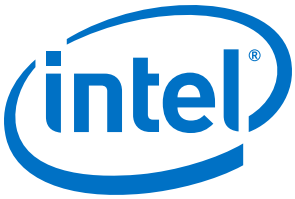


H-tile Hard IP for Ethernet Intel FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **19.3**

IP Version: **19.2.0**

Subscribe
Send Feedback

UG-20121 | 2019.10.31
Latest document on the web: **PDF** | **HTML**

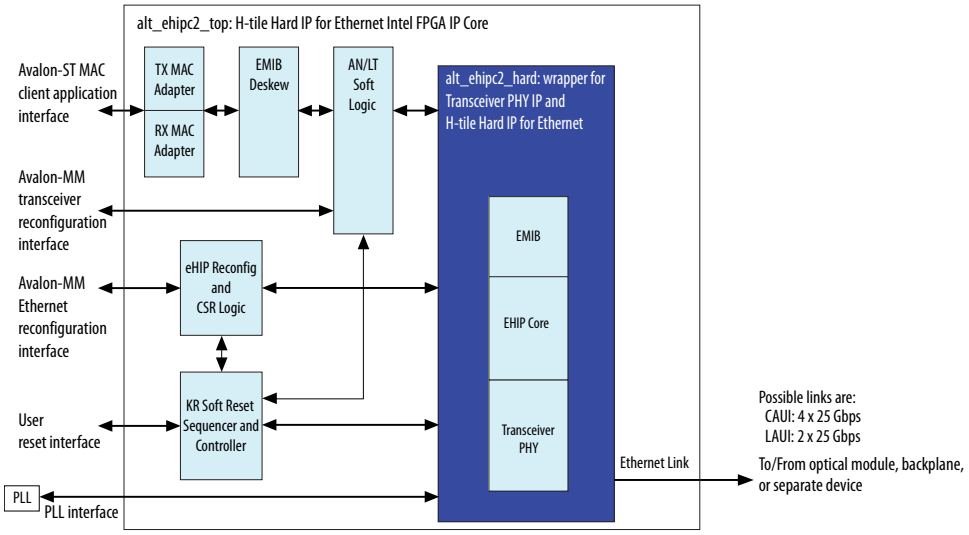
Contents

1. About the H-tile Hard IP for Ethernet Intel FPGA IP Core

Intel® Stratix® 10 H-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 25G Ethernet Consortium.

The H-tile Hard IP for Ethernet Intel FPGA IP core provides access to this hard IP at Ethernet data rates of 50 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.

Figure 1. H-tile Hard IP for Ethernet Intel FPGA IP Core



The IP core provides standard MAC and physical coding sublayer (PCS) functions with a variety of configuration and status registers.

The IP core is available with a 50GBASE-R2 Ethernet channel or a 100GBASE-R4 Ethernet channel. For either Ethernet data rate, you can choose a MAC+PCS, or a PCS Only, or an OTN, or a FlexE variation.

Note: The H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature. For further inquiries, contact your nearest Intel sales representative or file an Intel Premier Support (IPS) case on https://www.intel.com/content/www/us/en/programmable/my-intel/mal-home.html.

Table 1. Client Interfaces for IP Core Variations

IP Core Variation	Client Interface Type	Client Interface Width (Bits)	
		50GBASE-R2	100GBASE-R4
MAC+PCS	Avalon® Streaming (ST)	128	512
PCS Only	Media Independent Interface (MII)	128	256
OTN	PCS66 interface	128	256
FlexE	PCS66 interface	128	256

The 50GBASE-R2 Ethernet channel maps to two 25.78125 Gbps links and the 100GBASE-R4 Ethernet channel maps to four 25.78125 Gbps links. The FPGA serial transceivers are compliant with the *IEEE 802.3-2015 High Speed Ethernet Standard* CAUI-4 specification and the *25G & 50G Ethernet Specification, Draft 1.6*. The IP core configures the transceivers to implement the relevant specification for your IP core variation. You can connect the transceiver interfaces directly to an external physical medium dependent (PMD) optical module or to another device.

1.1. IP Core 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 H-tile Hard IP for Ethernet Intel FPGA IP core variations are in full-duplex mode. These IP core variations offer the following features:

Table 2. H-tile Hard IP for Ethernet Intel FPGA Features

Features	Description
PCS	Hard IP logic that interfaces seamlessly to Intel Stratix 10 FPGA 25.78125 Gbps serial transceivers.
	LAUI or CAUI-4 external interface consisting of two or four FPGA hard serial transceiver lanes operating at 25.78125 Gbps.
	Supports LAUI or CAUI-4 links based on 64B/66B encoding with data striping and alignment markers to align data from multiple lanes.
	Supports Autonegotiation (AN) as defined in *IEEE Standard 802.3-2015 Clause 73* and the *25G Ethernet Consortium Schedule Draft 1.6*
	Support link training (LT) as defined in *IEEE Standard 802.3-2015 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.
Optical Transport Network (OTN)	Optional 50/100GE constant bit rate (CBR), with TX and RX PCS66 bit encoding and scrambling disabled. *Note:* The H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature. For further inquiries, contact your nearest Intel sales representative or file an Intel Premier Support (IPS) case on https://www.intel.com/content/www/us/en/programmable/my-intel/mal-home.html.
Flexible Ethernet (FlexE)	Optional 50/100GE constant bit rate (CBR) with TX and RX PCS66 scrambler/descrambler.

continued...

Features	Description
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.
	RX and TX preamble pass-through options for applications that require proprietary user management information transfer.
	Optional TX MAC source address insertion.
	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.
	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.
Frame Monitoring and Statistics	RX cyclic redundancy check (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.
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.

continued...

Features	Description
	TX error insertion capability supports test and debug.
	Optional access to Native PHY Debug Master Endpoint (NPDME) for debugging or monitoring PHY signal integrity.
User System Interface	Avalon Memory-Mapped (Avalon-MM) management interface to access the IP core control and status registers.
	Avalon-ST data path interface connects the MAC to client logic with the start of frame in the most significant byte (MSB) in MAC+PCS variations. Interface for 50GBASE-R2 variations has data width 128 bits; interface for 100GBASE-R4 variations has 512 bits, to ensure the data rate despite this RX client interface SOP alignment and RX and TX preamble passthrough option.
	MII data path interface connects the PCS to client logic in PCS Only variations. Interface for 50GBASE-R2 variations has data width 128 bits; interface for 100GBASE-R4 variations has 256 bits.
	Hardware and software reset control.
	Supports Synchronous Ethernet (Sync-E) by providing a CDR recovered clock output signal to the device fabric.

Related Information

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

1.2. IP Core Device Family and Speed Grade Support

The following sections list the device family and device speed grade support offered by the Intel FPGA IP:

H-tile Hard IP for Ethernet Intel FPGA IP Core Device Family Support on page 12

H-tile Hard IP for Ethernet Intel FPGA IP Core Device Speed Grade Support on page 13

1.2.1. H-tile Hard IP for Ethernet Intel FPGA IP Core Device Family Support

Table 3. 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 4. H-tile Hard IP for Ethernet Intel FPGA IP Core Device Family Support

Shows the level of support offered by the H-tile Hard IP for Ethernet Intel FPGA IP core for each Intel FPGA device family.

Device Family	Support
Intel Stratix 10	Advance H-tile devices only
Other device families	No support

Related Information

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

1.2.2. H-tile Hard IP for Ethernet Intel FPGA IP Core Device Speed Grade Support

The H-tile Hard IP for Ethernet Intel FPGA IP core supports Intel Stratix 10 H-tile devices with these speed grade properties:

- Transceiver speed grade: -1 or -2
- Core speed grade: -1 or -2

1.3. IP Core Verification

To ensure functional correctness of the H-tile Hard IP for Ethernet Intel FPGA IP core, Intel performs extensive validation through both simulation and hardware testing. Before releasing a version of the H-tile Hard IP for Ethernet Intel FPGA IP core, 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. Any exceptions to this verification are reported in the *Intel FPGA IP Release Notes*. Intel does not verify compilation with IP core versions older than the previous release.

Related Information

- Knowledge Base errata for the H-tile Hard IP for Ethernet Intel FPGA IP core
  Exceptions to functional correctness that manifest in software releases 17.1 and later are documented in the H-tile Hard IP for Ethernet Intel FPGA IP core errata.
- H-tile Hard IP for Ethernet Intel FPGA IP Core Release Notes
  Changes to the H-tile Hard IP for Ethernet Intel FPGA IP core in major software releases are noted in the Intel FPGA IP Release Notes.

1.3.1. Simulation Environment

Intel performs the following tests on the 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

1.3.2. Compilation Checking

Intel performs compilation testing on an extensive set of 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.

1.3.3. Hardware Testing

Intel performs hardware testing of the key functions of the H-tile Hard IP for Ethernet Intel FPGA IP core using standard 50 and100Gbps Ethernet network test equipment and optical modules. The Intel hardware tests of the H-tile Hard IP for Ethernet Intel FPGA IP core also ensure reliable solution coverage for hardware related areas such as performance, link synchronization, and reset recovery.

1.4. Resource Utilization

Resource utilization changes depending on the parameter settings you specify in the H-tile Hard IP for Ethernet Intel FPGA parameter editor. This IP core is not as sensitive to parameter settings as other IP cores, because much of the functionality is in the Hard IP, but some parameters, such as the selection of a MAC+PCS, a PCS Only, an OTN[1], or a FlexE variation, do affect resource utilization on the device. If you select a MAC+PCS variation, the IP core requires additional resources to implement the additional functionality.

Note: Intel advises that resource utilization numbers are approximate, as the Intel Quartus Prime Pro Edition Fitter assigns resources based on the entirety of your design. The numbers below result from a single run on a simple design. Your results may vary.

(1) The H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature. For further inquiries, contact your nearest Intel sales representative or file an Intel Premier Support (IPS) case on https://www.intel.com/content/www/us/en/programmable/my-intel/mal-home.html.

Table 5. IP Core FPGA Resource Utilization

Lists the resources and expected performance for selected variations of the H-tile Hard IP for Ethernet Intel FPGA IP core in an Intel Stratix 10 device.

These results were obtained using the Intel Quartus Prime Pro Edition v18.1 software. All parameters are at their default values except the parameters listed in the table:

- **Ethernet rate**
- **Ethernet IP layers**
- **Enable AN/LT**, **Enable Auto-Negotiation**, and **Enable Link Training** are either all set to the value of **True** or all to the value of **False**.
- The numbers of ALMs and logic registers are rounded up to the nearest 100.
- The numbers of ALMs, before rounding, are the **ALMs needed** numbers from the Intel Quartus Prime Pro Edition Fitter Report.

Ethernet rate	Ethernet IP layers	Enable AN/LT	ALMs	Dedicated Logic Registers	Memory M20K
100G	MAC+PCS	True	12200	17530	12
		False	4900	7490	2
	PCS Only	False	1500	2513	0
	OTN(1)	False	1500	2355	0
	FlexE	False	1500	2355	0
50G	MAC+PCS	True	7000	9730	10
		False	2100	2941	0
	PCS Only	False	1080	1655	0
	OTN(1)	False	1270	1658	0
	FlexE	False	1100	1616	0

Related Information

Fitter Resources Reports in the Quartus Prime Help
Information about Quartus Prime resource utilization reporting, including **ALMs needed**.

1.5. 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 6. H-tile Hard IP for Ethernet Intel FPGA IP Core Current Release Information

Item	Description
IP Version	19.2.0
Intel Quartus Prime Version	19.3
Release Date	2019.10.31
Ordering Code	IP-ETH-HTILEHIP

2. Getting Started

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

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.

2.1. Installing and Licensing Intel FPGA IP Cores

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

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

Figure 2. IP Core Installation Path

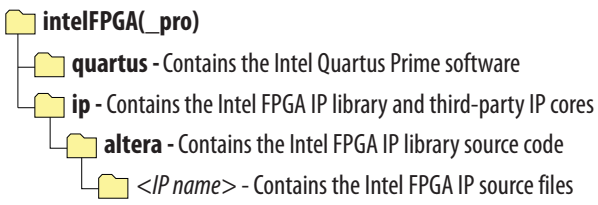


Table 7. IP Core Installation Locations

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*

2.2. Specifying the IP Core Parameters and Options

The H-tile Hard IP for Ethernet Intel FPGA 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 H-tile Hard IP for Ethernet Intel FPGA 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** and select a production H-tile device that meets the speed grade requirements for the IP core.
   c. Click **Finish**.
2. In the IP Catalog, locate and select **H-tile Hard IP for Ethernet**. 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 30 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 *Intel Stratix 10 H-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. Close the IP generator window. 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

H-tile Hard IP for Ethernet Intel FPGA Design Example User Guide
Information about generating the H-tile Hard IP for Ethernet Intel FPGA design example.

2.3. 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 *H-tile Hard IP for Ethernet Intel FPGA Design Example User Guide*.

Figure 3. H-tile Hard IP for Ethernet Intel FPGA IP Core Generated Files

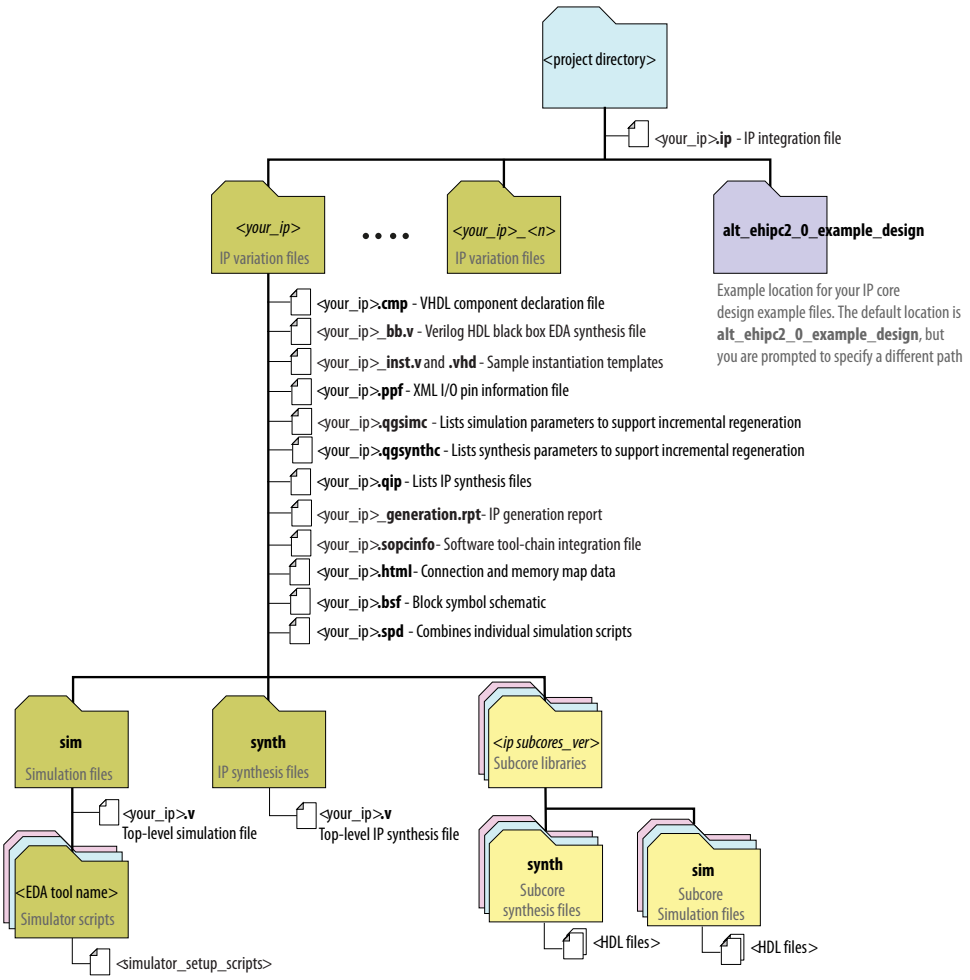


Table 8. IP Core Generated Files

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*>.bsf	A Block Symbol File (.bsf) representation of the IP variation for use in Quartus Prime Block Diagram Files (**.bdf**).
<*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*>.ppf	The Pin Planner File (.ppf) stores the port and node assignments for IP components created for use with the Pin Planner.
<*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*>.regmap	If IP contains register information, .regmap file generates. The .regmap file describes the register map information of master and slave interfaces. This file complements the .sopcinfo file by providing more detailed register information about the system. This enables register display views and user customizable statistics in the System Console.
<*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

H-tile Hard IP for Ethernet Intel FPGA Design Example User Guide
Information about the H-tile Hard IP for Ethernet Intel FPGA design example file structure.

2.4. 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 21

Pin Assignments on page 23

Adding the Transceiver PLLs on page 24

Clock Requirements on page 27

Placement Settings for the H-tile Hard IP for Ethernet Intel FPGA IP Core on page 28

2.4.1. Channel Placement

Each H-tile provides a single Hard IP for Ethernet block. 100GBASE-R4 variations of the IP core use channels 0 through 4 in the top transceiver bank of the tile, and leave channel 5 available for use by other parts of your design. 50GBASE-R2 variations use channels 0 and 1 in the top transceiver bank of the tile, and leave channels 2 through 5 available for use by other parts of your design.

Figure 4. 100GBASE-R4 IP Core Channel Placement in H-tile

fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4	GXT Channel 3	EMIB GXT Channel 4
	GXT Channel 3	GXT Channel 2	EMIB GXT Channel 3
fPLL	GX Channel 2	100G Ethernet HIP	EMIB GX Channel 2
ATXPLL	GXT Channel 1	GXT Channel 1	EMIB GXT Channel 1
	GXT Channel 0	GXT Channel 0	EMIB GXT Channel 0
fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4		EMIB GXT Channel 4
	GXT Channel 3		EMIB GXT Channel 3
fPLL	GX Channel 2		EMIB GX Channel 2
ATXPLL	GXT Channel 1		EMIB GXT Channel 1
	GXT Channel 0		EMIB GXT Channel 0
fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4		EMIB GXT Channel 4
	GXT Channel 3		EMIB GXT Channel 3
fPLL	GX Channel 2		EMIB GX Channel 2
ATXPLL	GXT Channel 1		EMIB GXT Channel 1
	GXT Channel 0		EMIB GXT Channel 0
fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4		EMIB GXT Channel 4
	GXT Channel 3		EMIB GXT Channel 3
fPLL	GX Channel 2		EMIB GX Channel 2
ATXPLL	GXT Channel 1		EMIB GXT Channel 1
	GXT Channel 0		EMIB GXT Channel 0

Figure 5. 50GBASE-R2 IP Core Channel Placement in H-tile

fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4	GXT Channel 3	EMIB GXT Channel 4
	GXT Channel 3	GXT Channel 2	EMIB GXT Channel 3
fPLL	GX Channel 2	100G Ethernet HIP	EMIB GX Channel 2
ATXPLL	GXT Channel 1	GXT Channel 1	EMIB GXT Channel 1
	GXT Channel 0	GXT Channel 0	EMIB GXT Channel 0
fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4		EMIB GXT Channel 4
	GXT Channel 3		EMIB GXT Channel 3
fPLL	GX Channel 2		EMIB GX Channel 2
ATXPLL	GXT Channel 1		EMIB GXT Channel 1
	GXT Channel 0		EMIB GXT Channel 0
fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4		EMIB GXT Channel 4
	GXT Channel 3		EMIB GXT Channel 3
fPLL	GX Channel 2		EMIB GX Channel 2
ATXPLL	GXT Channel 1		EMIB GXT Channel 1
	GXT Channel 0		EMIB GXT Channel 0
fPLL	GX Channel 5		EMIB GX Channel 5
ATXPLL	GXT Channel 4		EMIB GXT Channel 4
	GXT Channel 3		EMIB GXT Channel 3
fPLL	GX Channel 2		EMIB GX Channel 2
ATXPLL	GXT Channel 1		EMIB GXT Channel 1
	GXT Channel 0		EMIB GXT Channel 0

2.4.2. Pin Assignments

When you integrate your H-tile Hard IP for Ethernet Intel FPGA IP core instance into 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 H-tile devices offer a single hard IP for Ethernet block on each H-tile. Your design must not include pin assignments that conflict with its location. In devices with multiple H-tiles, you can specify the H-tile to which the Ethernet link serial pins

should map. Refer to *100G Configuration* and *50G Configuration* in the *Ethernet Hard IP* section of the *Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide* or the figures in *Channel Placement*.

Related Information

- Quartus Prime Help
  For information about the Quartus Prime software, including virtual pins and the IP Catalog.
- Ethernet Hard IP section of Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
  Information about constraints on transceiver configuration for Hard IP for Ethernet in Intel Stratix 10 H-tile devices.

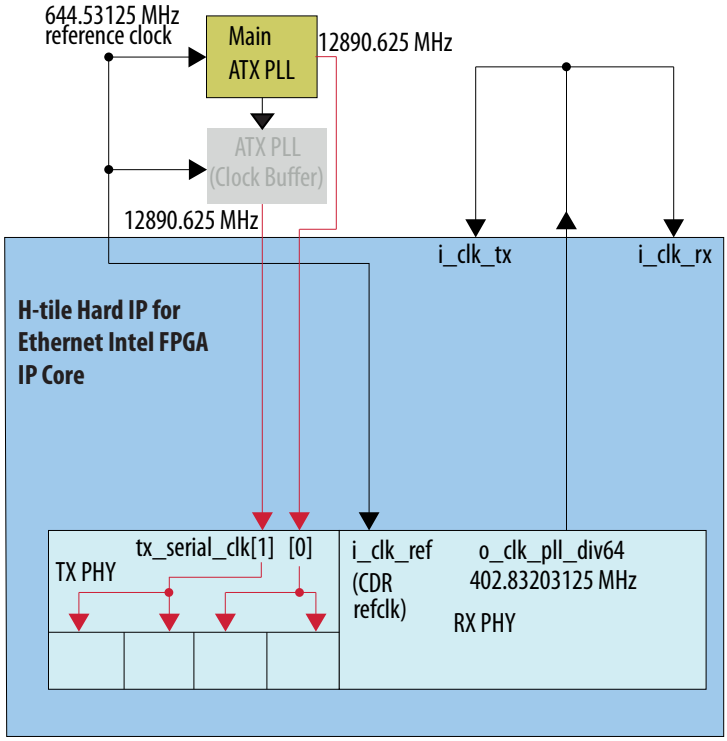
2.4.3. Adding the Transceiver PLLs

The H-tile Hard IP for Ethernet Intel FPGA IP core requires one or two TX transceiver PLLs that are not part of the IP core, to compile and to function correctly in hardware. On Intel Stratix 10 devices, only the ATX PLL supports the required data rate.

The transceiver PLLs you configure are physically present on the device, but the H-tile Hard IP for Ethernet Intel FPGA IP core does not configure and connect them. The required number of ATX PLLs is two for 100GBASE-R4 variations and one for 50GBASE-R2 variations. Each ATX PLL drives the clocks for two transceiver channels.

Figure 6. PLL Configuration Example for 100GBASE-R4 (Synchronous Mode) IP Core Variation

The TX transceiver PLLs are instantiated with two Intel Stratix 10 ATX PLL IP cores, one as the main ATX PLL and another as a clock buffer. The TX transceiver PLLs must always be instantiated outside the H-tile Hard IP for Ethernet Intel FPGA IP core.



To configure an ATX PLL as the main ATX PLL:

- Select **L-Tile/H-Tile Transceiver ATX PLL Intel Stratix 10 FPGA IP**.
- In the parameter editor, set the following parameter values:
  - Set **VCCR_GXB and VCCT_GXB supply voltage for the Transceiver** to **1_1V**.
  - Set **Primary PLL clock output buffer** to **GXT clock output buffer**.
  - Turn on **Enable GXT clock output port to above ATX PLL (gxt_output_to_abv_atx)** or **Enable GXT clock output port to below ATX PLL (gxt_output_to_blw_atx)**.
  - Turn on **Enable GXT local clock output port (tx_serial_clk_gxt)**.
  - Turn on **Enable GXT clock buffer to above ATX PLL**.
  - Set **GXT output clock source** to **Local ATX PLL**.
  - Set **PLL output frequency** to **12890.625 MHz**. The transceiver performs dual edge clocking, using both the rising and falling edges of the input clock from the PLL. Therefore, this PLL output frequency setting supports a 25.78125 Gbps data rate through the transceiver.
  - Set **PLL auto mode reference clock frequency** to the value you specified for the **PHY Reference Frequency** parameter.

