Address (DA)						Source Address (SA)						Type/ Length (TL)		Data (D)		
Octet	5	4	3	2	1	0	5	4	3	2	1	0	1	0	00	...	NN
Bit	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]	[15:8]	[7:0]	MSB[7 :0]	...	LSB[7:0]

For example, the destination MAC address includes the following six octets AC-DE-48-00-00-80. The first octet transmitted (octet 0 of the MAC address described in the 802.3 standard) is AC and the last octet transmitted (octet 7 of the MAC address) is 80. The first bit transmitted is the low-order bit of AC, a zero. The last bit transmitted is the high order bit of 80, a one.

The preceding table and the following figure show that in this example, 0xAC is driven on DA5 (`DA[47:40]`) and 0x80 is driven on DA0 (`DA[7:0]`).

Figure 18. Octet Transmission on the Avalon Streaming Interface Signals Without Preamble Pass-Through

The figure illustrates how the octets of the client frame are transferred over the TX datapath when preamble pass-through is turned off.

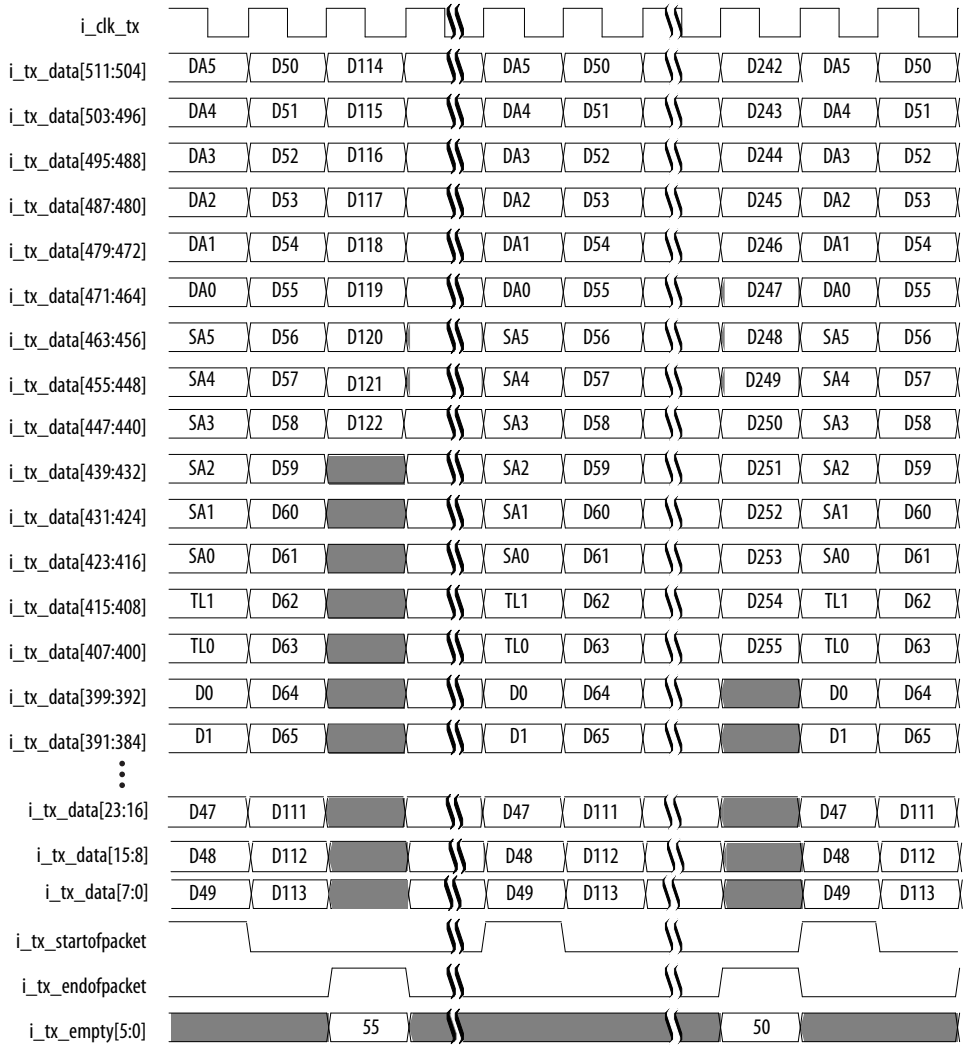


Figure 19. Byte Order on the Avalon Streaming Interface Lanes With Preamble Pass-Through

The figure describes the byte order on the Avalon streaming interface when the preamble pass-through feature is turned on.

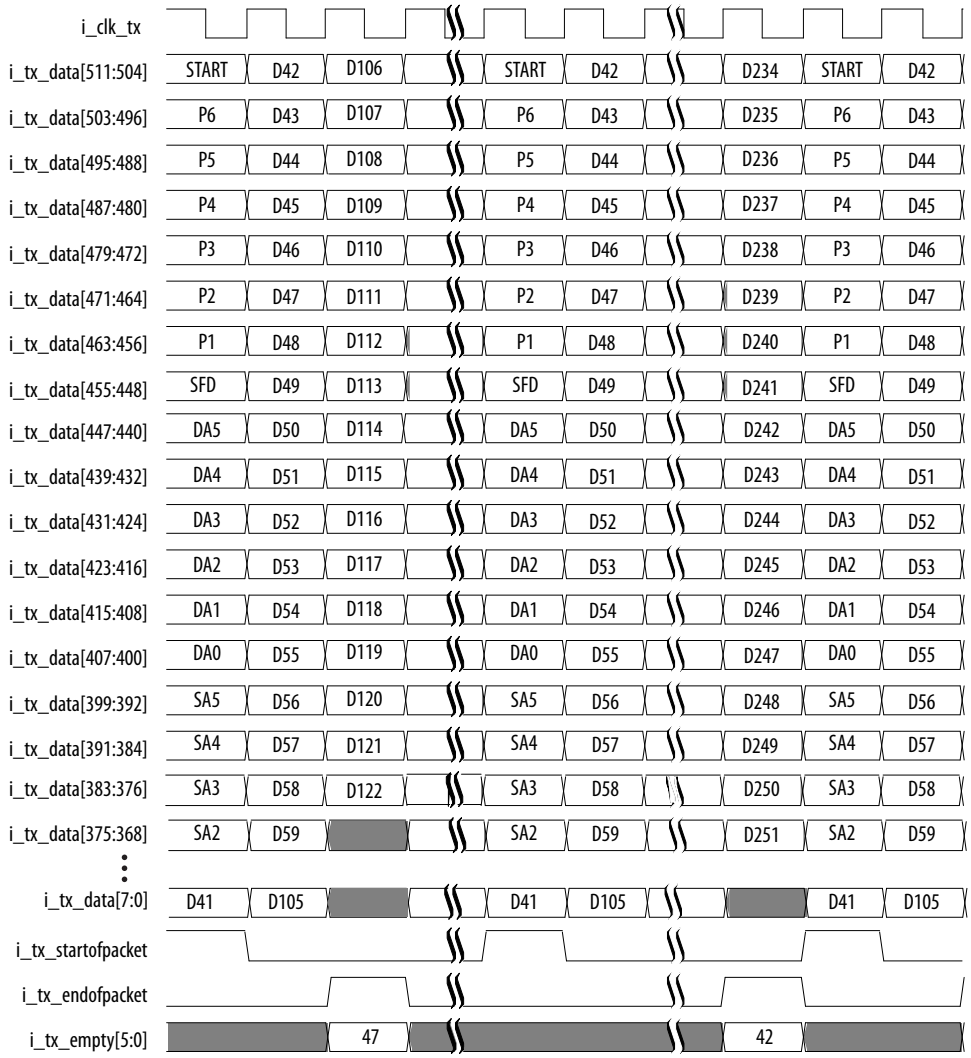
Destination Address[40] is the broadcast/multicast bit (a type bit), and Destination Address[41] is a locally administered address bit.

	SFD	Preamble						Start	Destination Address (DA)						Source Address (SA)						Type/ Length		Data (D)		
Octet	7	6	5	4	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0	1	0	00	...	NN
Bit	[63:56]	[55:48]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]	[15:8]	[7:0]	MSB[7:0]	...	LSB[7:0]

Figure 20. Octet Transmission on the Avalon Streaming Interface Signals With Preamble Pass-Through

The figure illustrates how the octets of the client frame are transferred over the TX datapath when preamble pass-through is turned on. The eight preamble bytes precede the destination address bytes. The preamble bytes are reversed: the application must drive the SFD byte on `i_tx_data[455:448]` and the START byte on `i_tx_data[511:504]`.

The destination address and source address bytes follow the preamble pass-through in the same order as in the case without preamble pass-through.



2.9.2. 1588 Precision Time Protocol Interfaces

If you turn on **Enable IEEE 1588 PTP**, the E-Tile Hard IP for Ethernet Intel FPGA IP processes and provides 1588 Precision Time Protocol (PTP) timestamp information as defined in the *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard*. This feature supports PHY operating speed with a constant timestamp accuracy as specified in the PTP Timestamp Accuracy per Ethernet Data Rate table.

1588 PTP packets carry timestamp information. The E-Tile Hard IP for Ethernet Intel FPGA IP updates the incoming timestamp information in a 1588 PTP packet to transmit a correct updated timestamp with the data it transmits on the Ethernet link, using a one-step or two-step clock.

A fingerprint can accompany a 1588 PTP packet. You can use this information for client identification and other client uses. If provided fingerprint information, the IP core passes it through unchanged.

The IP core connects to a time-of-day (TOD) module that continuously provides the current time of day based on the input clock frequency. Because the module is outside the E-Tile Hard IP for Ethernet Intel FPGA IP, you can use the same module to provide the current time of day for multiple modules in your system.

Related Information

IEEE website
The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

2.9.2.1. Implementing a 1588 System That Includes a E-Tile Hard IP for Ethernet Intel FPGA IP

The 1588 specification in *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* describes various systems you can implement in hardware and software to synchronize clocks in a distributed system by communicating time offset and frequency correction information between master and slave clocks in arbitrarily complex systems. A 1588 system that includes the E-Tile Hard IP for Ethernet Intel FPGA IP with 1588 PTP functionality uses the incoming and outgoing timestamp information from the IP core and the other modules in the system to synchronize clocks across the system.

The E-Tile Hard IP for Ethernet Intel FPGA IP with 1588 PTP functionality provides the timestamp manipulation and basic update capabilities required to integrate your IP core in a 1588 system. You can specify that packets are PTP packets, and how the IP core should update incoming timestamps from the client interface before transmitting them on the Ethernet link. The IP core does not implement the event messaging layers of the protocol, but rather provides the basic hardware capabilities that support a system in implementing the full 1588 protocol.

Related Information

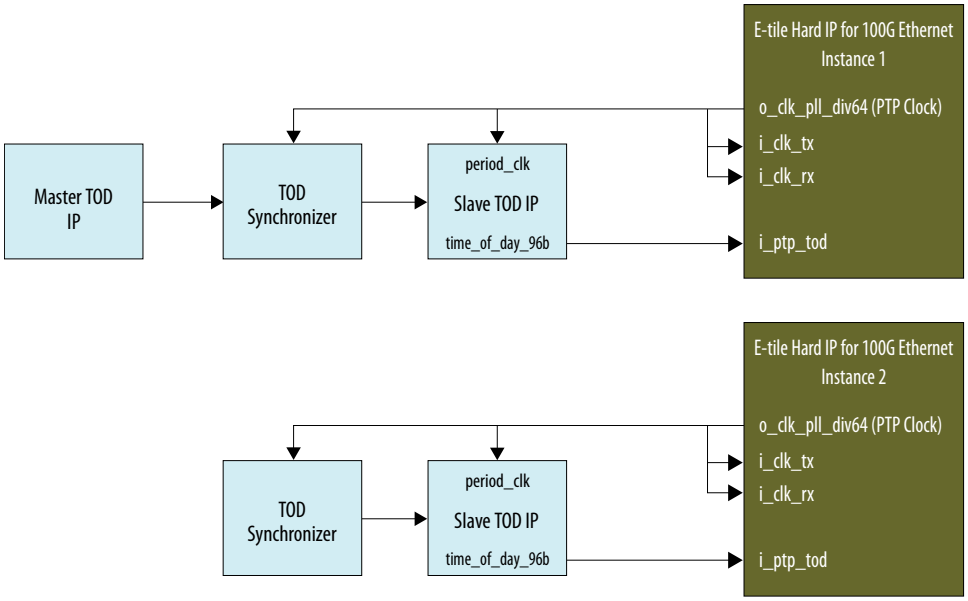
IEEE website
The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

2.9.2.2. PTP Timestamp Accuracy

The E-Tile Hard IP for Ethernet Intel FPGA IP 10G/25G variants support two PTP accuracy modes, **Basic Mode** and **Advanced Mode**, selectable from the **PTP Accuracy Mode** drop down menu under PTP options.

The E-Tile Hard IP for Ethernet Intel FPGA IP 100G variants support timestamping accuracy of +/-8ns.

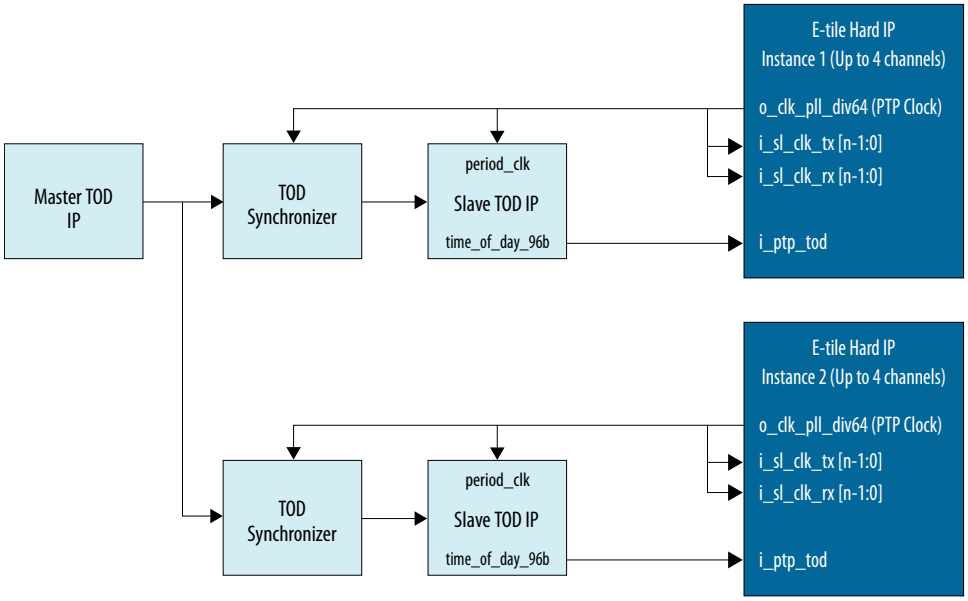
Figure 21. 100G PTP Accuracy Block Diagram



PTP Timestamp Accuracy in Basic Mode

When you select **Basic Mode** in the **PTP Accuracy Mode** under PTP Options in the parameter editor, IP core requires a single TOD IP and a single TOD synchronizer IP. The TOD blocks are shared between 1 to 4 10G/25G Ethernet channels in the same IP core. Note that sharing TOD IP with different E-Tile Hard IP for Ethernet IP instances is not supported.

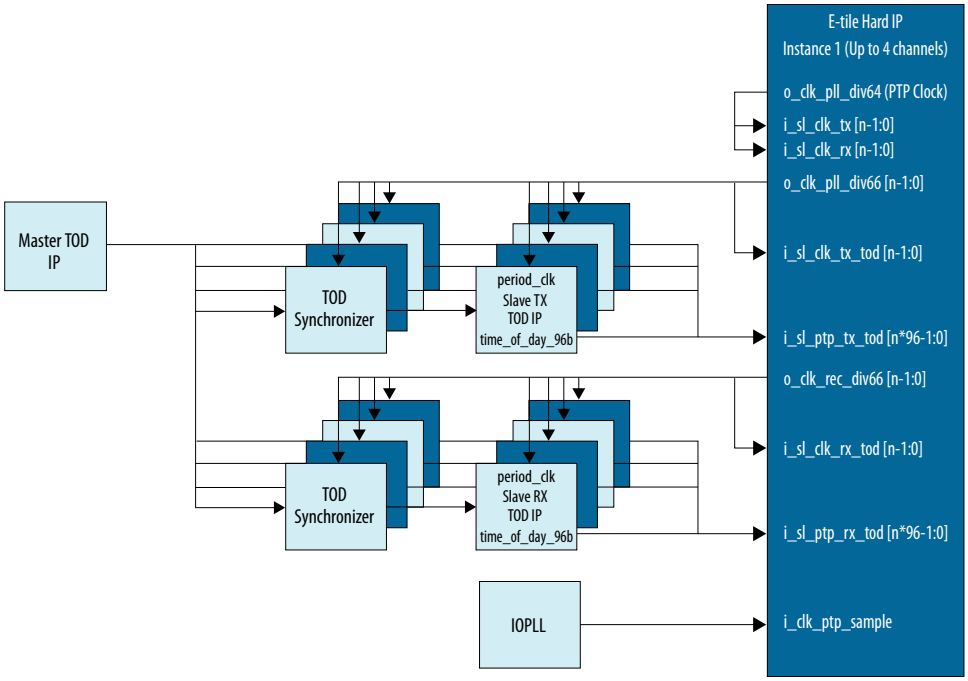
Figure 22. Basic Mode PTP Accuracy Block Diagram



PTP Timestamp Accuracy in Advanced Mode

When you select **Advanced Mode** in the **PTP Accuracy Mode** under PTP Options in the parameter editor, each channel in a 10G/25G Ethernet IP instance has two TOD interface ports, `i_sl_ptp_tx_tod[95:0]` (for TX port) and `i_sl_ptp_rx_tod[95:0]` (for RX port). Each channel in the same 10G/25G E-Tile Hard IP for Ethernet IP core requires two TOD IPs and two TOD Synchronizer IPs.

Figure 23. Advanced Mode PTP Accuracy Block Diagram



The functional usage of basic and advanced modes is the same. The advanced mode has additional clock inputs and TOD module requirements. Refer to the *10G/25G Ethernet Channel with Basic PTP Accuracy Mode* for the clock connections and the *External Time-Of-Day Module Variations with 1588 PTP Feature* section for the TOD module connections in basic and advanced modes.

To improve accuracy, the advanced mode utilizes additional logic and TOD modules that affects the resource utilization. The advanced mode also requires an IOPLL and logic lock region requirements. For more information, refer to the *Logic Lock Regions Requirements for PTP Accuracy Mode*.

Table 22. PTP Timestamp Accuracy per Ethernet Data Rate

Ethernet Data Rate	PTP Timestamp Accuracy		Parallel Clock Frequency
	Basic Mode	Advanced Mode	
10GE	±3 ns	± 1.5 ns	402.83 MHz
25GE	±3 ns	± 1.5 ns	402.83 MHz
100GE	±8 ns	N/A	402.83 MHz

Note: The 100G PAM4 configuration is not supported.

Related Information

- External Time-of-Day Module for Variations with 1588 PTP Feature on page 50
- 10G/25G Ethernet Channel with Basic PTP Accuracy Mode on page 172
- Logic Lock Regions Requirements for PTP Accuracy Advanced Mode on page 104

2.9.2.3. PTP Transmit Functionality

When you send a 1588 PTP packet to a E-Tile Hard IP for Ethernet Intel FPGA IP with **Enable IEEE 1588 PTP** turned on in the parameter editor, you must assert one and only one of the following input signals with the TX SOP signal to tell the IP core the incoming packet is a 1588 PTP packet:

- `i_sl_ptp_ts_req/i_ptp_ts_req`: assert this signal to tell the IP core to process the current packet in two-step processing mode.
- `i_sl_ptp_ins_ets/i_ptp_ins_ets`: assert this signal to tell the IP core to process the current packet in one-step processing mode and to insert the exit timestamp for the packet in the packet (insertion mode).
- `i_sl_ptp_ins_cf/i_ptp_ins_cf`: assert this signal to tell the IP core to process the current packet in one-step processing mode and to update the timestamp in the packet by adding the latency through the IP core (the residence time in the IP core) to the cumulative delay field maintained in the packet (correction mode). This mode supports transparent clock systems.

Note: For 100G variant, these signal names are sightly different. Refer to the 1588 PTP Interface on page 146 for signal name details.

All TX PTP operations assume the `o_sl_tx_ptp_ready` signal was asserted and is held high.

Figure 24. Example Waveform for 2-step TX Timestamp using `i_sl_ptp_ts_req` Signal

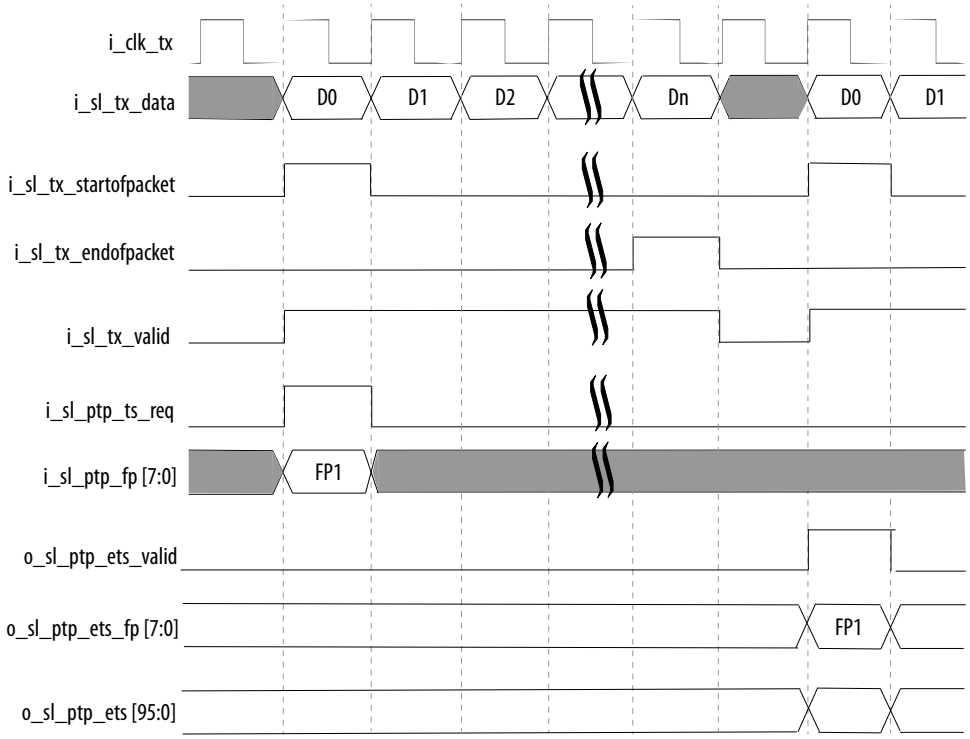


Figure 25. Example Waveform for 1-step TX Timestamp using `i_sl_ptp_ins_ets` Signal

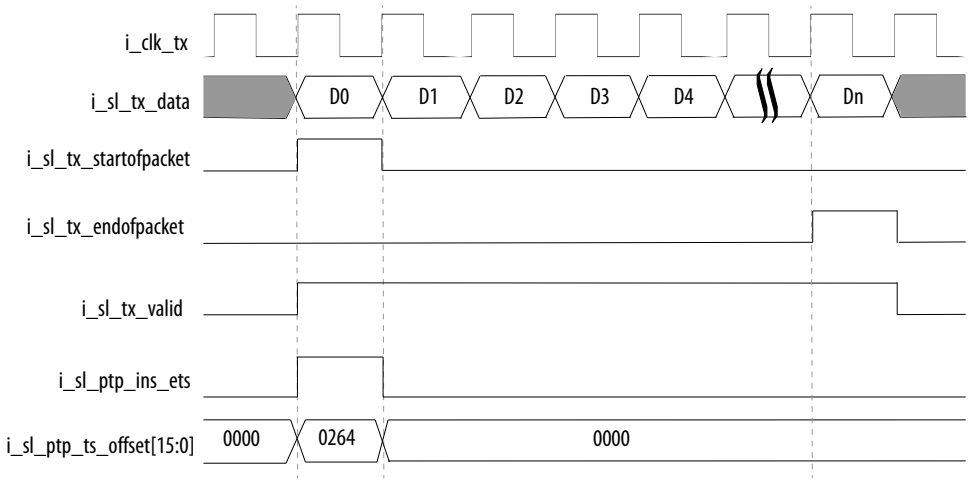
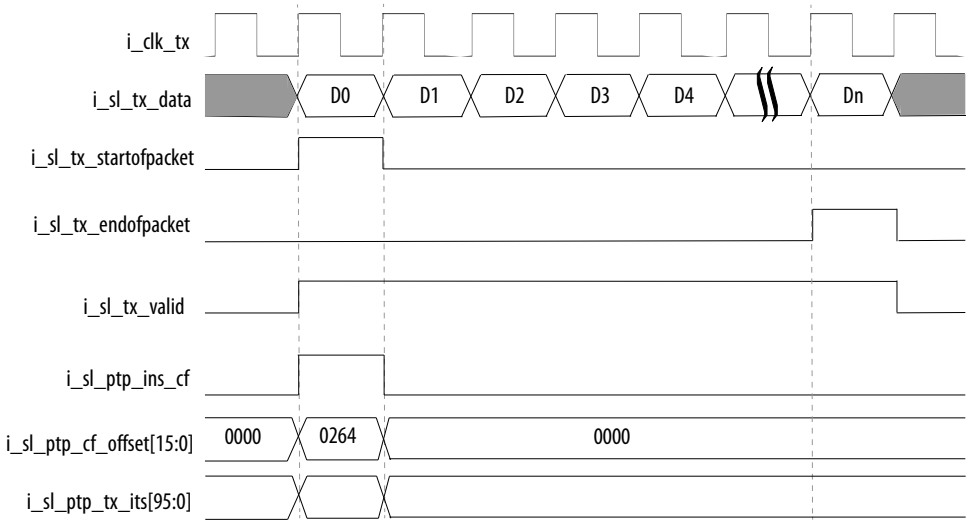
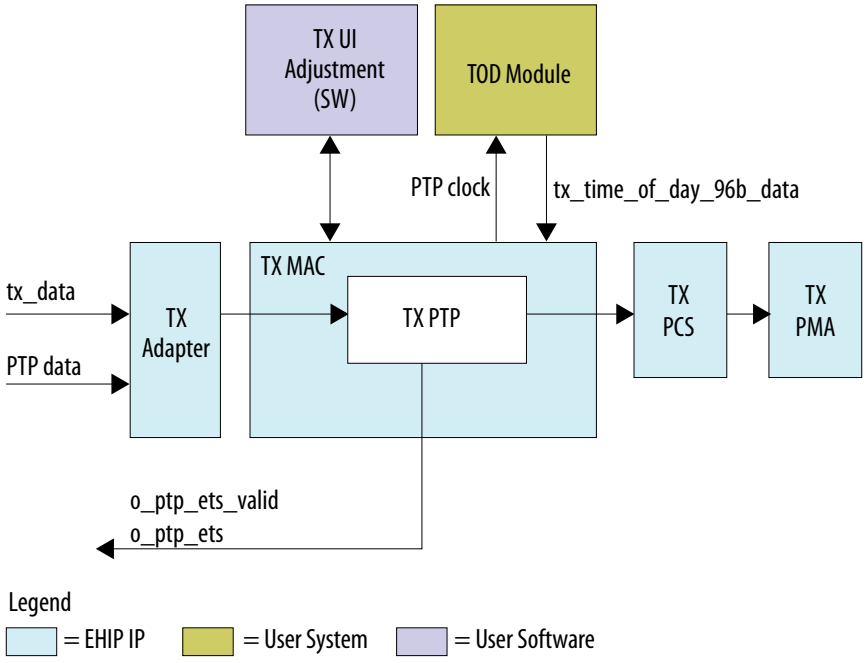


Figure 26. Example Waveform for 1-step TX Timestamp using `i_sl_ptp_ins_cf` Signal



The IP core transmits the 1588 PTP packet in an Ethernet frame after PTP processing.

Figure 27. PTP Transmit Block Diagram



In one-step mode, the IP core either overwrites the timestamp information provided at the user-specified offset with the packet exit timestamp (insertion mode), or adds the residence time in this system to the value at the specified offset (correction mode). You tell the IP core how to process the timestamp by asserting the appropriate signal with the TX SOP signal. You must specify the offset of the timestamp in the packet (`i_ptp_ts_offset`) in insertion mode, or the offset of the correction field in

the packet (`i_ptp_cf_offset`) in correction mode. In addition, the IP core zeroes out or updates the UDP checksum, or leaves the UDP checksum as is, depending on the mutually exclusive `i_ptp_zero_csum` and `i_ptp_update_eb` signals.

Note: If the PTP packet resides in the system for more than 4 seconds, the correction field will show a mismatched value with a very large number.

Two-step PTP processing ignores the values on the one-step processing signals. In two-step processing mode, the IP core does not modify the current timestamp in the packet. Instead, the IP core transmits a two-step derived timestamp on the separate `o_ptp_ets[95:0]` bus, when it begins transmitting the Ethernet frame. The value on the `o_ptp_ets` bus is the packet exit timestamp. The `o_ptp_ets` bus holds a valid value when the corresponding `o_ptp_ets_valid` signal is asserted.

In addition, to help the client to identify the packet, you can specify a fingerprint to be passed by the IP core in the same clock cycle with the timestamp. The E-Tile Hard IP for Ethernet Intel FPGA IP has a fixed 8 bit width for fingerprint. You provide the fingerprint value to the IP core in the `i_ptp_fp` signal. The IP core then drives the fingerprint on the appropriate `o_ptp_ets_fp` port with the corresponding output timestamp, when it asserts the `o_ptp_ets_valid` signal.

The IP core calculates the packet exit timestamp using reference block timing. The egress time of blocks which marked as references, are measured directly at the serializer, and are used to calculate the egress times of all other bits.

Related Information

IEEE website
The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

2.9.2.4. PTP Receive Functionality

If you turn on **Enable IEEE 1588 PTP** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor, the IP core provides a 96-bit (V2 format) with every packet on the RX client interface, whether it is a 1588 PTP packet or not. The value on the timestamp bus `o_ptp_rx_its` is valid in the same clock cycle as the RX SOP signal. The value on the timestamp bus is not the current timestamp; instead, it is the timestamp from the time when the IP core received the packet on the Ethernet link. The IP core captures the time-of-day from the TOD module on `i_ptp_tod` at the time it receives the packet on the Ethernet link, and sends that timestamp to the client on the RX SOP cycle on the timestamp bus `o_ptp_rx_its`. User software/hardware logic can use this timestamp or ignore it based on whether it is a 1588 PTP packet or not.

The RX PTP operation assumes the `o_sl_rx_ptp_ready` signal was asserted and is held high.

Figure 28. PTP Receive Block Diagram

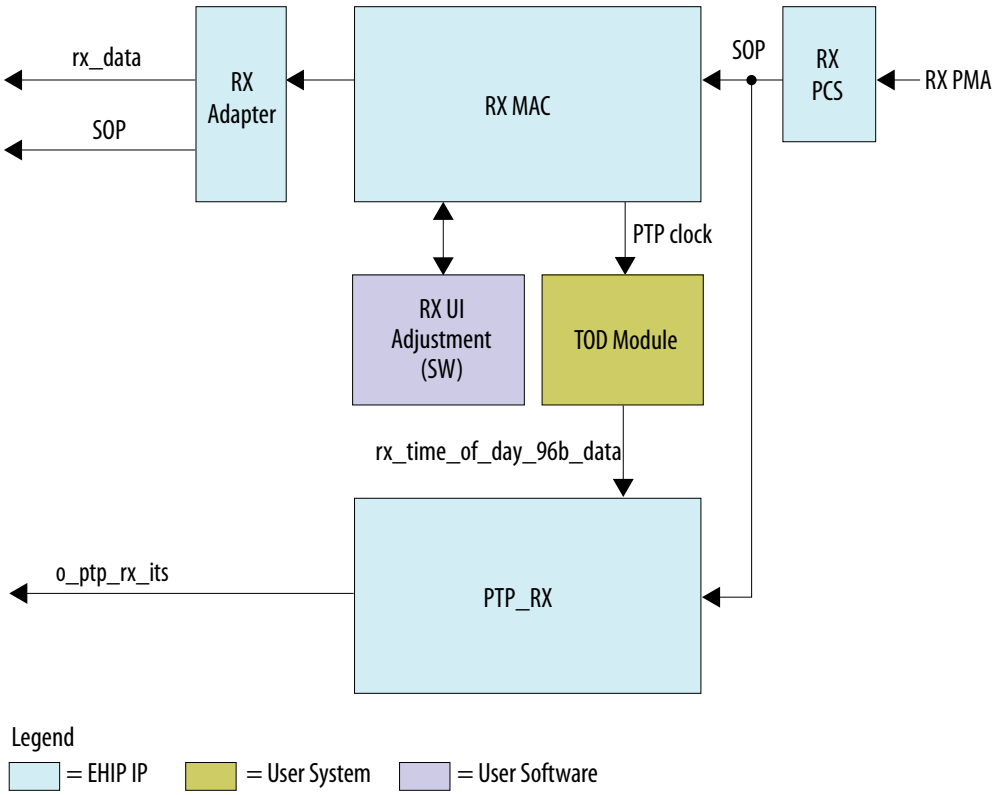
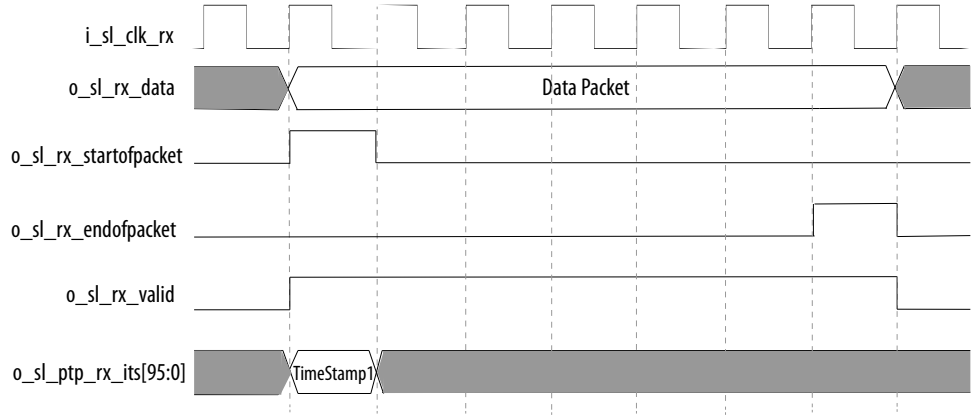


Figure 29. Example Waveform PTP Timestamp on RX PTP Interface



Related Information

IEEE website

The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

2.9.2.5. External Time-of-Day Module for 1588 PTP Variations

E-Tile Hard IP for Ethernet Intel FPGA IP that include the 1588 PTP module require an external time-of-day (TOD) module to provide the current time-of-day in each clock cycle, based on the incoming clock. For more information, refer to the *External Time-of-Day Module for Variations with 1588 PTP Feature*.

Note: You will observe a difference of 2 alignment marker period between the timestamp of the external ToD and TAM in the E-Tile Hard IP for Ethernet Intel FPGA IP when the external ToD module is set to use 16 bits fractional nanoseconds. However, the TAM module has a 24 bit fns representation to do the drift correction and does not rely on the external ToD timestamp.

Related Information

External Time-of-Day Module for Variations with 1588 PTP Feature on page 50

2.9.2.6. PTP Timestamp and TOD Formats

The E-Tile Hard IP for Ethernet Intel FPGA IP supports a 96-bit timestamp (V2 format) or a 64-bit timestamp (correction-field format) in PTP packets.

The IP core completes all internal processing in the V2 format. However, if you specify V1 format for a particular PTP packet in one-step insertion mode, the IP core inserts the appropriate V1-format timestamp in the outgoing packet on the Ethernet link.

V2 Format

The IP core maintains the time-of-day (TOD) in V2 format according to the IEEE specification::

- Bits [95:48]: Seconds (48 bits).
- Bits [47:16]: Nanoseconds (32 bits). This field overflows at 1 billion.
- Bits [15:0]: Fractions of nanosecond (16 bits). This field is a true fraction; it overflows at 0xFFFF.

V1 Format

V1 timestamp format is specified in the IEEE specification:

- Bits [63:32]: Seconds (32 bits).
- Bits [31:0]: Nanoseconds (32 bits). This field overflows at 1 billion.

Note: PTP packet with V1 timestamp format and Extended bytes is not supported for channel that uses a mixture of V1 and V2 timestamp formats.

Correction Field Format

The Correction Field format is distinct from the V1 format. It is intended for use in transparent clock systems, in which each node adds its own residence time to a running total latency through the system. This format matches the format of the correction field in the packet, as used in transparent clock mode.

- Bits [63:16]: Nanoseconds (48 bits).
- Bits [15:0]: Fractions of nanosecond (16 bits). This field is a true fraction; it overflows at 0xFFFF.

Related Information

IEEE website
The *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard* is available on the IEEE website.

2.9.2.7. 10G/25G TX and RX Unit Interval Adjustment

The accuracy drift occurs when there is a PPM difference between the MAC layer clock and the Master TOD clock. To correct the drift, you may use the RX and TX Unit Interval (UI) adjustment method. IEEE 802.3 standards permit the Ethernet clocks frequency to vary within ±100 PPM.

For the system with PPM, you are required to do at least one UI adjustment once out of reset. Any timestamp before the UI adjustment is invalid. The UI adjustment shall be performed after the `rx/tx_ptp_ready` is asserted and before starting any PTP operation. The UI adjustment applies to the system independent of PPM as long as the PPM is within the Ethernet specification. Once UI adjustment is completed, it takes two Alignment marker periods for UI to adjust and produce an accurate timestamp.

The UI adjustment is a software flow, implemented by user software, utilizing 10G/25G PTP PPM UI Adjustment registers in the 1588 PTP Registers on page 259 section to compute the new UI. Once determined, the UI register updates the UI value.

10G/25G UI Adjustment Calculation

The UI Adjustment Calculation applies to both, TX and RX paths. The software must take at least two snapshots of the 10G/25G PTP PPM UI Adjustment registers within 1s time frame. The new UI value is computed from the difference between the two snapshots. The longer the interval between the first and the next snapshot, the better accuracy the new UI value provides.[8] The accuracy can also be improved by taking multiple snapshots within 1s time frame. If two compared snapshots are taken outside of the 1s time frame, they shall be discarded.

Note: The UI adjustment shall not be performed when TOD changes are in progress or during reset.

User software performs the following steps to calculate the new UI value.

1. Set `TAM_SNAPSHOT` to 1 to take the first snapshot of the Time of Alignment Marker (TAM) and the Alignment Marker Count (AM_Count) values. The user software snapshots the TAM and AM_Count values into the `TX/RX_TAM_H/L` and `TX/RX_COUNT` registers located in the 1588 PTP Registers on page 259.
2. Read `TX/RX_TAM_H/L` and `TX/RX_COUNT` registers and save values from the first snapshot as:

[8] Comparing the first snapshot with a third snapshot taken at a later time will give a more accurate UI than comparing the first snapshot with the second snapshot taken at an earlier time.

- TX_TAM_0 = {TX_TAM_H, TX_TAM_L}
- TX_Count_0 = TX_COUNT
- RX_TAM_0 = {RX_TAM_H, RX_TAM_L}
- RX_Count_0 = RX_COUNT

3. Clear TAM_SNAPSHOT to 0 to complete the first snapshot.
4. Set TAM_SNAPSHOT to 1 again to take the N^{th} snapshot of the TAM and AM_Count values where N represents the number of taken snapshots.
5. Read TX/RX_TAM_H/L and TX/RX_COUNT registers and save values from the N^{th} snapshot:
   - TX_TAM_N = {TX_TAM_H, TX_TAM_L}
   - TX_Count_N = TX_COUNT
   - RX_TAM_N = {RX_TAM_H, RX_TAM_L}
   - RX_Count_N = RX_COUNT
6. Ensure the interval between the 1^{th} snapshot and the N^{th} snapshot is within 1s else the snapshots are invalid for use and the user software needs to restart from step 1.
7. Calculate the new UI value:

   UI = TAM_Interval / (AM_Count * Reference_Time_Load_Interval).

   Refer steps below to calculate TAM_Interval, AM_Count, and est_AM_Count.

   Note: For more information on Reference Time Load Interval, refer to the Table 23 on page 94.
8. Compute the TAM_Interval value. TAM is a time value in nanosecond (ns). Once the TAM value reaches 1 billion ns, it rolls over to 1s creating a rollover condition. When TAM value rolls over, the subsequent TAM_N value will appear smaller than the first TAM_0 value.
   - If (TAM_N > TAM_0): TAM_Interval (ns) = (TAM_N - TAM-_0)
   - If (TAM_N ≤ TAM_0): TAM_Interval (ns) = ((1s + TAM_N) - TAM_0)

   To calculate the TX_TAM_Interval, replace TAM_N with TX_TAM_N value and TAM_0 with TX_TAM_0 value. Same steps apply to calculate the RX_TAM_Interval.
9. Calculate the estimated AM Count (est_AM_Count). The est_AM_Count value is used to ensure that AM_Count, calculated in next step, is valid.

   est_AM_Count = INT (TAM_Interval / (Reference_Time_Load_Interval * 0ppm_UI)

   where INT() is a round up function to the nearest integer, the value of 0ppm_UI for 10GE variant is 96.969696 ps and the value of 0ppm_UI for 25GE variant is 38.787878 ps. Compare the est_AM_Count with 64,000, which is the maximum AM_Count with offset. If est_AM_count exceeds 64,000, discard the snapshot and start from step 1.
10. Calculate the AM_Count value. AM_Count can reach a rollover condition when reaching maximum value. Use the appropriate equation to calculate the AM_Count.

- If (Count_N > Count_0): AM_Count = (Count_N – Count_0)
- If (Count_N ≤ Count_0): AM_Count = ((65,535 – Count_0) + Count_N)

11. Write the calculated TX and RX UI values to TX_UI_REG and RX_UI_REG registers.

Table 23. Reference Time Load Interval

This table shows the Reference time load interval values for PPM UI adjustment calculation.

Configuration	Data Path	Reference_Time_Load_Interval value in bits
25G variant with RS-FEC	TX	81,920*66=5,406,720 *Note:* 81,920 is a 66-bit value of alignment marker interval (in terms of block numbers)
	RX	81,920*66=5,406,720
10GE/25GE variants without RS-FEC	TX	81,920*66=5,406,720
	RX	6,336
100G with and without RS-FEC	TX	16384*20*66=21,626,880 *Note:* 16384 is a 66-bit value of alignment marker interval (in terms of block numbers) per virtual lane.
	RX	

2.9.2.8. 10G/25G TX and RX PTP Extra Latency

TX_PTP_EXTRA_LATENCY and RX_PTP_EXTRA_LATENCY registers define extra latency that IP core adds to the outgoing TX and the incoming RX timestamps. This time offset applies to all time values processed by the TX and RX PTP logic. It can be used to account for known errors on the PCB, or in other parts of the system.

According to IEEE 1588 PTP Standard, the timestamping point is the time when the first bit after the Start of Frame Delimiter (SFD) crosses Medium Dependent Interface (MDI). The TX and RX PTP Extra Latency calculation shown below only generates a timestamp when the SFD crosses the serial pin of an FPGA. To represent the time at the MDI, you should consider adding a delay of external PHY to the TX and RX PTP Extra Latency registers.

Table 24. TX and RX PMA Delay

The table specifies transmitter and receiver PMA delay.

Configuration	Datapath	PMA Delay Constant (in Hardware)	PMA Delay Constant (in Simulation)
10G	TX	105	105
	RX	89	91
25G with and without RS-FEC	TX	105	107
	RX	89	94
100G with and without RS-FEC	TX	105	107
	RX	89	95

Steps to Calculate TX PTP Extra Latency

1. Determine TX PMA delay from the *TX and RX PMA Delay* table.
2. Calculate the TX PTP Extra Latency:
   - TX_PTP Extra Latency = TX PMA Delay * UI period (in ns)
3. Write the calculated TX PTP Extra Latency to the TX_PTP_EXTRA_LATENCY register.

Steps to Calculate RX PTP Extra Latency

1. Determine RX PMA delay from the *TX and RX PMA Delay* table.
2. Calculate the RX PTP Extra Latency:

   RxCWPos represents a number of bit slips required to achieve RS-FEC alignment. Read PMA AVMM register 0x29[4:0] to obtain this value.

   RxBitSlip represents the number of bit slips required to achieve a block alignment.
   - RX PTP Extra Latency = -((RX PMA Delay + RxCWPos) * UI period (in ns)) for 25G with RS-FEC.
   - RX PTP Extra Latency = -((RX PMA Delay + RxBitSlip - 66) * UI period (in ns)) for 10G/25G without RS-FEC when RxBitSlip is greater than 62.
   - RX PTP Extra Latency = -((RX PMA Delay + RxBitSlip) * UI period (in ns)) for 10G and 25G without RS-FEC when RxBitSlip is smaller than or equal to 62.
3. Write the calculated RX PTP Extra Latency to the RX_PTP_EXTRA_LATENCY register.

2.9.2.9. 100G PTP TX User Flow

In this section, the acronym PL and VL stands for Physical Lane and Virtual Lane respectively.

1. After power up or reset, wait until TX raw offset data are ready.

   The status could be monitored via:
   - Polling via CSR:

     ```
     csr_read (mlptp_status[0]) = 1'b1
     ```
2. Read TX raw offset data from IP:

   ```
   tx_const_delay = csr_read (mlptp_tx_const_delay[30:0])
   tx_const_delay_sign = csr_read (mlptp_tx_const_delay[31])

   for (pl = 0; pl < PL; pl++) {
       tx_apulse_offset[pl] = csr_read (mlptp_tx_l<3-0>_offset[30:0])
       tx_apulse_offset_sign[pl] = csr_read (mlptp_tx_l<3-0>_offset[31])
       tx_apulse_wdelay[pl] = csr_read (mlptp_tx_l<3-0>_wire_dly[19:0])
       tx_apulse_time[pl] = csr_read (mlptp_tx_l<3-0>_time[19:0])
   }
   ```
3. Determine TX reference lane:
   a. Detect if asynchronous pulse time rollover, and adjust accordingly:

      ```
      Expectation: Skew between async pulses < 256ns

      tx_ap_time_rollover = ((max(tx_apulse_time[PL-1:0]) –
      min(tx_apulse_time[PL-1:0])) > 256ns) ? 1 : 0
      ```

```
tx_ap_time_seconds_rollover = (tx_ap_time_rollover && (29'h10000000 –
max(tx_apulse_time[PL-1:0])) > 256ns) ? 1 : 0

for (pl = 0; pl < PL; pl++) {
tx_apulse_time_adj[pl] = tx_ap_time_seconds_rollover &&
(tx_apulse_time[pl] < 256ns) ? tx_apulse_time[pl] + {13'h0A00,
16'h0000} : tx_ap_time_rollover && (tx_apulse_time[pl] < 256ns) ?
tx_apulse_time[pl] + {13'h1000, 16'h0000} : tx_apulse_time[pl]
}
```

b. Calculate the actual time of TX Alignment Marker at TX PMA parallel data interface:

```
for (pl = 0; pl < PL; pl++) {
tx_am_actual_time[pl] = (tx_apulse_time_adj[pl]) +
(tx_apulse_offset_sign[pl] ? –tx_apulse_offset[pl] :
tx_apulse_offset[pl]) – (tx_apulse_wdelay[pl])
}
```

c. Determine TX reference lane:

```
tx_ref_pl = pl

where tx_am_actual_time[pl] is max(tx_am_actual_time[PL-1:0])
```

4. Calculate TX offsets:
   a. Calculate TX TAM adjust:

```
tx_tam_adjust = (tx_const_delay_sign ? –tx_const_delay :
tx_const_delay) + (tx_apulse_offset_sign[tx_ref_pl] ? –
tx_apulse_offset[tx_ref_pl] : tx_apulse_offset[tx_ref_pl]) –
(tx_apulse_wdelay[tx_ref_pl])
```

   b. Calculate TX extra latency:

      Convert unit of TX PMA delay from UI to nanoseconds:

```
tx_pma_delay_ns = tx_pma_delay_ui * UI
```

      Total up all extra latency together:

```
tx_extra_latency = tx_pma_delay_ns + tx_external_phy_delay +
tx_tam_adjust
```

5. Write the determined TX reference lane into IP:

```
csr_write (mlptp_tx_ref_lane, tx_ref_pl)
```

6. Write the calculated TX offsets to IP:
   a. Write TX extra latency:

```
csr_write (tx_ptp_extra_latency, tx_extra_latency)
```

7. UI value measurement. Follow the steps mentioned in section 100G UI Adjustment on page 101.

   Note: As UI measurement is a long process in simulation. Therefore, for simulation, Intel recommends to skip this step and program 0 ppm value.

8. Notify soft PTP that user flow configuration is completed:

```
csr_write (mlptp_tx_user_cfg, 1'b1)
```

9. Wait until TX PTP is ready.

The status could be monitored via:

- Output port:

```
o_tx_ptp_ready = 1'b1
```

or

- Polling via CSR:

```
csr_read (mlptp_status[2]) = 1'b1
```

10. TX PTP is up and running.
    a. Adjust TX UI value. You must perform TX UI adjustment of IP from time to time to prevent time counter drift from golden time-of-day in the system. Follow the steps mentioned in section 100G UI Adjustment on page 101.

       Note: UI measurement is a long process in simulation. Therefore, for simulation, Intel recommends to skip this step and program 0 ppm value.

2.9.2.10. 100G PTP RX User Flow

In this section, the acronym PL and VL stands for Physical Lane and Virtual Lane respectively.

1. Wait until RX raw offset data are ready.

   The status could be monitored via:

   - Polling via CSR:

```
csr_read (mlptp_status[1]) = 1'b1
```

2. Read RX raw offset data from IP:
   - All variants:

```
rx_const_delay = csr_read (mlptp_rx_const_delay[30:0])
rx_const_delay_sign = csr_read (mlptp_rx_const_delay[31])

for (pl = 0; pl < PL; pl++) {
rx_apulse_offset[pl] = csr_read (mlptp_rx_l<3-0>_offset[30:0])
rx_apulse_offset_sign[pl = csr_read (mlptp_rx_l<3-0>_offset[31])
rx_apulse_wdelay[pl] = csr_read (mlptp_rx_l<3-0>_wire_dly[19:0])
rx_apulse_time[pl] = csr_read (mlptp_rx_l<3-0>_time[19:0])
}
```

   - FEC variants:

```
rx_fec_cw_pos_lane0 = csr_read (Hard FEC, RSFEC_CW_POS_RX[0][14:0])

for (pl = 0; pl < PL; pl++) {
    rx_fec_ln_mapping[pl] = csr_read (Hard FEC, RSFEC_LN_MAPPING_RX[pl]
[6:0])
    rx_fec_ln_skew[pl] = csr_read (Hard FEC, RSFEC_LN_SKEW_RX[pl][6:0])
}

rx_fec_lane0_pl_map = pl, where rx_fec_ln_mapping[pl] = 0
```

   For more information on RS-FEC registers, refer to the E-Tile Transceiver PHY User Guide.

3. Determine RX reference lane:
   a. Determine sync pulse (Alignment Marker) offsets with reference to async pulse:

- FEC variants:

```
for (pl = 0; pl < PL; pl++) {
    rx_spulse_offset[pl] = ((rx_fec_ln_skew[rx_fec_lane0_pl_map] -
rx_fec_ln_skew[pl]) * 80 + rx_fec_cw_pos_lane0[4:0]) * UI
}
```

- No FEC variants:

```
for (vl = 0; vl < VL; vl++) {
    rx_spulse_offset[vl] = < Refer to section: 100G RX Virtual Lane
Offset Calculation for No FEC Variants on page 100 >
```

b. Detect if asynchronous pulse time rollover, and adjust accordingly:

```
Expectation: Skew between async pulses < 256ns

rx_ap_time_rollover = ((max(rx_apulse_time[PL-1:0]) –
min(rx_apulse_time[PL-1:0])) > 256ns) ? 1 : 0
rx_ap_time_seconds_rollover = (rx_ap_time_rollover && (29'h10000000 –
max(rx_apulse_time[PL-1:0])) > 256ns) ? 1 : 0
for (pl = 0; pl < PL; pl++) {
rx_apulse_time_adj[pl] = rx_ap_time_seconds_rollover &&
(rx_apulse_time[pl] < 256ns) ? rx_apulse_time[pl] + {13'h0A00,
16'h0000} : rx_ap_time_rollover && (rx_apulse_time[pl] < 256ns) ?
rx_apulse_time[pl] + {13'h1000, 16'h0000} : rx_apulse_time[pl]
}
```

c. Calculate the actual time of RX Alignment Marker at RX PMA parallel data interface:

- FEC variants: Skip this step.
- No FEC variants:

```
for (vl = 0; vl < VL; vl++) {
local_pl = vl_to_pl_map(vl)
rx_am_actual_time[vl] = (rx_apulse_time[local_pl]) +
(rx_apulse_offset_sign[local_pl] ? –rx_apulse_offset[local_pl] :
rx_apulse_offset[local_pl]) – (rx_apulse_wdelay[local_pl]) +
(rx_spulse_offset[vl])
}
```

d. Determine RX reference lane:

- FEC variants:

```
rx_ref_pl = pl

where rx_fec_ln_skew[pl] is min(rx_fec_ln_skew[PL-1:0])
```

- No FEC variants:

```
rx_ref_vl = vl

where rx_am_actual_time[vl] is max(rx_am_actual_time[VL-1:0])

rx_ref_pl = vl_to_pl_map(rx_ref_vl)
```

4. Calculate RX offsets:
   a. Calculate RX TAM adjust:

- FEC variants:

```
rx_tam_adjust = (rx_const_delay_sign ? –rx_const_delay :
rx_const_delay) + (rx_apulse_offset_sign[rx_ref_pl] ? –
rx_apulse_offset[rx_ref_pl] : rx_apulse_offset[rx_ref_pl]) –
(rx_apulse_wdelay[rx_ref_pl]) + (rx_spulse_offset[rx_ref_pl])
```

- No FEC variants:

```
rx_tam_adjust = (rx_const_delay_sign ? –rx_const_delay :
rx_const_delay) + (rx_apulse_offset_sign[rx_ref_pl] ? –
rx_apulse_offset[rx_ref_pl] : rx_apulse_offset[rx_ref_pl]) –
(rx_apulse_wdelay[rx_ref_pl]) + (rx_spulse_offset[rx_ref_vl])
```

b. Calculate RX extra latency:

Convert unit of RX PMA delay from UI to nanoseconds:

```
rx_pma_delay_ns = rx_pma_delay_ui * UI
```

Total up all extra latency together:

```
rx_extra_latency = –rx_pma_delay_ns – rx_external_phy_delay +
rx_tam_adjust
```

c. Calculate RX virtual lane offsets:

```
for (vl = 0; vl < VL; vl++) {
    if (remote_vl(vl) == 0 to 3)
         rx_vl_offset[vl]  = (2 – 330) * UI
    else
         rx_vl_offset[vl]  = 2 * UI
}
```

5. Write the determined RX reference lane into IP:

```
csr_write (mlptp_rx_ref_lane, rx_ref_pl)
```

6. Write the calculated RX offsets to IP:

a. Write RX virtual lane offsets:

```
for (vl = 0; vl < VL; vl++) {
    csr_write (vl<19-0>_offset_cfg_<0/1>,  rx_vl_offset[vl] )
}
```

b. Write RX extra latency:

```
csr_write (rx_ptp_extra_latency, rx_extra_latency)
```

7. UI value measurement. Follow the steps mentioned in section 100G UI Adjustment on page 101.

Note: As UI measurement is a long process in simulation. Therefore, for simulation, Intel recommends to skip this step and program 0 ppm value.

8. Notify soft PTP that user flow configuration is completed:

```
csr_write (mlptp_rx_user_cfg, 1’b1)
```

9. Wait until RX PTP is ready.

The status could be monitored via:

- Output port:

```
o_rx_ptp_ready = 1'b1
```

or

- Polling via CSR:

```
csr_read (mlptp_status[3]) = 1'b1
```

10. RX PTP is up and running.
    a. Adjust RX UI value. You must perform RX UI adjustment of IP from time to time to prevent time counter drift from golden time-of-day in the system. Follow the steps mentioned in section 100G UI Adjustment on page 101.

       Note: UI measurement is a long process in simulation. Therefore, for simulation, Intel recommends to skip this step and program 0 ppm value.

2.9.2.11. 100G RX Virtual Lane Offset Calculation for No FEC Variants

1. Lock the RX PCS. The RX PCS must be fully aligned before extracting VL Offfset data. Wait for `o_rx_pcs_fully_aligned` to be asserted.
2. Read the VL data for each of the Local Virtual Lanes (VL):

   Perform the read operation via CSR interface. This returns `VL data field` (offsets: 0xC40 - 0xC67) data for all 20 virtual lanes.

   For example, if you read VL data for Local lane 0, you get `GBSTATE[0]`, `BA_PHASE[0]`, `BA_POS[0]`, `AM_COUNT[0]`, and `REMOTE_VL[0]`. `LOCAL_VL` will be set to 0, and `LOCAL_PL` will be set to 0.
3. Set the Physical Lanes for each Remote Virtual Lane:

   Step through the data provided by each Local Virtual lane:

   - Each lane provides a `REMOTE_VL` and a `LOCAL_PL` value.

   For each of the 20 possible Remote virtual lanes, set `PL[REMOTE_VL] = LOCAL_PL`.

   For example, if you read Local Virtual lane 12, and get back `REMOTE_VL` = 5, `LOCAL_PL` = 2, that means the data for Virtual lane 5 from the link partner is coming in on our Physical lane 2. You store that as PL[5] = 2.
4. Calculate the Virtual Lane Offsets for each of the Remote Virtual Lanes in bits:

```
Step through the data provided by each Local Virtual lane
Note: % is the modulo operator.

Let sublane = LOCAL_VL % 5
    For example, if you are processing data from Local VL12, sublane is 2

Let bit_offset = ((BA_POS-21)*5-BA_PHASE*22+sublane)%66
    For example, suppose you get BA_POS=4, BA_PHASE=0 on sublane 2
    bit_offset = ((4-21)*5-0*22+2)%66 = (-85+2)%66 = 49

    Bit offset indicates the number of bits away from bit 0 of an incoming
    PL 66b block the alignment logic must shift data to align it

Let gb_shift = (GBSTATE-4 +BA_PHASE*3)%5
    For example, suppose you get BA_PHASE = 0, and GBSTATE = 3
    gb_shift = (2 – 4 + 0*3)%5 = (-1)%5 = 4

Let am_shift = (AM_COUNT > (am_interval+1)/2) ? (AM_COUNT-(am_interval+1)):
```

```
AM_COUNT
    For example, suppose you get AM_COUNT=63, and am_interval is set to 63
    63>(63+1)/2, so am_shift = (63 – (63+1)) = -1
    Note that it is ok for am_shift to be negative

Let am_offset = am_shift * 5 + gb_shift
    In this example, am_shift = -1, and gb_shift = 4,
    therefore am_offset = -1 * 5 +4 = -1

Set proc_offset as follows:
    If(BA_PHASE == 0) {
       if(BA_POS ==21) { proc_offset = 4;}
       else if(BA_POS > 7) { proc_offset = 5;}
       else if(sublane ==4 && BA_POS == 6){proc_offset = 5;}
       else{proc_offset = 6;}
    }
    else if(BA_PHASE == 1){
       if(BA_POS > 12) { proc_offset = 5;}
       else if(sublane > 0 && BA_POS == 12){proc_offset = 5;}
       else{proc_offset = 6;}
    }
    else{
       if(BA_POS > 17) { proc_offset = 5;}
       else if(sublane >2 && BA_POS == 16){proc_offset = 5;}
       else if(BA_POS >3){proc_offset = 6;}
       else if(sublane >1) && BA_POS==3){proc offset =6;}
       else{proc_offset = 7;}
    }
For example:
    if sublane == 2, BA_PHASE==0, and BA_POS==4, proc_offset = 6
    Set vl_offset_bits[REMOTE_VL] = (bit_offset – (am_offset +
proc_offset)*66)

For example:
    if REMOTE_VL=5, bit_offset = 49, am_offset = -1, and proc_offset = 6
    Set vl_offset_bits[5]  = (49 – (-1+6)*66) = -281 bits
In this case, the reference block actually arrived 281 UI later than the
Alignment marker. To compute the time that the alignment marker arrived,
take the time of the reference block arrival, and subtract 281.
```

5. Convert the `vl_offset_bits` to VL_OFFSET (in ns):

```
For each REMOTE_VL, VL_OFFSET[REMOTE_VL] =
vl_offset_bits_shifted[REMOTE_VL] * RX_UI
```

2.9.2.12. 100G UI Adjustment

This section provides the steps to measure UI value to prevent time drift due to clock PPM as reference to golden Time-of-Day (ToD) in the platform.

Tip: You should not perform the measurement if there is a large adjustment to the ToD value, as it will impact the measurement result which is based on principle that the ToD value will be incremented constantly based on golden ToD running at ideal clock with 0ppm.

TAM is reference time for Alignment Marker (AM) for variants with AM, and arbitrary selected reference bit for variants without AM.

1. Request snapshot of initial TX TAM and RX TAM:

```
csr_write (mlptp_tam_snapshot, 1'b1)
```

2. Read snapshotted initial TAM and counter values:

```
tx_tam_0_31_0 = csr_read (mlptp_tx_tam_l[31:0])
tx_tam_0_47_32 = csr_read (mlptp_tx_tam_h[15:0])
tx_tam_0_cnt = csr_read (mlptp_tx_am_count[15:0])

rx_tam_0_31_0 = csr_read (mlptp_rx_tam_l[31:0])
rx_tam_0_47_32 = csr_read (mlptp_rx_tam_h[15:0])
rx_tam_0_cnt = csr_read (mlptp_rx_am_count[15:0])

Form complete TAM by concatenation

tx_tam_0 = {tx_tam_0_47_32, tx_tam_0_31_0};
rx_tam_0 = {rx_tam_0_47_32, rx_tam_0_31_0};
```

3. Clear snapshot:

```
csr_write (mlptp_tam_snapshot, 1'b0)
```

4. Starting from time when step 1 is executed, wait for time duration as specified in section Minimum and Maximum Reference Time (TAM) Interval for UI Measurement (Hardware) on page 103.
5. Request snapshot of Nth TX TAM and RX TAM:

```
csr_write (mlptp_tam_snapshot, 1'b1)
```

6. Read snapshotted Nth TAM and counter values:

```
tx_tam_n_31_0 = csr_read (mlptp_tx_tam_l[31:0])
tx_tam_n_47_32 = csr_read (mlptp_tx_tam_h[15:0])
tx_tam_n_cnt = csr_read (mlptp_tx_am_count[15:0])

rx_tam_n_31_0 = csr_read (mlptp_rx_tam_l[31:0])
rx_tam_n_47_32 = csr_read (mlptp_rx_tam_h[15:0])
rx_tam_n_cnt = csr_read (mlptp_rx_am_count[15:0])

Form complete TAM by concatenation

tx_tam_n = {tx_tam_n_47_32, tx_tam_n_31_0};
rx_tam_n = {rx_tam_n_47_32, rx_tam_n_31_0};
```

7. Clear snapshot:

```
csr_write (mlptp_tam_snapshot, 1'b0)
```

8. Calculation:
   a. Get TAM interval (tx_tam_interval and rx_tam_interval) from Table 23 on page 94.
   b. Calculate time elapsed:

```
tx_tam_delta = (tx_tam_n <= tx_tam_0) ? [(tx_tam_n + 10^9ns) –
tx_tam_0] : (tx_tam_n – tx_tam_0)

rx_tam_delta = (rx_tam_n <= rx_tam_0) ? [(rx_tam_n + 10^9ns) –
rx_tam_0] : (rx_tam_n – rx_tam_0)
```

   Note: 10^9ns = 48'h 3B9A_CA00_0000

   c. Calculate estimated count value:

```
tx_tam_cnt_est    = round_up_to_nearest_integer(tx_tam_delta /
(tx_tam_interval * 0ppm_UI))
rx_tam_cnt_est    = round_up_to_nearest_integer(rx_tam_delta /
(rx_tam_interval * 0ppm_UI))

Where 0ppm_UI is 38.7878 ps.
```

```
If tx_tam_cnt_est or rx_tam_cnt_est > 64000, which indicates 16-bit
counters may have already overflow 2 times or more, you must restart
from step 1.
```

This step is optional, and required only if snapshot may happen longer than duration specified in Minimum and Maximum Reference Time (TAM) Interval for UI Measurement (Hardware) on page 103. For example: The read response of System-console via JTAG master is slow.

d. Calculate TAM count value:

```
tx_tam_cnt = (tx_tam_n_cnt < tx_tam_0_cnt) ? [(tx_tam_n_cnt + 2^16) –
tx_tam_0_cnt] : (tx_tam_n_cnt – tx_tam_0_cnt)

rx_tam_cnt = (rx_tam_n_cnt < rx_tam_0_cnt) ? [(rx_tam_n_cnt + 2^16) –
rx_tam_0_cnt] : (rx_tam_n_cnt – rx_tam_0_cnt)
```

e. Calculate UI value:

```
tx_ui = (tx_tam_delta) / (tx_tam_cnt * tx_tam_interval / 4)
rx_ui = (rx_tam_delta) / (rx_tam_ cnt * rx_tam_interval / 4) * 330
```

9. Write the calculated UI value to IP:

```
csr_write (mlptp_tx_ui, tx_ui)
csr_write (mlptp_rx_ui, rx_ui)
```

Ensure the format is {4-bit nanoseconds, 24-bit fractional nanoseconds}.

10. After first UI measurement, for every minimum TAM interval or longer duration, repeat step 1 to 9. This is to prevent time counter drift from golden Time-of-Day in the system whenever clock ppm changes.

2.9.2.13. Minimum and Maximum Reference Time (TAM) Interval for UI Measurement (Hardware)

Table 25. Minimum and Maximum Number of Reference Time Interval Allowed for UI Measurement

Mode	FEC Type	Minimum Number of Reference Time Load Interval				Maximum Number of Reference Time Load Interval			
		Number of Count, N		Miliseconds (ms)		Number of Count, N		Miliseconds (ms)	
		TX	RX	TX	RX	TX	RX	TX	RX
10G-1	No FEC	6	10	3.15	0.006	64000	64000	33554.43	39.32
25G-1	No FEC	6	10	1.26	0.002	64000	64000	13421.77	15.73
	KR-FEC	6	6	1.26	1.26	64000	64000	13421.77	13421.77
100G-4	No FEC KR-FEC	6	6	1.26	1.26	64000	64000	13421.77	13421.77

2.9.2.14. PTP System Considerations

This section provides list of generic guidelines required when using PTP IP.

- You need to wait for the o_tx_ptp_ready to be asserted before sending the PTP packet in TX direction.
- You shall ignore the RX timestamp when o_rx_ptp_ready is deasserted.
- Training sequence (of any packet type) is required on the RX direction to complete RX PTP deskew process and only then o_rx_ptp_ready can be asserted. After the reset and a PMA adaptation, the IP asserts the signal after the link partner sends up to 20 Ethernet packets. Therefore, the o_rx_ptp_ready signal can be asserted at a much later time than the o_tx_ptp_ready signal, depending on whether the link partner is sending any packets.
- The o_tx_ptp_ready is deasserted when triggering i_sl_tx_rsn_n or i_sl_csr_rst_n resets.
- The o_rx_ptp_ready is deasserted when triggering i_sl_tx_rsn_n, i_sl_rx_rsn_n, or i_sl_csr_rst_n. In the TX reset case, even though the o_rx_ptp_ready is deasserted, it doesn't reset RX PTP deskew logic and there is no need for training sequence to assert o_rx_ptp_ready again.
- The o_rx_ptp_ready signal is deasserted if the Ethernet link disconnects.
- If you run a PMA adaptation after the Ethernet link is established, the Ethernet link goes down again and come back once the PMA calibration is completed.
- During the reset or a major TOD update, you must wait for at most 2 Alignment Marker (AM) periods before sending a PTP packet in the TX direction. This provides sufficient time to load the new TAM value into the IP. You may observe a timestamp inaccuracy within these 2 AM periods. You can assume this AM period = (81.920 * 66 * 97 ps) = 524,451,840 ps for overall speed. You can ignore the RX timestamp within this period.

Note: For IEEE 1588 PTP enabled design, there is a temporary TX data corruption after the first link up event of power up sequence or a reset. As a result, the remote link partner observes temporary link down event and followed by the link up. TX packet corruption happens if there is packet transmission before the second link up event. Remote link partner is expected to handle link down event accordingly as per the IEEE specifications.

2.9.2.15. Logic Lock Regions Requirements for PTP Accuracy Advanced Mode

The PTP Accuracy advanced mode requires you to add the logic lock region assignment statement in the .qsf file. To add the logic lock assignments, copy the alt_ehipc3_10g25g_ptp_advancedmode_logiclockreg_gen.tcl script, located in the <Generated HDL directory>/alt_ehipc3_2021/synth/" directory into the same location as the .qsf project file.

Follow these steps to add logic lock regions to your .qsf file:

1. In Intel Quartus Prime Pro Edition software, run the initial compilation to place channels in the required location.
2. In the command line, execute the alt_ehipc3_10g25g_ptp_advancedmode_logiclockreg_gen.tcl script with the following command:

```
quartus_sta -t alt_ehipc3_10g25g_ptp_advancedmode_logiclockreg_gen.tcl -
project <project name> [-revision <revision name>] [-file <output file
name>]
```

Note: The project name is a required field. The revision name is an optional field.

3. Re-run the compilation step to compile the design with added logic lock regions `.qsf` assignments.

Whenever you perform additional design change or design optimization, you only need to execute step 3. If you change the channel placement or select a different E-tile, you must delete previously generated logic lock regions and execute steps 1 through 3 to regenerate the logic lock regions with the new channel location.

2.9.3. PCS, OTN, FlexE, and Custom PCS Modes

Each E-Tile Hard IP for Ethernet Intel FPGA IP instance contains a full featured multi-lane PCS layer, which offers a number of interfacing options from the FPGA fabric.

The IP offers the following PCS options:

- PCS Only - This mode uses the MII interface to transmit and receive Ethernet packets for data rate of 10/25/100 Gbps.
- OTN and FlexE - This mode uses the PCS66 interface to read and write 66 bit blocks data, from and to the PMA block.
- Custom PCS - This mode uses the MII interface to transmit and receive packets from non-Ethernet protocols with data rates with 2.5 to 28 Gbps.

Figure 30. TX PCS Datapath with and without RS-FEC

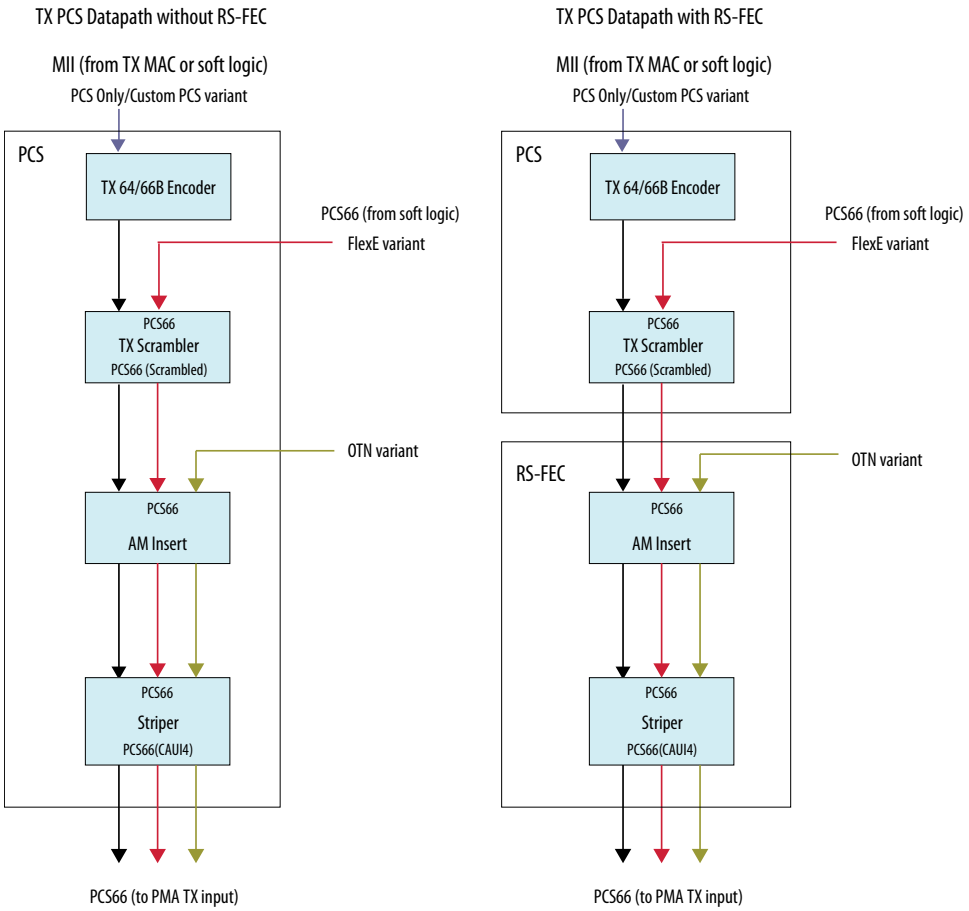
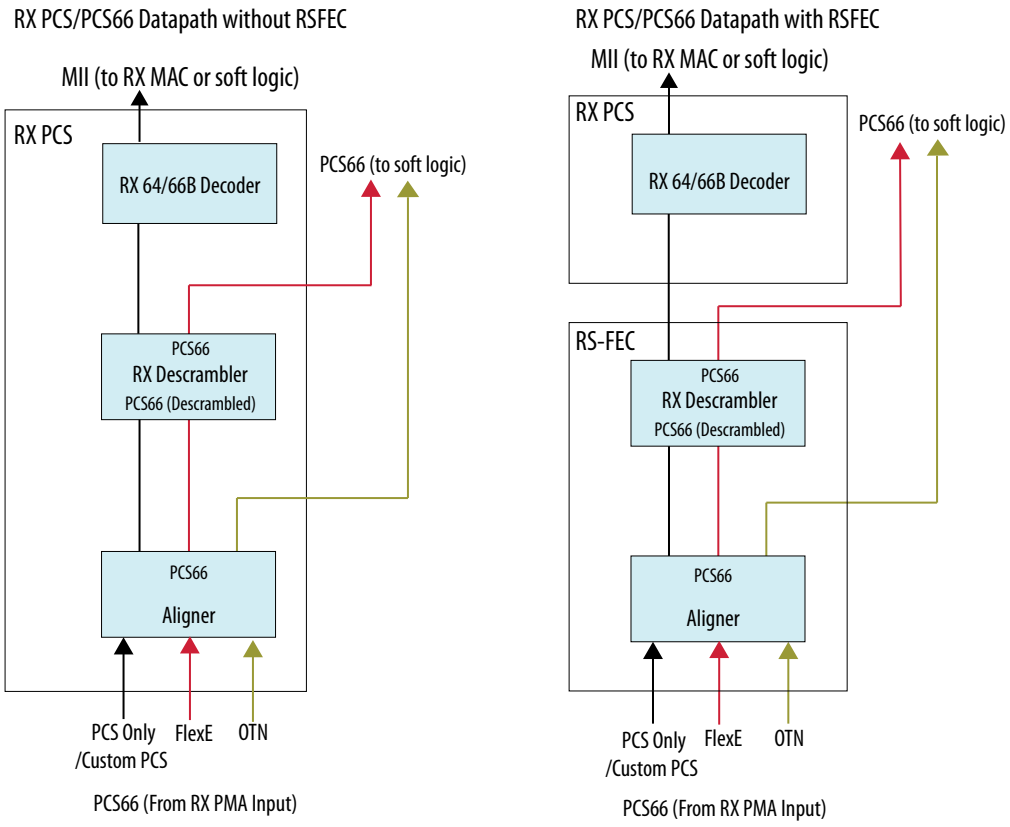


Figure 31. RX PCS Datapath with and without RS-FEC



2.9.3.1. PCS Only Mode

The E-Tile Hard IP for Ethernet Intel FPGA IP supports PCS only mode in 10/25G and 100G variants with optional RSFEC feature. It can support up to four PCS channels in 10/25G variant. This mode bypassed the Ethernet MAC and uses MII interface to read and write to the PMA block.

The PCS TX datapath consists of:

- TX PCS encoder—encodes the data from the PMA interface.
- TX PCS scrambler—enables the data to be scrambled. Channels will not lock correctly if the data is not scrambled.
- Alignment insertion—the TX PCS interface inserts alignment markers.
- Striper—enables logically sequential data to be segmented to increase data throughput.

The PCS RX datapath consists of:

- Aligner—enables the alignment of incoming data.
- RX PCS descrambler—enables the incoming scrambled data to be descrambled.
- RX PCS decoder—decodes the incoming encoded data from the PMA interface.

2.9.3.2. OTN Mode

The E-Tile Hard IP for Ethernet Intel FPGA IP supports OTN mode in 10/25G and 100G variants with optional RSFEC feature. It can support up to four OTN channels in 10/25G variant. This mode bypassed the Ethernet MAC and uses PCS66 interface to read and write to the PMA block.

The OTN TX datapath consists of:

- Alignment insertion—the TX PCS interface inserts alignment markers.
- Striper—enables logically sequential data to be segmented to increase data throughput.

Note: In OTN mode, scrambler is bypassed because the input data is expected to be scrambled.

The OTN RX datapath consists of an aligner block that enables the alignment of the incoming data.

2.9.3.3. FlexE Mode

The E-Tile Hard IP for Ethernet Intel FPGA IP supports FlexE mode in 10/25G and 100G variants with optional RSFEC feature. It can support up to four FlexE channels in 10/25G variant. This mode bypassed the Ethernet MAC and uses PCS66 interface to read and write to the PMA block.

The FlexE TX datapath consists of:

- TX PCS scrambler—enables the data to be scrambled. Channels will not lock correctly if the data is not scrambled.
- Alignment insertion—the TX PCS interface inserts alignment markers
- Striper—enables logically sequential data to be segmented to increase data throughput.

The FlexE RX datapath consists of:

- Aligner—enables the alignment of incoming data.
- RX PCS descrambler—enables the incoming scrambled data to be descrambled.

2.9.3.4. Custom PCS Mode

The E-Tile Hard IP for Ethernet Intel FPGA IP supports up to four custom PCS channels with RSFEC feature. This mode bypasses the Ethernet MAC and uses MII interface to transmit and receive packets with data rate between 2.5 to 28 Gbps.

Note: The custom PCS mode does not support auto-negotiation and link training features in the Intel Quartus Prime version 19.1.

The custom PCS TX datapath consists of:

- TX PCS encoder—encodes the data from the PMA interface.
- TX PCS scrambler—enables the data to be scrambled. Channels will not lock correctly if the data is not scrambled.
- Alignment insertion—the TX PCS interface inserts alignment markers.
- Striper—enables logically sequential data to be segmented to increase data throughput.

The PCS RX datapath consists of:

- Aligner—enables the alignment of incoming data.
- RX PCS descrambler—enables the incoming scrambled data to be descrambled.
- RX PCS decoder—decodes the incoming encoded data from the PMA interface.

2.9.4. Auto-Negotiation and Link Training

The E-Tile Hard IP for Ethernet Intel FPGA IP variations with auto-negotiation and link training implement the *IEEE Backplane Ethernet Standard 802.3-2012*.

The IP core includes the option to implement the following features:

- Auto-negotiation provides a process to explore coordination with a link partner on a variety of different common features. Turn on the **Enable AN/LT** and parameter to configure support for auto-negotiation. Turn on the **Enable Auto-Negotiation on Reset** parameter to enable auto-negotiation by default after reset.
- Link training provides a process for the IP core to train the link to the data frequency of incoming data while compensating for variations in process, voltage, and temperature. Turn on the **Enable AN/LT** parameter to configure support for link training. Turn on the **Enable Link Training on Reset** parameter to enable link training by default after reset. When enabled, link training performs the initial and continuous adaptation. For more details on adaptation modes, refer to the *E-Tile Transceiver PHY User Guide*.

The E-Tile Hard IP for Ethernet Intel FPGA IP includes separate auto-negotiation and link training modules for each of the 10G/25G channels. For 100G, the IP provides auto-negotiation functionality on a single channel specified by the **Auto-Negotiation Master** parameter and separate link training modules for each channel.

Related Information

- Auto Negotiation and Link Training Registers on page 184
- E-Tile Transceiver PHY User Guide

2.9.5. TX and RX RS-FEC

If you turn on **Enable RS-FEC** in the E-Tile Hard IP for Ethernet Intel FPGA IP parameter editor, the IP core includes Reed-Solomon forward error correction (FEC) in both the receive and transmit datapaths. This feature is only available in 25G and 100G variants.

The IP core implements Reed-Solomon FEC per Clause 91 of the IEEE Standard 802.3bj. The Reed-Solomon FEC algorithm includes the following modules:

- TX RS-FEC
  - 64b/66b to 256b/257b transcoding
  - High-Speed Reed-Solomon encoder
- RX RS-FEC
  - Alignment marker lock
  - 256b/257b to 64b/66b transcoding
  - High-Speed Reed-Solomon decoder

Related Information

E-Tile Transceiver PHY User Guide
More information about RS-FEC architecture.

2.9.6. PMA Direct Mode

The E-Tile Hard IP for Ethernet Intel FPGA IP provides an option to switch from 10G/25G MAC+PCS variant to use PMA only mode during run-time. In this mode, only the reconfiguration and PMA interfaces are enabled. The MAC interface to user logic connection is disabled. The additional interface when you enabled this mode are:

- `i_sl_tx_pma[ch-1:0]`
- `o_sl_rx_pma[ch-1:0]`

Enable the **Include alternate ports** in the parameter editor to expose the PMA direct signals. Set `tx_ehip_mode[2:0]` of register `Configuration Fields for TX PLD` (address 0x350) to 3'h7 to enable the PMA direct mode. This feature is supported only in **100GE or 1 to 4 10GE/25GE with optional RSFEC and 1588 PTP** variant.

Related Information

PMA Direct Interface on page 144

2.9.7. Dynamic Reconfiguration

You can dynamically reconfigure the settings in the E-Tile Intel FPGA IP to run your design in different data rates and features.

For more information on the steps and guidelines to use the dynamic reconfiguration feature, refer to the:

- E-Tile Dynamic Reconfiguration Design Example (for Intel Stratix 10 devices)
- E-Tile Dynamic Reconfiguration Design Example (for Intel Agilex devices)

2.9.8. Ethernet Adaptation Flow for 10G/25G and 100G/4x25G Dynamic Reconfiguration Design Example

Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage* sections in the *E-tile Transceiver PHY User Guide* for more details on the adaptation flow and how to get started.

This adaptation flow assumes a valid Ethernet traffic. 10GE/25GE variant uses the external AIB clocking. 100GE/4x25G variant is using a non-external AIB clocking.

1. Assert `i_sl_tx_rst_n/i_tx_rst_n/soft_tx_rst` and `i_sl_rx_rst_n/i_rx_rst_n/soft_rx_rst` signals.
2. Disable the PMA[9].
3. Trigger PMA analog reset[10]. Don't call the interrupt sequencer.
4. Perform dynamic reconfiguration sequence:
   a. Switch reference clock.
   b. Change reference clock ratio.
   c. Apply RX phase slip.
   d. Reconfigure AIB, EHIP, PCS, and enable/disable RS-FEC registers.
5. In 100GE/4x25GE variants, perform the dynamic reconfiguration reset. Dynamic Reconfiguration requires a staggered reset. For more information, refer to the reset sequence information in the *E-tile Hard IP Intel Stratix 10 Design Example User Guide: Ethernet, CPRI PHY, and Dynamic Reconfiguration*.
6. Enable the PMA[9].
7. Deassert the `i_sl_tx_rst_n/i_tx_rst_n/soft_tx_rst` signal.
8. If using a PMA configuration, load the PMA configuration using control status registers (CSR). This is loaded to the registers using PMA registers 0x200 to 0x203[11].
   a. Write 0x40143 = 0x80.
   b. Read 0x40144[0] until it changes to 1.
9. Enable internal serial loopback[12] and run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00.
10. Enable mission mode and disable internal serial loopback (skip this step if using internal serial loopback)[12].
11. Wait for valid data traffic on RX and then proceed to the next step.
12. Run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00 (skip this step if using internal serial loopback).
13. Run continuous adaptation[13].

---

(9) Refer to *0x0001: PMA Enable/Disable* section in the *E-tile Transceiver PHY User Guide*.

(10) Refer to *PMA Analog Reset* section in the *E-tile Transceiver PHY User Guide*.

(11) Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage* sections in the *E-tile Transceiver PHY User Guide*.

(12) For how to enable and disable internal serial loopback, refer to *0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control* section in the *E-tile Transceiver PHY User Guide*.

(13) During the continuous adaptation, the link partner must keep sending the data. If link goes down, the entire sequence must be repeated.

14. Deassert the `i_sl_rx_rst_n/i_rx_rst_n/soft_rx_rst` signal.
15. Optional: Verify that the link status signal `rx_aligned` transitions high.
16. Send packets.

Related Information

- E-Tile Hard IP for Ethernet Intel Agilex FPGA IP Design Example User Guide: Testing the 100G Ethernet Dynamic Reconfiguration Hardware Design Example
  More information about resetting sequence.
- E-Tile Transceiver PHY User Guide: Loading a PMA Configuration
  More information about loading a PMA configuration.
- E-Tile Transceiver PHY User Guide: PMA Registers 0x200 to 0x203 Usage
  More information about PMA registers usage.
- E-Tile Transceiver PHY User Guide: PMA Attribute Codes
  More information about RX phase slip usage.

2.9.9. Ethernet Adaptation Flow with PTP or with External AIB Clocking

This adaptation flow applies to single and multilane 10GE/25GE variants.

Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage* sections in the *E-tile Transceiver PHY User Guide* for more details on the adaptation flow and how to get started.

This adaptation flow assumes a valid Ethernet traffic.

1. Assert `i_sl_tx_rst_n/soft_tx_rst` and `i_sl_rx_rst_n/soft_rx_rst` signals.
2. Trigger PMA analog reset.[(14)]
3. Reload PMA settings (call the PMA attribute sequencer) using 0x91[0] = 1[(14)].
4. Deassert the `i_sl_tx_rst_n/soft_tx_rst` signal.
5. If using a PMA configuration, load the PMA configuration using control status registers (CSR). This is loaded to the registers using PMA registers 0x200 to 0x203[(15)].
   a. Write 0x40143 = 0x80.
   b. Read 0x40144[0] until it changes to 1.
6. Enable internal serial loopback[(16)] and run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00.
7. Enable mission mode and disable internal serial loopback (skip this step if using internal serial loopback)[(16)].

---

(14) Refer to *PMA Analog Reset* section in the *E-tile Transceiver PHY User Guide*.

(15) Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage* sections in the *E-tile Transceiver PHY User Guide*.

(16) For how to enable and disable internal serial loopback, refer to *0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control* section in the *E-tile Transceiver PHY User Guide*.

8. Wait for valid data traffic on RX and then proceed to the next step.
9. Run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00 (skip this step if using internal serial loopback).
10. Run continuous adaptation[17].
11. Deassert the `i_sl_rx_rst_n/soft_rx_rst` signal.
12. Optional: Verify that the link status signal `rx_aligned` transitions high.
13. Send packets.

Related Information

- E-Tile Transceiver PHY User Guide: Loading a PMA Configuration
  More information about loading a PMA configuration.
- E-Tile Transceiver PHY User Guide: PMA Registers 0x200 to 0x203 Usage
  More information about PMA registers usage.

2.9.10. Ethernet Adaptation Flow with Non-external AIB Clocking

This adaptation flow applies to single 10GE/25GE/100GE and multilane 10GE/25GE variants.

Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage* sections in the *E-tile Transceiver PHY User Guide* for more details on the adaptation flow and how to get started.

This adaptation flow assumes a valid Ethernet traffic.

1. Assert `i_sl_tx_rst_n/i_tx_rst_n/soft_tx_rst` and `i_sl_rx_rst_n/i_rx_rst_n/soft_rx_rst` signals.
2. Trigger PMA analog reset.[18][19]
3. Reload PMA settings (call the PMA attribute sequencer) on all lanes.
4. Apply control status registers (CSR) reset.
   a. For 100GE/25GE/10GE single instance, cycle control status registers (CSR) reset.
   b. For 25GE/10GE multilane instance, hold CSR reset on slave channels, cycle master channel CSR reset, then release CSR reset on slave channels as indicated in the below figure.

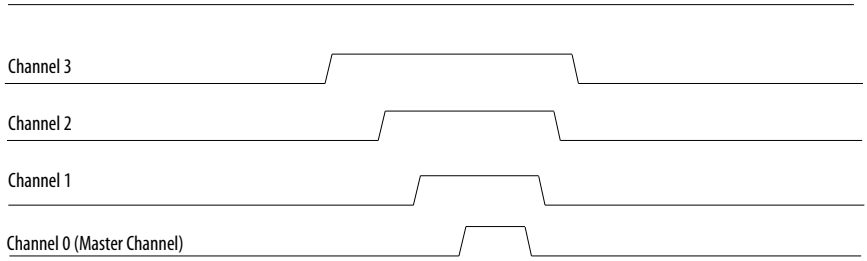
---

[17] During the continuous adaptation, the link partner must keep sending the data. If link goes down, the entire sequence must be repeated.

[18] Refer to *PMA Analog Reset* section in the *E-tile Transceiver PHY User Guide*.

[19] This step also resets the transmit side. For scenerios where the transmit side should left untouched, the PMA analog reset should never be asserted. For such scenerios, refer to the Signal Detect Algorithm section in the *E-tile Transceiver PHY User Guide*.

Figure 32. Reset Sequence in 10G/25G Multilane Mode



5. Deassert the `i_sl_tx_rst_n/i_tx_rst_n/soft_tx_rst` signal.
6. If using a PMA configuration, load the PMA configuration using control status registers (CSR). This is loaded to the registers using PMA registers 0x200 to 0x203[20].
   a. Write 0x40143 = 0x80.
   b. Read 0x40144[0] until it changes to 1.
7. Enable internal serial loopback[21] and run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00.
8. Enable mission mode and disable internal serial loopback (skip this step if using internal serial loopback)[21].
9. Wait for valid data traffic on RX and then proceed to the next step.
10. Run initial adaptation. Verify that the initial adaptation status is complete using interrupt code 0x0126 and data 0x0B00 (skip this step if using internal serial loopback).
11. Run continuous adaptation[22].
12. Deassert the `i_sl_rx_rst_n/i_rx_rst_n/soft_rx_rst` signal.
13. Optional: Verify that the link status signal `rx_aligned` transitions high.
14. Send packets.

Related Information

- E-Tile Transceiver PHY User Guide: Loading a PMA Configuration
  More information about loading a PMA configuration.
- E-Tile Transceiver PHY User Guide: PMA Registers 0x200 to 0x203 Usage
  More information about PMA registers usage.

---

[20] Refer to *Loading a PMA Configuration* and *PMA Registers 0x200 to 0x203 Usage* sections in the *E-tile Transceiver PHY User Guide*.

[21] For how to enable and disable internal serial loopback, refer to *0x0008: Internal Serial Loopback and Reverse Parallel Loopback Control* section in the *E-tile Transceiver PHY User Guide*.

[22] During the continuous adaptation, the link partner must keep sending the data. If link goes down, the entire sequence must be repeated.

2.10. Reset

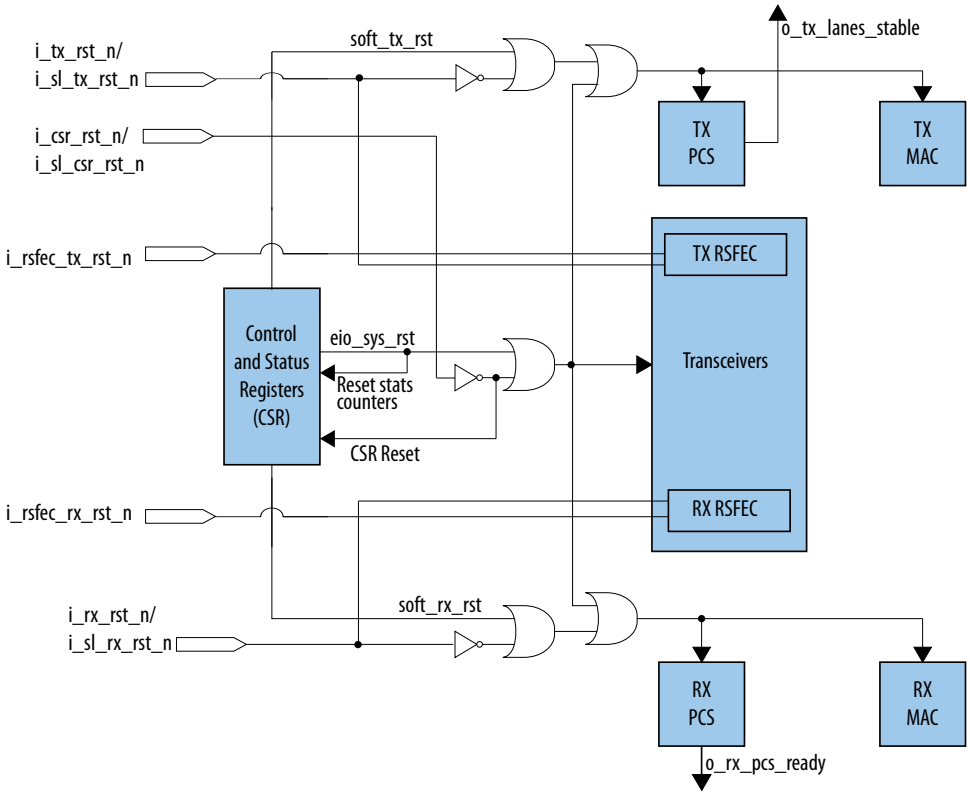
Ethernet registers control three distinct soft resets:

- `eio_sys_rst`
- `soft_tx_rst`
- `soft_rx_rst`

These soft resets are not self-clearing. The reconfig port clears the soft resets by writing to the appropriate register. The IP core also has three hard reset signals, which are active low:

- `i_csr_rst_n` (100G) / `i_sl_csr_rst_n` (10G/25G)
- `i_tx_rst_n` (100G) / `i_sl_tx_rst_n` (10G/25G)
- `i_rx_rst_n` (100G) / `i_sl_rx_rst_n` (10G/25G)

Figure 33. Conceptual Overview of General IP Core Reset Logic



Asserting the external hard reset `i_csr_rst_n`/`i_sl_csr_rst_n` or the soft reset `eio_sys_rst` returns all Ethernet registers to their original values, including the statistics counters. An additional dedicated reset signal, `i_reconfig_reset`, resets the transceiver reconfiguration, Ethernet reconfiguration interfaces, and some Ethernet soft registers.

Table 26. Reset Signal Functions

In this table, a tick (√) represents the block is reset by the specified reset signal. A dash (—) represents the block is not impacted by the specified reset signal.

Reset Signal	Block											
	TX EMIB Interface	TX MAC	TX PCS	TX FEC	TX PMA Interface	TX Statistics	RX EMIB Interface	RX MAC	RX PCS	RX FEC	RX PMA Interface	RX Statistics
i_sl_csr_rst_n	√	√	√	—	√	√	√	√	√	—	√	√
i_csr_rst_n eio_sys_rst	√	√	√	√	√	√	√	√	√	√	√	√
i_sl_tx_rst_n soft_tx_rst[23]	—	√	√	—	—	—	—	—	—	—	—	—
i_tx_rst_n soft_tx_rst[24]	—	√	√	√	—	—	—	—	—	—	—	—
i_sl_rx_rst_n soft_rx_rst[23]	—	—	—	—	—	—	—	√	√	—	—	—
i_rx_rst_n soft_rx_rst[24]	—	—	—	—	—	—	—	√	√	√	—	—
soft_clear_tx_stats	—	—	—	—	—	√	—	—	—	—	—	—
soft_clear_rx_stats	—	—	—	—	—	—	—	—	—	—	—	√

[23] This reset is applicable for the 10G/25G Ethernet variant.

[24] This reset is applicable for the 100G Ethernet variant.

The general reset signals reset the following functions:

- `soft_tx_rst`, `i_tx_rst_n/i_sl_tx_rst_n`:
  - Resets the IP core in the TX direction.
  - Resets TX PCS, TX MAC, and TX PMA interface.
  - This reset leads to deassertion of the `o_tx_lanes_stable` output signal.
  - In Ethernet Toolkit, perform this reset using the TX MAC and PCS Reset.
- `soft_rx_rst`, `i_rx_rst_n/i_sl_rx_rst_n`:
  - Resets the IP core in the RX direction.
  - Resets RX PCS, and RX MAC.
  - This reset leads to deassertion of the `o_rx_pcs_ready` output signal.
  - In Ethernet Toolkit, perform this reset using the RX MAC and PCS Reset.
- `eio_sys_rst`, `i_csr_rst_n/i_sl_csr_rst_n`:
  - Resets the IP core. `i_csr_rst_n` signal is edge sensitive. Perform the reset assertion and deassertion sequence at the `i_csr_rst_n` 0->1 edge.
  - Resets the TX and RX MAC, TX and RX EMIB interface, Ethernet reconfiguration registers, PCS, and TX and RX PMA interfaces.
  - This reset leads to deassertion of the `o_tx_lanes_stable` and `o_rx_pcs_ready` output signals.
  - In Ethernet Toolkit, perform this reset using the Full System Reset.

In addition, the synchronous `i_reconfig_reset` signal resets the IP core transceiver reconfiguration interface, Ethernet reconfiguration interfaces, and some Ethernet soft registers. `i_reconfig_reset` signal is synchronous to the `i_reconfig_clk` and is positive edge triggered.

PMA reset is only required when you change any PMA settings. For PMA reset information, refer to *PMA Reset* and *PMA Analog Reset* in the *E-Tile Transceiver PHY User Guide*.

System Considerations

- You should perform a system reset before beginning IP core operation, preferably by asserting and deasserting the `i_csr_rst_n/i_sl_csr_rst_n` and `i_reconfig_reset` signals together. Alternatively, you can use `eio_sys_rst` [(25)] register instead of `i_csr_rst_n/i_sl_csr_rst_n` signals.
  - To assert `i_csr_rst_n/i_sl_csr_rst_n`, drive the signal to 0. To deassert `i_csr_rst_n/i_sl_csr_rst_n`, drive the signal to 1.
  - To assert `i_reconfig_reset`, drive the signal to 1. To deassert `i_reconfig_reset`, drive the signal to 0.
  - To assert `eio_sys_rst`, write 1'b1 to the `0x310[0]` register. To deassert `eio_sys_rst`, write 1'b0 to the `0x310[0]` register. The IP core implements the correct reset sequence to reset the entire IP core.
  - For 10GE/25GE multi-channel master-slave configuration (with RS-FEC enabled), all channels must be taken out of the initial reset before each channel can start functioning independently. For more information, refer to Figure 32 on page 114.
  - For 10GE/25GE multi-channel non master-slave configuration (with RS-FEC disabled), the master/slave resets can be asserted/deasserted in any order for each of the individual channels.
- If you assert the transmit reset when the downstream receiver is already aligned, the receiver loses alignment. Before the downstream receiver loses lock, it might receive some malformed frames.
- If you assert the receive reset while the upstream transmitter is sending packets, the packets in transit are corrupted.

Related Information

- Reset Signals on page 157
- Register Descriptions on page 183
- PMA Reset
  More information about resetting PMA channels.
- PMA Analog Reset
  More information about resetting PMA internal controller.
- Ethernet Toolkit User Guide

2.10.1. Reset Sequence

The following waveforms shows the reset sequence using the `i_csr_rst_n` (100G)/ `i_sl_csr_rst_n` (10G/25G), `i_tx_rst_n` (100G)/`i_sl_tx_rst_n` (10G/25G), and `i_rx_rst_n` (100G)/`i_sl_rx_rst_n` (10G/25G) signals.

(25) When the `eio_sys_rst` signal is asserted, the access to any other registers is forbidden. When AN/LT is enabled, `eio_sys_rst` reset cannot be asserted until AN/LT is complete and enters the data mode.

Figure 34. External Hard Reset Sequence

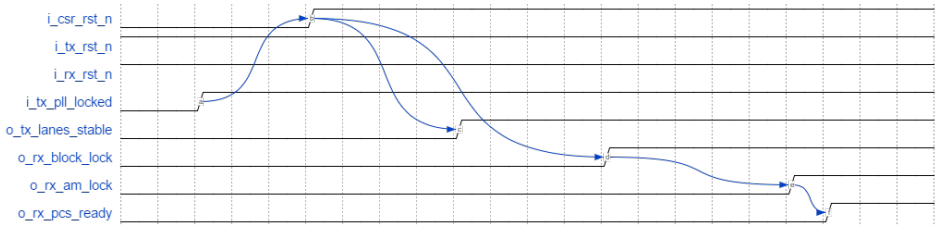


Figure 35. TX Datapath Reset Sequence

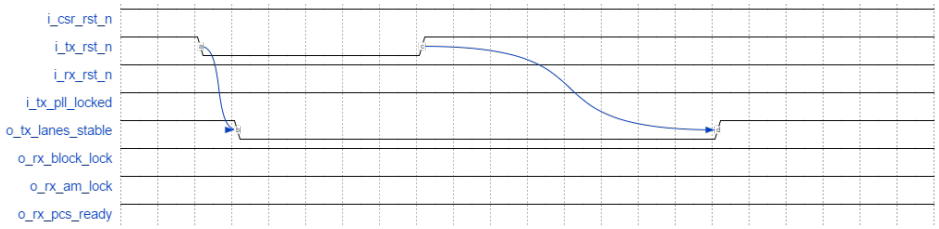
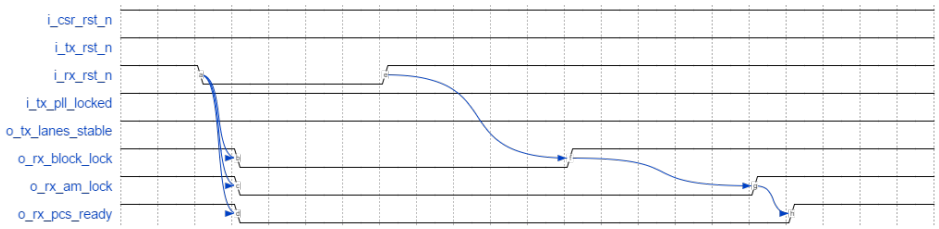


Figure 36. RX Datapath Reset Sequence



2.10.1.1. Reset Sequence with External AIB Clocking

Below table shows the reset recommendation when the external AIB signal is used to reset the E-Tile Hard IP for Ethernet Intel FPGA IP data channels.

For details on general reset signals used during the reset, refer to *Reset* and *Reset Signals* sections.

Table 27. External AIB Clocking Reset Signal Functions

This table is a reset sequence recommendation when external AIB clock is enabled. Signals marked by a tick (√) must be reset in the specified mode. Signals marked by a dash (—) don't required reset in the specified mode.

Modes	Signals		
	i_csr_rst_n	i_tx_rst_n	i_rx_rst_n
External AIB clock enable — Master Channel	—[26]	√	√
External AIB clock enable —	—[26]	√	√

continued...

[26] If the External AIB clock is enabled, there is no need to assert i_csr_rst_n reset. If you assert this reset on the master channel after power on, it will bring down slave channels.

Modes	Signals		
	i_csr_rst_n	i_tx_rst_n	i_rx_rst_n
Slave Channel			
External AIB clock disable — Master Channel	√[27]	√	√
External AIB clock disable — Slave Channel	√[27]	√	√

Use case example with 10G Master Ethernet channel and three 25G Slave Ethernet channels is shown in the *Master-Slave Configuration: Option 3- Dynamic Reconfiguration* clock network use case section.

For more information on PMA Analog Reset user cases, refer to the *E-tile Transceiver PHY User Guide*.

For more information on the dynamic reconfiguration, refer to the *Dynamic Reconfiguration Design Example User Guide*.

Related Information

- Master-Slave Configuration: Option 3 - Dynamic Reconfiguration on page 169
- Reset on page 115
- Reset Signals on page 157
- E-Tile Transceiver PHY User Guide
  Information about the Native PHY IP Core.
- E-Tile Hard IP Intel Stratix 10 Design Example User Guide
  Information about the Dynamic Reconfiguration.
- E-Tile Hard IP Intel Agilex Design Example User Guide
  Information about the Dynamic Reconfiguration.

2.11. Interfaces and Signals

All input signal names begin with `i_` and all output signal names begin with `o_`.

[27] This case has a limitation. `i_csr_rst_n` reset on the master channel will bring down the slave channel.

2.11.1. TX MAC Interface to User Logic

The E-Tile Hard IP for Ethernet Intel FPGA IP TX client interface in MAC+PCS variations employs the Avalon-ST protocol.(1) The Avalon-ST protocol is a synchronous point-to-point, unidirectional interface that connects the producer of a data stream (source) to a consumer of data (sink). The key properties of this interface include:

- Start of packet (SOP) and end of packet (EOP) signals delimit frame transfers.
- The SOP must always be in the MSB, simplifying the interpretation and processing of incoming data.
- A valid signal qualifies signals from source to sink.
- The sink applies backpressure to the source by using the ready signal. The source typically responds to the deassertion of the ready signal from the sink by driving the same data until the sink can accept it. The **Ready latency** defines the relationship between assertion and deassertion of the ready signal, and cycles which are considered to be `ready` for data transfer.

The client acts as a source and the TX MAC acts as a sink in the transmit direction.

Table 28. Signals of the AvalonStreaming TX Client Interface

All interface signals are clocked by the TX clock. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `i_sl_tx_data`
- For variants with more than 1 channel: `i_sl_tx_data[n-1:0]`
- For variants with single 100GE channel: `i_tx_data`

Signal Name	Width	Description

`i_sl_clk_tx`
`i_sl_clk_tx[n-1:0]`
`i_clk_tx`
	1 bit for each channel	The TX clock for the IP core that drives the channel.

`i_sl_tx_data`
`i_sl_tx_data[n-1:0]`
`i_tx_data`
	64 bits (10G/25G)
	512 bits (100G)
		TX data.
		The E-Tile Hard IP for Ethernet Intel FPGA IP does not process incoming packets of less than nine bytes. You must ensure such frames do not reach the TX client interface. The IP core marks incoming packets of 9 to 13 bytes as errored, by adding Error Control bytes to the packet upon transmission.
		You must send each TX data packet without intermediate IDLE cycles. Therefore, you must ensure your application can provide the data for a single packet in consecutive clock cycles. If data might not be available otherwise, you must buffer the data in your design and wait to assert the SOP signal when you are assured the packet data to send on the TX data is available or will be available on time.

`i_sl_tx_valid`
`i_sl_tx_valid[n-1:0]`
`i_tx_valid`
	1 bit for each channel	When asserted, the TX data signal is valid. This signal must be continuously asserted between the assertions of the start of packet and end of packet signals for the same packet.

`i_sl_tx_empty`
`i_sl_tx_empty[n-1:0]`
`i_tx_empty`
	3 bits for each channel (10G/25G)
	6 bits (100G)
		Indicates the number of empty bytes on the TX data when the EOP signal is asserted.

continued...

Signal Name	Width	Description
i_sl_tx_startofpacket i_sl_tx_startofpacket[n-1:0] i_tx_startofpacket	1 bit for each channel	When asserted, indicates that the TX data holds the first clock cycle of data in a packet (start of packet). Assert for only a single clock cycle for each packet. When the SOP signal is asserted, the MSB of the TX data drives the start of packet.
i_sl_tx_endofpacket i_sl_tx_endofpacket[n-1:0] i_tx_endofpacket	1 bit for each channel	When asserted, indicates that the TX data holds the final clock cycle of data in a packet (end of packet). Assert for only a single clock cycle for each packet. For some legitimate packets, the SOP and EOP signals are asserted on the same clock cycle.
o_sl_tx_ready o_sl_tx_ready[n-1:0] o_tx_ready	1 bit for each channel	When asserted, indicates that the MAC can accept the data **Ready latency** clock cycles after the current cycle. The IP core asserts the ready signal on clock cycle <*n*> to indicate that clock cycle <*n* + **Ready latency**> is a ready cycle. The client may only transfer data during ready cycles. If the IP core deasserts the ready during a packet transfer on the TX MAC client interface, the client must stall the data on the TX data. The ready signal indicates the MAC is ready to receive data in normal operational mode. However, the ready signal might not be an adequate indication following reset. To avoid sending packets before the Ethernet link is able to transmit them reliably, you should ensure that the application does not send packets on the TX client interface until after the `o_tx_lanes_stable` signal is asserted.
i_sl_tx_error i_sl_tx_error[n-1:0] i_tx_error	1 bit for each channel	When asserted in an EOP cycle (while the EOP signal is asserted), directs the IP core to insert an error in the packet before sending it on the Ethernet link. This signal supports the client in selectively invalidating a packet. It is also a test and debug feature. In loopback mode, the IP core recognizes the packet upon return as a malformed packet.
i_sl_tx_pause i_sl_tx_pause[n-1:0] i_tx_pause	1 bit for each channel	When asserted, directs the IP core to send a PAUSE XOFF frame on the Ethernet link. The rising edge triggers the request. You must maintain this signal at the value of 1 until you wish the IP core to end the PAUSE period. The IP core sends a PAUSE XOFF frame after it completes processing of the current in-flight TX packet, and periodically thereafter, until you deassert the `i_tx_pause` signal. When you deassert the `i_tx_pause` signal, the IP core sends a PAUSE XON frame on the Ethernet link. *Note:* For 10G/25G channels, you should hold the `i_sl_tx_pause` signal more than 205 ns to get the request captured by the MAC. This signal is functional only if standard Ethernet flow control is enabled.

continued...

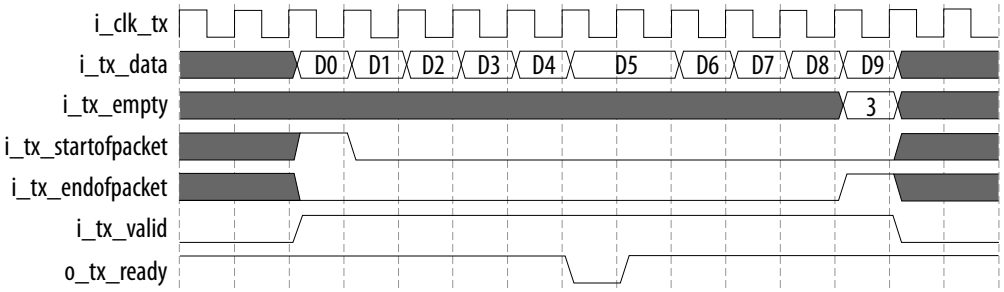
Signal Name	Width	Description
		Note: Standard Ethernet flow control is enabled if the value of the RTL parameter `flow_control` is one of `sfc`, `sfc_no_xoff`, `both`, or `both_no_xoff`. If you do not specify the value of the RTL parameter in your IP core instance, but you generate the IP core variation with the value of the **Stop TX traffic when link partner sends pause** set to **Yes** or **No**, pause flow control is also enabled.

`i_sl_tx_pfc`
`i_sl_tx_pfc[n-1:0`
`i_tx_pfc`
	8 bits for each channel	When a bit is asserted, directs the IP core to send a PFC XOFF frame on the Ethernet link for the corresponding priority queue. The rising edge triggers the request. You must maintain this signal at the value of 1 until you wish the IP core to end the pause period. The IP core sends a PFC XOFF frame after it completes processing of the current in-flight TX packet, and periodically thereafter, until you deassert the `i_tx_pfc` bit. When you deassert the bit, the IP core sends a PFC XON frame on the Ethernet link for the corresponding priority queue.
		Note: For 10G/25G channels, you should hold the `i_sl_tx_pfc` signal more than 205 ns to get the request captured by the MAC.
		This signal is functional only if priority flow control is enabled.
		Note: Priority flow control is enabled if the value of the RTL parameter `flow_control` is one of `pfc`, `pfc_no_xoff`, `both`, or `both_no_xoff`. If you do not specify the value of the RTL parameter in your IP core instance, but you generate the IP core variation with the value of the **Stop TX traffic when link partner sends pause** set to **Yes** or **No**, priority flow control is also enabled.

`i_sl_tx_skip_crc`
`i_sl_tx_skip_crc[n-1:0]`
`i_tx_skip_crc`
	1 bit for each channel	Specifies how the TX MAC should process the current TX MAC client interface packet. Use this signal to temporarily turn off source insertion for a specific packet and to override the default behaviors of padding to minimum packet size and inserting CRC.
		If this signal is asserted, directs the TX MAC to not insert CRC, not add padding bytes, and not implement source address insertion. You can use this signal to indicate the data on the TX data signal includes CRC, padding bytes (if relevant), and the correct source address.
		If this signal is not asserted, and source address insertion is enabled, directs the TX MAC to overwrite the source address. The MAC copies the new source address from the TXMAC_SADDR register.
		If this signal is not asserted, whether or not source address insertion is enabled, the TX MAC inserts padding bytes if needed and inserts CRC in the packet.
		The client must maintain the same value on this signal for the duration of the packet (from the cycle in which it asserts the SOP signal through the cycle in which it asserts the EOP signal, inclusive).

Figure 37. Transmitting Data Using the TX Avalon Streaming MAC Client Interface

Note: The transmit operations for the single lane ports (i_sl_*) are equivalent to the multi-lane ports.



The figure above shows how to transmit data using the TX MAC client interface. The interface complies with the Avalon streaming interface specification.

- Data valid (`i_sl_tx_valid`) must be held high from the start to end of a packet, and must be low outside of a packet.
- Packets always start on the leftmost of the byte of `i_tx_data` (SOP aligned).
- You can set the **Ready latency** through the parameter editor.
  - When `o_tx_ready` deasserts, `i_tx_data` must be paused for as many cycles as `o_tx_ready` is deasserted, starting **Ready latency** cycles later. In this example, **Ready latency** is 1. So the cycle after `o_tx_ready` deasserts for 1 cycle, `i_tx_data` is paused for 1 cycle.
- When the frame ends, `i_tx_empty` is set to the number of unused bytes in `i_tx_data`, starting from the right (byte 0).
  - In this example, `i_tx_data` on the last cycle of the packet has 3 empty bytes.
  - The minimum number of bytes on the last cycle is 1.

Figure 38. Fields and Frame Boundaries in an Ethernet Packet

When you turn off **Preamble Passthrough** in the parameter editor, `i_tx_data` must be written as shown below for the first cycle of data presented to the MAC.

Note: For 10G/25G channels, multiple cycles are required to write the header data.

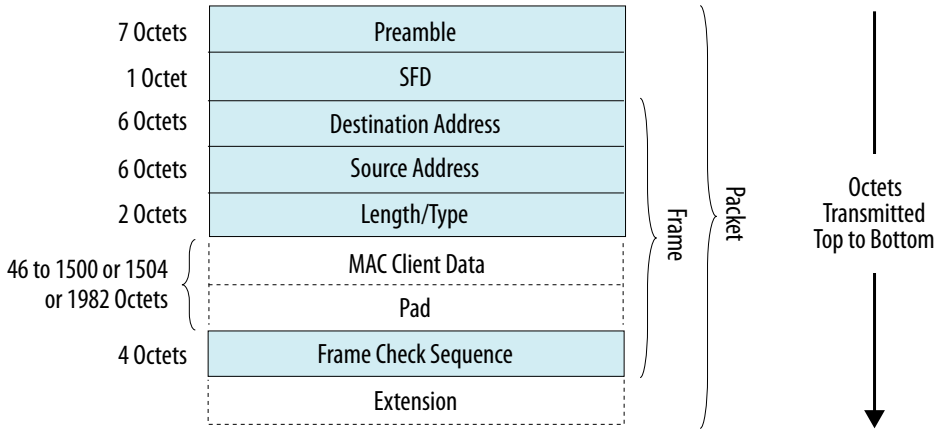


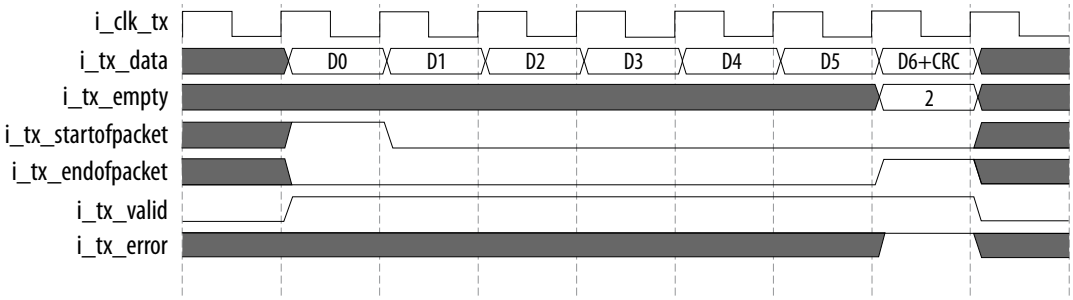
Table 29. TX MAC Field Positions in i_tx_data with Preamble Passthrough Disabled

10G/25G requires multiple transfer cycle for header data. The (') symbol in the **10G/25G i_sl_tx_data** column represents transfer on the subsequent cycle.

100G i_tx_data	10G/25G i_sl_tx_data	MAC Field	Note
[511:504]	[63:56]	Dest Addr[47:40]	The first octet of the Destination Address, follows Start Frame Delimiter (SFD).
[503:496]	[55:48]	Dest Addr[39:32]	
[495:488]	[47:40]	Dest Addr[31:24]	
[495:480]	[39:32]	Dest Addr[23:16]	
[479:472]	[31:24]	Dest Addr[15:8]	
[471:464]	[23:16]	Dest Addr[7:0]	
[463:456]	[15:8]	Src Addr[47:40]	When you turn on **Source Address Insertion**, contents are replaced by **txmac_saddr** unless `i_tx_skip_crc` is high.
[455:448]	[7:0]	Src Addr[39:32]	
[447:440]	[63:56]'	Src Addr[31:24]	
[439:432]	[55:48]'	Src Addr[23:16]	
[431:424]	[47:40]'	Src Addr[15:8]	
[423:416]	[39:32]'	Src Addr[7:0]	
[415:408]	[31:24]'	Length/Type[15:0]	
[407:400]	[23:16]'	Length/Type[7:0]	
[399:0]	[15:0]'	...	

The `i_tx_error` or `i_sl_tx_error` port allows packets to be marked as errored when they are complete.

Figure 39. Using i_tx_error



Because the core uses a cut-through interface, the core starts transmitting the packet data it is given as soon as possible. If the core discovers an error after the packet starts, e.g. in a bridging system where the receiver also uses a cut-through interface, you can use `i_tx_error` to invalidate the packet. You can also use `i_tx_error` for testing, to generate errored packets, and confirm that the other end of the link is able to reject the errored packets.

To invalidate an errored frame, end it with `i_tx_endofpacket` and assert `i_tx_error`. If the frame is good, deassert `i_tx_error`.

Note: Using `i_tx_error` will not provide a robust test of the remote CRC, because it uses MII Error Control bytes to indicate error, rather than relying on corrupted CRC bits.

Related Information

Avalon Interface Specifications

2.11.2. RX MAC Interface to User Logic

The E-Tile Hard IP for Ethernet Intel FPGA IP RX client interface in MAC+PCS variations employs the Avalon streaming interface protocol. The Avalon streaming interface protocol is a synchronous point-to-point, unidirectional interface that connects the producer of a data stream (source) to a consumer of data (sink). The key properties of this interface include:

- Start of packet (SOP) and end of packet (EOP) signals delimit frame transfers.
- The SOP must always be in the MSB, simplifying the interpretation and processing of data you receive on this interface.
- A valid signal qualifies signals from source to sink.

The RX MAC acts as a source and the client acts as a sink in the receive direction.

Table 30. Signals of the Avalon Streaming RX Client Interface

All interface signals are clocked by the RX clock. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `i_sl_rx_data`
- For variants with more than 1 channel: `i_sl_rx_data[n-1:0]`
- For variants with single 100GE channel:`i_rx_data`

Name	Width	Description

`i_sl_clk_rx`
`i_sl_clk_rx[n-1:0]`
`i_clk_rx`
	1 bit for each channel	The RX clock for the IP core that drives the channel.

`o_sl_rx_data`
`o_sl_rx_data[n-1:0]`
`o_rx_data`
	64 bits for each channel (10G/25G)
	512 bits (100G)
		RX data. The highest order bit is the MSB and bit 0 is the LSB. Bytes are read in the usual left to right order. The IP core reverses the byte order to meet the requirements of the Ethernet standard.

`o_sl_rx_valid`
`o_sl_rx_valid[n-1:0]`
`o_rx_valid`
	1 bit for each channel	When asserted, indicates that RX data is valid. Only valid between the SOP and EOP signals.
		This signal might be deasserted between the assertion of the SOP and EOP signals.

`o_sl_rx_empty`
`o_sl_rx_empty[n-1:0]`
`o_rx_empty`
	3 bits for each channel (10G/25G)
	6 bits for each channel (100G)
		Indicates the number of empty bytes on the RX data signal when EOP signal is asserted, starting from the least significant byte (LSB).

`o_sl_rx_startofpacket`
`o_sl_rx_startofpacket[n-1:0]`
`o_rx_startofpacket`
	1 bit for each channel	When asserted, indicates that the RX data signal holds the first clock cycle of data in a packet (start of packet). The IP core asserts this signal for only a single clock cycle for each packet.
		When the SOP signal is asserted, the MSB of the RX data signal drives the start of packet.

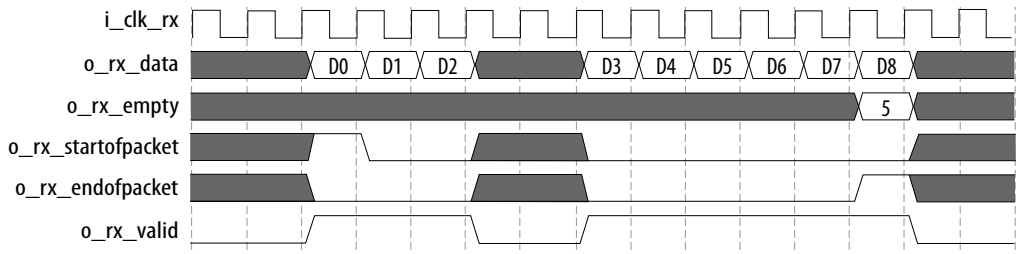
continued...

Name	Width	Description
o_sl_rx_endofpacket o_sl_rx_endofpacket[n-1:0] o_rx_endofpacket	1 bit for each channel	When asserted, indicates that the RX data signal holds the final clock cycle of data in a packet (end of packet). The IP core asserts this signal for only a single clock cycle for each packet. In the case of an undersized frame or in the case of a frame that is exactly 64 bytes long, the SOP and EOP signals might be asserted in the same clock cycle.
o_sl_rx_error o_sl_rx_error[n-1:0] o_rx_error	6 bits for each channel	Reports certain types of errors in the Ethernet frame whose contents are currently being transmitted on the client interface. This signal is valid in EOP cycles only. The individual bits report different types of errors: • Bit [0]: Malformed packet error. If this bit has the value of 1, the packet is malformed. The IP core identifies a malformed packet when it receives a control character that is not a terminate character. • Bit [1]: CRC error. If this bit has the value of 1, the IP core detected a CRC error or an Error character in the frame. • Bit [2]: undersized frame. If this bit has the value of 1, the frame size is between nine and 63 bytes, inclusive. In this case the IP core also sets o_rx_error[1] to signal a CRC error. The IP core does not recognize an incoming frame of size eight bytes or less as a frame, and those cases are not reported here. If the preamble-passthrough and CRC forwarding settings cause the RX MAC to strip out bytes such that only eight bytes or less remain in the frame, the IP core also does not recognize the frame, and those cases are not reported here. If the frame is malformed, the case is not reported here. • Bit [3]: oversized frame. If this bit has the value of 1, the frame size is greater than the maximum frame size you specified as the value of the parameter editor **RX Maximum Frame Size** parameter or overwrote with the rx_max_frame_size RTL parameter. If the frame is malformed, the case is not reported here. • Bit [4]: payload length error. If this bit has the value of 1, the payload received in the frame is shorter than the length field value, and the value in the length field is less than or equal 1500 bytes. If the frame is oversized or undersized, the case is not reported here. If the frame is malformed, the case is not reported here. • Bit [5]: Reserved.
o_sl_rxstatus_valid o_sl_rxstatus_valid[n-1:0] o_rxstatus_valid	1 bit for each channel	When asserted, indicates that o_rxstatus_data is driving valid data.
o_sl_rxstatus_data o_sl_rxstatus_data[n-1:0] o_rxstatus_data	40 bits for each channel	Specifies information about the received frame. The following fields are defined: • [Bit 39]: When asserted, indicates a PFC frame • [Bits 38:36]: Reserved • Bit[35]: When asserted, indicates a PAUSE frame • Bit[34]: When asserted, indicates a Control (Type is 0x8808) frame

continued...

Name	Width	Description
		• Bit[33]: When asserted, indicates a VLAN frame • Bit[32]: When asserted, indicates a stacked VLAN frame • Bits[31:0]: Reserved
o_sl_rx_pause o_sl_rx_pause[n-1:0] o_rx_pause	1 bit for each channel	When asserted, indicates the IP core received a PAUSE XOFF frame on the Ethernet link. The IP core deasserts this signal when the quanta count from the PAUSE XOFF request expires. If you set the parameter editor **Stop TX traffic when link partner sends pause** parameter to the value of **Yes**, or overwrite it with the sfc or both value for the flow_control RTL parameter, the TX MAC stops traffic in response to the PAUSE XOFF frame. In this case, the quanta count decrements while the IP core stops traffic. If the settings direct the TX MAC to not stop traffic in response to the PAUSE XOFF frame, the quanta counter decrements on every valid cycle on the TX MAC client interface. Each quanta represents 512 bits. Therefore, the counter decrements by one half in every valid clock cycle in 100G variations.
o_sl_rx_pfc o_sl_rx_pfc[n-1:0] o_rx_pfc	8 bits for each channel	When a bit is asserted, indicates the IP core received a PFC XOFF frame on the Ethernet link for the corresponding priority queue. The IP core deasserts each bit when the XOFF frame's quanta count expires. The PFC quanta counters decrement on every valid cycle on the TX MAC client interface. Each quanta represents 512 bits. Therefore, the counter decrements by one half in every valid clock cycle in 100G variations. In summary, the width of the pulse indicates the length of the requested pause in traffic for the queue.

Figure 40. Receiving Data Using the RX MAC Client Interface



The figure above shows how to receive data using the RX MAC client interface. The interface complies with the Avalon streaming interface specification.

- Packets always start on the leftmost of the byte of o_rx_data (SOP aligned).
- When the frame ends, o_rx_empty is set to the number of unused bytes in o_rx_data, starting from the right (byte 0).
  - In this example, o_rx_data on the last cycle of the packet has 5 empty bytes.
  - The minimum number of bytes on the last cycle is 1.
- The framing and data ports are only valid when o_rx_data is high.

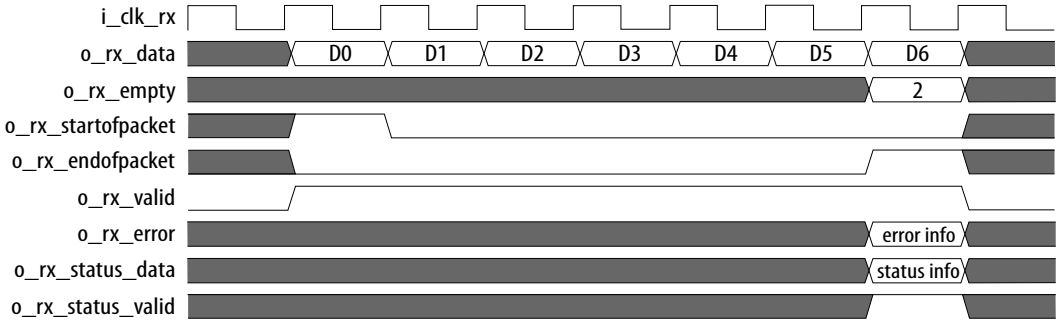
Note: The interface does not take direct backpressure

Table 31. RX MAC Field Positions in o_rx_data with Preamble Passthrough Disabled

100G i_tx_data	10G/25G i_sl_tx_data	MAC Field	Note
[511:504]	[63:56]’	Dest Addr[47:40]	The first octet of the Destination Address, follows Start Frame Delimiter (SFD).
[503:496]	[55:48]’	Dest Addr[39:32]	
[495:488]	[47:40]’	Dest Addr[31:24]	
[495:480]	[39:32]’	Dest Addr[23:16]	
[479:472]	[31:24]’	Dest Addr[15:8]	
[471:464]	[23:16]’	Dest Addr[7:0]	
[463:456]	[15:8]’	Src Addr[47:40]	
[455:448]	[7:0]’	Src Addr[39:32]	
[447:440]	[63:56]	Src Addr[31:24]	
[439:432]	[55:48]	Src Addr[23:16]	
[431:424]	[47:40]	Src Addr[15:8]	
[423:416]	[39:32]	Src Addr[7:0]	
[415:408]	[31:24]	Length/Type[15:0]	
[407:400]	[23:16]	Length/Type[7:0]	
[399:0]	[15:0]	…	

Figure 41. RX MAC Status and Errors

The figure shows how status and error information are presented with RX packets as they arrive.



The status valid port is provided for backward compatibility, but always asserts when `o_rx_endofpacket` is asserted and valid.

Related Information

Avalon Interface Specifications

2.11.3. TX PCS Interface to User Logic

The E-Tile Hard IP for Ethernet Intel FPGA IP TX client interface in PCS Only variations employs the Media Independent Interface (MII) protocol.

The client acts as a source and the TX PCS acts as a sink in the transmit direction.

Table 32. Signals of the MII TX Client Interface

All interface signals are clocked by the TX clock. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `i_sl_tx_data`
- For variants with more than 1 channel: `i_sl_tx_data[n-1:0]`
- For variants with single 100GE channel: `i_tx_data`

Signal Name	Width	Description

`i_sl_clk_tx`
`i_sl_clk_tx[n-1:0]`
`i_clk_tx`
	1 bit for each channel	The TX clock for the IP core that drives the channel.

`i_sl_tx_mii_d`
`i_sl_tx_mii_d[n-1:0]`
`i_tx_mii_d`
	64 bits for each channel (10G/25G)
	256 bits(100G)
		TX MII data. Data must be in MII encoding. `i_tx_mii_d[7:0]` holds the first byte the IP core transmits on the Ethernet link. `i_tx_mii_d[0]` holds the first bit the IP core transmits on the Ethernet link.
		While the TX MII valid signal has the value of 0 or the alignment marker insertion bit signal has the value of 1, and for one additional clock cycle, you must hold the value of this signal stable. We refer to this behavior as freezing the signal value.

`i_sl_tx_mii_c`
`i_sl_tx_mii_c[n-1:0]`
`i_tx_mii_c`
	8 bits for each channel (10G/25G)
	32 bits (100G)
		TX MII control bits. Each bit corresponds to a byte of the TX MII data signal. For example, `i_tx_mii_c[0]` corresponds to `i_tx_mii_d[7:0]`, `i_tx_mii_c[1]` corresponds to `i_tx_mii_d[15:8]`, and so on.
		If the value of a bit is 1, the corresponding data byte is a control byte. If the value of a bit is 0, the corresponding data byte is data.
		The Start of Packet byte (0xFB), End of Packet byte (0xFD), Idle bytes (0x07), and error byte (0xFE) are control bytes, but the preamble bytes, Start of Frame (SFD) byte (0xD5), CRC bytes, and payload bytes are data bytes.
		While the TX MII valid signal has the value of 0 or the alignment marker insertion bit signal has the value of 1, you must freeze the value of this signal.

`i_sl_tx_mii_valid`
`i_sl_tx_mii_valid[n-1:0]`
`i_tx_mii_valid`
	1 bit for each channel	Indicates that the TX MII data signal is valid.
		You must assert this signal a fixed number of clock cycles after the IP core raises ready signal, and must deassert this signal the same number of clock cycles after the IP core deasserts the ready signal. The number must be in the range of 1–10 clock cycles.
		While you hold the value of this signal at 0, you must freeze the values of both TX MII data and TX MII control bits signals stable.

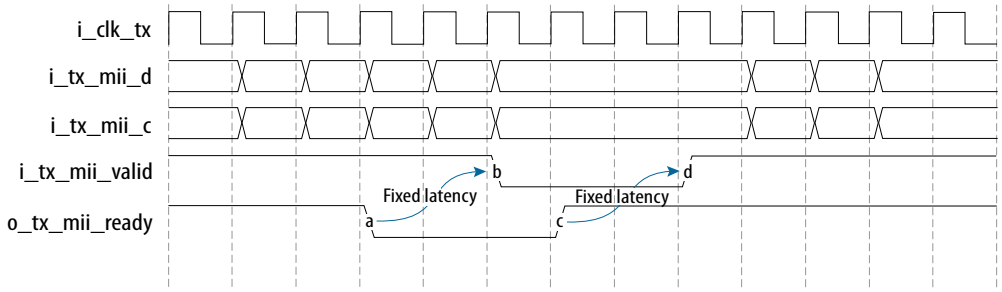
`o_sl_tx_mii_ready`
`o_sl_tx_mii_ready[n-1:0]`
`o_tx_mii_ready`
	1 bit for each channel	Indicates the PCS is ready to receive new data.

`i_sl_tx_mii_am`
`i_sl_tx_mii_am[n-1:0]`
`i_tx_mii_am`
	1 bit for each channel	Alignment marker insertion bit.
		• In 100G variations, you must hold this signal asserted for 5 consecutive clock cycles.
		• In 25Gx1 with RS-FEC variations, you must hold this signal asserted for 4 consecutive clock cycles.
		• In 10Gx1 or 25Gx1 without RS-FEC variations, you must tie this signal low.

continued...

Signal Name	Width	Description
		The number of valid clock cycles from deassertion of the alignment marker insertion bit signal to reassertion of the alignment marker insertion bit signal is the `am_period`. For an example that handles this setting for simulation and drives the `i_tx_mii_am` signal appropriately for simulation, refer to the IP core design example for PCS Only variations. For information about how to generate the IP core design example, refer to the *Design Example User Guide*. For information about the `sim_mode` RTL parameter, refer to the *RTL Parameters* section of this user guide. While you hold the value of this signal at 1, you must freeze the values of both TX MII data and TX MII control bits signals.