To configure an ATX PLL as a GXT transmit PLL with GXT clocks to adjacent GXT channels and GXT clock buffer ATX PLLs:

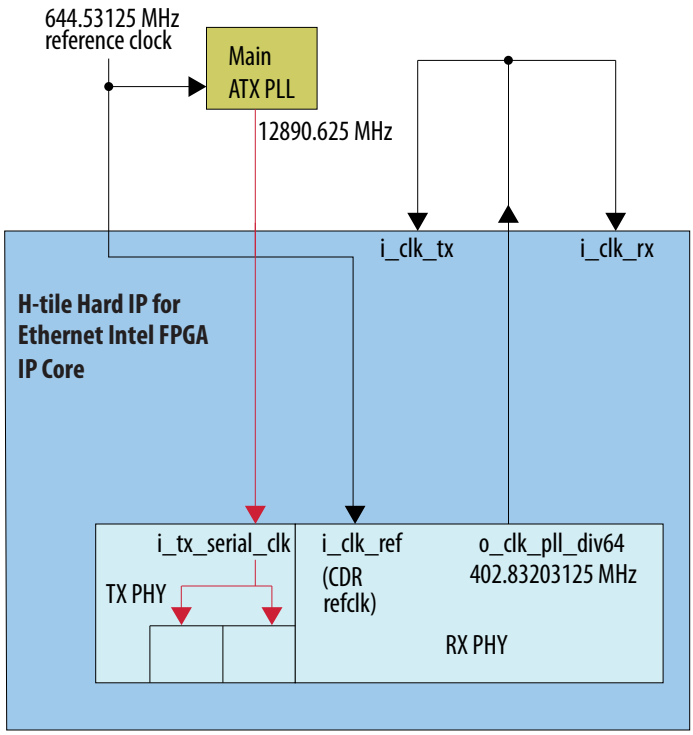
- Set the **ATX PLL operation mode** drop-down as **GXT mode**.
- Set the **Enable GXT local clock output port (tx_serial_clk_gxt)** .
- Set the **GXT output clock source** drop-down as **Local ATX PLL.**
- Select the **Enable GXT output port** to **Input from ATX PLL above (gxt_input_from_abv_atx)** or **Input from ATX PLL below (gxt_input_from_blw_atx)**.
- Tie off the `pll_refclk0` pin to REFCLK pin, if the GXT clock buffer ATX PLL is not reconfigured to a GXT transmit PLL or GX transmit PLL.

Each PLL drives the `tx_serial_clk` input of two of the H-tile Hard IP for Ethernet Intel FPGA IP core PHY links. You must connect the PLLs to the H-tile Hard IP for Ethernet Intel FPGA IP core as follows:
Connect `i_tx_pll_locked[0]` and `i_tx_pll_locked[1]` from the H-tile Hard IP for Ethernet Intel FPGA IP core to the `pll_locked` signal of the main ATX PLL.

PLL	PLL Signal	H-tile Hard IP for Ethernet Intel FPGA
Main ATX PLL	`tx_serial_clk_gxt`	`i_tx_serial_clk[0]`
Main ATX PLL	`pll_locked`	`i_tx_pll_locked[0]` `i_tx_pll_locked[1]`
Clock Buffer	`tx_serial_clk_gxt`	`i_tx_serial_clk[1]`

Refer to the example compilation project or design example for working user logic that demonstrates one correct method to instantiate and connect the external PLLs.

Figure 7. PLL Configuration Example for 50GBASE-R2 (Synchronous Mode) IP Core Variation



To configure ATX PLL in 50GBASE-R2 variant:

- Set **VCCR_GXB and VCCT_GXB supply voltage for the Transceiver** to **1_1V**.
- Set **Primary PLL clock output buffer** to **GXT clock output buffer**.
- Turn on **Enable GXT local clock output port (tx_serial_clk_gxt)**.
- Set **GXT output clock source** to **Local ATX PLL**.
- Set **PLL output frequency** to **12890.625 MHz**. The transceiver performs dual edge clocking, using both the rising and falling edges of the input clock from the PLL. Therefore, this PLL output frequency setting supports a 25.78125 Gbps data rate through the transceiver.
- Set **PLL auto mode reference clock frequency** to the value you specified for the **PHY Reference Frequency** parameter.

When you generate an H-tile Hard IP for Ethernet Intel FPGA IP core, the software also generates the HDL code for an ATX PLL, in the simulation file *<variation_name>*/altera_xcvr_atx_pll_s10_htile_180/sim/*<variation_name>*_altera_xcvr_atx_pll_s10_htile_180_*<random_string>*.sv and the synthesis file *<variation_name>*/altera_xcvr_atx_pll_s10_htile_180/synth/*<variation_name>*_altera_xcvr_atx_pll_s10_htile_180_*<random_string>*.sv . However, the HDL code for the H-tile Hard IP for Ethernet Intel FPGA IP core does not instantiate the ATX PLL. If you choose to use the ATX PLL provided with the H-tile

Hard IP for Ethernet Intel FPGA IP core, you must instantiate and connect the instances of the ATX PLL with the H-tile Hard IP for Ethernet Intel FPGA IP core in user logic.

If you generate your own ATX PLL, you must ensure its file name differs from the PLL provided with the IP core.

Note: If your design includes multiple instances of the H-tile Hard IP for Ethernet Intel FPGA IP core, do not use the ATX PLL HDL code provided with the IP core. Instead, generate new ATX PLL IP cores to connect in your design.

Related Information

- L-Tile/ H-Tile Transceiver Native PHY Intel Stratix 10 FPGA IP Core User Guide
  Information about the correspondence between PLLs and transceiver channels in Intel Stratix 10 devices, and information about how to configure an external transceiver PLL for your own design. Refer to the sections about the GXT clock network and about using the ATX PLL for GXT channels.
- H-tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
  Information about the H-tile Hard IP for Ethernet Intel FPGA design example, which connects the appropriate number of external PLLs to the IP core PHY links.

2.4.4. Clock Requirements

The following are the clock connections requirements for:

- Normal operation for MAC+PCS or PCS Only variations
- Asynchronous clock operation for MAC+PCS variation

Related Information

Clocks on page 79

2.4.4.1. Clock Connection Requirements for Synchronous Operation

For synchronous operation, you must make the following clock connections:

- The same clock should drive the `i_clk_ref` input signal to the IP core and the reference clocks of the ATX PLLs to which it is connected. If your design cannot drive `i_clk_ref` with the same clock as the PLL reference clocks, you must ensure the two clocks have the same nominal rate.
- The output clock `o_clk_pll_div64` drives both the `i_clk_rx` and the `i_clk_tx` input clocks.
- In case of multiple instances of the IP core, if the same clock drives the `i_clk_ref` input clock of all the instances and all of their ATX PLLs, the `o_clk_pll_div64` output clock from one instance can drive all instances of `i_clk_rx` and `i_clk_tx`.

2.4.4.2. Clock Connection Requirement for Asynchronous Clock Operation

When you enable **Enable asynchronous adapter clocks** in the parameter editor, the `i_clk_rx` and `i_clk_tx` can be asynchronous from each other and `o_clk_pll_div64` clock, without adding an asynchronous FIFO or special data valid sequence. However, the clock rates must meet the minimum frequency specify in the following table.

Table 9. Minimum Clock Rates for Asynchronous Clock Operation

Ethernet Rate	Minimum Frequency for `i_clk_tx`	Minimum Frequency for `i_clk_rx`
50-Gbps	390.625 MHz	Same frequency as `o_clk_rec_div66` or 390.635 MHz + 200ppm.
100-Gbps	280.90 MHz	280.90 MHz + 200 ppm

2.4.5. Placement Settings for the H-tile Hard IP for Ethernet Intel FPGA IP Core

The Intel Quartus Prime Pro Edition software provides the options to specify design partitions and Logic Lock (Standard) Plus regions for block-based design, to control placement on the device. To achieve timing closure for your design, you might need to provide floorplan guidelines using one or both of these features.

In all cases you must take into account the location of the hard IP for Ethernet on the target H-tile(s). Each H-tile offers a single hard IP for Ethernet block. Refer to the *Ethernet Hard IP* section of the *Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide* or the figures in *Channel Placement*.

The appropriate floorplan is always design-specific and depends on your full design.

Related Information

- Intel Quartus Prime Pro Edition Handbook Volume 2: Design Implementation and Optimization
  Describes design constraints and Logic Lock regions.
- Block-Based Design Flows
- Ethernet Hard IP section of Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
  Information about the location of the Hard IP for Ethernet block on Intel Stratix 10 H-tile devices.

2.5. IP Core Testbenches

Intel provides a design example and a testbench that you can generate for the H-tile Hard IP for Ethernet Intel FPGA IP core.

To generate the testbench, in the H-tile Hard IP for Ethernet Intel FPGA 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

H-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
Information about generating and running the design example and testbench files for the H-tile Hard IP for Ethernet Intel FPGA IP core. This testbench demonstrates a basic test of the IP core. It is not intended to be a substitute for a full verification environment.

2.6. Compiling the Full Design and Programming the FPGA

You can use the **Start Compilation** command on the Processing menu in the Intel Quartus Prime Pro Edition software to compile your design. After successfully compiling your design, program the targeted Intel device with the Programmer and verify the design in hardware.

Related Information

- Block-Based Design Flows
- Programming Intel FPGA Devices
- Stratix 10 Low Latency 100G Ethernet Design Example User Guide

3. Intel FPGA IP Parameters

3.1. Parameter Editor Parameters

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

The H-tile Hard IP for Ethernet Intel FPGA parameter has two tabs, an **IP** tab and an **Example Design** tab. For information about the **Example Design** tab, refer to the *H-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide*.

Table 10. H-tile Hard IP for Ethernet Intel FPGA 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
General Options			
Ethernet Rate	• 50G • 100G	50G	Selects the IP core Ethernet data rate.
Ethernet IP layers	• MAC+PCS • PCS Only • OTN • FlexE	MAC+PCS	Selects the type of Ethernet layer in your IP core variation. • MAC+PCS: Full Ethernet MAC and PCS • PCS Only: Ethernet PHY with MII interface • OTN: Ethernet PHY suitable for an OTN Ethernet PHY (PCS without 64/66B encoding/decoding and scrambling/de-scrambling). *Note:* The H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature. For further inquiries, contact your nearest Intel sales representative or file an Intel Premier Support (IPS) case on https://www.intel.com/content/www/us/en/programmable/my-intel/mal-home.html. • FlexE: Ethernet PHY suitable for Flex Ethernet PHY (PCS without 64/66B encoding/decoding).
Ready Latency	0–3	0	Selects the readyLatency value on the TX client interface. readyLatency is an Avalon-ST 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*.

continued...

Parameter	Range	Default Setting	Parameter Description
			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.
MAC Options: Basic Tab *Note:* In PCS Only 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. MAC+PCS variations support the entire range. In PCS Only, OTN, and FlexE variations, 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. MAC+PCS variations support the entire range. In PCS Only, OTN, and FlexE variations, this parameter has no effect and remains at the default value of 1518.
Enforce maximum frame size	• True • False	False	Specifies whether the IP core is able to receive an oversized packet or truncates these packets.
Link fault generation option	• OFF • Unidirectional • Bidirectional	OFF	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. 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. Flow control is not supported for PCS, OTN or FlexE only variations. Choose **Disable Flow Control** option when using PCS, OTN or FlexE only variations.
Bytes to remove from RX frames	• None • Remove CRC bytes • Remove CRC and PAD bytes	Remove CRC bytes	Selects whether the RX MAC should remove CRC bytes, or remove CRC and PAD bytes, or do not remove anything from incoming RX frames before passing them to the RX MAC

continued...

Parameter	Range	Default Setting	Parameter Description
			Client. If the PAD bytes and CRC are not needed downstream, this option can reduce the need for downstream packet processing logic
Forward RX pause requests	• True • False	False	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	• True • False	False	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.
TX MAC source address	0–(2^{48}–1)	0x00_11_22_33_44_55	Source address with which the IP core initializes the TXMAC_SADDR registers at offsets 0x40C and 0x40D. *Note:* In the Intel Quartus Prime Pro Edition software release v17.1, the default value displays in the parameter editor in decimal notation (as 7358829205), and if you modify the value, you must specify the new value in decimal notation. *Note:* In the Intel Quartus Prime Pro Edition software release v17.1, the parameter input field appears only when you turn on the **Use source address insertion** parameter, and the parameter name does not display. In future releases the parameter name will appear.
TX VLAN detection	• True • False	True	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.
RX VLAN detection	• True • False	True	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

continued...

Parameter	Range	Default Setting	Parameter Description
			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.
Enable asynchronous adapter clocks	• True • False	False	If turned on, the IP core is allowed to use different clock source for `i_clk_rx` and `i_clk_tx` signals. Only available for Mac + PCS variation.
MAC Options: Specialized Tab *Note:* In PCS, OTN, and FlexE Only variations, these parameters have no effect.			
Enable preamble passthrough	• True • False	False	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	• True • False	False	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	• True • False	False	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. 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 does not attempt to control the minimum IPG.
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 50GBASE-R2 variations, the default number is 4. In 100GBASE-R4 variations, the default number is 20. Each increment of 1 in the value of **Additional IPG removed per AM period** increases throughput by 6ppm in 50GBASE-R2 variations or by 3ppm in 100GBASE-R4 variations. To specify larger throughput increases, use the **Average Inter-packet Gap** parameter.
PMA Options			
PHY Reference Frequency	• 644.53125 MHz • 322.265625 MHz	644.53125 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).

continued...

Parameter	Range	Default Setting	Parameter Description
AN/LT Options			
Enable AN/LT	• True • False	False	If this parameter is turned on, the IP core supports 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*. If this parameter is turned off, the IP core does not support these features, and the other parameters on this tab are not available. Auto-negotiation and link training features are available only in MAC+PCS variation.
Status clock rate	100–162 MHz	100 MHz	Sets the expected incoming `i_reconfig_clk` frequency. The input clock frequency must match the frequency you specify for this parameter. The IP core is configured with this information to ensure the IP core measures the link fail inhibit time accurately (determines the value of the Link Fail Inhibit timer (IEEE 802.3 clause 73.10.2) correctly).
Auto-Negotiation			
Enable Auto-Negotiation	• True • False	True	If this parameter is turned on, the IP core includes logic 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 include auto-negotiation logic and cannot perform auto-negotiation.
Link fail inhibit time	500–510 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.
Enable CR Technology Ability	• True • False	True	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.
Auto-Negotiation Master	• Lane 0 • Lane 1 • Lane 2 • Lane 3	Lane 0	Selects the master channel for auto-negotiation. The IP core does not provide a mechanism to change the master channel dynamically. The value you set in the parameter editor cannot be changed during operation. For 50G **Ethernet rate**, only Lane 0 and Lane 1 options are available. For 100G **Ethernet rate**, all options are available.
Pause ability–C0	• True • False	True	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*.

continued...

Parameter	Range	Default Setting	Parameter Description
Pause ability–C1	• True • False	True	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			
Enable Link Training	• True • False	True	If this parameter is turned on, the IP core includes the link training module, 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*.
Number of frames to send at end of training	• 127 • 255	127	Specifies the number of additional training frames the local link partner delivers after training is complete to ensure that the link partner can correctly detect the local receiver state.
Enable Clause 72 PRBS11 generation	• True • False	False	If turned on, the IP core includes logic to generate the legacy Clause 72 PRBS pattern, in addition to the 25G Link Training patterns specified in Clause 92 of the *IEEE Std 802.3–2015*. If turned off, the IP core generates only the 25G Link Training patterns specified in Clause 92 of the *IEEE Std 802.3–2015*.
Link Training: PMA Parameters			
VMAXRULE	0–31	30	Specifies the maximum V_{OD}. The default value, 30, represents 1200 mV. This default value is the maximum value the device should drive.
VMINRULE	0–31	6	Specifies the minimum V_{OD}. The default value, 6, represents 165 mV. This default value is the minimum value the device should drive.
VODMINRULE	0–31	14	Specifies the minimum V_{OD} for the first tap. The default value, 14, represents 440 mV.
VPOSTRULE	0–25	25	Specifies the maximum value that the internal algorithm for pre-emphasis will ever test in determining the optimum post-tap setting.
VPRERULE	0–16	16	Specifies the maximum value that the internal algorithm for pre-emphasis will ever test in determining the optimum pre-tap setting.
PREMAINVAL	0–31	30	Specifies the Preset V_{OD} value. This value is set by the Preset command of the link training protocol, defined in Clause 72.6.10.2.3.1 of *IEEE Std 802.3–2015*.
PREPOSTVAL	0–**25**	0	Specifies the preset Post-tap value.
PREPREVAL	0–16	0	Specifies the preset Pre-tap value.
INITMAINVAL	0–**31**	25	Specifies the initial V_{OD} value. This value is set by the Initialize command of the link training protocol, defined in Clause 72.6.10.2.3.2 of *IEEE Std 802.3–2015*.
INITPOSTVAL	0–**25**	13	Specifies the initial Post-tap value.
INITPREVAL	0–16	3	Specifies the initial Pre-tap value.

continued...

Parameter	Range	Default Setting	Parameter Description
Configuration, Debug and Extension Options			
Enable Native PHY Debug Master Endpoint (NPDME)	• True • False	False	When you turn on this option, the Transceiver Native PHY IP includes an embedded Native PHY Debug Master Endpoint (NPDME) that connects internally to the Avalon-MM slave interface for dynamic reconfiguration. The NPDME 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	• True • False	False	Turn on this option to enable an internal JTAG connection to the Avalon-MM Master Bridge for register reconfigurations. This connection allows the System Console to run the Ethernet Link Inspector.

Related Information

- H-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
  Information about the **Example Design** tab in the H-tile Hard IP for Ethernet Intel FPGA parameter editor.
- Avalon Interface Specifications
  Detailed information about Avalon-ST interfaces and the Avalon-ST readyLatency parameter.

3.2. RTL Parameters

The H-tile Hard IP for Ethernet Intel FPGA IP core 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.

Table 11. H-tile Hard IP for Ethernet Intel FPGA 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 `enable` (default value): 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. • Value `disable`: The IP core MAC implements standard alignment marker periods as specified in the *IEEE Standard 802.3–2015*.

continued...

Parameter	Parameter Description
	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 Offset 0x360. • `am_period[15:0]` field (bits [31:16]) of the TXMAC_EHIP_CFG register at Offset 0x40B.
Parameters Available for MAC+PCS IP Core Variations Only	
`rx_pause_daddr`	Sets the destination address 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 Offset 0x40A.

continued...

Parameter	Parameter Description
tx_pause_daddr	Sets the destination address 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 address 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 address 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.

Related Information

Advanced RTL Parameters on page 85
Additional RTL parameters are available for advanced applications.

4. Functional Description

The H-tile Hard IP for Ethernet Intel FPGA IP core 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 50-Gbps and 100-Gbps Ethernet PHY implemented in hard IP.

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 H-tile Hard IP for Ethernet Intel FPGA IP core also supports PCS Only variations. These variations provide an MII interface to the client and transmit and receive Ethernet packets through a 50-Gbps and 100-Gbps Ethernet PHY implemented in hard IP.

Other than MAC+PCS and PCS Only variations, the IP core also supports OTN or FlexE only variations. These variations enable the application to read or write 66b blocks to the RX and TX PCS, bypassing the MAC through a 50-Gbps and 100-Gbps Ethernet PHY implemented in hard IP. There is no encoder/decoder and scrambler/descrambler in OTN variation. There is no encoder/decoder in FlexE variation.

Note: The H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature. For further inquiries, contact your nearest Intel sales representative or file an Intel Premier Support (IPS) case on https://www.intel.com/content/www/us/en/programmable/my-intel/mal-home.html.

4.1. High Level System Overview

Figure 8. H-tile Hard IP for Ethernet Intel FPGA IP Core MAC + PCS Variant

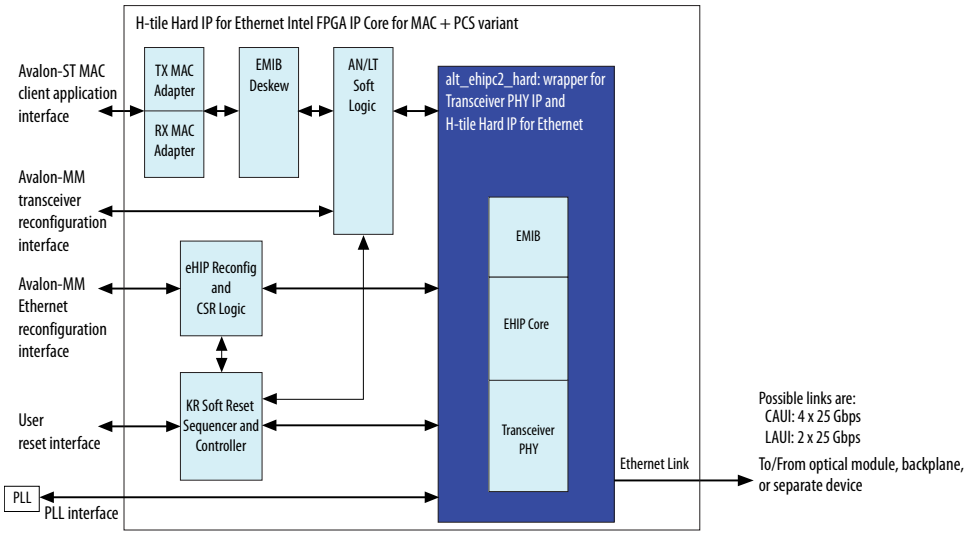


Figure 9. H-tile Hard IP for Ethernet Intel FPGA IP Core PCS Variant

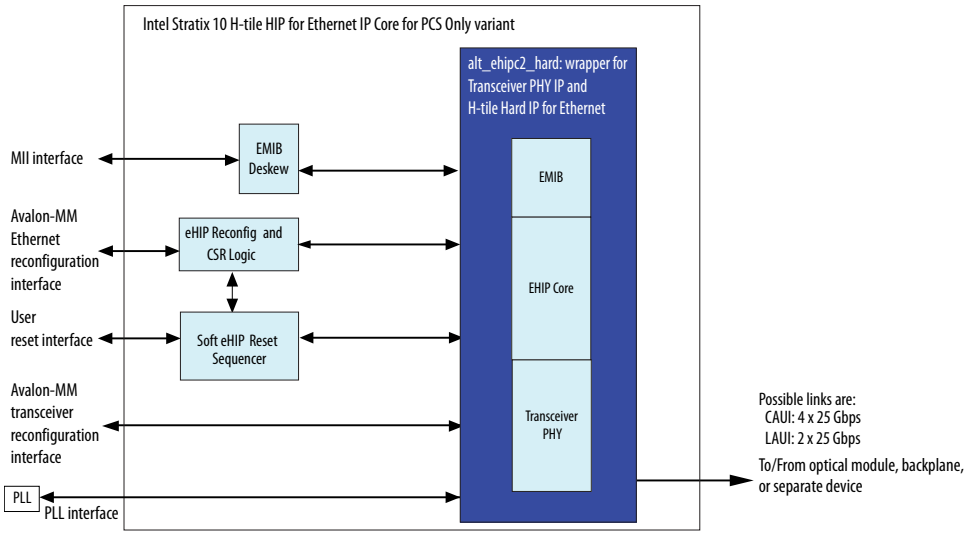
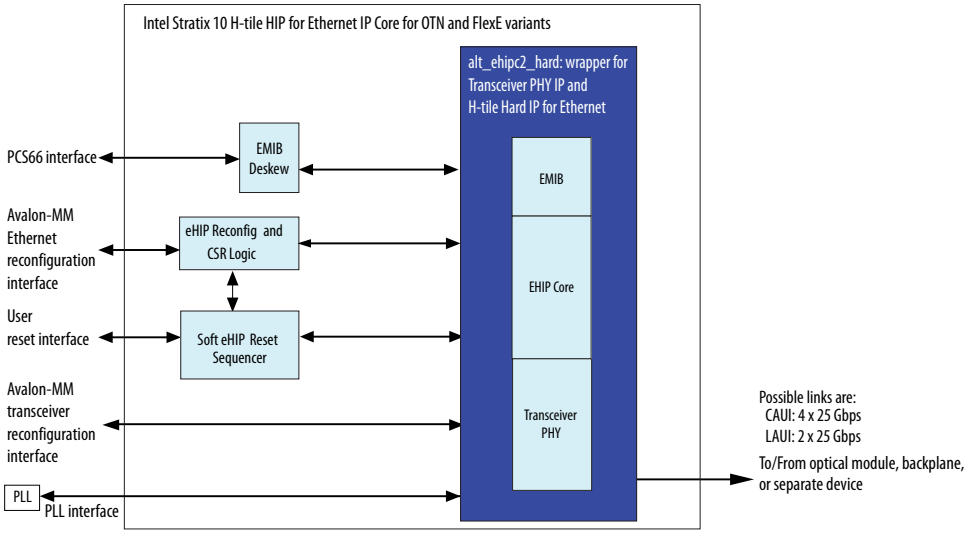


Figure 10. H-tile Hard IP for Ethernet Intel FPGA IP Core OTN and FlexE Variants



4.2. H-tile Hard IP for Ethernet Intel FPGA MAC Interface

4.2.1. H-tile Hard IP for Ethernet Intel FPGA IP Core TX Datapath

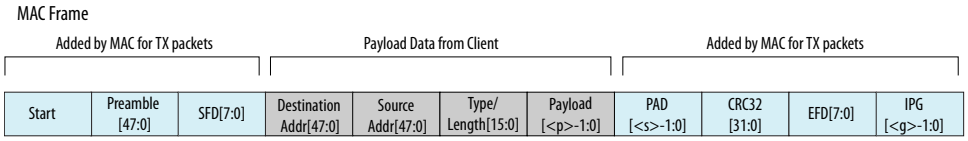
The TX MAC module receives the client payload data with the destination and source addresses and then adds, appends, or updates various header fields in accordance with the configuration specified. The MAC does not modify the destination address, the source address, or the payload received from the client. 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 drive `i_tx_skip_crc` port to 0, the MAC or turn on flow control calculates the CRC over the entire frame. If padding is added, it is also included in the CRC calculation. If you drive `i_tx_skip_crc` port to 1, the client must provide the CRC bytes and must provide frames that have a minimum size of 64 bytes and therefore do not require padding. The TX MAC module always inserts IDLE bytes to maintain an average IPG.

The H-tile Hard IP for Ethernet Intel FPGA IP core does not process incoming frames of less than nine bytes correctly. You must ensure such frames do not reach the TX client interface.

Figure 11. Typical Client Frame at the Transmit Interface

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:

4.2.1.1. 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. If you set **Link fault generation option**, this MAC module also incorporates the functions of the reconciliation sublayer.

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

If the preamble pass-through feature is turned on, the client provides the eight-byte preamble (including Start byte) on the data bus. The client is responsible for providing the correct Start byte.

4.2.1.2. Length/Type Field Processing

This two-byte header 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.

4.2.1.3. Frame Padding

When the length of client frame is less than 64 bytes (meaning the payload is less than 46 bytes) and greater than eight bytes, the TX MAC module inserts pad bytes (0x00) 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 H-tile Hard IP for Ethernet Intel FPGA IP core does not process incoming (egress) frames of less than nine bytes correctly. You must ensure such frames do not reach the TX client interface.

Note: TX MAC padding is only available for **MAC + PCS** IP core variation.

4.2.1.4. 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.

4.2.1.5. Inter-Packet Gap Generation and Insertion

If you set **Average Inter-packet Gap** to **12** in the H-tile Hard IP for Ethernet Intel FPGA 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, without inserting any extra idle Control words to maintain IPG at a specified average. In this case 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.

The TX MAC also supports inter-packet gap reduction in increments of 3 ppm, using the **Additional IPG removed per AM period** parameter or dynamically set through the `ipg_col_rem` register.

Note: The inter-packet gap generation and insertion feature is only available for **MAC + PCS** IP core variation.

4.2.2. H-tile Hard IP for Ethernet Intel FPGA IP Core 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 **Remove pads** and **Keep RX CRC** parameters.

Figure 12. Flow of Frame Through the MAC RX Without Preamble Pass-Through

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).

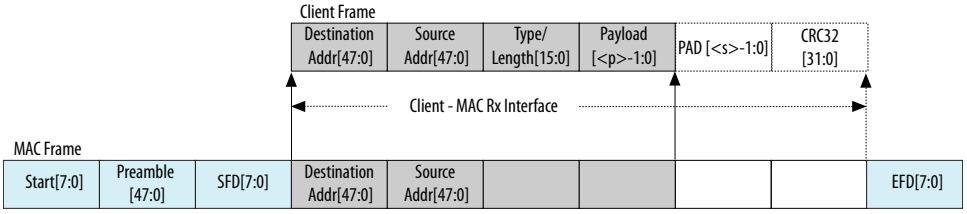
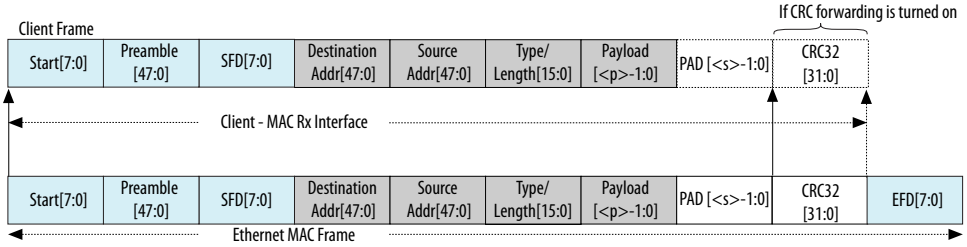


Figure 13. Flow of Frame Through the MAC RX With Preamble Pass-Through Turned On

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:

4.2.2.1. H-tile Hard IP for Ethernet Intel FPGA IP Core RX Filtering

The H-tile Hard IP for Ethernet Intel FPGA IP core processes all incoming valid frames. However, the IP core does not forward pause frames to the Avalon-ST RX client interface by default.

If you set the **Forward RX pause requests** parameter, the IP core forwards pause frames to the Avalon-ST RX client interface.

4.2.2.2. H-tile Hard IP for Ethernet Intel FPGA IP Core 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. The MAC RX looks for the Start, six preamble bytes and SFD, depending on the strict SFD checking settings of the IP core.

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 H-tile Hard IP for Ethernet Intel FPGA parameter editor, the MAC RX does not remove the eight-byte preamble sequence.

4.2.2.3. IP Core Strict SFD Checking

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

If you turn on **Enable strict preamble check** in the H-tile Hard IP for Ethernet Intel FPGA 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).

Table 12. 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	

4.2.2.4. 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.

4.2.2.5. 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. If a control byte other than Error or Terminate is found inside a frame, the RX MAC asserts `o_rx_error[0]` when the frame ends to indicate that it was a malformed packet.

4.2.2.6. 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 pad and CRC 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.

Note: Forwarding CRC in RX frames feature is only available for **MAC + PCS** IP core variation.

4.2.2.7. 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.

4.2.2.8. 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.

Note: The inter-packet gap generation and insertion feature is only available for **MAC + PCS** IP core variation.

4.2.3. 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 H-tile Hard IP for Ethernet Intel FPGA IP core 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.

Note: Flow control feature is only available in **MAC + PCS** IP core variation.

4.2.3.1. Conditions Triggering XOFF Frame Transmission

The H-tile Hard IP for Ethernet Intel FPGA IP core 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.

4.2.3.2. 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.

4.2.4. 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, there is no option to stop traffic automatically when a PFC XOFF frame arrives.

Table 13. 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 H-tile Hard IP for Ethernet Intel FPGA parameter editor.

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.

4.2.5. Pause Control Frame Filtering

The H-tile Hard IP for Ethernet Intel FPGA IP core 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.

4.2.6. Link Fault Signaling

If you choose **Unidirectional** or **Bidirectional** in the **Link fault generation option** parameter, 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.

4.2.7. Statistics Counters Interface

The H-tile Hard IP for Ethernet Intel FPGA provides the following methods to read the statistic counters:

- Assert `i_stats_snapshot` signal to freeze and capture a snapshot of the statistic counter values
- Request through statistic shadow registers
- Read statistic counters through Avalon-MM interface

4.2.7.1. Statistic Counters Rollover Limitations

The statistic counters that filter the packets based on frame length may increment incorrectly when the MAC transmits or receives frames larger than 2^{16} – 1 bytes with **Enforce maximum frame size** parameter is disabled.

To prevent incorrect statistic counts, limit the packet length for transmission to less or equal to 2^{16} – 1 bytes and enable **Enforce maximum frame size** parameter. The **Enforce maximum frame size** feature truncates received frames that exceed 2^{16} – 1 bytes.

4.2.8. 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 14. Byte Order on the Client Interface Lanes Without Preamble Pass-Through

The figure describes the byte order on the Avalon-ST 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 figures show that in this example, 0xAC is driven on DA5 (`DA[47:40]`) and 0x80 is driven on DA0 (`DA[7:0]`).

Figure 15. 50-Gbps Octet Transmission on the Avalon-ST Interface Without Preamble Pass-Through

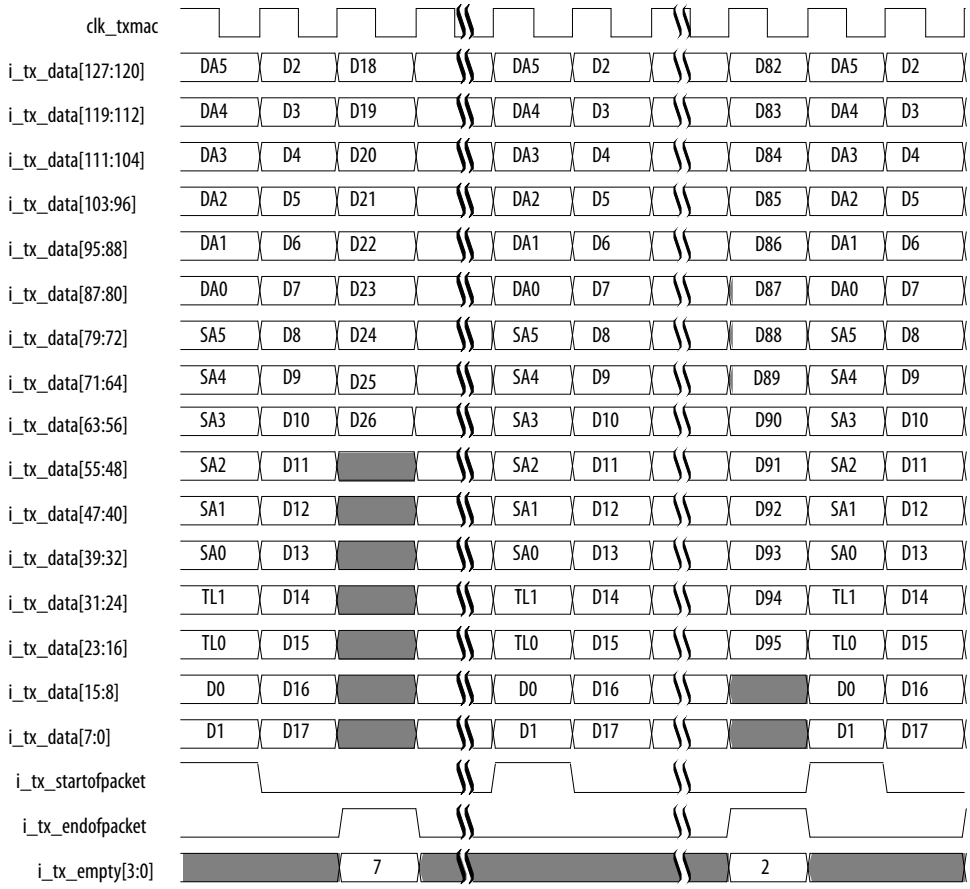


Figure 16. 100-Gbps Octet Transmission on the Avalon-ST Signals Without Preamble Pass-Through

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

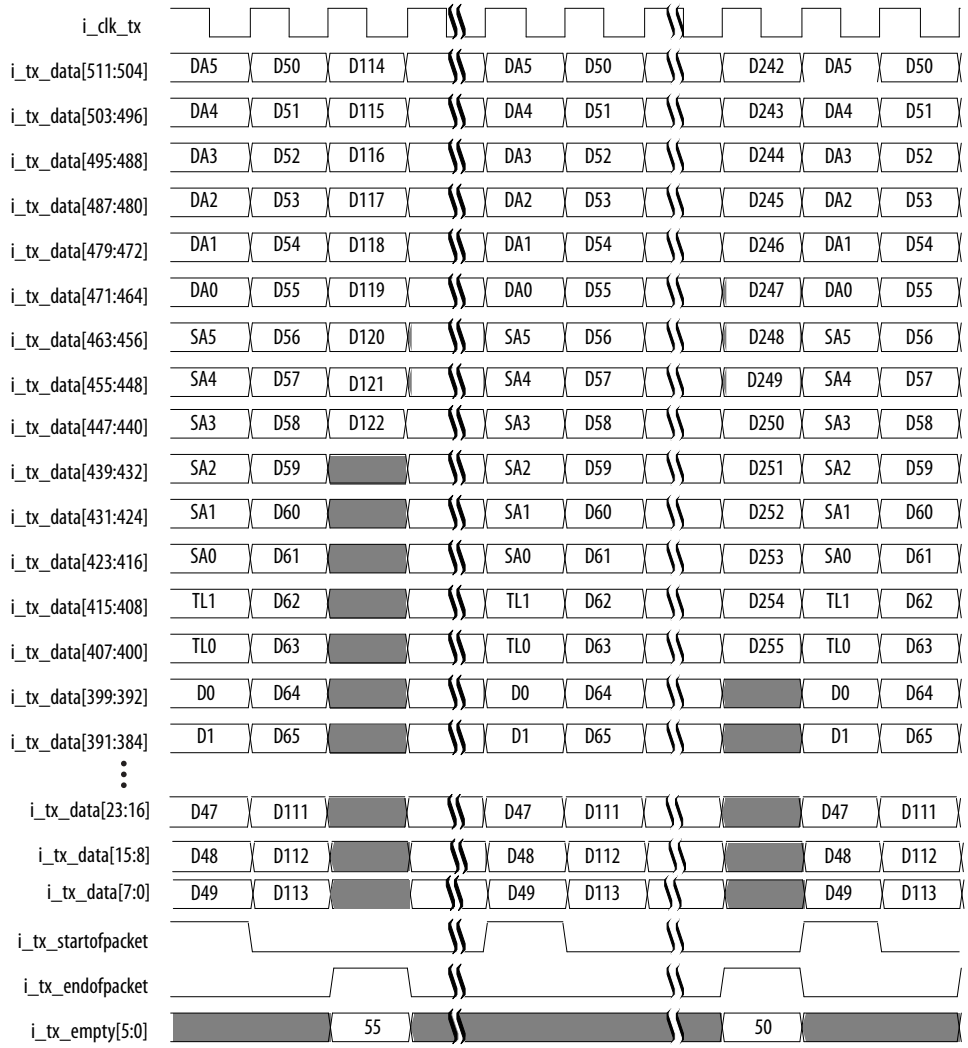


Figure 17. Byte Order on the Avalon-ST Interface Lanes With Preamble Pass-Through

The figure describes the byte order on the Avalon-ST 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 18. 50-Gbps Octet Transmission on the Avalon-ST 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[71:64]` and the START byte on `i_tx_data[127:120]`.

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

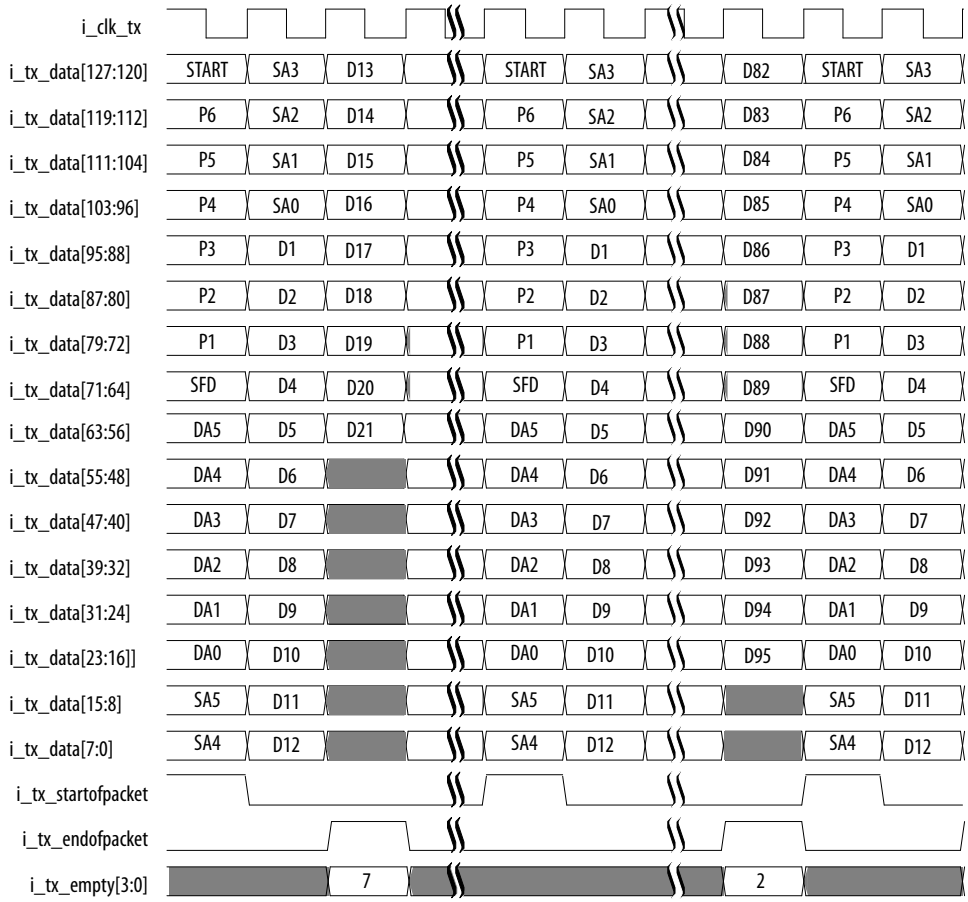
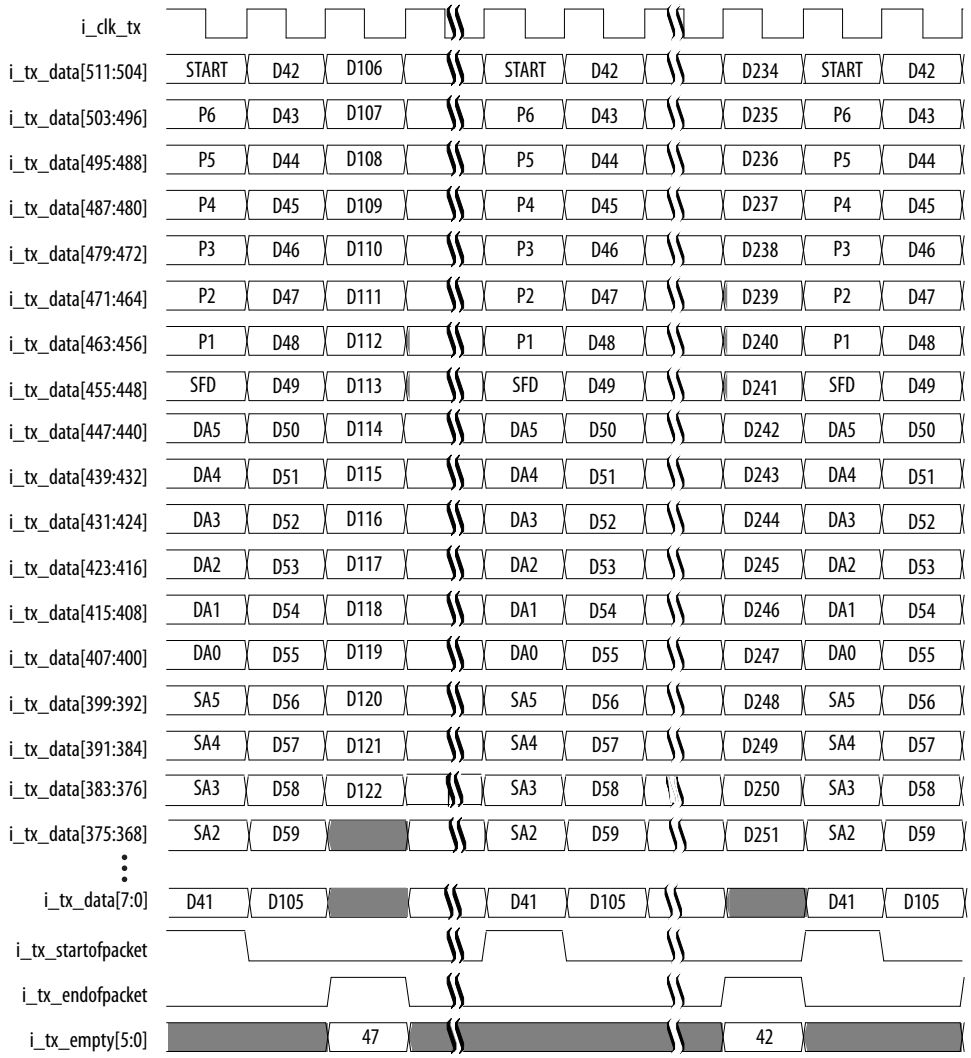


Figure 19. 100-Gbps Octet Transmission on the Avalon-ST Signals With Preamble Pass-Through

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.

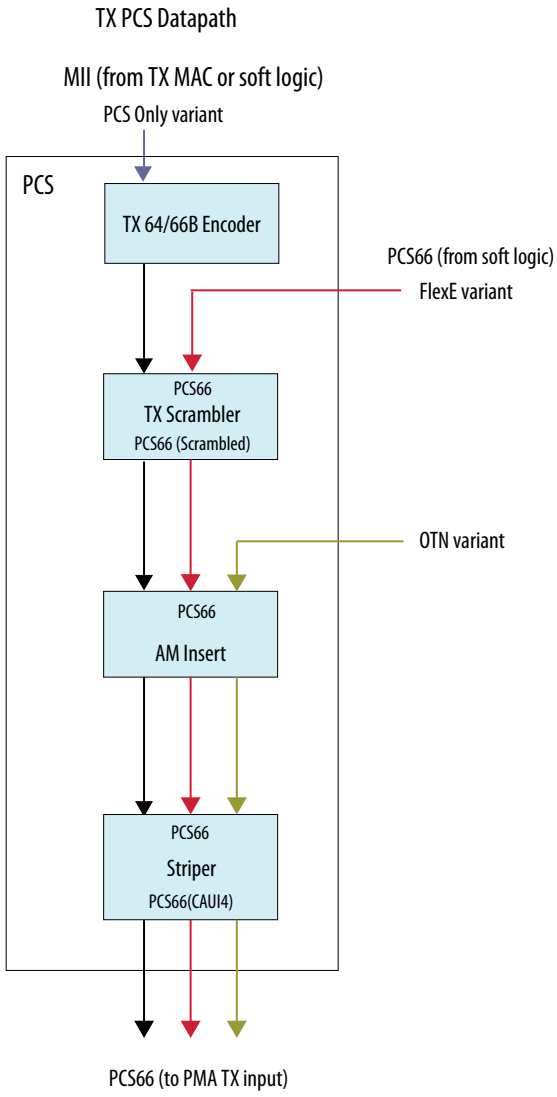


4.3. H-tile Hard IP for Ethernet Intel FPGA PCS Only/PCS66 Interface

4.3.1. TX PCS and RX PCS Datapath

Each H-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.

Figure 20. TX PCS Datapath

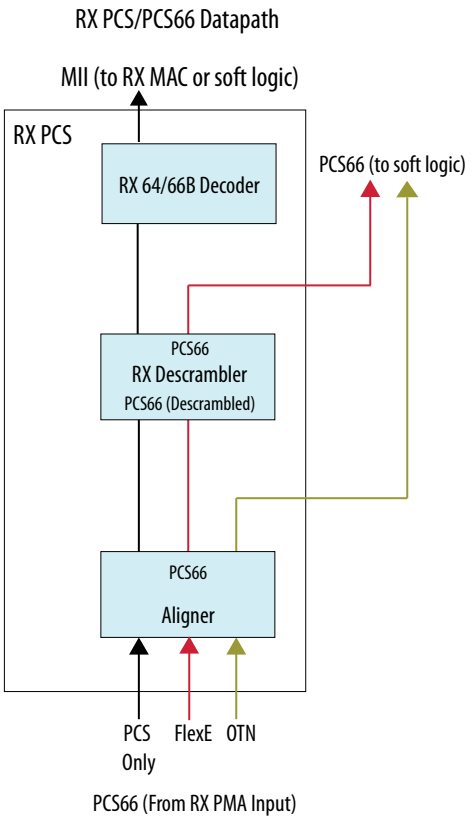


In FlexE mode, the TX Encoder in the PCS is bypassed; in OTN mode, both the TX Encoder and Scrambler are bypassed.

- TX PCS encoder: Enables the data to written in encoded form from the PCS66 interface.
- TX PCS scrambler: Enables the data to be scrambled. Channels will not lock correctly if the data is not scrambled.

  Note: In OTN mode, scrambler is bypassed because the input data is expected to be scrambled.
- Alignment insertion: The TX PCS interface inserts alignment markers
- Striper: Enables logically sequential data to be segmented to increase data throughput.

Figure 21. RX PCS Datapath



In FlexE mode, RX data is aligned and descrambled, but not decoded; in OTN mode, it is only aligned.

In both modes, data are not decoded as the incoming data is not encoded.

- Aligner: Enables the alignment of incoming data.
- RX PCS descrambler: Enables the incoming scrambled data to be descrambled.

  Note: In OTN mode, descrambler is bypassed.
- RX PCS decoder: Enables the incoming encoded data to written in decoded form from the PCS66 interface.

4.4. Auto-Negotiation and Link Training

The H-tile Hard IP for Ethernet Intel FPGA IP core 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 **Enable Auto-Negotiation** parameters to configure support for auto-negotiation.
- 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** and **Enable Link Training** parameters to configure support for link training.

The H-tile Hard IP for Ethernet Intel FPGA IP core includes separate link training modules for each of the two or four Ethernet lanes, and a single auto-negotiation module. You specify the master lane for performing auto-negotiation in the parameter editor.

Note: The auto-negotiation and link training features are only available for **MAC + PCS** IP core variation.