Figure 42. Transmitting Data Using the PCS Mode TX Interface



The figure above shows how to write packets directly to the PCS mode TX interface.

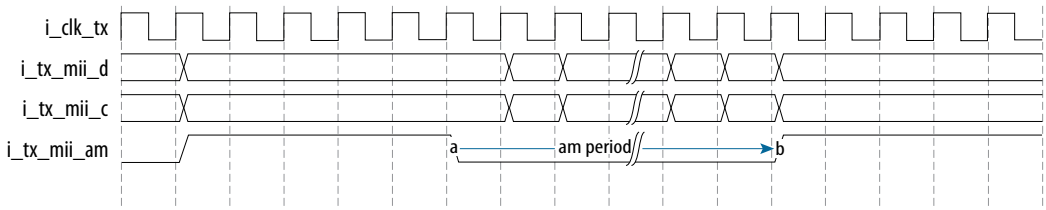
- The packets are written using MII.
  - Each byte in `i_tx_mii_d` has a corresponding bit in `i_tx_mii_c` that indicates whether the byte is a control byte or a data byte; for example, `i_tx_mii_c[1]` is the control bit for `i_tx_mii_d[15:8]`.
- `i_tx_mii_valid` should conform to these conditions:
  - Assert the valid signal only when the ready signal is asserted, and deassert only when the ready signal is deasserted.
  - The two signals can be spaced by a fixed latency between 1 and 10 cycles.
  - When the valid signal deasserts, `i_tx_mii_d` and `i_tx_mii_c` must be paused.
- The byte order for the PCS mode TX interface is opposite of the byte order for the MAC client. Bytes flow from right to left; the first byte to be transmitted from the interface is `i_tx_mii_d[7:0]`.
- The bit order for the PCS mode TX interface is the same as the bit order of the MAC client. The first bit to be transmitted from the interface is `i_tx_mii_d[0]`.

Note: The PCS mode TX interface is not SOP aligned. Any legal ordering of packets in MII format is accepted.

Table 33. Writing a Start Packet Block with Preamble to the PCS Mode TX Interface

MII Data		MII Control		Ethernet Packet Byte
i_tx_mii_d[7:0]	0xFB	i_tx_mii_c[0]	1	Start of Packet
i_tx_mii_d[15:8]	0x55	i_tx_mii_c[1]	0	Preamble
i_tx_mii_d[23:16]	0x55	i_tx_mii_c[2]	0	Preamble
i_tx_mii_d[31:24]	0x55	i_tx_mii_c[3]	0	Preamble
i_tx_mii_d[39:32]	0x55	i_tx_mii_c[4]	0	Preamble
i_tx_mii_d[47:40]	0x55	i_tx_mii_c[5]	0	Preamble
i_tx_mii_d[55:48]	0x55	i_tx_mii_c[6]	0	Preamble
i_tx_mii_d[63:56]	0xD5	i_tx_mii_c[7]	0	SFD

Figure 43. Inserting Alignment Markers



The timing of alignment marker insertion is very rigid. Alignment markers cannot be delayed without disrupting the Ethernet link. Use valid cycles to count the alignment markers. When `i_tx_mii_valid` is low, the alignment marker counters and input must freeze.

The number of cycles for `i_tx_mii_am` to remain high depends on the rate of the interface:

- 100G: 5 cycles
- 25Gx1 with RS-FEC: 4 cycles
- 10Gx1 or 25x1 without PTP or RS-FEC: 0 cycle (tie low)

The number of cycles for am period depends on the rate of the interface and whether in simulation or hardware:

- In simulation, it is common to use a reduced am period for both sides of the link to increase lock-time speed.
  - 100G link: 315
  - 25Gx1 link with RS-FEC: 5119
- In hardware.
  - 100G link: 81915
  - 25Gx1 link with RS-FEC: 81916

2.11.4. RX PCS Interface to User Logic

The E-Tile Hard IP for Ethernet Intel FPGA IP RX client interface in PCS Only variations employs the Media Independent Interface (MII) protocol.

The RX PCS acts as a source and the client acts as a sink in the receive direction.

Table 34. Signals of the MII RX Client Interface

All interface signals are clocked by the RX clock. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `i_sl_rx_data`
- For variants with more than 1 channels: `i_sl_rx_data[n-1:0]`
- For variants with single 100E channel:`i_rx_data`

Signal Name	Width	Description

`i_sl_clk_rx`
`i_sl_clk_rx[n-1:0]`
`i_clk_rx`
	1 bit for each channel	The RX clock for the IP core that drives the channel.

`o_sl_rx_mii_d`
`o_sl_rx_mii_d[n-1:0]`
`o_rx_mii_d`
	64 bits for each channel (10G/25G)
	256 bits (100G)
		RX MII data. Data is in MII encoding. `o_rx_mii_d[7:0]` holds the first byte the IP core received on the Ethernet link. `o_rx_mii_d[0]` holds the first bit the IP core received on the Ethernet link.
		When RX MII valid signal has the value of 0 or the RX valid alignment marker signal has the value of 1, the value on this signal is invalid.

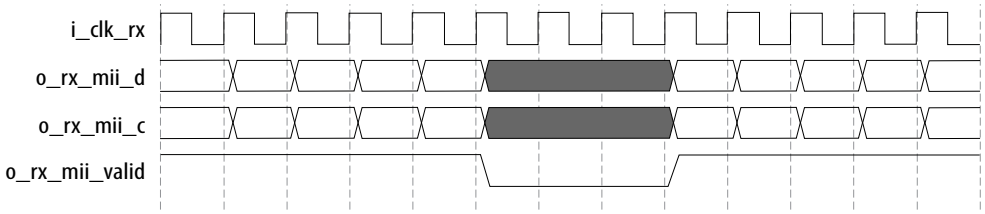
`o_sl_rx_mii_c`
`o_sl_rx_mii_c[n-1:0]`
`o_rx_mii_c`
	8 bits for each channel (10G/25G)
	32 bits (100G
		RX MII control bits. Each bit corresponds to a byte of RX MII data. `o_rx_mii_c[0]` corresponds to `o_rx_mii_d[7:0]`, `o_rx_mii_c[1]` corresponds to `o_rx_mii_d[15:8]`, and so on.
		If the value of a bit is 1, the corresponding data byte is a control byte. If the value of a bit is 0, the corresponding data byte is data.
		The Start of Packet byte (0xFB), End of Packet byte (0xFD), Idle bytes (0x07), and error byte (0xFE) are control bytes, but the preamble bytes, Start of Frame (SFD) byte (0xD5), CRC bytes, and payload bytes are data bytes.
		When RX MII valid signal has the value of 0 or the RX valid alignment marker signal has the value of 1, the value on this signal is invalid.

`o_sl_rx_mii_valid`
`o_sl_rx_mii_valid[n-1:0]`
`o_rx_mii_valid`
	1 bit for each channel	Indicates that the RX MII data, RX MII control bits, and the RX valid alignment marker signals are valid.

`o_sl_rx_mii_am_valid`
`o_sl_rx_mii_am_valid[n-1:0]`
`o_rx_mii_am_valid`
	1 bit for each channel	Indicates the IP core received a valid alignment marker on the Ethernet link.
		When the RX MII valid signal has the value of 0, the value on this signal is invalid. The value of the RX MII valid signal may fall while the IP core is asserting this signal.

`o_sl_rx_pcs_fully_aligned`
`o_sl_rx_pcs_fully_aligned[n-1:0]`
`o_rx_pcs_fully_aligned`
	1 bit for each channel	Asserts when RX PCS is ready to receive data.

Figure 44. Receiving Data Using the PCS Mode RX Interface



The figure above shows how to read packets from the RX PCS using the PCS mode RX interface.

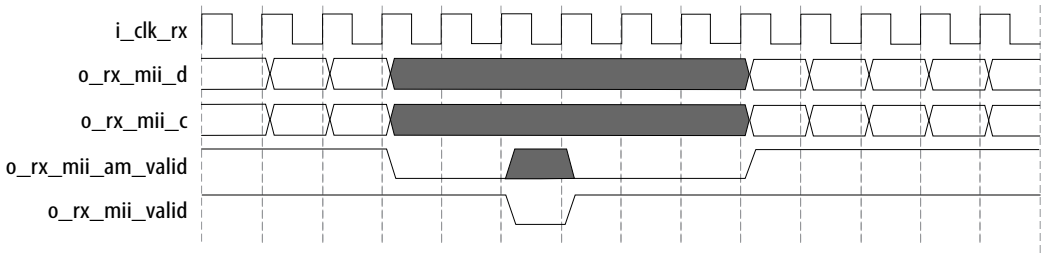
- The packets are MII encoded.
  - Each byte in `o_rx_mii_d` has a corresponding bit in `o_rx_mii_c` that indicates whether the byte is a control byte or a data byte; for example, `o_rx_mii_c[2]` is the control bit for `o_rx_mii_d[23:16]`.
- The data is only valid when `o_rx_mii_valid` is high. The contents of the `o_rx_mii_d` and `o_rx_mii_c` buses are not defined when `o_rx_mii_valid` is low.
- The byte order for the PCS mode RX interface is opposite of the byte order for the MAC client. Bytes flow from right to left; the first byte that the core receives is `o_rx_mii_d[7:0]`.
- The bit order for the PCS mode RX interface is the same as the bit order of the MAC client. The first bit that the core receives is `o_rx_mii_d[0]`.

Note: The PCS mode RX interface is not SOP aligned. New packets can begin on any byte position that is divisible by 8 (PCS data is transferred in 8-byte blocks).

Table 35. Reading a Start Packet Block with Preamble from a PCS Mode TX Interface

MII Data		MII Control		Ethernet Packet Byte
o_rx_mii_d[7:0]	0xFB	o_rx_mii_c[0]	1	Start of Packet
o_rx_mii_d[15:8]	0x55	o_rx_mii_c[1]	0	Preamble
o_rx_mii_d[23:16]	0x55	o_rx_mii_c[2]	0	Preamble
o_rx_mii_d[31:24]	0x55	o_rx_mii_c[3]	0	Preamble
o_rx_mii_d[39:32]	0x55	o_rx_mii_c[4]	0	Preamble
o_rx_mii_d[47:40]	0x55	o_rx_mii_c[5]	0	Preamble
o_rx_mii_d[55:48]	0x55	o_rx_mii_c[6]	0	Preamble
o_rx_mii_d[63:56]	0xD5	o_rx_mii_c[7]	0	SFD

Figure 45. Receiving Alignment Markers



`o_rx_mii_am_valid` indicates the arrival of the alignment markers from the RX PCS. The alignment markers also depend on `o_rx_mii_valid`. When `o_rx_mii_valid` is low, `o_rx_mii_am_valid` is not valid.

The contents of the `o_rx_mii_d` and `o_rx_mii_c` buses are not defined when `o_rx_mii_valid` is low. This is because alignment markers are not part of the 64b/66b encoding, and do not have an MII equivalent.

2.11.5. FlexE and OTN Mode TX Interface

The E-Tile Hard IP for Ethernet Intel FPGA IP TX client interface in FlexE and OTN variations employs the PCS66 interface protocol.

The FlexE and OTN variations allow the application to write 66b blocks to the TX PCS, bypassing the TX MAC.

- In FlexE mode, the TX encoder in the PCS is also bypassed.
- In OTN mode, both the TX encoder and the scrambler are bypassed.

The client acts as a source and the TX PCS acts as a sink in the transmit direction.

Note: The E-Tile Hard IP for Ethernet Intel FPGA IP provides support for the OTN feature. For further inquiries, contact your nearest Intel sales representative.

Table 36. Signals of the PCS66 TX Interface

All interface signals are clocked by the TX clock. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `i_sl_tx_pcs66_d`
- For variants with more than 1 channel: `i_sl_tx_pcs66_d[ch-1:0]`
- For variants with single 100GE channel: `i_tx_pcs66_d`

Signal Name	Width	Description

`i_sl_tx_pcs66_d`
`i_sl_tx_pcs66_d[ch-1:0]`
`i_tx_pcs66_d`
	66 bits for each channel (10G/25G)
	264 bits (100G)
		TX PCS 66b data for 1 block.
		• In FlexE mode, the data presented is scrambled.
		• In OTN mode, the data goes directly to the RS-FEC or PMA.

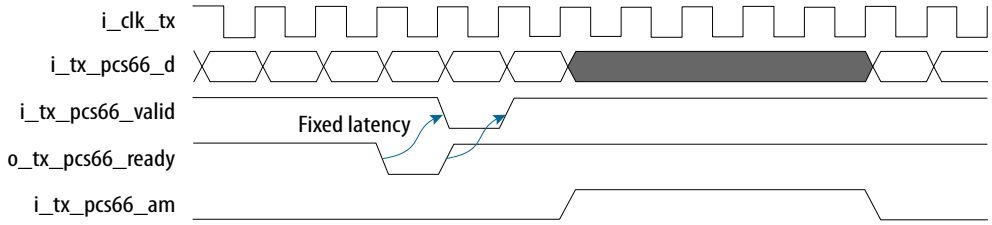
`i_sl_tx_pcs66_valid` `i_sl_tx_pcs66_valid[ch-1:0]`	1 bit for each channel	When asserted, indicates that the TX PCS 66b data is valid. Must be asserted when the TX PCS 66b ready signal is asserted.

continued...

Signal Name	Width	Description
i_tx_pcs66_valid		
o_sl_tx_pcs66_ready o_sl_tx_pcs66_ready[ch-1:0] o_tx_pcs66_ready	1 bit for each channel	TX PCS 66b ready signal. When asserted, indicates the PCS is ready to receive new data.
i_sl_tx_pcs66_am i_sl_tx_pcs66_am[ch-1:0] i_tx_pcs66_am	1 bit for each channel	Alignment marker insertion bit. In FlexE and OTN modes, asserting this signal causes the PCS to allow gaps for the alignment markers in place of the data presented on the TX PCS data signal. The application marks the block as an alignment marker and the scrambler does not process the data.

Figure 46. Transmitting Data Using the PCS66 TX Interface

The figure shows how to write the 66b blocks directly to the TX PCS in FlexE and OTN mode using the PCS66 TX Interface.



TX data is written as 66b blocks. The blocks are expected to be 66b encoded, with the sync header bits in the rightmost bit positions (bits 1 and 0).

- In FlexE mode, the PCS scrambles and stripes the blocks for transmission.
- In OTN mode, the PCS only stripes the blocks for transmission. The input data is expected to be already scrambled.

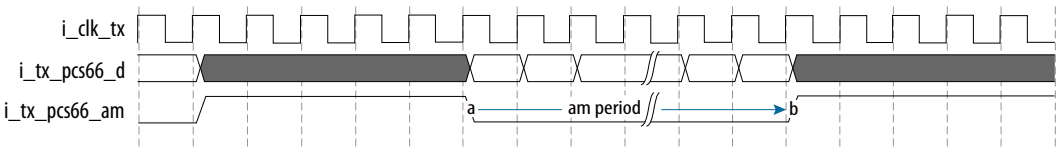
`i_tx_pcs66_valid` should conform to these conditions:

- Assert the valid signal only when the ready signal is asserted, and deassert only when the ready signal is deasserted.
- The two signals can be spaced by a fixed latency between 1 and 10 cycles.
- When the valid signal deasserts, `i_tx_pcs66_d` must be paused.

The block order for the PCS66 mode TX interface is the same as the TX PCS interface. Blocks are transmitted from right to left; the first block to be transmitted from the interface is `i_tx_pcs66_d[65:0]`.

The bit order for the PCS66 mode TX interface is the same as the TX PCS interface. Bits are transmitted from right to left; the first bit to be transmitted from the interface is `i_tx_pcs66_d[0]`.

Figure 47. Inserting Alignment Markers



When PCS66 TX interface is used for FlexE mode, the timing of alignment marker insertion can be controlled from the fabric. The same operations can be performed on *_sl* versions of the ports, with slight variance:

- For 100G channels, the signal causes the alignment markers to be inserted.
- For 10G/25G channels, the signal causes the cycle to be treated as invalid for PCS processing (no changes to scramble).

In FlexE mode, the timing of alignment marker insertion is very rigid. Alignment markers cannot be delayed without disrupting the Ethernet link. Use valid cycles to count the alignment markers. When `i_tx_pcs66_valid` is low, the alignment marker counters and input must freeze.

- The number of cycles for `i_tx_pcs66_am` to remain high for a 100G link is 5 cycles.
- The number of cycles for am period for a 100G link is typically 315 in simulation and 81915 in hardware.

OTN streams are expected to include their own alignment markers. In OTN mode with FEC, you must assert `i_tx_pcs66_am` to indicate the position of the alignment markers. In OTN mode without FEC, `i_tx_pcs66_am` is optional and you can tie the signal low.

Table 37. Alignment Markers Insertion for PCS Direct, FlexE, and OTN Modes

Mode	AM Insertion Bit	AM Date Insertion	TX Data on AM Cycles	Scrambler	64b/66b Encoding/ Decoding
PCS Direct	User-driven	Done by PCS	Ignored	Enabled	Enabled
FlexE	User-driven	Done by PCS	Ignored	Enabled	Disabled
OTN	User-driven	Done by user	AM data	Bypassed	Disabled

2.11.6. FlexE and OTN Mode RX Interface

The E-Tile Hard IP for Ethernet Intel FPGA IP RX client interface in FlexE and OTN variations employs the PCS66 interface protocol.

The FlexE and OTN variations allow the application to read 66b blocks from the RX PCS, bypassing the RX MAC.

The RX PCS acts as a source and the client acts as a sink in the receive direction.

Note: The E-Tile Hard IP for Ethernet Intel FPGA IP provides support for the OTN feature. For further inquiries, contact your nearest Intel sales representative.

Table 38. Signals of the PCS66 RX Interface

All interface signals are clocked by the RX clock. The signal names are standard Avalon-ST signals with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `o_sl_rx_pcs66_d`
- For variants with more than 1 channel: `o_sl_rx_pcs66_d[ch-1:0]`
- For variants with single 100GE channel: `o_rx_pcs66_d`

Name	Width	Description

`o_sl_rx_pcs66_d`
`o_sl_rx_pcs66_d[ch-1:0]`
`o_rx_pcs66_d`
	66 (10G/25G)
	264 (100G)
		RX PCS 66b data for 1 block.
		• In FlexE mode, the RX PCS 66b data is aligned and descrambled but not decoded.
		• In OTN mode, the RX PCS 66b data is only aligned.

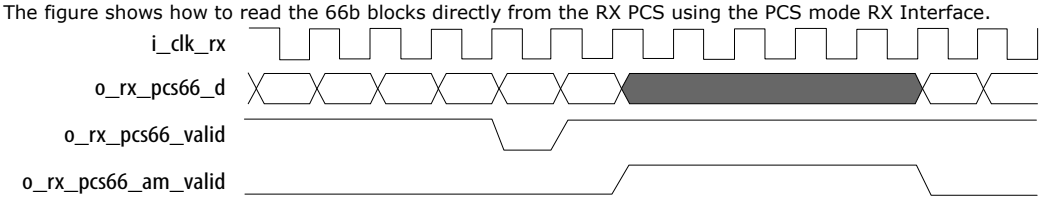
`o_sl_rx_pcs66_valid`
`o_sl_rx_pcs66_valid[ch-1:0]`
`o_rx_pcs66_valid`
	1	When asserted, indicates that the RX PCS 66b data is valid.

`o_sl_rx_pcs66_am_valid`
`o_sl_rx_pcs66_am_valid[ch-1:0]`
`o_rx_pcs66_am_valid`
	1	Alignment marker indicator.
		When asserted, Indicates the blocks on the RX PCS 66b data signal are identified as RS-FEC codeword markers.

`o_sl_rx_pcs_fully_aligned[n-1:0]`
`o_rx_pcs_fully_aligned`
`o_sl_rx_pcs_fully_aligned`
	1 bit for each channel	Asserts when RX PCS is ready to receive data.

Figure 48. Receiving Data Using the PCS66 RX Interface

The figure shows how to read the 66b blocks directly from the RX PCS using the PCS mode RX Interface.



The 66b blocks follow Ethernet 64b/66b convention. The rightmost 2 bits of each 66 block is a 2b sync header and the remaining 64b are data.

- In FlexE mode, the data is aligned and descrambled..
- In OTN mode, the data is only aligned.

The data is only valid when `o_rx_pcs66_valid` is high. The contents of the `o_rx_pcs66_d` bus are not defined when `o_rx_pcs66_valid` is low.

The block order for the PCS66 mode RX interface is the same as the RX PCS interface. Blocks flow from right to left; the first block that the core receives is `o_rx_pcs66_d[65:0]`.

The bit order for the PCS66 mode RX interface is the same as the RX PCS interface. Bits flow from right to left; the first bit that the core receives is `o_rx_pcs66_d[0]`.

o_rx_pcs66_am_valid indicates the arrival of the alignment markers from the RX PCS. The alignment markers also depend on o_rx_pcs66_valid. When o_rx_pcs66_valid is low, o_rx_pcs66_am_valid is not valid.

- In FlexE mode, when o_rx_pcs66_am_valid is high, o_rx_pcs66_d is undefined because the alignment markers do not get descrambled.
- In OTN mode, when o_rx_pcs66_am_valid is high, o_rx_pcs66_d presents the received alignment markers.

Figure 49. Receiving Alignment Markers for FlexE Mode

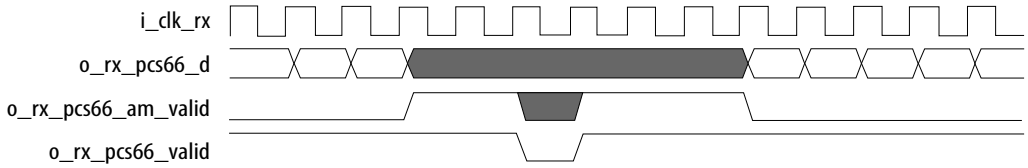
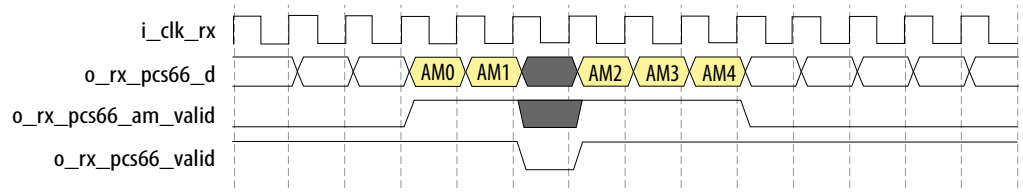


Figure 50. Receiving Alignment Markers for OTN Mode



2.11.7. TX Custom PCS Interface to User Logic

The E-Tile Hard IP for Ethernet Intel FPGA IP TX client interface in custom PCS variation employs the Media Independent Interface (MII) protocol.

The client acts as a source and the TX PCS acts as a sink in the transmit direction.

Table 39. Signals of the MII TX Client Interface

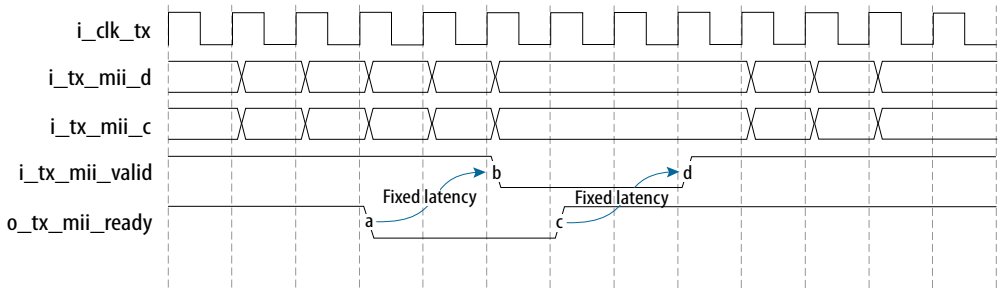
All interface signals are clocked by the TX clock. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. The letter *n* in the signal name is referring to the number of channel. For example, the name of the MII TX data for channel 1 is *i_sl_tx_mii_d[(n*width)-1:0]*.

Signal Name	Width	Description
i_sl_clk_tx[n-1:0]	1 bit for each channel	The TX clock for the IP core that drives the channel.
i_sl_tx_mii_d[(n*width)-1:0]	64 bits for each channel	TX MII data. Data must be in MII encoding. i_tx_mii_d[7:0] holds the first byte the IP core transmits on the Ethernet link. i_tx_mii_d[0] holds the first bit the IP core transmits on the Ethernet link. While the TX MII valid signal has the value of 0 or the alignment marker insertion bit signal has the value of 1, and for one additional clock cycle, you must hold the value of this signal stable. We refer to this behavior as freezing the signal value.

continued...

Signal Name	Width	Description
i_sl_tx_mii_c[(n*width):0]	8 bits for each channel	TX MII control bits. Each bit corresponds to a byte of the TX MII data signal. For example, i_tx_mii_c[0] corresponds to i_tx_mii_d[7:0], i_tx_mii_c[1] corresponds to i_tx_mii_d[15:8], and so on. If the value of a bit is 1, the corresponding data byte is a control byte. If the value of a bit is 0, the corresponding data byte is data. The Start of Packet byte (0xFB), End of Packet byte (0xFD), Idle bytes (0x07), and error byte (0xFE) are control bytes, but the preamble bytes, Start of Frame (SFD) byte (0xD5), CRC bytes, and payload bytes are data bytes. While the TX MII valid signal has the value of 0 or the alignment marker insertion bit signal has the value of 1, you must freeze the value of this signal.
i_sl_tx_mii_valid[n-1:0]	1 bit for each channel	Indicates that the TX MII data signal is valid. You must assert this signal a fixed number of clock cycles after the IP core raises ready signal, and must deassert this signal the same number of clock cycles after the IP core deasserts the ready signal. The number must be in the range of 1–10 clock cycles. While you hold the value of this signal at 0, you must freeze the values of both TX MII data and TX MII control bits signals stable.
o_sl_tx_mii_ready[n-1:0]	1 bit for each channel	Indicates the PCS is ready to receive new data.
i_sl_tx_mii_am i_sl_tx_mii_am[n-1:0]	1 bit for each channel	Alignment marker insertion bit. • In 25Gx1 with RS-FEC variations, you must hold this signal asserted for 4 consecutive clock cycles. • In 10Gx1 or 25Gx1 without RS-FEC variations, you must tie this signal low. The number of valid clock cycles from deassertion of the alignment marker insertion bit signal to reassertion of the alignment marker insertion bit signal is the am_period. For an example that handles this setting for simulation and drives the i_tx_mii_am signal appropriately for simulation, refer to the IP core design example for PCS Only variations. For information about how to generate the IP core design example, refer to the *E-Tile Hard IP for Ethernet Design Example User Guide*. For information about the sim_mode RTL parameter, refer to the *RTL Parameters* section of this user guide. While you hold the value of this signal at 1, you must freeze the values of both TX MII data and TX MII control bits signals.

Figure 51. Transmitting Data Using the PCS Mode TX Interface



The figure above shows how to write packets directly to the PCS mode TX interface.

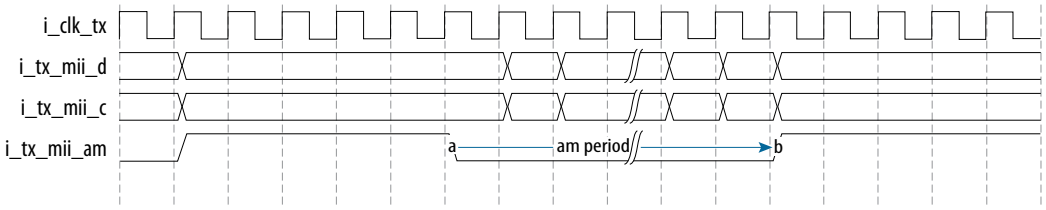
- The packets are written using MII.
  - Each byte in `i_tx_mii_d` has a corresponding bit in `i_tx_mii_c` that indicates whether the byte is a control byte or a data byte; for example, `i_tx_mii_c[1]` is the control bit for `i_tx_mii_d[15:8]`.
- `i_tx_mii_valid` should conform to these conditions:
  - Assert the valid signal only when the ready signal is asserted, and deassert only when the ready signal is deasserted.
  - The two signals can be spaced by a fixed latency between 1 and 10 cycles.
  - When the valid signal deasserts, `i_tx_mii_d` and `i_tx_mii_c` must be paused.
- The byte order for the PCS mode TX interface is opposite of the byte order for the MAC client. Bytes flow from right to left; the first byte to be transmitted from the interface is `i_tx_mii_d[7:0]`.
- The bit order for the PCS mode TX interface is the same as the bit order of the MAC client. The first bit to be transmitted from the interface is `i_tx_mii_d[0]`.

Note: The PCS mode TX interface is not SOP aligned. Any legal ordering of packets in MII format is accepted.

Table 40. Writing a Start Packet Block with Preamble to the PCS Mode TX Interface

MII Data		MII Control		Ethernet Packet Byte
i_tx_mii_d[7:0]	0xFB	i_tx_mii_c[0]	1	Start of Packet
i_tx_mii_d[15:8]	0x55	i_tx_mii_c[1]	0	Preamble
i_tx_mii_d[23:16]	0x55	i_tx_mii_c[2]	0	Preamble
i_tx_mii_d[31:24]	0x55	i_tx_mii_c[3]	0	Preamble
i_tx_mii_d[39:32]	0x55	i_tx_mii_c[4]	0	Preamble
i_tx_mii_d[47:40]	0x55	i_tx_mii_c[5]	0	Preamble
i_tx_mii_d[55:48]	0x55	i_tx_mii_c[6]	0	Preamble
i_tx_mii_d[63:56]	0xD5	i_tx_mii_c[7]	0	SFD

Figure 52. Inserting Alignment Markers



The timing of alignment marker insertion is very rigid. Alignment markers cannot be delayed without disrupting the Ethernet link. Use valid cycles to count the alignment markers. When `i_tx_mii_valid` is low, the alignment marker counters and input must freeze.

The number of cycles for `i_tx_mii_am` to remain high depends on the rate of the interface:

- Links with RS-FEC: 4 cycles
- Links without RS-FEC: 0 cycle (tie low)

The number of cycles for am period depends on the rate of the interface and whether in simulation or hardware:

- In simulation, it is common to use a reduced am period for both sides of the link to increase lock-time speed. The am period for link with RSFEC enabled is set to 5119.
- In hardware, the am period for link with RSFEC enabled is set to 81916.

2.11.8. RX Custom PCS Interface to User Logic

The E-Tile Hard IP for Ethernet Intel FPGA IP RX client interface in custom PCS variations employs the MII protocol.

The RX PCS acts as a source and the client acts as a sink in the receive direction.

Table 41. Signals of the MII RX Client Interface

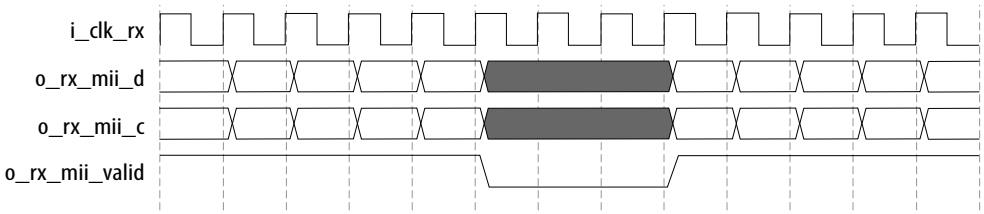
All interface signals are clocked by the RX clock. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. The letter *n* in the signal name is referring to the number of channel. For example, the name of the MII TX data for channel 1 is *i_sl_tx_mii_d[(n*width)-1:0].*

Signal Name	Width	Description
i_sl_clk_rx i_sl_clk_rx[n-1:0]	1 bit for each channel	The RX clock for the IP core that drives the channel.
o_sl_rx_mii_d[(n*width)-1:0]	64 bits for each channel	RX MII data. Data is in MII encoding. o_rx_mii_d[7:0] holds the first byte the IP core received on the Ethernet link. o_rx_mii_d[0] holds the first bit the IP core received on the Ethernet link. When RX MII valid signal has the value of 0 or the RX valid alignment marker signal has the value of 1, the value on this signal is invalid.
o_sl_rx_mii_c[(n*width)-1:0]	8 bits for each channel	RX MII control bits. Each bit corresponds to a byte of RX MII data. o_rx_mii_c[0] corresponds to o_rx_mii_d[7:0], o_rx_mii_c[1] corresponds to o_rx_mii_d[15:8], and so on. If the value of a bit is 1, the corresponding data byte is a control byte. If the value of a bit is 0, the corresponding data byte is data. The Start of Packet byte (0xFB), End of Packet byte (0xFD), Idle bytes (0x07), and error byte (0xFE) are control bytes, but the preamble bytes, Start of Frame (SFD) byte (0xD5), CRC bytes, and payload bytes are data bytes. When RX MII valid signal has the value of 0 or the RX valid alignment marker signal has the value of 1, the value on this signal is invalid.
o_sl_rx_mii_valid[n-1:0]	1 bit for each channel	Indicates that the RX MII data, RX MII control bits, and the RX valid alignment marker signals are valid.
o_sl_rx_mii_am_valid[n-1:0]	1 bit for each channel	Indicates the IP core received a valid alignment marker on the Ethernet link.

continued...

Signal Name	Width	Description
		When the RX MII valid signal has the value of 0, the value on this signal is invalid. The value of the RX MII valid signal may fall while the IP core is asserting this signal.

Figure 53. Receiving Data Using the PCS Mode RX Interface



The figure above shows how to read packets from the RX PCS using the PCS mode RX interface.

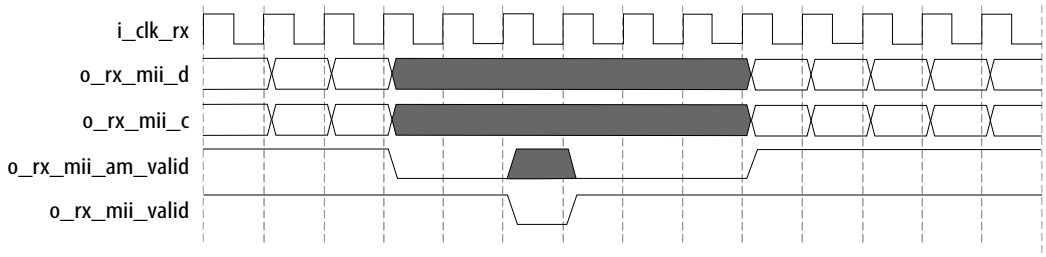
- The packets are MII encoded.
  - Each byte in `o_rx_mii_d` has a corresponding bit in `o_rx_mii_c` that indicates whether the byte is a control byte or a data byte; for example, `o_rx_mii_c[2]` is the control bit for `o_rx_mii_d[23:16]`.
- The data is only valid when `o_rx_mii_valid` is high. The contents of the `o_rx_mii_d` and `o_rx_mii_c` buses are not defined when `o_rx_mii_valid` is low.
- The byte order for the PCS mode RX interface is opposite of the byte order for the MAC client. Bytes flow from right to left; the first byte that the core receives is `o_rx_mii_d[7:0]`.
- The bit order for the PCS mode RX interface is the same as the bit order of the MAC client. The first bit that the core receives is `o_rx_mii_d[0]`.

Note: The PCS mode RX interface is not SOP aligned. New packets can begin on any byte position that is divisible by 8 (PCS data is transferred in 8-byte blocks).

Table 42. Reading a Start Packet Block with Preamble from a PCS Mode TX Interface

MII Data		MII Control		Ethernet Packet Byte
o_rx_mii_d[7:0]	0xFB	o_rx_mii_c[0]	1	Start of Packet
o_rx_mii_d[15:8]	0x55	o_rx_mii_c[1]	0	Preamble
o_rx_mii_d[23:16]	0x55	o_rx_mii_c[2]	0	Preamble
o_rx_mii_d[31:24]	0x55	o_rx_mii_c[3]	0	Preamble
o_rx_mii_d[39:32]	0x55	o_rx_mii_c[4]	0	Preamble
o_rx_mii_d[47:40]	0x55	o_rx_mii_c[5]	0	Preamble
o_rx_mii_d[55:48]	0x55	o_rx_mii_c[6]	0	Preamble
o_rx_mii_d[63:56]	0xD5	o_rx_mii_c[7]	0	SFD

Figure 54. Receiving Alignment Markers



`o_rx_mii_am_valid` indicates the arrival of the alignment markers from the RX PCS. The alignment markers also depend on `o_rx_mii_valid`. When `o_rx_mii_valid` is low, `o_rx_mii_am_valid` is not valid.

The contents of the `o_rx_mii_d` and `o_rx_mii_c` buses are not defined when `o_rx_mii_valid` is low. This is because alignment markers are not part of the 64b/66b encoding, and do not have an MII equivalent.

2.11.9. PMA Direct Interface

The E-Tile Hard IP for Ethernet Intel FPGA IP PMA Direct TX and RX Interfaces are available when you turn on **Include alternate ports** for 10G/25G channels in **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP** variation.

These signals are never connected to the Ethernet hard logic. They are available when you need to switch at run time to PMA modes.

Table 43. Signals of the PMA Direct Interface

Signal Name	Width	Description
`i_sl_tx_pma[ch-1:0]`	80	PMA Direct TX datapath for corresponding transceiver. For all Ethernet cores, this signal does nothing until core is reconfigured at run-time to enter PMA Direct mode.
`o_sl_rx_pma[ch-1:0]`	80	PMA Direct RX datapath for corresponding transceiver. For all Ethernet cores, this signal does nothing until core is reconfigured at run-time to enter PMA Direct mode.

2.11.10. Custom Rate Interface

The E-Tile Hard IP for Ethernet Intel FPGA IP Custom Rate Interface is available when you turn on **Enable custom rate** for 10G/25G channels in **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP** variation. This parameter is not available when **Enable SyncE** is turned on.

Note: Exposing custom rate cadence interface does not change the Ethernet operation. Ethernet protocol does not use this interface.

Note: Enabling this feature exposes the interface from Stratix 10 E-Tile Transceiver Native PHY to the user. It does not enable a custom cadence feature within the E-Tile Hard IP for Ethernet. To enable the custom cadence feature in the Stratix 10 E-Tile Transceiver Native PHY, you must set `flowreg_rate` register in the EHIP TX MAC Feature Configuration on page 224 to 0x7.

Note: Refer to the E-Tile CPRI PHY IP for an example of the interface usage.

Table 44. Signals of the Custom Rate Interface

All of the Custom Rate Interface signals except the `i_sl_custom_cadence[ch-1:0]` signal are asynchronous.

Signal Name	Width	Description
`i_sl_custom_cadence[ch-1:0]`	1	Custom data valid signal. Connect this signal either to a counter that produces a steady data valid cadence that corresponds to the ratio between the clock rate used and the clock rate required, or a system that increases or decreases the data valid cadence based on the current occupancy of transceiver TX FIFO.
`o_sl_txfifo_pfull[ch-1:0]`	1	When asserted, indicates that the transceiver TX FIFO is partially full. At this point, the transceiver FIFO exceeds the programmed *Partially Full* watermark.
`o_sl_txfifo_pempty[ch-1:0]`	1	When asserted, indicates that the transceiver TX FIFO is partially empty. At this point, the transceiver FIFO is below the programmed *Partially Full* watermark.
`o_sl_txfifo_overflow[ch-1:0]`	1	When asserted, indicates that the transceiver TX FIFO has overflowed, and should be reset.
`o_sl_txfifo_underflow[ch-1:0]`	1	When asserted, indicates that the transceiver TX FIFO has underflowed, and should be reset.

2.11.11. Deterministic Latency Interface

The E-Tile Hard IP for Ethernet Intel FPGA IP Deterministic Latency Interface is available when you turn on **Include deterministic latency measurement interface** for 10G/25G channels in **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP** variation.

Note: This feature is for internal use only, do not enable this setting.

When setting is turned on, you can view the deterministic latency interface directly from the Stratix 10 E-Tile Transceiver Native PHY.
Use the deterministic latency interface if you want to measure the latency of the datapath when running a stack that does not include MAC.

Note: Exposing deterministic latency interface does not change the Ethernet operation. Ethernet protocol does not use this interface. Enabling this feature exposes the interface from Stratix 10 E-Tile Transceiver Native PHY to the user.

Note: Refer to the E-Tile CPRI PHY IP in Deterministic Latency Calculation on page 303 for an example of the interface usage.

Table 45. Signals of the Deterministic Latency Interface

All of the Deterministic Latency Interface signals are asynchronous.

Signal Name	Width	Description
o_tx_dl_async_pulse[ch-1:0]	1	Asynchronous output pulse signal for the transmitter latency measurement[28] of the deterministic latency application. There is a start pulse and a stop pulse.
o_sl_rx_dl_async_pulse[ch-1:0]	1	Asynchronous output pulse signal for the receiver latency measurement[28] of the deterministic latency application. There is a start pulse and a stop pulse.
i_sl_latency_sclk[ch-1:0]	1	Clock signal for latency measurement[28] of the deterministic latency application.
i_sl_tx_dl_measure_sel[ch-1:0]	1	Mux select signal for the transmitter latency measurement.[28] 1 is for the datapath latency. 0 is for the wire delay.
i_sl_rx_dl_measure_sel[ch-1:0]	1	Mux select signal for the receiver latency measurement.[28] 1 is for the datapath latency. 0 is for the wire delay.

2.11.12. 1588 PTP Interface

The E-Tile Hard IP for Ethernet Intel FPGA IP 1588 PTP Interface is available for 10G/25G designs when you turn on **Enable IEEE 1588 PTP** for 10G/25G channels in **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP** variation. The 1588 Precision Time Protocol (PTP) timestamp information provided is as defined in the *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard*.

These signals are active only when your selected channel is configured to provide a MAC+PTP+PCS stack.

All 1-step and 2-step TX/RX Timestamp interface signals are common for both, basic and advanced PTP accuracy modes with the exception of these TOD interface signals:

- In basic PTP accuracy mode: `i_ptp_tod`
- In advanced PTP accuracy mode: `i_sl_ptp_tx_tod/i_sl_ptp_tx_tod`

All interface signals, except the TOD interface signals[29] and PTP ready signals[30], are clocked by the TX or RX clock.

- TX clock represents `i_sl_clk_tx` clock when the asynchronous adapter is disabled and `i_sl_async_clk_tx` clock when the asynchronous adapter is enabled.
- RX clock represents `i_sl_clk_rx` clock when the asynchronous adapter is disabled and `i_sl_async_clk_rx` clock when the asynchronous adapter is enabled.

---

(28) For more information, review the *Latency Measurement* section.

(29) The applicable TOD interface signals: `i_ptp_tod` in the basic PTP accuracy mode, `i_sl_ptp_tx_tod/i_sl_ptp_rx_tod` in the advance PTP accuracy mode.

(30) The applicable PTP ready signals: `o_tx_ptp_ready/o_sl_tx_ptp_ready` and `o_rx_ptp_ready/o_sl_rx_ptp_ready`.

The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. For example:

- For 100GE channel or single channel 10GE/25GE: `i_ptp_ins_ets`
- For selected 10GE/25GE channel: `i_sl_ptp_ins_ets[(n*width)-1:0]`

Table 46. Signals of the 1-Step TX Timestamp Interface

Signal Name	Width	Description
`i_ptp_ins_ets` `i_sl_ptp_ins_ets[n-1:0]`	1	Egress timestamp into the current TX Packet on the respective channel. • Valid only when the TX valid and TX SOP signals are asserted. • Do not use when the TX skip CRC signal (e.g. `i_tx_skip_crc`) is asserted. The CRC for the TX packet must be recalculated after the egress timestamp is written. • Do not use when the residence time timestamp signal (e.g. `i_ptp_ins_cf`) is asserted. You cannot update residence time and insert an egress timestamp on the same packet. • Set the position of the PTP timestamp field in the TX packet (e.g. `i_ptp_ts_offset`) to the byte position of the start of the timestamp field in the PTP header. • Set the format for the PTP 1-step operation (e.g. `i_ptp_ts_format`) to the desired timestamp format. • If the selected timestamp format requires a 96b timestamp, set the PTP correction field in the TX packet (e.g. `i_ptp_cf_offset`) to the byte position of the start of the correction field in the PTP header.
`i_ptp_ins_cf` `i_sl_ptp_ins_cf[n-1:0]`	1	Residence time timestamp into the correction field in the current TX packet on the respective channel. • Valid only when the TX valid and TX SOP signals are asserted. • Do not use when the TX skip CRC signal (e.g. `i_tx_skip_crc`) is asserted. The CRC for the TX packet must be recalculated after the residence time is written. • Do not use when the egress time timestamp signal (e.g. `i_ptp_ins_ets`) is asserted. You cannot update residence time and insert an egress timestamp on the same packet. • Provide the core with the ingress timestamp of the current packet (e.g. assert `i_ptp_tx_its`) when it entered the system, so that a residence time can be calculated. • Set the position of the PTP correction field in the TX packet (e.g. `i_ptp_cf_offset`) to the byte position of the start of the correction field in the PTP header. *Note:* If the PTP packet resides in the system for more than 4 seconds, the correction field will show a mismatched value with a very large number. • Set `i_ptp_ts_format` to 0.

continued...

Signal Name	Width	Description
i_ptp_zero_csum i_sl_ptp_zero_csum[n-1:0]	1	Overwrites the checksum in a UDP packet carried inside the current TX packet with zeros. • Valid only when the TX valid and TX SOP signals are asserted. • Do not use when the TX skip CRC signal (e.g. i_tx_skip_crc) is asserted. The CRC for the TX packet must be recalculated after the checksum is changed. • Do not use when the update extended bytes field signal (e.g i_ptp_update_eb) is asserted. You cannot set a UDP checksum to 0, and update an extension field to cancel out checksum changes on the same packet. • Set the position of the UDP checksum field in the TX packet (e.g. i_ptp_csum_offset) to the byte position of the start of the UDP checksum in the TX packet.
i_ptp_update_eb i_sl_ptp_update_eb[n-1:0]	1	Overwrites the extended bytes field in an IPv6 packet carried inside the current TX packet with a value that cancels out changes to the checksum due to changes to the UDP packet. • Valid only when the TX valid and TX SOP signals are asserted. • Do not use when the TX skip CRC signal (e.g. i_tx_skip_crc) is asserted. The CRC for the TX packet must be recalculated after the checksum is changed. • Do not use when the overwrite a UDP checksum with zeros signal (e.g. i_ptp_zero_csum) is asserted. You cannot set a UDP checksum to 0, and update an extension field to cancel out checksum changes on the same packet. • Set the position of the first byte of the extended bytes field in the TX packet (e.g. i_ptp_eb_offset) to the byte position of the start of the UDP checksum in the TX packet.
i_ptp_ts_format i_sl_ptp_ts_format[n-1:0]	1	Format of the PTP 1-step operation on the respective channel. • 0: Use IEEE 1588v2 timestamp and correction field formats (96 bits) • 1: Use IEEE 1588v1 timestamp format (64 bits) Valid only when either the egress time timestamp signal (i_ptp_ins_ets) or the residence time timestamp signal (i_ptp_ins_cf), and the TX valid signal, and SOP signal are asserted. When i_ptp_ins_cf is asserted, this port must be set to 0. The correction field is only applicable to IEEE 1588v2 format.
i_ptp_ts_offset i_sl_ptp_ts_offset[(n*16)-1:0]	16	Position of the PTP timestamp field in the current TX packet. • Valid only when the TX valid and TX SOP signals are asserted. • It is the offset of the first octet of the field from the start of the frame, where the first byte of the frame (the first destination MAC address octet) is position 0. • The IEEE 1588v2 PTP timestamp field is 10 octets long (80 bits), and the IEEE 1588v1 timestamp is 8 octets long (64bits), starting from the position given by the offset. Because the IEEE 1588v2 timestamps are actually 96 bits long, the lower 16 bits of the timestamp are placed in the lower 2 octets of the correction field. ***Caution:*** You must set the offset to a position within the TX packet, or the PTP insertion operation will fail. You must not also overlap the PTP fields.

continued...

Signal Name	Width	Description
i_ptp_cf_offset i_sl_ptp_cf_offset[(n*16)-1:0]	16	Position of the PTP correction field in the current TX packet. • Valid only when the TX valid and TX SOP signals are asserted. • It is the offset of the first octet of the field from the start of the frame, where the first byte of the frame (the first destination MAC address octet) is position 0. • The PTP correction field is 8 octets long, starting from the position given by the offset. When 96-bit timestamps are used, the MAC places the lower 16 bits of the timestamp in the lower 2 octets of the correction field. ***Caution:*** You must set the offset to a position within the TX packet, or the PTP insertion operation will fail. You must not also overlap the PTP fields.
i_ptp_csum_offset i_sl_ptp_csum_offset[(n*16)-1:0]	16	Position of the first byte of a UDP checksum field in the current TX packet. • Valid only when the checksum overwrite in a UDP packet (e.g. i_ptp_zero_csum), TX valid, TX SOP signals are asserted. • It is the offset of the first octet of the field from the start of the frame, where the first byte of the frame (the first destination MAC address octet) is position 0. ***Caution:*** You must set the offset to a position within the TX packet, or the PTP insertion operation will fail. You must not also overlap the PTP fields.
i_ptp_eb_offset i_sl_ptp_eb_offset[(n*16)-1:0]	16	Position of the first byte of extended bytes field in the current TX packet. • Valid only when the extended bytes overwrite in an IPv6 packet (e.g. i_ptp_update_eb), TX valid, TX SOP signals are asserted. • It is the offset of the first octet of the field from the start of the frame, where the first byte of the frame (the first destination MAC address octet) is position 0. ***Caution:*** You must set the offset to a position within the TX packet, or the PTP insertion operation will fail. You must not also overlap the PTP fields.
i_ptp_tx_its i_sl_ptp_tx_its[(n*96)-1:0]	96	Ingress timestamp for a TX packet that requires a residence time calculation (e.g. i_ptp_ins_cf = 1). This timestamp is the time at which the packet arrives in the system. The TX MAC compares this time to the time at which the packet leaves the system to generate a residence time. You must set the i_ptp_ts_format to 0. Residence time calculation/correction is only applicable to IEEE 1588v2 format. Valid only when the TX valid and TX SOP signals are asserted.

Table 47. Signals of the 2-Step TX Timestamp Interface

Use the 2-step TX Timestamp to request for 2-step TX Timestamps when a packet is transmitted.

Signal Name	Width	Description
i_ptp_ts_req i_sl_ptp_ts_req[n-1:0]	1	Request a 2-step timestamp signal for the current TX packet. When asserted, generates a TX timestamp for the current packet. Valid only when the TX valid and TX SOP signals are asserted.
i_ptp_fp i_sl_ptp_fp[(n*8)-1:0]	8	Fingerprint signal for current TX packet. Assigns an 8-bit fingerprint to a TX packet that is being transmitted, so that the 2-step or 1-step PTP timestamp associated with the TX packet can be identified. The timestamp returns with the same fingerprint.

continued...

Signal Name	Width	Description
		• Use a range of fingerprints from 0..31 or larger, to avoid the possibility of assigning the same fingerprint to 2 TX packets that are being processed. • Choose an easy-to-decode null fingerprint for any packets that do not require an egress timestamp. For example, if you use a range of 0..31, make 32 the null fingerprint. Valid only when the TX valid and TX SOP signals are asserted.
o_ptp_ets_valid o_sl_ptp_ets_valid[n-1:0]	1	1-step or 2-step egress timestamp valid signal. When asserted, the fingerprint and egress timestamp signals present valid output on this cycle. This signal is asserted after the timestamp is generated in one of these scenarios: • You assert i_ptp_ts_req to request 2-step timestamp. • You assert i_ptp_ins_ets to request 1-step timestamp insertion.
o_ptp_ets o_sl_ptp_ets[(n*96)-1:0]	96	2-step or 1-step egress timestamp signal. This port presents an egress timestamp for the TX Packet that was transmitted with the fingerprint given by o_ptp_ets_fp. Following conditions apply to this signal: • Valid only when the egress timestamp valid (o_ptp_ets_valid) signal is asserted. • The timestamp is in 1588v2 format (96b). • This corresponds to the generated timestamp when you assert i_ptp_ts_req to request 2-step timestamp or when you assert i_ptpt_ins_ets to request 1-step timestamp insertion. • The timestamp is for the packet whose fingerprint matches the fingerprint with the egress timestamp. • All timestamps are referenced to the copy of the Time-of-Day provided to the IP core through the i_ptp_tod port.
o_ptp_ets_fp o_sl_ptp_ets_fp[(n*8)-1:0]	8	Fingerprint for the current 2-step or 1-step egress timestamp. You can use the fingerprint to determine which TX packet the timestamp belongs to. Valid only when the egress timestamp valid signal (o_ptp_ets_valid) is asserted.

Table 48. Signals of the Time of Day Interface

The time of day interface allows the core to reference all of its timestamps to the time of day as it is known locally.

Signal Name	Width	Description
i_ptp_tod	96	When you set the **PTP Accuracy Mode** to **Basic Mode**, this signal presents the current Time of Day, according to the local clock, to the Ethernet IP core. All channels in the same IP core share the same ToD port. When your design implements multiple IP cores, each IP core requires a ToD IP. The timestamp is in IEEE 1588v2 format (96b).
i_sl_ptp_tx_tod[(n*96)-1:0]	96	When you set the **PTP Accuracy Mode** to **Advanced Mode**, represents the TX Time of Day, according to the local clock. The timestamp is in IEEE 1588v2 format (96 bits).
i_sl_ptp_rx_tod[(n*96-1):0]	96	When you set the **PTP Accuracy Mode** to **Advanced Mode**, represents the RX Time of Day, according to the local clock. The timestamp format is in IEEE 1588v2 format (96 bits).

Table 49. Signals of the RX Timestamp Interface

The RX Timestamp interface allows each channel to provide RX timestamps when packets arrive.

Signal Name	Width	Description
o_ptp_rx_its o_sl_ptp_rx_its[(n*96)-1:0]	96	Ingress RX timestamp signal. Presents the ingress timestamp for the incoming RX packet on the respective channel. Valid only when the RX valid and RX SOP signals are asserted. The timestamp is in 1588v2 format (96b).

Table 50. Signals of the PTP Status Interface

The PTP Status interface lets applications using PTP functions know when the PTP timestamp logic is ready for use.

Signal Name	Width	Description
o_tx_ptp_ready o_sl_tx_ptp_ready[n-1:0]	1	TX PTP ready signal. When asserted, the core to ready to request for TX PTP functions on the respective channel.
o_rx_ptp_ready o_sl_rx_ptp_ready[n-1:0]	1	RX PTP ready signal. When asserted,the RX PTP logic ready for use on the respective channel. After reset and PMA adaptation, the signal gets asserted after link partner sends up to 20 Ethernet packets.

2.11.13. Ethernet Link and Transceiver Signals

The E-Tile Hard IP for Ethernet Intel FPGA IP includes transceivers that implement two or four physical lanes at the line rates required for Ethernet channels.

Table 51. Transceiver Signals

Note: n = number of channels.

Signal	Description
o_tx_serial[n-1:0] (10GE/25GE) o_tx_serial[3:0] (100GE)	TX transceiver data. Each o_tx_serial bit becomes two physical pins that form a differential pair.
i_rx_serial[n-1:0] (10GE/25GE) i_rx_serial[3:0] (100GE)	RX transceiver data. Each i_rx_serial bit becomes two physical pins that form a differential pair.
i_clk_ref[n-1:0](10GE/25GE) i_clk_ref(100GE)	The input clock i_clk_ref is the reference clock for the high-speed serial clocks. This clock must have the same frequency as specified in **PHY Reference Frequency** parameter with a ±100 ppm accuracy per the *IEEE 802.3-2015 Ethernet Standard*. This signal supports the following frequencies: • 156.25 MHz • 322.265625 MHz • 312.5 MHz • 644.53125 MHz In addition, this clock must meet the jitter specification of the *IEEE 802.3-2015 Ethernet Standard*. The PLL and clock generation logic use this reference clock to derive the transceiver and PCS clocks. The input clock should be a high quality signal on the appropriate dedicated clock pin. Refer to the *Intel Stratix 10 Device Data Sheet* or *Intel Agilex Device Data Sheet* for transceiver reference clock phase noise specifications.

continued...

Signal	Description
	The index represents the number of reference clocks supported by this IP. The number is equivalent to the number of channels when `refclk_mux` is 0 with maximum value of 5 when `refclk_mux` is 1. *Note:* By default, all channels are mapped to `i_clk_ref[0]` regardless of port's width or `refclk_mux` setting. For more information on how to change to a different reference clock, refer to the *Switching Reference Clocks* section in the *E-Tile Transceiver PHY User Guide*.
`o_tx_pll_locked[n-1:0]`	The `o_tx_pll_locked[n-1:0]` signal indicates when the transceiver PLL output clocks are locked. The `o_clk_pll_div64` and `o_clk_pll_div66` clocks are reliable only after this signal bits are all high.

Related Information

- Intel Stratix 10 Device Data Sheet
- Intel Agilex Device Data Sheet
- E-Tile Transceiver PHY User Guide
  Information about the Intel Stratix 10 Native PHY IP core.

2.11.14. Reconfiguration Interfaces and Signals

2.11.14.1. Ethernet Reconfiguration Interfaces

You access Ethernet control and status registers of the E-Tile Hard IP for Ethernet Intel FPGA IP during normal operation using an Avalon memory-mapped interface. The interface responds regardless of the link status. It also responds when the IP core is in a reset state driven by any reset signal or soft reset other than the `i_csr_rst_n` signal.

Asserting the `i_csr_rst_n` signal resets all Ethernet control and status registers, including the statistics counters; while this reset is in process, reads or writes to addresses in the Ethernet Hard IP will be delayed.

Table 52. Ethernet Reconfiguration Interface

The signals in this interface are clocked by the `i_reconfig_clk` clock and reset by the `i_reconfig_reset` signal. This clock and reset are used for all the reconfiguration interfaces in the IP core. However, the two interfaces access disjoint sets of registers. The signal names are standard Avalon streaming interface signals with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `i_sl_eth_reconfig_addr`
- For variants with more than 1 channel: `i_sl_eth_reconfig_addr[n-1:0]`
- For variants with single 100GE channel: `i_eth_reconfig_addr`

Port Name	Width	Description

`i_sl_eth_reconfig_addr`
`i_sl_eth_reconfig_addr[n-1:0]`
`i_eth_reconfig_addr`
	21 (100GE)
	19 (10GE/25GE)
		Address bus for Ethernet control and status registers in the respective channel.

`i_sl_eth_reconfig_write`
`i_sl_eth_reconfig_write[n-1:0]`
`i_eth_reconfig_write`
	1	Write request signal for Ethernet control and status registers in the respective channel.

continued...

Port Name	Width	Description
i_sl_eth_reconfig_read i_sl_eth_reconfig_read[n-1:0] i_eth_reconfig_read	1	Read request signal for Ethernet control and status registers in the respective channel.
i_sl_eth_reconfig_writedata i_sl_eth_reconfig_writedata[n-1:0] i_eth_reconfig_writedata	32	Write data for Ethernet control and status registers in the respective channel.
i_sl_eth_reconfig_readdata i_sl_eth_reconfig_readdata[n-1:0] i_eth_reconfig_readdata	32	Read data from reads to Ethernet control and status registers in the respective channel.
o_sl_eth_reconfig_readdata_valid o_sl_eth_reconfig_readdata_valid[n-1:0] o_eth_reconfig_readdata_valid	1	Read data from Ethernet control and status registers is valid in the respective channel.
i_sl_eth_reconfig_waitrequest i_sl_eth_reconfig_waitrequest[n-1:0] i_eth_reconfig_waitrequest	1	Avalon memory-mapped interface stalling signal for operations on Ethernet control and status registers in the respective channel.

Related Information

E-Tile Transceiver PHY User Guide
Provides more information about the transceiver reconfiguration interface in E-tile devices, including timing diagrams for reads and writes.

2.11.14.2. Transceiver Reconfiguration Interfaces

You access the control and status registers of the Intel Stratix 10 E-tile transceivers during normal operation using an Avalon memory-mapped interface. The interface responds regardless of the link status.

Asserting the `i_csr_rst_n` signal resets all Ethernet control and status registers, including the statistics counters; while this reset is in process, the Ethernet reconfiguration interface does not respond.

Table 53. Transceiver Reconfiguration Interface Ports to Native PHY Reconfiguration Interfaces

The signals in this interface are clocked by the `i_reconfig_clk` clock and reset by the `i_reconfig_reset` signal. All interface signals are clocked by the RX clock. The signal names are standard Avalon memory-mapped interface signals with slight differences for different variations. For example:

- For single 10GE/25GE channel variant: `i_xcvr_reconfig_address`
- For 1–4 10GE/25GE channels variant: `i_xcvr_reconfig_address[n-1:0]`
- For single 100GE channel variant:`i_xcvr_reconfig_address[19*w-1:0]`; each lane = 19 bit, w = 4
- For single 100GE or 1–4 10GE/25GE channels variant: `i_xcvr_reconfig_address[ch-1:0]`; ch = number of transceivers

Port Name	Width	Description
`i_xcvr_reconfig_address`	19	Address bus for transceiver control and status registers.
`i_xcvr_reconfig_write`	1	Transceiver write signal. When asserted, writes data on the reconfiguration write data bus.
`i_xcvr_reconfig_read`	1	Transceiver read signal. When asserted, starts a read cycle.
`i_xcvr_reconfig_writedata`	8 bits each lane	Transceiver write data bus. When asserted, presents transceiver data written on a write cycle.
`o_xcvr_reconfig_readdata`	8 bits each lane	Transceiver read data bus. When asserted, presents transceiver data read on a read cycle.
`o_xcvr_reconfig_waitrequest`	1	Indicates the Avalon memory-mapped interface interface is busy. The read or write cycle is only complete when this signal goes low.

Related Information

E-Tile Transceiver PHY User Guide
Provides more information about the transceiver reconfiguration interface in E-tile devices, including timing diagrams for reads and writes.

2.11.14.3. RS-FEC Reconfiguration Interfaces

You access RS-FEC control and status registers of the E-Tile Hard IP for Ethernet Intel FPGA IP during normal operation using an Avalon-MM interface.

Table 54. RS-FEC Reconfiguration Interface

The signals in this interface are clocked by the `i_reconfig_clk` clock and reset by `i_reconfig_reset`.

Port Name	Width	Description
`i_rsfec_reconfig_addr`	11	Address bus for RS-FEC control and status registers in the respective channel.
`i_rsfec_reconfig_write`	1	Write request signal for RS-FEC control and status registers in the respective channel.
`i_rsfec_reconfig_read`	1	Read request signal for RS-FEC control and status registers in the respective channel.
`i_rsfec_reconfig_writedata`	8	Write data for RS-FEC control and status registers in the respective channel.
`o_rsfec_reconfig_readdata`	8	Read data from reads to RS-FEC control and status registers in the respective channel.
`o_rsfec_reconfig_waitrequest`	1	Avalon-MM stalling signal for operations on RS-FEC control and status registers in the respective channel.

Related Information

Intel Stratix 10 E-Tile Transceiver PHY User Guide
Provides more information about the transceiver reconfiguration interface in E-tile devices, including timing diagrams for reads and writes.

2.11.14.4. PTP Reconfiguration Interfaces

When PTP is used, you access the control and status registers controlling the transceiver channels used for the PTP interface of the E-Tile Hard IP for Ethernet Intel FPGA IP during normal operation using an Avalon memory-mapped interface.

The PTP reconfiguration interfaces are available when you use the 100G channel with 1 to 4 10G/25G channels, RS-FEC, and PTP variant.

Table 55. PTP Reconfiguration Interface

The signals in this interface are clocked by the `i_reconfig_clk` clock and reset by `i_reconfig_reset`. Use the default AIB and transceiver configurations for PTP channels..

Note: Ports with a width including p are allocated 1 per PTP EMIB instance in the module.

Port Name	Width	Description
`i_ptp_reconfig_address[p*1-9:10]`	19 bits each lane	Control and status register address bus for PTP channel.
`i_ptp_reconfig_write[p-1:0]`	1	PTP channel write signal asserted to write data on reconfiguration write data bus.
`i_ptp_reconfig_read[p-1:0]`	1	PTP channel read signal asserted to start a read cycle.
`i_ptp_reconfig_writedata[p*8-1:0]`	8	PTP channel data to be written on a write cycle.
`o_ptp_reconfig_readdata[p*8-1:0]`	8	PTP channel data that was read by a read cycle.
`o_ptp_reconfig_waitrequest[p-1:0]`	1	Avalon memory-mapped interface stalling signal for operations on PTP control and status registers in the respective channel. The read/write cycle is only complete when this signal goes low.

Related Information

E-Tile Transceiver PHY User Guide
Provides more information about the transceiver reconfiguration interface in E-tile devices, including timing diagrams for reads and writes.

2.11.15. Miscellaneous Status and Debug Signals

The E-Tile Hard IP for Ethernet Intel FPGA IP provides a handful of status and debug signals to support visibility into the actions of the IP core and the stability of IP core output clocks.

Table 56. Miscellaneous Status and Debug Signals

All of the miscellaneous output status and debug signals except the `i_stats_snapshot` signal are asynchronous and must be synchronized before they are used. The signal names are standard with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: `o_sl_tx_lanes_stable`
- For variants with more than 1 channel: `o_sl_tx_lanes_stable[ch-1:0]`
- For variants with single 100GE channel: `o_tx_lanes_stable`

Signal	Width	Description
`o_cdr_lock[n-1:0]` (*n* is the number of transceivers)	[n-1:0]	Indicates that the recovered clocks are locked to data. The `o_clk_rec_div64[n]` and `o_clk_rec_div66[n]` clocks are reliable only after `o_cdr_lock[n]` is asserted.

`o_sl_tx_lanes_stable`
`o_sl_tx_lanes_stable[n-1:0]`
`o_tx_lanes_stable`
	1	Asserted when all physical TX lanes are stable and ready to transmit data for the corresponding Ethernet channel. Each channel has its own `o_tx_lanes_stable`.

`o_sl_rx_block_lock`
`o_sl_rx_block_lock[n-1:0]`
`o_rx_block_lock`
	1	Asserted when the corresponding Ethernet channel completes 66-bit block boundary alignment on all PCS lanes. Each channel has its own block lock signal.

`o_sl_rx_am_lock`
`o_sl_rx_am_lock[n-1:0]`
`o_rx_am_lock`
	1	Asserted when the RX PCS completes detection of alignment markers and deskew of the virtual PCS lanes in the corresponding Ethernet 100G channel.

`o_sl_rx_pcs_ready`
`o_sl_rx_pcs_ready[n-1:0]`
`o_rx_pcs_ready`
	1	Asserted when the RX lanes of the corresponding Ethernet channel are fully aligned and ready to receive data.

`o_sl_local_fault_status`
`o_sl_local_fault_status[n-1:0]`
`o_local_fault_status`
	1	Asserted when the RX MAC of the corresponding Ethernet channel detects a local fault: the RX PCS detected a problem that prevents it from receiving data. This signal is functional only if you set the **Choose Link Fault generation option** parameter to the value of **Bidirectional** or **Unidirectional** in the parameter editor or if you overwrite the parameter editor parameter by setting the `link_fault_mode` RTL parameter to the value of `lf_bidir` or `lf_unidir`.