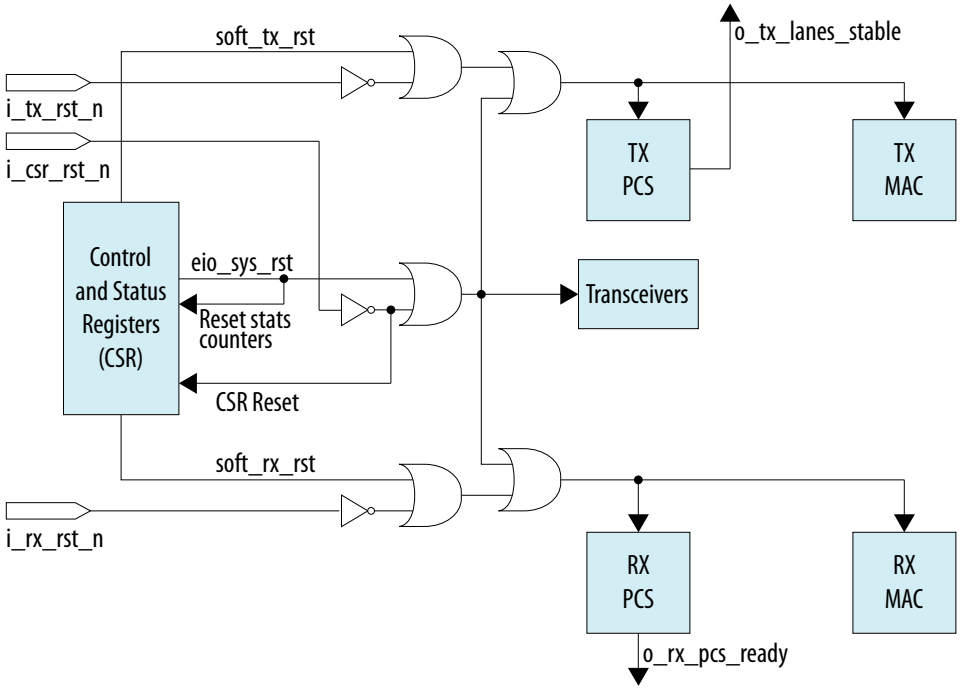
5. Reset

Ethernet registers control three distinct soft resets. These soft resets are not self-clearing. Software clears them by writing to the appropriate register. In addition, the IP core has three hard reset signals.

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

Figure 22. Conceptual Overview of General IP Core Reset Logic

The three hard resets are top-level ports. The three soft resets are internal signals which are outputs of the `PHY_CONFIG` register. Software writes the appropriate bit of the `PHY_CONFIG` register to assert a soft reset.



The general reset signals reset the following functions:

- `soft_tx_rst`, `i_tx_rst_n`: Resets the TX PCS and TX MAC. This reset leads to deassertion of the `o_tx_lanes_stable` output signal.
- `soft_rx_rst`, `i_rx_rst_n`: Resets the RX PCS and RX MAC. This reset leads to deassertion of the `o_rx_pcs_ready` output signal.
- `eio_sys_rst`, `i_csr_rst_n`: Resets the IP core. Resets the TX and RX MACs, Ethernet reconfiguration registers, PCS, and transceivers. This reset leads to deassertion of the `o_tx_lanes_stable` and `o_rx_pcs_ready` output signals. Use this signal to reset the IP core whenever the transceiver is recalibrated.

Table 14. Reset Signal Functions

Reset Signal	Block							
	MAC TX Datapath	PCS TX Datapath	MAC RX Datapath	PCS RX Datapath	PHY	CSRs (MAC/ PHY)	TX Statistics	RX Statistics
`i_csr_rst_n,eio_sys_rst`	√	√	√	√	√	√	√	√
`i_tx_rst_n,soft_tx_rst`	√	√	X	X	X	X	X	X
`i_rx_rst_n,soft_rx_rst`	X	X	√	√	X	X	X	X
`soft_sys_rst`	√	√	√	√	√	√(2)	√	√
`soft_clear_tx_stats`	X	X	X	X	X	X	√	X
`soft_clear_rx_stats`	X	X	X	X	X	X	X	√

In addition, the synchronous `i_reconfig_reset` signal resets the IP core transceiver reconfiguration interface and the Ethernet reconfiguration interface. Associated clock is the `i_reconfig_clk`, which clocks the two interfaces.

System Considerations

You should perform a system reset before beginning IP core operation by asserting and deasserting `i_csr_rst_n` and `i_reconfig_reset` signals together. To assert both signals, set `i_csr_rst_n` signal to 0 and `i_reconfig_reset` signal to 1. To deassert both signals, set `i_csr_rst_n` to 1 and `i_reconfig_reset` to 0. 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 are corrupted.

(2) Resets only the CSRs implemented inside the IP core into the original SOF values.

If the ATX PLL loses lock, the IP core forces a transmit side and a receive side reset. To ensure the IP core also resets the Hard IP for Ethernet, you must assert the `i_csr_rst_n` signal after the ATX PLL loses lock.

If the IP core suffers loss of signal on the serial links, it asserts the receive reset.

The following diagrams show the reset sequences for TX and RX datapaths when you assert `i_tx_rst_n` and `i_rx_rst_n` reset signals.

Figure 23. TX Datapath Reset Sequence

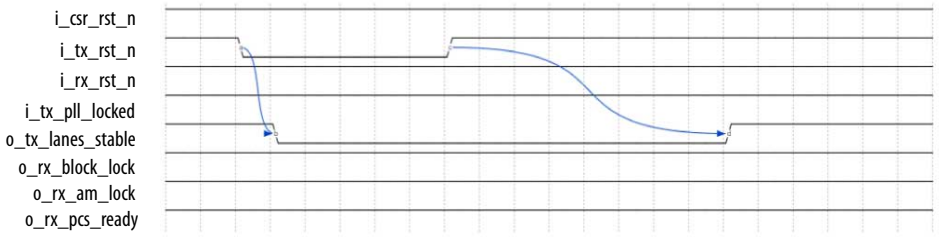
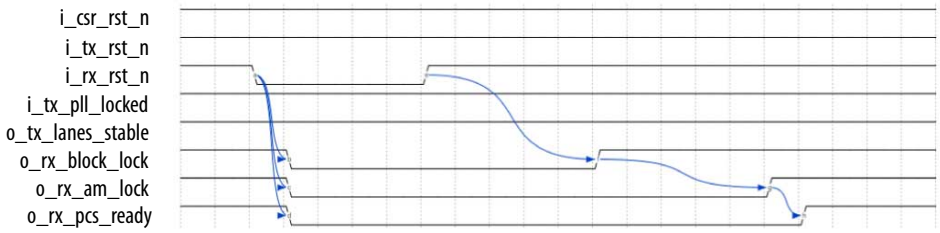


Figure 24. RX Datapath Reset Sequence



Related Information

Reset Signals on page 78

6. Interfaces and Signal Descriptions

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

Figure 25. H-tile Hard IP for Ethernet Intel FPGA 100GBASE-R4 Signals and Interfaces

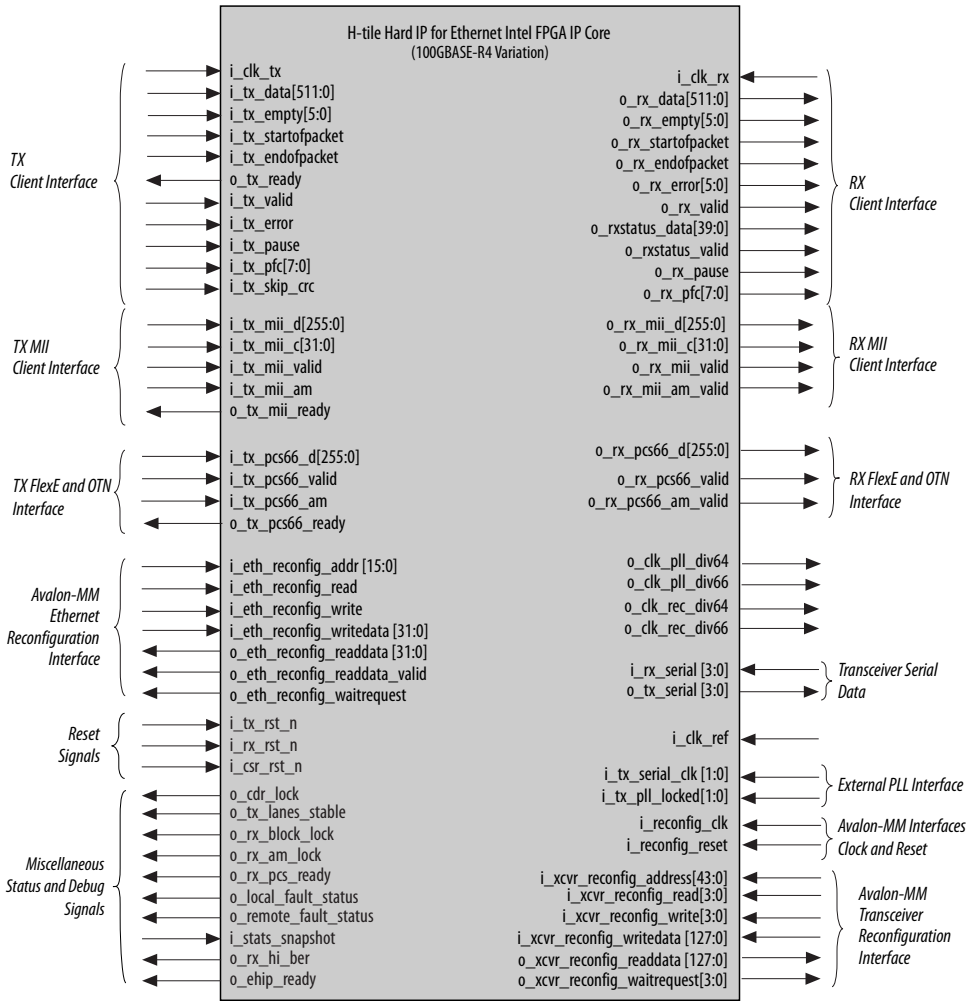
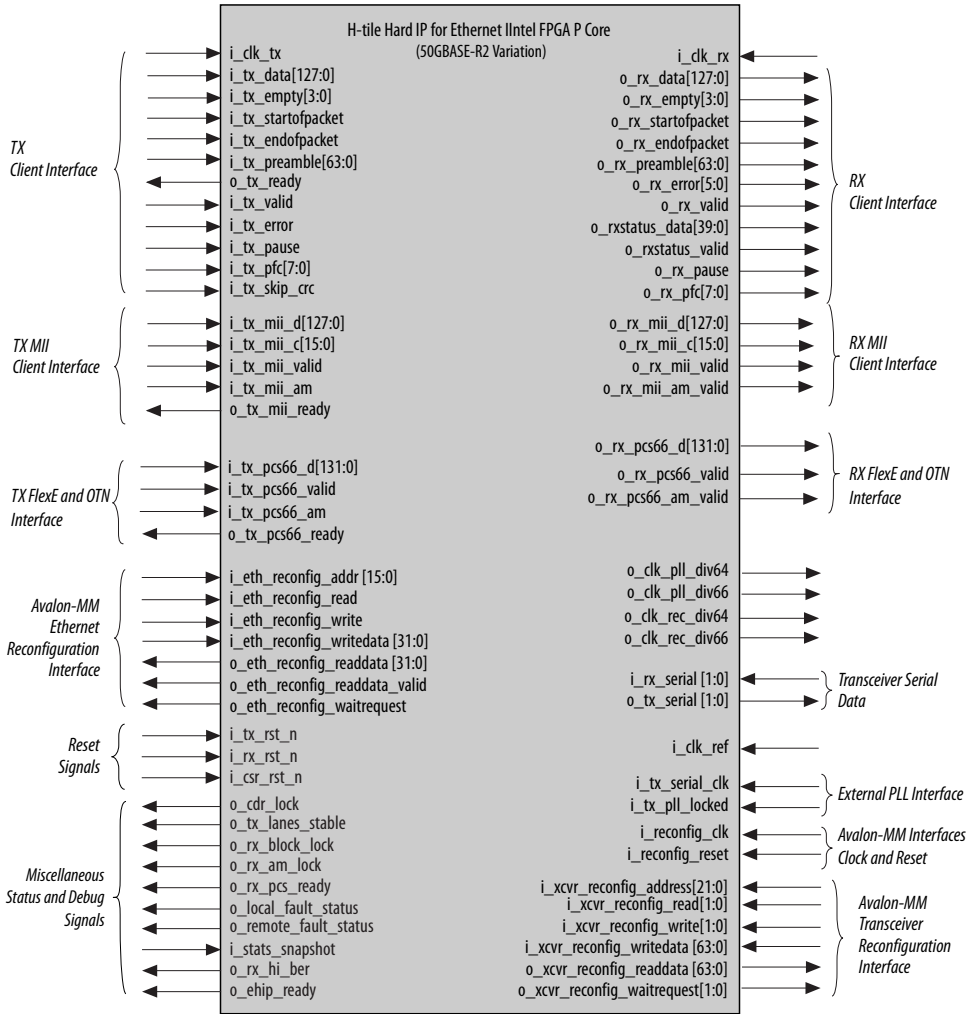


Figure 26. H-tile Hard IP for Ethernet Intel FPGA 50GBASE-R2 Signals and Interfaces



6.1. TX MAC Interface to User Logic

The H-tile Hard IP for Ethernet Intel FPGA IP core TX client interface in MAC+PCS variations employs the Avalon-ST protocol. 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 `readyLatency` 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 15. Signals of the Avalon-ST TX Client Interface

All interface signals are clocked by the `i_clk_tx` clock. The value you specify for **Enter Ready Latency** in the H-tile Hard IP for Ethernet Intel FPGA parameter editor is the Avalon-ST readyLatency value on this interface.

Signal Name	Description
`i_clk_tx`	The TX clock for the IP core is `i_clk_tx`. The frequency of this clock is 402.832 MHz.
`i_tx_data[127:0]` (in 50GBASE-R2 variations) `i_tx_data[511:0]` (in 100GBASE-R4 variations)	TX data. If the preamble pass-through feature is enabled, data in 100GBASE-R4 variations begins with the preamble. The H-tile Hard IP for Ethernet Intel FPGA IP core 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 error packets, by asserting the `i_tx_error` signal in the end-of-packet clock cycle. 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 `i_tx_startofpacket` when you are assured the packet data to send on `i_tx_data` is available or will be available on time. `i_tx_data[0]` is LSB.
`i_tx_valid`	When asserted `i_tx_data` is valid. This signal must be continuously asserted between the assertions of `i_tx_startofpacket` and `i_tx_endofpacket` for the same packet.
`i_tx_empty[3:0]` (in 50GBASE-R2 variations) `i_tx_empty[5:0]` (in 100GBASE-R4 variations)	Indicates the number of empty bytes on `i_tx_data` when `i_tx_endofpacket` is asserted.
`i_tx_startofpacket`	When asserted, indicates that `i_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 `i_tx_startofpacket` is asserted, the MSB of `i_tx_data` drives the start of packet.
`i_tx_endofpacket`	When asserted, indicates that `i_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, `i_tx_startofpacket` and `i_tx_endofpacket` are asserted on the same clock cycle.
`o_tx_ready`	When asserted, indicates that the MAC can accept the data `readyLatency` clock cycles after the current cycle. The IP core asserts the `o_tx_ready` signal on clock cycle <*n*> to indicate that clock cycle <*n* + readyLatency> is a ready cycle. The client may only transfer data during ready cycles. If the IP core deasserts `o_tx_ready` during a packet transfer on the TX MAC client interface, the client must stall the data on `i_tx_data`. The `o_tx_ready` signal indicates the MAC is ready to receive data in normal operational mode. However, the `o_tx_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_tx_preamble[63:0]`	User preamble data. This signal is available in 50GBASE-R2 variations when you turn on **Enable preamble passthrough** in the IP core parameter editor. 100GBASE-R4 variations accept the preamble on `i_tx_data` and do not provide the `i_tx_preamble` signal. User logic drives the custom preamble data when `i_tx_startofpacket` is asserted.

continued...

Signal Name	Description
i_tx_error	When asserted in an EOP cycle (while i_tx_endofpacket 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_tx_pause	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. This signal is functional only if standard Ethernet flow control is enabled. *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_tx_pfc[7:0]	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.. 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_tx_skip_crc	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 i_tx_data 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 i_tx_startofpacket through the cycle in which it asserts i_tx_endofpacket, inclusive).

Below examples show the impact on transferred data due to the readyLatency.

Figure 27. TX Avalon-ST MAC Client Interface for H-tile Hard IP for Ethernet Intel FPGA IP Core with `readyLatency` = 1

In this example, data rate is 100-Gbps and `readyLatency` is 1.

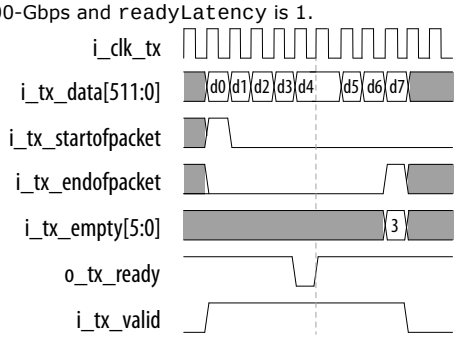
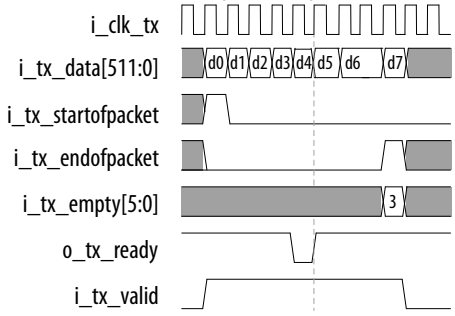


Figure 28. TX Avalon-ST MAC Client Interface for H-tile Hard IP for Ethernet Intel FPGA IP Core with `readyLatency` = 3

This example shows the 100-Gbps data rate with `readyLatency` set to 3.



Related Information

Avalon Interface Specifications
For more information about the Avalon-ST interface.

6.2. RX MAC Interface to User Logic

The H-tile Hard IP for Ethernet Intel FPGA IP core RX client interface in MAC+PCS variations employs the Avalon-ST protocol. 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 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 16. Signals of the Avalon-ST RX Client Interface

All interface signals are clocked by the i_clk_rx clock.

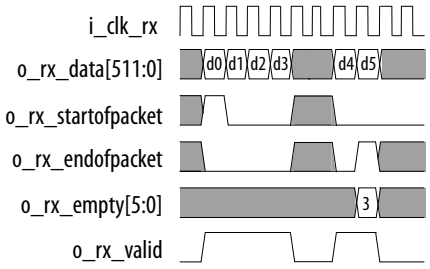
Name	Description
i_clk_rx	The RX clock for the IP core is i_clk_rx. The frequency of this clock is 402.832 MHz.
o_rx_data[127:0] (in 50GBASE-R2 variations) o_rx_data[511:0] (in 100GBASE-R4 variations)	RX data. The highest order bit is the MSB and bit 0 is the LSB. Bytes are read in the usual left to right order. The IP core reverses the byte order to meet the requirements of the Ethernet standard.
o_rx_valid	When asserted, indicates that RX data is valid. Only valid between the o_rx_startofpacket and o_rx_endofpacket signals. This signal might be deasserted between the assertion of o_rx_startofpacket and o_rx_endofpacket.
o_rx_empty[3:0] (in 50GBASE-R2 variations) o_rx_empty[5:0] (in 100GBASE-R4 variations)	Indicates the number of empty bytes on o_rx_data when o_rx_endofpacket is asserted, starting from the least significant byte (LSB).
o_rx_startofpacket	When asserted, indicates that o_rx_data 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 o_rx_startofpacket is asserted, the MSB of o_rx_data drives the start of packet.
o_rx_endofpacket	When asserted, indicates that o_rx_data 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, o_rx_startofpacket and o_rx_endofpacket might be asserted in the same clock cycle.
o_rx_preamble[63:0]	RX frame preamble value. This signal is available in 50GBASE-R2 variations when you turn on **Enable preamble passthrough** in the IP core parameter editor. 100GBASE-R4 variations send the preamble on o_rx_data and do not provide the o_rx_preamble signal. The IP core drives the custom preamble data when o_rx_startofpacket is asserted.
o_rx_error[5:0]	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 50GBASE-R2 IP core identifies a malformed packet when it receives a control character that is not a terminate character nor an Eror character, while receiving the packet. The 100GBASE-R4 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.

continued...

Name	Description
	• 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 1536 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_rxstatus_valid`	When asserted, indicates that `o_rxstatus_data` is driving valid data.
`o_rxstatus_data[39:0]`	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 • Bit[33]: When asserted, indicates a VLAN frame • Bit[32]: When asserted, indicates a stacked VLAN frame • Bits[31:0]: Reserved
`o_rx_pause`	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 100GBASE-R4 variations, and by one quarter in every valid clock cycle in 50GBASE-R2 variations.
`o_rx_pfc[7:0]`	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 100GBASE-R4 variations, and by one quarter in every valid clock cycle in 50GBASE-R2 variations. In summary, the width of the pulse indicates the length of the requested pause in traffic for the queue.

Figure 29. RX Avalon-ST MAC Client Interface for H-tile Hard IP for Ethernet Intel FPGA Intel FPGA IP Core

Shows typical traffic for the RX client interface of the IP core.



Related Information

Avalon Interface Specifications
For more information about the Avalon-ST interface.

6.3. TX PCS Interface to User Logic

The H-tile Hard IP for Ethernet Intel FPGA IP core 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 17. Signals of the MII TX Client Interface

All interface signals are clocked by the `i_clk_tx` clock.

Signal Name	Description
`i_clk_tx`	The TX clock for the IP core is `i_clk_tx`. The frequency of this clock is 402.832 MHz.
`i_tx_mii_d[127:0]` (in 50GBASE-R2 variations) `i_tx_mii_d[255:0]` (in 100GBASE-R4 variations)	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 `i_tx_mii_valid` has the value of 0 or `i_tx_mii_am` has the value of 1, and for one additional clock cycle, you must hold the value of `i_tx_mii_d` stable. We refer to this behavior as freezing the signal value.
`i_tx_mii_c[15:0]` (in 50GBASE-R2 variations) `i_tx_mii_c[31:0]` (in 100GBASE-R4 variations)	TX MII control bits. Each bit corresponds to a byte of `i_tx_mii_d`. `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. When `i_tx_mii_valid` has the value of 0 or `i_tx_mii_am` has the value of 1, you must freeze the value of `i_tx_mii_c`.
`i_tx_mii_valid`	Indicates that `i_tx_mii_d` is valid. You must assert this signal a fixed number of clock cycles after the IP core raises `o_tx_mii_ready`, and must deassert this signal the same number of clock cycles after the IP core deasserts `o_tx_mii_ready`. 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 `i_tx_mii_d` and `i_tx_mii_c` stable.
`o_tx_mii_ready`	Indicates the PCS is ready to receive new data.
`i_tx_mii_am`	Alignment marker insertion bit. In 100GBASE-R4 variations of the IP core, you must hold this signal asserted for 5 consecutive clock cycles, counting only valid cycles, to drive the insertion of an alignment marker on the Ethernet link. In 50GBASE-R2 variations, you must hold this signal asserted for 2 consecutive clock cycles, counting only the valid cycles, to drive the insertion of an alignment marker. A valid cycle is one in which `i_tx_mii_valid` has the value of 1.

continued...

Signal Name	Description
	The number of valid clock cycles from deassertion of `i_tx_mii_am` (alignment marker insertion bit signal) to reassertion of `i_tx_mii_am` is the `am_period`. • For normal operation of the Ethernet link, you must ensure that the value of `am_period` is 81915 clock cycles in 100GBASE-R4 variations, and 32766 clock cycles in 50GBASE-R2 variations. • In simulation you can reduce this value to 315 in 100GBASE-R4 variations and to 510 in 50GBASE-R2 variations. This change decreases the simulation time to RX PCS alignment. You can set the IP core to expect this interval by setting the `sim_mode` RTL parameter to `Enable`. *Note:* The value for the MAC+PCS variations is different, to ensure the value on the internal signal has the appropriate `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 *H-tile Hard IP for Ethernet Intel FPGA 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 `i_tx_mii_d` and `i_tx_mii_c`.

Figure 30. TX MII Client Interface for H-tile Hard IP for Ethernet Intel FPGA IP Core

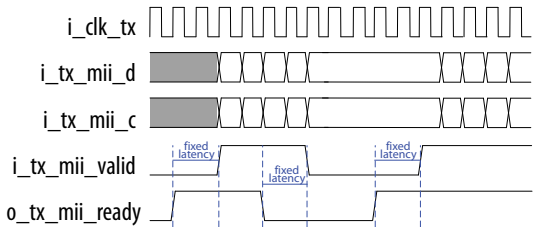
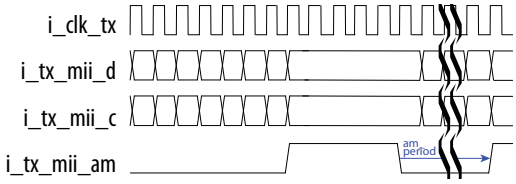


Figure 31. Alignment Marker Insertion on the TX MII Client Interface



Related Information

H-Tile Hard IP for Ethernet Intel Stratix 10 FPGA IP Design Example User Guide
The H-tile Hard IP for Ethernet Intel FPGA simulation design example sets `am_period` to 315 in 100GBASE-R4 variations and to 510 in 50GBASE-R2 variations, using the `sim_mode` RTL parameter

6.4. RX PCS Interface to User Logic

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

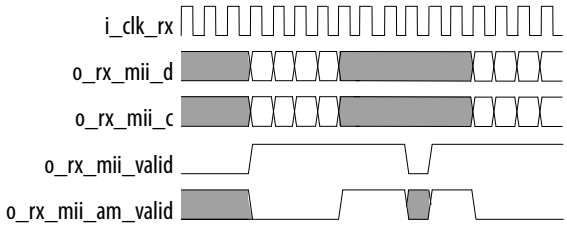
The RX PCS acts as a source and the client acts as a sink in the receive direction.

Table 18. Signals of the MII RX Client Interface

All interface signals are clocked by the `i_clk_rx` clock.

Signal Name	Description
`i_clk_rx`	The RX clock for the IP core is `i_clk_rx`. The frequency of this clock is 402.83203125 MHz.
`o_rx_mii_d[127:0]` (in 50GBASE-R2 variations) `o_rx_mii_d[255:0]` (in 100GBASE-R4 variations)	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 `o_rx_mii_valid` has the value of 0 or `o_rx_mii_am_valid` has the value of 1, the value on `o_rx_mii_d` is invalid.
`o_rx_mii_c[15:0]` (in 50GBASE-R2 variations) `o_rx_mii_c[31:0]` (in 100GBASE-R4 variations)	RX MII control bits. Each bit corresponds to a byte of `o_rx_mii_d`. `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 `o_rx_mii_valid` has the value of 0 or `o_rx_mii_am_valid` has the value of 1, the value on `o_rx_mii_c` is invalid.
`o_rx_mii_valid`	Indicates that `o_rx_mii_d`, `o_rx_mii_c`, and `o_rx_mii_am_valid` are valid.
`o_rx_mii_am_valid`	Indicates the IP core received a valid alignment marker on the Ethernet link. When `o_rx_mii_valid` has the value of 0, the value on `o_rx_mii_am_valid` is invalid. The value of `o_rx_mii_valid` may fall while the IP core is asserting `o_rx_mii_am_valid`.

Figure 32. RX MII Client Interface for H-tile Hard IP for Ethernet Intel FPGA IP Core



6.5. FlexE and OTN Mode TX Interface

The H-tile Hard IP for Ethernet Intel FPGA IP core 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 H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature. For further inquiries, contact your nearest Intel sales representative or file an Intel Premier Support (IPS) case on https://www.intel.com/content/www/us/en/programmable/my-intel/mal-home.html.

Table 19. Signals of the PCS66 TX Interface

Signal Name	Description
`i_tx_pcs66_d [127:0] (in 50GBASE-R2 variations)` `i_tx_pcs66_d[255:0] (in 100GBASE-R4 variations)`	TX PCS 66b data for 2 blocks in 50GBASE-R2 variations and 4 blocks in 100GBASE-R4 variations. • In FlexE mode, the data presented is scrambled and bit-interleaved for 100GBASE-R4 transmission. • In OTN mode, the data is only bit-interleaved and unscrambled.
`i_tx_pcs66_valid`	When asserted, indicates that the TX PCS 66b data is valid. Must be asserted when the TX PCS 66b ready signal is asserted.
`o_tx_pcs66_ready`	TX PCS 66b ready signal. When asserted, indicates the PCS is ready to receive new data.
`i_tx_pcs66_am`	Alignment marker insertion bit. In FlexE mode, 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. In OTN mode, this signal is not used. The input stream is expected to include its alignment markers.

Figure 33. 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.

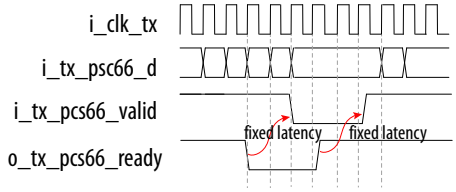
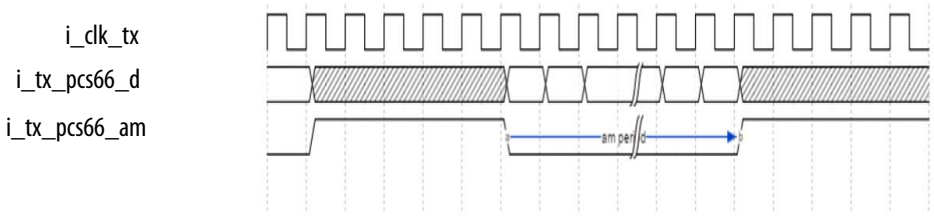


Figure 34. Alignment Marker Insertion on the TX PCS66 Interface



6.6. FlexE and OTN Mode RX Interface

The H-tile Hard IP for Ethernet Intel FPGA IP core 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 H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature. For further inquiries, contact your nearest Intel sales representative or file an Intel Premier Support (IPS) case on https://www.intel.com/content/www/us/en/programmable/my-intel/mal-home.html.

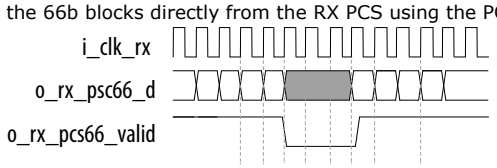
Table 20. Signals of the PCS66 RX Interface

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:

Name	Description
o_rx_pcs66_d [127:0] (in 50GBASE-R2 variations) o_rx_pcs66_d [256:0] (in 100GBASE-R4 variations)	RX PCS 66b data for 2 blocks in 50GBASE-R2 variations and 4 blocks in 100GBASE-R4 variations. • In FlexE mode, the RX PCS 66b data is aligned and descrambled. • In OTN mode, the RX PCS 66b data is only aligned.
o_rx_pcs66_valid	When asserted, indicates that the RX PCS 66b data is valid.
o_rx_pcs66_am_valid	Alignment marker indicator. When asserted, Indicates the blocks on the RX PCS 66b data signal are identified as alignment markers.

Figure 35. Receiving Data Using the PCS66 RX Interface for H-tile Hard IP for Ethernet Intel FPGA IP Core

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



6.7. Ethernet Link and Transceiver Signals

The H-tile HIP for Ethernet includes transceivers that implement two or four physical lanes at 25.78125 MHz and require one or two separately instantiated advanced transmit (ATX) PLLs to generate the high speed serial clocks. On Stratix 10 devices, only the ATX PLL supports the required data rate.

Table 21. Transceiver Signals

Signal	Description
o_tx_serial[1:0] (in 50GBASE-R2 variations) o_tx_serial[3:0] (in 100GBASE-R4 variations)	TX transceiver data. Each o_tx_serial bit becomes two physical pins that form a differential pair.
i_rx_serial[1:0] (in 50GBASE-R2 variations) i_rx_serial[3:0] (in 100GBASE-R4 variations)	RX transceiver data. Each i_rx_serial bit becomes two physical pins that form a differential pair.
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 a frequency of 322.265625 MHz or 644.53125 MHz with a ±100 ppm accuracy per the *IEEE 802.3-2015 Ethernet Standard*. In addition, i_clk_ref must meet the jitter specification of the *IEEE 802.3-2015 Ethernet Standard*.

continued...

Signal	Description
	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 Datasheet* for transceiver reference clock phase noise specifications.
`i_tx_serial_clk` (in 50GBASE-R2 variations) `i_tx_serial_clk[1:0]` (in 100GBASE-R4 variations)	High speed serial clocks driven by the ATX PLLs. 50GBASE-R2 IP core variations have a single serial clock. 100GBASE-R4 IP core variations have two serial clocks, each driven from a separate ATX PLL. The frequency of these clocks is 12.890625 GHz. You must drive these clocks from the ATX PLL or ATX PLLs that you configure separately from the H-tile Hard IP for Ethernet Intel FPGA IP core. Refer to *Adding the Transceiver PLLs*.
`i_tx_pll_locked` (in 50GBASE-R2 variations) `i_tx_pll_locked[1:0]` (in 100GBASE-R4 variations)	Lock signals from the ATX PLLs. Each bit indicates the corresponding ATX PLL is locked. 50GBASE-R2 IP core variations have a single PLL locked signal. 100GBASE-R4 variations have two PLL locked signals, each driven from a separate ATX PLL. You must drive these clocks from the ATX PLL or ATX PLLs that you configure separately from the H-tile Hard IP for Ethernet Intel FPGA IP core. Refer to *Adding the Transceiver PLLs*. The `o_clk_pll_div64` and `o_clk_pll_div66` clocks are reliable only after the `i_tx_pll_locked` bits are all high.

Related Information

- Adding the Transceiver PLLs on page 24
  Describes the required connections between the ATX PLL or ATX PLLs and the IP core.
- Intel Stratix 10 Device Datasheet
- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
  Information about the Intel Stratix 10 ATX PLL and Native PHY IP core.

6.8. Transceiver Reconfiguration Signals

You access the transceiver dynamic reconfiguration registers using the transceiver reconfiguration interface. This Avalon-MM interface provides access to the transceiver registers.

The Avalon-MM interface implements a standard memory-mapped protocol. You can connect an Avalon master to this bus to access the registers of the embedded Intel Stratix 10 Native PHY IP cores.

Table 22. 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. Each bus is a concatenation of the signals for the individual transceiver channels. 50GBASE-R2 IP core variations have two individual transceiver channels; 100GBASE-R4 variations have four individual transceiver channels.

Port Name	Description
`i_xcvr_reconfig_write[1:0]` (in 50GBASE-R2 variations) `i_xcvr_reconfig_write[3:0]` (in 100GBASE-R4 variations)	Write request signal. Signal is active high. To request to write to any of the transceiver reconfiguration registers of the transceiver channel that is configured for lane *n*, assert `i_xcvr_reconfig_write[n]`.
`i_xcvr_reconfig_read[1:0]` (in 50GBASE-R2 variations)	Read request signal. Signal is active high.

continued...

Port Name	Description
i_xcvr_reconfig_read[3:0] (in 100GBASE-R4 variations)	To request to read from any of the transceiver reconfiguration registers of the transceiver channel that is configured for lane *n*, assert i_xcvr_reconfig_read[*n*].
i_xcvr_reconfig_address[21:0] (in 50GBASE-R2 variations) i_xcvr_reconfig_address[43:0] (in 100GBASE-R4 variations)	Address bus. Drive the register address for the transceiver reconfiguration register to which you wish to write or from which you wish to read, on the corresponding 11 bits of i_xcvr_reconfig_address. For example, if you wish to read the value in the transceiver reconfiguration register at offset 0x4E0 for lane 1, drive the value of 0x4E0 on i_xcvr_reconfig_address[21:11] while you assert i_xcvr_reconfig_read[1].
i_xcvr_reconfig_writedata[32:0] (in 50GBASE-R2 variations) i_xcvr_reconfig_writedata[63:0] (in 100GBASE-R4 variations)	Write data bus. i_xcvr reconfig_address[(11(n+1)-1:11n] specifies the write address for the write data on i_xcvr_reconfig_writedata[16(n+1)-1:16n]. For example, to write to the transceiver reconfiguration register address at offset 0x4E0 for lane 1, drive the register address on i_xcvr reconfig_address[21:11], assert i_xcvr_reconfig_read[1], and write the data to i_xcvr_reconfig_writedata[31:16].
o_xcvr_reconfig_readdata[32:0] (in 50GBASE-R2 variations) o_xcvr_reconfig_readdata[63:0] (in 100GBASE-R4 variations)	Read data bus. i_xcvr reconfig_address[(11(n+1)-1:11n] specifies the read address for the read data on o_xcvr_reconfig_readdata[16(n+1)-1:16n]. For example, to read from the transceiver reconfiguration register address at offset 0x4E0 for lane 1, drive the register address on i_xcvr reconfig_address[21:11], assert i_xcvr_reconfig_write[1], and after o_xcvr_reconfig_waitrequest[1] is deasserted, read the data on o_xcvr_reconfig_readdata[31:16]. Note that the o_xcvr_reconfig_readdata bit range for a lane is valid only after the corresponding bit of o_xcvr_reconfig_waitrequest is deasserted.
o_xcvr_reconfig_waitrequest[1:0] (in 50GBASE-R2 variations) o_xcvr_reconfig_waitrequest[3:0] (in 100GBASE-R4 variations)	Indicates the Avalon-MM interface is busy. Keep each i_xcvr_reconfig_write or i_xcvr_reconfig_read bit asserted until the corresponding o_xcvr_reconfig_waitrequest bit is deasserted.

Figure 36. Writing to Transceiver Reconfiguration CSRs through Transceiver Avalon Memory-Mapped (MM) Interface

Requirements when performing a write to transceiver reconfiguration CSRs:

- The write request must held high until o_xcvr_reconfig_waitrequest is de-asserted if the write begins while o_xcvr_reconfig_waitrequest is high.
- Wait for o_xcvr_reconfig_waitrequest to go high before asserting a write command. Hold the write command until o_xcvr_reconfig_waitrequest goes low again.
- Reads and writes cannot be performed simultaneously.
- When multiple CSRs have the same address, you may need to perform a Read-Modify-Write to change the desired CSR without changing the value of the CSRs in the same address.

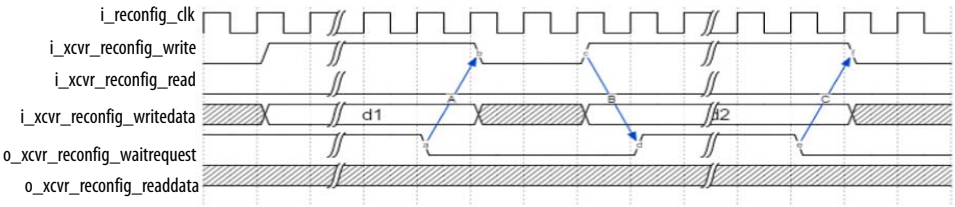
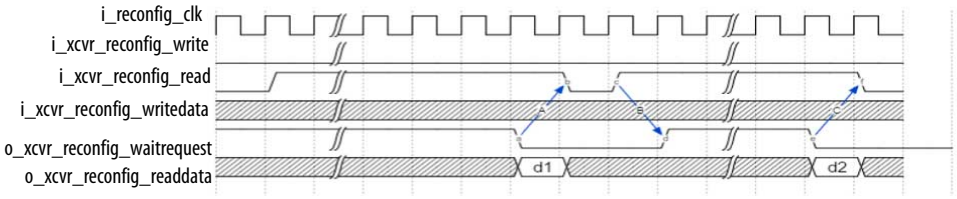


Figure 37. Reading from Transceiver Reconfiguration CSRs through Transceiver Avalon Memory-Mapped (MM) Interface

Requirements when performing a write to transceiver reconfiguration CSRs:

- The read request must held high until `o_xcvr_reconfig_waitrequest` is de-asserted if the write begins while `o_xcvr_reconfig_waitrequest` is high.
- Wait for `o_xcvr_reconfig_waitrequest` to go high before asserting a write command. Hold the read command until `o_xcvr_reconfig_waitrequest` goes low again.
- Reads and writes cannot be performed simultaneously.



Related Information

- Intel Stratix 10 L- and H-Tile Transceiver PHY User Guide
  Provides more information about the transceiver reconfiguration interface in H-tile devices, including timing diagrams for reads and writes.
- Intel Stratix 10 H-tile Transceiver Registers

6.8.1. Disabling Background Calibration

H-tile Hard IP for Ethernet Intel FPGA IP implements the Auto Adaptation triggering for RX PMA CTLE/DFE mode.

For Intel Stratix 10 H-tile production devices, disable the background calibration first prior to accessing the transceiver core reconfiguration register. The Intel Stratix 10 H-tile ES devices do not have background calibration.

In Intel Quartus Prime software version 19.2 onwards, use the following steps to access the transceiver core reconfiguration registers:

1. Write `0x0` into register `0x542[0]` of the transceiver control and status registers using the transceiver reconfiguration Avalon-MM interface to disable background calibration.
2. Access the transceiver register, for example, to perform the transceiver reconfiguration.
3. Once completed, write `0x1` into register `0x542[0]` of the transceiver control and status registers using the transceiver reconfiguration Avalon-MM interface to enable background calibration.

Related Information

Intel Stratix 10 10 L- and H-Tile Transceiver PHY User Guide

6.9. Ethernet Reconfiguration Interface

You access Ethernet control and status registers of the H-tile Hard IP for Ethernet Intel FPGA IP core during normal operation using an Avalon-MM interface called the Ethernet reconfiguration 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` and `soft_sys_rst` signals. 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 23. 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 control this interface and the transceiver reconfiguration interface. However, the two interfaces access disjoint sets of registers.

Signal	Description
`i_eth_reconfig_addr[15:0]`	Drives the Avalon-MM register address.
`i_eth_reconfig_read`	When asserted, specifies a read request.
`i_eth_reconfig_write`	When asserted, specifies a write request.
`o_eth_reconfig_readdata[31:0]`	Drives read data. Valid when `o_eth_reconfig_readdata_valid` is asserted.
`o_eth_reconfig_readdata_valid`	When asserted, indicates that `i_eth_reconfig_read_data[31:0]` is valid.
`i_eth_reconfig_writedata[31:0]`	Drives the write data.
`o_eth_reconfig_waitrequest`	Indicates that the Ethernet reconfiguration interface is not ready to complete the read or write transaction.

Related Information

Typical Read and Write Transfers section in the *Avalon Interface Specifications*
Describes typical Avalon-MM read and write transfers with a slave-controlled waitrequest signal.

6.10. Miscellaneous Status and Debug Signals

The H-tile Hard IP for Ethernet Intel FPGA IP core 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 24. Miscellaneous Status and Debug Signals

All of the miscellaneous status and debug signals except the `i_stats_snapshot` signal are asynchronous.

Signal	Description
`o_cdr_lock`	Indicates that the recovered clocks are locked to data. The `o_clk_rec_div64` and `o_clk_rec_div66` clocks are reliable only after `o_cdr_lock` is asserted.
`o_tx_lanes_stable`	Asserted when all physical TX lanes are stable and ready to transmit data.
`o_rx_block_lock`	Asserted when the IP core completes 66-bit block boundary alignment on all PCS lanes.

continued...

Signal	Description
o_rx_am_lock	Asserted when the RX PCS completes detection of alignment markers and deskew of the PCS lanes.
o_rx_pcs_ready	Asserted when the RX lanes are fully aligned and ready to receive data.
o_local_fault_status	Asserted when the RX MAC 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_remote_fault_status	Asserted when the RX MAC detects a remote fault: the remote link partner sent remote fault order 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_stats_snapshot	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. This signal is synchronous with the i_clk_tx clock. In asynchronous clock mode, use o_clk_pll_div64 as clock source.
o_rx_hi_ber	Asserted to indicate the RX PCS 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_ehip_ready	The IP core 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 IP core 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 IP core datapath is not ready for data on the client interface nor ready for register accesses on the Ethernet reconfiguration interface.

Figure 38. Status Interface Behavior during Link Startup with Bidirectional Link Fault

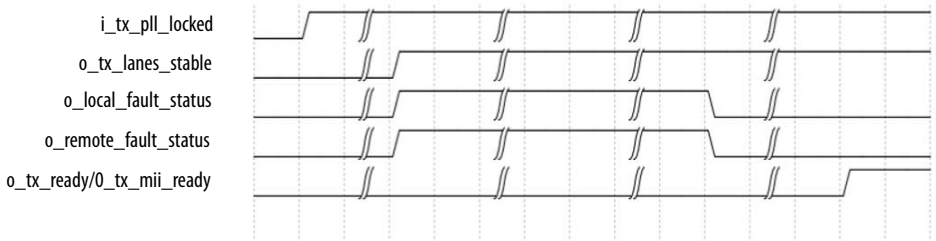
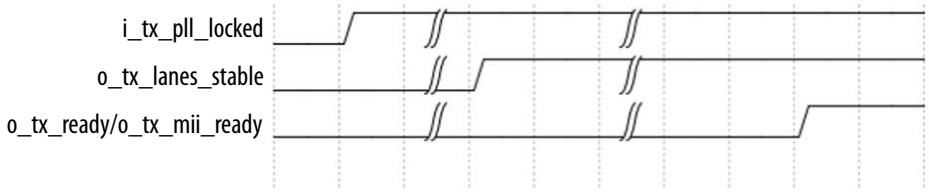


Figure 39. Status Interface Behavior during Link Startup with Unidirectional or Link Fault Disable



Related Information

Reset Signals on page 78

6.11. 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 transceiver and Ethernet reconfiguration registers.

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 25. Reset Signals

Signal	Description
`i_tx_rst_n`	Active-low hard-reset asynchronous 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_rx_rst_n`	Active-low hard-reset asynchronous 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_csr_rst_n`	Active-low hard asynchronous global reset. Resets the full IP core. 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 H-tile Hard IP for Ethernet Intel FPGA IP core Avalon-MM interfaces, both the transceiver reconfiguration interface and the Ethernet reconfiguration interface, but not the registers to which they provide access. This signal is synchronous with the `i_reconfig_clk` clock.

Related Information

- Reset on page 58
- Miscellaneous Status and Debug Signals on page 76

6.12. Clocks

You must set the transceiver reference clock (`i_clk_ref`) frequency to a value that the IP core supports. The H-tile Hard IP for Ethernet Intel FPGA IP core supports a `clk_ref` frequency of 644.53125 MHz ±100 ppm or 322.265625 MHz ±100 ppm. The ±100ppm value is required for any clock source providing the transceiver reference clock.

All H-tile Hard IP for Ethernet Intel FPGA IP core variations support the Synchronous Ethernet standard, whether or not you turn on the **Enable SyncE** parameter in the parameter editor. Sync-E variations provide the RX recovered clock as a top-level output signal.

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 TX PLL reference clock 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 H-tile Hard IP for Ethernet Intel FPGA IP core performs the filtering.

Table 26. Clock Inputs

Describes the input clocks that you must provide.

Signal Name	Description
`i_clk_tx`	The TX clock for the IP core is `i_clk_tx`. The frequency of this clock is 402.83203125 MHz.
`i_clk_rx`	The RX clock for the IP core is `i_clk_rx`. The frequency of this clock is 402.83203125 MHz.
`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 a frequency of 322.265625 MHz or 644.53125 MHz with a ±100 ppm accuracy per the *IEEE 802.3-2015 Ethernet Standard*. 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 Datasheet* for transceiver reference clock phase noise specifications.
`i_tx_serial_clk` (in 50GBASE-R2 variations) `i_tx_serial_clk[1:0]` (in 100GBASE-R4 variations)	High speed serial clocks driven by the ATX PLLs. 50GBASE-R2 IP core variations have a single serial clock. 100GBASE-R4 IP core variations have two serial clocks, each driven from a separate ATX PLL. The frequency of these clocks is 12.890625 GHz. You must drive these clocks from the ATX PLL or ATX PLLs that you configure separately from the H-tile Hard IP for Ethernet Intel FPGA IP core. Refer to *Adding the Transceiver PLLs*.
`i_reconfig_clk`	Avalon clock for the H-tile Hard IP for Ethernet Intel FPGA IP core transceiver reconfiguration interface and Ethernet reconfiguration interface. The clock frequency is 100-162 MHz. All transceiver reconfiguration interface and Ethernet reconfiguration interface signals are synchronous to `i_reconfig_clk`.

Table 27. 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	Hard IP for Ethernet block clock. The clock frequency is 402.83203125 MHz. This clock is reliable only after i_tx_pll_locked is asserted.
o_clk_pll_div66	Hard IP for Ethernet block clock × 64/66. The clock frequency is 390.625 MHz. This clock is reliable only after i_tx_pll_locked is asserted.
o_clk_rec_div64	Derived from RX recovered clock. This clock supports the Synchronous Ethernet standard. The RX recovered clock frequency is 402.83203125 MHz ±100 ppm during normal operation. This clock is reliable only after o_cdr_lock is asserted. 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 in your Synchronous Ethernet system. To do so you must include an additional component on your board. The IP core does not provide filtering.
o_clk_rec_div66	Derived from RX recovered clock. This clock supports the Synchronous Ethernet standard. The RX recovered clock frequency is 390.625 MHz ±100 ppm during normal operation. This clock is reliable only after o_cdr_lock is asserted. 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 in your Synchronous Ethernet system. To do so you must include an additional component on your board. The IP core does not provide filtering.

Related Information

- Clock Requirements on page 27
- Intel Stratix 10 Device Datasheet
  Provides transceiver reference clock phase noise specifications.

7. H-tile Hard IP for Ethernet Intel FPGA IP User Guide Archives

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, IPs have a new IP versioning scheme.