`o_sl_remote_fault_status`
`o_sl_remote_fault_status[n-1:0]`
`o_remote_fault_status`
	1	Asserted when the RX MAC of the corresponding Ethernet channel detects a remote fault: the remote link partner sent remote fault ordered sets indicating that it is unable to receive data. This signal is functional only if you set the **Choose Link Fault generation option** parameter to the value of **Bidirectional** in the parameter editor or if you overwrite the parameter editor parameter by setting the `link_fault_mode` RTL parameter to the value of `lf_bidir`.

`i_sl_stats_snapshot`
`i_sl_stats_snapshot[n-1:0]`
`i_stats_snapshot`
	1	Directs the IP core to record a snapshot of the current state of the statistics registers. Assert this signal to perform the function of both the TX and RX statistics register shadow request fields at the same time, or to perform that function for multiple instances of the IP core simultaneously. Refer to *TX Statistics Counters* and *RX Statistics Counters*.
		Assert the signal for the desired duration of the freeze of read values in the statistics counters. The rising edge sets the `tx_shadow_on` field (bit [1]) of the CNTR_TX_STATUS register at offset 0x846 and the `rx_shadow_on` field (bit [1]) of the CNTR_RX_STATUS register at offset 0x946. to the value of 1 and the falling edge resets these bits.

continued...

Signal	Width	Description
		This signal is synchronous with the i_clk_tx clock.
o_sl_rx_hi_ber o_sl_rx_hi_ber[n-1:0] o_rx_hi_ber	1	Asserted to indicate the RX PCS of the corresponding Ethernet channel is in a HI BER state according to Figure 82-15 in the *IEEE 802.3-2015 Standard*. The IP core uses this signal in autonegotiation and link training.
o_sl_ehip_ready o_sl_ehip_ready[n-1:0] o_ehip_ready	1	The Ethernet channel deasserts this signal in response to an i_csr_rst_n or i_tx_rst_n reset, or either of the corresponding soft resets. After the reset process completes, the channel reasserts this signal to indicate that the Hard IP for Ethernet block has completed initialization and is ready to interoperate with the main Intel Stratix 10 die. While the o_ehip_ready signal is low, the channel's datapath is not ready for data on the client interface nor ready for register accesses on the Ethernet reconfiguration interface.

Related Information

Monitoring Transceiver Signals

2.11.16. Reset Signals

The IP core has three external hard reset inputs. These resets are asynchronous and are internally synchronized. In addition the IP core supports a dedicated reset signal that resets the transceiver and Ethernet reconfiguration interfaces but not the registers they control.

Assert the asynchronous resets for ten i_reconfig_clk cycles or until you observe the effect of their specific reset. Asserting the external hard reset i_csr_rst_n returns all Ethernet reconfiguration registers to their original values. o_rx_pcs_ready and o_tx_lanes_stable are asserted when the core has exited reset successfully.

Table 57. Reset Signals

All of the IP core reset signals except the i_reconfig_reset signal are asynchronous. The signal names are standard with slight differences to indicate the variations. For example:

- For variants with single 10GE/25GE channel: i_sl_tx_rst_n
- For variants with more than 1 channel: i_sl_tx_rst_n[n-1:0]
- For variants with single 100E channel: i_tx_rst_n

Signal	Description
i_sl_tx_rst_n i_sl_tx_rst_n[n-1:0] i_tx_rst_n	Active-low hard reset signal. Resets the TX interface, including the TX PCS and TX MAC. This reset leads to the deassertion of the o_tx_lanes_stable output signal.
i_sl_rx_rst_n i_sl_rx_rst_n[n-1:0] i_rx_rst_n	Active-low hard reset signal. Resets the RX interface, including the RX PCS and RX MAC. This reset leads to the deassertion of the o_rx_pcs_ready output signal.
i_sl_csr_rst_n i_sl_csr_rst_n[n-1:0] i_csr_rst_n	Active-low hard global reset. Resets the full IP core.

continued...

Signal	Description
	Resets the TX MAC, RX MAC, TX PCS, RX PCS, transceivers (transceiver reconfiguration registers and interface), and Ethernet reconfiguration registers. This reset leads to the deassertion of the `o_tx_lanes_stable` and `o_rx_pcs_ready` output signals.
`i_reconfig_reset`	Resets the E-Tile Hard IP for Ethernet Intel FPGA IP core Avalon memory-mapped interfaces, both the transceiver reconfiguration interface and the Ethernet reconfiguration interface and some Ethernet soft registers.. This signal is synchronous with the `i_reconfig_clk` clock.

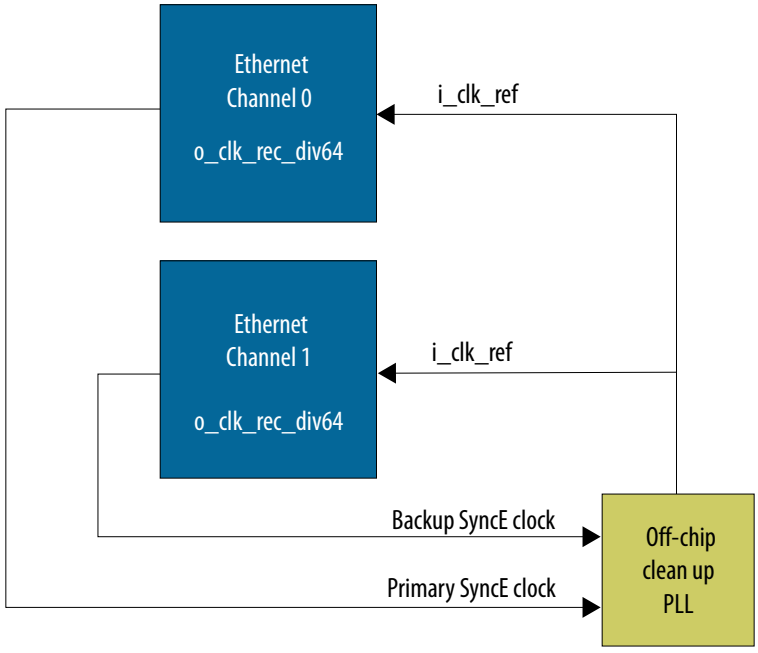
2.11.17. Clocks

You must set the transceiver reference clock (`i_clk_ref`) frequency to a value that the IP supports.

The Synchronous Ethernet standard, described in the ITU-T G.8261, G.8262, and G.8264 recommendations, requires that the TX clock be filtered to maintain synchronization with the RX reference clock through a sequence of nodes. The expected usage is that user logic drives the transceiver reference clocks with a filtered version of the RX recovered clock signal, to ensure the receive and transmit functions remain synchronized. In this usage model, a design component outside the E-Tile Hard IP for Ethernet Intel FPGA IP performs the filtering.

An alternate clocking arrangement for `i_clk_ref` can be used to enable the Synchronous Ethernet (SyncE) operation. Two or more channels can share the Off-chip Cleanup PLL clock output. The Primary SyncE clock and the Backup SyncE clock come from the recovered clock output pins of channels connected to the same SyncE network while `i_clk_ref` connects to the cleanup PLL. SyncE clocking can be also combined with the data path clocking.

Figure 55. Clock Connection in SyncE Operation



SyncE mode is supported when you turn on **Enable SyncE** for single/multiple 10G and 25G Ethernet channels. When SyncE is enabled, **Enable external AIB clocking** and **Enable RS-FEC** options are still available but **Enable Custom Rate** option is disabled as custom cadence connection is done internally.

When SyncE is enabled, TX elastic FIFO (eFIFO) is instantiated within the IP. The number of TX eFIFO generated is based on the number of 10G/25G Ethernet channels. The write clock of the eFIFO is `o_clk_pll_div66` from TX PMA while the read clock is `i_sl_clk_tx` input clock and the FIFO has a depth of 32. The write data valid signal of the FIFO is connected to the custom cadence port of the hard IP, hence the write data valid is driven high for 32 write clocks and low for 1 clock cycle.

Table 58. Clock Inputs

Describes the input clocks that you must provide.

Signal Name	Description
`i_sl_clk_tx`	This clock drives both, the TX datapath and TX interface, for 10G/25G channel. • Frequency of 402.83203125 MHz for all modes on a 25G channel. Also applicable for 10G channels when overclocked or when PTP is enabled. • Frequency of 161.1328125 MHz for all modes on a 10G channel except for enabled PTP. This clock must be active during dynamic reconfiguration. *Note:* Applicable only when you select **Single 10GE/25GE**.
`i_sl_clk_rx`	This clock drives both, the RX datapath and TX interface, for 10/25G channel. • Frequency of 402.83203125 MHz for all modes on a 25G channel. Also applicable for 10G channels when overclocked or when PTP is enabled. • Frequency of 161.1328125 MHz for all modes on a 10G channel except for enabled PTP. This clock must be active during dynamic reconfiguration. *Note:* Applicable only when you select **Single 10GE/25GE**.
`i_sl_clk_tx[n]`	These clocks drive the active TX datapath and TX interface for 10/25G channel when Asynchronous mode is disabled. These clocks drive the active TX datapath for 10/25G channel when Asynchronous mode is enabled while `i_sl_async_clk_tx` clocks the TX interface. Each channel has its own clock input. • Frequency of 402.83203125 MHz for all modes on a 25G channel. Also applicable for 10G channels when overclocked or when PTP is enabled. • Frequency of 161.1328125 MHz for all modes on a 10G channel except for enabled PTP. This clock must be active during dynamic reconfiguration. *Note:* Applicable only when you select **1 to 4 10GE/25GE with optional RS-FEC** or **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP**.
`i_sl_clk_rx[n]`	These clocks drive the active RX datapath and RX interface for 10/25G channel when Asynchronous mode is disabled. These clocks drive the active RX datapath for 10/25G channel when Asynchronous mode is enabled while `i_sl_async_clk_rx` clocks the RX interface.. Each channel has its own clock input. • Frequency of 402.83203125 MHz for all modes on a 25G channel. Also applicable for 10G channels when overclocked or when PTP is enabled. • Frequency of 161.1328125 MHz for all modes on a 10G channel except for enabled PTP. This clock must be active during dynamic reconfiguration. *Note:* Applicable only when you select **1 to 4 10GE/25GE with optional RS-FEC** or **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP**.
`i_clk_tx`	This clock drives the TX interface for 100G channel.

continued...

Signal Name	Description
	The frequency of this clock is 402.83203125 MHz for all modes on a 100G channel RS-FEC(544,514) modes, where the frequency is 415.0390625 MHz. *Note:* Applicable only when you select **Single 100GE with optional RS-FEC** or **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP**.
i_clk_rx	This clock drives the RX interface for 100G channel. The frequency of this clock is 402.83203125 MHz for all modes on a 100G channel RS-FEC(544,514) modes, where the frequency is 415.0390625 MHz. *Note:* Applicable only when you select **Single 100GE with optional RS-FEC** or **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP**.
i_clk_ref	The input clock i_clk_ref is the reference clock for the high-speed serial clocks and the datapath parallel clocks. This clock must have the following frequencies with a ±100 ppm accuracy per the *IEEE 802.3-2015 Ethernet Standard*: • 156.25 MHz (10G/25G/100G) • 322.265625 MHz (10G/25G/100G) • 312.500000 MHz (100G) • 644.531250 MHz (100G) Variants with (544,514) RSFEC option only support 156.25 MHz and 312.5 MHz PHY i_clk_ref reference frequency. The reference clock must be at 156.25 MHz frequency to support Auto Negotiation and Link Training. In addition, i_clk_ref must meet the jitter specification of the *IEEE 802.3-2015 Ethernet Standard* The PLL and clock generation logic use this reference clock to derive the transceiver and PCS clocks. The input clock should be a high quality signal on the appropriate dedicated clock pin. Refer to the *Intel Stratix 10 Device Data Sheet* or *Intel Agilex Device Data Sheet* for transceiver reference clock phase noise specifications. When using this clock for Synchronous Ethernet, the expected usage is that this signal being driven by a filtered and divided version of o_clk_rec_div64 or o_clk_rec_div66, to ensure the receive and transmit functions remain synchronized. Therefore, you must include an additional component on your board. The IP core does not provide filtering. *Note:* Applicable only when you select **Single 100GE with optional RS-FEC** or **100GE or 1 to 4 10GE/25GE with optional RS-FEC and 1588 PTP**.
i_reconfig_clk	Avalon clock for the E-Tile Hard IP for Ethernet Intel FPGA IP transceiver reconfiguration interface, RS-FEC reconfiguration interface, PTP reconfiguration interface, and Ethernet reconfiguration interface. The clock frequency is 100-125 MHz. All reconfiguration interface signals are synchronous to i_reconfig_clk. *Note:* When you turn on **Enable AN/LT** or select any 10G/25G channel variant in the **Select Core Variant**, i_reconfig_clk is set to 500 MHz in simulation to accelerate IP's simulation.
i_aib_clk	This clock is used for all internal datapath provided externally by user. This clock must be driven by a clock running at the fastest line rate in the design divided by 64 and must be frequency locked to i_aib_2x_clk clock. Phase offset between these two clocks are allowed. *Note:* Applicable only when you select **Enable external AIB clocking** parameter in 10/25GE variants.
i_aib_2x_clk	This clock must have double the frequency of i_aib_clk clock and is provided externally by user. It is used for clock crossing handling in EMIB interface. This clock must be frequency locked to i_aib_clk clock. Phase offset between these two clocks are allowed. *Note:* Applicable only when you select **Enable external AIB clocking** parameter in 10/25GE variants.
i_sl_async_clk_tx[n]	This clock drives the TX interface for 25G channel when Asynchronous mode is enabled. The clock frequency must be within 390.625 MHz to 402.83203125 MHz.

continued...

Signal Name	Description
	Note: Applicable only when you select **Enable asynchronous adapter clocks** parameter.
i_sl_async_clk_rx[n]	This clock drives the RX interface for 25G channel when Asynchronous mode is enabled. The clock frequency must be within 390.625 MHz to 402.83203125 MHz. *Note:* Applicable only when you select **Enable asynchronous adapter clocks** parameter.
i_sl_clk_tx_tod[n-1:0]	TX interface Time Of Day (ToD) Clock. The o_clk_pll_div66[n-1:0] must drive this clock. The clock frequency varies based on the Ethernet variant: • 156.25 MHz (10G) • 390.625 MHz (25G) *Note:* Applicable only when you set the **PTP Accuracy Mode** parameter to **Advanced Mode**. For more information, refer to the *PTP Timestamp Accuracy* section.
i_sl_clk_rx_tod[n-1:0]	RX interface Time Of Day (ToD) Clock. The o_clk_rec_div66[n-1:0] must drive this clock. The clock frequency varies based on the Ethernet variant: • 156.25 MHz (10G) • 390.625 MHz (25G) *Note:* Applicable only when you set the **PTP Accuracy Mode** parameter to **Advanced Mode**. For more information, refer to the *PTP Timestamp Accuracy* section.
i_clk_ptp_sample	Sample clock for PTP measurement. This is an external clock provided to the design with frequency of 114.285714 MHz (with required period of 8.75 ns) with ± 100 ppm. *Note:* Applicable only when you set the **PTP Accuracy Mode** parameter to **Advanced Mode**. For more information, refer to the *PTP Timestamp Accuracy* section.

Table 59. Clock Outputs

Describes the output clocks that the IP core provides. In most cases these clocks participate in internal clocking of the IP core as well.

Signal Name	Description
o_clk_pll_div64[n]	Hard IP for Ethernet block clock. Supports the following clock frequencies: • 402.83203125 MHz for 25G and 100G with optional RS-FEC(528,514) channels • 402.83203125 MHz for 10G PTP and 25G PTP channels • 415.0390625 MHz for 100G with RS-FEC(544,514) channel • 161.1328125 MHz for 10G channels This clock is reliable only after o_tx_pll_locked is asserted.
o_clk_pll_div66[n]	Hard IP for Ethernet block clock times 64/66. Supports the following clock frequencies: • 390.625 MHz for 25G and 100G with optional RS-FEC(528,514) channels • 402.4621 MHz for 100G with RS-FEC(544,514) • 156.25 MHz for 10G channels This clock is reliable only after o_tx_pll_locked is asserted.
o_clk_rec_div64[n]	Derived from RX recovered clock. This clock supports the SyncE standard. The RX recovered clock frequency is:

continued...

Signal Name	Description
	• 161.1328125 MHz ±100 ppm for 10G channels • 402.83203125 MHz ±100 ppm for 25G channels • 402.83203125 MHz ±100 ppm for 100G with optional RS-FEC(528,514) channels • 415.0390625 MHz ±100 ppm for 100G with RS-FEC(544,514) channels This clock is reliable only after `o_cdr_lock[n]` is asserted. When using this clock for Synchronous Ethernet, the expected usage is that you drive the TX transceiver reference clock with a filtered and divided version of `o_clk_rec_div64` or `o_clk_rec_div66`, to ensure the receive and transmit functions remain synchronized. To do so you must include an additional component on your board. The IP core does not provide filtering. *Note:* The RX recovered clock is not available for PTP channels when PTP enabled.
`o_clk_rec_div66[ch]`	Derived from RX recovered clock. This clock supports the Synchronous Ethernet standard. The RX recovered clock frequency is: • 156.25 MHz ±100 ppm for 10G channels • 390.625 MHz ±100 ppm for 25G channels • 390.625 MHz ±100 ppm for 100G with optional RS-FEC(528,514) channels • 402.4621 MHz ±100 ppm for 100G with optional RS-FEC(528,514) channels This clock is reliable only after `o_cdr_lock[n]` is asserted. When using this clock for Synchronous Ethernet, the expected usage is that you drive the TX transceiver PLL reference clock with a filtered and divided version of `o_clk_rec_div64` or `o_clk_rec_div66`, to ensure the receive and transmit functions remain synchronized. To do so you must include an additional component on your board. The IP core does not provide filtering. *Note:* The RX recovered clock is not available for PTP channels when PTP enabled.

Related Information

- Intel Stratix 10 Device Data Sheet
  Provides transceiver reference clock phase noise specifications.
- Intel Agilex Device Data Sheet

2.11.17.1. Asynchronous Adapter Clock in 10G/25G Mode

When enabling asynchronous adapter clocks, you may clock the TX/RX interface in TX MAC Interface to User Logic on page 121 using the `i_sl_async_clk_tx/rx` clock asynchronous to the `i_sl_sync_clk_tx/rx` signals used in the internal IP datapath.

Note: In 10G mode, the asynchronous adapter clock is available only when PTP is enabled.

(31) When Asynchronous mode is disabled, `i_sl_clk_tx/rx` signals drive both, TX/RX interface and TX/RX datapath in 25G mode.

Figure 56. Clock Connection in Asynchronous FIFO Operation

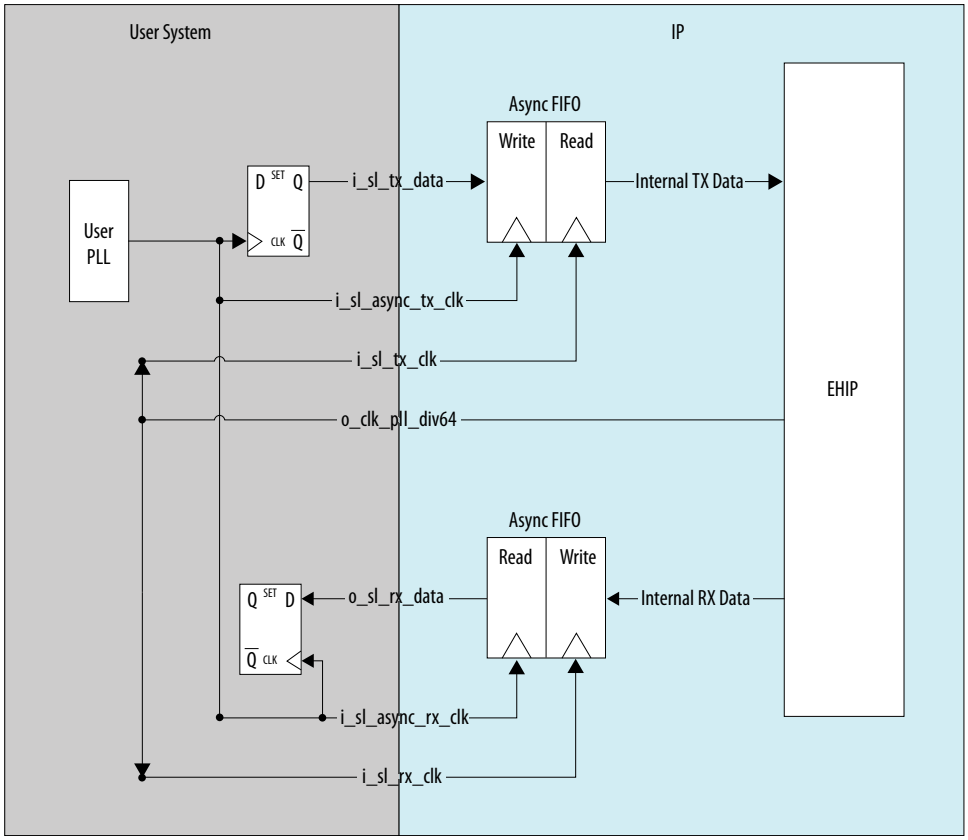


Table 60. Supported Clock Rates for MAC Client Asynchronous FIFO Operation in 25G Mode

The below rates assume 1 byte IPG and disabled preamble-pass-through.

Rate	Clock Rate			
	Min `i_sl_async_clk_tx`	Max `i_sl_async_clk_tx`	Min `i_sl_async_clk_rx`	Max `i_sl_async_clk_rx`
10G[32]/25G	390.625 MHz	402.83203215 MHz	390.625 MHz	402.83203215 MHz

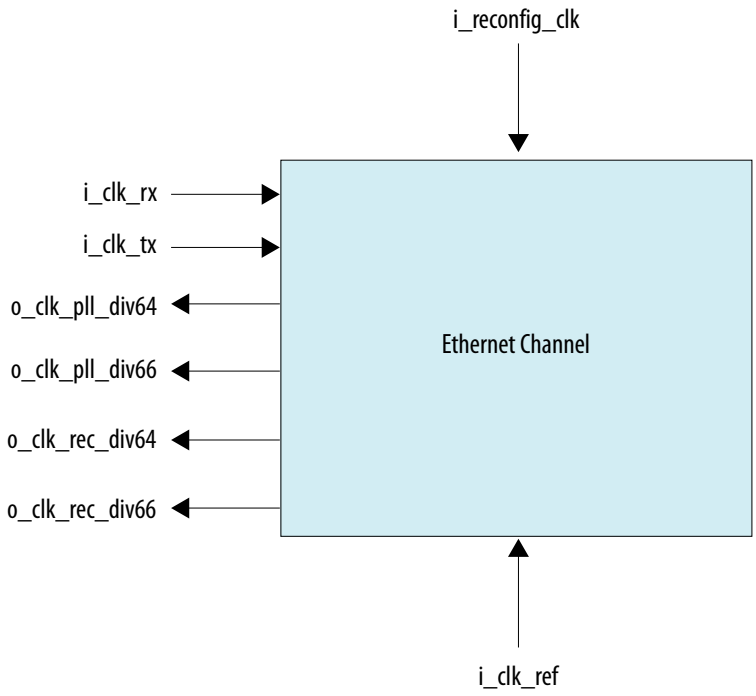
2.11.17.2. Asynchronous Adapter Clock in 100G Mode

When **Enable asynchronous adapter clocks** is enabled, `i_clk_rx` and `i_clk_tx` can be asynchronous from each other and from `o_clk_pll_div64` clock as long as the clocks are fast enough to ensure all data is processed by a channel.

Note: For 100G mode, the asynchronous adapter clocks are only available when PTP is disabled.

(32) In 10G mode, the asynchronous adapter clock is available only when PTP is enabled.

Figure 57. Clock Connection in Asynchronous FIFO Operation



Below table summarizes minimum and maximum frequencies required for `i_clk_rx` and `i_clk_tx` during the Asynchronous mode.

Table 61. Supported Clock Rates for MAC Client Asynchronous FIFO Operation in 100G Mode

The below rates assume 1 byte IPG and disabled preamble-pass-through.

Rate	Clock Rate			
	Min `i_clk_tx`	Max `i_clk_tx`	Min `i_clk_rx`	Max `i_clk_rx`
100G	340 MHz	420 MHz	340 MHz + 200 ppm	420 MHz + 200 ppm

2.11.17.3. Clock Network Use Cases

These use cases provide guidance about how you can connect various clocks through the GUI for different use cases.

2.11.17.3.1. Single 25G Ethernet Channel (with FEC)

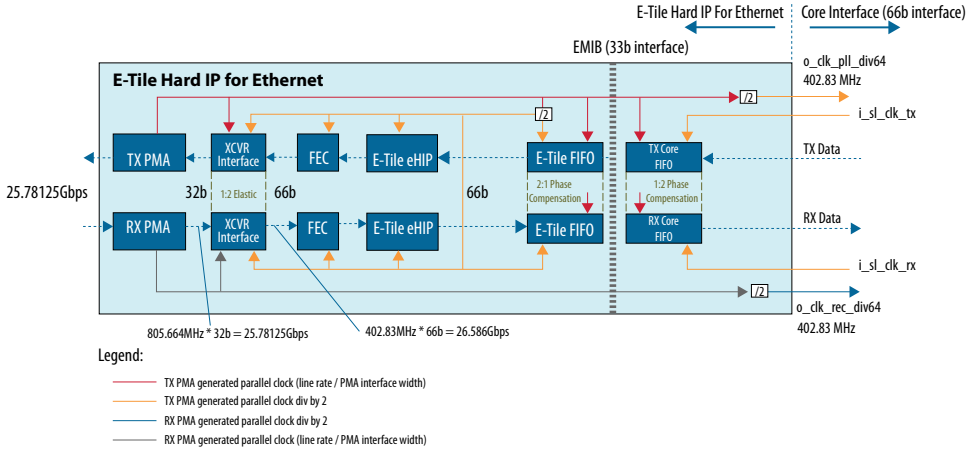
Table 62. Use Case Configuration

Data Rate	Core Interface
25.78125 Gbps	64 bits

Connect `o_clk_pll_div64` (402.83MHz) to the `i_sl_clk_tx` and `i_sl_clk_rx`. If you use any other source for `i_sl_clk_tx` or `i_sl_clk_rx`, make sure that `i_sl_clk_tx` and `i_sl_clk_rx` have 0 PPM difference with respect to `o_clk_pll_div64`.

Figure 58. Ethernet 25G x 1

RX FEC is also clocked by the TX PMA generated clock.



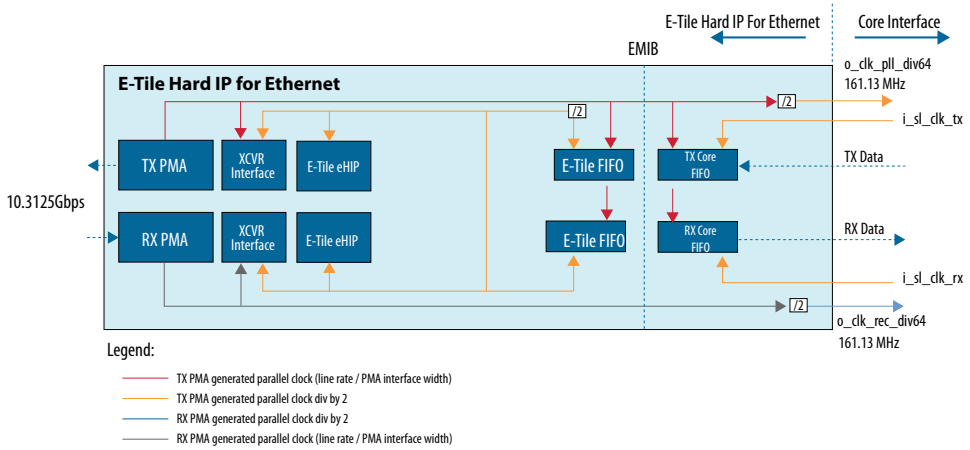
2.11.17.3.2. Single 10G Ethernet Channel (without FEC)

Table 63. Use Case Configuration

Data Rate	Core Interface
10.3125 Gbps	64 bits

Connect `o_clk_pll_div64` (161.13MHz) to the `i_sl_clk_tx` and `i_sl_clk_rx`. If you use any other source for `i_sl_clk_tx` or `i_sl_clk_rx`, make sure `i_sl_clk_tx` and `i_sl_clk_rx` have 0 PPM difference with respect to `o_clk_pll_div64`.

Figure 59. Ethernet 10G x 1



2.11.17.3.3. Four 25G Ethernet Channels (with FEC) within a Single FEC Block

Table 64. Use Case Configuration

Data Rate per Channel	Number of Channels	Core Interface
25.78125 Gbps	4	64 bits

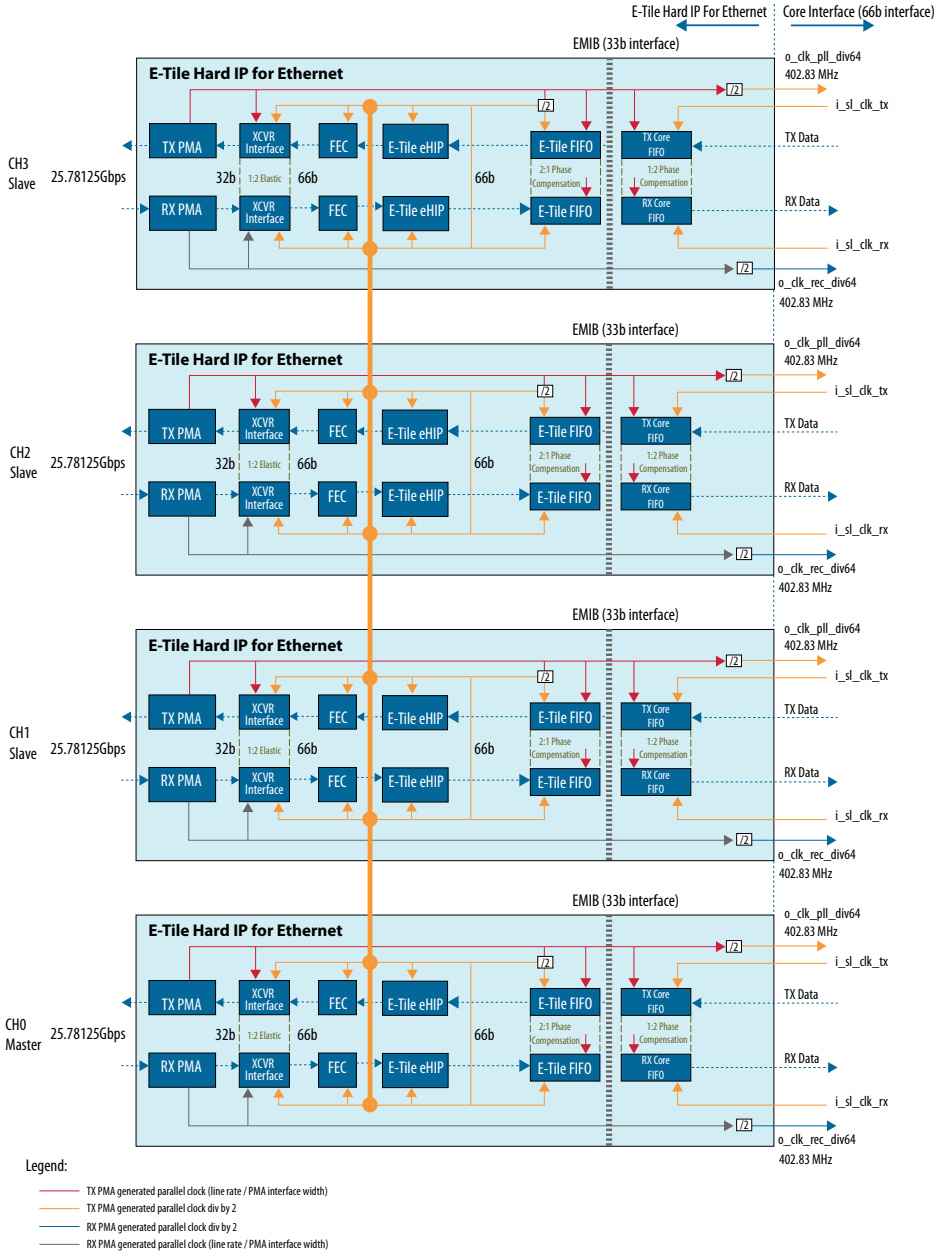
Master-Slave Configuration: Option 1

All four channels use a common FEC block but FEC will use only one clock from the 4 available channels. The channel that provides the FEC clock is considered as a master. The other 3 channels use that same clock for clocking their TX and RX data path, and are considered as slave channels. Any interruption on master channel PMA, a PMA reset, for example, impacts the slave channels. This creates a dependency between the master and the slave channels.

Connect `o_clk_pll_div64` (402.83MHz) to the `i_sl_clk_tx` and `i_sl_clk_rx`. If you use any other source for `i_sl_clk_tx` or `i_sl_clk_rx`, make sure `i_sl_clk_tx` and `i_sl_clk_rx` have 0 PPM difference with the `o_clk_pll_div64`.

Figure 60. Ethernet 25G x 4 (FEC On) Master-Slave Configuration Option 1

RX FEC is also clocked by the TX PMA generated clock.



Master-Slave Configuration: Option 2 - External AIB Clocking Scheme

In this configuration, you can select to import the TX and RX datapath clocks and EMIB clock from an external source outside of the targeted transceiver channels. Enable this by selecting the checkbox **Enable External AIB Clocking** from IP Parameter Editor. An extra input port is exposed in the transceiver channel core interface to drive the individual EMIB clock for each 25 Gbps channel. The FEC clock is still provided by the Master channel. The Stratix 10 E-Tile Transceiver Native PHY Intel

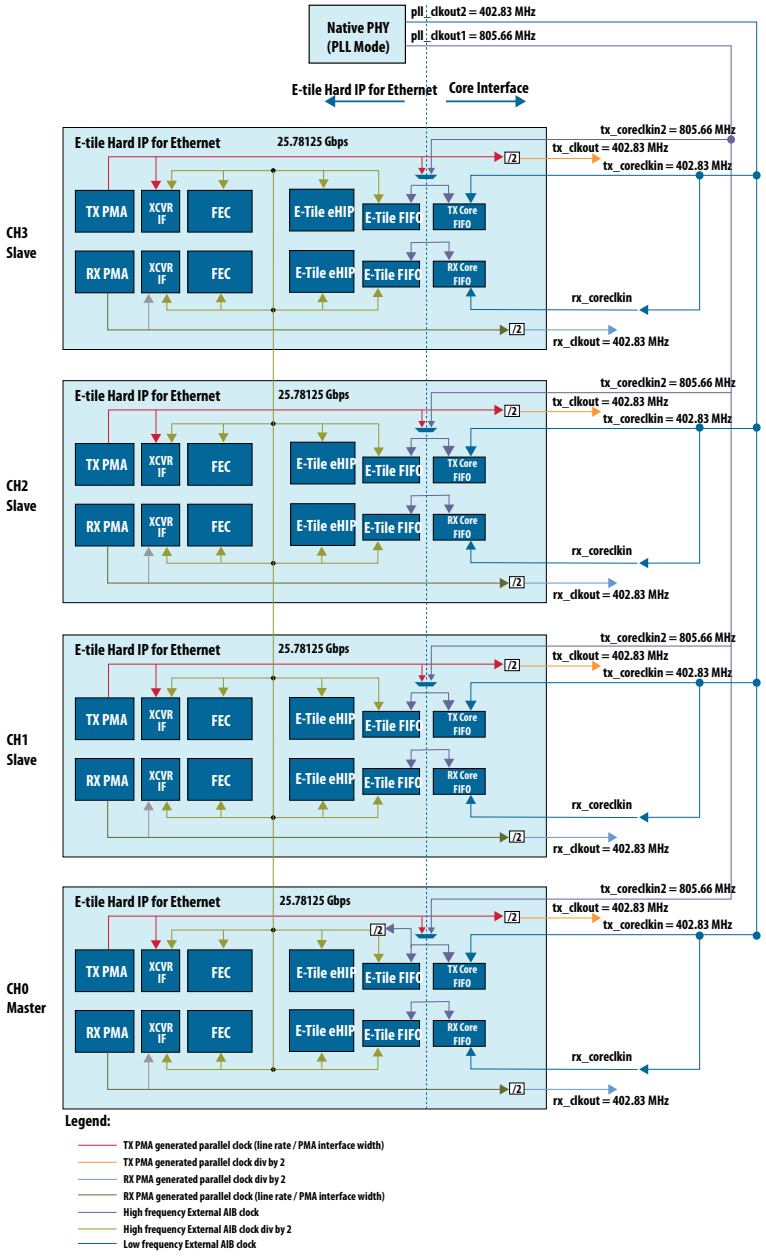
FPGA IP in PLL mode acts as the external source to provide clock to transceiver channel. Before resetting the transceiver channel, you must read the `o_tx_pll_locked` output from the PLL channel:

- Wait until `o_tx_pll_locked` output from the PLL channel is asserted before deasserting the transceiver channel reset at power-up.
- If `o_tx_pll_locked` from the PLL channel is deasserted at any time, hold the respective transceiver channel in reset until `o_tx_pll_locked` is reasserted.

The PLL Channel and the Ethernet channel should have the same reference clock source.

The following figure shows one master 25 Gbps channel providing the datapath clock to other three slave 25 Gbps channels. This method removes the dependency of a PMA reset between the Master and Slave channels.

Figure 61. Ethernet 25G x 4 (FEC On) Master-Slave Configuration: External AIB Clocking

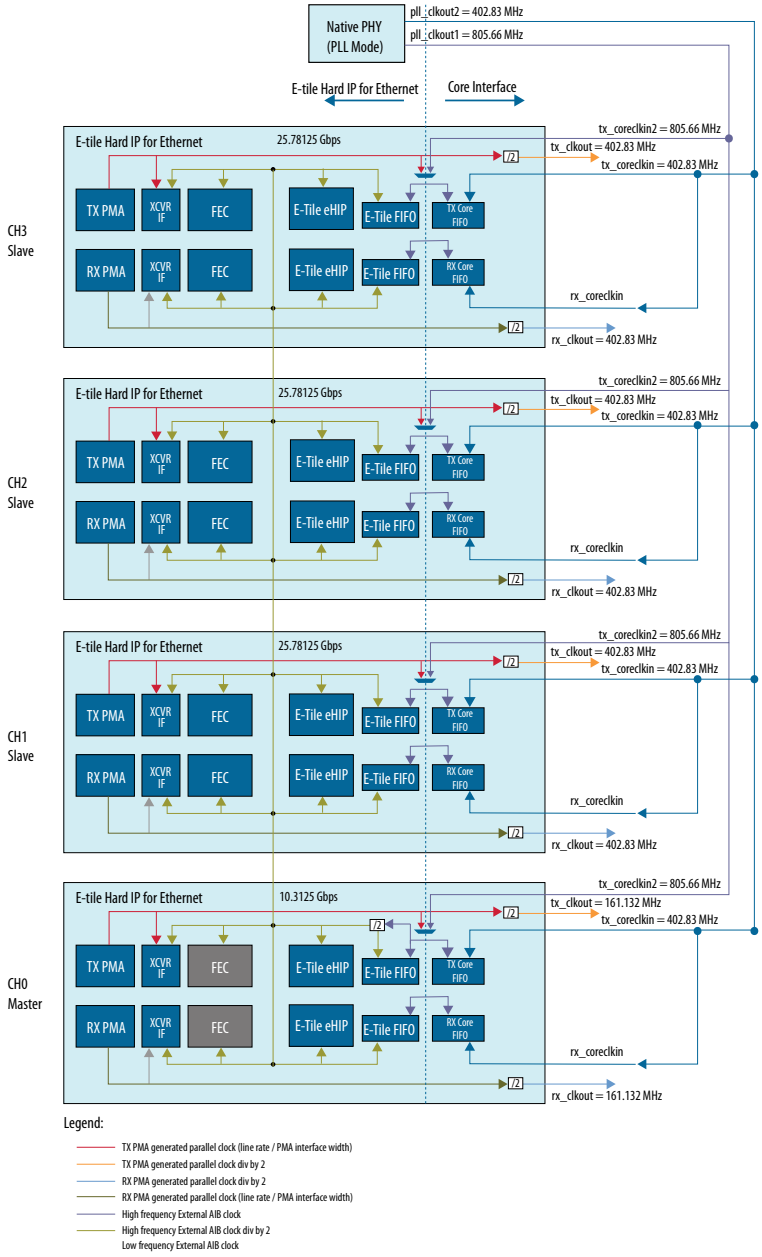


Master-Slave Configuration: Option 3 - Dynamic Reconfiguration

In this configuration, you can dynamically reconfigure the Master Channel 0 from 25G to 10G. This configuration uses four 25G Ethernet channels with RS-FEC enabled at power on. Channel 0 is considered the Master channel and channel 1 ~ channel 3 are considered Slave channels. After power on, the 25G Master Channel with RS-FEC enable reconfigures from 25.78125 Gbps to 10.3125 Gbps.

Below figure is an example of possible dynamic reconfiguration on Master channel. You can reconfigure the master channel to any mode without bringing down the slave channel functionality with the exception of the direct PMA.

Figure 62. Ethernet 25G x 4 (FEC On) Master-Slave Configuration: Dynamic Reconfiguration



For more information on the dynamic reconfiguration examples, refer to the *Dynamic Reconfiguration Design Example User Guide*.

Related Information

- E-Tile Hard IP Intel Stratix 10 Design Example User Guide
  Information about the Dynamic Reconfiguration.
- E-Tile Hard IP Intel Agilex Design Example User Guide
  Information about the Dynamic Reconfiguration.

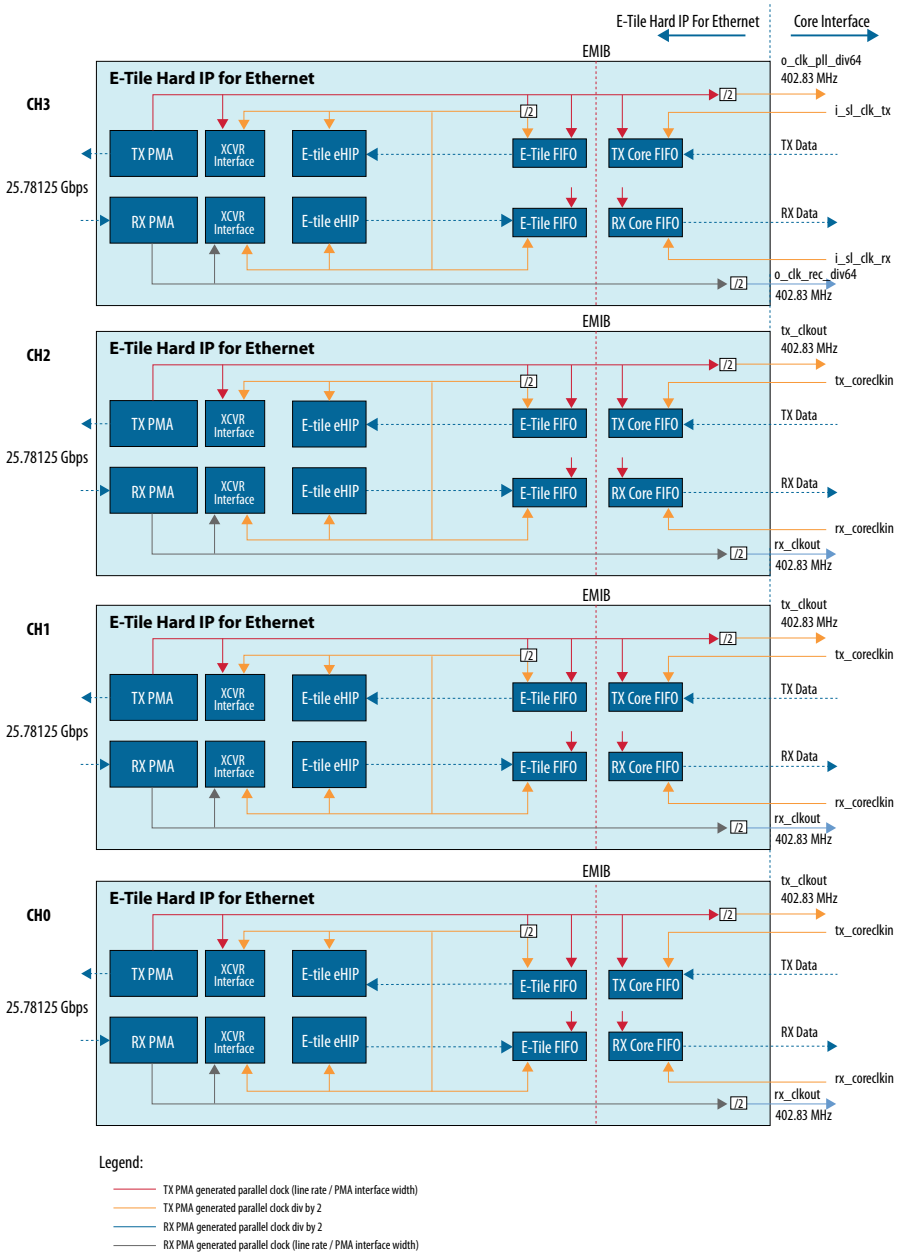
2.11.17.3.4. Ethernet 25G x 4 (FEC Off)

This use case does not include FEC; therefore, there is no need for clock sharing between the four 25G Ethernet channels. Connect `o_clk_pll_div64` (402.83 MHz) to `i_sl_clk_tx` and `i_sl_clk_rx`. Due to the timing constraint, `i_sl_clk_tx` and `i_sl_clk_rx` channels can only be assigned to channel 0, channel 1, or channel 2.

Note: Due to the clock assignment dependency, if clock arrives from other than a master channel, the clock's appropriate channel impacts all other channels.

If you use any other source for `i_sl_clk_tx` or `i_sl_clk_rx`, make sure `i_sl_clk_tx` and `i_sl_clk_rx` have 0 PPM difference with the `o_clk_pll_div64`.

Figure 63. Ethernet 25G x 4 (FEC Off)



2.11.17.3.5. 10G/25G Ethernet Channel with Basic PTP Accuracy Mode

When PTP is enabled, the external AIB clocking is inherently enabled. Do not enable **External AIB clocking** parameter in the Parameter Editor, it is unnecessary and redundant.

Table 65. Use Case Configuration

Number of Ethernet Channels	Data Rate	Core Interface	External AIB Clocking
2	25.78125 Gbps	64 bits	Disabled

This use case covers a scenario when PTP is enabled and the **PTP Accuracy Mode** is set to **Basic Mode**.

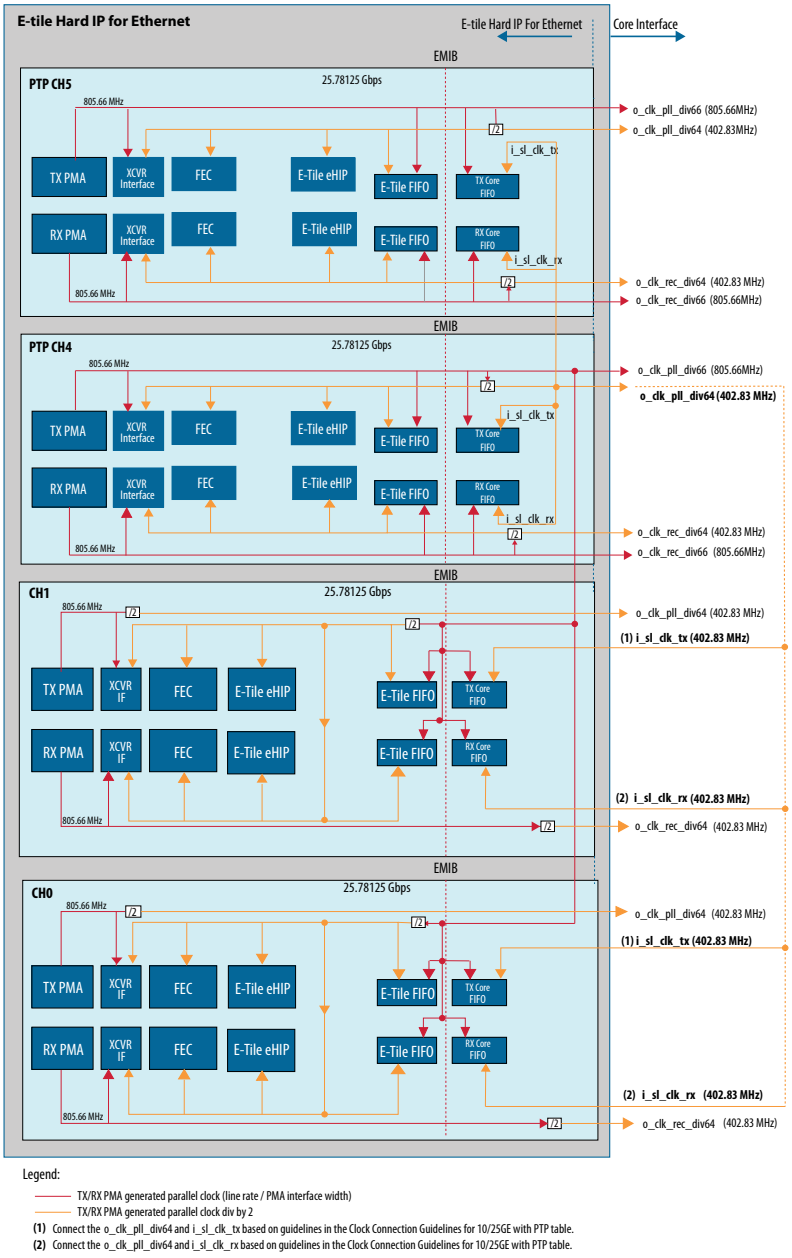
With PTP enabled, a PTP channel and its source clock called PTP clock becomes the master channel regardless of FEC configuration.

Connect `o_clk_pll_div64[number of channel]` (402.83MHz) to the `i_sl_clk_tx` and `i_sl_clk_rx` of each Ethernet channel based on the following guidelines:

Table 66. Clock Connection Guidelines for 10GE/25GE with enabled PTP and Basic PTP Accuracy Mode

Number of Channels of 10G/25G	Clock Port	PTP Clock	Clock Connection Guideline
Single channel	`i_sl_clk_tx` `i_sl_clk_rx`	`o_clk_pll_div64[1]`	Connect `o_clk_pll_div64[1]` to `i_sl_clk_tx` and `i_sl_clk_rx`.
2 channels	`i_sl_clk_tx[1:0]` `i_sl_clk_rx[1:0]`	`o_clk_pll_div64[2]`	Connect `o_clk_pll_div64[2]` to `i_sl_clk_tx[1:0]` and `i_sl_clk_rx[1:0]`.
3 channels	`i_sl_clk_tx[2:0]` `i_sl_clk_rx[2:0]`	`o_clk_pll_div64[3]`	Connect `o_clk_pll_div64[3]` to `i_sl_clk_tx[2:0]` and `i_sl_clk_rx[2:0]`.
4 channels	`i_sl_clk_tx[3:0]` `i_sl_clk_rx[3:0]`	`o_clk_pll_div64[4]`	Connect `o_clk_pll_div64[4]` to `i_sl_clk_tx[3:0]` and `i_sl_clk_rx[3:0]`.

Figure 64. Ethernet 10/25G with Basic PTP Accuracy Mode



2.11.17.3.6. 10G/25G Ethernet Channel with Advanced PTP Accuracy Mode

When PTP is enabled, the external AIB clocking is inherently enabled. Do not enable **External AIB clocking** parameter in the Parameter Editor, it is unnecessary and redundant.

Table 67. Use Case Configuration

Number of Ethernet Channels	Data Rate	Core Interface	External AIB Clocking
2	25.78125 Gbps	64 bits	Disabled

This use case covers a scenario when PTP is enabled and the **PTP Accuracy Mode** is set to **Advanced Mode**.

With PTP enabled, a PTP channel and its source clock called PTP clock becomes the master channel regardless of FEC configuration.

Connect `o_clk_pll_div66[number of channel]` to the `i_sl_clk_tod[number or channels]`, and `o_clk_rec_div66[number of channels]` to the `i_sl_clk_rx_tod[number of channels]` of each Ethernet channel based on the following guidelines:

Table 68. Clock Connection Guidelines for 10GE/25GE with enabled PTP and Advanced PTP Accuracy Mode

Number of Channels of 10G/25G	Clock Port	PTP Clock	Clock Connection Guideline

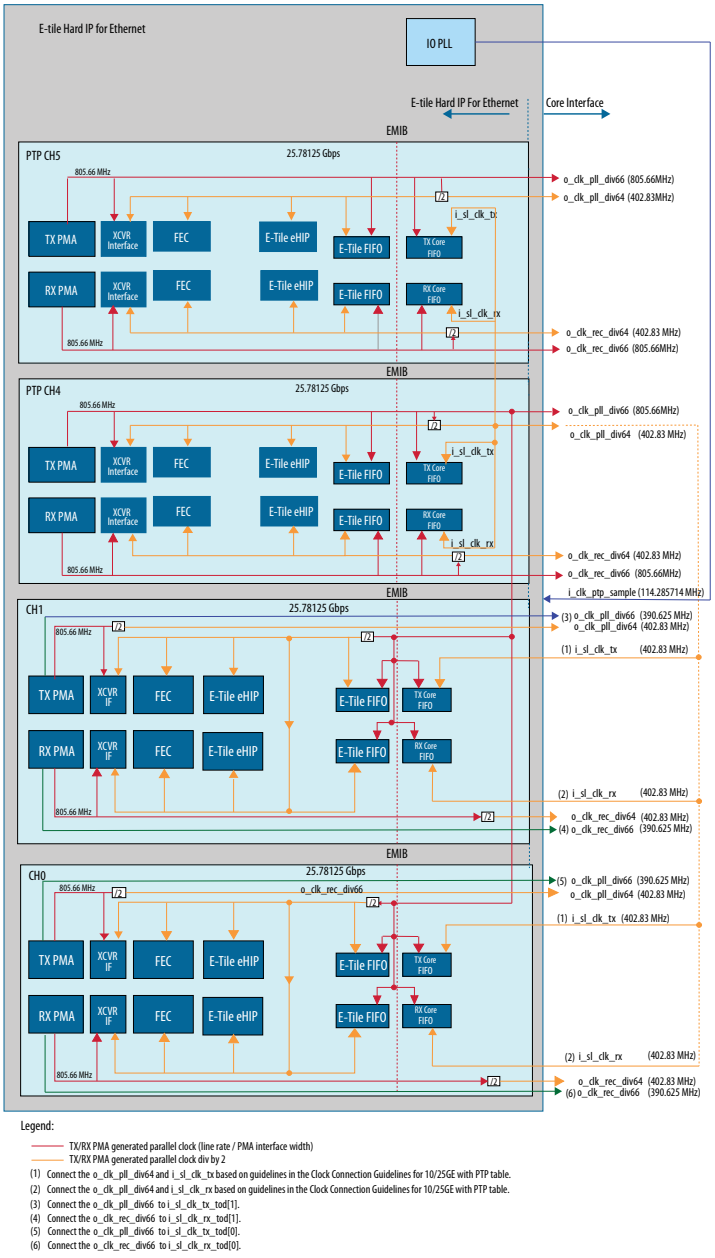
Single channel	`i_sl_clk_tx`
	`i_sl_clk_rx`
	`i_sl_clk_tx_tod`
	`i_sl_clk_rx_tod`
		`o_clk_pll_div64[1]`	Connect `o_clk_pll_div64[1]` to `i_sl_clk_tx` and `i_sl_clk_rx`.
			Connect `o_clk_pll_div66` to `i_sl_clk_tx_tod`.
			Connect `o_clk_rec_div66` to `i_sl_clk_rx_tod`.

2 channels	`i_sl_clk_tx[1:0]`
	`i_sl_clk_rx[1:0]`
	`i_sl_clk_tx_tod[1:0]`
	`i_sl_clk_rx_tod[1:0]`
		`o_clk_pll_div64[2]`	Connect `o_clk_pll_div64[2]` to `i_sl_clk_tx[1:0]` and `i_sl_clk_rx[1:0]`.
			Connect `o_clk_pll_div66[1:0]` to `i_sl_clk_tx_tod[1:0]`.
			Connect `o_clk_rec_div66[1:0]` to `i_sl_clk_rx_tod[1:0]`.

3 channels	`i_sl_clk_tx[2:0]`
	`i_sl_clk_rx[2:0]`
	`i_sl_clk_tx_tod[2:0]`
	`i_sl_clk_rx_tod[2:0]`
		`o_clk_pll_div64[3]`	Connect `o_clk_pll_div64[3]` to `i_sl_clk_tx[2:0]` and `i_sl_clk_rx[2:0]`.
			Connect `o_clk_pll_div66[2:0]` to `i_sl_clk_tx_tod[2:0]`.
			Connect `o_clk_rec_div66[2:0]` to `i_sl_clk_rx_tod[2:0]`.

4 channels	`i_sl_clk_tx[2:0]`
	`i_sl_clk_rx[2:0]`
	`i_sl_clk_tx_tod[3:0]`
	`i_sl_clk_rx_tod[3:0]`
		`o_clk_pll_div64[4]`	Connect `o_clk_pll_div64[4]` to `i_sl_clk_tx[3:0]` and `i_sl_clk_rx[3:0]`.
			Connect `o_clk_pll_div66[3:0]` to `i_sl_clk_tx_tod[3:0]`.
			Connect `o_clk_rec_div66[3:0]` to `i_sl_clk_rx_tod[3:0]`.

Figure 65. Ethernet 25G with Advanced PTP Accuracy Mode



In addition, the advanced mode requires an IOPLL to provide a clock frequency of 114.285714 MHz to connect to `i_clk_ptp_sample`. The IOPLL must be a stable, free running clock with the reference frequency meeting the IOPLL reference clock requirement as specified in the *Intel Stratix 10 Clocking and PLL User Guide*. You can also use any existing IOPLL in the your design to generate this required clock frequency.

Table 69. Recommended Connection Guidelines for IOPLL Signals

IOPLL Signals	Description
refclk	Any system clock that meets the reference clock requirement. The default frequency is 100 MHz. The minimum and maximum frequency value depends on the selected device.
rst	Global/system reset
locked	Logic reset
outclk_0	i_clk_ptp_sample

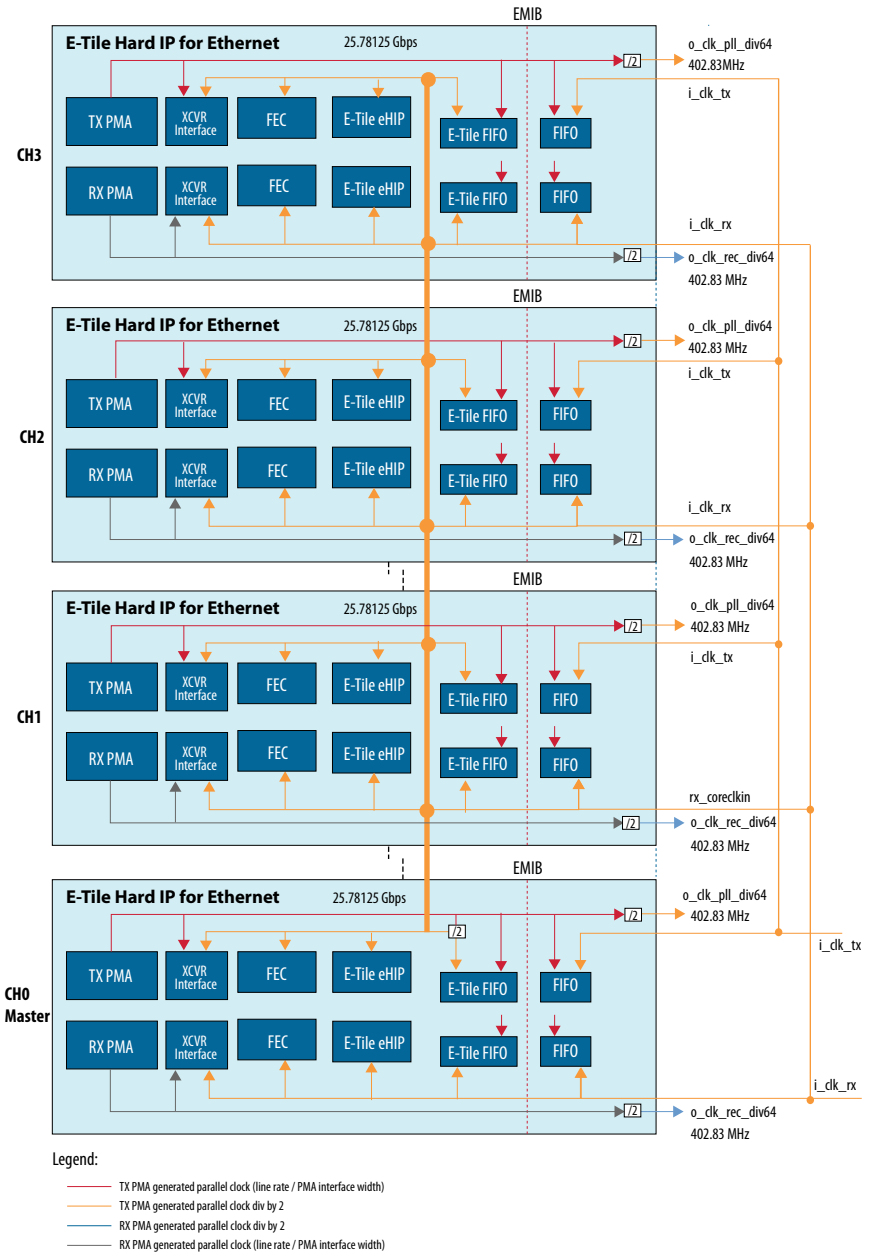
Related Information

- Intel Stratix 10 Clocking and PLL User Guide
  Information on Intel Stratix 10 Clocking.
- Intel Agilex Clocking and PLL User Guide
  Information on Intel Agilex Clocking.

2.11.17.3.7. 100G Ethernet with Aggregated FEC

This use case is implemented in the case of multi-lane protocols like 100GbE, for example. This uses four transceiver lanes of 25 Gbps each, where all four lanes use the same FEC block. There is an inherent dependency between channels in this configuration as described in *Master-Slave Configuration: Option 1* section. However, for applications like 100 GbE, dependency is acceptable and sometimes required.

Figure 66. 100G Ethernet with Aggregated FEC



2.11.17.3.8. 100G Ethernet with PTP

Connect `o_clk_pll_div64[4]` (402.83MHz) to the `i_clk_tx` and `i_clk_rx` of each Ethernet channel based on the following guidelines:

Table 70. Clock Connection Guidelines for 100GbE with enabled PTP and Basic PTP Accuracy Mode

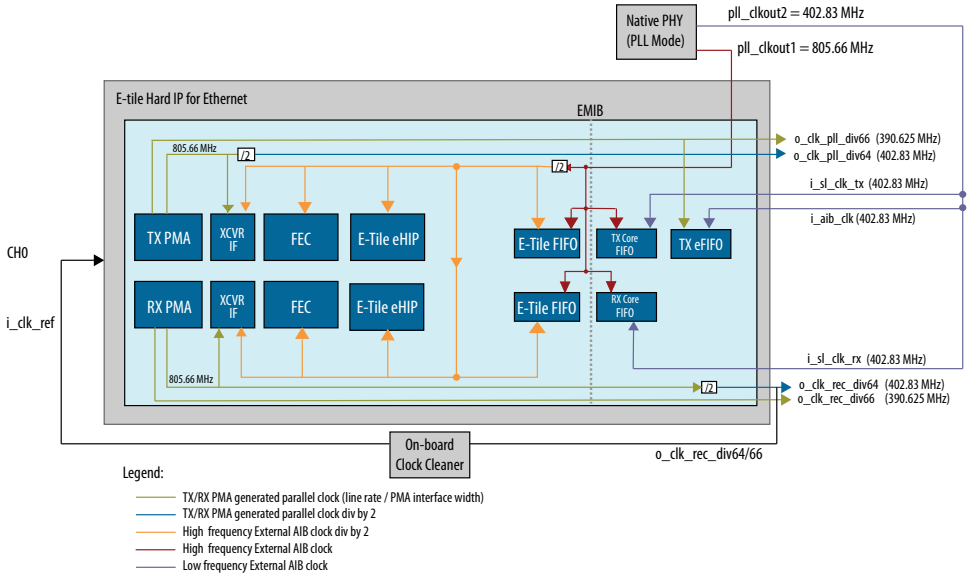
Clock Port	PTP Clock	Clock Connection Guideline
i_clk_tx i_clk_rx	o_clk_pll_div64[4]	Connect o_clk_pll_div64[4] to i_clk_tx and i_clk_rx.

2.11.17.3.9. Single 25G Synchronous Ethernet Channel

External AIB Clocking Scheme

When **Enable External AIB Clocking** is enabled, external clock source can be used to drive TX and RX datapath. In this configuration, the Intel Stratix 10 10 E-Tile Transceiver Native PHY Intel FPGA IP in PLL mode can act as Channel PLL to drive i_aib_clk, i_sl_clk_tx and i_sl_clk_rx input clocks. The RX recovered clock (o_clk_rec_div66 or o_clk_rec_div64) can be fed to on-board clock cleaner. The filtered recovered clock should be connected to transceiver reference clock (i_clk_ref). The clock cleaner should also be configured to generate correct clock frequency for i_clk_ref.

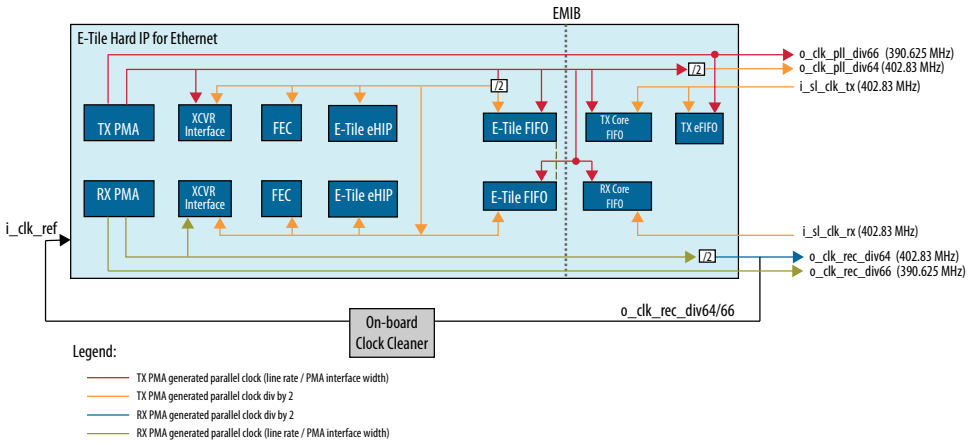
Figure 67. Single Channel 25G SyncE with External AIB Clocking (FEC On)



Without External AIB Clocking Scheme

When **Enable External AIB Clocking** is disabled, i_sl_clk_tx and i_sl_clk_rx input clocks should be connected to output clock o_clk_pll_div64 (402.83MHz). Similarly, connect filtered and divided version of RX recovered clock (o_clk_rec_div64 or o_clk_rec_div66) to i_clk_ref.

Figure 68. Single Channel 25G SyncE without External AIB Clocking (FEC On)



2.11.17.3.10. Multiple 25G Synchronous Ethernet Channels

For multi-channel synchronous Ethernet configuration, one TX eFIFO is instantiated for each Ethernet channel. The write data valid signal of the FIFO is connected internally to the custom cadence port of hard IP.