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

IP Core Version	User Guide
18.1	Intel Stratix 10 H-tile Hard IP for Ethernet IP User Guide 18.1
18.0	Intel Stratix 10 H-tile Hard IP for Ethernet IP User Guide 18.0
17.1	Intel Stratix 10 H-tile Hard IP for Ethernet IP User Guide 17.1

8. Document Revision History for the H-Tile Hard IP for Ethernet FPGA IP User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.10.31	19.3	19.2.0	• Replaced Altera Debug Master Endpoint (ADME) with Native PHY Debug Master Endpoint (NPDME). • Added **Enable JTAG to Avalon Master Bridge** parameter in the *Parameter Editor Parameters* section. • Updated *Release Information* section: — Added IP versioning description — Updated release information — Updated ordering code from IP-ETH-HTILEHARDIP to IP-ETH-HTILEHIP • Added section *Disabling Background Calibration* to clarify support for Auto adaptation triggering for RX PMA CTLE/DFE mode. • Removed RS-FEC from the *TX PCS Datapath* and *RX PCS Datapath* figures. RS-FEC is not supported in the H-tile Hard IP for Ethernet Intel FPGA IP. • Updated *System Consideration* by including the `i_reconfig_reset` signal in the system reset operation. • In *PLL Configuration Example for 100GBASE-R4 (Synchronous Mode) IP Core Variation* figure, added arrow between the Main ATX PLL block and ATX PLL (Clock Buffer) block. • Added *TX Avalon-ST MAC Client Interface for H-tile Hard IP for Ethernet Intel FPGA IP Core with `readyLatency` = 3* figure in the *TX MAC Interface to User Logic* section.
2019.02.12	18.1	18.1	Corrected the signal width error for the `i_eth_reconfig_addr` signal in the *Interfaces and Signal Descriptions* and *Ethernet Reconfiguration Interface* sections. The correct width is 16 bits, not 12 bits.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
2019.01.21	18.1	18.1	• Created chapters for MAC and PCS interfaces: — *H-tile Hard IP for Ethernet Intel FPGA MAC Interface* — *H-tile Hard IP for Ethernet Intel FPGA PCS Only/ PCS66 Interface* • Renamed the Soft eHIP Reset Sequencer and KR Reset Controller block to KR Soft Reset Sequencer and Controller in the H-tile Hard IP for Ethernet FPGA IP block diagram for better clarity. • Edited the note about generating transceiver PLLs in the *Adding the Transceiver PLLs* section to clarify that you need to generate ATX PLL IP cores to connect in your design if it includes multiple instances of the H-tile Hard IP for Ethernet FPGA IP. • Corrected the PLL signal name from `tx_serial_clk` to `tx_serial_clk_gxt` in the *Adding the Transceiver PLLs* section. • Updated the descriptions for the following registers: — ANLT Sequencer Status — Auto Negotiation Status Register — Auto Negotiation Config Register 3 — Link Training Config Register 1 — Link Training Status Register 1 — Local Link Training Parameters
2018.08.10	18.0	18.0	Added a note to clarify that the H-tile Hard IP for Ethernet Intel FPGA IP provides preliminary support for the OTN feature in the following sections: • *About the H-tile Hard IP for Ethernet Intel FPGA* IP Core • *IP Core Supported Features* • *Resource Utilization* • *Parameter Editor Parameters* • *Functional Description* • *FlexE and OTN Mode TX Interface* • *FlexE and OTN Mode RX Interface*
2018.07.20	18.0	18.0	• Added Functional Description section. • Added OTN and FlexE features in the *H-tile Hard IP for Ethernet Intel FPGA IP Core Supported Features* section. • Added **Link fault generation option**, **Enable asynchronous adapter clock**, and **Enable Altera Debug Master Endpoint (ADME)** parameters in the *Parameter Editor Parameters* section. • Added OTN and FlexE variants in the *Parameter Editor Parameters* section. • Added *Ethernet Reconfiguration and Status Register Descriptions* section. • Updated *IP Core FPGA Resource Utilization* table with resource utilization for: — 50-Gbps and 100-Gbps PCS only variants — 50-Gbps and 100-Gbps OTN only variants — 50-Gbps and 100-Gbps FlexE only variants • Updated configuration for transceiver PLLs when using 100-Gbps data rate in 50-Gbps and 100-Gbps MAC + PCS with Auto Negotiation and Link Training variants*Adding the Transceiver PLLs* topic.

continued...

Document Version	Intel Quartus Prime Version	IP Version	Changes
			• Added configuration for transceiver PLLs when using 50-Gbps data in *Adding the Transceiver PLLs* topic. • Added waveform diagrams for: — 50-Gbps and 100-Gbps MAC + PCS Alignment marker on TX PCS interface — Data transmit on TX FlexE and OTN interfaces — Alignment marker on TX FlexE and OTN interfaces — Data received on TX FlexE and OTN interfaces — Write to/read from transceiver reconfiguration CSRs using Avalon-MM interface — Status interface behavior during link startup with and without bidirectional link fault enabled — TX and RX datapath reset sequences • Added *Reset Signal Functions* table in *Reset* topic. • Renamed Intel Stratix 10 H-Tile Hard IP for Ethernet to H-tile Hard IP for Ethernet Intel FPGA as per Intel rebranding.
2018.01.12	17.1	17.1	Initial public release. At this time, the *Registers* and *Functional Description* chapters are pending.

A. Advanced RTL Parameters

The H-tile Hard IP for Ethernet Intel FPGA IP core provides advanced parameters in the generated RTL that you can modify for your IP core instance. In most cases you should leave these parameters at their default values.

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.

The most useful RTL parameters are listed in the *RTL Parameters* section. The RTL parameters in this appendix are provided for advanced applications. In most cases these parameters are not useful, either because all IP core instances in the same design usually have the same value and the parameter editor parameter suffices to specify the value, or because the Intel PSG team recommends that you use the default value.

Table 28. H-tile Hard IP for Ethernet Intel FPGA RTL Parameters

Parameter	Parameter Description
Parameters Available for all IP Core Variations	
`hi_ber_monitor`	Enables the RX PCS hi-BER monitor. • Value `enable` (default value): Ethernet standard compliant setting. The IP core hi-BER monitor is turned on. The RX PCS counts the number of RX frame header errors according to Figure 82-15 in the *IEEE Standard 802.3–2015*, using the `xus_timer_window` and `ber_invalid_count` values appropriate for the IP core data rate. When the IP core detects a hi_ber condition, it sets the value of the `o_hi_ber` signal to the value of 1. • Value `disable`: The hi-BER monitor is turned off. The value of this parameter determines the initial and reset values of the `use_hi_ber_monitor` field (bit [20]) of the RXPCS_CONF register at 0ffset 0x360.
`rx_pcs_max_skew`	Specifies the maximum RX PCS skew the IP core allows. • Value range is 1 through 47 (decimal). • Default value is 47 decimal. • You can modify this value for testing. The value of this parameter determines the initial and reset values of the `rc_pcs_max_skew[5:0]` field (bits [19:14]]) of the RXPCS_CONF register at 0ffset 0x360.
Parameters Available for 50GBASE-R2 Variations Only	
`am_encoding40g_0`	50GBASE-R2 alignment marker encoding for PCS lane number 0

continued...

Parameter	Parameter Description
	• Default value is Ethernet standard value, set according to Table 82-3 in the *IEEE Standard 802.3–2015*: 0x90_76_47. • Value range is 0 through 2^{24}–1. • Ethernet link partner must expect the same value for PCS lane 0. Otherwise, alignment cannot succeed. • The IntelFPGA team recommends that you maintain this parameter at its default value. The value of this parameter determines the initial and reset values of the `AM_ENCODING_0` register at offset 0x376.
`am_encoding40g_1`	50GBASE-R2 alignment marker encoding for PCS lane number 1 • Default value is Ethernet standard value, set according to Table 82-3 in the *IEEE Standard 802.3–2015*: 0xF0_C4_E6. • Value range is 0 through 2^{24}–1. • Ethernet link partner must expect the same value for PCS lane 1. Otherwise, alignment cannot succeed. • The IntelFPGA team recommends that you maintain this parameter at its default value. The value of this parameter determines the initial and reset values of the `AM_ENCODING_1` register at offset 0x377.
`am_encoding40g_2`	50GBASE-R2 alignment marker encoding for PCS lane number 2 • Default value is Ethernet standard value, set according to Table 82-3 in the *IEEE Standard 802.3–2015*: 0xC5_65_9B. • Value range is 0 through 2^{24}–1. • Ethernet link partner must expect the same value for PCS lane 2. Otherwise, alignment cannot succeed. • The IntelFPGA team recommends that you maintain this parameter at its default value. The value of this parameter determines the initial and reset values of the `AM_ENCODING_2` register at offset 0x378.
`am_encoding40g_3`	50GBASE-R2 alignment marker encoding for PCS lane number 3 • Default value is Ethernet standard value, set according to Table 82-3 in the *IEEE Standard 802.3–2015*: 0xA2_79_3D. • Value range is 0 through 2^{24}–1. • Ethernet link partner must expect the same value for PCS lane 3. Otherwise, alignment cannot succeed. • The IntelFPGA team recommends that you maintain this parameter at its default value. The value of this parameter determines the initial and reset values of the `AM_ENCODING_3` register at offset 0x379.
Parameters Available for MAC+PCS IP Core Variations Only	
`enforce_max_frame_size`	Specifies whether the IP core is able to receive an oversized packet or truncates these packets. • Default value is the value you specified for the **Enforce maximum frame size** parameter. • Value `disable`: For RX frames that exceed the value of the RTL parameter `rx_max_frame_size`, the IP core increments the RX oversized packets counter at offsets 0x924 and 0x925 and sets bit [3] (oversized) of the `o_rx_error` bus on the RX client interface at the appropriate time. • Value `enable`: For RX frames that exceed the value of the RTL parameter `rx_max_frame_size`, the IP core truncates the packet to the maximum RX frame size, increments the RX oversized packets counter at offsets 0x924 and 0x925, increments the appropriate RX FCS errored packet counters, and sets bits [3] (oversized) and [1] (CRC error) of the `o_rx_error` bus on the RX client interface at the appropriate time. The value of this parameter determines the initial and reset values of the `enforce_max_rx` field (bit [7]) of the `RXMAC_CONTROL` register at 0ffset 0x50A.

continued...

Parameter	Parameter Description
`flow_control`	Sets the flow control mode for the TX and RX MAC. • Value `none` (default if the parameter editor **Stop TX traffic when link partner sends pause** parameter has the value of **Disable Flow Control**): The IP core does not implement flow control. • Value `sfc`: The IP core implements strictly compliant Ethernet standard flow control. Both the `i_tx_pause` and `o_rx_pause` ports are functioning, and the TX MAC stops traffic if the IP core receives a PAUSE XOFF frame on the Ethernet link. • Value `sfc_no_xoff`: Both the `i_tx_pause` and `o_rx_pause` ports are functioning, but the TX MAC does not stop traffic if the IP core receives a PAUSE XOFF frame on the Ethernet link. • Value `pfc`: Both the `i_tx_pfc` and `o_rx_pfc` ports are functioning, and the TX MAC stops traffic if the IP core receives a PAUSE XOFF frame on the Ethernet link. • Value `pfc_no_xoff`: The IP core implements strictly compliant Ethernet priority flow control. Both the `i_tx_pfc` and `o_rx_pfc` ports are functioning, but the TX MAC does not stop traffic if the IP core receives a PAUSE XOFF frame on the Ethernet link. • Value `both` (default if the parameter editor **Stop TX traffic when link partner sends pause** parameter has the value of **Yes**): All of the `i_tx_pause`, `o_rx_pause`, `i_tx_pfc`, and `o_rx_pfc` ports are functioning, and the TX MAC stops traffic if the IP core receives a PAUSE XOFF frame on the Ethernet link. • Value `both_no_xoff` (default if the parameter editor **Stop TX traffic when link partner sends pause** parameter has the value of **No**): All of the `i_tx_pause`, `o_rx_pause`, `i_tx_pfc`, and `o_rx_pfc` ports are functioning, but the TX MAC does not stop traffic if the IP core receives a PAUSE XOFF frame on the Ethernet link. The value of this parameter determines the initial and reset values of these register fields: • `en_pfc_port[8:0]` field (bits [8:0]) of the TX_PAUSE_EN register at 0ffset 0x605. • `en_xoff_qnum_sel[2:0]` field (bits [2:0]) of the TX_XOF_EN_TX_PAUSE_QNUMBER register at 0ffset 0x60A. • `en_sfc` field (bit [0]) of the TXSFC_EHIP_CFG register at 0ffset 0x611. • `en_pfc` field (bit [1]) of the TXSFC_EHIP_CFG register at 0ffset 0x611. • `en_rx_pause[7:0]` field (bits [7:0]) of the RX_PAUSE_ENABLE register at 0ffset 0x705. • `en_sfc` field (bit [0]) of the RXSFC_EHIP_CFG register at 0ffset 0x709. • `en_pfc` field (bit [1]) of the RXSFC_EHIP_CFG register at 0ffset 0x709.
`flow_control_holdoff_mode`	Sets the holdoff timer source for the TX PAUSE and TX PFC queues. • Value `per_queue` (default value): The IP core regulates TX PAUSE (`i_tx_pause`) and each TX PFC queue (each `i_tx_pfc[n]`) with their own individual holdoff timers. • Value `uniform`: The IP core regulates TX PAUSE (`i_tx_pause`) and each TX PFC queue (each `i_tx_pfc[n]`) with the uniform holdoff timer. Each of these queues is regulated with the shorter of the uniform holdoff timer and its own individual holdoff timer. In priority flow control, if all queues have the same holdoff time, using this mode increases the chance that the IP core sends triggered XOFF frames for all PFC queues in the same Ethernet frame, increasing throughput. • Value `no_holdoff`: No holdoff timers. Not recommended. The value of this parameter determines the initial and reset values of these register fields: • `ldoff[8:0]` field (bits [8:0]) of the RETRANSMIT_XOFF_HOLDOFF_EN register at 0ffset 0x607. • `en_holdoff_all` field (bit [0]) of the CFG_RETRANSMIT_HOLDOFF_EN register at 0ffset 0x60B.
`forward_rx_pause_requests`	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.

continued...

Parameter	Parameter Description
	• Default value is the value you specified for the **Forward RX pause requests** parameter. • Value `enable`: The RX MAC forwards incoming PAUSE and PFC frames on the RX client interface. • Value `disable`: If flow control is turned on, the IP core does not forward incoming PAUSE and PFC frames on the RX client interface. The value of this parameter determines the initial and reset values of the `rx_pause_fwd` field (bit [0]) of the RX_PAUSE_FWD register at 0ffset 0x706.
`holdoff_quanta`	Sets the holdoff timer for the standard Ethernet flow control (PAUSE XOFF). • Default value is 32768. • Range is 1 through the largest number of quanta that ensures any PAUSE XOFF frame the IP core sends will arrive before the previous PAUSE XOFF frame expires. • The value of this parameter (actually, the register fields it affects) determines the frequency with which the TX MAC resends PAUSE XOFF frames while the corresponding PAUSE request is asserted. • This parameter counts quanta in 100GBASE-R4 variations, and half-quanta in 50GBASE-R2 variations. The value of this parameter determines the initial and reset values of the `holdoff_quanta[15:0]` field (bits [15:0]) of the RETRANSMIT_XOFF_HOLDOFF_QUANTA register at 0ffset 0x608.
`ipg_removed_per_am_period`	Specifies the number of inter-packet gaps the IP core removes per alignment marker period. • In 50GBASE-R2 variations, the default value of this parameter is 2 plus the value you specified for the parameter editor **Additional IPG removed per AM period** parameter. The correct value of the RTL parameter `ipg_removed_per_am_period` for standard 50-Gbps operation is 2, which causes the IP core to remove 4 IPGs per alignment marker period. • In 100GBASE-R4 variations, the default value of this parameter is 20 plus the value you specified for the parameter editor **Additional IPG removed per AM period** parameter. The correct value of the RTL parameter `ipg_removed_per_am_period` for standard 100-Gbps operation is 20. • Range is 1 through 2^{16}–1. • Each increment of 1 in the value of this parameter increases throughput by 6ppm in 50GBASE-R2 variations or by 3ppm in 100GBASE-R4 variations. • To specify large throughput increases, use the `tx_ipg_size` RTL parameter instead of the `ipg_removed_per_am_period` RTL parameter. The value of this parameter determines the initial and reset values of the `ipg_col_rem[15:0]` field (bits [15:0]) of the IPG_COL_REM register at 0ffset 0x406.
`link_fault_mode`	Specifies the IP core TX MAC and RX MAC responses to link fault events. • Default value is the value you specified for the parameter editor **Choose Link Fault generation option** parameter. • Value `lf_bidir`: Complies with the Ethernet specification. Implements the state diagram in IEEE 802.3 Figure 81-11. • Value `lf_unidir`: 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. • Value `lf_off`: Turns off TX MAC link fault responses. This option if provided for backward compatibility. The value of this parameter determines the initial and reset values of these register fields: • `en_lf` field (bit [0]) of the LINK_FAULT_CONFIG register at 0ffset 0x405. • `en_unidir` field (bit [1]) of the LINK_FAULT_CONFIG register at 0ffset 0x405.
`pause_quanta`	Specifies the number of quanta the TX MAC writes in PAUSE XOFF frames it transmits. • Default value is 65535. • Range is 1 through 65535 (2^{16}–1). • The Intel FPGA team recommends you maintain the maximum value for this parameter. The IP core can send a PAUSE XON packet at any time, to cancel the PAUSE XOFF request.

continued...

Parameter	Parameter Description
	The value of this parameter determines the initial and reset values of the `pause_quanta[15:0]` field (bits [15:0]) of the TX_PAUSE_QUANTA register at 0ffset 0x609.

`pfc_holdoff_quanta_0`
`pfc_holdoff_quanta_1`
`pfc_holdoff_quanta_2`
`pfc_holdoff_quanta_3`
`pfc_holdoff_quanta_4`
`pfc_holdoff_quanta_5`
`pfc_holdoff_quanta_6`
`pfc_holdoff_quanta_7`
	Each parameter sets the holdoff timer for the priority flow control (PFC XOFF) for the corresponding queue. For each parameter:
	• Default value is 32768.
	• Range is 1 through the largest number of quanta that ensures any PFC XOFF frame the IP core sends will arrive before the previous PFC XOFF frame expires.
	• The value of this parameter (actually, the register field it affects) determines the frequency with which the TX MAC resends PFC XOFF frames while the corresponding PFC request is asserted.
	• This parameter counts quanta in 100GBASE-R4 variations, and half-quanta in 50GBASE-R2 variations.
	The values of each of these parameters determines the initial and reset values of the following register for the corresponding queue:
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_0 register at 0ffset 0x628
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_1 register at 0ffset 0x629
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_2 register at 0ffset 0x62A
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_3 register at 0ffset 0x62B
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_4 register at 0ffset 0x62C
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_5 register at 0ffset 0x62D
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_6 register at 0ffset 0x62E
	• `holdoff_quanta[15:0]` field (bits [15:0]) of the PFC_HOLDOFF_QUANTA_7 register at 0ffset 0x62F

`pfc_pause_quanta_0`
`pfc_pause_quanta_1`
`pfc_pause_quanta_2`
`pfc_pause_quanta_3`
`pfc_pause_quanta_4`
`pfc_pause_quanta_5`
`pfc_pause_quanta_6`
`pfc_pause_quanta_7`
	Each parameter specifies the number of quanta the TX MAC writes in PFC XOFF frames it transmits for the corresponding queue. For each parameter:
	• Default value is 65535.
	• Range is 1 through 65535 (2^{16}–1).
	• The Intel FPGA team recommends you maintain the maximum value for this parameter. The IP core can send a PFC XON packet at any time, to cancel the PFC XOFF request.
	The values of each of these parameters determines the initial and reset values of the following register for the corresponding queue:
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_0 register at 0ffset 0x620
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_1 register at 0ffset 0x621
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_2 register at 0ffset 0x622
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_3 register at 0ffset 0x623
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_4 register at 0ffset 0x624
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_5 register at 0ffset 0x625
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_6 register at 0ffset 0x626
	• `pause_quanta[15:0]` field (bits [15:0]) of the PFC_PAUSE_QUANTA_7 register at 0ffset 0x627

continued...

Parameter	Parameter Description
remove_pads	Selects padding byte removal. If turned on, the IP core strips the padding bytes from the Ethernet packets before sending the data on the RX client interface. If turned off, the IP core maintains the padding bytes and includes them in the data on the RX client interface. • Default value is the value you specified for the **Remove pads** parameter. • Value enable: The RX MAC strips the padding bytes from RX packets for which the Length/Type field holds a value smaller than the payload length of the packet. If the packet is a VLAN or Stacked VLAN packet, the IP core reads the Length/Type field in the VLAN or Stacked VLAN header. • If RX CRC forwarding is turned on (the parameter editor **Keep RX CRC** parameter is turned on), you must set the remove_pads parameter to the value of disable. • Value disable: The RX MAC does not remove padding bytes from RX packets before sending the data on the RX client interface. The value of this parameter determines the initial and reset values of the remove_rx_pad field (bit [8]) of the RXMAC_CONTROL register at Offset 0x50A.
rx_length_checking	Selects whether the IP core checks TX and RX packets for length errors. Length errors include only cases where the payload is shorter than the length indicated in the appropriate Length/Type field. • Value enable (default value): The RX MAC indicates an undersized error if the payload size is smaller than the length indicated in the appropriate Length/Type field. — If the Length/Type field value is greater than 1500, it does not indicate a length, and the RX MAC does not check the packet for length errors. — If the payload is larger than the Length/Type field indicates, the RX MAC does not indicate an error, because the discrepancy could be padding bytes. The IP core accepts over-padded frames. — If RX VLAN detection is enabled, the IP core compares the payload length of a VLAN or Stacked VLAN RX packet to the Length/Type field specific to this frame type. • Value disable: The RX MAC does not check RX packets for length errors. • Despite the rx in the name of this parameter, both the TX MAC and the RX MAC count length errors in the corresponding statistics counters if this parameter has the value of enable and software has not modified the resulting value of 1 in the en_plen field (bit [0]) of the RXMAC_CONTROL register at Offset 0x50A. The value of this parameter determines the initial and reset values of the en_plen field (bit [0]) of the RXMAC_CONTROL register at Offset 0x50A.
rx_max_frame_size	Sets the maximum packet size (in bytes) the IP core can receive on the Ethernet link without reporting an oversized packet in the RX statistics counters. • Default value is the value of the parameter editor **RX maximum frame size** parameter. • Range is 65 through $2^{16}-1$. • If you turn on the **Enforce maximum frame size** parameter and do not modify the value of the enforce_max_frame_size RTL parameter, or you set the enforce_max_frame_size RTL parameter to enable, the IP core truncates incoming Ethernet packets that exceed this size. The value of this parameter determines the initial and reset values of the max_rx[15:0] field (bits [15:0]) of the MAX_RX_SIZE_CONFIG register at Offset 0x506.
rx_vlan_detection	Specifies whether the IP core treats RX VLAN and Stacked VLAN Ethernet frames as regular control frames or detects them and handles them differently. • Default value is the value you specify for the parameter editor **RX VLAN detection** parameter. • Value enable: If the parameter has this value, the IP core recognizes RX VLAN and Stacked VLAN Ethernet frames, 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. • Value disable: The IP core treats these frames as regular control frames. The value of this parameter determines the initial and reset values of the disable_rxvlan field (bit [1]) of the RXMAC_CONTROL register at Offset 0x50A.
rxcrc_covers_preamble	Specifies whether the RX MAC checks CRC under the assumption that it covers the preamble and the standard Ethernet frame (the full Ethernet packet), or only the standard Ethernet frame (without the preamble included in the calculation).

continued...

Parameter	Parameter Description
	• Value `disable` (default value): The RX MAC performs the CRC check assuming a standard Ethernet CRC calculation, which does not include the preamble. • Value `enable`: The RX MAC performs the CRC check assuming the original CRC calculation includes the preamble. This option is useful if preamble passthrough is turned on and the Ethernet link partner also considers the preamble in decoding the CRC. The value of this parameter determines the initial and reset values of the `rxcrc_covers_preamble` field (bit [1]) of the RXMAC_EHIP_CFG register at Offset 0x50B.
`strict_preamble_checking`	Determines whether the IP core rejects RX packets whose preamble is not the standard Ethernet preamble (0x55_55_55_55_55_55). • The default value is the value of the parameter editor **Enable strict preamble check** parameter. • Value `enable`: The RX MAC drops packets whose preamble (bytes [6:1]) is not the standard Ethernet preamble (0x55_55_55_55_55_55). • Value `disable`: The RX MAC does not examine the preamble bytes of incoming Ethernet packets. • This feature is available whether or not you turn on preamble-passthrough. However, the usual reason to turn on preamble-passthrough is to provide a non-standard preamble. In that case strict preamble checking defeats the purpose. • This option provides an additional layer of protection against spurious Start frames that can occur at startup or when bit errors occur. The value of this parameter determines the initial and reset values of the `en_strict_preamble` field (bit [4]) of the RXMAC_CONTROL register at Offset 0x50A.
`strict_sfd_checking`	Determines whether the IP core rejects RX packets whose SFD byte is not the standard Ethernet SFD (0xD5). • The default value is the value of the parameter editor **Enable strict SFD check** parameter. • Value `enable`: The RX MAC drops packets whose SFD (byte [0]) is not the standard Ethernet SFD (0xD5). • Value `disable`: The RX MAC does not examine the SFD byte of incoming Ethernet packets. • This feature is available whether or not you turn on preamble-passthrough. • This option provides an additional layer of protection against spurious Start frames that can occur at startup or when bit errors occur. The value of this parameter determines the initial and reset values of the `en_check_sfd` field (bit [3]) of the RXMAC_CONTROL register at Offset 0x50A.
`tx_ipg_size`	Specifies the average minimum inter-packet gap (IPG) the IP core maintains on the TX Ethernet link. • The default value is the value of the parameter editor **Average Inter-packet Gap** parameter. • Value `ipg_12`: The TX MAC maintains an average minimum IPG of 12 bytes. This value complies with the Ethernet standard. • Value `ipg_10`: The TX MAC maintains an average minimum IPG of 10 bytes. This non-compliant option increases throughput. • Value `ipg_8`: The TX MAC maintains an average minimum IPG of 8 bytes. This non-compliant option increases throughput further. • Value `ipg_1`: The TX MAC does not attempt to control the minimum IPG. This non-compliant option increases throughput. You can set this value and cede control of the IPG to the application. The value of this parameter determines the initial and reset values of the `ipg[1:0]` field (bits [2:1]) of the TXMAC_EHIP_CONFIG register at Offset 0x40B.
`tx_max_frame_size`	Sets the maximum packet size (in bytes) the IP core can transmit on the Ethernet link without reporting an oversized packet in the TX statistics counters. • Default value is the value of the parameter editor **TX maximum frame size** parameter. • Range is 65 through $2^{16}-1$.

continued...