Similar to single channel configuration, RX recovered clock of each channel is fed to on-board clock cleaner for jitter removal. The clock cleaner should filter and generate the correct frequency for `i_clk_ref` input reference clock. After power up, the clock cleaner has to provide default reference clock before RX recovered clock is available at `o_clk_rec_div64/66`.

There are two ways to connect filtered and divided RX recovered clock to `i_clk_ref`:

1. Dedicated reference clock: Recovered clock from each channel is fed to input reference clock of respective channel. For example, connect filtered RX clock from Channel 0 to `i_clk_ref` of Channel 0, filtered RX clock from Channel 1 to `i_clk_ref` of Channel 1.
2. Common reference clock: Only one of the RX recovered clocks to be connected to input reference clock of all channels. For example, connect filtered RX clock from Channel 0 to `i_clk_ref` of Channel 0 and Channel 1. Make sure that all channels have the same input reference clock frequency.

Figure 69. 25G x4 (FEC On) SyncE Without External AIB Clocking

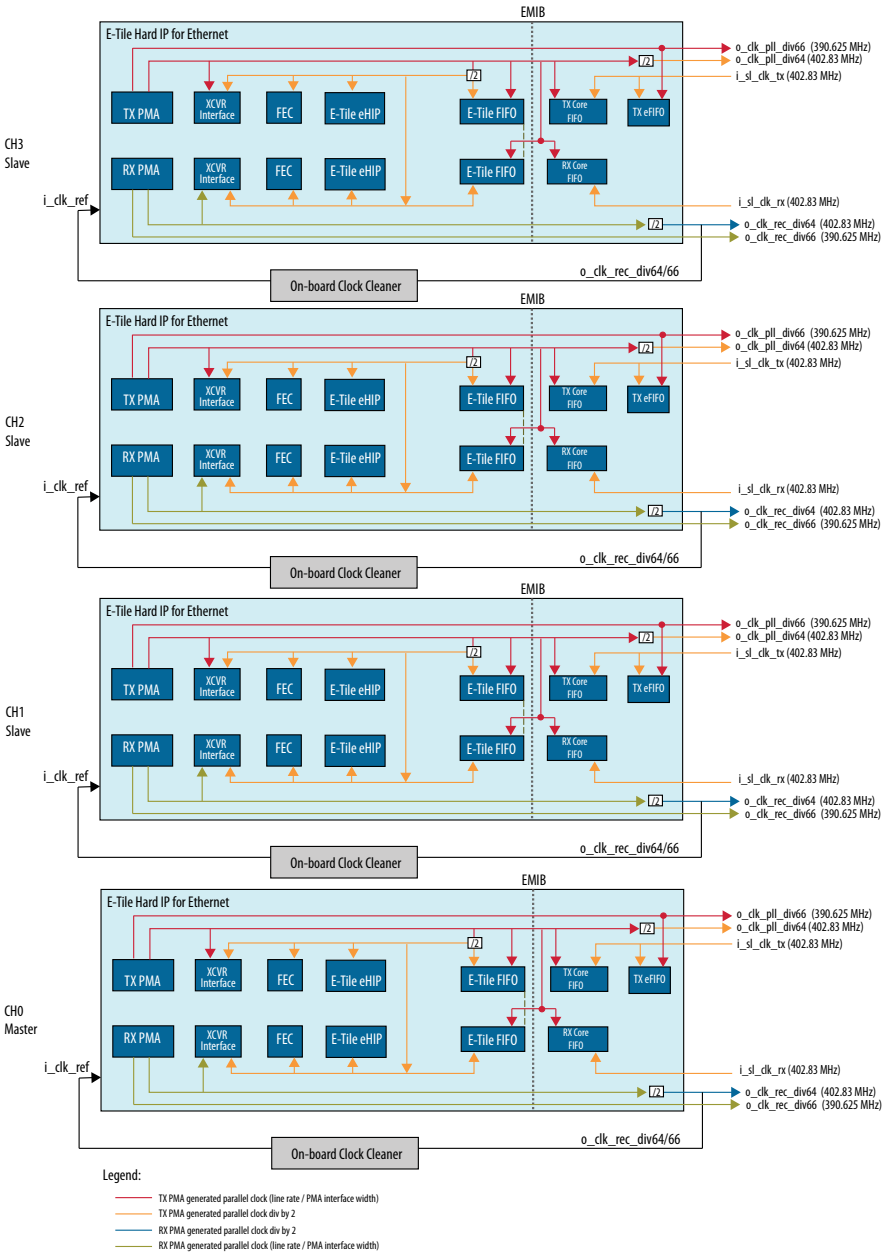


Figure 70. 25G x4 (FEC On) SyncE With External AIB Clocking, Dedicated Reference Clock

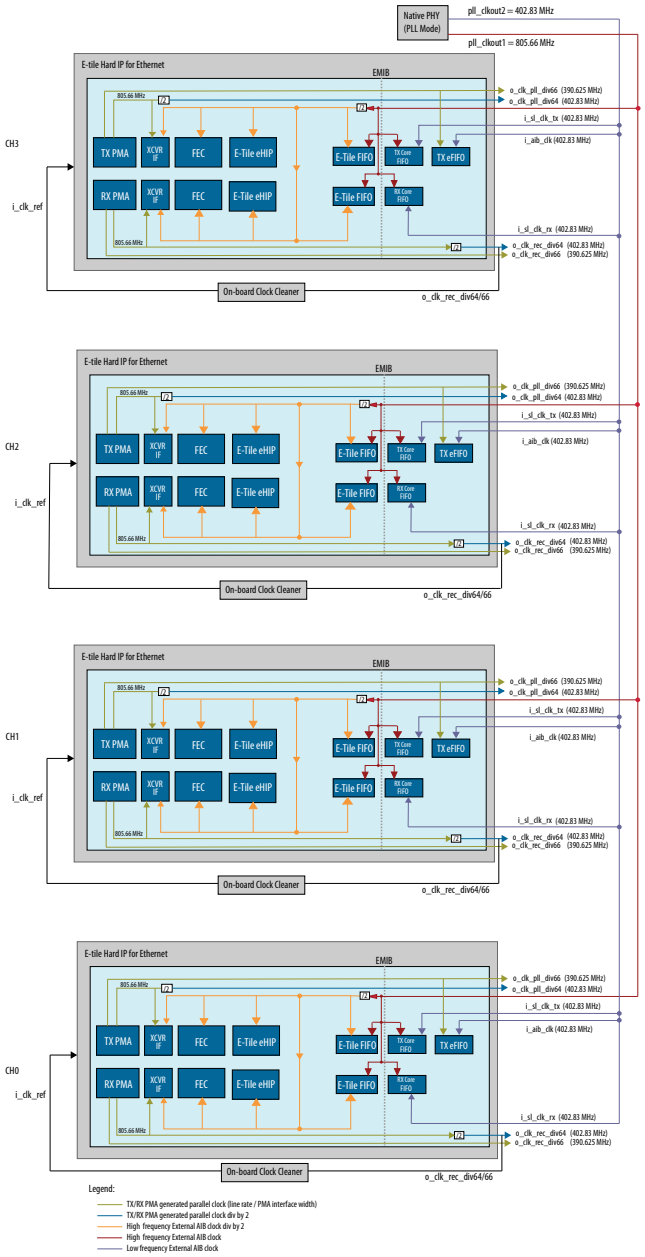
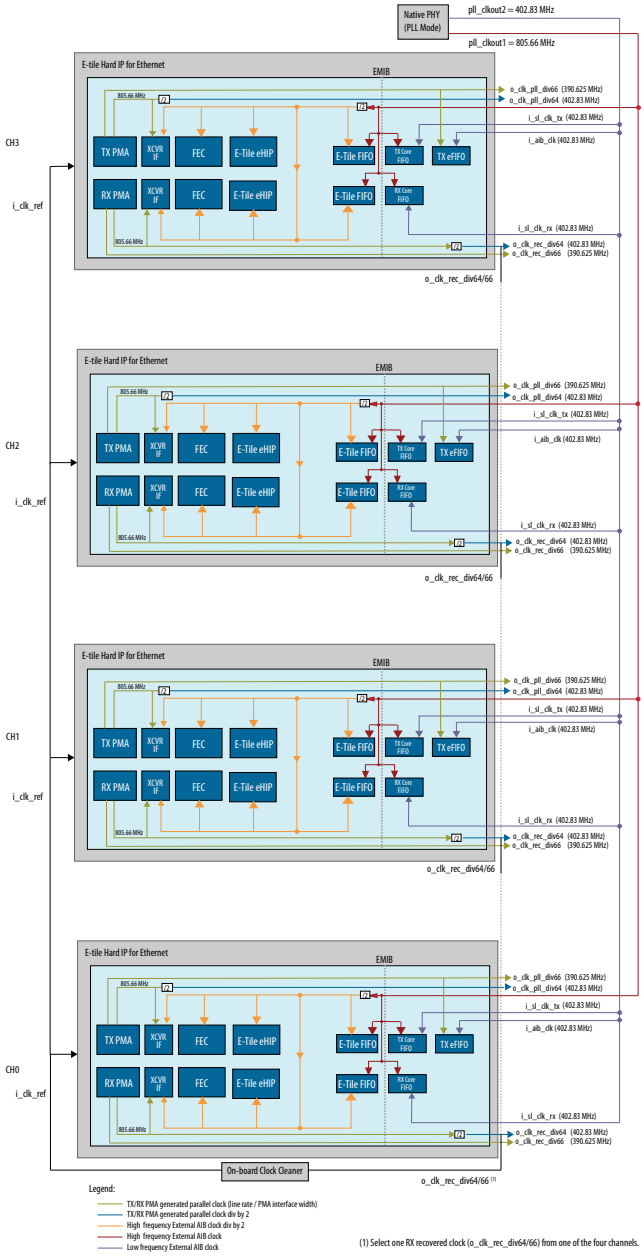


Figure 71. 25G x4 (FEC On) SyncE With External AIB Clocking, Shared Reference Clock



2.12. Register Descriptions

You access the Ethernet registers for the E-Tile Hard IP for Ethernet Intel FPGA IP using the Avalon memory-mapped Ethernet reconfiguration interface on each channel. The TX and RX RS-FEC registers are accessible through the RS-FEC reconfiguration interface.

Write operations to a read-only register field have no effect. Read operations that address a Reserved register return an unspecified result. Write operations to Reserved registers have no effect. Accesses to registers that do not exist in your IP core variation, or to register bits that are not defined in your IP core variation, have an unspecified result. You should consider these registers and register bits Reserved. Although you can only access registers in 32-bit read and write operations, you should not attempt to write or ascribe meaning to values in undefined register bits.

These registers use 32-bit addresses; they are not byte addressable.

Table 71. Register Base Addresses

The reset values in register type sections represent register values after a reset has completed.

Word Offset	Register Type
0x0B0-0x0E8	Auto Negotiation and Link Training registers
0x300-0x3FF	PHY registers
0x400-0x4FF	TX MAC registers
0x500-0x5FF	RX MAC registers
0x600-0x7FF	Pause and Priority- Based Flow Control registers
0x800-0x8FF	TX Statistics Counter registers
0x900-0x9FF	RX Statistics Counter registers
0xA00-0xAFF	TX 1588 PTP registers
0xB00-0xBFF	RX 1588 PTP registers

Table 72. RS-FEC Register Base Addresses

Word Offset	Register Type
0x000-0x2FF	TX and RX RS-FEC registers

Table 73. Transceiver Register Base Addresses

PTP transceiver channel register is accessed through the PTP reconfiguration interface. Data transceiver channel register is accessed through the Transceiver reconfiguration interface.

Word Offset	Register Type
0x40000-0x40144	PMA Capability registers
0x000-0x207	PMA AVMM registers

Note: Do not attempt to access any register address that is Reserved or undefined. Accesses to registers that do not exist in your IP core variation have unspecified results.

2.12.1. Auto Negotiation and Link Training Registers

2.12.1.1. ANLT Sequencer Config

Offset: 0xB0

ANLT Sequencer Config Fields

Bit	Name	Description	Access	Reset
31	kr_pause	**Pauses ANLT Function** 1: Pauses ANLT function when kr_paused bit is high. 0: Normal ANLT function Set this bit before accessing PMA registers via the transceiver reconfiguration interface to ensure no conflict with the ANLT function.	RW	0x0
29:26	anlt_seq_cfg_rxinv	**RX Polarity Inversion for Lane 0 to Lane 3** Sets RX Polarity Inversion on lanes 3:0 (for 100G NRZ), lanes 2 and 0 (for 100G PAM4) or current lane (for 25/10G). • [29] = Inverts RX PMA polarity on lane 3 • [28] = Inverts RX PMA polarity on lane 2 • [27] = Inverts RX PMA polarity on lane 1 • [26] = Inverts RX PMA polarity on lane 0	RW	0x0
25:22	anlt_seq_cfg_txinv	**TX Polarity Inversion for Lane 0 to Lane 3** Sets TX Polarity Inversion on lanes 3:0 (for 100G NRZ), lanes 2 and 0 (for 100G PAM4) or current lane (for 25/10G). • [25] = Inverts TX PMA polarity on lane 3 • [24] = Inverts TX PMA polarity on lane 2 • [23] = Inverts TX PMA polarity on lane 1 • [22] = Inverts TX PMA polarity on lane 0 Setting takes effect upon KR restart	RW	0x0
21	rsfec_request	**Request RS-FEC mode during AN** 1: Request RS-FEC mode during AN 0: Do not request RS-FEC during AN • Defaults to 1 if parameter REQUEST_RSFEC is set to 1 • This feature is new in E-Tile	RW	0x0
20	rsfec_capable	**Enables RS-FEC Negotiation** 1: Enable RS-FEC negotiation 0: Do not negotiate for RS-FEC Defaults to 1 if parameter ENABLE_RSFEC is set to 1	RW	0x0
19:16	anlt_seq_cfg_ilpbk	**Internal Loopback for Lane 0 to Lane 3** Sets internal loopback mode on lanes 3:0 (for 100G NRZ), lanes 0 and 2 (for 100G PAM4) or current lane (for 25/10G). • [16] = Internal loopback for lane 0 • [17] = Internal loopback for lane 1 • [18] = Internal loopback for lane 2 • [19] = Internal loopback for lane 3 Loopback takes effect upon KR restart.	RW	0x0
14	skip_lt_on_an_timeout	**Skip Link Training on AutoNegotiation Timeout** 1: If AN times out skip LT before attempting data mode, and use the previous LT settings 0: Use the normal ANLT sequence, even if link_fail_if_hiber = 0 • This option is provided to speed up re-lock times when the link is known not to be resetting due to problems with link integrity	RW	0x0
13	link_fail_if_hiber	**Link Fail if HiBER** 1: Trigger a link failure if a HiBER condition is detected in the PCS during data mode (default)	RW	0x1

continued...

Bit	Name	Description	Access	Reset
		0: Ignore HiBER		
12	lt_failure_response	**Link Training Failure Response** 1: Upon LT failure, PHY will go to data mode 0: Upon LT failure, PHY will restart AN, or if AN is disabled, skip AN and restart LT	RW	0x0
7:4	seq_force_mode	**Force the sequencer into a specific protocol, ignoring autonegotiation result** [6:4] = 3'b000: None [6:4] = 3'b001: 25G-R1 [6:4] = 3'b010: Reserved [6:4] = 3'b011: 100G-R4 [6:4] = 3'b100: Reserved [6:4] = 3'b101: 10G-R1 [6:4] = 3'b110: Reserved [6:4] = 3'b111: 100G-P2 [7] = 1'b1: Force RS-FEC on if capable (never for 10G, always for 100G-P2) • Forces the ANLT Sequencer into a specific protocol, ignoring the AN result • ANLT will still be cycled if enabled; configure AN and LT using their respective CFG registers *Note:* Not all protocols are available in all configurations. You must enable the protocols when generating the IP to be functional at run-time.	RW	0x0
2	disable_lf_timer	**Disable Link Fail Inhibit Timer** 1: Disable the link fail inhibit timer 0: If PCS link fails, then AN will restart • The most common reason to disable the link fail inhibit timer is to characterize the link's behavior with link training • Turning off the link fail inhibit timer prevents link training from cycling, allowing each failure to be examined individually • Disabling the LFI Timer also disable transitioning from data mode to auto-negotiation phase when the link status goes down, even if the LFI timer would have expired. Effectively the system stays in Data Mode state until reset or user intervention.	RW	0x0
1	disable_an_timer	**Disable Auto-Negotiation Timer** Enable this bit to allow operation with link partners that do not support auto-negotiation. 1: AN will wait for valid partner without timing out (default). Auto-negotiation timeout is set to approximately 1 second. 0: If AN fails, the Sequencer will try a different protocol	RW	0x1
0	reset_seq	**Reset ANLT Sequencer** 1: Reset only the ANLT Sequencer. Initiates a PCS reconfiguration and/or ANLT reset 0: Normal operation This bit is self-cleared when the ANLT sequence restarts.	RW	0x0

2.12.1.2. ANLT Sequencer Status

Offset: 0xB1

ANLT Sequencer Status Fields

Bit	Name	Description	Access	Reset
31	kr_paused	**Indicates ANLT Function is paused due to kr_pause bit** 1: ANLT Function Paused 0: Normal ANLT Function		
18	rsfec_ability	**Indicates local RS-FEC support** 1: RS-FEC supported 0: RS-FEC not supported Defaults to 1 if parameter ENABLE_RSFEC is set to 1	RO	0x0
15:8	seq_reconfig_mode	**Sequencer mode for PCS reconfiguration** [8] = AN mode [9] = LT mode (Clause 93) [10] = 10G data mode [11] = 25G data mode [12] = Reserved [13] = 100G-R4 data mode [14] = Reserved [15] = 100G-P2 data mode All other settings reserved. • The sequencer modifies the datapath as required to move through the stages of ANLT • This status register lets you know which step is in progress, and how the datapath is configured	RO	0x0
2	seq_lt_timeout	**Sequencer Link Training Timeout** 1: Sequencer had LT Timeout 0: No timeout occurred This status bit is sticky, and stays high until the next time LT restarts.	RO	0x0
1	seq_an_timeout	**Sequencer AutoNegotiation Timeout** 1: Sequencer had AN Timeout 0: No timeout occurred This status bit is sticky, and stays high until the next time AN restarts.	RO	0x0
0	seq_link_ready	**Sequencer Link Ready** 1: Link is ready for data mode 0: Link not ready	RO	0x0

2.12.1.3. Auto Negotiation Config Register 1

Offset: 0xC0

Auto Negotiation Config Register 1 Fields

Bit	Name	Description	Access	Reset
31:16	consortium_oui	**Consortium Organizationally Unique Identifier (OUI) (lower 16 bits)** Sets the lower bits of the OUI (as defined in IEEE 802.3 Annex 73A) used in sending and receiving Next Pages.	RW	0x737D
11	ignore_consortium_next_page_tech_ability_code	**Ignore Consortium NextPage Tech Ability Code** 1: AN function accepts any unformatted Next Page after a formatted Next Page tagged with the proper OUI for resolving Consortium AN modes	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		0: The AN function only accepts an unformatted Next Page with the code 0x003 in bits D8:D0		
10	enable_consortium_next_page_override	**Enable Consortium Next Page override** 1: Data sent to the consortium Next Page comes from the *Consortium Next Page Override* Register (0xCD) instead of being set automatically 0: Data is filled based on supported IP modes	RW	0x0
9	enable_consortium_next_page_receive	**Enable Consortium Next Page receive** 1: Enable decoding received Consortium Next Pages for purpose of resolving AN 0: The AN function ignores the received next pages	RW	0x1
8	enable_consortium_next_page_send	**Enable Consortium Next Page send** Send Consortium-standard next pages immediately after the base page. If User next pages are enabled, consortium pages is sent after the last user next page.	RW	0x0
7	ignore_nonce_field	**Ignore Nonce Field** 1: Ignore the Nonce field during AN 0: Normal operation • AN will normally fail in loopback due to the Nonce field • To use AN with loopback, disable Nonce bit checking using this feature The default value for this bit in synthesis is 0 and in simulation is 1.	RW	0x0
6	override_an_chan_enable	**Override AN Master Channel (as set by AN_CHAN)** 1: Use AN Master channel selection from 0xCC[1:0] 0: Use AN Master channel specified via generation parameter	RW	0x0
5	override_an_parameters_enable	**Override AN Parameters** 1: Use the bits from parameter override CSRs to compose the default base page 0: Normal operation	RW	0x0
3	local_device_remote_fault	**Force Local device remote fault** 1: Signal a remote fault using appropriate bit in the AN pages 0: Normal operation	RW	0x0
2	an_next_pages_ctrl	**Enable User Controlled AN Next Pages** 1: User-controlled next pages are enabled. You can send any arbitrary data via the User next page high/low bits (0xC6[31:0]/0xC5[15:0]). The IP pauses on each next page transaction until you set 0xC1[8]. 0: User-controlled next page control is disabled. The IP generates the null message to send as next pages, and does not pause on next page transactions. Set this bit before AN begins, and clear this bit once your next page activity completes to allow AN to finish.	RW	0x0
1	an_base_pages_ctrl	**Enable User Controlled AN Base Pages** To ensure proper HCD resolution when changing the Tech Ability and FEC fields, set the equivalent bits in the parameter override CSR through override_an_parameters_enable bit. 1: User controlled base pages are enabled; the User Base page CSRs control the base page used for AN	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		0: The AN logic will automatically generate base pages based on the Ethernet Core Variant and its parameters • Enable this feature if you need to control the content of the AN Base page • Leave this feature disabled if you want the core to perform default negotiation for its type		
0	enable_an	**Enable AutoNegotiation** 1: Enable AutoNegotiation (default) 0: Disable AutoNegotiation • Equivalent to state variable mr_autoneg_enable in IEEE 802.3 CL73.10.1	RW	0x1

2.12.1.4. Auto Negotiation Config Register 2

Offset: 0xC1

Auto Negotiation Config Register 2 Fields

Bit	Name	Description	Access	Reset
23:16	consortium_oui_upper	**Consortium Organizationally Unique Identifier (OUI) (upper 8b)** Sets the upper bits of the OUI (as defined in IEEE 802.3 Annex 73A) used in sending and receiving Next Pages.	RW	0x6A
8	an_next_page	**AN Next Page** 1: New user Next Page data to send is loaded in 0xC5-0xC6 and finished reading LP next page data from 0xC9-0xCA. Only available if 0xC0[2] is set to 1. If there are no more Next Pages to send, but you still wishes to read LP Next Pages, you must set 0xC5-0xC6 to the IEEE null message as defined in 73A.1 (bit D0=1, all other bits=0) before setting this bit. To determine if the LP has more Next Pages to send, view the NP bit (D15) of the LP base page or previous LP Next Page. After the IP has sent the user Next Page data and loaded the new LP Next Page data to 0xC9-0xCA (if available), this bit will self-clear (0 = LP next page data ready to read)	RW	0x0
0	reset_an	**Reset all AN state machines** 1: Reset all the AN state machines 0: Normal operation Maps to state variable mr_main_reset in IEEE 802.3 CL 73.10.1. This bit is self-cleared when auto-negotiation restarts.	RW	0x0

2.12.1.5. Auto Negotiation Status Register

Offset: 0xC2

Auto Negotiation Status Register Fields

Bit	Name	Description	Access	Reset
30	rs_fec_negotiated	**RS-FEC Negotiated** Indicates AN negotiated RS-FEC operation. 1: Link uses RS-FEC	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		0: Link doesn't use RS-FEC		
27:24	consortium_negotiated_port_type	**Consortium negotiated Port Type** Indicates the negotiated HCD port type for Consortium modes. [24] = 25GBASE-KR1 [25] = 25GBASE-CR1 [26] = 50GBASE-KR2 [27] = 50GBASE-CR2	RO	0x0
23:12	ieee_negotiated_port_type	**IEEE Negotiated Port Type** Indicates the negotiated HCD port type for IEEE modes. [12] = 1000BASE-KX [13] = 10GBASE-KX4 [14] = 10GBASE-KR [15] = 40GBASE-KR4 [16] = 40GBASE-CR4 [17] = 100GBASE-CR10 [18] = 100GBASE-KP4 [19] =100GBASE-KR4 [20] = 100GBASE-CR4 [21] = 25GBASE-KR-S/CR-S [22] = 25GBASE-KR/CR [23] = 100GBASE-KR2/CR2	RO	0x0
11	negotiation_failure	**AN complete, but unable to resolve PHY** 1: AN completed, but was unable to find a Highest Common Denominator rate, or a common FEC 0: Normal operation	RO	0x0
10	consortium_next_page_received	**Consortium Next Page received** 1: Consortium Next Page identified from a link partner 0: No Consortium Next page found	RO	0x0
7	an_lp_ability	**Link Partner Auto Negotiation Ability** 1: Link Partner is able to perform AN 0: Link Partner is not able to perform AN	RO	0x0
6	an_status	**Auto Negotiation Status** 1: Link is up 0: Link is down	RO	0x0
5	an_ability	**PHY Auto Negotiation Ability** 1: PHY is able to perform AN 0: PHY is not able to perform AN • This bit is tied high when AN module is included in the Ethernet core, low otherwise	RO	0x0
3	an_adv_remote_fault	**Auto Negotiation ADV Remote Fault** 1: Fault information sent to link partner 0: Normal operation • Remote Fault is encoded in bit D13 of the Base link codeword • See IEEE 802.3 CL 73.6.7 for more information • See mr_adv_ability in CL 73.10.1	RO	0x0
2	an_complete	**Auto Negotiation Complete** 1: AN Complete	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		0: AN in progress • Corresponds to state variable mr_autoneg_complete in CL 73.10.1		
1	an_page_received	**AN Page Received** 1: A page has been received 0: No page received • Corresponds to state variable mr_page_rx in IEEE 802.3 Cl 73.10.1	RO	0x0

2.12.1.6. Auto Negotiation Config Register 3

Offset: 0xC3

Auto Negotiation Config Register 3 Fields

Bit	Name	Description	Access	Reset
30:28	override_an_pause	**AN_PAUSE Override Value** When Override AN Parameters is enabled (override_an_parameters_enable=1), this register controls the value of AN_PAUSE used in the AN Base page [0]: Pause Ability [1]: Asymmetric Direction [2]: Reserved	RW	0x0
27:24	override_an_fec	**AN_FEC Override Value** When Override AN Parameters is enabled (override_an_parameters_enable=1), this register controls the value of AN_FEC used in the AN Base page [24] = 10G BASE-R RS-FEC Capability [25] = 10G BASE-R RS-FEC Request [26] = 25G IEEE RS-FEC Request [27] = 25G IEEE BASE-R RS-FEC Request	RW	0x0
23:16	override_an_tech	**AN_TECH Override Value, bits [7:0]** When Override AN Parameters is enabled (override_an_parameters_enable=1), this register controls the value of AN_TECH used in the AN Base page [16] = Reserved [17] = 10GBASE-KX4 (XAUI) [18] = 10GBASE-KR [19] = Reserved [20] = Reserved [21] = Reserved [22] = 100GBASE-KP4 [23] = 100GBASE-KR4	RW	0x0
15:0	user_base_page_low	**User Controlled AN Base page (lower bits)** When User Controlled Base pages are turned on (an_base_pages_ctrl=1), this register provides the lower bits of the User base page that is used instead of the default page [15] = Next page bit [14] = ACK bit (controlled by State Machine) [13] = Remote Fault bit [12:10]: Pause bits [9:5] = Echoed Nonce (set by SM) [4:0] = Selector	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		Note: Bit 49 (the PRBS bit of the AN BASE page) is generated by the SM.		

2.12.1.7. Auto Negotiation Config Register 4

Offset: 0xC4

Auto Negotiation Config Register 4 Fields

Bit	Name	Description	Access	Reset
31:0	user_base_page_high	**User Controlled AN Base page (upper bits)** [31:30] = FEC bits [29:5] = Technology Ability bits [4:0] = TX Nonce bits	RW	0x0

2.12.1.8. Auto Negotiation Config Register 5

Offset: 0xC5

Auto Negotiation Config Register Fields

Bit	Name	Description	Access	Reset
30:16	override_an_tech_22_8	**AN_TECH Override Value, bits [22:8]** When Override AN Parameters is enabled (override_an_parameters_enable=1), this register controls the lower bits of AN_TECH used in the AN Base page. [16] = 100GBASE-CR4 [17] = 25GBASE-KR-S/CR-S [18] = 25GBASE-KR/CR [19] = 2.5GBASE-KX [20] = 5GBASE-KR [21] = 50GBASE-KR/CR [22] = 100GBASE-KR2/CR2 [23] = 200GBASE-KR4/CR4 All other settings Reserved	RW	0x0
15:0	user_next_page_low	**User Controlled AN Next Page (lower bits)** The AN TX state machine uses these bits when user controlled Next Page is set (an_next_pages_ctrl=1). [15]: Next Page bit [14]: ACK bit (controlled by the state machine) [13]: MP bit [12]: ACK2 bits [11]: Toggle bit (controlled by the state machine) [10:0]: Message code field [10:0]/Unformatted code field[10:0] *Note:* When Consortium Next Page send is enabled (consortium_next_page_send=1), the Consortium Next Page sequence is sent after the last user Next Page.	RW	0x0

2.12.1.9. Auto Negotiation Config Register 6

Offset: 0xC6

Auto Negotiation Config Register 6 Fields

Bit	Name	Description	Access	Reset
31:0	user_next_page_high	**User Controlled AN Next page (upper bits)** [31:0]: Unformatted Code Field (or [47:16] when MP bit is low) *Note:* When Consortium Next Page Send is enabled (consortium_next_page_send=1), the first two User Next Pages will be ignored and replaced with the Consortium Next Page sequence	RW	0x0

2.12.1.10. Auto Negotiation Status Register 1

Offset = 0xC7

Auto Negotiation Status Register 1 Fields

Bit	Name	Description	Access	Reset
15:0	lp_base_page_low	**Link Partner Base Page (lower bits)** [15] = Link partner next page bit [14] = Link partner ACK [13] = Link partner RF bit [12:10] = Link partner PAUSE bits [9:5] = Link partner Echoed Nonce bits [4:0] = Link partner Selector bits	RO	0x0

2.12.1.11. Auto Negotiation Status Register 2

Offset: 0xC8

Auto Negotiation Status Register 2 Fields

Bit	Name	Description	Access	Reset
31:0	lp_base_page_high	**Link Partner Base Page (upper bits)** [[31:30] = Link partner FEC bits [29:5] = Link partner Technology Ability bits [4:0] = TX Nonce bits	RO	0x0

2.12.1.12. Auto Negotiation Status Register 3

Offset: 0xC9

Auto Negotiation Status Register 3 Fields

Bit	Name	Description	Access	Reset
15:0	lp_next_page_low	**Link Partner Next Page (lower bits)** [15] = Link partner next page bit [14] = Link partner ACK [13] = Link partner MP bit [12] = Link partner ACK2 bit [11] = Link partner Toggle bit [10:0] = Link partner Message/Unformatted bits	RO	0x0

2.12.1.13. Auto Negotiation Status Register 4

Offset: 0xCA

an_status4 Fields

Bit	Name	Description	Access	Reset
31:0	lp_next_page_high	**Link Partner Next Page (upper bits)** [31:0]: Link partner Unformatted bits	RO	0x0

2.12.1.14. Auto Negotiation Status Register 5

Offset: 0xCB

Auto Negotiation Status Register 5 Fields

Bit	Name	Description	Access	Reset
30:28	an_lp_adv_pause	**Link Partner PAUSE Ability bits** [28] = PAUSE as defined in Annex 28B [29] = ASM_DIR as defined in Annex 28B [30] = Reserved	RO	0x0
27	an_lp_adv_remote_fault	**Link Partner Remote Fault** Remote fault bit from Link Partner	RO	0x0
26:23	an_lp_adv_fec_f	**Link Partner FEC Ability Field** [23] = 25G RS-FEC requested [24] = 25G BASE-R FEC (CL 74 Firecode) requested [25] = FEC Ability [26] = FEC Requested	RO	0x0
22:0	an_lp_adv_tech_a	**Link Partner Technology Ability Field** [0] = 1000BASE-KX [1] = 10GBASE-KX4 [2] = 10GBASE-KR [3] = 40GBASE-KR4 [4] = 40GBASE-CR4 [5] = 100GBASE-CR10 [6] = 100GBASE-KP4 [7] = 100GBASE-KR4 [8] = 100GBASE-CR4 [9] = 25GBASE-KR-S/CR-S [10] = 25GBASE-KR/CR [11] = 2.5GBASE-KX [12] = 5GBASE-KR [13] = 50GBASE-KR/CR [14] = 100GBASE-KR2/CR2 [15] = 200GBASE-KR4/CR4 [22:16] = Reserved	RO	0x0

2.12.1.15. AN Channel Override

Offset: 0xCC

AN Channel Override Fields

Bit	Name	Description	Access	Reset
1:0	override_an_channel	**AN Channel Override** When override_an_chan_enable (0xC0[6]) is 1, this register selects AN channel, range 0-3.	RW	0x0

2.12.1.16. Consortium Next Page Override

Offset: 0xCD

Consortium Next Page Override Fields

Bit	Name	Description	Access	Reset
27:24	override_consortium_next_page_fec_control	**Override Consortium Next PAGE FEC Control** When Enable Consortium Next Page override is enabled (enable_consortium_next_page_override=1), this register overrides bits D27:D24 in the Unformatted Next Page with the following bits defined by the consortium:. [24] = F1- CL91 RS-FEC ability [25] = F2- CL74 RS-FEC ability [26] = F3- CL91 RS-FEC request [27] = F4- CL74 RS-FEC request	RW	0x0
19:0	override_consortium_next_page_tech	**Override Consortium Next Page Technology Ability** [8:0] = Override bits D8:D0 in the Unformatted Next Page from default of 0x003 to indicate extended technology abilities. [19:9] = Override bits D26:D16 in the Unformatted Next Page with the following bits defined by the consortium: • [12:9] = Reserved, set to 0 • [13] = 25GBASE-KR1 ability • [14] = 25GBASE-CR1 ability • [16:15] = Reserved, set to 0 • [17] = 50GBASE-KR2 ability • [18] = 50GBASE-CR2 ability	RW	0x3

2.12.1.17. Consortium Next Page Link Partner Status

Offset: 0xCE

Consortium Next Page Link Partner Status Fields

Bit	Name	Description	Access	Reset
27:24	lp_consortium_next_page_fec	**Link Partner Consortium Next Page FEC Ability** Contain bits D27:D24 from the decoded consortium Unformatted Next Page with the following bits defined by the consortium: [24]: F1- CL91 RS-FEC ability [25]: F2- CL74 RS-FEC ability [26]: F3- CL91 RS-FEC request [27]: F4- CL74 RS-FEC request	RO	0x0
19:0	lp_consortium_next_page_tech	**Link Partner Consortium Next Page Technology Ability** [8:0]: Contain bits D8:D0 from the decoded consortium Unformatted Next Page (default of 0x003) to indicate extended technology abilities.	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		[19:9]: Contain bits D26:D16 from the decoded consortium Unformatted Next Page with the following bits defined by the consortium: • [12:9]: Reserved • [13]: 25GBASE-KR1 ability • [14]: 25GBASE-CR1 ability • [16:15]: Reserved • [17]: 50GBASE-KR2 ability • [18]: 50GBASE-CR2 ability • [19]: Reserved		

2.12.1.18. Link Training Config Register 1

Offset: `0xD0`

Link Training Config Register 1 Fields

Set bits [19:17] to 1 for internal loopback (ILB) or very short connections to disable LT optimizations.

Bit	Name	Description	Access	Reset
31:28	`lt_cfg1_ovrd_bw`	**CTLE-BW Override** When `train_start_initpre` set to 1, use this value for serdes CTLE-BW initial setting.	RW	`0x0`
27:24	`lt_cfg1_ovrd_hf`	**CTLE-HF Override** When `train_start_initpre` set to 1, use this value for serdes CTLE-HF initial setting.	RW	`0x0`
23:20	`lt_cfg1_ovrd_lf`	**CTLE-LF Override** When `train_start_initpre` set to 1, use this value for serdes CTLE-LF initial setting.	RW	`0x0`
19	`lt_cfg1_disable_postlt`	**Disables Post-LT optimized serdes settings** 1: Disables optimizing serdes settings for KR post-LT and use default settings. May be required for internal loopback (ILB) or very short connections.	RW	`0x0`
18	`lt_cfg1_disable_prelt`	**Disables Pre-LT optimized serdes settings** 1: Disables optimizing serdes settings for KR pre-LT and use default settings. May be required for internal loopback (ILB) or very short connections.	RW	`0x0`
17	`lt_cfg1_disable_prxcal`	**Disables Periodic RX Adaptation during Data Mode** 1: Disables Periodic RX Adaptation during Data Mode May be required for internal loopback (ILB) or very short connections	RW	`0x0`
16	`lt_cfg1_disable_rxcal`	**Disables Initial RX Adaptation during LT** 1: Disables Initial RX Adaptation during LT	RW	`0x0`
15:4	`lt_mw_time`	**Sets LT Max_wait_timer timeout value** Each step is ~419.4us for NRZ, 2516.4us for PAM4. Default value is 1192, or ~500ms for NRZ, ~3s for PAM4.	RW	`12'h 4a8`
3	`train_start_initpre`	**Enable serdes initial RX setting override** 1: Override initial serdes RX settings via 0xD0[31:20] 0: Use default serdes RX settings for Link Training (default)	RW	`0x0`

continued...

Bit	Name	Description	Access	Reset
		Pre-LT optimized settings must be enabled for this to take effect (0xD0[18]==0)		
2	high_effort_train	**Use high-effort training** 1: LT will use higher effort for RX train (may take more than 500ms) 0: LT will use standard effort for RX train (500ms)	RW	0x0
1	dis_max_wait_tmr	**Disable Max Wait Timer** 1: Disable Max Wait Timer 0: Use Max Wait Timer (default)	RW	0x0
0	enable_link_training	**Enable Link Training** 1: Enable link training 0: Disable link training	RW	0x1

2.12.1.19. Link Training Status Register 1

Offset: 0xD2

Link Training Status Register 1 Fields

Bit	Name	Description	Access	Reset
27	link_training_failure_ln3	**Link Training Failure on Lane 3 (only applicable to 100G NRZ)** 1: Link Training Failed on Lane 3 0: Normal operation • Corresponds to state variable training_failure as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 4 lanes	RO	0x0
26	link_training_startup_ln3	**Link Training Startup up Protocol in Progress on Lane 3 (only applicable to 100G NRZ)** 1: Start-up protocol in progress 0: Start-up protocol complete • Corresponds to state variable training as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 4 lanes	RO	0x0
25	link_training_frame_lock_ln3	**Link Training Frame Lock Achieved on Lane 3 (only applicable to 100G NRZ)** 1: Training frame delineation detected 0: Searching for training frame boundaries • Corresponds to state variable frame_lock as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 4 lanes	RO	0x0
24	link_trained_ln3	**Receiver Trained on Lane 3 (only applicable to 100G NRZ)** 1: Receiver training completed 0: Training in progress • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only on links with 4 lanes	RO	0x0
19	link_training_failure_ln2	**Link Training Failure on Lane 2 (only applicable to 100G NRZ)** 1: Link Training Failed on Lane 2	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		0: Normal operation • Corresponds to state variable training_failure as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 4 lanes		
18	link_training_startup_ln2	**Link Training Startup up Protocol in Progress on Lane 2 (only applicable to 100G NRZ)** 1: Start-up protocol in progress 0: Start-up protocol complete • Corresponds to state variable training as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 4 lanes	RO	0x0
17	link_training_frame_lock_ln2	**Link Training Frame Lock Achieved on Lane 2 (only applicable to 100G NRZ)** 1: Training frame delineation detected 0: Searching for training frame boundaries • Corresponds to state variable frame_lock as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 4 lanes	RO	0x0
16	link_trained_ln2	**Receiver Trained on Lane 2 (only applicable to 100G NRZ)** 1: Receiver training completed 0: Training in progress • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only on links with 4 lanes	RO	0x0
11	link_training_failure_ln1	**Link Training Failure on Lane 1** 1: Link Training Failed on Lane 1 0: Normal operation • Corresponds to state variable training_failure as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 2 or 4 lanes	RO	0x0
10	link_training_startup_ln1	**Link Training Startup up Protocol in Progress on Lane 1** 1: Start-up protocol in progress 0: Start-up protocol complete • Corresponds to state variable training as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 2 or 4 lanes	RO	0x0
9	link_training_frame_lock_ln1	**Link Training Frame Lock Achieved on Lane 1** 1: Training frame delineation detected 0: Searching for training frame boundaries • Corresponds to state variable frame_lock as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 2 or 4 lanes	RO	0x0
8	link_trained_ln1	**Receiver Trained on Lane 1** 1: Receiver training completed 0: Training in progress • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1 • Valid only for links with 2 or 4 lanes	RO	0x0
3	link_training_failure_ln0	**Link Training Failure on Lane 0** 1: Link Training Failed on Lane 0	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		0: Normal operation • Corresponds to state variable training_failure as defined in IEEE 802.3 CL72.6.10.3.1		
2	link_training_startup_ln0	**Link Training Startup up Protocol in Progress on Lane 0** 1: Start-up protocol in progress 0: Start-up protocol complete • Corresponds to state variable training as defined in IEEE 802.3 CL72.6.10.3.1	RO	0x0
1	link_training_frame_lock_ln0	**Link Training Frame Lock Achieved on Lane 0** 1: Training frame delineation detected 0: Searching for training frame boundaries • Corresponds to state variable frame_lock as defined in IEEE 802.3 CL72.6.10.3.1	RO	0x0
0	link_trained_ln0	**Receiver Trained on Lane 0** 1: Receiver training completed 0: Training in progress • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1	RO	0x0

2.12.1.20. Link Training Config Register for Lane 0

Offset: 0xD3

Link Training Config Register for Lane 0 Fields

Bit	Name	Description	Access	Reset
26:16	lt_prbs_seed_ln0	**Link Training PRBS Seed for Lane 0 (only applicable to 100G and 25G NRZ)** Sets the initial seed for PRBS. Default value is 11'h57e	RW	0x57E
2:0	lt_prbs_pattern_select_ln0	**Link Training PRBS Pattern Select for Lane 0 (only applicable to 100G and 25G NRZ)** 0: Use Clause 92 Polynomial 0 1: Use Clause 92 Polynomial 1 2: Use Clause 92 Polynomial 2 3: Use Clause 92 Polynomial 3 4: Use Clause 72 Polynomial (if CL72 PRBS parameter is enabled) All other settings reserved • Default value for lane 0 is 0	RW	0x0

2.12.1.21. Link Training Config Register for Lane 1

Offset: 0xE0

Link Training Config Register for Lane 1 Fields

Bit	Name	Description	Access	Reset
26:16	lt_prbs_seed_ln1	**Link Training PRBS Seed for Lane 1 (only applicable to 100G)**	RW	0x645

continued...

Bit	Name	Description	Access	Reset
		Sets the initial seed for PRBS. Default value is 11'h645		
2:0	lt_prbs_pattern_select_ln1	**Link Training PRBS Pattern Select for Lane 1 (only applicable to 100G)** 0: Use Clause 92 Polynomial 0 1: Use Clause 92 Polynomial 1 2: Use Clause 92 Polynomial 2 3: Use Clause 92 Polynomial 3 4: Use Clause 72 Polynomial (if CL72 PRBS parameter is enabled) All other settings reserved • Default value for lane 1 is 1	RW	0x1

2.12.1.22. Link Training Config Register for Lane 2

Offset: 0xE4

Link Training Config Register for Lane 2 Fields

Bit	Name	Description	Access	Reset
26:16	lt_prbs_seed_ln2	**Link Training PRBS Seed for Lane 2 (only applicable to 100G NRZ)** Sets the initial seed for PRBS. Default value is 11'h72d	RW	0x72D
2:0	lt_prbs_pattern_select_ln2	**Link Training PRBS Pattern Select for Lane 2 (only applicable to 100G NRZ)** 0: Use Clause 92 Polynomial 0 1: Use Clause 92 Polynomial 1 2: Use Clause 92 Polynomial 2 3: Use Clause 92 Polynomial 3 4: Use Clause 72 Polynomial (if CL72 PRBS parameter is enabled) All other settings reserved • Default value for lane 2 is 2	RW	0x2

2.12.1.23. Link Training Config Register for Lane 3

Offset: 0xE8

Link Training Config Register for Lane 3 Fields

Bit	Name	Description	Access	Reset
26:16	lt_prbs_seed_ln3	**Link Training PRBS Seed for Lane 3 (only applicable to 100G NRZ)** Sets the initial seed for PRBS. Default value is 11'h7b6	RW	0x7B6
2:0	lt_prbs_pattern_select_ln3	**Link Training PRBS Pattern Select for Lane 3 (only applicable to 100G NRZ)** 0: Use Clause 92 Polynomial 0 1: Use Clause 92 Polynomial 1 2: Use Clause 92 Polynomial 2 3: Use Clause 92 Polynomial 3 4: Use Clause 72 Polynomial (if CL72 PRBS parameter is enabled) All other settings reserved • Default value for lane 3 is 3	RW	0x3

2.12.2. PHY Registers

Related Information

Register Map—E-Tile Transceiver PHY User Guide

2.12.2.1. PHY Module Revision ID

Offset: 0x300

PHY Module Revision ID Fields

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module.	RO	0x11112015

2.12.2.2. PHY Scratch Register

Offset: 0x301

PHY Scratch Register Fields

Bit	Name	Description	Access	Reset
31:0	scratch		RW	0x0

2.12.2.3. Loopback Mode

Offset: 0x30D

Loopback Mode

Registers to control the muxes that determine the loopback mode.

Bit	Name	Description	Access	Reset
23:21	rxpldmux_sel	**Select the input to the RX PLD Interface** • 7: TX PLD Interface • 1: RX PCS • 0: RX MAC	RW	0x0
20:18	rxmacmux_sel	**Select the input to the RX MAC** • 2: TX MAC • 0: RX PCS	RW	0x0
17:15	rxpcsmux_sel	**Select the input to the RX PCS** • 3: TX PCS • 0: RX PMA Interface The RX datapath must be reset after changing this setting.	RW	0x0
5:3	txpcsmux_sel	**Select input to TX PCS** • 2: RX PCS • 1: TX PLD Interface • 0: TX MAC	RW	0x0

2.12.2.4. PHY Configuration

Offset: 0x310

PHY Configuration Fields

Bit	Name	Description	Access	Reset
5	set_data_lock	Set data lock 1: Force PLL to lock to data	RW	0x0
4	set_ref_lock	Set ref lock 1: Force PLL to lock to reference	RW	0x0
2	soft_rx_rst	Soft RXP Reset 1: Resets the RX PCS and RX MAC.	RW	0x0
1	soft_tx_rst	Soft TXP Reset 1: Resets the TX PCS and TX MAC.	RW	0x0
0	eio_sys_rst	Ethernet IO System Reset 1: Resets the IP core (TX and RX MACs, Ethernet reconfiguration registers, PCS, and transceivers).	RW	0x0

2.12.2.5. Reset Sequencer RS-FEC Disable

Offset: 0x313

Reset Sequencer RS-FEC Disable Fields

Bit	Name	Description	Access	Reset
5	rsfec_disable	**Reset Sequencer RS-FEC Disable** Indicates to the reset sequencer that RSFEC has been disabled or bypassed. 0: RSFEC is enabled 1: RSFEC is disabled	RW	0x0

2.12.2.6. RX CDR PLL Locked

Offset: 0x321

RX CDR PLL Locked Fields

Bit	Name	Description	Access	Reset
3:0	eio_freq_lock	**CDR PLL locked** 1: Corresponding physical lane's CDR has locked to data for a 100G link.	RO	0x0

2.12.2.7. TX Datapath Ready

Offset: 0x322

TX Datapath Ready Fields

Bit	Name	Description	Access	Reset
0	tx_pcs_ready	**TX Ready** 1: TX Datapath is out of reset, stable, and ready for use.	RO	0x0

2.12.2.8. Frame Errors Detected

Offset: 0x323

Frame Errors Detected Fields

Bit	Name	Description	Access	Reset
19:0	frmerr	**Frame error(s) detected** • 1: A frame error was detected on corresponding lane • For single lanes, only bit 0 is used • For 100G links, bits 19:0 are used, corresponding to Virtual lanes 0 to 19 • This bit is sticky, and must be cleared by asserting sclr_frame_error	RO	0x0

2.12.2.9. Clear Frame Errors

Offset: 0x324

Clear Frame Errors Fields

Bit	Name	Description	Access	Reset
0	clr_frmerr	**Clear PHY frame error(s).** 1: Return all sticky frame error bits to 0.	RW	0x0

2.12.2.10. RX PCS Status for AN/LT

Offset: 0x326

RX PCS Status for AN/LT Fields

Bit	Name	Description	Access	Reset
1	hi_ber	**Hi-BER** 1: One or more virtual lanes are in the Hi-BER state defined in the Ethernet specification	RO	0x0
0	rx_aligned	**RX PCS fully aligned** 1: The RX PCS is fully aligned and ready to start decoding data *Note:* Not valid when RS-FEC is enabled, Use the 0x180[0] register in the E-Tile Transceiver PHY UG to verify if the RX lanes are aligned. For more information on register 0x180 (rsfec_lanes_rx_stat), refer to the RS-FEC Registers map in the *E-Tile Transceiver PHY UG*.	RO	0x0

2.12.2.11. PCS Error Injection

Offset: 0x327

PCS Error Injection Fields

Bit	Name	Description	Access	Reset
19:0	inj_err	**Inject Error** 0->1: Flip bits to inject encoding errors in corresponding virtual lane 0 :Clear all error injection settings • For EHIP with rate set to 100G, bits 0 to 19 are valid, and correspond to virtual lanes 0...19	RW	0x0

2.12.2.12. Alignment Marker Lock

Offset: 0x328

Alignment Marker Lock Fields

Bit	Name	Description	Access	Reset
0	am_lock	**AM Lock** 1: RX PCS has achieved Alignment Marker lock *Note:* Not valid for single-lane EHIP *Note:* Not valid when RS-FEC is enabled, Use the 0x180[0] register in the E-Tile Transceiver PHY UG to verify if the RX lanes are aligned.(33)	RO	0x0

2.12.2.13. Change in RX PCS Deskew Status

Offset: 0x329

lanes_deskewed Fields

Bit	Name	Description	Access	Reset
1	dskew_chng	**Change in deskewed status** 1: RX PCS went from deskewed to not deskewed, or from not deskewed to deskewed • Not valid for single lane channels (10G/25G) • This bit is sticky. Use clr_frmerr to set this bit back to 0. • Resetting the RX datapath, or the entire core will also clear the bit	RO	0x0
0	dskew_status	**Deskewed status** 1: RX PCS is deskewed 0: RX PCS is not currently deskewed. *Note:* There is some latency between this status bit and the actual state. • Not valid for single lane channels (10G/25G) • This bit is not sticky. Intel recommends that you replace this bit with a soft logic that can be made sticky based on the deskew_done port.	RO	0x0

2.12.2.14. BER Count

Offset: 0x32A

ber_count Fields

Bit	Name	Description	Access	Reset
31:0	count	**BER Count** • 32b count that increments each time the BER_BAS_SH state is entered • Rolls over when maximum count is reached • Clears when the channel is reset • Can be captured using snapshot or RX shadow request	RO	0x0

(33) For more information on register 0x180 (rsfec_lanes_rx_stat), refer to the RS-FEC Registers map in the *E-Tile Transceiver PHY UG*.

2.12.2.15. Transfer Ready (AIB reset) Status for EHIP, ELANE, and PTP Channels

Offset: 0x32B

aib_transfer_ready_status Fields

Bit	Name	Description	Access	Reset
21:20	ptp_rx_transfer_ready	**PTP TX Channels Transfer Ready Status** 1: transfer_ready is 1.	RO	0x0
19:16	ehip_rx_transfer_ready	**EHIP/ELANE RX Channels Transfer Ready Status** 1: transfer_ready is 1.	RO	0x0
5:4	ptp_tx_transfer_ready	**PTP TX Channels Transfer Ready Status** 1: transfer_ready is 1.	RO	0x0
3:0	ehip_tx_transfer_ready	**EHIP/ELANE TX Channels Transfer Ready Status** 1: transfer_ready is 1.	RO	0x0

2.12.2.16. EHIP, ELANE, and RS-FEC Reset Status

Offset: 0x32C

soft_rc_reset_status Fields

Bit	Name	Description	Access	Reset
5	ehip_rsfec_rx_reset	**EHIP rsfec RX reset from the reset controller**	RO	0x0
4	ehip_rsfec_tx_reset	**EHIP rsfec TX reset from the reset controller**	RO	0x0
3	ehip_rsfec_reset	**EHIP rsfec reset from the reset controller**	RO	0x0
2	ehip_rx_reset	**EHIP RX reset from the reset controller**	RO	0x0
1	ehip_tx_reset	**EHIP TX reset from the reset controller**	RO	0x0
0	ehip_reset	**EHIP reset from the reset controller**	RO	0x0

2.12.2.17. PCS Virtual Lane 0

Offset: 0x330

PCS Virtual Lane 0 Fields

Bit	Name	Description	Access	Reset
29:25	vlane5	**Virtual lane mapping** Original virtual lane position of the data mapped to the PCS lane with this index. For example, if you read the value 5 from vlane 12, it means the virtual lane data that the link partner transmitted on virtual lane 5 is being received on virtual lane 12. EHIP will reorder the data automatically	RO	0x1F
24:20	vlane4			
19:15	vlane3			
14:10	vlane2			
9:5	vlane1			
4:0	vlane0			

2.12.2.18. PCS Virtual Lane 1

Offset: 0x331

PCS Virtual Lane 1 Fields

Bit	Name	Description	Access	Reset
29:25	vlane11	**Virtual lane mapping** Original virtual lane position of the data mapped to the PCS lane with this index. For example, if you read the value 5 from vlane12, it means the virtual lane data that the link partner transmitted on virtual lane 5 is being received on virtual lane 12. EHIP will reorder the data automatically	RO	0x1F
24:20	vlane10			
19:15	vlane9			
14:10	vlane8			
9:5	vlane7			
4:0	vlane6			

2.12.2.19. PCS Virtual Lane 2

Offset: 0x332

PCS Virtual Lane 2 Fields

Bit	Name	Description	Access	Reset
29:25	vlane17	**Virtual lane mapping** Original virtual lane position of the data mapped to the PCS lane with this index. For example, if you read the value 5 from vlane 12, it means the virtual lane data that the link partner transmitted on virtual lane 5 is being received on virtual lane 12. EHIP will reorder the data automatically	RO	0x1F
24:20	vlane16			
19:15	vlane15			
14:10	vlane14			
9:5	vlane13			
4:0	vlane12			

2.12.2.20. PCS Virtual Lane 3

Offset: 0x333

PCS Virtual Lane 3 Fields

Bit	Name	Description	Access	Reset
9:5	vlane19	**Virtual lane mapping** Original virtual lane position of the data mapped to the PCS lane with this index. For example, if you read the value 5 from vlane 12, it means the virtual lane data that the link partner transmitted on virtual lane 5 is being received on virtual lane 12. EHIP will reorder the data automatically	RO	0x1F
4:0	vlane18			

2.12.2.21. Recovered Clock Frequency in KHz

Offset: 0x341

Recovered Clock Frequency in KHz Fields

Bit	Name	Description	Access	Reset
31:0	khz_rx	**Recovered clock frequency** Recovered clock frequency/10, in KHz.	RO	0x0

2.12.2.22. TX Clock Frequency in KHz

Offset: 0x342

TX Clock Frequency in KHz Fields

Bit	Name	Description	Access	Reset
31:0	khz_tx	**TX clock frequency** TX clock frequency/10, in KHz.	RO	0x0

2.12.2.23. Configuration Fields for TX PLD

Offset: 0x350

Configuration Fields for TX PLD Fields

Bit	Name	Description	Access	Reset
23	sel_50gx2	**Select 100G mode** Selects whether EHIP receives EMIB data from 2 or 4 lanes • 0: Use 4 EMIB channels for data input • 1: Use 2 EMIB channels for data input • The TX datapath must be reset after changing this field • Not used for single lane channels (10G/25G) • Defaults to 0 after power up • After i_csr_rst_n, default value depends on the what you selected in the **Select Ethernet Rate** parameter. — When **Select Ethernet Rate** = 100G, sel_50gx2 = 0	RW	0x0
22	tx_deskew_clear	**EMIB Deskew clear** Reset signal for the TX PLD deskew logic. • 0: Normal deskew operation • 1: TX EMIB deskew circuit in reset • Defaults to 0 after power up and i_csr_rst_n	RW	0x0
21:16	tx_deskew_chan_sel	**Deskew channel select** Specifies which channels participate in the deskew procedure • For single lane channels (10G/25G) — Only used when single lane is in EHIP_MAC_PTP mode — [0]=1: include EHIP lane datapath EMIB in deskew; defaults to 1 — [4]=1: include PTP EMIB from EHIP core in deskew; defaults to 1 • After reset, defaults to 0 • After i_csr_rst_n, default value depends on the **Select Ethernet Rate** parameter • For 100Gx4 channels — [0]=1: include EMIB0 in deskew; defaults to 1 — [1]=1: include EMIB1 in deskew; defaults to 1 — [2]=1: include EMIB2 in deskew; defaults to 1 — [3]=1: include EMIB3 in deskew; defaults to 1 — [4]=1: include EMIB4 in deskew; defaults to 1, only available in PTP mode — [5]=1: include EMIB5 in deskew; defaults to 1, only available in PTP mode	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		The TX datapath must be reset after changing values in this field.		
12:8	tx_fifo_afull	**TX FIFO almost full level** This is a debug feature that has been deprecated.	RW	0x0
2:0	tx_ehip_mode	**Portmap select** Selects how the synchronous input to the EHIP will be mapped. • 3h0: MAC interface • 3h1: MAC interface with PTP • 3h2: PCS (MII) interface • 3h3: PCS66 interface with forced encoder and scrambler bypass • 3h4: PCS66 interface • 3'h5: Reserved • 3'h6: Reserved • 3'h7: PMA direct interface • After power up, defaults to 0 • After i_csr_rst_n, default depends on the **Select Ethernet IP Layers** parameter — When **Select Ethernet IP Layers** = MAC+PCS, tx_ehip_mode = 3'd0 — When **Select Ethernet IP Layers** = PCS-only, tx_ehip_mode = 3'd2 — When **Select Ethernet IP Layers** = FlexE PHY, tx_ehip_mode = 3'd4 — When **Select Ethernet IP Layers**= OTN PHY, tx_ehip_mode = 3'd3 The TX datapath must be reset after changing this field.	RW	0x0

2.12.2.24. Status for TX PLDs

Offset: 0x351

Status for TX PLDs Fields

—

Bit	Name	Description	Access	Reset
24	err_tx_avst_fifo_overflow	**TX AVST FIFO Overflow** • Indicates that the FIFO was written while full • Overflow would never happen—if it does, this indicates a problem with the way i_valid is being driven • Once asserted this bit will hold value until the i_clear_internal_error port is asserted to clear it • This bit doesn't need to be polled—o_internal_err will be asserted if this signal goes high.	RO	0x0
23	err_tx_avst_fifo_empty	**TX AVST FIFO ran empty unexpectedly** • Asserts when the TX FIFO runs empty (regardless of read enable) • Does not apply when in MAC Mode • Empty should never happen—if it does, this indicates a problem with the way i_valid is being driven	RO	0x1
22	err_tx_avst_fifo_underflow	**TX AVST FIFO Underflow**	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		• Indicates that the FIFO was read when empty after steady state reading was established • Underflow should never happen—if it does, this indicates a problem with the way `i_valid` is being driven • Once asserted this bit will hold value until the `i_clear_internal_error` port is asserted to clear it, or the TX datapath is reset • This bit doesn't need to be polled—`o_internal_err` will be asserted if this signal goes high.		
21:16	`tx_dsk_active_chans`	**Active Channels.** [n]=1: Corresponding channel is part of the deskew set and has received a deskew marker since reset • This is a sticky bit that clears on reset and `dsk_clear` • Use this set of status bits to confirm that channels are receiving deskew markers • Remember that single lane channels will only use this when PTP is active, and in that case, will use bits 0 and 4 only *Note:* This status is for TX EMIB deskew, and has nothing to do with RX skew from the serial line.	RO	0x0
13:8	`tx_dsk_monitor_err`	**Skew Monitor Error Detected** [n]=1: An out-of-alignment EMIB deskew marker was detected on EMIB channel *n* after deskew • In single lane mode, channels 0 and 4 are used when PTP is active, where channel 4 is the PTP channel • Valid only when `tx_dsk_eval_done` = 1 • . *Note:* This status is for TX EMIB deskew, and has nothing to do with RX skew from the serial line.	RO	0x0
3:1	`tx_dsk_status`	**EMIB Deskew Status** 0: 0 cycles of delay added to remove TX EMIB skew 1: 1 cycle of delay added to remove TX EMIB skew 2: 2 cycles of delay added to remove TX EMIB skew 3: 3 cycles of delay added to remove TX EMIB skew 4: 4 cycles of delay added to remove TX EMIB skew 5: 5 cycles of delay added to remove TX EMIB skew 6: Reserved 7: Deskew Error—too much EMIB skew was detected • Valid only when `tx_dsk_eval_done` = 1 • When an error is detected, `deskew_clear` can be used to restart the deskew state machine *Note:* This status is for TX EMIB deskew, and has nothing to do with RX skew from the serial line.	RO	0x0
0	`tx_dsk_eval_done`	**Deskew evaluation is complete** 1: The TX PLD has finished attempting to deskew the EMIB channels connected to EHIP 0: TX PLD is still waiting for enough TX deskew markers to evaluate deskew *Note: Evaluation complete* does not mean that deskew was successful; it just means that the deskew state machine has come to a conclusion. • This signal is always required for `hip_ready` for multilane channels unless EMIB channels are deliberately excluded from deskew • For single lane channels, this signal is only needed when using PTP	RO	0x0

Bit	Name	Description	Access	Reset
		Note: This deskew is has nothing to do with RX PCS deskew or the serial input.		

2.12.2.25. Status for Dynamic Deskew Buffer

Offset: 0x354

Status for Dynamic Deskew Buffer Fields

Bit	Name	Description	Access	Reset
16	err_skew	**Dynamic Deskew Buffer overflow** 1: At least one lane of the Dynamic Deskew Buffer overflowed sometime in the past since the last time it was reset • Once asserted, the value will hold until the you clears it using the i_clear_internal_err port, or by resetting the RX datapath • The dynamic deskew buffer should be cleared by deasserting i_signal_ok, or by resetting the RX datapath • When a dynamic deskew buffer overflows, RX data is lost, which can cause packets to be lost, and frame errors • Even if no packets are lost, and the channel maintains integrity, an overflow should never happen, and is a sign that something in the channel did not follow the specification	RO	0x0
15:12	err_overflow	**Per lane Dynamic Deskew Buffer overflow indicator** [n]=1: The dynamic deskew buffer for lane n overflowed sometime in the past since the last time it was reset • Once asserted, the value will hold until you clear it using the i_clear_internal_err port, or by resetting the RX datapath • The dynamic deskew buffer should be cleared by deasserting i_signal_ok, or by resetting the RX datapath • When a dynamic deskew buffer overflows, RX data is lost, which can cause packets to be lost, and frame errors • Even if no packets are lost, and the channel maintains integrity, an overflow should never happen, and is a sign that something in the channel did not follow the specification	RO	0x0
11:8	rd_numdata	**Per lane Dynamic Deskew Buffer Almost Full** [n]=1: The occupancy of the dynamic deskew buffer in lane *n* has exceeded the watermark set by rxpma_max_skew • Valid only for lanes actually in used by a multi-lane EHIP core • Exceeding the watermark doesn't indicate an error, but may be a sign that the a problem in the past is now limiting the amount of skew variation the core can tolerate	RO	0x0

2.12.2.26. Configuration for RX PLD Block

Offset: 0x355

Configuration for RX PLD Block Fields

Bit	Name	Description	Access	Reset
4	sel_50gx2	**Select 50Gx2 mode** • 0: Use 4 EMIB channels for data output • 1: Use 2 EMIB channels for data output The RX datapath must be reset after changing this field. • Defaults to 0 after power up • After i_csr_rst_n, default value depends on the **Select Ethernet Rate** parameter — When **Select Ethernet Rate** = 100G, sel_50gx2 = 0 The RX datapath must be reset after changing this field.	RW	0x0
3	use_lane_ptp	**Select the input for the PTP channels** Valid for multilane EHIP (ehip_core) only • 0: PTP RX data comes from EHIP core • 1: PTP RX data comes from connected EHIP lanes Default value after power up and i_csr_rst is 0	RW	0x0
2:0	rx_ehip_mode	**Select RX Port map** Selects how data from the EHIP is presented through the EMIB • 3'h0: MAC interface • 3'h1: MAC interface with PTP • 3'h2: PCS (MII) interface • 3'h3: PCS66 interface for OTN (forced descrambler bypass) • 3'h4: PCS66 interface (descrambler optional) • 3'h7: PMA direct interface • After power up, defaults to 0 • After i_csr_rst_n, default depends on the **Select Ethernet IP Layers** parameter — When e**Select Ethernet IP Layers** = MAC+PCS, rx_ehip_mode = 3'd0 — When **Select Ethernet IP Layers** = PCS-only, rx_ehip_mode = 3'd2 — When **Select Ethernet IP Layers** = FlexE PHY, rx_ehip_mode = 3'd4 — When e**Select Ethernet IP Layers** = OTN PHY, rx_ehip_mode = 3'd3	RW	0x0

2.12.2.27. Configuration for RX PCS

Offset: 0x360

Configuration for RX PCS Fields

Bit	Name	Description	Access	Reset
20	use_hi_ber_monitor	**Enable Hi-BER Monitor** 0: Turn off Hi-BER monitor 1: Turn on Hi-BER monitor • The Hi-BER monitor is turned on by default because it is used for standard compliance • Hi-BER is needed to support Auto-Negotiation, and is generally used to report poor link conditions • When the Hi-BER monitor is turned on, if a Hi-BER condition is detected, the PCS will replace incoming data with Local Fault blocks	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		• Disable the Hi-BER monitor if you need to monitor RX data while in a Hi-BER state • At power-on, this register defaults to 0 • After `i_csr_rst_n` is asserted, the register is set to the value given by the `hi_ber_monitor` module parameter		
19:14	`rx_pcs_max_skew`	**Sets the maximum skew allowed by the RX PCS deskew logic** • This parameter is set by default to the maximum safe limit for RX PCS deskew, which is higher than the value required by the Ethernet Standard • The max skew can be lowered for testing • Raising the max skew beyond the default can be dangerous, since some of the margin left by the limit is used to absorb dynamically changing induced skews due to the operation of the internal PCS logic • At power-on, this register defaults to 6'h3F • When `i_csr_rst_n` is asserted, this register is set to the value given by the `rx_pcs_max_skew` module parameter	RW	0x3F
13:0	`am_interval`	**Expected interval between Alignment markers per Virtual lane** • This register is used only for multilane RX alignment. It is not used when RS-FEC is active, or for single lane channels • The interval is set by default to the number of valid blocks *per virtual lane* between alignment marker blocks required by the Ethernet Standard • For 100G links, RX alignment interval is TX alignment period/5 • Alignment interval can be reduced to the time required to link (especially in simulation), but it is critical that both sides of the link have compatible AM spacing • The RX PCS must be reset using `i_signal_ok`, RX datapath reset, or RX PCS reset, after changing this register • At power-on, this is set to 14'h3FFF; • After `i_csr_rst_n`, if the module parameter `sim_mode` is enabled, this parameter is set to a sim mode value appropriate for the selected rate • After `i_csr_rst_n`, if the module parameter `sim_mode` is disabled, this parameter is set to mission mode value appropriate for the selected rate	RW	0x3FFF

2.12.2.28. BIP Counter 0

Offset: `0x361`

BIP Error Count from RX Virtual Lane 0 Fields

Bit	Name	Description	Access	Reset
15:0	`count`	**BIP Counter**	RO	0x0

Bit	Name	Description	Access	Reset
		Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter		

2.12.2.29. BIP Counter 1

Offset: 0x362

BIP Error Count from RX Virtual Lane 1 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.30. BIP Counter 2

Offset: 0x363

BIP Counter 2 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multi-lane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX Datapath Reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.31. BIP Counter 3

Offset: 0x364

BIP Counter 3 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.32. BIP Counter 4

Offset: 0x365

BIP Counter 4 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.33. BIP Counter 5

Offset: 0x366

BIP Counter 5 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.34. BIP Counter 6

Offset: 0x367

BIP Counter 6 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.35. BIP Counter 7

Offset: 0x368

BIP Counter 7 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.36. BIP Counter 8

Offset: 0x369

BIP Counter 8 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors)	RO	0x0

Bit	Name	Description	Access	Reset
		• Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter		

2.12.2.37. BIP Counter 9

Offset: 0x36A

BIP Counter 9 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.38. BIP Counter 10

Offset: 0x36B

BIP Counter 10 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.39. BIP Counter 11

Offset: 0x36C

BIP Counter 11 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter**	RO	0x0

Bit	Name	Description	Access	Reset
		Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (`i_rx_rst_n`) • RX Stats Reset CSR does not reset this counter		

2.12.2.40. BIP Counter 12

Offset: `0x36D`

BIP Counter 12 Fields

Bit	Name	Description	Access	Reset
15:0	`count`	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (`i_rx_rst_n`) • RX Stats Reset CSR does not reset this counter	`RO`	`0x0`

2.12.2.41. BIP Counter 13

Offset: `0x36E`

BIP Counter 13 Fields

Bit	Name	Description	Access	Reset
15:0	`count`	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (`i_rx_rst_n`) • RX Stats Reset CSR does not reset this counter	`RO`	`0x0`

2.12.2.42. BIP Counter 14

Offset: `0x36F`

BIP Counter 14 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.43. BIP Counter 15

Offset: 0x370

BIP Counter 15 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.44. BIP Counter 16

Offset: 0x371

BIP Counter 16 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.45. BIP Counter 17

Offset: 0x372

BIP Counter 17 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.46. BIP Counter 18

Offset: 0x373

BIP Counter 18 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors) • Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (i_rx_rst_n) • RX Stats Reset CSR does not reset this counter	RO	0x0

2.12.2.47. BIP Counter 19

Offset: 0x374

BIP Counter 19 Fields

Bit	Name	Description	Access	Reset
15:0	count	**BIP Counter** Shows current BIP count for corresponding PCS lane. • Used only for multilane Ethernet links • Increments for a given virtual lane when the BIP calculated over all the data received since the last alignment marker does not match the BIP value in the current Alignment Marker • Valid only after Alignment Marker lock • Rolls over at max count (2^16 BIP errors)	RO	0x0

Bit	Name	Description	Access	Reset
		• Can be captured by snapshot or RX Shadow request • Resets on RX datapath reset (`i_rx_rst_n`) • RX Stats Reset CSR does not reset this counter		

2.12.2.48. Timer Window for Hi-BER Checks

Offset: `0x37A`

Timer Window for Hi-BER Checks Fields

Bit	Name	Description	Access	Reset
20:0	`cycles`	**Timer window for BER measurements** Sets the timer window for BER measurements in clock cycles. The Ethernet Standard (IEEE 802.3) defines the required times for Hi-BER measurements for each rate. These times must be converted to clock cycles with accurate within +1% and -25% of the specified times. *Note:* The clock rate you are using is different from the clock rate used to calculate the cycle count, you will need to scale the cycle count. • 100GBASE-R4: 21'd201415 (from Clause 82, 0.5ms +1%,-25% at 402.3 MHz • 100GBASE-R2/4: 21'd207518 (from Clause 82, 0.5ms +1%,-25% at 415.039 MHz • 25GBASE-R1: 21'd806451 (from Clause 107, 2.0 ms +1%, -25% at 402.3 MHz • 10GBASE-R1: 21'd20141 (from Clause 49, 0.125ms +1%, -25% at 161.13 MHz • 10GBASE-R1: 21'd50403 (from Clause 49, 0.125ms +1%, -25% at 402.83 MHz The RX PCS must be reset after changing this value.	RW	`0x31 2C7`

2.12.2.49. Hi-BER Frame Errors

Offset: `0x37B`

Hi-BER Frame Errors Fields

Bit	Name	Description	Access	Reset
6:0	`count`	**Hi-BER Frame Errors** Sets the BER count that triggers `hi_ber`. The Ethernet Standard (IEEE 802.3) defines the appropriate setting for `ber_invalid_count` based on rate. • 100GBASE-R4: 7'd97 (from Clause 82) • 25GBASE-R1: 7'd97 (from Clause 107) • 10GBASE-R1: 7'd16 (from Clause 49) The RX PCS must be reset after changing this value.	RW	`0x61`

2.12.2.50. Error Block Count

Offset: `0x37C`

Error Block Count Fields

Bit	Name	Description	Access	Reset
31:0	count	**Error block count** • Counts the number of Error blocks produced by the RX PCS Decoder • Valid only when the RX PCS Decoder is used and either alignment is achieved or alignment is not used and i_signal_ok = 1 • Error blocks can be received from the remote link, or generated by violations of the Ethernet Standard 64B66B encoding specification • The counter is 32b wide and rolls over when the max count is reached • The counter's output can be frozen while still incrementing using i_snapshot or rx_shadow_req. Snapshot/rx_shadow_req is recommended for all reads, since the counter is wider than 1 byte • The counter is reset when i_signal_ok = 0, the RX datapath is reset, or the RX PCS is reset	RO	0x0

2.12.2.51. Deskew Depth 0

Offset: 0x37F

Deskew Depth 0 Fields

Bit	Name	Description	Access	Reset
29:24	depth4	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
23:18	depth3	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
17:12	depth2	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
11:6	depth1	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
5:0	depth0	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0

2.12.2.52. Deskew Depth 1

Offset: 0x380

Deskew Depth 1 Fields

Bit	Name	Description	Access	Reset
29:24	depth4	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
23:18	depth3	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
17:12	depth2	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
11:6	depth1	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
5:0	depth0	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0

2.12.2.53. Deskew Depth 2

Offset: 0x381

Deskew Depth 2 Fields

Bit	Name	Description	Access	Reset
29:24	depth4	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
23:18	depth3	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
17:12	depth2	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
11:6	depth1	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
5:0	depth0	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0

2.12.2.54. Deskew Depth 3

Offset: 0x382

Deskew Depth 3 Fields

Bit	Name	Description	Access	Reset
29:24	depth4	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
23:18	depth3	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
17:12	depth2	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
11:6	depth1	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0
5:0	depth0	**Deskew depth for one of the PCS Virtual lanes**	RO	0x0

2.12.2.55. RX PCS Test Error Count

Offset: 0x383

RX PCS Test Error Count Fields

Bit	Name	Description	Access	Reset
31:0	count	**RX PCS Test Error Count** The register is reset by resetting the RX datapath.	RO	0x0

2.12.3. TX MAC Registers

2.12.3.1. TX MAC Module Revision ID

Offset: 0x400

TX MAC Module Revision ID Fields

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module Returns a 4 byte value indicating the revision of this design	RO	0x11112015

2.12.3.2. TX MAC Scratch Register

Offset: 0x401

TX MAC Scratch Register Fields

Bit	Name	Description	Access	Reset
31:0	scratch	32 bits of scratch register space for testing	RW	0x0

2.12.3.3. Link Fault Configuration

Offset: 0x405

Link Fault Configuration Fields

Bit	Name	Description	Access	Reset
3	force_rf	**Force the TX MAC to transmit Remote Faults when link fault signaling is on** 1: TX MAC transmits Remote Faults 0: TX MAC operates normally	RW	0x0
2	disable_rf	**Send idles instead of remote faults for local faults in unidirectional mode** 1: In unidirectional mode, local faults cause the TX to transmit Idles 0: In unidirectional mode, local faults cause the TX to transmit Remote Faults (spec default)	RW	0x0
1	en_unidir	**Enable Unidirectional Link Fault** 1: EHIP enables support for unidirectional link fault signaling as described in Clause 66 Remote faults will have no impact on TX data, and Local faults will cause the TX to transmit Remote fault Ordered sets between frames • After power-on, en_unidir is set to 0 • After i_csr_rst_n, en_unidir is set according to the module parameter link_fault_mode	RW	0x0
0	en_lf	**Enable Link Fault Reporting** 1: The TX PCS will transmit link fault messages based on link faults detected by the RX • After power-on, en_lf is set to 1'b1 • After i_csr_rst_n, en_lf is set according to the module parameter link_fault_mode 0: The TX PCS will not respond to link faults	RW	0x1

2.12.3.4. IPG Words to remove per Alignment Marker Period

Offset: 0x406

IPG Words to remove per Alignment Marker Period Fields

Bit	Name	Description	Access	Reset
15:0	ipg_col_rem	**IPG_COL_REM** 16b value that sets the number of IPG words that will be removed during an alignment marker period for a fully occupied link to make space for alignment markers. This parameter can also be used to scale IPG in ppm increments for rate balance. • After power-on, ipg_col_rem is set to 16'd20 • After i_cfg_rst_n, ipg_col_rem is set to the standard value required for the selected line rate, plus the value of the module parameter ipg_removed_per_am_period.	RW	0x14

2.12.3.5. Maximum TX Frame Size

Offset: 0x407

Maximum TX Frame Size Fields

Bit	Name	Description	Access	Reset
15:0	max_tx	**MAX_TX_SIZE_CONFIG** 16 bits value that sets the maximum TX frame size. When TX frames exceed this size, the CNTR_TX_OVERSIZE statistic is incremented • After power-up, max_tx is set to 16'd9600 • After i_csr_rst_n is asserted, max_tx is set to the value given by the module parameter tx_max_frame_size	RW	0x2580

2.12.3.6. TX MAC Configuration

Offset: 0x40A

TX MAC Configuration Fields

Bit	Name	Description	Access	Reset
3	en_saddr_insert	**Enable Source Address Insertion** 0: Client provides Source Address 1: TX MAC inserts source addresses stored in CSRs • At power-up, en_saddr_insert is set to 0 • After i_csr_rst_n, en_saddr_insert is set to the value given by source_address_insertion	RW	0x0
2	disable_txmac	**Disable TX MAC** 0: TX MAC operates normally 1: TX MAX is disabled - it behaves as though it has been PAUSED by the remote link until disable is turned off	RW	0x0
1	disable_txvlan	**Disable VLAN detection for TX Stats** 0: TX frames with VLAN headers will be counted as VLAN frames in the TX stats 1: VLAN headers will not be considered by the TX stats block • At power-on, disable_txvlan is set to 1 • After i_csr_rst_n is asserted, disable_vlan is set to the value given by module parameter tx_vlan_detection	RW	0x0

2.12.3.7. EHIP TX MAC Feature Configuration

Offset: 0x40B

EHIP TX MAC Feature Configuration Fields

Bit	Name	Description	Access	Reset
31:15	am_period	**TX Alignment Marker Period** Sets the number of TX clock cycles that are used to send regular data between Alignment Markers	RW	0x13FFB

continued...

Bit	Name	Description	Access	Reset
		• At power-on, this is set to 17'd81915 • After `i_csr_rst_n`, if the module parameter **sim_mode** is enabled, this parameter is set to a simulation mode value appropriate for the selected rate • After `i_csr_rst_n`, if the module parameter **sim_mode** is disabled, this parameter is set to mission mode value appropriate for the selected rate		
9	`txcrc_covers_preamble`	**Enable CRC over preamble** 0: TX CRC calculated over Ethernet Frame (default) 1: TX CRC calculated over frame plus preamble • At power-on, `txcrc_covers_preamble` is set to 0 • After `i_csr_rst_n` is asserted, `txcrc_covers_preamble` is set to the value given by module parameter **txcrc_covers_preamble**	RW	0x0
8:6	`flowreg_rate`	**Sets the valid toggle rate of the TX MAC flow regulator** 0: 100G 1: Reserved 2: Reserved 3:25G 4:10G 5: Reserved 6: Reserved 7: Use custom cadence	RW	0x0
5:3	`am_width`	**Sets the number of cycles for each AM pulse** Sets the number of TX clock cycle that the AM pulse is held high • After power-up, `am_width` is set to 5 • After `i_csr_rst_n` is asserted, `am_width` is set according to the rate of the channel • Set to 5 for 100G channels • Set to 4 for 25G channels that use RS-FEC • Set to 1 for all other types of channels	RW	0x5
2:1	`ipg`	**DIC Average Min IPG** Sets the average minimum IPG enforced by the Deficit Idle Counter: • 2'd0: 12 bytes (default) • 2'd1:10 bytes • 2'd2:8 bytes • 2'd3:1 byte • After power-up, ipg is set to 0 (12 bytes) • After `i_csr_rst_n` is asserted, ipg is set to the value given by the module parameter **tx_ipg_size**	RW	0x0
0	`en_pp`	**Enable TX Preamble Passthrough** 1: Preamble-passthrough mode enabled - bytes 1 to 7 of each SOP word will be used as preamble bytes at the start of the Ethernet packet 0: A standard Ethernet preamble will be used for TX packets	RW	0x0

2.12.3.8. TX MAC Source Address Lower Bytes

Offset: `0x40C`

TX MAC Source Address Lower Bytes Fields

Bit	Name	Description	Access	Reset
31:0	saddrl	**Source Address Insertion Source Address lower bytes** Lower 4 bytes of the 6 byte source address that is inserted by the TX MAC when TX source address insertion is enabled • At power-on, saddrl is set to 1 • After i_csr_rst_n is asserted, saddrl is set to the value given by module parameter txmac_saddr[31:0]	RW	0x22334455

2.12.3.9. TX MAC Source Address Higher Bytes

Offset: 0x40D

TX MAC Source Address Higher Bytes Fields

Bit	Name	Description	Access	Reset
15:0	saddrh	**Source Address Insertion Source Address upper bytes** Upper 2 bytes of the 6 byte source address that is inserted by the TX MAC when TX source address insertion is enabled • At power-on, saddrh is set to 1 • After i_csr_rst_n is asserted, saddrh is set to the value given by module parameter txmac_saddr[47:32]	RW	0x11

2.12.4. RX MAC Registers

2.12.4.1. RX MAC Module Revision ID

Offset: 0x500

RX MAC Module Revision ID Fields

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module Returns a 4 byte value indicating the revision of this design	RO	0x11112015

2.12.4.2. RX MAC Scratch Register

Offset: 0x501

RX MAC Scratch Register Fields

Bit	Name	Description	Access	Reset
31:0	scratch	32 bits of scratch register space for testing	RW	0x0

2.12.4.3. Maximum RX Frame Size

Offset: 0x506

Maximum RX Frame Size Fields

Bit	Name	Description	Access	Reset
15:0	max_rx	**MAX_RX_SIZE_CONFIG** 16b value that sets the maximum RX frame size. When RX frames exceed this size, the CNTR_RX_OVERSIZE statistic is incremented, and the appropriate rx_error bit is asserted with EOP on the frame to indicate the frame is oversize Sets the maximum size of a RX frame in octets before it will be counted as an oversize frame • When enforce_max_frame_size is enabled, frames longer than max_rx will be truncated on arrival, and marked as oversize, with an FCS error • After power-up, max_rx is set to 16'd9600 • After i_csr_rst_n, max_rx is set to the value given by the module parameter **RX maximum frame size**	RW	0x2580

2.12.4.4. RX CRC Forwarding

Offset: 0x507

RX CRC Forwarding Fields

Bit	Name	Description	Access	Reset
0	forward_rx_crc	**Forward RX CRC values** 0: Remove CRC from RX frames 1: Leave CRC in RX frames and forward it to RX Client logic	RW	0x0

2.12.4.5. Link Fault Status

Offset: 0x508

Link Fault Status Fields

Bit	Name	Description	Access	Reset
1	rfault	**Remote Fault detected** 1: EHIP detected a remote fault	RO	0x0
0	lfault	**Local Fault detected** 1: EHIP detected a local fault	RO	0x0

2.12.4.6. RX MAC Configuration

Offset: 0x50A

RX MAC Configuration Fields

Bit	Name	Description	Access	Reset
8	remove_rx_pad	**Remove PADs from padded frames** 0: Padded frames are not altered 1: Pads are removed from padded frames	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		• After power-on, remove_rx_pad defaults to 0 • Afteri_csr_rst_n, remove_rx_pad is set to the value given by the parameter **Bytes to remove from RX frames** in the parameter editor.		
7	enforce_max_rx	**Enforce Maximum frame size on RX packets** 0: Oversized frames are not altered 1: Frames are ended with FCS error if they exceed the programmed RX maximum frame size • After power on, this register defaults to 0 • After i_csr_rst_n, this register is set to value of the parameter **Enforce Maximum Frame Size** in the parameter editor.	RW	0x0
4	en_strict_preamble	**Enable Strict Preamble Checking** 0: Custom Preamble bytes are allowed between SOP and SFD 1: Packets will be dropped if they do not have standard preamble bytes • After power-up, en_strict_preamble is set to 0 • After i_csr_rst_n is asserted, en_strict_preamble is set to the value given by the parameter **Enable strict preamble check** in the parameter editor.	RW	0x0
3	en_check_sfd	**Enable Start Frame Delimiter Checking** 0:Custom SFD bytes are allowed in preambles 1:Packets will be dropped if they do not have a standard Start Frame Delimiter • After power-up, en_check_sfd is set to 0 • After i_csr_rst_n is asserted, en_check_sfd is set to the value given by the parameter **Enable strict SFD checking** in the parameter editor.	RW	0x0
1	disable_rxvlan	**Disable RX VLAN detection** 0: EHIP detects VLAN frames, counts them separately in stats, and marks them at EOP 1:EHIP ignores VLAN in RX data, and treats VLAN headers as payload bytes • At power-on, this register defaults to 0 • When i_csr_rst_n is asserted, this register is set to the value given by the parameter **RX VLAN detection** in the parameter editor.	RW	0x0
0	en_plen	**Enable Packet Length Checking** 1: EHIP will assert the length error bit of rx_error at EOP for Frames where the Type/Length field is a length, and the length advertised is greater than the length of the frame that was received • After power-on, en_plen is set to 1 • After i_csr_rst_n,en_plen is set according to the module parameter rx_length_checking	RW	0x1

2.12.4.7. EHIP RX MAC Feature Configuration

Offset: 0x50B

EHIP RX MAC Feature Configuration Fields

Bit	Name	Description	Access	Reset
1	rxcrc_covers_preamble	**Enable CRC over preamble** 0: RX CRC calculated over Ethernet Frame (default) 1: RX CRC calculated over frame plus preamble • At power-on, this register is set to 0 • When i_csr_rst_n is asserted, this register is set to the value given by the module rxcrc_covers_preamble	RW	0x0
0	en_pp	**Enable RX Preamble Passthrough** 1: Preamble-passthrough mode enabled - the preamble received with each packet will be passed to the user 0: RX preamble will not be passed to the user	RW	0x0

2.12.5. Pause and Priority- Based Flow Control Registers

2.12.5.1. TXSFC Module Revision ID

Offset: 0x600

TXSFC Module Revision ID Fields

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module Returns a 4 byte value indicating the revision of this design	RO	0x11112015

2.12.5.2. TX SFC Scratch Register

Offset: 0x601

TX SFC Scratch Register Fields

Bit	Name	Description	Access	Reset
31:0	scratch	32 bits of scratch register space for testing	RW	0x0

2.12.5.3. Enable TX Pause Ports

Offset: 0x605

Enable TX Pause Ports Fields

Bit	Name	Description	Access	Reset
8:0	en_pfc_port	**Enable TX PAUSE or TX PFC port.** Bits [7:0]: For PFC Bit [8]: For PAUSE 1: Corresponding tx_pfc_pause port can be used to trigger TX PFC frames • After power on, bit 8 defaults to 1 • After i_csr_rst_n, the value of bit 8 is set based on the module parameter **Stop TX traffic when link partner sends PAUSE?**	RO	0x1

2.12.5.4. TX Pause Request

Offset: 0x606

TX Pause Request Fields

Bit	Name	Description	Access	Reset
8:0	req_pause	**Request TX PAUSE or TX PFC.** Bits [7:0]: For PFC Bit [8]: For PAUSE Set to request the transmission of TX Pause frames Works the same way as the corresponding tx_pause port or tx_pfc port	RW	0x0

2.12.5.5. Enable Automatic TX Pause Retransmission

Offset: 0x607

Enable Automatic TX Pause Retransmission Fields

Bit	Name	Description	Access	Reset
8:0	en_holdoff	**Enable Holdoff timer.** Turns on automatic XOFF pause frame retransmission using a holdoff timer for the corresponding tx_pfc_pause port Bits [7:0]: For PFC Bit [8]: For PAUSE 1: Holdoff timer enabled. • EHIP will transmit a new set of XOFF frames whenever the holdoff timer expires while a port or CSR request is still high for the corresponding queue • At power up this register defaults to 1 • After i_csr_rst_n is asserted, this register value is set according to the module parameter flow_control_holdoff_mode	RW	0x1

2.12.5.6. Retransmit Holdoff Quanta

Offset: 0x608

Retransmit Holdoff Quanta Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**Retransmit Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1	RW	0xFFFF

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, `holdoff_quanta` defaults to 16'hFFFF • After `i_cfg_rst_n`, `holdoff_quanta` defaults to the value in the module parameter `holdoff_quanta` for pause, and `pfc_holdoff_quanta_n` for PFC		

2.12.5.7. Retransmit Pause Quanta

Offset: `0x609`

Retransmit Pause Quanta Fields

Bit	Name	Description	Access	Reset
15:0	`pause_quanta`	**Retransmit Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, `pause_quanta` is set to the default value (16'hFFFF) • After `i_csr_rst_n`, `pause_quanta` is set to the value given by the module parameter `pause_quanta` for PAUSE, and `pfc_pause_quanta_n` for PFC	RW	`0xFFFF`

2.12.5.8. Enable TX XOFF

Offset: `0x60A`

Enable TX XOFF Fields

Bit	Name	Description	Access	Reset
2:0	en_xoff_qnum_sel	**Enable XOFF** Activates automatic TX response to XOFF requests from the link partner n standard flow control mode, 1=EHIP responds to XO FF requests it receives by stopping the flow of TX data • After power on, this register defaults to 0 • After i_csr_rst_n, this register is set to the value based on the module parameter flow_control	RW	0x0

2.12.5.9. Enable Uniform Holdoff

Offset: 0x60B

Enable Uniform Holdoff Fields

Bit	Name	Description	Access	Reset
0	en_holdoff_all	**Enable uniform holdoff** All queues must use a holdoff at least as long as the holdoff programmed into Set Uniform Holdoff register. • At power up this register defaults to 0 • After i_csr_rst_n is asserted, this register value is set according to the module parameter flow_control_holdoff_mode	RW	0x0

2.12.5.10. Set Uniform Holdoff

Offset: 0x60C

Set Uniform Holdoff Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_all_quanta	**Uniform holdoff time** 16b minimum holdoff time required of all PFC queues whenen_holdoff_all = 1.	RW	0x0

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles • Minimum value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Maximum value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: min (Pause Quanta register value) - (60 + Maximum TX Frame Size register value/8) • For example, if the minimum pfc pause quanta over all queues is 500, and the max tx frame size is 800 bytes, the max holdoff quanta is 500-(60+100) = 340 • These values are based on the maximum overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: min(Pause Quanta register value) - (50 + Maximum TX Frame Size register value/32) • At power up this register defaults to 0 • After `i_csr_rst_n` is asserted, this register value is set according to the module parameter `uniform_holdoff_quanta`		

2.12.5.11. Lower 4 bytes of the Destination address for Flow Control

Offset: `0x60D`

Lower 4 bytes of the Destination address for Flow Control Fields

Bit	Name	Description	Access	Reset
31:0	`daddrl`	**Flow control Destination Address** Lower 4 bytes of the 6 byte destination address used for SFC and PFC frames • At power-on, daddrl is set to 32'hC2000001 • After `i_csr_rst_n` is asserted, daddrl is set to the value given by module parameter `tx_pause_daddr[31:0]`	RW	`0xC2000001`

2.12.5.12. Higher 2 bytes of the Destination address for Flow Control

Offset: `0x60E`

Higher 2 bytes of the Destination address for Flow Control Fields

Bit	Name	Description	Access	Reset
15:0	`daddrh`	**Flow control Destination Address** Upper 2 bytes of the 6 byte destination address used for SFC and PFC frames • At power-on, `daddrh` is set to 16'h0180 • After `i_csr_rst_n` is asserted, `daddrh` is set to the value given by module parameter `tx_pause_daddr[47:32]`	RW	`0x180`

2.12.5.13. Lower 4 bytes of the Source address for Flow Control frames

Offset: 0x60F

Lower 4 bytes of the Source address for Flow Control frames Fields

Bit	Name	Description	Access	Reset
31:0	saddrl	**Lower 4 bytes of the Flow control Source Address** Lower 4 bytes of the 6 byte source address used for SFC and PFC frames • At power-on, saddrl is set to 32'hCBFC5ADD • After i_csr_rst_n is asserted, saddrl is set to the value given by module parameter tx_pause_saddr[31:0]	RW	0xCBFC5ADD

2.12.5.14. Higher 2 bytes of the Source address for Flow Control frames

Offset: 0x610

Higher 2 bytes of the Source address for Flow Control frames Fields

Bit	Name	Description	Access	Reset
15:0	saddrh	**Higher 2 bytes of the Flow control Source Address** Higher 2 bytes of the 6 byte source address used for SFC and PFC frames • At power-on, saddrh is set to 16'hE100 • After i_csr_rst_n is asserted, saddrh is set to the value given by module parameter tx_pause_saddr[47:32]	RW	0xE100

2.12.5.15. TX Flow Control Feature Configuration

Offset: 0x611

txsfc_ehip_cfg Fields

Bit	Name	Description	Access	Reset
1	en_pfc	**Enable Priority Flow Control TX** 1: Enable Priority Flow Control • This feature requires the TX MAC • Enabling this feature allows the TX MAC to transmit PFC frames when requested, even if the flow of data through the datapath is inhibited • The TX datapath must be reset after changing this field • To shut off TX PFC without resetting the datapath, use tx_pause_en • To request the transmission of PFC frames through AVMM, use tx_pause_request • After power-on, this reset is set to 0 • After i_csr_rst_n, this register is set to a value given by the module parameter flow_control	RW	0x0
0	en_sfc	**Enable Standard Flow Control TX** 1:Enable Standard Flow Control (link PAUSE)	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		• This feature requires the TX MAC • Enabling this feature allows the TX MAC to transmit PAUSE frames when requested, even if the flow of data through the datapath is inhibited • The TX datapath must be reset after changing this field • To shut off TX PAUSE without resetting the datapath, use `tx_pause_en` • To request the transmission of PAUSE frames through AVMM, use `tx_pause_request` • After power-on, this reset is set to 0 • After `i_csr_rst_n`, this register is set to a value given by the module parameter `flow_control`		

2.12.5.16. Pause Quanta 0

Offset: `0x620`

Pause Quanta 0 Fields

Bit	Name	Description	Access	Reset
15:0	`pause_quanta`	**Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, `pause_quanta` is set to the default value (16'hFFFF) • After `i_csr_rst_n`, `pause_quanta` is set to the value given by the module parameter `pause_quanta` for PAUSE, and `pfc_pause_quanta_n` for PFC	RW	`0xFFFF`

2.12.5.17. Pause Quanta 1

Offset: `0x621`

Pause Quanta Fields

Bit	Name	Description	Access	Reset
15:0	`pause_quanta`	**Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames	RW	`0xFFFF`

Bit	Name	Description	Access	Reset
		• The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, `pause_quanta` is set to the default value (16'hFFFF) • After `i_csr_rst_n`, `pause_quanta` is set to the value given by the module parameter `pause_quanta` for PAUSE, and `pfc_pause_quanta_n` for PFC		

2.12.5.18. Pause Quanta 2

Offset: `0x622`

Pause Quanta 2 Fields

Bit	Name	Description	Access	Reset
15:0	`pause_quanta`	**Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, `pause_quanta` is set to the default value (16'hFFFF) • After `i_csr_rst_n`, `pause_quanta` is set to the value given by the module parameter `pause_quanta` for PAUSE, and `pfc_pause_quanta_n` for PFC	RW	`0xFFFF`

2.12.5.19. Pause Quanta 3

Offset: `0x623`

Pause Quanta 3 Fields

Bit	Name	Description	Access	Reset
15:0	`pause_quanta`	**Pause quanta**	RW	`0xFFFF`

Bit	Name	Description	Access	Reset
		16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, `pause_quanta` is set to the default value (16'hFFFF) • After `i_csr_rst_n`, `pause_quanta` is set to the value given by the module parameter `pause_quanta` for PAUSE, and `pfc_pause_quanta_n` for PFC		

2.12.5.20. Pause Quanta 4

Offset: `0x624`

Pause Quanta 4 Fields

Bit	Name	Description	Access	Reset
15:0	`pause_quanta`	**Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, `pause_quanta` is set to the default value (16'hFFFF) • After `i_csr_rst_n`, `pause_quanta` is set to the value given by the module parameter `pause_quanta` for PAUSE, and `pfc_pause_quanta_n` for PFC	RW	`0xFFFF`

2.12.5.21. Pause Quanta 5

Offset: `0x625`

Pause Quanta 5 Fields

Bit	Name	Description	Access	Reset
15:0	pause_quanta	**Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, pause_quanta is set to the default value (16'hFFFF) • After i_csr_rst_n, pause_quanta is set to the value given by the module parameter pause_quanta for PAUSE, and pfc_pause_quanta_n for PFC	RW	0xFFFF

2.12.5.22. Pause Quanta 6

Offset: 0x626

Pause Quanta 6 Fields

Bit	Name	Description	Access	Reset
15:0	pause_quanta	**Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, pause_quanta is set to the default value (16'hFFFF) • After i_csr_rst_n, pause_quanta is set to the value given by the module parameter pause_quanta for PAUSE, and pfc_pause_quanta_n for PFC	RW	0xFFFF

2.12.5.23. Pause Quanta 7

Offset: 0x627

Pause Quanta 7 Fields

Bit	Name	Description	Access	Reset
15:0	pause_quanta	**Pause quanta** 16b value specifying the Quanta value transmitted in XOFF frames • The Quanta value indicates to the remote link partner the amount of time to apply flow control • 1 Quanta corresponds to 512 bit times. — On a 10Gx1 or 25Gx1 link, 512 bit times is 8 valid clock cycles — On a 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF • The default value for quanta is 16'hFFFF. Using the max value simplifies the use of flow control by making it directly controlled by XON and XOFF, and reduces the bandwidth required for retransmitted control frames • After power-up, pause_quanta is set to the default value (16'hFFFF) • After i_csr_rst_n, pause_quanta is set to the value given by the module parameter pause_quanta for PAUSE, and pfc_pause_quanta_n for PFC	RW	0xFFFF

2.12.5.24. PFC Holdoff Quanta 0

Offset: 0x628

PFC Holdoff Quanta 0 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.25. PFC Holdoff Quanta 1

Offset: 0x629

PFC Holdoff Quanta 1 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.26. PFC Holdoff Quanta 2

Offset: 0x62A

PFC Holdoff Quanta 2 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.27. PFC Holdoff Quanta 3

Offset: 0x62B

PFC Holdoff Quanta 3 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.28. PFC Holdoff Quanta 4

Offset: 0x62C

PFC Holdoff Quanta 4 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.29. PFC Holdoff Quanta 5

Offset: 0x62D

PFC Holdoff Quanta 5 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.30. PFC Holdoff Quanta 6

Offset: 0x62E

PFC Holdoff Quanta 6 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.31. PFC Holdoff Quanta 7

Offset: 0x62F

PFC Holdoff Quanta 7 Fields

Bit	Name	Description	Access	Reset
15:0	holdoff_quanta	**PFC Holdoff Quanta** 16b value specifying holdoff time before another XOFF is transmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1 • Times are programmed in holdoff quanta — For 10G and 25G links, 1 Holdoff Quanta = 8 clock cycles — For 100G links, 1 Holdoff Quanta = 2 clock cycles • Min value is 1, but to minimize wasted bandwidth, holdoff should be set as large as possible without exceeding the recommended max value • Max value for correct operation where holdoff retransmits PFC requests before the previously transmitted Quanta expires is: — For 10G and 25G links: corresponding pause/pfc quanta - (60 + Maximum TX Frame Size register value/8) • For example, if the corresponding pause quanta is 1000, and the max tx frame size is 880 bytes, the max holdoff quanta is 1000-(60+110) = 830 • These values are based on the max overrun limits defined in IEEE 802.3 2015 Annex 31B — For 100Gx4 links: corresponding pause/pfc quanta - (50 + Maximum TX Frame Size register value/32) • After power-on, holdoff_quanta defaults to 16'hFFFF • After i_cfg_rst_n, holdoff_quanta defaults to the value in the module parameter holdoff_quanta for pause, and pfc_holdoff_quanta_n for PFC	RW	0xFFFF

2.12.5.32. RXSFC Module Revision ID

Offset: 0x700

RXSFC Module Revision ID Fields

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module Returns a 4 byte value indicating the revision of this design	RO	0x11112015

2.12.5.33. RXSFC Scratch Register

Offset: 0x701

RXSFC Scratch Register Fields

Bit	Name	Description	Access	Reset
31:0	scratch	32 bits of scratch register space for testing	RW	0x0

2.12.5.34. Enable RX Pause Frame Processing

Offset: 0x705

Enable RX Pause Frame Processing Fields

Bit	Name	Description	Access	Reset
7:0	en_rx_pause	**Enable Rx Pause** 1:Enable PFC port for selected queue • After power-on, this reset is set to 0 • After i_csr_rst_n, this register is set to a value given by the module parameter flow_control	RW	0x1

2.12.5.35. Forward Flow Control Frames

Offset: 0x706

Forward Flow Control Frames Fields

Bit	Name	Description	Access	Reset
0	rx_pause_fwd	**Forward Flow Control Frames** Sets whether PAUSE and PFC frames are sent to the MAC Client Interface • 1: Forwards all flow control frames to the application • 0: Does not forward flow control frames that match the RX destination address for flow control to the application Be careful when turning off Flow Control frame forwarding • This feature requires EHIP to be in a mode with the MAC turned on • When flow control forwarding is turned off, flow control frames will be dropped regardless of whether flow control processing is turned on • Packets are considered to be flow control if they have T/L = 16'h8808, MAC Control opcode =0x0001 (PAUSE) or 0x0101 (PFC) and their destination address matches rx_pause_daddr • Flow Control packets are only processed by the MAC if they are also exactly 72 bytes long (including Preamble) and are error free • When Flow Control forwarding is turned off, Flow control packets will be dropped and not processed • When this setting is changed, the RX MAC must be reset • At power-on, this register defaults to 0 • When i_csr_rst_n is asserted, this register is set to the value given by the module parameter forward_rx_pause_requests	RW	0x0

2.12.5.36. Lower 4 bytes of the Destination address for RX Pause Frames

Offset: 0x707

Lower 4 bytes of the Destination address for RX Pause Frames Fields

Bit	Name	Description	Access	Reset
31:0	rx_pause_daddrl	**Lower bytes of the RX Flow Control Destination Address** Lower 4 bytes of the 6 byte destination address that must be found in incoming SFC and PFC frames.	RW	0xB83537 15

Bit	Name	Description	Access	Reset
		• This feature requires EHIP to be in a mode with the MAC turned on • When this setting is changed, the RX MAC must be reset • At power-on, this register defaults to 32'hB8353715 • When i_csr_rst_n is asserted, this register is set to the value given by the module parameter rx_pause_daddr[31:0]		

2.12.5.37. Higher 2 bytes of the Destination address for RX Pause Frames

Offset: 0x708

Higher 2 bytes of the Destination address for RX Pause Frames Fields

Bit	Name	Description	Access	Reset
15:0	rx_pause_daddrh	**Higher bytes of the RX Flow Control Destination Address** Higher 2 bytes of the 6 byte destination address that must be found in incoming SFC and PFC frames • This feature requires EHIP to be in a mode with the MAC turned on • When this setting is changed, the RX MAC must be reset • At power-on, this register defaults to 16'h0F2C • When i_csr_rst_n is asserted, this register is set to the value given by the module parameter rx_pause_daddr[47:32]	RW	0x00000F2C

2.12.5.38. RX Flow Control Feature Configuration

Offset: 0x709

RX Flow Control Feature Configuration Fields

Bit	Name	Description	Access	Reset
1	en_pfc	**Enable Priority Flow Control RX** 1: Enable Priority Flow Control • After power-on, this reset is set to 0 • After i_csr_rst_n, this register is set to a value given by the module parameter flow_control	RW	0x0
0	en_sfc	**Enable Standard Flow Control RX** 1: Enable Standard Flow Control (link PAUSE) • After power-on, this reset is set to 0 • After i_csr_rst_n, this register is set to a value given by the module parameter flow_control	RW	0x0

2.12.6. TX Statistics Counter Registers

2.12.6.1. TX Statistics Registers

Table 74. Transmit Side Statistics Registers

Address	Name-	Description	Access
0x800	TX_FRAGMENTS_31_0	Number of transmitted frames less than 64 bytes and reporting a CRC error (lower 32 bits)	RO
0x801	TX_FRAGMENTS_63_32	Number of transmitted frames less than 64 bytes and reporting a CRC error (upper 32 bits)	RO
0x802	TX_JABBERS_31_0	Number of transmitted oversized frames reporting a CRC error (lower 32 bits)	RO
0x803	TX_JABBERS_63_32	Number of transmitted oversized frames reporting a CRC error (upper 32 bits)	RO
0x804	TX_FCSERR_31_0	Number of transmitted packets with FCS errors. (lower 32 bits)	RO
0x805	TX_FCSERR_63_32	Number of transmitted packets with FCS errors. (upper 32 bits)	RO
0x806	TX_CRCERR_OKPKT_31_0	Number of frames of any size that are malformed but are neither undersized or oversized with a CRC error (lower 32 bits)	RO
0x807	TX_CRCERR_OKPKT_63_32	Number of frames of any size that are malformed but are neither undersized or oversized with a CRC error (upper 32 bits)	RO
0x808	TX_MCAST_DATA_ERR_31_0	Number of errored multicast frames transmitted, excluding control frames (lower 32 bits)	RO
0x809	TX_MCAST_DATA_ERR_63_32	Number of errored multicast frames transmitted, excluding control frames (upper 32 bits)	RO
0x80A	TX_BCAST_DATA_ERR_31_0	Number of errored broadcast frames transmitted, excluding control frames (lower 32 bits)	RO
0x80B	TX_BCAST_DATA_ERR_63_32	Number of errored broadcast frames transmitted, excluding control frames (upper 32 bits)	RO
0x80C	TX_UCAST_DATA_ERR_31_0	Number of errored unicast frames transmitted, excluding control frames (lower 32 bits)	RO
0x80D	TX_UCAST_DATA_ERR_63_32	Number of errored unicast frames transmitted, excluding control frames (upper 32 bits)	RO
0x80E	TX_MCAST_CTRL_ERR_31_0	Number of errored multicast control frames transmitted (lower 32 bits)	RO
0x80F	TX_MCAST_CTRL_ERR_63_32	Number of errored multicast control frames transmitted (upper 32 bits)	RO

continued...

Address	Name-	Description	Access
0x810	TX_BCAST_CTRL_ERR_31_0	Number of errored broadcast control frames transmitted (lower 32 bits)	RO
0x811	TX_BCAST_CTRL_ERR_63_32	Number of errored broadcast control frames transmitted (upper 32 bits)	RO
0x812	TX_UCAST_CTRL_ERR_31_0	Number of errored unicast control frames transmitted (lower 32 bits)	RO
0x813	TX_UCAST_CTRL_ERR_63_32	Number of errored unicast control frames transmitted (upper 32 bits)	RO
0x814	TX_PAUSE_ERR_31_0	Number of errored pause frames transmitted (lower 32 bits)	RO
0x815	TX_PAUSE_ERR_63_32	Number of errored pause frames transmitted (upper 32 bits)	RO
0x816	TX_64B_31_0	Number of 64-byte transmitted frames (lower 32 bits), including the CRC field but excluding the preamble and SFD bytes	RO
0x817	TX_64B_63_32	Number of 64-byte transmitted frames (upper 32 bits), including the CRC field but excluding the preamble and SFD bytes	RO
0x818	TX_65to127B_31_0	Number of transmitted frames between 65–127 bytes (lower 32 bits)	RO
0x819	TX_65to127B_63_32	Number of transmitted frames between 65–127 bytes (upper 32 bits)	RO
0x81A	TX_128to255B_31_0	Number of transmitted frames between 128–255 bytes (lower 32 bits)	RO
0x81B	TX_128to255B_63_32	Number of transmitted frames between 128–255 bytes (upper 32 bits)	RO
0x81C	TX_256to511B_31_0	Number of transmitted frames between 256–511 bytes (lower 32 bits)	RO
0x81D	TX_256to511B_63_32	Number of transmitted frames between 256–511 bytes (upper 32 bits)	RO
0x81E	TX_512to1023B_31_0	Number of transmitted frames between 512–1023 bytes (lower 32 bits)	RO
0x81F	TX_512to1023B_63_32	Number of transmitted frames between 512–1023 bytes (upper 32 bits)	RO
0x820	TX_1024to1518B_31_0	Number of transmitted frames between 1024–1518 bytes (lower 32 bits)	RO
0x821	TX_1024to1518B_63_32	Number of transmitted frames between 1024–1518 bytes (upper 32 bits)	RO

continued...

Address	Name-	Description	Access
0x822	TX_1519toMAXB_31_0	Number of transmitted frames of size between 1519 bytes and the number of bytes specified in the MAX_TX_SIZE_CONFIG register (lower 32 bits)	RO
0x823	TX_1519toMAXB_63_32	Number of transmitted frames of size between 1519 bytes and the number of bytes specified in the MAX_TX_SIZE_CONFIG register (upper 32 bits)	RO
0x824	TX_OVERSIZE_31_0	Number of oversized frames (frames with more bytes than the number specified in the MAX_TX_SIZE_CONFIG register) transmitted (lower 32 bits)	RO
0x825	TX_OVERSIZE_63_32	Number of oversized frames (frames with more bytes than the number specified in the MAX_TX_SIZE_CONFIG register) transmitted (upper 32 bits)	RO
0x826	TX_MCAST_DATA_OK_31_0	Number of valid multicast frames transmitted, excluding control frames (lower 32 bits)	RO
0x827	TX_MCAST_DATA_OK_63_32	Number of valid multicast frames transmitted, excluding control frames (upper 32 bits)	RO
0x828	TX_BCAST_DATA_OK_31_0	Number of valid broadcast frames transmitted, excluding control frames (lower 32 bits)	RO
0x829	TX_BCAST_DATA_OK_63_32	Number of valid broadcast frames transmitted, excluding control frames (upper 32 bits)	RO
0x82A	TX_UCAST_DATA_OK_31_0	Number of valid unicast frames transmitted, excluding control frames (lower 32 bits)	RO
0x82B	TX_UCAST_DATA_OK_63_32	Number of valid unicast frames transmitted, excluding control frames (upper 32 bits)	RO
0x82C	TX_MCAST_CTRL_OK_31_0	Number of valid multicast frames transmitted, excluding data frames (lower 32 bits)	RO
0x82D	TX_MCAST_CTRL_OK_63_32	Number of valid multicast frames transmitted, excluding data frames (upper 32 bits)	RO
0x82E	TX_BCAST_CTRL_OK_31_0	Number of valid broadcast frames transmitted, excluding data frames (lower 32 bits)	RO
0x82F	TX_BCAST_CTRL_OK_63_32	Number of valid broadcast frames transmitted, excluding data frames (upper 32 bits)	RO
0x830	TX_UCAST_CTRL_OK_31_0	Number of valid unicast frames transmitted, excluding data frames (lower 32 bits)	RO

continued...

Address	Name-	Description	Access
0x831	TX_UCAST_CTRL_OK_63_32	Number of valid unicast frames transmitted, excluding data frames (upper 32 bits)	RO
0x832	TX_PAUSE_31_0	Number of valid pause frames transmitted (lower 32 bits)	RO
0x833	TX_PAUSE_63_32	Number of valid pause frames transmitted (upper 32 bits)	RO
0x834	TX_RNT_31_0	Number of transmitted runt packets (lower 32 bits). The IP core does not transmit frames of length less than nine bytes. The IP core pads frames of length nine bytes to 64 bytes to extend them to 64 bytes. Therefore, this counter does not increment in normal operating conditions.	RO
0x835	TX_RNT_63_32	Number of transmitted runt packets (upper 32 bits). The IP core does not transmit frames of length less than nine bytes. The IP core pads frames of length nine bytes to 64 bytes to extend them to 64 bytes. Therefore, this counter does not increment in normal operating conditions.	RO
0x836	TX_st_31_0	Number of TX frame starts (lower 32 bits)	RO
0x837	TX_st_63_32	Number of TX frame starts (upper 32 bits)	RO
0x838	TX_lenerr_31_0	Number of frames where the length of the frame advertised in the L/T field was larger than the frame that was received (lower 32 bits). Length checking must be enabled	RO
0x839	TX_lenerr_63_32	Number of frames where the length of the frame advertised in the L/T field was larger than the frame that was received (upper 32 bits). Length checking must be enabled	RO
0x83A	TX_pfc_err_31_0	Number of malformed TX PFC frames with CRC errors (lower 32 bits)	RO
0x83B	TX_pfc_err_63_32	Number of malformed TX PFC frames with CRC errors (upper 32 bits)	RO
0x83C	TX_pfc_31_0	Number of TX PFC frames without error (lower 32 bits)	RO
0x83D	TX_pfc_63_32	Number of TX PFC frames without error (upper 32 bits)	RO
0x840	txstat_revid	Returns a 4 byte value indicating the revision of this design	RO
0x841	txstat_scratch	32 bits of scratch register space for testing	RO
0x842 to 0x844	Reserved		
0x845	TX_CNTR_CONFIG	Bits[2:0]: Configuration of TX statistics counters:	RW

continued...

Address	Name-	Description	Access
		• Bit[2] = 1: Freeze stats CSRs so that all TX Stats values read from the registers will be from the same moment: — Note that the actual stats collection counters are not frozen, but because they are all 'read' at the time of the freeze, they are cleared. — If a shadow request is started while snapshot is active, a new capture will be executed. — Likewise, if a shadow request is active while snapshot is asserted, a new capture will be executed. — While either a shadow request or a capture is active, tx_shadow_on will be high. — Snapshot and shadow requests apply to several of the RX PCS counters as well as MAC statistics. • Bit[1] = 1: Reset the parity error bit in cntr_TX_status. — Parity error bit will remain in reset until rst_tx_parity is set back to 0 • Bit[0] = 1: Reset all TX Stats counters — TX stats will stay in reset until reset is set back to 0. — Reset also applies when snapshot or shadow is active, and will clear the AVMM visible registers. — `rst_tx_stats` does not clear the parity error bit. Bits[31:3] are Reserved.	
			continued...

Address	Name-	Description	Access
0x846	TX_CNTR_STATUS	• Bit[1] =1: The CSRs for the TX Statistics are currently frozen, and holding the statistic values from the last time a shadow request was made. Shadow on is asserted for either a shadow request or a snapshot • Bit[0] = 1: A parity error was detected on at least one of the statistics counters since the last time this bit was cleared — Statistics counter values are stored periodically by EHIP for long term storage. — Whenever a counter value is stored, a parity value is calculated for the new value. — Whenever a stats value is updated, the parity value of the old value is calculated. If it doesn't match the stored value, the sticky parity error bit is asserted. — If tx_parity_err is high, it means sometime in the past, a parity error was detected on the stats memory. Bits[31:2] are Reserved.	RO
0x847–0x85F	Reserved		
0x860	TX_Payload_OctetsOK_31_0	Number of transmitted payload bytes in frames with no FCS, undersized, oversized, or payload length errors. • When TX VLAN/SVLAN detection is enabled, VLAN/SVLAN header bytes are also removed from the count • For single lane EHIP modules (10G or 25G), packets that start within 4 bytes of a the previous packet's TERM are not counted (malformed).	RO
0x861	TX_Payload_OctetsOK_63_32		RO
0x862	TX_Frame_OctetsOK_31_0	Number of transmitted bytes in frames with no FCS, undersized, oversized, or payload length errors. For single lane EHIP modules (10G or 25G), packets that start within 4 bytes of a the previous packet's TERM are not counted (malformed).	RO
0x863	TX_Frame_OctetsOK_63_32		RO
0x864	TX_Malformed_CTRL_31_0	Records the number of TX packets that were malformed. • A packet is malformed if it is interrupted by an MII Control byte other than TERM and ERROR • Packets that have ERROR control bytes but end with a TERM are not considered malformed • For single lane EHIP modules (10G or 25G), packets that start within 4 bytes of a the previous packet's TERM are not counted (malformed).	RO
0x865	TX_Malformed_CTRL_63_32		RO

continued...

Address	Name-	Description	Access
0x866	TX_Dropped_CTRL_31_0	Records the number of TX packets dropped due to errors. • The TXMAC automatically pads short frames, except when *i_skip_crc* is asserted from the packet • When CRC is skipped, if the packet is shorter than 21 bytes, it will be counted as a TX dropped packet	RO
0x867	TX_Dropped_CTRL_63_32		RO
0x868	TX_BadLt_CTRL_31_0	Records the number of TX frames that arrived with a Length/Type field that was neither a length nor a type. • L/T is considered to be a Length field if the value in the field is 16'd1500 or less • L/T is considered to be a Type field if the value in the field is 16'd1536 or more • If a packet has a L/T field with value between 16'd1501 and 16'd1535 (inclusive), the L/T field is considered bad, and the counter is incremented *Note:* If TX_VLAN/SVLAN detection is turned on, it is the L/T field inside the VLAN/SVLAN header that is evaluated.	RO
0x869	TX_BadLt_CTRL_63_32		RO

2.12.7. RX Statistics Counter Registers

2.12.7.1. RX Statistics Registers

Table 75. Receive Side Statistics Registers

Address	Name	Description	Access
0x900	RX_FRAGMENTS_31_0	Number of received frames less than 64 bytes and reporting a CRC error (lower 32 bits)	RO
0x901	RX_FRAGMENTS_63_32	Number of received frames less than 64 bytes and reporting a CRC error (upper 32 bits)	RO
0x902	RX_JABBERS_31_0	Number of received oversized frames reporting a CRC error (lower 32 bits)	RO
0x903	RX_JABBERS_63_32	Number of received oversized frames reporting a CRC error (upper 32 bits)	RO
0x904	RX_FCSERR_31_0	Number of received packets with FCS errors. This register maintains a count of the number of pulses on the l<n>_rx_fcs_error or rx_fcs_error output signal (lower 32 bits)	RO
0x905	RX_FCSERR_63_32	Number of received packets with FCS errors. This register maintains a count of the number of pulses on the l<n>_rx_fcs_error output signal (upper 32 bits)	RO

continued...

Address	Name	Description	Access
0x906	RX_CRCERR_OKPKT_31_0	Number of received frames with a frame of length at least 64, with CRC error (lower 32 bits)	RO
0x907	RX_CRCERR_OKPKT_63_32	Number of received frames with a frame of length at least 64, with CRC error (upper 32 bits)	RO
0x908	RX_MCAST_DATA_ERR_31_0	Number of errored multicast frames received, excluding control frames (lower 32 bits)	RO
0x909	RX_MCAST_DATA_ERR_63_32	Number of errored multicast frames received, excluding control frames (upper 32 bits)	RO
0x90A	RX_BCAST_DATA_ERR_31_0	Number of errored broadcast frames received, excluding control frames (lower 32 bits)	RO
0x90B	RX_BCAST_DATA_ERR_63_32	Number of errored broadcast frames received, excluding control frames (upper 32 bits)	RO
0x90C	RX_UCAST_DATA_ERR_31_0	Number of errored unicast frames received, excluding control frames (lower 32 bits)	RO
0x90D	RX_UCAST_DATA_ERR_63_32	Number of errored unicast frames received, excluding control frames (upper 32 bits)	RO
0x90E	RX_MCAST_CTRL_ERR_31_0	Number of errored multicast control frames received (lower 32 bits)	RO
0x90F	RX_MCAST_CTRL_ERR_63_32	Number of errored multicast control frames received (upper 32 bits)	RO
0x910	RX_BCAST_CTRL_ERR_31_0	Number of errored broadcast control frames received (lower 32 bits)	RO
0x911	RX_BCAST_CTRL_ERR_63_32	Number of errored broadcast control frames received (upper 32 bits)	RO
0x912	RX_UCAST_CTRL_ERR_31_0	Number of errored unicast control frames received (lower 32 bits)	RO
0x913	RX_UCAST_CTRL_ERR_63_32	Number of errored unicast control frames received (upper 32 bits)	RO
0x914	RX_PAUSE_ERR_31_0	Number of errored pause frames received (lower 32 bits)	RO
0x915	RX_PAUSE_ERR_63_32	Number of errored pause frames received (upper 32 bits)	RO
0x916	RX_64B_31_0	Number of 64-byte received frames (lower 32 bits), including the CRC field but excluding the preamble and SFD bytes	RO
0x917	RX_64B_63_32	Number of 64-byte received frames (upper 32 bits), including the CRC field but excluding the preamble and SFD bytes	RO
0x918	RX_65to127B_31_0	Number of received frames between 65–127 bytes (lower 32 bits)	RO
0x919	RX_65to127B_63_32	Number of received frames between 65–127 bytes (upper 32 bits)	RO
0x91A	RX_128to255B_31_0	Number of received frames between 128 –255 bytes (lower 32 bits)	RO
0x91B	RX_128to255B_63_32	Number of received frames between 128 –255 bytes (upper 32 bits)	RO
0x91C	RX_256to511B_31_0	Number of received frames between 256 –511 bytes (lower 32 bits)	RO

continued...

Address	Name	Description	Access
0x91D	RX_256to511B_63_32	Number of received frames between 256 –511 bytes (upper 32 bits)	RO
0x91E	RX_512to1023B_31_0	Number of received frames between 512–1023 bytes (lower 32 bits)	RO
0x91F	RX_512to1023B_63_32	Number of received frames between 512 –1023 bytes (upper 32 bits)	RO
0x920	RX_1024to1518B_31_0	Number of received frames between 1024–1518 bytes (lower 32 bits)	RO
0x921	RX_1024to1518B_63_32	Number of received frames between 1024–1518 bytes (upper 32 bits)	RO
0x922	RX_1519toMAXB_31_0	Number of received frames between 1519 bytes and the maximum size defined in the MAX_RX_SIZE_CONFIG register (lower 32 bits)	RO
0x923	RX_1519toMAXB_63_32	Number of received frames between 1519 bytes and the maximum size defined in the RXMAC_SIZE_CONFIG register (upper 32 bits)	RO
0x924	RX_OVERSIZE_31_0	Number of oversized frames (frames with more bytes than the number specified in the RXMAC_SIZE_CONFIG register) received (lower 32 bits)	RO
0x925	RX_OVERSIZE_63_32	Number of oversized frames (frames with more bytes than the number specified in the RXMAC_SIZE_CONFIG register) received (upper 32 bits)	RO
0x926	RX_MCAST_DATA_OK_31_0	Number of valid multicast frames received, excluding control frames (lower 32 bits)	RO
0x927	RX_MCAST_DATA_OK_63_32	Number of valid multicast frames received, excluding control frames (upper 32 bits)	RO
0x928	RX_BCAST_DATA_OK_31_0	Number of valid broadcast frames received, excluding control frames (lower 32 bits)	RO
0x929	RX_BCAST_DATA_OK_63_32	Number of valid broadcast frames received, excluding control frames (upper 32 bits)	RO
0x92A	RX_UCAST_DATA_OK_31_0	Number of valid unicast frames received, excluding control frames (lower 32 bits)	RO
0x92B	RX_UCAST_DATA_OK_63_32	Number of valid unicast frames received, excluding control frames (upper 32 bits)	RO
0x92C	RX_MCAST_CTRL_OK_31_0	Number of valid multicast frames received, excluding data frames (lower 32 bits)	RO
0x92D	RX_MCAST_CTRL_OK_63_32	Number of valid multicast frames received, excluding data frames (upper 32 bits)	RO
0x92E	RX_BCAST_CTRL_OK_31_0	Number of valid broadcast frames received, excluding data frames (lower 32 bits)	RO
0x92F	RX_BCAST_CTRL_OK_63_32	Number of valid broadcast frames received, excluding data frames (upper 32 bits)	RO
0x930	RX_UCAST_CTRL_OK_31_0	Number of valid unicast frames received, excluding data frames (lower 32 bits)	RO
0x931	RX_UCAST_CTRL_OK_63_32	Number of valid unicast frames received, excluding data frames (upper 32 bits)	RO

continued...

Address	Name	Description	Access
0x932	RX_PAUSE_31_0	Number of received pause frames, with or without error (lower 32 bits)	RO
0x933	RX_PAUSE_63_32	Number of received pause frames, with or without error (upper 32 bits)	RO
0x934	RX_RNT_31_0	Number of received runt packets (lower 32 bits) A run is a packet of size less than 64 bytes but greater than eight bytes. If a packet is eight bytes or smaller, it is considered a decoding error and not a runt frame, and the IP core does not flag it nor count it as a runt.	RO
0x935	RX_RNT_63_32	Number of received runt packets (upper 32 bits) A run is a packet of size less than 64 bytes but greater than eight bytes. If a packet is eight bytes or smaller, it is considered a decoding error and not a runt frame, and the IP core does not flag it nor count it as a runt.	RO
0x936	RX_st_31_0	Number of RX frame starts (lower 32 bits)	RO
0x937	RX_st_63_32	Number of RX frame starts (upper 32 bits)	RO
0x938	RX_lenerr_31_0	Number of RX length errors (lower 32 bits)	RO
0x939	RX_lenerr_63_32	Number of RX length errors (upper 32 bits)	RO
0x93A	RX_pfc_err_31_0	Number of RX PFC frame with CRC error (lower 32 bits)	RO
0x93B	RX_pfc_err_63_32	Number of RX PFC frame with CRC error (upper 32 bits)	RO
0x93C	RX_pfc_31_0	Number of RX PFC frames without error (lower 32 bits)	RO
0x93D	RX_pfc_63_32	Number of RX PFC frames without error (upper 32 bits)	RO
0x93E to 0x93F	Reserved		
0x940	rxstat_revid	Returns a 4 byte value indicating the revision of this design	RO
0x941	rxstat_scratch	32 bits of scratch register space for testing	RW
0x942 to 0x944	Reserved		
0x945	RX_CNTR_CONFIG	Bits[2:0]: Configuration of RX statistics counters:	RW

continued...

Address	Name	Description	Access
		• Bit[2] = 1: Freeze stats CSRs so that all RX Stats values read from the registers will be from the same moment. — Note that the actual stats collection counters are not frozen, but because they are all 'read' at the time of the freeze, they are cleared. — If a shadow request is started while snapshot is active, a new capture will be executed. — Likewise, if a shadow request is active while snapshot is asserted, a new capture will be executed. — While either a shadow request or a capture is active, rx_shadow_on will be high. — Snapshot and shadow requests apply to several of the RX PCS counters as well as MAC statistics • Bit[1] = 1:Reset the parity error bit in RX Statistics Counter Status — Parity error bit will remain in reset until rst_rx_parity is set back to 0 • Bit[0] = 1: Reset all RX Stats counters — RX stats will stay in reset until reset is set back to 0 — Reset also applies when snapshot or shadow is active, and will clear the AVMM visible registers — `rst_rx_stats` does not clear the parity error bit Bits[31:3] are Reserved.	
0x946	RX_CNTR_STATUS	• Bit[1] = 1: The CSRs for the RX Statistics are currently frozen, and holding the Stats values from the last time a shadow request was made. • Bit[0] = 1: A parity error was detected on at least one of the statistics counters since the last time this bit was cleared. Bits [31:2] are Reserved.	RO
0x947–0x95F	Reserved		
0x960	RX_Payload_OctetsOK_31_0	Number of received payload bytes in frames with no FCS, undersized, oversized, or payload length errors. When RX VLAN/SVLAN detection is enabled, VLAN/SVLAN header bytes are also removed from the count. Use snapshot or shadow to freeze the count before reading it to avoid value change while reading the register.	RO
0x961	RX_Payload_OctetsOK_63_32		RO
0x962	RX_Frame_OctetsOK_31_0	Number of received bytes in frames with no FCS, undersized, oversized, or payload length errors. Use snapshot or shadow to freeze the count before reading it to avoid value change while reading the register.	RO
0x963	RX_Frame_OctetsOK_63_32		RO

2.12.8. 1588 PTP Registers

The 1588 PTP registers together with the 1588 PTP signals process and provide Precision Time Protocol (PTP) timestamp information as defined in the *IEEE 1588-2008 Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Standard*. The 1588 PTP module provides you the support to implement the 1588 Precision Time Protocol in your design.

Table 76. TX 1588 PTP Registers

Addr	Name	Bit	Description	HW Reset Value	Access
0xA00	TXPTP_REVID	[31:0]	IP core revision ID.	0x0504_2018	RO
0xA01	TXPTP_SCRATCH	[31:0]	Scratch register available for testing.	32'b0	RW
0xA05	TX_PTP_CLK_PERIOD	[19:0]	20-bit register holding the datapath clock period in the IEEE 1588v2 format. This value is used to estimate delays through the datapath. Period of the 402.83 MHz EHIP clock in 1588v2 format. Bits[19:16]: nanoseconds (ns) Bits[15:0]: fractions of nanosecond	0x27B81	RW
0xA0A	TX_PTP_EXTRA_LATENCY	[31:0]	User-defined extra latency the IP core adds to outgoing TX 1-step and 2-step timestamps. [31]: Sign bit, set to 1 for negative extra latency Bits[30:16]: Extra latency in nanoseconds Bits[15:0]: Extra latency in fractions of nanosecond (value/17'h10000) For example, to set a TX extra latency of +2.5 ns, set `tx_ptp_extra_latency` to 32'h00028000.	32'b0	RW
0xA0D	PTP_DEBUG	[31:0]	Controls a small number of PTP debug features. • Bit[0] = 1: instead of inserting PTP field values in TX packets when executing PTP TX 1-step commands, insert fixed values — Insert 8'hAA in all bytes that would have been used for timestamp bytes — Insert 8'hBB in all bytes that would have been used for correction field bytes — Insert 8'hCC in all bytes that would have been used for Extension bytes • Bit[31:1]: Reserved	0x0	RW
0xB10	TX_UI_REG	[31:0]	Sets the time of a single serial bit on the TX Serial interface. Sets the time for a single TX serial bit. This time is used to generate TX timestamp estimates. • Bit[31:24]: Nanoseconds field for the time of a single serial TX bit • [23:0]: Fractional nanoseconds field for the time of a single serial TX bit	0x9EE01/0x18D302	RW
0xB11	RX_UI_REG	[31:0]	Sets the time of a single serial bit on the RX Serial interface. Sets the time for a single RX serial bit. This time is used to generate RX timestamp estimates. • Bit[31:24]: Nanoseconds field for the time of a single serial RX bit • [23:0]: Fractional nanoseconds field for the time of a single serial RX bit	0x9EE01/0x18D302	RW

Table 77. RX 1588 PTP Registers

Addr	Name	Bit	Description	HW Reset Value	Access
0xB00	RXPTP_REVID	[31:0]	IP core revision ID.	0x0504 2018	RO
0xB01	RXPTP_SCRATCH	[31:0]	Scratch register available for testing.	32'b0	RW
0xB06	RX_PTP_EXTRA_LATENCY	[31:0]	32-bit specifying extra latency that EHIP adds to the incoming RX timestamps. • Bit[31]: Sign bit, set to 1 for negative latency • Bits[30:16]: Extra latency in nanoseconds • Bits[15:0]: Extra latency in fractions of a nanosecond For example, to set an RX extra latency of 5.00 ns, set rx_ptp_extra_latency to 32'h00050000.	32'b0	RW

Table 78. 10G/25G PTP PPM UI Adjustment Registers

Addr	Name	Bit	Description	HW Default Value	Access
0xB19	TAM_SNAPSHOT	[0]	Time value control register. When set, the values of the reference time value and AM count number are recorded in TX_TAM and TX_COUNT registers.	0x0	RW
0xB1A	TX_TAM_L	[31:0]	This register represents the lower 32-bits of the TX TAM value. TX_TAM[31:0]: • Bits[31:16]: Nanosecond field for the LSB • Bits[15:0]: Fractional nanoseconds field	0x0	RO
0xB1B	TX_TAM_H	[15:0]	This register represents the upper 16-bits of the TX_TAM value. TX_TAM[47:32]: Nanosecond field for the MSB	0x0	RO
0xB1C	TX_COUNT	[15:0]	Contains the TX_AM count value.	0x0	RO
0xB1D	RX_TAM_L	[31:0]	This register represents the lower 32-bits of the RX TAM value. RX_TAM[31:0]: • Bits[31:16]: Nanosecond field for the LSB • Bits[15:0]: Fractional nanoseconds field	0x0	RO
0xB1E	RX_TAM_H	[15:0]	This register represents the upper 16-bits of the RX_TAM value. RX_TAM[47:32]: Nanosecond field for the MSB	0x0	RO
0xB1F	RX_COUNT	[15:0]	Contains the RX_AM count value.	0x0	RO

Table 79. 100G PTP PPM UI Adjustment Registers

Addr	Name	Description	HW Default Value	Access
0xC00	mlptp_tx_ui	UI time in ns and fns. Sets the time for a single TX serial bit. This time is used to generate TX timestamp estimates.	0x0009 EE01	RW

continued...