Parameter	Parameter Description
	The value of this parameter determines the initial and reset values of the `max_tx[15:0]` field (bits [15:0]) of the `MAX_TX_SIZE_CONFIG` register at Offset 0x407.
`tx_vlan_detection`	Specifies whether the IP core treats TX VLAN and Stacked VLAN Ethernet frames as regular control frames or detects them and handles them differently. • Default value is the value you specify for the parameter editor **TX VLAN detection** parameter. • Value `enable`: If the parameter has this value, the IP core recognizes TX VLAN and Stacked VLAN Ethernet frames, 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. • Value `disable`: The IP core treats these frames as regular control frames. The value of this parameter determines the initial and reset values of the `disable_txvlan` field (bit [1]) of the `TXMAC_CONTROL` register at Offset 0x40A.
`txcrc_covers_preamble`	Specifies whether the TX MAC generates CRC that covers the preamble and the standard Ethernet frame, or only the standard Ethernet frame • Value `disable` (default value): If `i_tx_skip_crc` has the value of 0, the TX MAC performs a standard Ethernet CRC calculation, which does not include the preamble, and inserts the result in the outgoing Ethernet frame. • Value `enable`: If `i_tx_skip_crc` has the value of 0, the TX MAC performs a CRC calculation that includes the preamble, and inserts the result in the outgoing Ethernet frame. This option is useful if preamble passthrough is turned on and the Ethernet link partner also considers the preamble in decoding the CRC. The value of this parameter determines the initial and reset values of the `txcrc_covers_preamble` field (bit [9]) of the `TXMAC_EHIP_CFG` register at Offset 0x40B.
`uniform_holdoff_quanta`	Sets the uniform holdoff timer for the TX PFC queues. • Default value is 32768. • Range is 1 through the largest number of quanta that ensures any PFC XOFF frame the IP core sends will arrive before the previous PFC XOFF frame expires. • The value of this parameter, in combination with the values of the individual holdoff timers and `flow_control_holdoff_mode` (the register fields it affects) determines the frequency with which the TX MAC resends PFC XOFF frames while the corresponding PFC request is asserted. • This parameter counts quanta in 100GBASE-R4 variations, and half-quanta in 50GBASE-R2 variations. The value of this parameter determines the initial and reset values of the `holdoff_all_quanta[15:0]` field (bits [15:0]) of the `CFG_REATRANSMIT_HOLDOFF_QUANTA` register at Offset 0x60C.

Related Information

RTL Parameters on page 36
The basic RTL parameters suffice for most applications.

B. Ethernet Reconfiguration and Status Register Descriptions

You access the Ethernet registers for the H-tile Hard IP for Ethernet Intel FPGA using the Avalon-MM Ethernet reconfiguration interface on each channel. These registers use 32-bit addresses; they are not byte addressable.

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.

Table 29. Register Base Addresses

Word Offset	Register Type
0xB0-0x0E8	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

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.

B.1. Auto Negotiation and Link Training Registers

B.1.1. ANLT Sequencer Config

Provides the following config bits:

- Reset ANLT Sequencer
- Disable AN Timer
- Disable Link Fail Timer
- Force Sequencer Mode

- Link Training failure response
- Link Fail if HiBER on/off
- Skip LT on AN timeout when HiBER not used on/off

Offset: 0xB0

Access: RW

ANLT Sequencer Config Fields

Bit	Name	Description	Access	Reset
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) 0: Ignore HiBER	RW	0x1
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 • This CSR defaults to 0 in hardware (synthesis) • Set this to 1 for simulation to avoid the need to model line conditions	RW	0x0
7:4	seq_force_mode	**Force the sequencer into a specific protocol** 4'b0000: No force 4'b0010: 50GBASE-R2 4'b0011: 100GBASE-R4 All other settings are reserved • 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	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	RW	0x0
1	disable_an_timer	**Disable Autonegotiation Timer** 1: AN will wait for valid partner without timing out (default) 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. May initiate a PCS reconfiguration and/or ANLT reset 0: Normal operation	RW	0x0

B.1.2. ANLT Sequencer Status

Provides the following status bits:

- Link Ready
- AN Timeout
- LT Timeout
- Sequencer mode for PCS reconfiguration

Offset: 0xB1

Access: RO

ANLT Sequencer Status Fields

Bit	Name	Description	Access	Reset
13:8	seq_reconfig_mode	**Sequencer mode for PCS reconfiguration** 6'b000001: AN mode 6'b000010: LT mode (Clause 93) 6'b010000: 50G data mode 6'b100000: 100G data mode All other settings are 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: The ANLT Sequencer thinks the link is ready for data mode 0: Link not ready • This bit is determined by the RX PCS lane alignment status and hi_ber indication, depending on the setting of 0xB0[13] Link Fail if HiBER. • If 0xB0[2] Disable Link Fail Inhibit Timer is set, then the sequencer will continue to indicate Link Ready and stay in data mode even if the link status goes down.	RO	0x0

B.1.3. Auto Negotiation Config Register 1

Provides the following configuration options:

- Enable AN
- Enable User controlled base pages
- Enable User controlled next pages
- Local device remote fault

- Force TX nonce value
- Override AN parameters
- Ignore Nonce field
- Enable Consortium Next Page Send
- Enable Consortium Next Page Receive
- Enable Consortium Next Page Override
- Ignore Consortium Next Page Tech Ability Code
- Consortium OUI (lower 16b)

Offset: 0xC0

Access: RW

Auto Negotiation Config Register 1 Fields

Bit	Name	Description	Access	Reset
31:16	consortium_oui	**Lower 16b of Consortium Organizationally Unique Identifier** Sets the lower bits of the OUI (as defined in IEEE 802.3 Annex 73A) used to send and receive Next pages	RW	0x737D
11	ignore_consortium_next_page_tech_ability_code	**Ignore Consortium NextPage Tech Ability Code** 1: AN function will accept any unformatted Next Page after a formatted next page tagged with the proper OUI for resolving Consortium AN modes 0: The AN function will only accept an unformatted Next Page with the code 0x003 in bits D8:D0 of the page to resolve Consortium AN modes	RW	0x0
10	enable_consortium_next_page_override	**Enable Consortium Next Page override** 1: Data sent to the consortium next page will come from a Consortium Override Register instead of being set automatically 0: Normal operation	RW	0x0
9	enable_consortium_next_page_receive	**Enable Consortium Next Page receive** 1: Enable decoding received Consortium Next Pages to resolve Autonegotiation 0: Normal operation • Consortium next pages are needed to auto-negotiate with 50GBASE-R2 links defined by the Consortium standard • Turn this feature on for 50GBASE-R2 links • If this feature is disabled, Next pages will be ignored for the purpose of resolving Autonegotiation	RW	0x1
8	enable_consortium_next_page_send	**Enable Consortium Next Page send** 1: Send Consortium next pages immediately after the base page 0: Normal operation • Consortium next pages are needed to auto-negotiate with 50GBASE-R2 links defined by the Consortium standard • This feature defaults to 1 in the 50GBASE-R2 configuration, and to 0 for 100GBASE-R4 • If this feature is enabled while User Next Pages are enabled, the Consortium pages will replace the first 2 User defined pages	RW	0x0
7	ignore_nonce_field	**Ignore Nonce Field**	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		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 • In simulation, the default value is 1. In synthesis, the default value is 0.		
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
4	force_tx_nonce_value	**Force TX Nonce value** 1: Force the TX Nonce value to support UNH testing 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; the User Next page CSRs control the next page use for AN 0: The AN logic will automatically generate next pages based on the Ethernet Core Variant and its parameters • Enable this feature if you need to control the content of the AN Next page • Leave this feature disabled if you want the core to perform default negotiation for its type	RW	0x0
1	an_base_pages_ctrl	**Enable User Controlled AN Base Pages** 1: User controlled base pages are enabled; the User Base page CSRs control the base page used for AN 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	RW	0x0
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

B.1.4. Auto Negotiation Config Register 2

Provides the following configuration options:

- Reset AN
- Restart AN TXSM
- AN Next Page
- Consortium OUI (upper 8b)

Offset: 0xC1

Access: RW

Auto Negotiation Config Register 2 Fields

Bit	Name	Description	Access	Reset
23:16	consortium_oui_upper	**Consortium Organizationally Unique Identifier (upper 8b)** Sets the upper bits of the OUI (as defined in IEEE 802.3 Annex 73A) used to send and receive Next pages	RW	0x6A
8	an_next_page	**AN Next Page** 1: Indicate new next page info to send. The data in the XNP TX registers 0: No next pages to send, send TX Null pages	RW	0x0
4	restart_an_txsm	**Restart AN TXSM** 1: Restart the AN Transmit State Machine 0: Normal operation Maps to state variable mr_restart_negotiation in IEEE 802.3 CL 73.10.1.		
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	RW	0x0

B.1.5. Auto Negotiation Status Register

Provides the following status information:

- AN page received
- AN complete
- AN ADV Remote Fault
- AN RXSM Idle
- AN Ability
- AN Status
- LP An Ability
- SEQ AN Failure
- Consortium Next Page received
- Negotiation Failure
- IEEE Negotiated Port Type
- Consortium Negotiated Port Type

Offset: 0xC2

Access: RO

Auto Negotiation Status Register Fields

Bit	Name	Description	Access	Reset
27:24	consortium_negotiated_port_type	**Consortium Negotiated Port Type** 4'b0100: 50GBASE-KR2	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		4'b1000: 50GBASE-CR2		
22:12	ieee_negotiated_port_type	**IEEE Negotiated Port Type** [7]: 100GBASE-KR4 [8]: 100GBASE-CR4	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 link partner 0: No Consortium Next page found	RO	0x0
9	an_failure	**Autonegotiation Failure** 1: AN failure detected 0: Normal operation	RO	0x0
7	an_lp_ability	**Link Partner Autonegotiation Ability** 1: Link Partner is able to perform AN 0: Link Partner is not able to perform AN • This bit is set when the AN Arbitration state successfully enters ACKNOWLEDGE_DETECT • This bit is cleared upon entering ABILITY_DETECT, or when AN restarts in the case of link failure or an incompatible link *Note:* Autonegotiation allows broadcasting of the AN ability including data rate and FEC capability. You need to ensure that the link partner's data rate matches the IP core.	RO	0x0
6	an_status	**Autonegotiation Status** 1: Link is up 0: Link is down This status bit is sticky, and stays low until the CSR is read	RO	0x0
5	an_ability	**PHY Autonegotiation 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
4	an_rxsm_idle	**AN RX State Machine Idle** 1: The AN RXSM is in the Idle state. This means the incoming data is not CL73 compatible 0: AN operating normally	RO	0x0
3	an_adv_remote_fault	**Autonegotiation 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 • This status bit is sticky, and stays low until the CSR is read	RO	0x0
2	an_complete	**Autonegotiation 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 • This status bit is sticky, and stays low until the CSR is read	RO	0x0

B.1.6. Auto Negotiation Config Register 3

Provides the following configuration options:

- User base page low
- Override AN_TECH [7:0]
- Override AN_PAUSE

Offset: 0xC3

Access: RW

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
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 [7]: 100GBASE-KR4 [8]: 100GBASE-CR4	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 *Note:* Bit 49 (the PRBS bit of the AN BASE page) is generated by the SM.	RW	0x0

B.1.7. Auto Negotiation Config Register 4

Provides the upper bits of the User Controlled Autonegotiation Base Page

Offset: 0xC4

Access: RW

Auto Negotiation Config Register 4 Fields

Bit	Name	Description	Access	Reset
31:0	user_base_page_high	**User Controlled AN Base page (upper bits)** [29:5]: Technology Ability bits [4:0]: TX Nonce bits	RW	0x0

B.1.8. Auto Negotiation Config Register 5

Provides the following configuration options

- User next page (lower bits)
- Override AN_TECH []

Offset: 0xC5

Access: RW

Auto Negotiation Config Register Fields

Bit	Name	Description	Access	Reset
31:16	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 upper bits of AN_TECH used in the AN Base page [0]: 100GBASE-CR4 All other settings Reserved	RW	0x0
15:0	user_next_page_low	**User Controlled AN Next page (lower bits)** When User Controlled next gates are turned on (an_next_pages_ctrl=1), this register provides the lower bits of the User Next page that is used instead of the default page [15]: Next page bit [14]: ACK bit (controlled by the TX SM) [13]: MP bit (Message vs. Unformatted) [12]: ACK2 bits [11]: Toggle bit (controlled by the TX SM) [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 first two User Next Pages will be ignored and replaced with the Consortium Next Page sequence.	RW	0x0

B.1.9. Auto Negotiation Config Register 6

Provides the upper bits of the User Controlled Autonegotiation Next Page

Offset: 0xC6

Access: RW

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

B.1.10. Auto Negotiation Status Register 1

This register provides the lower bits of the AN RX Base page received from the link partner

Offset: 0xC7

Access: RO

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

B.1.11. Auto Negotiation Status Register 2

This register provides the upper bits of the AN RX Base page received from the link partner.

Offset: 0xC8

Access: RO

Auto Negotiation Status Register 2 Fields

Bit	Name	Description	Access	Reset
31:0	lp_base_page_high	**Link Partner Base Page (upper bits)** [29:5]: Link partner Technology Ability bits [4:0]: TX Nonce bits	RO	0x0

B.1.12. Auto Negotiation Status Register 3

This register provides the lower bits of the AN RX Next page received from the link partner.

Offset: 0xC9

Access: RO

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

B.1.13. Auto Negotiation Status Register 4

This register provides the upper bits of the AN RX Next page received from the link partner.

Offset: 0xCA

Access: RO

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 [47:16] or [31:0]	RO	0x0

B.1.14. Auto Negotiation Status Register 5

This register provides the following status information:

- Link Partner Technology Ability Field
- Link Partner Remote Fault
- Link Partner PAUSE Ability

Offset: 0xCB

Access: RO

Auto Negotiation Status Register 5 Fields

Bit	Name	Description	Access	Reset
30:28	an_lp_adv_pause	**Link Partner PAUSE Ability bits** [0]: PAUSE as defined in Annex 28B [1]: ASM_DIR as defined in Annex 28B [2]: Reserved	RO	0x0
27	an_lp_adv_remote_fault	**Link Partner Remote Fault** Remote fault bit from Link Partner	RO	0x0
22:0	an_lp_adv_tech_a	**Link Partner Technology Ability Field** [5]: 100GBASE-CR10 [6]: 100GBASE-KP4 [7]: 100GBASE-KR4 [8]: 100GBASE-CR4 [22:11]: Reserved	RO	0x0

B.1.15. Consortium Next Page Override

Provides override CSRs for the following fields of the Consortium Next Page.

- Override Consortium Next Page Technology Ability

Offset: 0xCD

Access: RW

Consortium Next Page Override Fields

Bit	Name	Description	Access	Reset
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 abilities [16:9]: Reserved, set to 0 [17]: 50GBASE-KR2 Ability [18]: 50GBASE-CR2 Ability [19]: Reserved, set to 0	RW	0x3

B.1.16. Consortium Next Page Link Partner Status

Consortium Next Page Link Partner Status This register provides the following status information:

- Link Partner Consortium Next Page Technology Ability

Offset: 0xCE

Access: RO

Consortium Next Page Link Partner Status Fields

Bit	Name	Description	Access	Reset
19:0	lp_consortium_next_page_tech	**Link Partner Consortium Next Page Technology Ability** [8:0]: Bits D8:D0 in the unformatted next page from default of 0x003 to indicate extended abilities [16:9]: Reserved [17]: 50GBASE-KR2 Ability [18]: 50GBASE-CR2 Ability [19]: Reserved	RO	0x0

B.1.17. Link Training Config Register 1

Provides CSRs for the following link training features:

- Enable Link Training
- Disable Max Wait Timer
- Disable Initialize PMA on Max Wait Timeout
- Enable Link Partner TX EQ Coefficient Override
- Enable Local TX EQ Coefficient Override
- Enable Manual RX Settings for Link Training

- Manual CTLE AC set by IP during Link Training
- Manual CTLE DC set by IP during Link Training
- Manual VGA set by IP during Link Training

Offset: `0xD0`

Access: `RW`

Link Training Config Register 1 Fields

Bit	Name	Description	Access	Reset
31:28	`lt_rx_vga`	**Manual VGA set by IP during Link Training** The IP multiply the value set in this field by 2, e.g. if the value is 7, VGA is set to 14. This file is only valid when `lt_rx_manual_mode` is set to 1.	RW	0x0
27:23	`lt_rx_clte_dc`	**Manual CTLE DC set by IP during Link Training** The IP multiply the value set in this field by 2, e.g. if the value is 7, the CTLE DC is set to 14. This file is only valid when `lt_rx_manual_mode` is set to 1.	RW	0x0
22:20	`lt_rx_clte_ac`	**Manual CTLE AC set by IP during Link Training** The IP multiply the value set in this field by 2, e.g. if the value is 7, the CTLE AC is set to 14. This file is only valid when `lt_rx_manual_mode` is set to 1.	RW	0x0
19	`lt_rx_manual_mode`	**Enable Manual RX Settings for Link Training** 1: Link training use manual RX settings from this register. 0: Link training automatically adapt RX settings. The default value is 1 in simulation and 0 in synthesis.	RW	0x0
17	`ovride_local_coef_enable`	**Enable Local TX EQ Coefficient Override** 1: Override the Local device TX EQ coefficients 0: Let the Link Partner decide the local TX EQ coefficients	RW	0x0
16	`ovride_lp_coef_enable`	**Enable Link Partner Coefficient Override** 1: Override the Link Partner EQ coefficients 0: Use the Link Training logic to decide the Link Partner TX EQ coefficients When this field is set to 1, user logic must decide the Link Partner TX EQ coefficient values.	RW	0x0
15	`disable_initialize_pma_on_max_wait_timeout`	**Disable initialize PMA on max_wait_timeout** 1: Don't initialize TX EQ to INIT values upon entry into the Training_Failure state of link training 0: Set TX EQ to INIT values upon entry into the Training_Failure state of link training (default)	RW	0x0
1	`dis_max_wait_tmr`	**Disable Max Wait Timer** 1: Disable Max Wait Timer 0: Use Max Wait Timer (default) • When Max Wait Timer is disabled, Link Training will not fail, but the IP can potentially stay in Link Training state indefinitely • Intel recommends that you enable this bit when using Link Training without Auto-negotiation • When this bit is enabled, Link Fail Timeout will also be ignored during Link Training	RW	0x0
0	`enable_link_training`	**Enable Link Training** 1: Enable link training 0: Disable link training	RW	0x1

B.1.18. Link Training Config Register 2

Provides CSRs for the following link training features:

- Restart Link Training on Lane 0
- Restart Link Training on Lane 1
- Restart Link Training on Lane 2
- Restart Link Training on Lane 3
- Updated Link Partner TX EQ Override Settings ready to be sent for Lane 0
- Updated Link Partner TX EQ Override Settings ready to be sent for Lane 1
- Updated Link Partner TX EQ Override Settings ready to be sent for Lane 2
- Updated Link Partner TX EQ Override Settings ready to be sent for Lane 3
- Updated Local TX EQ Override Settings ready to be sent for Lane 0
- Updated Local TX EQ Override Settings ready to be sent for Lane 1
- Updated Local TX EQ Override Settings ready to be sent for Lane 2
- Updated Local TX EQ Override Settings ready to be sent for Lane 3

Offset: `0xD1`

Access: `RW`

Link Training Config Register 2 Fields

Bit	Name	Description	Access	Reset
11	`updated_local_coef_ln3`	**Updated Local TX EQ Override Settings ready to be set for Lane 3** 1: Trigger LT logic to set new Local TX EQ Override settings for Lane 3 0: Normal operation Valid only when `ovride_local_coef_enable=1`. Takes values from corresponding TX EQ Override CSR. Valid only for links with 4 lanes.	`RW`	`0x0`
10	`updated_local_coef_ln2`	**Updated Local TX EQ Override Settings ready to be set for Lane 2** 1: Trigger LT logic to set new Local TX EQ Override settings for Lane 2 0: Normal operation Valid only when `ovride_local_coef_enable=1`. Takes values from corresponding TX EQ Override CSR Valid only for links with 4 lanes.	`RW`	`0x0`
9	`updated_local_coef_ln1`	**Updated Local TX EQ Override Settings ready to be set for Lane 1** 1: Trigger LT logic to set new Local TX EQ Override settings for Lane 1 0: Normal operation Valid only when `ovride_local_coef_enable=1`. Takes values from corresponding TX EQ Override CSR.	`RW`	`0x0`
8	`updated_local_coef_ln0`	**Updated Local TX EQ Override Settings ready to be set for Lane 0** 1: Trigger LT logic to set new Local TX EQ Override settings for Lane 0	`RW`	`0x0`

continued...

Bit	Name	Description	Access	Reset
		0: Normal operation Valid only when `ovride_local_coef_enable=1` Takes values from corresponding TX EQ Override CSR		
7	`updated_lp_coef_ln3`	**Updated Link Partner TX EQ Override Settings ready to be sent for Lane 3** 1: Trigger LT logic to transmit new TX EQ Override settings for Link Partner Lane 3 0: Normal operation Valid only when `ovride_lp_coef_enable=1` Takes values from corresponding TX EQ Override CSR. Valid for links with 4 lanes only.	RW	0x0
6	`updated_lp_coef_ln2`	**Updated Link Partner TX EQ Override Settings ready to be sent for Lane 2** 1: Trigger LT logic to transmit new TX EQ Override settings for Link Partner Lane 2 0: Normal operation Valid only when `ovride_lp_coef_enable=1`. Takes values from corresponding TX EQ Override CSR. Valid for links with 4 lanes only.	RW	0x0
5	`updated_lp_coef_ln1`	**Updated Link Partner TX EQ Override Settings ready to be sent for Lane 1** 1: Trigger LT logic to transmit new TX EQ Override settings for Link Partner Lane 1 0: Normal operation Valid only when `ovride_lp_coef_enable=1` Takes values from corresponding TX EQ Override CSR.	RW	0x0
4	`updated_lp_coef_ln0`	**Updated Link Partner TX EQ Override Settings ready to be sent for Lane 0** 1: Trigger LT logic to transmit new TX EQ Override settings for Link Partner Lane 0 0: Normal operation Valid only when `ovride_lp_coef_enable=1`. Takes values from corresponding TX EQ Override CSR.	RW	0x0
3	`restart_link_training_ln3`	**Restart Link Training on lane 3** 1: Restart Clause 93 start-up protocol 0: Normal operation • Corresponds to state variable mr_restart_training as defined in IEEE 802.3 CL 72.6.10.3.1 • Valid for links with 4 lanes only	RW	0x0
2	`restart_link_training_ln2`	**Restart Link Training on lane 2** 1: Restart Clause 93 start-up protocol 0: Normal operation • Corresponds to state variable mr_restart_training as defined in IEEE 802.3 CL 72.6.10.3.1 • Valid for links with 4 lanes only	RW	0x0
1	`restart_link_training_ln1`	**Restart Link Training on lane 1** 1: Restart Clause 93 start-up protocol 0: Normal operation • Corresponds to state variable mr_restart_training as defined in IEEE 802.3 CL 72.6.10.3.1	RW	0x0
0	`restart_link_training_ln0`	**Restart Link Training on lane 0** 1: Restart Clause 93 start-up protocol	RW	0x0

Bit	Name	Description	Access	Reset
		0: Normal operation • Corresponds to state variable mr_restart_training as defined in IEEE 802.3 CL 72.6.10.3.1		

B.1.19. Link Training Status Register 1

Provides Status for the following link training features:

- Receiver Trained (Lanes 0 to 3)
- Link Training Frame Lock Achieved (Lanes 0 to 3)
- Link Training Startup Protocol Status (Lanes 0 to 3)
- Link Training Failure on Lane 0 (Lanes 0 to 3)

Offset: 0xD2

Access: RO

Link Training Status Register 1 Fields

Bit	Name	Description	Access	Reset
27	link_training_failure_ln3	**Link Training Failure on Lane 3** 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 • When link training fails, the FPGA IP stops sending LT traffic • Valid only for links with 4 lanes	RO	0x0
26	link_training_startup_ln3	**Link Training Startup up Protocol in Progress on Lane 3** 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 • Bit is cleared when link training has completed or when link training fails • Valid only for links with 4 lanes	RO	0x0
25	link_training_frame_lock_ln3	**Link Training Frame Lock Achieved on Lane 3** 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** 1: Receiver training has completed successfully 0: Training has not yet completed • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1 • Bit is cleared when link training fails • Valid only on links with 4 lanes	RO	0x0
19	link_training_failure_ln2	**Link Training Failure on Lane 2** 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** 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** 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** 1: Receiver training has completed successfully 0: Training has not yet completed • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1 • Bit is cleared when link training fails • 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 has completed successfully 0: Training has not yet completed • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1 • Bit is cleared when link training fails • 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 has completed successfully 0: Training has not yet completed • Corresponds to state variable rx_trained as defined in IEEE 802.3 CL72.6.10.3.1 • Bit is cleared when link training fails	RO	0x0

B.1.20. Link Training Config Register for Lane 0

Provides CSRs for the following link training features:

- LT PRBS Pattern Select for lane 0
- LT PRBS Seed for lane 0

Offset: 0xD3

Access: RW

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** 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** 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

B.1.21. Link Training Frame Contents for Lane 0

Provides CSRs for the following fields that are transmitted during link training to the link partner for Lane 0

- TX EQ Coefficient Request to Link Partner on Lane 0
- INIT Coefficients command to Link Partner on Lane 0
- PRESET Coefficients command to Link Partner on Lane 0
- Local TX EQ Coefficient Status for Lane 0
- Local Receiver Ready Status for Lane 0
- Most Recent TX EQ Coefficient Request from Link Partner on Lane 0
- Most Recent INIT command from Link Partner on Lane 0
- Most Recent PRESET command from Link Partner on Lane 0
- Most Recent TX EQ Status from Link Partner on Lane 0
- Most Recent Receiver Ready Status from Link Partner on Lane 0

Offset: 0xD4

Access: RO and RW

Link Training Frame Contents Fields

Bit	Name	Description	Access	Reset
30	lp_receiver_ready_ln0	**Link Partner Receiver Ready Status for Lane 0** 1: The link partner receiver has determined that training is complete and is prepared to receive data 0: The link partner receiver is requesting that training continue	RO	0x0
29:24	lp_coefficient_status_ln0	**TX EQ Coefficient Status from Link Partner for Lane 0** [5:4] Status of Link Partner (+1) TX EQ Coefficient [3:2] Status of Link Partner (0) TX EQ Coefficient [1:0] Status of Link Partner (-1) TX EQ Coefficient The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RO	0x0
23	lp_preset_coefficients_ln0	**PRESET Command from Link Partner on Lane 0** 1: Set local TX EQ to PRESET 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this Field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln0=1 to write the local values.	RW	0x0
22	lp_initialize_coefficients_ln0	**INIT Command from Link Partner on Lane 0** 1: Set local TX EQ to INIT 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln0=1 to write the local values.	RW	0x0

continued...