Addr	Name	Description	HW Default Value	Access
		• Bit [31:24]: Nanoseconds field for the time of a single serial TX bit • Bit [23:0]: Fractional nanoseconds field for the time of a single serial TX bit		
0xC01	mlptp_rx_ui	UI time in ns and fns. Sets the time for a single RX serial bit. This time is used to generate RX timestamp estimates. • Bit [31:24]: Nanoseconds field for the time of a single serial RX bit • Bit [23:0]: Fractional nanoseconds field for the time of a single serial RX bit	0x0CCC CCCD	RW
0xC10	vl0_offset_data_0	• Bit [31]: vl_offset_data_valid—Indicate VL Offset Data is valid to be used for calculation • Bit [23:10]: AM_COUNT—Alignment Marker Count • Bit [9:5]: BA_POS—Block Aligner Position • Bit [4:3]: BA_PHASE—Block Aligner Phase • Bit [2:0]: GBSTATE—Gearbox state for the selected virtual lane	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx x000	RO
0xC12	vl1_offset_data_0			
0xC14	vl2_offset_data_0			
0xC16	vl3_offset_data_0			
0xC18	vl4_offset_data_0			
0xC1A	vl5_offset_data_0			
0xC1C	vl6_offset_data_0			
0xC1E	vl7_offset_data_0			
0xC20	vl8_offset_data_0			
0xC22	vl9_offset_data_0			
0xC24	vl10_offset_data_0			
0xC26	vl11_offset_data_0			
0xC28	vl12_offset_data_0			
0xC2A	vl13_offset_data_0			
0xC2C	vl14_offset_data_0			
0xC2E	vl15_offset_data_0			
0xC30	vl16_offset_data_0			
0xC32	vl17_offset_data_0			
0xC34	vl18_offset_data_0			
0xC36	vl19_offset_data_0			
0xC11	vl0_offset_data_1	• Bit [31]: vl_offset_data_valid—Indicate VL Offset Data is valid to be used for calculation. • Bit [11:10]: LOCAL_PL—Local Physical Lane Index Number • Bit [9:5]: REMOTE_VL—Remote Virtual Lane Index Number • Bit [4:0]: LOCAL_VL—Local Virtual Lane Index Number	0bxxxx xxxx xxxx xxxx xxxx xxxx xxx0 0000	RO
0xC13	vl1_offset_data_1			
0xC15	vl2_offset_data_1			
0xC17	vl3_offset_data_1			
0xC19	vl4_offset_data_1			
0xC1B	vl5_offset_data_1			
0xC1D	vl6_offset_data_1			
0xC1F	vl7_offset_data_1			
0xC21	vl8_offset_data_1			

continued...

Addr	Name	Description	HW Default Value	Access
0xC23	vl9_offset_data_1			
0xC25	vl10_offset_data_1			
0xC27	vl11_offset_data_1			
0xC29	vl12_offset_data_1			
0xC2B	vl13_offset_data_1			
0xC2D	vl14_offset_data_1			
0xC2F	vl15_offset_data_1			
0xC31	vl16_offset_data_1			
0xC33	vl17_offset_data_1			
0xC35	vl18_offset_data_1			
0xC37	vl19_offset_data_1			
0xC40	vl0_offset_cfg_0	• Bit [6:5]: LOCAL_PL—Physical Lane for the VL Offset • Bit [4:0]: REMOTE_VL—Virtual Lane Index for the VL Offset	0bxxxx xxxx xxxx xxxx xxxx xxxx xxx0 0000	RW
0xC42	vl1_offset_cfg_0			
0xC44	vl2_offset_cfg_0			
0xC46	vl3_offset_cfg_0			
0xC48	vl4_offset_cfg_0			
0xC4A	vl5_offset_cfg_0			
0xC4C	vl6_offset_cfg_0			
0xC4E	vl7_offset_cfg_0			
0xC50	vl8_offset_cfg_0			
0xC52	vl9_offset_cfg_0			
0xC54	vl10_offset_cfg_0			
0xC56	vl11_offset_cfg_0			
0xC58	vl12_offset_cfg_0			
0xC5A	vl13_offset_cfg_0			
0xC5C	vl14_offset_cfg_0			
0xC5E	vl15_offset_cfg_0			
0xC60	vl16_offset_cfg_0			
0xC62	vl17_offset_cfg_0			
0xC64	vl18_offset_cfg_0			
0xC66	vl19_offset_cfg_0			
0xC41	vl0_offset_cfg_1	• Bit [31]: `sign_bit`—Sign bit. • Bit [30:16]: `offset_ns`—Virtual lane offset (nanoseconds) • Bit [15:0]: `offset_frac_ns`—Virtual lane offset (fractional nanoseconds)		RW
0xC43	vl1_offset_cfg_1			
0xC45	vl2_offset_cfg_1			
0xC47	vl3_offset_cfg_1			
0xC49	vl4_offset_cfg_1			

continued...

Addr	Name	Description	HW Default Value	Access
0xC4B	vl5_offset_cfg_1			
0xC4D	vl6_offset_cfg_1			
0xC4F	vl7_offset_cfg_1			
0xC51	vl8_offset_cfg_1			
0xC53	vl9_offset_cfg_1			
0xC55	vl10_offset_cfg_1			
0xC57	vl11_offset_cfg_1			
0xC59	vl12_offset_cfg_1			
0xC5B	vl13_offset_cfg_1			
0xC5D	vl14_offset_cfg_1			
0xC5F	vl15_offset_cfg_1			
0xC61	vl16_offset_cfg_1			
0xC63	vl17_offset_cfg_1			
0xC65	vl18_offset_cfg_1			
0xC67	vl19_offset_cfg_1			
0xC70	mlptp_tam_snapshot	Time value control register. When set, the values of the reference time value and AM count number are recorded TX and RX registers. Request TAM snapshot to calculate unit interval (ui). TX and RX TAM snapshot are independent: • Bit [0]: `tx_tam_snapshot` Set this bit to 1'b1 to request TX TAM snapshot. Asserting this bit generates a single pulse to hardware. • Bit [1]: `rx_tam_snapshot` Set this bit to 1'b1 to request RX TAM snapshot. Asserting this bit generates a single pulse to hardware.	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx xxx0	RW
0xC71	mlptp_tx_tam_l	TX TAM lower bits. Bit [31:0] of TAM Result returned upon TAM snapshot request, consists of TAM partial bits.	0x0000 0000	RO
0xC72	mlptp_tx_tam_h	TX TAM upper bits. Bit [15:0] of TAM Result returned upon TAM snapshot request, consists of TAM partial bits.	0xXXXX 0000	RO
0xC73	mlptp_tx_am_count	TX TAM counter. Bit [15:0]—16 bits of TAM counter. Result returned upon TAM snapshot request, consists of TAM partial bits.	0xXXXX 0000	RO
0xC74	mlptp_rx_tam_l	RX TAM lower bits. Bit [31:0] of TAM Result returned upon TAM snapshot request, consists of TAM partial bits.	0x0000 0000	RO
0xC75	mlptp_rx_tam_h	RX TAM upper bits. Bit [15:0] of TAM Result returned upon TAM snapshot request, consists of TAM partial bits.	0xXXXX 0000	RO

continued...

Addr	Name	Description	HW Default Value	Access
0xC76	mlptp_rx_am_count	RX TAM counter. Bit [15:0]—16 bits of TAM counter. Result returned upon TAM snapshot request, consists of TAM partial bits.	0xXXXX 0000	RO
0xC77	mlptp_tx_ref_lane	Physical lane that TAM associates with. • Bit [2:0]: `tx_ref_lane` **TX reference lane**—3 bit number of TX physical lane (any can be reference): — Resetting TX datapath, or the entire core will clear this bit to 0 Only applicable to multi-lane designs.	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx x000	WO
0xC78	mlptp_rx_ref_lane	Physical lane that TAM associates with. • Bit [2:0]: `rx_ref_lane` **RX reference lane**—3 bit number of RX physical lane which packet last arrives at. — `rx_pcs_fully_aligned` deassertion will clear the bits to 0 — Resetting RX datapath or the entire core will also clear the bits to 0 Only applicable to multi-lane designs.	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx x000	WO
0xC79	mlptp_tx_user_cfg	Indicates user has completed all necessary TX configuration: • Bit [0] = 1 means configured. • Bit [0] = 0 means not configured yet: — has programmed TX extra latency to Hard IP — has programmed TX VL offset to Hard IP (Multi-lane variant only) Register value will be automatically cleared in the following condition: • `tx_lanes_stable` deassertion • Resetting TX datapath or the entire core	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx xxx0	WO
0xC7A	mlptp_rx_user_cfg	Indicates user has completed all necessary RX configuration: • Bit [0] = 1 means configured. • Bit [0] = 0 means not configured yet: — has programmed RX extra latency to Hard IP — has programmed RX VL offset to Hard IP (Multi-lane variant only) Register value will be automatically cleared in the following condition: • `rx_pcs_fully_aligned` deassertion • Resetting RX datapath or the entire core	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx xxx0	WO
0xC7B	mlptp_status	PTP Status Register: • Bit [3]: `rx_ptp_ready` (RX PTP Ready). 1'b1 means ready. • Bit [2]: `tx_ptp_ready` (TX PTP Ready). 1'b1 means ready. • Bit [1]: `rx_ptp_offset_data_valid` (RX calculation data valid/ready to read). 1'b1 means valid. • Bit [0]: `tx_ptp_offset_data_valid` (TX calculation data valid/ready to read). 1'b1 means valid.	0xXXXX XXX0	RO

continued...

Addr	Name	Description	HW Default Value	Access
0xC7C	mlptp_tx_const_delay	Constant delay value used in TAM adjustment calculation. Constant delay for specific physical lane. • Bit [31:0]: data_constdelay	0x0000 0000	RO
0xC7D	mlptp_rx_const_delay	Constant delay value used in TAM adjustment calculation. Constant delay for specific physical lane. • Bit [31:0]: data_constdelay	0x0000 0000	RO
0xC7E	mlptp_tx_l0_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: data_offset	0x0000 0000	RO
0xC7F	mlptp_rx_l0_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: data_offset	0x0000 0000	RO
0xC80	mlptp_tx_l0_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: data_time	0xX000 0000	RO
0xC81	mlptp_rx_l0_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: data_time	0xX000 0000	RO
0xC82	mlptp_tx_l0_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: data_wiredelay	0xXXX0 0000	RO
0xC83	mlptp_rx_l0_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: data_wiredelay	0xXXX0 0000	RO
0xC84	mlptp_tx_l1_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: data_offset	0x0000 0000	RO
0xC85	mlptp_rx_l1_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: data_offset	0x0000 0000	RO
0xC86	mlptp_tx_l1_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: data_time	0xX000 0000	RO
0xC87	mlptp_rx_l1_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: data_time	0xX000 0000	RO
0xC88	mlptp_tx_l1_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: data_wiredelay	0xXXX0 0000	RO

continued...

Addr	Name	Description	HW Default Value	Access
0xC89	mlptp_rx_l1_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: `data_wiredelay`	0xXXX0 0000	RO
0xC8A	mlptp_tx_l2_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: `data_offset`	0x0000 0000	RO
0xC8B	mlptp_rx_l2_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: `data_offset`	0x0000 0000	RO
0xC8C	mlptp_tx_l2_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: `data_time`	0xX000 0000	RO
0xC8D	mlptp_rx_l2_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: `data_time`	0xX000 0000	RO
0xC8E	mlptp_tx_l2_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: `data_wiredelay`	0xXXX0 0000	RO
0xC8F	mlptp_rx_l2_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: `data_wiredelay`	0xXXX0 0000	RO
0xC90	mlptp_tx_l3_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: `data_offset`	0x0000 0000	RO
0xC91	mlptp_rx_l3_offset	Time offset value used in TAM adjustment calculation. Time offset for specific physical lane. • Bit [31:0]: `data_offset`	0x0000 0000	RO
0xC92	mlptp_tx_l3_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: `data_time`	0xX000 0000	RO
0xC93	mlptp_rx_l3_time	Time value used in TAM adjustment calculation. Marker time for specific physical lane. • Bit [27:0]: `data_time`	0xX000 0000	RO
0xC94	mlptp_tx_l3_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: `data_wiredelay`	0xXXX0 0000	RO
0xC95	mlptp_rx_l3_wire_dly	Wire delay value used in TAM adjustment calculation. Wire delay for specific physical lane. • Bit [19:0]: `data_wiredelay`	0xXXX0 0000	RO

2.12.9. RS-FEC Registers

For information on RS-FEC registers, refer to the *E-Tile Transceiver PHY User Guide*: *RS-FEC Registers*.

Related Information

E-Tile Transceiver PHY User Guide: RS-FEC Registers

2.12.10. PMA Registers

For information on PMA registers, refer to the *E-Tile Transceiver PHY User Guide*: *PMA Register Map*.

Related Information

- E-Tile Transceiver PHY User Guide
- PMA Register Map

2.13. Document Revision History for the E-tile Hard IP for Ethernet Intel FPGA IP Core

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.08.04	21.2	20.2.1	Made the following changes: • Corrected reset value for *Lower 4 bytes of the Destination address for RX Pause Frames* and *Higher 2 bytes of the Destination address for RX Pause Frames*. • Added SyncE support for 10G variant. • Added clarifying notes in section: *PTP Timestamp and TOD Formats* and *PTP System Considerations*. • Corrected descriptions for *PHY Registers: Loopback Mode*.
2021.03.29	21.1	20.2.1	Made the following changes: • Added a new section: *SDC for Multiple E-Tile Instances*. • Added a note about AN/LT in section: *Parameter Editor Parameters*. • Removed duplicate entries in Table: *PTP Timestamp Accuracy per Ethernet Data Rate*. • Added reset information while using the Ethernet Toolkit in section: *Reset*. • Updated description for *Timer Window for Hi-BER Checks* register. • Added new parameter: **Enable SyncE**. • Added new sections: — *Single 25G Synchronous Ethernet Channel* — *Multiple 25G Synchronous Ethernet Channels*
2020.12.14	20.4	20.2.1	Made the following changes: • Added new information for 100G PTP support. • Corrected the following figures: — *Ethernet Cores Position on an E-tile Device* — *Ethernet 25G x 1* — *Ethernet 25G x 4 (FEC On) Master-Slave Configuration Option* • Clarified the availability of asynchronous adapter clocks for 100G mode. • Added the PTP Accuracy Advanced Mode support for the Intel Agilex devices. • Corrected the script location in section: *Logic Lock Regions Requirements for PTP Accuracy Advanced Mode*. • Added new register in *PHY Registers*: *Reset Sequencer RS-FEC Disable*. • Added new registers for 100G PTM PPM in section: *1588 PTP Registers*.
2020.10.05	20.3	20.2.0	Made the following changes:

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added clarifying note regarding the Avalon-ST interface modification in the E-Tile Hard IP for Ethernet Intel FPGA IP. The note was added in the *Client Interfaces for IP Core Variations* table and the *TX MAC Interface to User Logic* section. • Added KDB link in the *About the E-Tile Hard IP for Ethernet Intel FPGA IP Core* section. • Updated list of supported Ethernet channel protocols in the *Ethernet Protocols* table. • Updated *External Time-of-Day Module for Variations with 1588 PTP Feature* section. Added new subsections describing the PTP Accuracy modes: — *TOD Module Required Connections in Basic Mode* — *TOD Module Required Connections in Advanced Mode* • Updated the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: IP Tab* table: — Added **PTP Accuracy Mode** parameter. • Updated the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: 10GE/25GE Tab*: — Updated **Enable asynchronous adapter clocks** parameter description. — Added **Include refclk_mux** parameter. • Updated the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: 100GE Tab* table: — Added **Preserve Unused Transceiver Channels** parameter. — Added **Reference Clock Frequency for Preserved Channels** parameter. • Revised text in the *Determining Link Fault Condition* section. Removed outdated Intel Quartus Prime software version. • Updated *PTP System Consideration* topic. • Added new topics: — *PTP Timestamp Accuracy* — *Logic Lock Regions Requirements for PTP Accuracy in Advanced Mode* • Added links and updated adaptation flows with `soft_tx_rst` and `soft_rx_rst` resets in the following sections: — *Ethernet Adaptation Flow for 10G/25G and 100G/4x25G Dynamic Reconfiguration Design Example* — *Ethernet Adaptation Flow with PTP or with External AIB Clocking* — *Ethernet Adaptation Flow with Non-external AIB Clocking* • Updated *Reset* section: — Renamed reset from `soft_sys_rst` to `eio_sys_rst` in the *Reset Signal Functions* table. — Added `soft_tx/rx_rst` signals in the *Reset Signal Functions* table. — Removed `i_rsfec_tx_rst_n/i_rsfec_rx_rst_n` resets. — Revised *System Consideration* section. Included 10G/25G multi-channel master-slave configuration considerations. • Updated *1588 PTP Interface* section to include the PTP accuracy mode feature. Revised and added new PTP accuracy mode related signals in the *Signals of the Time of Day Interface* table. • Updated `i_clk_ref[n-1:0]/i_clk_ref` signals description in the *Transceiver Signals* table.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Updated *Clocks* section: — Revised description of `i_sl_clk_tx/rx` and `i_sl_clk_tx/rx[n]` signals in the *Clock Inputs* table when PTP is enabled. — Revised description of `i_sl_async_clk_tx[n]` and `i_sl_async_clk_rx[n]` signals in the *Clock Inputs* table. — Added new signals: `i_sl_tx_tod[n-1:0]`, `i_sl_rx_tod[n-1:0]`, and `i_clk_ptp_sample`. — Revised description of the `o_clk_pll_div64[n]` signal in the *Clock Outputs* table. — Renamed section from *10G/25G Ethernet Channel with PTP, without External AIB Clocking* to *10G/25G Ethernet Channel with Basic PTP Accuracy Mode*. Added Basic PTP accuracy mode information. — Added new topic: *10G/25G Ethernet Channel with Advanced PTP Accuracy Mode*. • Revised the Async FIFO description in the *Clock Connection in Asynchronous FIFO Operation* diagram. • Revised `0x326` and `0x328` registers description in the *RX PCS Status for AN/LT* section to include the RX alignment limitation. • Corrected the `0xB10` register width in the *TX 1588 PTP Registers* table.
2020.07.13	20.2	20.1.0	Made the following changes: • Corrected parallel clock frequency value for a 10GE data rate variant in the *PTP Timestamp Accuracy per Ethernet Data Rate* table. The frequency is 402.83 MHz. • Added `i_reconfig_clk` specific note in the *Clock Inputs* section. The note states that the clock is set to a higher frequency in the simulation of specified variants.
2020.06.29	20.2	20.1.0	Made the following changes: • Updated Debug and Testability feature in the *E-Tile Hard IP for Ethernet Intel FPGA IP Features* section. Added support for the Ethernet Tool Kit. • Updated device family support in the *E-Tile Hard IP for Ethernet Intel FPGA IP Device Family Support* table: — Intel Stratix 10 support moved from Preliminary to Final. — Intel Agilex support moved from Advance to Preliminary. • Removed *<your_ip>*`.regmap` file from the *IP Core Generated Files* table. The file doesn't appear in the list of IP core generated files. • Updated *External Time-of-Day Module for Variations with 1588 PTP Feature* section. • Updated **Average Inter-packet Gap** description in the *Intel Agilex Parameters: IP Tab* table and the *Inter-Packet Generation and Insertion* section. • Added clarifying note in the **Include deterministic latency measurement interface** description. in the *IP Parameter* section. This feature is for internal use only and should not be enabled. • Updated **ReadyLatency** description in the *Intel Agilex Parameters: 100GE Tab*. Ready Latency parameter only supports MAC+PCS variant. • Updated *RTL Parameters* section to clarify the access to the RTL generated parameters in simulation and synthesis.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Removed 64-bit timestamp format from the following topics: — *PTP Transmit Functionality* — *PTP Receive Functionality* — *External Time-of-Day Module for 1588 PTP Variations* — *PTP Timestamp and TOD Formats* • Renamed and removed outdated signals in the *PTP Transmit Block Diagram* and the *PTP Receive Block Diagram* figures. • Updated the *TX and RX PTP Extra Latency* section: — Added text to inform user to consider SFD and MDI boundaries when calculating TX/RX PTP Extra Latency. — Added step to store the calculated TX/RX PTP Extra Latency in the TX/RX_PTP_EXTRA_LATENCY registers. • Added a note in the *Deterministic Latency Interface* section. This feature is for internal use only. • Updated signal descriptions in the *1588 PTP Interface* section. • Added PTP clock assignments in the *10G/25G Ethernet Channel (with PTP and without External AIB Clocking)* section. • Corrected `eio_freq_lock` description in the *RX CDR PLL Locked* register section. Corresponding physical lane's CDR are locked to data. • Removed text: *The reset values in this table represents register values after a reset has completed.* globally. Added the text in the *Reconfiguration and Status Register Description* section. • Added Loopback Mode register in the *PHY Registers* section.
2020.01.31	19.4	19.4.0	Made the following changes: • Updated *Parameter Editor Parameters* section: — Removed **Reconfig clock rate** parameter in the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: IP Tab* table. — Added **Enable Dynamic RSFEC for KR** parameter in the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: IP Tab* table. — Added **Enable asynchronous adapter clocks** parameter in the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: 10GE/25GE Tab* table. — Added **Ready latency** parameter in the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: 10GE/25GE Tab* table. — Added **Enable asynchronous adapter clocks** parameter in the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: 100GE Tab* table. • Updated the parallel clock frequency value to 161.132 MHz for 10GE data rate in the *PTP Timestamp Accuracy per Ethernet Data Rate* table. • Updated *PTP Transmit Functionality* section: — Added `o_sl_ptp_ets_valid`, `o_sl_ptp_ets_fp[7:0]`, and `o_sl_ptp_ets[95:0]` signals in the *Example Waveform for 2-step TX Timestamp using `i_sl_ptp_ts_req` Signal* figure. — Added `i_sl_ptp_cf_offset` and `i_sl_ptp_tx_its[95:0]` signals in the *Example Waveform for 1-step TX Timestamp using `i_sl_ptp_ins_cf` Signal* figure. — Accurately updated all signal names.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added new section: *TX and RX PTP Extra Latency*. • Updated `o_sl_ptp_rx_its` signal's width in the *Example Waveform PTP Timestamp on RX PTP Interface* figure. • Added link to the *Ethernet Design Example Components User Guide* in the *PTP Timestamp and TOD Formats* section. • Clarified the `i_csr_rst_n/i_sl_csr_rst_n` reset sequence in the *Reset* section. • Added clarifying notes in the *Custom Rate Interface* section. • Added clarifying notes in the *Deterministic Latency Interface* section. • Updated *1588 PTP Interface* section: — Added comment clarifying 1-step and 2-step TX/RX timestamp interface signals behavior when Asynchronous mode is enabled. — Updated `o_sl_tx_ptp_ready[n-1:0]` and `o_sl_rx_ptp_ready[n-1:0]` signals description in the *Signals of the PTP Status Interface* table. • Updated text in the *Miscellaneous Status and Debug Signals* section. • Added three new adaptation flows: — *Ethernet Adaptation Flow for 10G/25G and 100G/4x25G Dynamic Reconfiguration Design Example* — *Ethernet Adaptation Flow with External AIB Clocking and PTP* — *Ethernet Adaptation Flow with non-external AIB Clocking* • Updated *Clocks* section. — Updated `i_reconfig_clk` clock frequency range to 100-125 MHz in the *Clock Inputs* table. — Updated description of `i_sl_clk_tx/rx` and `i_sl_async_clk_tx/rx` signals in the *Clock Inputs* table. — Added new section: *Asynchronous Adapter Clock in 25G Mode*. — Added new section: *Asynchronous Adapter Clock in 100G Mode*.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Updated *Reset* section: — Added Intel recommendation details on how to perform reset an IP. — Updated the `i_csr_rst_n`/`i_tx_rsn_n`, and `i_rx_rsn_n` signals for RX_FEC and TX_FEC blocks in the *Reset Signals Function* table • Updated *Auto Negotiation and Link Training Registers* section: — Added the `rsfec_capable`, `rsfec_request`, `anlt_seq_cfg_txinv`, and `anlt_seq_cfg_rxinv` signals in the *ANLT Sequencer Config*. — Added the `enable_consortium_next_page_send`, `enable_consortium_next_page_receive`, `enable_consortium_next_page_override`, `ignore_consortium_next_page_tech_ability_code`, and `consortium_oui` signals in the *Auto Negotiation Config Register 1*. — Added the `an_next_page` and `consortium_oui_upper` in the *Auto Negotiation Config Register 2*. — Added the `consortium_next_page_received`, `consortium_negotiated_port_type`, and `rs_fec_negotiated` in the *Auto Negotiation Status Register*. — Updated description of `ieee_negotiated_port_type` in the *Auto Negotiation Status Register*. — Added the `user_next_page_low` in the *Auto Negotiation Config Register5*. — Added new *Auto Negotiation Config Register 6*. — Updated the `an_lp_adv_tech_a` in the *Auto Negotiation Status Register 5*. — Updated *AN Channel Override* register description. — Updated description of `override_consortium_next_page_tech` and `override_consortium_next_page_fec_control` in the *Consortium Next Page Override* register. — Added new *Consortium Next Page Link Partner Status* register. • Updated `flowreg_rate` signal in the *EHIP TX MAC Feature Configuration* section.
2020.01.16	19.3	19.3.0	Made the following changes: • Updated `tx_clkout` and `rx_clkout` to 161.132 MHz in the *Ethernet 25G x 4 (FEC On) Master-Slave Configuration: Dynamic Reconfiguration* figure. • Updated legend with missing clocks in the *Ethernet 25G x 4 (FEC On) Master-Slave Configuration: Dynamic Reconfiguration* and *Ethernet 25G x 4 (FEC On) Master-Slave Configuration: External AIB Clocking* figures.
2019.12.30	19.3	19.3.0	Made the following changes: • Added 0x864 ~ 0x869 registers and their descriptions in the *Reconfiguration and Status Register Descriptions: TX Statistics Registers* section.
2019.11.15	19.3	19.3.0	Made the following changes:

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added Intel Agilex device family support. • Added Ethernet Link Inspector in the *E-Tile Hard IP for Ethernet Intel FPGA IP Features* table. • Added note clarifying transceiver speed grade -3 support in the *E-Tile Hard IP for Ethernet Intel FPGA IP Device Speed Grade Support*. • Updated IP version, Intel Quartus Prime version, and release date in the *E-tile Hard IP for Ethernet Intel FPGA IP Current Release Information* table. • Updated parameters description in the *Parameter Editor Parameters* section: — **Enable Auto-Negotiation on Reset** — **Link Fall Inhibit Time** — **Auto-Negotiation Master** — **Advertise both 10G and 25G during AN** — **Enable Link Training on Reset** • Restructured *1588 Precision Time Protocol Interface* section: — Updated the *PTP Receive Block Diagram* to include the RX and TX UI Adjustment block — Added new *TX and RX Unit Interval Adjustment* section • Updated PTP timestamp accuracy for 100GE in the *PTP Timestamp Accuracy per Ethernet Data Rate* table. • Updated ordering code from IP-ETH-ETILEHARDIP to IP-ETH-ETILEHIP in the *E-tile Hard IP for Ethernet Intel FPGA IP Current Release Information* table. • Renamed *Guidelines and Restrictions* section to *Guidelines and Restrictions for 24-channel placement variant*. • Added new section *Guidelines and Restrictions for 16-channel placement variant*. • Added clarification in *PTP Transmit Functionality* and *PTP Receive Functionality* sections stating that TX/RX PTP operations start only after the `o_sl_rx/tx_ready` signal is set. • Clarified statement on asserting the external hard reset in the *Reset* section. • Updated statement to include `i_sl_tx_rst_n` reset for 10G/25G variant in the *Reset Sequence*. • Reworded description of PTP registers in the *1588 PTP Interface* section. • Updated registers description in the *Deterministic Latency Interface* section. • Updates signal's description in the *Signals of the PTP Status Interface* table. • Updated *Reset Sequence with External AIB Clocking* section. • Added new dynamic reconfiguration clock network use case in the *Four 25G Ethernet Channels (with FEC)* section. • Removed unclear statement from `i_reconfig_reset` description in the *Reset Signals* section. • Added clarification in the *Auto-Negotiation and Link Training* section: When enabled, link training supports the initial and continuous adaptation. • Added note to clarify that RX recovered clock is not available for PTP channels when PTP enabled, in the *Clock Outputs* table.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.09.18	19.2	19.2.0	• Updated Variant Selection figure in the *About the E-tile Hard IP for Ethernet Intel FPGA IP Core* section. — Single 25G MAC with PCS and optional 1588PTP and (528,514) RS-FEC is not supported — 1 to 4-10GE/25G MAC with PCS and optional 1588PTP and (528,514) RS-FEC is not supported • Updated E-Tile Channel Placement Tool link in the *Channel Placement* section. • Updated Example Waveform PTP Timestamp on RX PTP Interface table in the *PTP Receive Functionality* table.
2019.08.07	19.2	19.2.0	• Updated Variant Selection figure in the *About the E-tile Hard IP for Ethernet Intel FPGA IP Core* section. • Clarified comments in the *MAC TX Datapath* and the *TX Preamble, Start, and SFD Insertion* sections. • Removed MAC adapters support for the 100G channel in the *E-Tile Hard IP for Ethernet Intel FPGA IP Overview* and the *Resource Utilization* sections. • Updated the *Release Information* section. • Updated the *Specifying the IP Core Parameters and Options* section. • Added the *Reset Sequence with External AIB Clocking* section. • Updated the *Clock Requirements* section. • Updated the *100GE with RS FEC Variant* section. • Added the *Two Channels 100GE/25GE with RS-FEC and PTP Variant* and the *Three Channels 100GE/25GE with RS-FEC and PTP Variant* sections. • Added link to Intel Stratix 10 Device Datasheet in the *E-tile Hard IP for Ethernet Intel FPGA IP Device Speed Grade Support* section. • Updated the *Reconfig clock rate* parameter description in the *E-tile Hard IP for Ethernet Intel FPGA IP Parameters: IP Tab* section. • Added reset information in the *Four 25G Ethernet Channels (with FEC) within the Single FEC block - Master-Slave Configuration: Option 2* section. • Updated the *Ethernet 25G x 4 (FEC Off)* section. • Updated the *25G Ethernet Channel (with PTP and External AIB clocking)* section. • Updated behavior of `i_rx_rst_n` signal in RX PCS and RX FEC blocks in the *Reset Signal Functions* table. • Renamed the *PMA Register Base Addresses* table with *Transceiver Reconfiguration Interface Register Base Addresses* name. • Added the *PTP Reconfiguration Interface Register Base Addresses* table.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Updated following bits in *Auto Negotiation and Link Training Registers*: — Removed RX polarity inversion for lane 0 to lane 3 signals `anlt_seq_cfg_rxinv` in ANLT Sequencer Config Fields register — Removed TX polarity inversion for lane 0 to lane 3 signals `anlt_seq_cfg_txinv` in ANLT Sequencer Config Fields register — Removed support for dynamic RS-FEC signals `rsfec_request`, `rsfec_capable` in ANLT Sequencer Config Fields register — Removed `rs_fec_negotiated` in Auto Negotiation Status Register — Removed `consortium_next_page_received` in Auto Negotiation Status Register — Updated `override_an_tech_22_8`, bits [23:19] in Auto Negotiation Config Register 5 — Updated `an_lp_adv_tech_a`, bits [15:11] in Auto Negotiation Status Register 5 • Added *Minimizing PMA Adaptation Time* section in the *PMA Registers* chapter. • Added *Ethernet Link Inspector* chapter.
2019.05.17	19.1	19.1	• Added information for Custom PCS variations. • Added 1588 PTP feature support for 100G Ethernet rate variations. • Added information for external AIB clocking feature support. • Added the following parameters in *Parameter Editor Parameters* section: — **RSFEC Clocking Mode** — **Enable external AIB clocking** — **Enable JTAG to Avalon Master Bridge** — **Number of PCS Channels in core** — **Custom PCS mode** — **RSFEC Fibre Channel(s) mode** — **Custom PCS Rate** — **PMA modulation type** — **PMA reference clock frequency** — **Enable custom rate regulation** • Added 312.5 and 644.53125 MHz options for **PHY Reference Frequency** parameter in *E-tile Hard IP for Ethernet Intel FPGA Parameters: 10GE/25GE Tab* table. • Updated the width of the `i_rsfec_reconfig_writedata` and `o_rsfec_reconfig_readdata` RS-FEC reconfiguration signals to 8 bits. • Added the following PTP timestamp diagrams in *PTP Transmit Functionality* and *PTP Receive Functionality*: — Example Waveform for 2-step TX Timestamp using `i_ptp_ts_req` Signal — Example Waveform for 1-step TX Timestamp using `i_ptp_ins_ets` Signal — Example Waveform for for 1-step TX Timestamp using `i_ptp_ins_cf` Signal — Example Waveform PTP Timestamp on RX PTP Interface • Added *PTP Timestamp Accuracy and Parallel Clock Frequency Support per Ethernet Data Rate* table in *Implementing a 1588 System That Includes a E-Tile Hard IP for Ethernet Intel FPGA IP Core* section.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Removed the following diagrams from *Implementing a 1588 System That Includes a E-Tile Hard IP for Ethernet Intel FPGA IP Core*: — Example Ethernet System with Ordinary Clock Master and Ordinary Clock Slave — Hardware Configuration Example Using E-Tile Hard IP for Ethernet Intel FPGA IP in a 1588 System in Transparent Clock Mode — Software Flow Using Transparent Clock Mode System — Example Boundary Clock with One Slave Port and Two Master Ports • Added dynamic reconfiguration clock requirements to `i_sl_clk_tx`/`i_sl_clk_tx[n]` and `i_sl_clk_rx`/`i_sl_clk_rx[n]` in *Clock Inputs* table. • Added the following topics in the *Clock Network Use Cases* section: — *10G Ethernet Channel (with PTP and without External AIB Clocking)* — *25G Ethernet Channel (with PTP and without External AIB Clocking)* — *10G Ethernet Channel (with PTP and External AIB Clocking)* — *25G Ethernet Channel (with PTP and External AIB Clocking)* • Added RX recovered clock frequency for SyncE support in *Clock Outputs* table. • Added *10/25G Ethernet Channel (with PTP and without External AIB Clocking)* and *10/25G Ethernet Channel (with PTP and External AIB Clocking)* sections.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added the following bits in *Auto Negotiation and Link Training Registers*: — `seq_force_mode` in ANLT Sequencer Config — `anlt_seq_cfg_ilpbk` in ANLT Sequencer Config — `anlt_seq_cfg_txinv` in ANLT Sequencer Config — `anlt_seq_cfg_rxinv` in ANLT Sequencer Config — `kr_pause` in ANLT Sequencer Config — `high_effort_train` in Link Training Config Register 1 — `train_start_initpre` in Link Training Config Register 1 — `lt_cfg1_disable_rxcal` in Link Training Config Register 1 — `lt_cfg1_disable_prxcal` in Link Training Config Register 1 — `lt_cfg1_disable_prelt` in Link Training Config Register 1 — `lt_cfg1_disable_postlt` in Link Training Config Register 1 — `lt_cfg1_ovrd_lf` in Link Training Config Register 1 — `lt_cfg1_ovrd_hf` in Link Training Config Register 1 — `lt_cfg1_ovrd_bw` in Link Training Config Register 1 — `restart_link_training_ln0` in Link Training Config Register 2 — `restart_link_training_ln2` in Link Training Config Register 2 — `restart_link_training_ln3` in Link Training Config Register 2 — `force_tx_nonce_value` in Auto Negotiation Config Register 1 — `consortium_oui_upper` in Auto Negotiation Config Register 2 • Added AN Channel Override register. • Added Transfer Ready (AIB reset) Status for EHIP, ELANE, and PTP Channels register. • Added EHIP, ELANE, and RS-FEC Reset Status register. • Removed the following registers: — Reference Clock Frequency in KHz — Internal Error Vector for RX PCS — Internal Error Mask for RX PCS • Rebranded Altera Debug Master Endpoint to Native PHY Debug Master Endpoint in the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters: IP Tab* table.
2019.04.19	18.1.1	18.1.1	Updated *RX Malformed Packet Handling* section to clarify that packets with Error bytes are considered as malformed packets.
2019.01.11	18.1.1	18.1.1	• Added RS-FEC support 25G and 100G Ethernet rate with PCS Only, OTN, and FlexE variations. • Updated the *Variant Selection* figure with new variations. • Updated the *Resource Utilization for Selected Variations* table.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added the following parameters in the *E-Tile Hard IP for Ethernet Intel FPGA IP Parameters* table: — IEEE1588/PTP channel placement restriction — First RSFEC Lane — Request RSFEC — Advertise both 10G and 25G during AN — Enable Link Training on Reset — Enable Altera Debug Master Endpoint (ADME) — Ready latency • Added a note to clarify that TX MAC does not support non-contiguous packet transfers in *MAC TX Datapath* section. • Added TX and RX PCS datapath for RS-FEC variant block diagrams in *TX PCS and RX PCS Datapath* topic. • Added 1588 Precision Time Protocol Interfaces topic. • Added clock network use cases topic for the following use case: — Single 25G Ethernet channel with single FEC block — Single 10G Ethernet channel without FEC — Four 25G Ethernet channel with single FEC block — Four 25G Ethernet channel without FEC — 100G Ethernet channel with aggregate FEC • Added *Guidelines and Restrictions* topic to describe supported configurations for multi Native PHY channels with RS-FEC and optional PTP. • Added *Channel Placement Guidelines and Restrictions*. • Restructured *Functional Description* section into *MAC*, *PCS/PCS66*, *PMA Direct Mode*, *Auto-Negotiation and Link Training*, and *TX and RX RS-FEC*. • Added *Determining Link Fault Condition*. • Added reset sequences in *Reset* chapter. • Changed the following registers to reserved. These registers are not used in the IP core. — Asymmetric PTP Latency address 0xA0B — TX Extra Latency Information for PTP address 0xA0C — TX Extra Latency Information for PTP address 0xA0E — RX Extra Latency Information for PTP address 0xB07 — RX Extra Latency Information for PTP address 0xB08 • Removed *PTP Asymmetric Latency* feature support from *E-Tile Hard IP for Ethernet Intel FPGA IP* table. This feature is not supported in the IP.
2018.08.10	18.0	18.0	Initial release.

3. About the E-Tile CPRI PHY Intel FPGA IP

The E-Tile CPRI PHY Intel FPGA IP implements the physical layer (layer 1) specification based on the *Common Public Radio Interface (CPRI) v7.0 Specification (2015-10-09)* in Intel Stratix 10 and Intel Agilex E-tile FPGA production devices. The IP supports up to 23 CPRI channels and the CPRI line rates of 2.4376, 3.0720, 4.9152, 6.1440, 9.8304 Gbps. This IP also supports 10.1376, 12.1651 and 24.33024 Gbps CPRI line rate with and without Reed-Solomon Forward Error Correction (RS-FEC).

Related Information

- E-tile Hard IP Intel Stratix 10 Design Examples User Guide
  Describes the Ethernet, CPRI PHY, and Dynamic Reconfiguration design example generation, simulation, compilation, and testing for Intel Stratix 10 devices.
- E-tile Hard IP Intel Agilex Design Examples User Guide
  Describes the Ethernet, CPRI PHY, and Dynamic Reconfiguration design example generation, simulation, compilation, and testing for Intel Agilex devices.
- E-Tile Transceiver PHY User Guide

3.1. Supported Features

The E-Tile CPRI PHY Intel FPGA IP core has the following features:

- Compliant with the *CPRI Specification V7.0 (2015-10-09)*
- Supports up to 23 CPRI channels
- Supports configurable CPRI communication line bit rate of 2.4376, 3.0720, 4.9152, 6.1440, 9.8304, 10.1376, 12.1651 and 24.33024 Gbps using Intel Stratix 10 and Intel Agilex E-tile transceivers
- Supports dynamic reconfiguration to different line bit rates during run time
- Supports RS-FEC block for 10.1376, 12.1651 and 24.33024 Gbps line bit rate
- Supports deterministic latency measurement
- Provides register access interface to external or on-chip processor, using the Intel Avalon Memory-Mapped (Avalon-MM) interconnect specification
- Supports Physical Medium Attachment (PMA) adaptation

Table 80. E-Tile CPRI PHY IP Core Feature Matrix

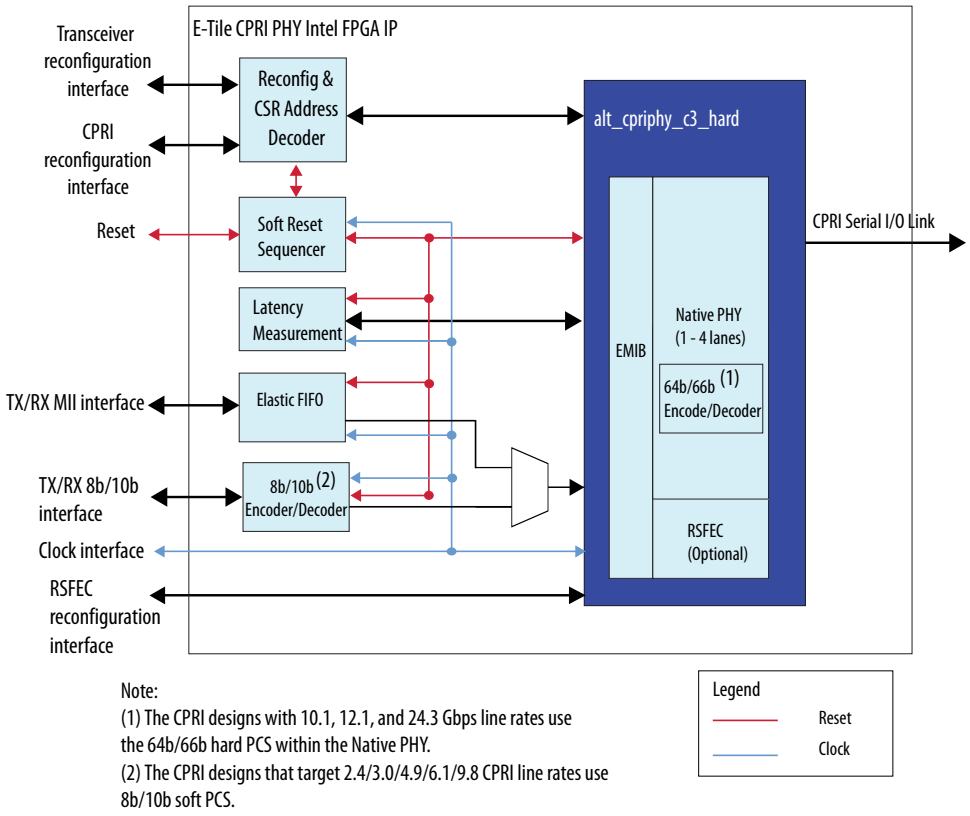
The Intel Quartus Prime Pro Edition software supports the following combinations

CPRI Line Bit Rate (Gbps)	RS-FEC Support	Reference Clock (MHz)	Deterministic Latency Support
2.4376	No	153.6	Yes
3.0720	No	153.6	Yes
4.9152	No	153.6	Yes
6.1440	No	153.6	Yes
9.8304	No	153.6	Yes
10.1376	With and Without	184.32	Yes
12.1651	With and Without	184.32	Yes
24.33024	With and Without	184.32	Yes

3.2. E-Tile CPRI PHY Intel FPGA IP Overview

The E-Tile CPRI PHY Intel FPGA IP block diagrams show the main blocks, and internal and external connections for each variant.

Figure 72. E-Tile CPRI PHY IP Block Diagram



Note:
(1) The CPRI designs with 10.1, 12.1, and 24.3 Gbps line rates use the 64b/66b hard PCS within the Native PHY.
(2) The CPRI designs that target 2.4/3.0/4.9/6.1/9.8 CPRI line rates use 8b/10b soft PCS.

- The E-Tile CPRI PHY IP core supports line bit rate of 2.4376, 3.0720, 4.9152, 6.144, 9.8304, 10.1376, 12.1651, and 24.33024 Gbps up to four channels. The RS-FEC block is optional for the IP core variations that target 10.1376, 12.1651, and 24.33024 Gbps CPRI line rate.
- The soft reset sequencer implements the reset sequence of the IP core.
- The IP variations with 2.4376, 3.0720, 4.9152, 6.144, and 9.8304 Gbps CPRI line rates include 8b/10b soft PCS and the IP variations that target CPRI line rates 10 .1376, 12.1651, and 24.33024 Gbps use 64b/66b hard PCS within the Native PHY.
- It supports latency measurement for delay calculation between the FPGA pins to the core.

Note: You need to configure an E-tile Native PHY instance as a PLL that drives CPRI PHY's Embedded Multi-die Interconnect Bridge (EMIB) interface across all the channels. Refer to section *Master-Slave Configuration: Option 2* to see how to configure E-tile Native PHY as a PLL. The E-Tile CPRI PHY IP does not support *Master-Slave Configuration: Option 1*

Related Information

- Master-Slave Configuration: Option 1 on page 166
- Master-Slave Configuration: Option 2 - External AIB Clocking Scheme on page 167

3.3. E-Tile CPRI PHY Device Family Support

Table 81. Intel FPGA IP Core Device Support Levels

Device Support Level	Definition
Advance	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 (datapath width, burst depth, I/O standards tradeoffs).
Preliminary	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	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.

Table 82. E-Tile CPRI PHY IP Core Device Family Support

Shows the level of support offered by the E-Tile CPRI PHY IP core for each Intel FPGA device family.

Device Family	Support
Intel Stratix 10 E-tile devices only	Advance
Intel Agilex E-tile devices only	Advance
Other device families	No support

3.4. Resource Utilization

The resources for the E-Tile CPRI PHY IP core were obtained from the Intel Quartus Prime Pro Edition software version 20.1

Table 83. Resource Utilization for Intel Stratix 10 Devices

CPRI Line Bit Rate (With number of channel1)	Enable Native PHY Debug Master Endpoint Parameter	ALMs	ALUTs	Dedicated Logic Registers	Memory 20K
24.33024 Gbps with RS-FEC	On	2,382	3,203	3,645	6
24 Gbps without RS-FEC	On	2,246	3,050	3,402	6
12.1651 Gbps with RS-FEC	On	2,389	3,195	3,647	6
12.1651 Gbps without RS-FEC	On	2,247	3,052	3,411	6
10.1376 Gbps with RS-FEC	On	2,393	3,197	3,657	6
10.1376 Gbps without RS-FEC	On	2,254	3,078	3,400	6
9.8304 Gbps	On	2,768	3,782	4,680	6
6.144 Gbps	On	2,764	3,781	4,768	6
4.9152 Gbps	On	2,781	3,796	4,699	6
3.0720 Gbps	On	3,774	3,787	4,725	6
2.4376 Gbps	On	2,771	3,819	4,694	6

Table 84. Resource Utilization for Intel Agilex Devices

CPRI Line Bit Rate (With number of channel1)	Enable Native PHY Debug Master Endpoint Parameter	ALMs	ALUTs	Dedicated Logic Registers	Memory 20K
24.33024 Gbps with RS-FEC	On	2,421	3,216	3,705	6
24 Gbps without RS-FEC	On	2,271	3,104	3,445	6
12.1651 Gbps with RS-FEC	On	2,395	3,180	3,639	6
12.1651 Gbps without RS-FEC	On	2,251	3,081	3,457	6
10.1376 Gbps with RS-FEC	On	2,397	3,192	3,658	6
10.1376 Gbps without RS-FEC	On	2,254	3,082	3,405	6
9.8304 Gbps	On	3,769	3,807	4,716	6
6.144 Gbps	On	2,772	3,788	4,709	6

continued...

CPRI Line Bit Rate (With number of channel1)	Enable Native PHY Debug Master Endpoint Parameter	ALMs	ALUTs	Dedicated Logic Registers	Memory 20K
4.9152 Gbps	On	2,801	3,786	4,715	6
3.0720 Gbps	On	2,772	3,793	4,748	6
2.4376 Gbps	On	2,747	3,788	4,673	6

3.5. Release Information

Intel FPGA IP versions match the Intel Quartus Prime Design Suite software versions until v19.1. Starting in Intel Quartus Prime Design Suite software version 19.2, Intel FPGA IP has a new versioning scheme.

The Intel FPGA IP version (X.Y.Z) number can change with each Intel Quartus Prime software version. A change in:

- X indicates a major revision of the IP. If you update the Intel Quartus Prime software, you must regenerate the IP.
- Y indicates the IP includes new features. Regenerate your IP to include these new features.
- Z indicates the IP includes minor changes. Regenerate your IP to include these changes.

Table 85. E-Tile CPRI PHY Intel FPGA IP Core Release Information

Item	Description
IP Version	19.4.2
Intel Quartus Prime Version	21.1
Release Date	2021.03.29
Ordering Code	IP-CPRI-v7-E-PHY

Related Information

E-Tile CPRI PHY Intel FPGA IP Release Notes
Describes changes to the IP in a particular release.

3.6. E-Tile CPRI PHY Intel FPGA IP Core Device Speed Grade Support

The E-Tile CPRI PHY Intel FPGA IP core supports Intel Stratix 10 and Intel Agilex devices with these speed grade properties:

- Transceiver speed grade: -1 or -2
- Core speed grade: -1 or -2

3.7. Getting Started

The following sections explain how to install, parameterize, simulate, and initialize the E-tile CPRI PHY Intel FPGA IP core:

3.7.1. Installing and Licensing Intel FPGA IP Cores

The Intel Quartus Prime Pro Edition 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 73. IP Core Installation Path

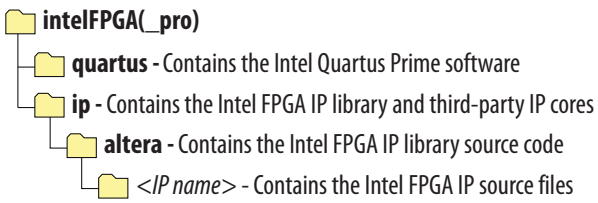


Table 86. IP Core Installation Locations

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

3.7.1.1. 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.

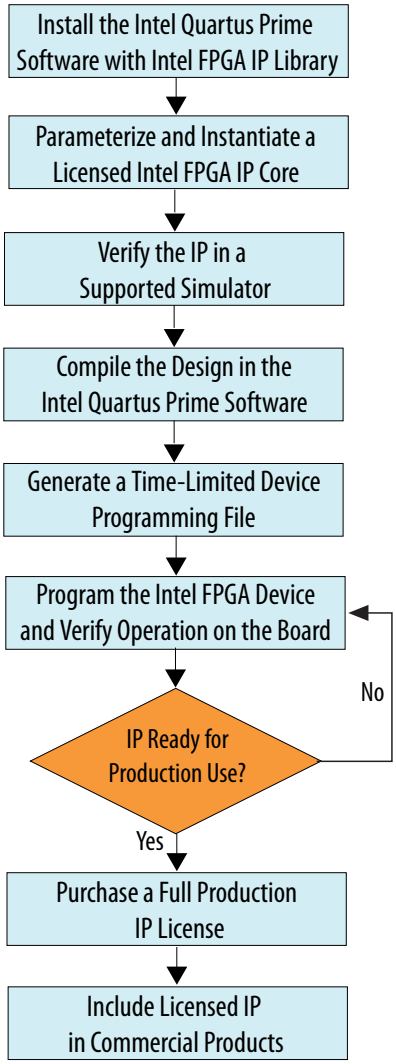
Intel FPGA IP Evaluation Mode supports the following operation modes:

- **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 74. 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 FPGA Licensing Support Center
- Introduction to Intel FPGA Software Installation and Licensing

3.7.2. Specifying the IP Core Parameters and Options

The IP parameter editor allows you to quickly configure your custom IP variation. Use the following steps to specify IP core options and parameters in the Intel Quartus Prime Pro Edition software.

1. If you do not already have an Intel Quartus Prime Pro Edition project in which to integrate your E-Tile CPRI PHY IP core, you must create one.
   a. In the Intel Quartus Prime Pro Edition, click **File ➤ New Project Wizard** to create a new Quartus Prime project, or **File ➤ Open Project** to open an existing Quartus Prime project. The wizard prompts you to specify a device.
   b. Specify the device family **Intel Stratix 10** or **Agilex (FB/FA)** and select a production E-tile device that meets the speed grade requirements for the IP core.
   c. Click **Finish**.
2. In the IP Catalog, locate and select **E-tile CPRI PHY Intel FPGA IP**. The **New IP Variation** window appears.
3. Specify a top-level name for your new custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>.ip.
4. Click **OK**. The parameter editor appears.
5. Specify the parameters for your IP core variation. Refer to Parameter Editor Parameters on page 53 for information about specific IP core parameters.
6. Optionally, to generate a simulation testbench or compilation and hardware design example, follow the instructions in the *Design Example User Guides*.
7. Click **Generate HDL**. The **Generation** dialog box appears.
8. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.
9. Click **Finish**. The parameter editor adds the top-level `.ip` file to the current project automatically. If you are prompted to manually add the `.ip` file to the project, click **Project ➤ Add/Remove Files in Project** to add the file.
10. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports and set any appropriate per-instance RTL parameters.

Related Information

- E-tile Hard IP Intel Stratix 10 Design Examples User Guide
- E-tile Hard IP Intel Agilex Design Examples User Guide

3.7.3. Generated File Structure

The Intel Quartus Prime Pro Edition software generates the following IP core output file structure.

For information about the file structure of the design example, refer to the device specific *E-tile Hard IP Intel Stratix 10 Design Examples User Guides* or *E-tile Hard IP Intel Agilex FPGA IP Design Examples User Guides*.

Figure 75. E-Tile CPRI PHY IP Core Generated Files

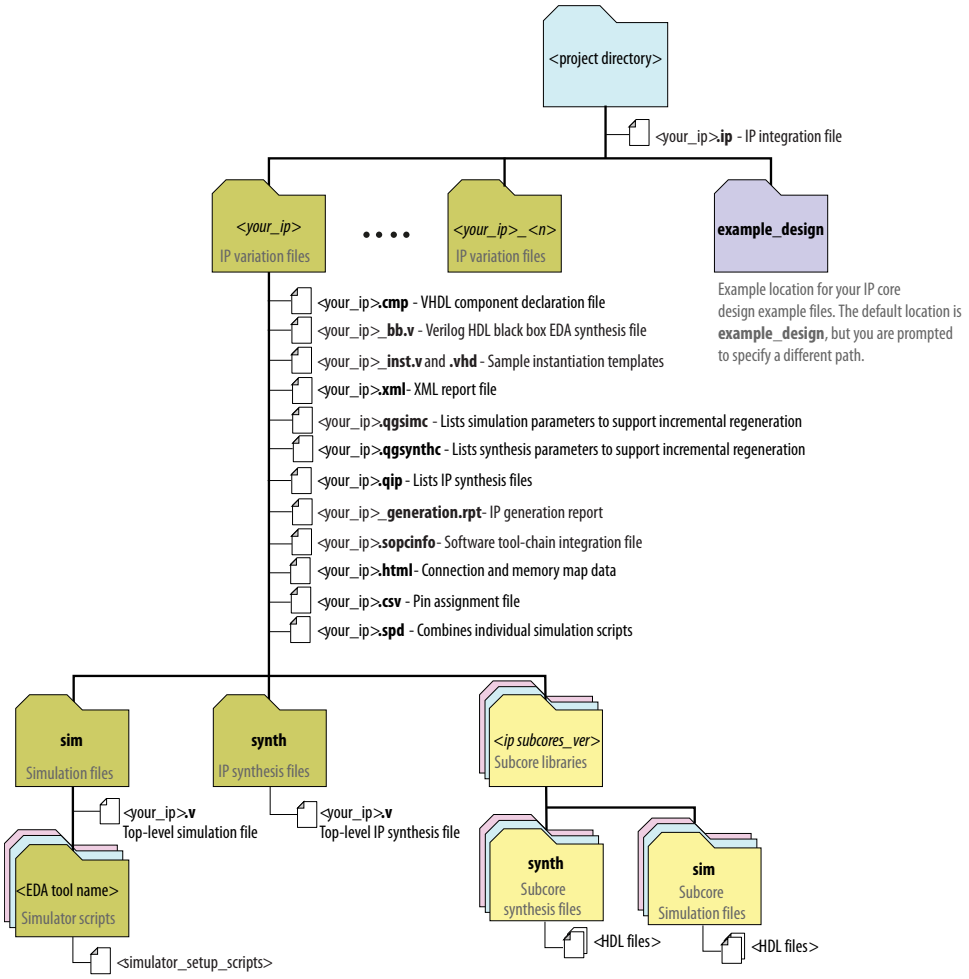


Table 87. E-Tile CPRI PHY IP Core Generated Files

File Name	Description
<*your_ip*>.ip	The Platform Designer system or top-level IP variation file. <*your_ip*> is the name that you give your IP variation.
<*your_ip*>.cmp	The VHDL Component Declaration (.cmp) file is a text file that contains local generic and port definitions that you can use in VHDL design files.
<*your_ip*>.html	A report that contains connection information, a memory map showing the address of each slave with respect to each master to which it is connected, and parameter assignments.
<*your_ip*>_generation.rpt	IP or Platform Designer generation log file. A summary of the messages during IP generation.
<*your_ip*>.qgsimc	Lists simulation parameters to support incremental regeneration.

continued...

File Name	Description
<your_ip>.qgsynthc	Lists synthesis parameters to support incremental regeneration.
<your_ip>.qip	Contains all the required information about the IP component to integrate and compile the IP component in the Intel Quartus Prime software.
<your_ip>.sopcinfo	Describes the connections and IP component parameterizations in your Platform Designer system. You can parse its contents to get requirements when you develop software drivers for IP components. Downstream tools such as the Nios II tool chain use this file. The .sopcinfo file and the system.h file generated for the Nios II tool chain include address map information for each slave relative to each master that accesses the slave. Different masters may have a different address map to access a particular slave component.
<your_ip>.csv	Contains information about the upgrade status of the IP component.
<your_ip>.spd	Required input file for ip-make-simscript to generate simulation scripts for supported simulators. The .spd file contains a list of files generated for simulation, along with information about memories that you can initialize.
<your_ip>_bb.v	You can use the Verilog black-box (_bb.v) file as an empty module declaration for use as a black box.
<your_ip>_inst.v or _inst.vhd	HDL example instantiation template. You can copy and paste the contents of this file into your HDL file to instantiate the IP variation.
<your_ip>.svd	Allows hard processor system (HPS) System Debug tools to view the register maps of peripherals connected to HPS in a Platform Designer system. During synthesis, the .svd files for slave interfaces visible to System Console masters are stored in the .sof file in the debug section. System Console reads this section, which Platform Designer can query for register map information. For system slaves, Platform Designer can access the registers by name.
<your_ip>.v or <your_ip>.vhd	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a ModelSim script msim_setup.tcl to set up and run a simulation.
synopsys/vcs/ synopsys/vcsmx/	Contains a shell script vcs_setup.sh to set up and run a VCS simulation. Contains a shell script vcsmx_setup.sh and synopsys_ sim.setup file to set up and run a VCS MX simulation.
cadence/	Contains a shell script ncsim_setup.sh and other setup files to set up and run an NCSIM simulation.
submodules/	Contains HDL files for the IP core submodules.
<child IP cores>/	For each generated child IP core directory, Platform Designer generates synth/ andsim/ sub-directories.

Related Information

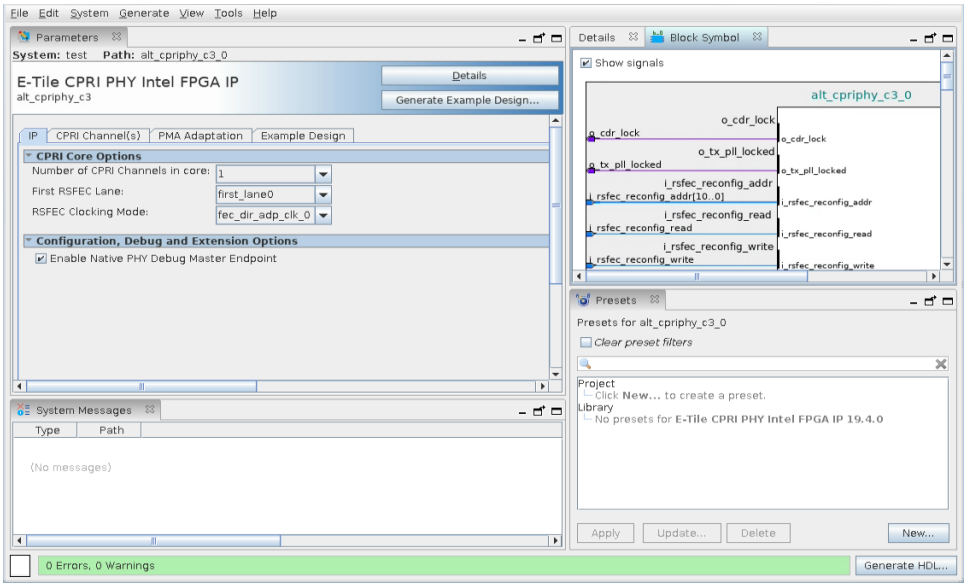
- E-tile Hard IP Intel Stratix 10 Design Examples User Guides
- E-tile Hard IP Intel Agilex Design Examples User Guides

3.7.4. E-Tile CPRI PHY Intel FPGA IP Channel Placement

Each E-tile offers up to 24 fractured mode channels. For E-Tile CPRI PHY IP, each E-tile provides up to 23 channels since the 24th channel has to be configured in PLL mode to provide 403 MHz/806 MHz clock to other CPRI channels.

The section below explains how the **First RSFEC Lane** and **RSFEC Clocking Mode** parameter works for the E-Tile CPRI PHY IP.

Figure 76. E-Tile CPRI PHY IP Parameter Editor



The following figure shows one master 24 Gbps channel providing the datapath clock to other three slave 24 Gbps channels.

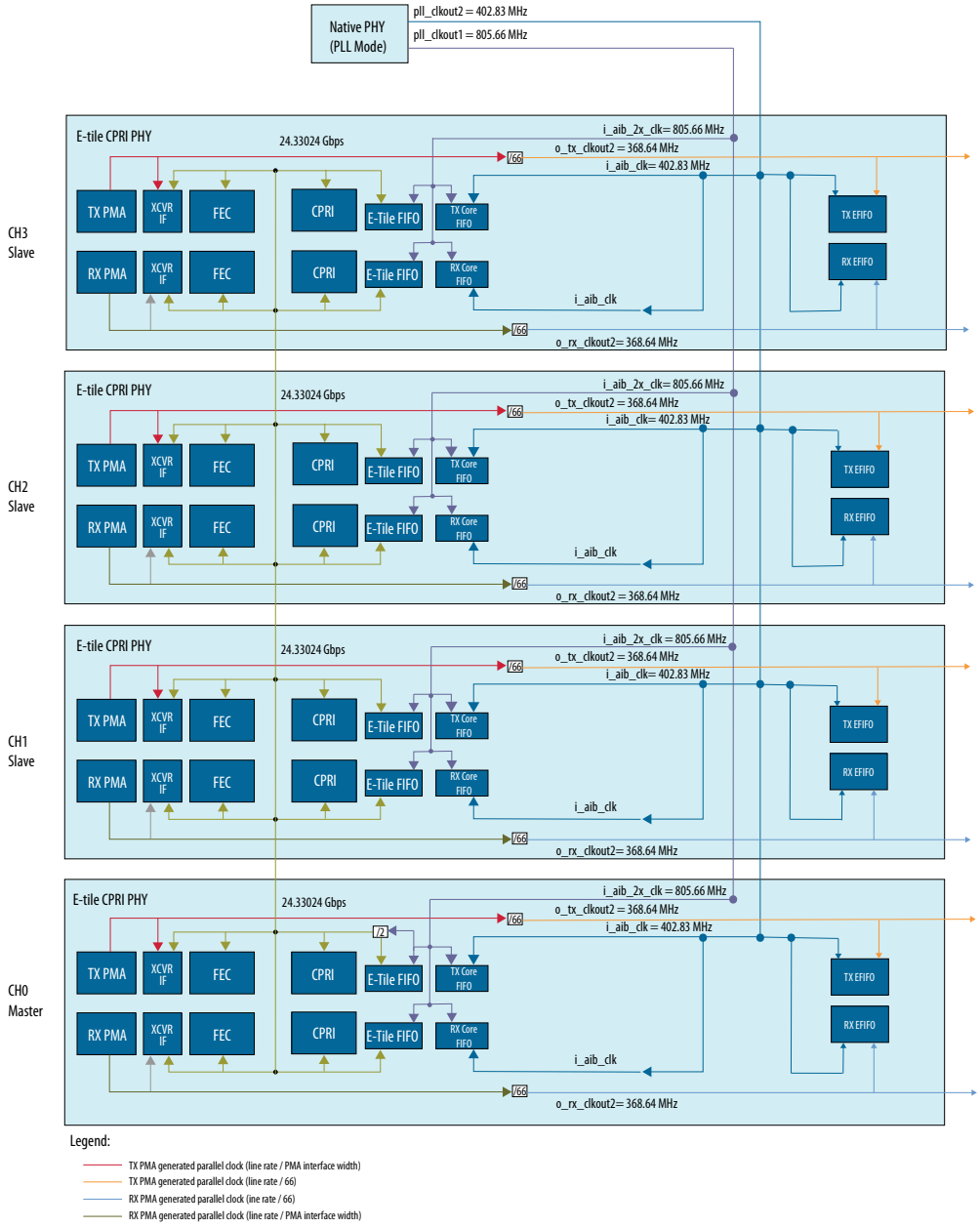
The E-Tile CPRI PHY IP uses Native PHY in PLL mode to provide 402.83 MHz and 805.66 MHz clock. All CPRI data rates use the Native PHY in PLL mode. For more information on how to implement the native PHY in PLL mode, refer to the PLL Mode section of *E-Tile Transceiver PHY User Guide*.

As shown in the figure below, the E-tile CPRI PHY uses Ethernet's clock frequencies because the Ethernet to CPRI dynamic reconfiguration uses Ethernet as the power up protocol so that AIB clock is fixed at Ethernet line rate and it remains unchanged after reconfiguration to CPRI line rates.

For more information on the dynamic reconfiguration, refer to the:

- E-Tile Dynamic Reconfiguration Design Example (for Intel Stratix 10 devices)
- E-Tile Dynamic Reconfiguration Design Example (for Intel Agilex devices)

Figure 77. E-Tile CPRI PHY (FEC On) Master-Slave Configuration



All four channels use a common RS-FEC block. However, the RS-FEC block uses only one clock from the available four channels. The channel that provides the FEC clock works as a master. The other three channels that uses the same clock for clocking their TX and RX data path works as slave channels. This creates a dependency between the master and the slave channels.

When you implement one or multiple high speed CPRI data rates with RS-FEC channels, you select the first RS-FEC lane option. You have the flexibility to select which first lane to be your master channel.

Related Information

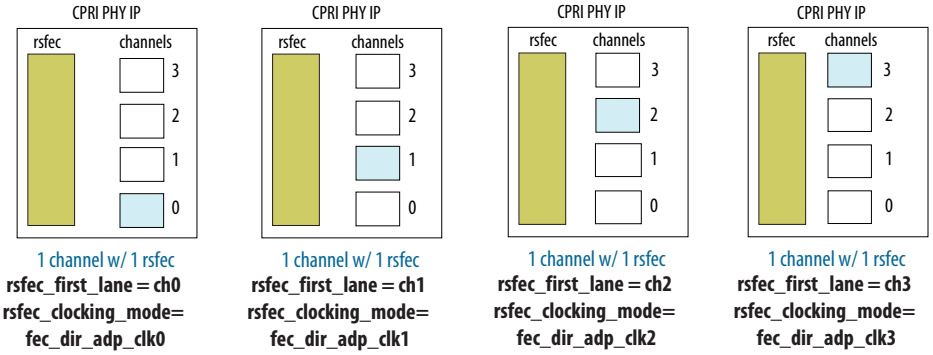
E-Tile Channel Placement Tool

3.7.4.1. One 24.33024 Gbps channel with RS-FEC

You can place the channel to **first_lane0**, or **first_lane1**, or **first_lane2** or **first_lane3**.

Note: The parameter options **first_lane0** refers to ch0, **first_lane1** refers to ch1, **first_lane2** refers to ch2, and **first_lane3** refers to ch3 in all the figures throughout this document.

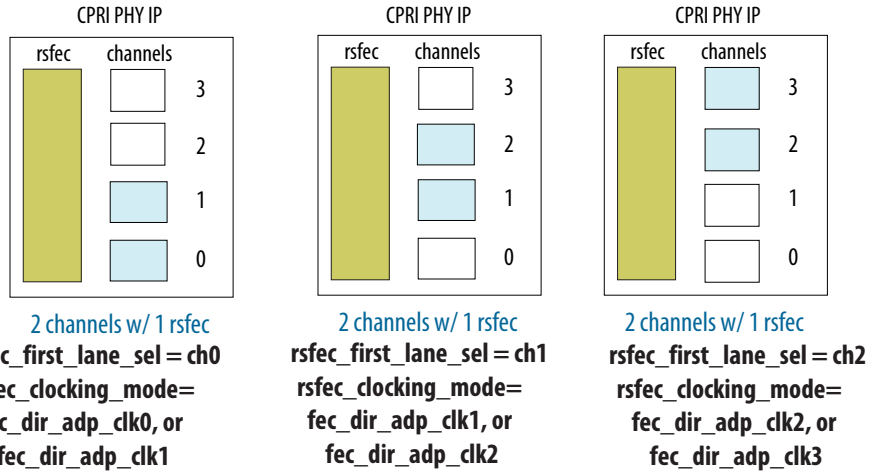
Figure 78. One 24.33024 Gbps Channel with RS-FEC



3.7.4.2. Two 24.33024 Gbps Channel with RS-FEC

You can place the first channel of your two channel to **first_lane0** or **first_lane1** or **first_lane2**. You can choose any channel as a master channel from those two channels using the **RS-FEC Clocking Mode** parameter. For example, for **first_lane1**, you can select either **fec_dir_adp_clk_1**, or **fec_dir_adp_clk_2** as your master channel.

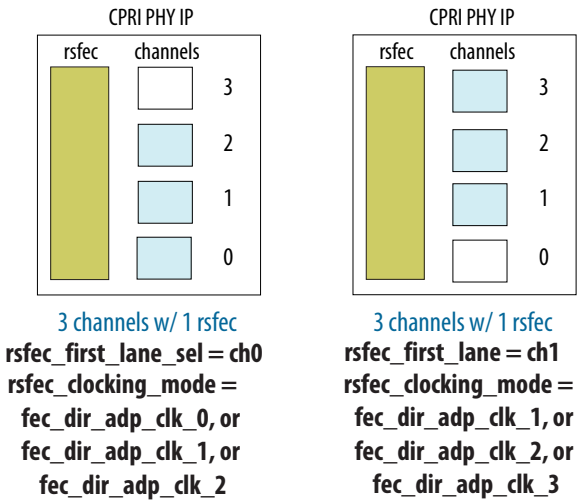
Figure 79. Two 24.33024 Gbps Channel with RS-FEC



3.7.4.3. Three 24.33024 Gbps channels with RS-FEC

You can place the first channel of your three channel to **first_lane0** or **first_lane1**. You can choose any channel as a master channel from those three channels using the **RS-FEC Clocking Mode** parameter. For example, for **first_lane0**, you can select **fec_dir_adp_clk_0**, **fec_dir_adp_clk_1**, or **fec_dir_adp_clk_2** as your master channel.

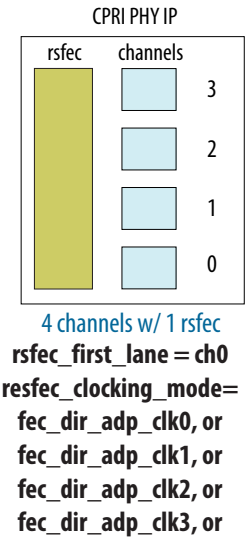
Figure 80. Three 24.33024 Gbps Channels with RS-FEC



3.7.4.4. Four 24.33024 Gbps channels with RS-FEC

You can place the first channel of your three channel to **first_lane0** only. You can choose any channel as a master channel from four channels using the **RS-FEC Clocking Mode** parameter. You can select **fec_dir_adp_clk_0**, **fec_dir_adp_clk_1**, **fec_dir_adp_clk_2**, or **fec_dir_adp_clk_3** as your master channel.

Figure 81. Four 24.33024 Gbps Channels with RS-FEC



3.7.4.5. Restrictions

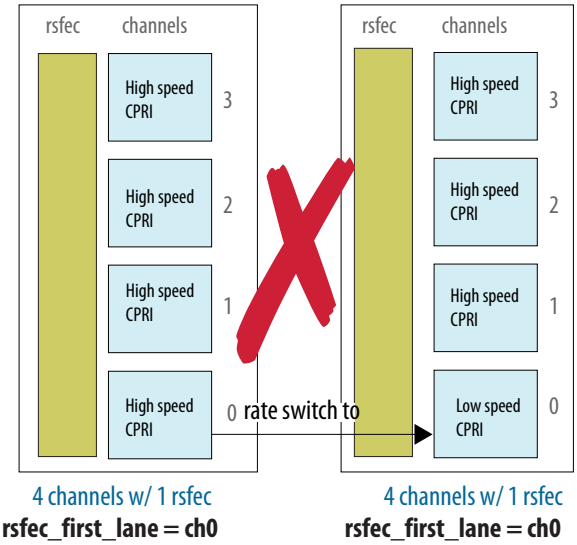
The master channel provides the clock for slave channels to use for clocking their TX and RX data paths. Any interruption to that clock from master channel impacts the already running slave channels. This creates a dependency between the master and the slave channels.

If master channel and slave channels are running at high speed CPRI data rates, and you switch the master channel high speed CPRI data rate to any of the low speed CPRI data rates, the slave channels go down since the clock from master channel is interrupted, as shown in *Figure: Master Channel Switch from high speed CPRI data rates with or without RS-FEC to low speed CPRI data rates* below.

Note: The high speed CPRI data rates are 24.3, 12.1, and 10.1 Gbps with and without RS-FEC. The low speed CPRI data rates are 2.4, 3.0, 4.9, 6.1, and 9.8 Gbps.

Figure 82. Master Channel Switch from high speed CPRI data rates with or without RS-FEC to low speed CPRI data rates

In the figure below, three high speed CPRI data rates slave channels are down.



In order to not impact slave channels which are already running at high speed CPRI data rates as shown in the above case, the master channel's rate reconfiguration should be within the data rates listed as high speed CPRI data rates. In addition to that, if master channel runs at any of the high speed CPRI data rates, the slave channels have the ability to switch to any CPRI data rates as shown in *Figure: Slave Channels Free to switch to any CPRI line rates*.

Figure 83. Master Channel Switch within high speed CPRI data rates

It does not impact slave channels running at high speed CPRI data rates.

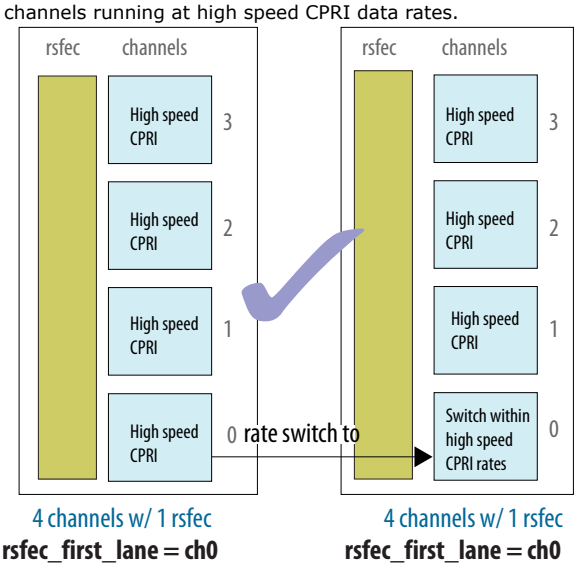
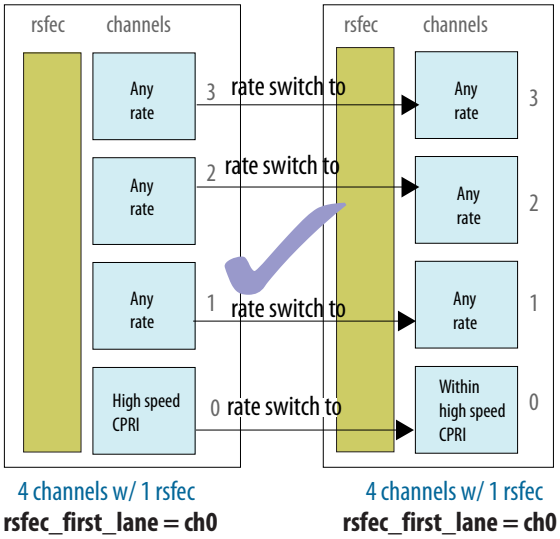


Figure 84. Slave Channels Free to switch to any CPRI line rates



Related Information

E-Tile Channel Placement Tool

3.7.5. IP Core Testbenches

Intel provides a compilation-only design example and a testbench that you can generate for the E-Tile CPRI PHY IP core.

To generate the testbench, in the E-Tile CPRI PHY parameter editor, you must first set the parameter values for the IP core variation you intend to generate in your end product. If you do not set the parameter values for your DUT to match the parameter values in your end product, the testbench you generate does not exercise the IP core variation you intend.

The testbench demonstrates XGMII data transfer to PHY with internal serial loopback and performs basic latency calculations. It is not intended to be a substitute for a full verification environment.

3.7.6. Compiling the Full Design

You can use the **Start Compilation** command on the Processing menu in the Intel Quartus Prime Pro Edition software to compile your design.

Related Information

- Block-Based Design Flows
- Programming Intel FPGA Devices

3.8. Parameter Settings

You customize the IP core by specifying parameters in the IP parameter editor.

Table 88. Parameter Settings: IP Tab

Parameter	Supported Values	Default Setting	Description
CPRI Core Options			
Number of CPRI Channels in core	1, 2, 3, 4	1	Sets the number of CPRI channels included in the CPRI core.
First RSFEC Lane	• first_lane0 • first_lane1 • first_lane2 • first_lane3	first_lane0	Sets the first RS-FEC lane. This parameter is only available in IP core variations that target CPRI line rates with RS-FEC block.
RSFEC Clocking Mode	• fec_dir_adp_clk_0 • fec_dir_adp_clk_1 • fec_dir_adp_clk_2 • fec_dir_adp_clk_3	fec_dir_adp_clk_0	Sets the clocking mode for the RS-FEC block. This parameter is only available in IP core variations that target CPRI line rates with RS-FEC block.
Configuration, Debug and Extension Options			
Enable Native PHY Debug Master Endpoint	• On • Off	On	When you turn on this parameter, the Native PHY Debug Master Endpoint instantiates an Avalon-MM master and connects the Avalon-MM slave inside the PHY. This allows access to the PHY registers for debug using the Intel Transceiver Toolkit via JTAG.

Table 89. Parameter Settings: CPRI Channel(s) Tab

Parameter	Supported Values	Default Setting	Description
CPRI General Options			
CPRI Rate	• 2.4376G (8b/10b) • 3.072G (8b/10b) • 4.9152G (8b/10b) • 6.144G (8b/10b) • 9.8304G (8b/10b) • 10.1376G (64b/66b) • 10.1376G (64b/66b) with RSFEC • 12.16512G (64b/66b) • 12.16512G (64b/66b) with RSFEC • 24.33024G (64b/66b) • 24.33024G (64b/66b) with RSFEC	10.1376G (64b/66b)	Selects the CPRI data rate. The hard RS-FEC block is included in the core if you select 10.1376, 12.1651, and 24.33024 Gbps (64b/66b) with the RS-FEC option.
Enable reconfiguration to 8b/10b datapath	• On • Off	Off	Turn on this parameter if you plan to reconfigure the CPRI line rate of your channels from 64b/66b datapath rates to 8b/10b datapath rates at run-time.

continued...

Parameter	Supported Values	Default Setting	Description
			If this option is not enabled, the CPRI IP core uses fewer resources, and not be able to change to 8b/10b datapath rates at run-time.
CPRI PMA Options			
PHY Reference frequency	• 153.6 MHz • 184.32 MHz	184.32 MHz	Support this value of the reference clock frequency for each CPRI line rate. The CPRI line rates that include 8b/10b soft PCS use a reference clock of 153.6 MHz and the CPRI line rates that include 64b/66b hard PCS use a reference clock of 184.32 MHz. This option is grayed out and always disabled in the current version of the Intel Quartus Prime software.

For parameters in the **PMA Adaptation** tab, refer to the *PMA Adaptation* topic in the *E-Tile Transceiver PHY User Guide.*

For parameters in the **Example Design** tab, refer to the device specific *E-tile Hard IP Intel Stratix 10 Design Examples User Guides*, or *E-tile Hard IP Intel Agilex Design Examples User Guides*.

Related Information

- E-Tile Transceiver PHY User Guide: PMA Parameters
  Information about PMA Adaptation parameters.
- E-Tile Transceiver PHY User Guide: Dynamic Reconfiguration Examples
  Information about configuring PMA parameters.
- E-tile Hard IP Intel Stratix 10 Design Example User Guides
- E-tile Hard IP Intel Agilex Design Example User Guides
- E-tile Transceiver PHY User Guide

3.9. Functional Description

3.9.1. CPRI PHY Functional Blocks

The E-Tile CPRI PHY Intel FPGA IP consists of the following modules:

- Native PHY—E-tile transceiver channels which consists of PMA and RS-FEC hard logic to support CPRI and Ethernet protocols. The native PHY also contains the following block:
  - 64b/66b Decoder: A hard PCS block within the Native PHY that provides encoding scheme for 10.1376, 12.1651 and 24.33024 Gbps CPRI line rates.
- Soft reset sequencer—A reset sequencer that staggers and asserts digital reset signals according to the E-Tile CPRI PHY Intel FPGA IP requirements.
- Elastic FIFO (EFIFO)—A dual clock FIFO that match the rate differences between the E-tile hard logic and soft logic.

- Latency measurement—A module that generates sync pulse to measure the datapath delay of the E-Tile CPRI PHY Intel FPGA IP.
- Reconfiguration and Control Status Register (CSR) address decoder—This is an address decoder for PHY reconfiguration interface and soft CSR.
- 8b/10b Decoder: A soft PCS block that provides encoding scheme for 2.4/3.0/4.9/6.14/9.8 CPRI line rates.

3.9.1.1. E-tile Native PHY

The E-tile Native PHY consists of PMA and RS-FEC hard logic blocks which supports CPRI and Ethernet protocols.

For more information about the E-tile Native PHY, refer to the *E-Tile Transceiver PHY User Guide*.

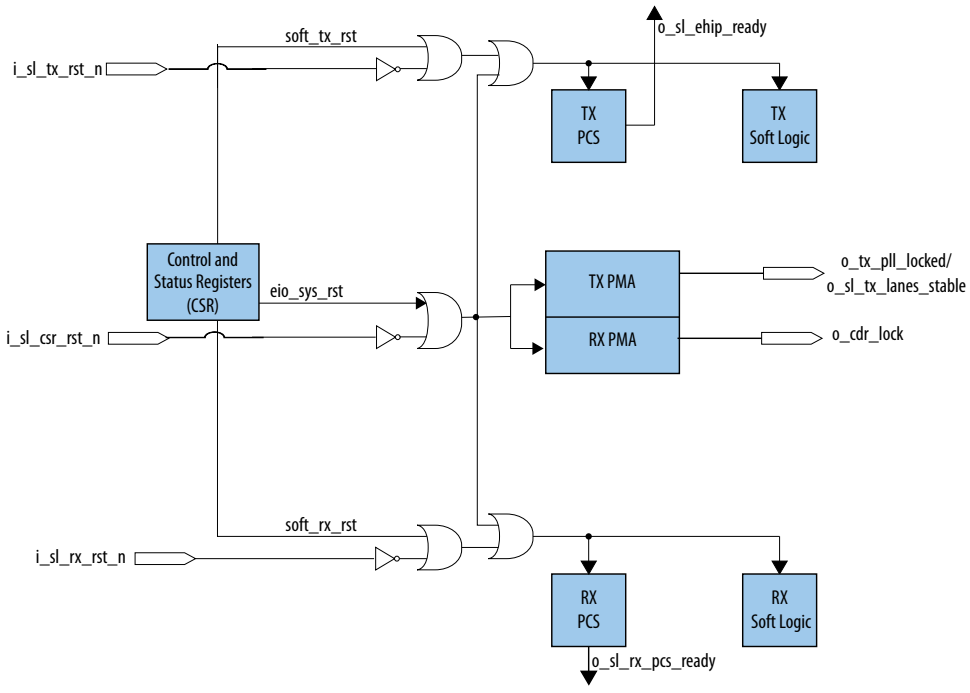
Related Information

E-Tile Transceiver PHY User Guide

3.9.1.2. Soft Reset Sequencer

The soft reset sequencer block manages the digital reset sequence in the soft logic of the E-Tile CPRI PHY IP.

Figure 85. Conceptual Overview of General IP Core Reset Logic



The IP has four input reset signals and three reset registers. The following table shows the functionality of each reset port and register.

Table 90. Reset Signal and Register Functions

In this table, a tick (√) indicates the block is reset by the specified reset signal. A dash (—) indicates the block is not impacted by the specified reset signal.

Reset Port/ Register	Block							
	TX EMIB Interface	TX PCS	TX PMA Interface	RX EMIB Interface	RX PCS	RX PMA Interfaces	Hard CSR	Soft CSR
i_sl_csr_rst_n, soft_sys_rst (34)	√(35)	√	√	√	√	√	√(36)	—
i_sl_tx_rst_n, soft_tx_rst	—	√	√	—	—	—	—	—
i_sl_rx_rst_n, soft_rx_rst	—	—	—	—	√	—	—	—
i_reconfig_reset	—	—	—	—	—	—	—	√

Reset Sequence

The following waveforms show the reset sequence using the i_sl_csr_rst_n, i_sl_tx_rst_n, and i_sl_rx_rst_n signals.

System Considerations

You should perform a system reset before beginning IP core operation, preferably by asserting the i_csr_rst_n and i_reconfig_reset signals together. The IP core implements the correct reset sequence to reset the entire IP core.

If you assert the transmit reset when the downstream receiver is already aligned, the receiver loses alignment. Before the downstream receiver loses lock, it might receive some malformed frames.

If you assert the receive reset while the upstream transmitter is sending packets, the packets in transit get corrupted.

(34) For the CPRI data rates with RS-FEC variant, deasserting the master channel's the i_sl_csr_rst_n signal interrupts all slave channels.

(35) Reset a subset of the PMA functions.

(36) soft_sys_rst resets only the registers in the hard logic and returns the register values to the original SOF values.

Figure 86. External Hard Reset Sequence

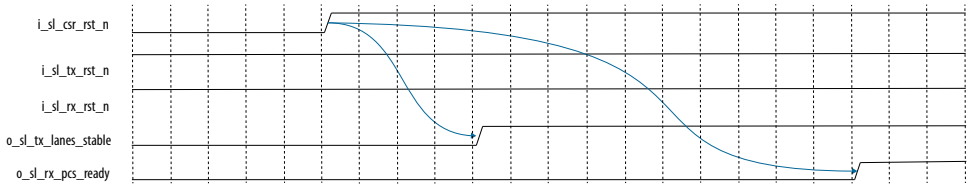


Figure 87. TX Datapath Reset Sequence

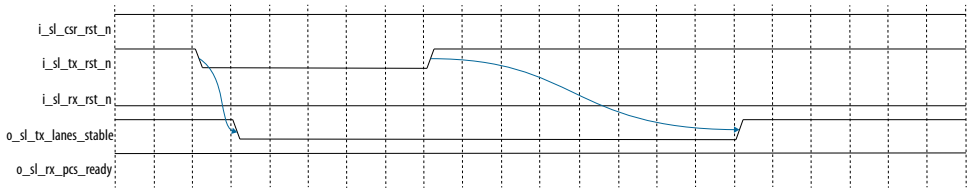
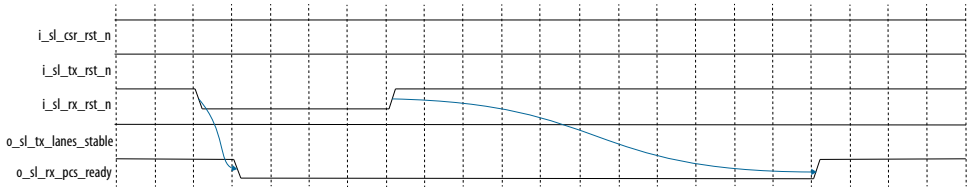


Figure 88. RX Datapath Reset Sequence



Related Information

- PMA Reset
  More information about resetting PMA channels.
- PMA Analog Reset
  More information about resetting PMA internal controller.

3.9.1.3. Latency Measurement

The latency measurement in the E-Tile CPRI PHY Intel FPGA IP measures the delay between the FPGA core and the serial pins.

3.9.1.3.1. Deterministic Latency Calculation

The Deterministic Latency (DL) term used across this document refers to the ability to precisely determine the delay between the FPGA core and the PMA pins. Such delay varies from reset to reset and device to device. In most applications the variability is acceptable in order to determine the actual delay within a given reset. The below example shows the calculation delay between pins and FPGA core for the E-Tile CPRI PHY Intel FPGA IP.

The deterministic latency measurement methodology for Intel Stratix 10 and Intel Agilex E-tile devices is based on the concept of measuring the time when a given word is at the interface to the PMA and when that same word is at the FPGA core. The difference in time between these two events, when added to the PMA propagation delay, determines the total latency between the FPGA core and the serial pins. Such a calculation intrinsically includes all delays due to intermediate logic, FIFOs and all other effects.

Table 91. Deterministic Latency Measurement for Each Variant

Variant	TX Delay (ns)	RX Delay (ns)
2.4376/3.0720 Gbps	TxDL * (sampling_clock_period in ns) / (2^8) + (365 * UI period in ns)	RxDL * (sampling_clock_period) / (2^8) + (255 * UI period in ns) + (RxBitSlipL * UI period in ns)
4.9152/6.144/9.8304 Gbps	TxDL * (sampling_clock_period in ns) / (2^8) + (367 * UI period in ns)	RxDL * (sampling_clock_period) / (2^8) + (255 * UI period in ns) + (RxBitSlipL * UI period in ns)
10.1316/12.1651/24.33024 Gbps without RS-FEC	TxDL * (sampling_clock_period in ns) / (2^8) + (569 * UI period in ns)	RxDL * (sampling_clock_period) / (2^8) + (-347) * (UI period in ns) + (RxBitSlipH * UI period in ns)
10.1316/12.1651/24.33024 Gbps with RS-FEC	TxDL * (sampling_clock_period in ns) / (2^8) + (537 * UI period in ns)	RxDL * (sampling_clock_period) / (2^8) + (-315) * (UI period in ns) - (RxCwPos * UI period in ns)

The actual latency is a function of multiple factors. The following are the description of the usage of these factors to calculate the resulting TX and RX latencies.

Table 92. Latency Calculation Description

Factor	Description
TxDL	Transmitter delay in sampling clock cycle. To calculate the TxDL value, read CPRI PHY register 0xC02 bit[20:0]. The register provides value in fixed point format. Bit[20:8] represents integer and bit[7:0] represents fractional number. For example, if bit[20:8] = 0x27 and bit [7:0] = 0xF4, the integer value is 39 and the fractional value is 0.953125 clock cycles. Therefore, the total delay is 39.953125 clock cycles. *Note:* These values are available in the design example log file at `\alt_cpriphy_c3_0_example_design \hardware_test_design\hwtest_sl \c3_cpri_test.log`.
RxDL	Receiver delay in sampling clock cycle. To calculate the RxDL value, read CPRI PHY register 0xC03 bit [20:0]. The register provides value in fixed point format. Bit[20:8] represents integer and bit[7:0] represents fractional number. For example, if bit[20:8] = 0x27 and bit [7:0] = 0xF4, the integer value is 39 and the fractional value is 0.953125 clock cycles. Therefore, the total delay is 39.953125 clock cycles. *Note:* These values are available in the design example log file at `\alt_cpriphy_c3_0_example_design \hardware_test_design\hwtest_sl \c3_cpri_test.log`.
sampling_clock_period	For E-Tile CPRI PHY Intel FPGA IP, sampling clock is 250 MHz and the period is 4 ns.
RxBitSlipH[37]	Number of bit slip required to achieve block alignment for 10.1 Gbps or higher CPRI line rates without RS-FEC. Read PMA AVMM register 0x28[6:0] to obtain this value. This

continued...

(37) If RxBitSlipH value is greater than or equal to 63, then the RxBitSlipH value is (66-RxBitSlipH). Else, the value is just RxBitSlipH.

Factor	Description
	value is a constant per link up. This value is added to the RX latency calculation. It is assumed that the CPU aggregating the delays know the UI.
RxBitSlipL	Number of bit slip required to achieve block alignment for 9.8 Gbps or lower CPRI line rates. Read CPRI AVMM register 0xC00[9:5] to obtain this value. This value is a constant per link up. This value is added to the RX latency calculation. It is assumed that the CPU aggregating the delays know the UI.
RxCwPos	Number of bit slip required to achieve FEC alignment for 10.1 Gbps or higher CPRI line rates with RS-FEC. Read PMA AVMM register 0x29[4:0] to obtain this value. This value is a constant per link up. This value is added to the RX latency calculation. It is assumed that the CPU aggregating the delays know the UI.

3.9.2. Dynamic Reconfiguration

You can dynamically reconfigure the settings in the E-Tile Intel FPGA IP to run your design in different data rates and features.

For more information on the steps and guidelines to use the dynamic reconfiguration feature, refer to the:

- E-Tile Dynamic Reconfiguration Design Example (for Intel Stratix 10 devices)
- E-Tile Dynamic Reconfiguration Design Example (for Intel Agilex devices)

3.10. E-Tile CPRI PHY Intel FPGA IP Interface Signals

All input signal names begin with `i_` and all output signal names begin with `o_`.

Multi-channel signal names contain an array index [n] to the end of their name, where n= 0 to 3.

3.10.1. Clock Signals

Each CPRI PHY channel has its own pair of datapath clocks and each transceiver has its own reference clock.

Table 93. CPRI PHY Clock Input Signals

Signal Name	Width (Bits)	I/O Direction	Description
`i_sl_clk_tx[n]`	1	Input	Single lane transmit datapath clock. These clocks drive the internal TX datapath for the CPRI PHY channel. Each CPRI PHY channel has its own clock input. The default frequency value is 402.8320 MHz.
`i_sl_clk_rx[n]`	1	Input	Single lane receive datapath clock. These clocks drive the internal RX datapath for the

continued...

Signal Name	Width (Bits)	I/O Direction	Description
			CPRI PHY channel. Each CPRI PHY channel has its own clock input. The default frequency value is 402.8320 MHz.
i_clk_ref	5	Input	Transceiver reference clock for each channel. An input multiplexer that supports five reference clocks. The default clock is index 0. You can select only 1 clock at any one time for a given channel. You can switch the clock through the transceiver reconfiguration interface. Use a 184.32 MHz reference clock to generate the high speed serial clock and datapath parallel clocks for CPRI line rates 10.1 and 24.3 Gbps with and without RS-FEC. Use a 153.6 MHz transceiver reference clock for CPRI line rates 2.4/3.0/4.9/6.1/9.8 Gbps.
i_aib_clk	1	Input	Clock for application interface block (AIB). This clock drives the AIB interface across all channels. The default frequency value is 402.8320 MHz.
i_aib_x2_clk	1	Input	Double frequency clock for AIB from external source. This clock also drives the AIB interface across all channels. The default frequency value is 805.6640 MHz.
i_reconfig_clk	1	Input	Reconfiguration clock. Frequency of 100 MHz for CSR access on all the Avalon-MM interfaces.
i_sampling_clk	1	Input	Sampling clock for deterministic latency logic. The default frequency value is 250 MHz.

Table 94. Clock Source Signals

Lists the clock source ports for the CPRI core. The core provides locally generated PLL clocks and recovered clocks that can be used for the datapath.

Signal Name	Width (Bits)	I/O Direction	Description
o_tx_clkout[n]	1	Output	Parallel TX clock running at line rate/64.
o_tx_clkout2[n]	1	Output	Parallel TX clock running at line rate/66.

continued...

Signal Name	Width (Bits)	I/O Direction	Description
			This clock drives the active TX and RX MII interface for the CPRI PHY channel.
o_rx_clkout[n]	5	Output	Parallel RX recovered clock running at line rate/64.
o_rx_clkout2[n]	1	Output	Parallel RX recovered clock running at line rate/66.

Table 95. Clock Status Signals

Lists the clock status ports for the CPRI core. Use these ports to hold the circuits that use clock sources from the core in reset until the PLLs driving the clocks are locked.

Signal Name	Width	I/O Direction	Description
o_tx_pll_locked[n]	1	Output	Indicates the TX PLL driving clock signals from the core is locked. Intel recommends not to use the o_tx_clkout or o_tx_clkout2 clocks until the o_tx_pll_locked clock is high.
o_cdr_lock[n]	1	Output	Indicates that the recovered clocks are locked to data. Intel recommends not to use the o_rx_clkout or o_rx_clkout2 clocks until the o_cdr_lock clock is high.

3.10.2. TX MII Interface

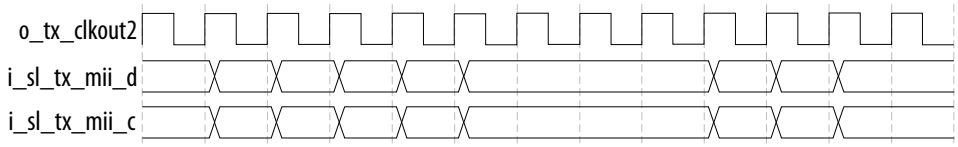
Table 96. CPRI PHY TX MII Interface

Port Name	Width	Domain	Description
i_sl_tx_mii_d[n]	64 bits per channel	o_tx_clkout2[n]	TX MII data. Data must be in MII encoding. i_tx_mii_d[7:0] holds the first byte the IP core transmits on the Ethernet link. i_tx_mii_d[0] holds the first bit the IP core transmits on the Ethernet link.
i_sl_tx_mii_c[n]	8 bits per channel	o_tx_clkout2[n]	TX MII control bits. Each bit corresponds to a byte of the TX MII data signal. For example, i_tx_mii_c[0] corresponds to i_tx_mii_d[7:0], i_tx_mii_c[1] corresponds to i_tx_mii_d[15:8], and so on.

continued...

Port Name	Width	Domain	Description
			If the value of a bit is 1, the corresponding data byte is a control byte. If the value of a bit is 0, the corresponding data byte is data. The Start of Packet byte (0xFB) and End of Packet byte (0xFD) are control bytes.

Figure 89. Transmitting Data Using TX MII Interface



The figure above shows how to write packets directly to the TX MII interface.

- The packets are written using MII.
  - Each byte in `i_sl_tx_mii_d` has a corresponding bit in `i_sl_tx_mii_c` that indicates whether the byte is a control byte or a data byte; for example, `i_sl_tx_mii_c[1]` is the control bit for `i_sl_tx_mii_d[15:8]`.
- The byte order for the TX MII interface flows from right to left; the first byte to be transmitted from the interface is `i_sl_tx_mii_d[7:0]`.
- The first bit to be transmitted from the interface is `i_sl_tx_mii_d[0]`.

3.10.3. RX MII Interface

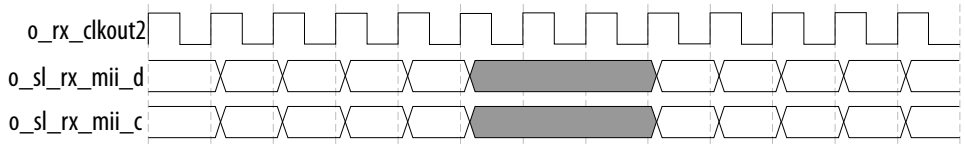
Table 97. CPRI PHY RX MII Interface

Port Name	Width	Domain	Description
`o_sl_rx_mii_d[n]`	64 per channel	`o_rx_clkout2[n]`	RX MII data. Data is in MII encoding. `o_sl_rx_mii_d[7:0]` holds the first byte the IP core received on the Ethernet link. `o_sl_rx_mii_d[0]` holds the first bit the IP core received on the Ethernet link.
`o_sl_rx_mii_c[n]`	8 per channel	`o_rx_clkout2[n]`	RX MII control bits. Each bit corresponds to a byte of RX MII data. `o_sl_rx_mii_c[0]` corresponds to `o_sl_rx_mii_d[7:0]`, `o_sl_rx_mii_c[1]` corresponds to `o_sl_rx_mii_d[15:8]`, and so on.

continued...

Port Name	Width	Domain	Description
			If the value of a bit is 1, the corresponding data byte is a control byte. If the value of a bit is 0, the corresponding data byte is data. The Start of Packet byte (0xFB) and End of Packet byte (0xFD) are control bytes.

Figure 90. Receiving Data Using the RX MII Interface



The figure above shows how to read packets from the RX MII interface.

- The packets are MII encoded.
  - Each byte in `o_sl_rx_mii_d` has a corresponding bit in `o_sl_rx_mii_c` that indicates whether the byte is a control byte or a data byte; for example, `o_sl_rx_mii_c[2]` is the control bit for `o_sl_rx_mii_d[23:16]`.
- The byte order for the RX MII interface flows from right to left; the first byte that the core receives is `o_sl_rx_mii_d[7:0]`.
- The first bit that the core receives is `o_sl_rx_mii_d[0]`.

3.10.4. TX 8B/10B Interface

The TX 8b/10b interface is available only when you select the **Enable reconfiguration to 8b/10b datapath** parameter or when you select the 8b/10b CPRI line rate. For the CPRI PHY core power up in 64b/66b line rate, the IP core asserts these signals when you reconfigure the core at runtime to enter 8b/10b line rate.

Table 98. CPRI PHY TX 8B/10B Interface

Port Name	Width	Domain	Description
`i_sl_tx_d[n]`	16 bits per channel	`o_tx_clkout2[n]`	Indicates 8b/10b TX data for the corresponding CPRI PHY channel.
`i_sl_tx_c[n]`	2 bits per channel	`o_tx_clkout2[n]`	Indicates 8b/10b TX control for the corresponding CPRI PHY channel.

When you transmit the data using the TX 8b/10b interface:

- The frames are 8b/10b encoded.
  - Each byte in `i_sl_tx_d` has a corresponding bit in `i_sl_tx_c` that indicates whether the byte is a control byte or a data byte. For example, `i_sl_tx_c[1]` is the control bit for `i_sl_tx_d[15:8]`.
- The byte order for the TX interface flows from right to left and the first byte that the core transmits is `i_sl_tx_d[7:0]`.
- The first bit that the core transmits is `i_sl_tx_d[0]`.

3.10.5. RX 8B/10B Interface

The RX 8b/10b interface is available only when you select the **Enable reconfiguration to 8b/10b datapath** parameter or you select the 8b/10b CPRI line rate. For the CPRI PHY core power up in 64b/66b line rate, the IP core asserts these signals when you reconfigure the core at runtime to enter 8b/10b line rate.

Table 99. CPRI PHY RX 8B/10B Interface

Port Name	Width	Domain	Description
`i_sl_rx_d[n]`	16 bits per channel	`o_rx_clkout2[n]`	Indicates 8b/10b RX data for the corresponding CPRI PHY channel.
`i_sl_rx_c[n]`	2 bits per channel	`o_rx_clkout2[n]`	Indicates 8b/10b RX control for the corresponding CPRI PHY channel.

When you transmit the data using the RX 8b/10b interface:

- The frames are 8b/10b encoded.
  - Each byte in `i_sl_rx_d` has a corresponding bit in `i_sl_rx_c` that indicates whether the byte is a control byte or a data byte. For example, `i_sl_rx_c[0]` is the control bit for `i_sl_rx_d[7:0]`.
- The byte order for the RX interface flows from right to left and the first byte that the core receives is `i_sl_rx_d[7:0]`.
- The first bit that the core receives is `i_sl_rx_d[0]`.

3.10.6. Status Interface for 64B/66B Line Rate

This section lists the status ports for the CPRI PHY 64b/66b line rate. Each CPRI PHY channel has its own status ports.

Table 100. CPRI PHY Status Interface Signals for 64B/66B Interface

Port Name	Width	Domain	Description
`o_sl_tx_lanes_stable[n]`	1 bit per channel	Asynchronous	The IP core asserts this signal to indicate that TX PMA is ready. The signal deasserts when `i_csr_rst_n` or `i_tx_rst_n` is deasserted.
`o_sl_rx_pcs_ready[n]`	1 bit per channel	Asynchronous	The IP core asserts this signal to indicate that the corresponding RX datapath is ready to receive data. The signal deasserts when `i_csr_rst_n` or `i_rx_rst_n` is deasserted.

continued...

Port Name	Width	Domain	Description
o_sl_rx_block_lock[n]	1 bit per channel	Asynchronous	The IP core asserts this signal to indicate that 66b block alignment has completed for the corresponding CPRI PHY channel.
o_sl_rx_hi_ber[n]	1 bit per channel	Asynchronous	The IP core asserts this signal in accordance with IEEE 802.3 to indicate RX PCS is in Hi-Bit Error Rate (BER) state for the corresponding CPRI PHY channel.
o_sl_ehip_ready[n]	1 bit per channel	Asynchronous	The IP core asserts this signal after i_sl_csr_rst_n and i_sl_tx_rst_n is asserted to indicate that the CPRI PHY has completed all internal initialization, is ready to accept reconfiguration transactions and send data.

3.10.7. Status Interface for 8B/10B Line Rate

This section lists the status ports for the CPRI PHY 8b/10b line rate. Each CPRI PHY channel has its own status ports.

Table 101. CPRI PHY Status Interface Signals for 8B/10B Interface

Port Name	Width	Domain	Description
o_sl_tx_ready[n]	1 bit per channel	Asynchronous	The IP core asserts this signal to indicate that TX is ready for the corresponding transceiver in PMA direct mode. • 1: TX ready • 0: TX not ready
o_sl_rx_ready[n]	1 bit per channel	Asynchronous	The IP core asserts this signal to indicate that RX is ready for the corresponding transceiver in PMA direct mode. • 1: RX ready • 0: RX not ready
o_sl_rx_patterndetect[n]	1 bit per channel	o_rx_clkout2[n]	The IP core asserts this signal to indicate that K28.5 has been detected in the current word boundary of o_sl_rx_d or o_sl_rx_c and the received data from the RX PMA achieved the word alignment. This interface should be observed in conjunction with o_sl_rx_disperr and i_sl_rx_errdetect.
o_sl_rx_disperr[n]	2 bit per channel	o_rx_clkout2[n]	The IP core asserts this signal to indicate that it received 10-bit code or data group in the current word boundary of o_sl_rx_d or o_sl_rx_c has a disparity error. • Bit 0: Indicates status for lower data group. • Bit 1: Indicates status for higher data group.
o_sl_rx_errdetect[n]	2 bit per channel	o_rx_clkout2[n]	The IP core asserts this signal to indicate that it received 10-bit data group in the o_sl_rx_d or o_sl_rx_c has an 8b/10b code violation. • Bit 0: Indicates status for lower data group. • Bit 1: Indicates status for higher data group.

3.10.8. Serial I/O Pins

The CPRI PHY IP core always includes the serial I/O pins. The simulation files use these pins to provide serial connections to the core and for synthesis to define the pin positions of the transceivers used by the core.

Table 102. CPRI PHY Serial I/O Pins

Port Name	Width	Description
o_tx_serial[n]	1 bit per channel	TX side transceiver serial pins. One for each channel.
i_rx_serial[n]	1 bit per channel	RX side transceiver serial pins. One for each channel.

3.10.9. Reconfiguration Interfaces (Avalon-MM)

The E-Tile CPRI PHY IP core has the following reconfiguration interfaces:

- CPRI PHY core reconfiguration: This interface provides access to the Avalon-MM interface in the CPRI PHY core for each of the CPRI PHY channels.
- Transceiver reconfiguration: This interface provides access to the Avalon-MM interface in the transceivers and to the other Native PHY components.
- RS-FEC reconfiguration: This interface provides access to the Avalon-MM interface in the RS-FEC core. Four CPRI PHY channels share one RS-FEC core.

3.10.9.1. CPRI PHY Reconfiguration Interface

Table 103. CPRI PHY Reconfiguration Interface

Port Name	Width	Domain	Description
i_sl_cpri_reconfig_addr[n]	19 bits per channel	i_reconfig_clk	Indicates address for the CPRI PHY Avalon-MM interface in a selected channel.
i_sl_cpri_reconfig_read[n]	1 bit per channel	i_reconfig_clk	Read command for the CPRI PHY Avalon-MM interface in a selected channel.
i_sl_cpri_reconfig_write[n]	1 bit per channel	i_reconfig_clk	Write command for the CPRI PHY Avalon-MM interface in a selected channel.
o_sl_cpri_reconfig_readdata[n]	32 bits per channel	i_reconfig_clk	Data read from the CPRI PHY Avalon-MM interface in a selected channel.
o_sl_cpri_reconfig_readdata_valid[n]	1 bit per channel	i_reconfig_clk	When the signal is high, it indicates that read data from CPRI PHY Avalon-MM interface is valid in a selected channel.
i_sl_cpri_reconfig_writedata[n]	32 bits per channel	i_reconfig_clk	Data write to the CPRI PHY Avalon-MM interface in a selected channel.
o_sl_cpri_reconfig_writerequest[n]	1 bit per channel	i_reconfig_clk	Avalon-MM stalling signal for operations on the CPRI PHY Avalon-MM interface in a selected channel.

3.10.9.2. Transceiver Reconfiguration Interface

Table 104. Transceiver Reconfiguration Signals

Port Name	Width	Domain	Description
i_xcvr_reconfig_address[n]	19 bits per channel	i_reconfig_clk	Specifies transceiver Avalon memory-mapped interface address in a selected channel.
i_xcvr_reconfig_read[n]	1 bit per channel	i_reconfig_clk	The IP core asserts this transceiver read signal to start a read cycle in a selected channel.
i_xcvr_reconfig_write[n]	1 bit per channel	i_reconfig_clk	The IP core asserts this transceiver write signal to write data on reconfig_writedata bus in a selected channel.
i_xcvr_reconfig_writedata[n]	8 bits per channel	i_reconfig_clk	Specifies transceiver data to be written on a write cycle in a selected channel.
o_xcvr_reconfig_readdata[n]	8 bits per channel	i_reconfig_clk	Specifies transceiver data to be read by a ready cycle in a selected channel.
o_xcvr_reconfig_waitrequest[n]	1 bit per channel	i_reconfig_clk	Represents transceiver Avalon memory-mapped interface stalling signal in selected channel. The read and write cycle is only complete when this signal is low.

3.10.9.3. RS-FEC Reconfiguration Interface

The RS-FEC reconfiguration interface is only available when you generate the IP core variation for 10.13, 12.16, and 24.3 Gbps CPRI line bit rates.

Table 105. RS-FEC Reconfiguration Signals

Port Name	Width	Domain	Description
i_rsfec_reconfig_address[n]	11 bits per channel	i_reconfig_clk	Specifies the RS-FEC Avalon memory-mapped interface address in the selected channel.
i_rsfec_reconfig_read[n]	1 bit per channel	i_reconfig_clk	The IP core asserts RS-FEC read signal to start a read cycle in a selected channel.
i_rsfec_reconfig_write[n]	1 bit per channel	i_reconfig_clk	The IP core asserts RS-FEC write signal to write data on the reconfig_writedata bus in a selected channel.

continued...

Port Name	Width	Domain	Description
i_rsfec_reconfig_writedata[n]	8 bits per channel	i_reconfig_clk	Specifies RS-FEC data to be written on a write cycle in a selected channel.
o_rsfec_reconfig_readdata[n]	8 bits per channel	i_reconfig_clk	Specifies RS-FEC data to be read by ready cycle in a selected channel.
o_rsfec_reconfig_waitrequest[n]	1 bit per channel	i_reconfig_clk	Represents RS-FEC Avalon memory-mapped interface stalling signal in a selected channel. The read and write cycle is complete when this signal is low.

3.11. Registers

You can access the CPRI registers for the E-Tile CPRI PHY Intel FPGA IP using the Avalon-MM reconfiguration interface on each channel. The TX and RX RS-FEC registers are accessible through the RS-FEC reconfiguration interface.

Table 106. E-Tile CPRI PHY IP Core AVMM Address Ranges

Register Type	Address Range
PCS Registers	0x300-0x3FF
TX MAC Registers	0x400-0x4FF
CPRI PHY Registers	0xC00-0xC0F

Table 107. RS-FEC Reconfiguration Interface Register Base Addresses

Register Type	Address Range
TX and RX RS-FEC registers	0x000-0x2FF

3.11.1. PHY Registers

Table 108. PHY Registers

Address	Bit	Name	Description	Access	Reset
0x310	5	set_data_lock	Set data lock 1: Force PLL to lock to data.	RW	0x0
	4	set_ref_lock	Set ref lock 1: Force PLL to lock to reference.	RW	0x0
	2	soft_rx_rst	Soft RXP Reset 1: Resets the RX PCS and RX MAC.	RW	0x0
	1	soft_tx_rst	Soft TXP Reset 1: Resets the TX PCS and TX MAC.	RW	0x0
	0	eio_sys_rst	Ethernet IO System Reset 1: Resets the IP core (TX and RX MACs, Ethernet reconfiguration registers, PCS, and transceivers).	RW	0x0
0x321	3:0	eio_freq_lock	**Clock Data Recovery (CDR) PLL locked**	RO	0x0

continued...

Address	Bit	Name	Description	Access	Reset
			1: Corresponding physical lane's CDR has locked to reference for 10 and 25G links.		
0x30E	9	use_aligner	**Use RX PCS Alignment** 1:RX PCS has aligner turned on to align incoming data. 0: The RX PCS expects to receive aligned data, and its internal alignment logic is bypassed. • After power on, this register defaults to 0 • After i_csr_rst_n, this register is set depending on the **Select Ethernet IP Layers** parameter • In all modes that include RS-FEC, this register is set to 0 • In modes that do not include RS-FEC, this register is set to 1	RW	0x0
0x322	0	tx_pcs_ready	**TX Ready** 1: TX Datapath is out of reset, stable, and ready for use.	RO	0x0
0x323	19:0	frmerr	**Frame error(s) detected** 1: A frame error was detected on the corresponding lane. • For single lanes, only bit 0 is used • This bit is sticky, and must be cleared by asserting sclr_frame_error	RO	0x0
0x324	0	clr_frmerr	**Clear PHY frame error(s).** 1: Return all sticky frame error bits to 0.	RW	0x0
0x325	19	rx_pcs_in_rst	**Reset RX PCS** 1: Reset RX PCS. • Defaults to 0 after power-up and i_csr_rst_n asserted	RW	0x1
	17	tx_pcs_in_rst	**Reset TX PCS** 1: Reset TX PCS. • Defaults to 0 after power-up and i_csr_rst_n asserted.	RW	0x1
	14	force_hip_ready	**Override Hard IP ready** 1: Assert force_hip_ready, even if all the conditions for Hard IP ready have not been met. • Note that one of the conditions for force_hip_ready is the completion of configuration loading. If there is a problem with the configuration logic, force_hip_ready may be prevented from taking effect • This feature is provided for test and debug only • Defaults to 0 after power-up and i_csr_rst_n asserted.	RW	0x0
	2	trst	**TX Datapath reset**	RW	0x0

continued...

Address	Bit	Name	Description	Access	Reset
			1: Hold TX datapath in reset, including TX PLD, TX MAC, and TX PCS. • Performs same function as the `i_tx_rst_n` port • The IP core asserts `o_tx_rst` signal when this reset is active • Does not reset TX MAC statistics • Space the assertion and deassertion of reset to prevent sudden changes in power consumption • Defaults to 0 after power-up and `i_csr_rst_n` asserted.		
	0	`rrst`	**RX Datapath reset** 1: Hold RX datapath in reset, including RX PLD, RX MAC, and RX PCS. • Performs same function as the `i_rx_rst_n` port • The IP core asserts `o_rx_rst` when this reset is active • Does not reset RX MAC statistics • Space the assertion and deassertion of reset to prevent sudden changes in power consumption • Defaults to 1 after power-up • Defaults to 0 after `i_csr_rst_n` asserted.	RW	0x0
0x326	1	`hi_ber`	**Hi-BER** 1: One or more virtual lanes are in Hi-BER state.	RO	0x0
	0	`rx_aligned`	**RX PCS fully aligned** 1: The RX PCS is fully aligned and ready to start decoding data	RO	0x0
0x32A	31:0	`count`	**BER Count** • 32-bit count that increments each time the core enters BER_BAS_SH state. • Rolls over when maximum count is reached • Clears when the channel is reset • Can be captured using snapshot or RX shadow request	RO	0x0
0x32B	19:16	`ehip_rx_transfer_ready`	**EHIP/ELANE RX Channels Transfer Ready Status** 1: `transfer_ready` is 1.	RO	0x0
	3:0	`ehip_tx_transfer_ready`	**EHIP/ELANE TX Channels Transfer Ready Status** 1: `transfer_ready` is 1.	RO	0x0
0x341	31:0	`khz_rx`	**Recovered clock frequency** Recovered clock frequency/100, in KHz.	RO	0x0
0x342	31:0	`khz_tx`	**TX clock frequency** TX clock frequency/100, in KHz.	RO	0x0
0x351	24	`err_tx_avst_fifo_overflow`	**TX AVST FIFO Overflow**	RO	0x0

continued...

Address	Bit	Name	Description	Access	Reset
			• Indicates that the FIFO was written while full • Overflow would never happen—if it does, this indicates a problem with the way `i_valid` is being driven • Once asserted this bit holds value until the `i_clear_internal_error` port is asserted to clear it • This bit doesn't need to be polled— the IP asserts `o_internal_err` signal if this signal goes high		
	23	`err_tx_avst_fifo_empty`	**TX AVST FIFO ran empty unexpectedly** • Asserts when the TX FIFO runs empty (regardless of read enable) • Does not apply when in MAC mode • Empty should never happen—if it does, this indicates a problem with the way `i_valid` is being driven	RO	0x1
	22	`err_tx_avst_fifo_underflow`	**TX AVST FIFO Underflow** • Indicates that the FIFO was read when empty after steady state reading was established • Underflow should never happen—if it does, this indicates a problem with the way `i_valid` is being driven • Once asserted this bit holds value until the `i_clear_internal_error` port is asserted to clear it, or the TX datapath is reset • This bit doesn't need to be polled— `o_internal_err` is asserted if this signal goes high	RO	0x0
0x360	20	`use_hi_ber_monitor`	**Enable Hi-BER Monitor** 0: Turn off Hi-BER monitor 1: Turn on Hi-BER monitor • The Hi-BER monitor is turned on by default because it is used for standard compliance • Hi-BER is needed to support Auto-Negotiation, and is generally used to report poor link conditions • When the Hi-BER monitor is turned on, if a Hi-BER condition is detected, the PCS replaces incoming data with Local Fault blocks • Disable the Hi-BER monitor if you need to monitor RX data while in a Hi-BER state • At power-on, this register defaults to 0 • After `i_csr_rst_n` is asserted, the register is set to the value given by the `hi_ber_monitor` module parameter	RW	0x0
0x37A	20:0	`cycles`	**Timer window for BER measurements** Sets the timer window for BER measurements in clock cycles. The Ethernet Standard (IEEE 802.3) defines the required times for Hi-BER measurements for each rate. These times must be converted to clock cycles with the accuracy of within +1% and -25% of the specified times.	RW	0x312C7

continued...

Address	Bit	Name	Description	Access	Reset
			Note: If the clock rate you are using is different from the clock rate used to calculate the cycle count, you need to scale the cycle count. • 100GBASE-R4: 21'd201415 (from Clause 82, 0.5ms +1%,-25% at 402.3 MHz • 25GBASE-R1: 21'd806451 (from Clause 107, 2.0 ms +1%, -25% at 402.3 MHz • 10GBASE-R1: 21'd20141 (from Clause 49, 0.125ms +1%, -25% at 161.13 MHz • 10GBASE-R1: 21'd50403 (from Clause 49, 0.125ms +1%, -25% at 402.83 MHz The RX PCS must be reset after changing this value.		
0x37B	6:0	count	**Hi-BER Frame Errors** Sets the BER count that triggers hi_ber. The Ethernet Standard (IEEE 802.3) defines the appropriate setting for ber_invalid_count based on rate. • 100GBASE-R4: 7'd97 (from Clause 82) • 25GBASE-R1: 7'd97 (from Clause 107) • 10GBASE-R1: 7'd16 (from Clause 49) The RX PCS must be reset after changing this value.	RW	0x61
0x37C	31:0	count	**Error block count** • Counts the number of error blocks produced by the RX PCS decoder • Valid only when the RX PCS decoder is used and alignment is achieved • The error blocks can be received from the remote link, or generated by violations of the Ethernet Standard 64B66B encoding specification • The counter is 32-bit wide and rolls over when the max count is reached • The counter is reset when the RX datapath is reset, or the RX PCS is reset	RO	0x0

3.11.2. CPRI PHY Registers

These registers use 32-bit addresses; they are not byte-addressable.

Table 109. CPRI PHY Registers

Address	Bit	Name	Description	Access	Reset
0xC00	31:10	Reserved			
	9:5	rx_bitslipboundary_sel	Reports the number of bits the 8B/10B RX PCS block slipped to achieve a deterministic latency.	RO	0x0
	4	cpri_fec_en	Indicates whether the RS-FEC block is enabled. Deterministic latency uses this register. • 0: Disable RS-FEC • 1: Enable RS-FEC You must reset TX and RX datapaths after changing this bit.	RW	The reset value depends on the selected IP variant. For example, the reset value is 1 if the instantiated IP variant is 24.33024G (64/66b) with RS-FEC.
	3:0	cpri_rate_sel	Selects the CPRI speed. EFIFO and deterministic latency use this register. • 0x2: 2.4 Gbps • 0x3: 3 Gbps • 0x4: 4.9 Gbps • 0x5: 6.1 Gbps • 0x6: 9.8 Gbps • 0x9: 10.1 Gbps • 0xA: 12.1 Gbps • 0xB: 24.3 Gbps You must reset TX and RX datapaths after changing this bit.	RW	The reset value depends on the selected IP variant. For example, the reset value is 0xB if the instantiated IP variant is 24.33024G (64/66b) with RS-FEC.
0xC01	31:2	Reserved			
	1	dl_master_reset	Indicates the master reset that allows deterministic latency (DL) measurement to be retaken: • 0: Reset disabled • 1: Reset enabled	RW	0x1
	0	measure_valid	Indicates whether the deterministic values are valid • 0: Invalid • 1: Valid	RO	0x0
0xC02	31:21	Reserved			
	20:0	tx_delay	Indicates deterministic latency measurement values for TX data path latency in fixed format (Q13.8). This value is valid only if measure_valid = 1.	RO	0x0
0xC03	31:21	Reserved			
	20:0	rx_delay	Indicates deterministic latency measurement values for RX data path latency in fixed format (Q13.8).	RO	0x0

continued...

Address	Bit	Name	Description	Access	Reset
			This value is valid only if `measure_valid = 1`.		
0xC04-0xC0F	31:0	Reserved			

3.11.3. PMA Registers

For information on PMA registers, refer to the *E-Tile Transceiver PHY User Guide*: *PMA Register Map*.

Related Information

- E-Tile Transceiver PHY User Guide
- PMA Register Map

3.11.3.1. Minimizing PMA Adaptation Time

When you change the line bit rate of the E-Tile CPRI PHY Intel FPGA IP, you need to calibrate the PMA to obtain the optimal performance. After you initiate a rate switch, the E-Tile CPRI PHY IP requires 100 ms for PMA to be ready. To meet this requirement, you need to minimize the PMA adaptation time by configuring the PMA adaptive engine to use adaptation presets through the PMA registers.

Select the PMA parameter by setting the PMA attribute code 0x2C to PMA attribute value 0x118 into the PMA attribute code registers.

1. Write 0x84[7:0] = 0x18.
2. Write 0x85[7:0] = 0x1.
3. Write 0x86[7:0] = 0x2C.
4. Write 0x87[7:0] = 0x0.
5. Write 0x90[0] = 1'b1.
6. Read 0x8A[7]. It should be 1.
7. Read 0x8B[0], until it changes to 0.
8. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

Write a value to the PMA parameter by setting the PMA attribute code 0x6C to PMA attribute value 0x0 into the PMA attribute code registers.

9. Write 0x84[7:0] = 0x0.
10. Write 0x85[7:0] = 0x00.
11. Write 0x86[7:0] = 0x6C.
12. Write 0x87[7:0] = 0x00.
13. Write 0x90[0] = 1'b1.
14. Read 0x8A[7]. It should be 1.
15. Read 0x8B[0], until it changes to 0.
16. Write 0x8A[7] to 1 to clear the 0x8A[7] flag.

Related Information

- Configuring a PMA Parameter Tunable by the Adaptive Engine
  More information about setting fixed value to PMA adaptive engine.
- Receiver PMA
  More information about PMA adaptation.

3.11.4. RS-FEC Registers

For information on RS-FEC registers, refer to the *E-Tile Transceiver PHY User Guide*: *RS-FEC Registers*.

Related Information

E-Tile Transceiver PHY User Guide: RS-FEC Registers

3.12. Document Revision History for the E-tile CPRI PHY Intel FPGA IP

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.03.29	21.1	19.4.2	Modified the values in *Table: Deterministic Latency Measurement for Each Variant*.
2020.06.29	20.2	19.4.1	• Modified the value of UI constant in TX delay deterministic latency calculation formula for 2.4376/3.0720 Gbps CPRI line rates in *Table: Deterministic Latency Measurement for Each Variant*. • Added new register `dl_master_reset` in *Table: CPRI PHY Registers*.
2020.04.28	20.1	19.4.1	• Added resource utilization numbers for Intel Agilex devices in section *Resource Utilization*. • RX clock domain changed to use the RX recovered clock. • Updated *Figure: E-tile CPRI PHY (FEC On) Master-Slave Configuration* to include RX clock domain change. • Updated information in *Table: Reset Signal and Register Functions* . • Updated signal domain information from `o_tx_clkout2[n]` to `o_rx_clkout2[n]`in the following tables: — *Table: CPRI PHY RX MII Interface* — *Table: CPRI PHY RX 8B/10B Interface* — *Table: CPRI PHY Status Interface Signals for 8B/10B Interface* • Corrected address range for the CPRI PHY registers in *Table: E-Tile CPRI PHY IP Core AVMM Address Ranges*. • Updated [3:0] bit description of the 0xC00 register for 3.0, 6.1, and 10.1 Gbps data rates in *Table: CPRI PHY Registers*.
2020.03.30	19.4	19.4.0	Modified the description of latency factors RxBitSlipH and RxBitSlipL in *Table: Latency Calculation Description*.
2020.03.09	19.4	19.4.0	• Corrected TX Delay and RX Delay values in *Table: Deterministic Latency Measurement for Each Variant* • Added factors RxBitSlipH and RxBitSlipL in *Latency Calculation Description*

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.12.16	19.4	19.4.0	• Added support for the Intel Agilex device with E-tile transceivers. • The E-Tile CPRI PHY IP now supports the following new CPRI line rates: 3.0720, 6.1440, 10.1316 (with RS-FEC), 12.1651 (with and without RS-FEC). • Updated resource utilization numbers in *Table: Resource Utilization for Selected Variations*. • Added *Figure: E-Tile CPRI PHY (FEC On) Master-Slave Configuration* in section *E-Tile CPRI PHY Intel FPGA IP Channel Placement*. • Updated the following sections to include RS-FEC clocking mode information: — *One 24.33024 Gbps channel with RS-FEC* — *Two 24.33024 Gbps Channel with RS-FEC* — Three 24.33024 Gbps Channel with RS-FEC — Four 24.33024 Gbps Channel with RS-FEC • Updated *Restrictions* section to include new CPRI line rates information. • Added new parameter **RSFEC Clocking Mode** in *Table: Parameter Settings: IP Tab*. • Updated **PHY Reference Clock Frequency** for new supported CPRI line rates in *Table: Parameter Settings: IP Tab*. • Updated description for the UI_constant_offset_[rx,tx] in *Table: Latency Calculation Description*. • Updated *Table: Reset Signal and Register Functions* in section *Soft Reset Sequencer*.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.10.22	19.2	19.2.0	Corrected the frequency value of `i_clk_ref` for CPRI line rates 2.4/4.9/9.8 Gbps in *Table: CPRI PHY Clock Input Signals*.
2019.08.07	19.2	19.2.0	• The E-Tile CPRI PHY IP now supports CPRI line rates: 2.4376, 4.9152, 9.8304, and 24.33024 Gbps (without RS-FEC). • Updated *Figure: E-tile CPRI PHY Block Diagram*. • Added new parameters **First RSFEC Lane** and **Enable reconfiguration to 8b/10b datapath** in *Table: Parameter Settings: IP Tab*. • Updated the **NPDME** parameter description in section *Parameter Settings*. • Added the following new sections: — TX 8B/10B Interface — RX 8B/10B Interface — Status Interface for 8B/10B Line Rate • Clarified that RX signals are resynchronized to the TX domain. • Added deterministic latency calculation equation for the new supported CPRI line rates in section *Deterministic Latency Calculation*. • Updated section *E-Tile CPRI PHY Intel FPGA IP Channel Placement*. • Updated section *CPRI PHY Functional Blocks*. • Modified port names in the following sections: — CPRI PHY Reconfiguration Interface — Transceiver Reconfiguration Interface — RS-FEC Reconfiguration Interface • Added calibration requirement in section *Minimizing PMA Adaptation Time*.
2019.05.17	19.1	19.1	Initial release.

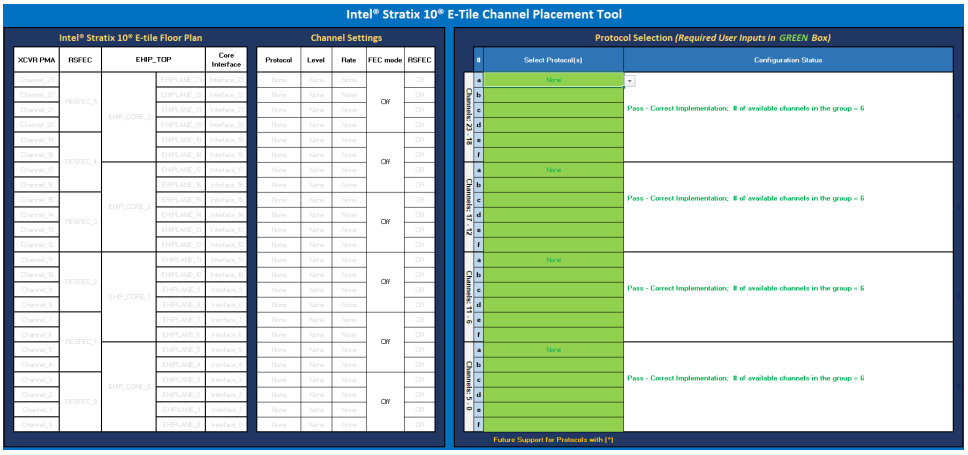
4. Supported Tools

4.1. E-Tile Channel Placement Tool

E-tile supports datacenters, 5G networks, Smart Grid and other market segments. Ethernet, CPRI and OTN are the backbone of these emerging and traditional technologies. The *E-Tile Channel Placement Tool*, in conjunction with the *Device Family Pin Connection Guidelines*, allows you to swiftly plan protocol placements in the product prior to reading comprehensive documentation and implementing designs in Intel Quartus Prime.

The Excel-based *E-Tile Channel Placement Tool*, supplemented with **Instructions**, **Legend**, and **Revision** tabs, is available for download at E-Tile Channel Placement Tool.

Figure 91. E-Tile Channel Placement Tool



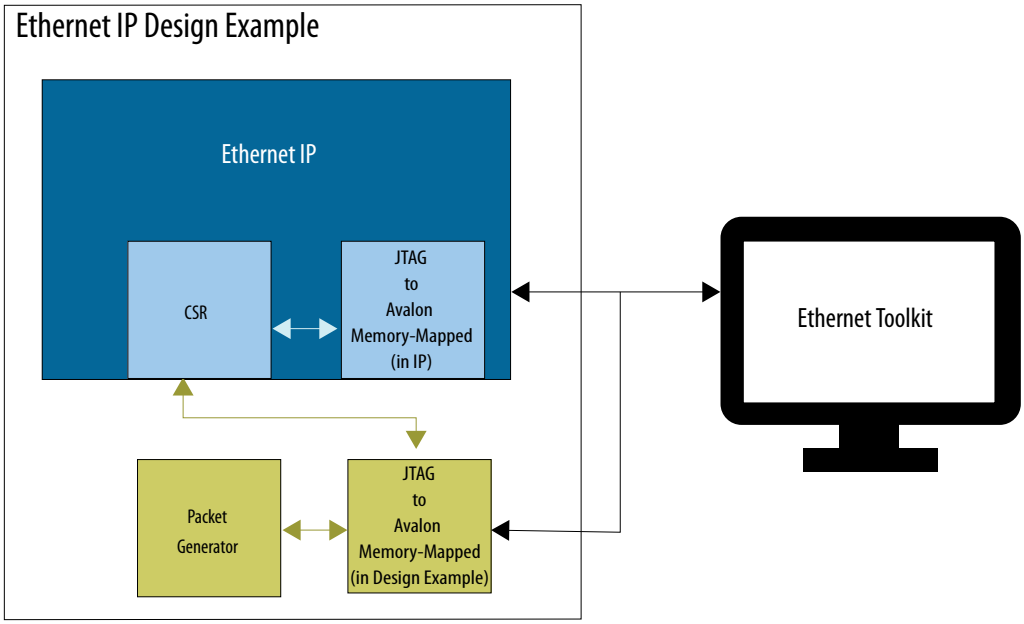
Related Information

- Intel Stratix 10 Device Family Pin Connection Guidelines
- Intel Agilex Device Family Pin Connection Guidelines

4.2. Ethernet Toolkit Overview

The Ethernet Toolkit is a TCL based debugging tool that allows you to interact with an Ethernet Intel FPGA IP in real time.

Figure 92. Block Diagram of the Ethernet Toolkit



You can use the Ethernet Toolkit with hardware design that has standalone Ethernet IP. You can also use the Ethernet Toolkit with an Intel Quartus Prime generated Ethernet IP design example.

4.2.1. Features

The Ethernet Toolkit offers the following features when used with hardware design that has standalone Ethernet IP as well as with an Intel Quartus Prime generated Ethernet IP design example:

- Verifies the status of the Ethernet link.
- Reads and writes to status and configuration registers of the IP.
- Displays the values of TX/RX status and statistics registers.
- Ability to assert and deassert IP resets.
- Verifies the IPs error correction capability.

The Ethernet Toolkit also offers some additional features when used with an Intel Quartus Prime generated Ethernet IP design example:

- Provides access to the example design packet generator.
- Execute testing procedures to verify the functionality of Ethernet IPs.
- Enable and disable MAC loopback.
- Set source and destination MAC addresses.

Related Information

Ethernet Toolkit User Guide

4.3. Document Revision History for the E-tile Channel Placement Tool and the Ethernet Link Inspector

Document Version	Intel Quartus Prime Version	Changes
2020.10.05	20.3	Added new topic: *Ethernet Toolkit Overview*.
2020.01.31	19.4	Added the *Intel Agilex Device Family Pin Connections Guidelines* link.
2019.11.15	19.3	Added Related Information link for the Intel Agilex device documents.
2019.08.07	19.2	Added *Ethernet Link Inspector* section.
2019.04.19	18.1.1	Initial release.