Bit	Name	Description	Access	Reset
21:16	lp_coefficient_update_ln0	**TX EQ Coefficient Request from Link Partner on Lane 0** [5:4]: Control for Local (+1) TX EQ Coefficient [3:2]: Control for Local (0) TX EQ Coefficient [1:0]: Control for Local (-1) TX EQ Coefficient This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln0=1 to write the local values. The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RW	0x0
14	ld_receiver_ready_ln0	**Local Receiver Ready Status for Lane 0** 1: The local device receiver has determined that training is complete and is prepared to receive data 0: The local device receiver is requesting that training continue	RO	0x0
13:8	ld_coefficient_status_ln0	**Local TX EQ Coefficient Status for Lane 0** [5:4] Status of Local (+1) TX EQ Coefficient [3:2] Status of Local (0) TX EQ Coefficient [1:0] Status of Local (-1) TX EQ Coefficient The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RO	0x0
7	ld_preset_coefficients_ln0	**PRESET Coefficients command to Link Partner on Lane 0** 1: PRESET Coefficients 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln0=1 to transmit the values to the Link Partner. The PRESET command is defined in IEEE 802.3 CL72.6.10.2.3.1.	RW	0x0
6	ld_initialize_coefficients_ln0	**INIT Coefficients command to Link Partner on Lane 0** 1: INIT Coefficients 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln0=1 to transmit the values to the Link Partner.	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		The INIT command is defined in IEEE 802.3 CL72.6.10.2.3.2.		
5:0	ld_coefficient_update_ln0	**TX EQ Coefficient Request to Link Partner on Lane 0** [5:4] Control for Link Partner (+1) TX EQ Coefficient [3:2] Control for Link Partner (0) TX EQ Coefficient [1:0] Control for Link Partner (-1) TX EQ Coefficient This Field is normally Read-only, and the values are normally controlled by the Link Training SM When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln0=1 to transmit the values to the Link Partner The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RW	0x0

B.1.22. Local Transceiver TX EQ 1 Settings for Lane 0

Provides the following Local TX EQ 1 Settings for Lane 0

- Local TX EQ VOD Setting for Lane 0
- Local TX EQ Post-Tap Setting for Lane 0
- Local TX EQ Pre-Tap Setting for Lane 0

Offset: 0xD5

Access: R0

Local Transceiver TX EQ 1 Settings for Lane 0 Fields

Bit	Name	Description	Access	Reset
20:16	lt_pretap_setting_ln0	**Local TX EQ Pre-tap Setting for Lane 0** This register returns the most recent Pre-tap setting that was written to the local transceiver	RO	0x0
13:8	lt_posttap_setting_ln0	**Local TX EQ Post-tap Setting for Lane 0** This register returns the most recent Post-tap setting that was written to the local transceiver .	RO	0x0
4:0	lt_vod_setting_ln0	**Local TX EQ VOD Setting for Lane 0** This register returns the most recent VOD setting that was written to the local transceiver	RO	0x0

B.1.23. Local Transceiver TX EQ 2 Settings for Lane 0

Provides the following Local TX EQ 2 Settings for Lane 0

- VMAXRULE Override value
- Enable VMAXRULE Override
- VODMINRULE Override value
- Enable VODMINRULE Override

- VPOSTRULE Override value
- Enable VPOSTRULE Override
- VPRERULE Override value
- Enable VPRERULE Override

Offset: 0xD6

Access: RW

Local Transceiver TX EQ 2 Settings Fields

Bit	Name	Description	Access	Reset
29	lt_vpre_ovrd_en_ln0	**Enable VPRERULE Override for Lane 0** 1: Use the value of lt_vpre_ovrd to set VPRERULE 0: Use the value of VPRERULE set by the parameters that were used at compile time	RW	0x0
28:24	lt_vpre_ovrd_ln0	**VPRERULE Override value for Lane 0** When lt_vpre_ovrd_en=1, this CSR sets the maximum value of the Pre-tap on Lane 0. VPRERULE must be set to a value greater than INITPREVAL.	RW	0x0
22	lt_vpost_ovrd_en_ln0	**Enable VPOSTRULE Override for Lane 0** 1: Use the value of lt_vpost_ovrd to set VPOSTRULE. 0: Use the value of VPOSTRULE set by the parameters that were used at compile time.	RW	0x0
21:16	lt_vpost_ovrd_ln0	**VPOSTRULE Override value for Lane 0** When lt_vpost_ovrd_en=1, this CSR sets the maximum value of the Post-tap on Lane 0. VPOSTRULE must be set to a value greater than INITPOSTVAL.	RW	0x0
13	lt_vodmin_ovrd_en_ln0	**Enable VODMINRULE Override for Lane 0** 1: Use the value of lt_vodmin_ovrd to set VODMINRULE 0: Use the value of VODMINRULE set by the parameters that were used at compile time	RW	0x0
12:8	lt_vodmin_ovrd_ln0	**VODMINRULE Override value for Lane 0** When lt_vodmin_ovrd_en=1, this CSR sets the minimum setting of VOD allowed during link training for Lane 0. VODMINRULE must be set to a value less than INITMAINVAL. VODMINRULE must also be set to a value greater than VMINRULE.	RW	0x0
5	lt_vodmax_ovrd_en_ln0	**Enable VMAXRULE Override for Lane 0** 1: Use the value of lt_vodmax_ovrd to set VMAXRULE 0: Use the value of VMAXRULE set by the parameters that were used at compile time	RW	0x0
4:0	lt_vodmax_ovrd_ln0	**VMAXRULE Override Value for Lane 0** When lt_vodmax_ovrd_en=1, this CSR sets the maximum Voltage allowed during link training for Lane 0. VMAXRULE must be set to a value greater than INITMAINVAL. Note that this value also changes PREMAINVAL.	RW	0x0

B.1.24. Local Link Training Parameters

Provides the following Link Training Parameters

- Max wait timeout multiplier
- Enable Wait for frame lock before staring max wait timer
- Disable canceling link ready if `remote_rx_ready` deasserts

Offset: `0xD7`

Access: `RO` and `RW`

Local Link Training Parameters Fields

Bit	Name	Description	Access	Reset
9	`disable_link_ready_cancel`	**Disable canceling link ready if remote_rx_ready deasserts** 1: Link ready will not be automatically canceled when `remote_rx_ready` is deasserted. 0: Link ready will be canceled if remote_rx_ready is deasserted (default).	`RW`	`0x0`

B.1.25. Link Training Config Register for Lane 1

Provides CSRs for the following link training features:

- LT PRBS Pattern Select for lane 1
- LT PRBS Seed for lane 1

Offset: `0xE0`

Access: `RW`

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** Sets the initial seed for PRBS. Default value is 11'h645	`RW`	`0x645`
2:0	`lt_prbs_pattern_select_ln1`	**Link Training PRBS Pattern Select for Lane 1** 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`

B.1.26. Link Training Frame Contents for Lane 1

Provides CSRs for the following fields that are transmitted during link training to the link partner for Lane 1

- TX EQ Coefficient Request to Link Partner on Lane 1
- INIT Coefficients command to Link Partner on Lane 1
- PRESET Coefficients command to Link Partner on Lane 1
- Local TX EQ Coefficient Status for Lane 1

- Local Receiver Ready Status for Lane 1
- Most Recent TX EQ Coefficient Request from Link Partner on Lane 1
- Most Recent INIT command from Link Partner on Lane 1
- Most Recent PRESET command from Link Partner on Lane 1
- Most Recent TX EQ Status from Link Partner on Lane 1
- Most Recent Receiver Ready Status from Link Partner on Lane 1

Offset: 0xE1

Access: RO and RW

Link Training Frame Contents for Lane 1 Fields

Bit	Name	Description	Access	Reset
30	lp_receiver_ready_ln1	**Link Partner Receiver Ready Status for Lane 1** 1: The link partner receiver has determined that training is complete and is prepared to receive data 0: The link partner receiver is requesting that training continue	RO	0x0
29:24	lp_coefficient_status_ln1	**TX EQ Coefficient Status from Link Partner for Lane 1** [5:4] Status of Link Partner (+1) TX EQ Coefficient [3:2] Status of Link Partner (0) TX EQ Coefficient [1:0] Status of Link Partner (-1) TX EQ Coefficient The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RO	0x0
23	lp_preset_coefficients_ln1	**PRESET Command from Link Partner on Lane 1** 1: Set local TX EQ to PRESET 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln1=1 to write the local values.	RW	0x0
22	lp_initialize_coefficients_ln1	**INIT Command from Link Partner on Lane 1** 1: Set local TX EQ to INIT 0: Normal Operation This field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln1=1 to write the local values	RW	0x0
21:16	lp_coefficient_update_ln1	**TX EQ Coefficient Request from Link Partner on Lane 1** [5:4] Control for Local (+1) TX EQ Coefficient [3:2] Control for Local (0) TX EQ Coefficient [1:0] Control for Local (-1) TX EQ Coefficient This field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM.	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln1=1 to write the local values. The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved		
14	ld_receiver_ready_ln1	**Local Receiver Ready Status for Lane 1** 1: The local device receiver has determined that training is complete and is prepared to receive data 0: The local device receiver is requesting that training continue	RO	0x0
13:8	ld_coefficient_status_ln1	**Local TX EQ Coefficient Status for Lane 1** [5:4] Status of Local (+1) TX EQ Coefficient [3:2] Status of Local (0) TX EQ Coefficient [1:0] Status of Local (-1) TX EQ Coefficient The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RO	0x0
7	ld_preset_coefficients_ln1	**PRESET Coefficients command to Link Partner on Lane 1** 1: PRESET Coefficients 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln1=1 to transmit the values to the Link Partner. The PRESET command is defined in IEEE 802.3 CL72.6.10.2.3.1.	RW	0x0
6	ld_initialize_coefficients_ln1	**INIT Coefficients command to Link Partner on Lane 1** 1: INIT Coefficients 0: Normal Operation This field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln1=1 to transmit the values to the Link Partner. The INIT command is defined in IEEE 802.3 CL72.6.10.2.3.2.	RW	0x0
5:0	ld_coefficient_update_ln1	**TX EQ Coefficient Request to Link Partner on Lane 1** [5:4] Control for Link Partner (+1) TX EQ Coefficient [3:2] Control for Link Partner (0) TX EQ Coefficient [1:0] Control for Link Partner (-1) TX EQ Coefficient This field is normally Read-only, and the values are normally controlled by the Link Training SM.	RW	0x0

Bit	Name	Description	Access	Reset
		When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln0=1 to transmit the values to the Link Partner. The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved		

B.1.27. Local Transceiver TX EQ 1 Settings for Lane 1

Provides the following Local TX EQ 1 Settings for Lane 1

- Local TX EQ VOD Setting for Lane 1
- Local TX EQ Post-Tap Setting for Lane 1
- Local TX EQ Pre-Tap Setting for Lane 1

Offset: 0xE2

Access: R0

Local Transceiver TX EQ 1 Settings for Lane 1 Fields

Bit	Name	Description	Access	Reset
20:16	lt_pretap_setting_ln1	**Local TX EQ Pre-tap Setting for Lane 1** This register returns the most recent Pre-tap setting that was written to the local transceiver.	RO	0x0
13:8	lt_posttap_setting_ln1	**Local TX EQ Post-tap Setting for Lane 1** This register returns the most recent Post-tap setting that was written to the local transceiver	RO	0x0
4:0	lt_vod_setting_ln1	**Local TX EQ VOD Setting for Lane 1** This register returns the most recent VOD setting that was written to the local transceiver	RO	0x0

B.1.28. Local Transceiver TX EQ 2 Settings for Lane 1

Provides the following Local TX EQ 2 Settings for Lane 1

- VMAXRULE Override value
- Enable VMAXRULE Override
- VODMINRULE Override value
- Enable VODMINRULE Override
- VPOSTRULE Override value
- Enable VPOSTRULE Override
- VPRERULE Override value
- Enable VPRERULE Override

Offset: 0xE3

Access: RW

Local Transceiver TX EQ 2 Settings for Lane 1 Fields

Bit	Name	Description	Access	Reset
29	lt_vpre_ovrd_en_ln1	**Enable VPRERULE Override for Lane 1** 1: Use the value of lt_vpre_ovrd to set VPRERULE 0: Use the value of VPRERULE set by the parameters that were used at compile time	RW	0x0
28:24	lt_vpre_ovrd_ln1	**VPRERULE Override value for Lane 1** When lt_vpre_ovrd_en=1, this CSR sets the maximum value of the Pre-tap on Lane 1. VPRERULE must be set to a value greater than INITPREVAL.	RW	0x0
22	lt_vpost_ovrd_en_ln1	**Enable VPOSTRULE Override for Lane 1** 1: Use the value of lt_vpost_ovrd to set VPOSTRULE. 0: Use the value of VPOSTRULE set by the parameters that were used at compile time.	RW	0x0
21:16	lt_vpost_ovrd_ln1	**VPOSTRULE Override value for Lane 1** When lt_vpost_ovrd_en=1, this CSR sets the maximum value of the Post-tap on Lane 1. VPOSTRULE must be set to a value greater than INITPOSTVAL.	RW	0x0
13	lt_vodmin_ovrd_en_ln1	**Enable VODMINRULE Override for Lane 1** 1: Use the value of lt_vodmin_ovrd to set VODMINRULE 0: Use the value of VODMINRULE set by the parameters that were used at compile time	RW	0x0
12:8	lt_vodmin_ovrd_ln1	**VODMINRULE Override value for Lane 1** When lt_vodmin_ovrd_en=1, this CSR sets the minimum setting of VOD allowed during link training for Lane 1. VODMINRULE must be set to a value less than INITMAINVAL. VODMINRULE must also be set to a value greater than VMINRULE.	RW	0x0
5	lt_vodmax_ovrd_en_ln1	**Enable VMAXRULE Override for Lane 1** 1: Use the value of lt_vodmax_ovrd to set VMAXRULE 0: Use the value of VMAXRULE set by the parameters that were used at compile time	RW	0x0
4:0	lt_vodmax_ovrd_ln1	**VMAXRULE Override Value for Lane 1** When lt_vodmax_ovrd_en=1, this CSR sets the maximum Voltage allowed during link training for Lane 1. VMAXRULE must be set to a value greater than INITMAINVAL. Note that this value also changes PREMAINVAL.	RW	0x0

B.1.29. Link Training Config Register for Lane 2

Link Training Config Register for Lane 2 Provides CSRs for the following link training features

- LT PRBS Pattern Select for lane 2
- LT PRBS Seed for lane 2

Offset: 0xE4

Access: RW

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** 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** 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

B.1.30. Link Training Frame Contents for Lane 2

Provides CSRs for the following fields that are transmitted during link training to the link partner for Lane 2

- TX EQ Coefficient Request to Link Partner on Lane 2
- INIT Coefficients command to Link Partner on Lane 2
- PRESET Coefficients command to Link Partner on Lane 2
- Local TX EQ Coefficient Status for Lane 2
- Local Receiver Ready Status for Lane 2
- Most Recent TX EQ Coefficient Request from Link Partner on Lane 2
- Most Recent INIT command from Link Partner on Lane 2
- Most Recent PRESET command from Link Partner on Lane 2
- Most Recent TX EQ Status from Link Partner on Lane 2
- Most Recent Receiver Ready Status from Link Partner on Lane 2

Offset: 0xE5

Access: RO and RW

Link Training Frame Contents for Lane 2 Fields

Bit	Name	Description	Access	Reset
30	lp_receiver_ready_ln2	**Link Partner Receiver Ready Status for Lane 2** 1: The link partner receiver has determined that training is complete and is prepared to receive data 0: The link partner receiver is requesting that training continue	RO	0x0
29:24	lp_coefficient_status_ln2	**TX EQ Coefficient Status from Link Partner for Lane 2** [5:4] Status of Link Partner (+1) TX EQ Coefficient [3:2] Status of Link Partner (0) TX EQ Coefficient [1:0] Status of Link Partner (-1) TX EQ Coefficient The Coefficient values are encoded as follows:	RO	0x0

continued...

Bit	Name	Description	Access	Reset
		• 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved		
23	lp_preset_coefficients_ln2	**PRESET Command from Link Partner on Lane 2** 1: Set local TX EQ to PRESET 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln2=1 to write the local values.	RW	0x0
22	lp_initialize_coefficients_ln2	**INIT Command from Link Partner on Lane 2** 1: Set local TX EQ to INIT 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln2=1 to write the local values.	RW	0x0
21:16	lp_coefficient_update_ln2	**TX EQ Coefficient Request from Link Partner on Lane 2** [5:4] Control for Local (+1) TX EQ Coefficient [3:2] Control for Local (0) TX EQ Coefficient [1:0] Control for Local (-1) TX EQ Coefficient This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this Field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln2=1 to write the local values. The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RW	0x0
14	ld_receiver_ready_ln2	**Local Receiver Ready Status for Lane 2** 1: The local device receiver has determined that training is complete and is prepared to receive data 0: The local device receiver is requesting that training continue	RO	0x0
13:8	ld_coefficient_status_ln2	**Local TX EQ Coefficient Status for Lane 2** [5:4] Status of Local (+1) TX EQ Coefficient [3:2] Status of Local (0) TX EQ Coefficient [1:0] Status of Local (-1) TX EQ Coefficient The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RO	0x0
7	ld_preset_coefficients_ln2	**PRESET Coefficients command to Link Partner on Lane 2**	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		1: PRESET Coefficients 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, useupdated_lp_coef_ln2=1 to transmit the values to the Link Partner. The PRESET command is defined in IEEE 802.3 CL72.6.10.2.3.1.		
6	ld_initialize_coefficients_ln2	**INIT Coefficients command to Link Partner on Lane 2** 1: INIT Coefficients 0: Normal Operation This field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, useupdated_lp_coef_ln2=1 to transmit the values to the Link Partner. The INIT command is defined in IEEE 802.3 CL72.6.10.2.3.2.	RW	0x0
5:0	ld_coefficient_update_ln2	**TX EQ Coefficient Request to Link Partner on Lane 2** [5:4] Control for Link Partner (+1) TX EQ Coefficient [3:2] Control for Link Partner (0) TX EQ Coefficient [1:0] Control for Link Partner (-1) TX EQ Coefficient This field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln2=1 to transmit the values to the Link Partner. The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RW	0x0

B.1.31. Local Transceiver TX EQ 1 Settings for Lane 2

Provides the following Local TX EQ 1 Settings for Lane 2

- Local TX EQ VOD Setting for Lane 2
- Local TX EQ Post-Tap Setting for Lane 2
- Local TX EQ Pre-Tap Setting for Lane 2

Offset: 0xE6

Access: R0

Local Transceiver TX EQ 1 Settings for Lane 2 Fields

Bit	Name	Description	Access	Reset
20:16	lt_pretap_setting_ln2	**Local TX EQ Pre-tap Setting for Lane 2** This register returns the most recent Pre-tap setting that was written to the local transceiver.	RO	0x0
13:8	lt_posttap_setting_ln2	**Local TX EQ Post-tap Setting for Lane 2** This register returns the most recent Post-tap setting that was written to the local transceiver	RO	0x0
4:0	lt_vod_setting_ln2	**Local TX EQ VOD Setting for Lane 2** This register returns the most recent VOD setting that was written to the local transceiver	RO	0x0

B.1.32. Local Transceiver TX EQ 2 Settings for Lane 2

Provides the following Local TX EQ 2 Settings for Lane 2

- VMAXRULE Override value
- Enable VMAXRULE Override
- VODMINRULE Override value
- Enable VODMINRULE Override
- VPOSTRULE Override value
- Enable VPOSTRULE Override
- VPRERULE Override value
- Enable VPRERULE Override

Offset: 0xE7

Access: RW

Local Transceiver TX EQ 2 Settings for Lane 2 Fields

Bit	Name	Description	Access	Reset
29	lt_vpre_ovrd_en_ln2	**Enable VPRERULE Override for Lane 2** 1: Use the value of lt_vpre_ovrd to set VPRERULE 0: Use the value of VPRERULE set by the parameters that were used at compile time	RW	0x0
28:24	lt_vpre_ovrd_ln2	**VPRERULE Override value for Lane 2** When lt_vpre_ovrd_en=1, this CSR sets the maximum value of the Pre-tap on Lane 2. VPRERULE must be set to a value greater than INITPREVAL.	RW	0x0
22	lt_vpost_ovrd_en_ln2	**Enable VPOSTRULE Override for Lane 2** 1: Use the value of lt_vpost_ovrd to set VPOSTRULE. 0: Use the value of VPOSTRULE set by the parameters that were used at compile time.	RW	0x0
21:16	lt_vpost_ovrd_ln2	**VPOSTRULE Override value for Lane 2** When lt_vpost_ovrd_en=1, this CSR sets the maximum value of the Post-tap on Lane 2. VPOSTRULE must be set to a value greater than INITPOSTVAL.	RW	0x0

continued...

Bit	Name	Description	Access	Reset
13	lt_vodmin_ovrd_en_ln2	**Enable VODMINRULE Override for Lane 2** 1: Use the value of lt_vodmin_ovrd to set VODMINRULE 0: Use the value of VODMINRULE set by the parameters that were used at compile time	RW	0x0
12:8	lt_vodmin_ovrd_ln2	**VODMINRULE Override value for Lane 2** When lt_vodmin_ovrd_en=1, this CSR sets the minimum setting of VOD allowed during link training for Lane 2. VODMINRULE must be set to a value less than INITMAINVAL. VODMINRULE must also be set to a value greater than VMINRULE.	RW	0x0
5	lt_vodmax_ovrd_en_ln2	**Enable VMAXRULE Override for Lane 2** 1: Use the value of lt_vodmax_ovrd to set VMAXRULE 0: Use the value of VMAXRULE set by the parameters that were used at compile time	RW	0x0
4:0	lt_vodmax_ovrd_ln2	**VMAXRULE Override Value for Lane 2** When lt_vodmax_ovrd_en=1, this CSR sets the maximum Voltage allowed during link training for Lane 2. VMAXRULE must be set to a value greater than INITMAINVAL. Note that this value also changes PREMAINVAL.	RW	0x0

B.1.33. Link Training Config Register for Lane 3

Provides CSRs for the following link training features:

- LT PRBS Pattern Select for lane 3
- LT PRBS Seed for lane 3

Offset: 0xE8

Access: RW

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** 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** 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

B.1.34. Link Training Frame Contents for Lane 3

Provides CSRs for the following fields that are transmitted during link training to the link partner for Lane 3

- TX EQ Coefficient Request to Link Partner on Lane 3
- INIT Coefficients command to Link Partner on Lane 3
- PRESET Coefficients command to Link Partner on Lane 3
- Local TX EQ Coefficient Status for Lane 3
- Local Receiver Ready Status for Lane 3
- Most Recent TX EQ Coefficient Request from Link Partner on Lane 3
- Most Recent INIT command from Link Partner on Lane 3
- Most Recent PRESET command from Link Partner on Lane 3
- Most Recent TX EQ Status from Link Partner on Lane 3
- Most Recent Receiver Ready Status from Link Partner on Lane 3

Offset: 0xE9

Access: RO and RW

Link Training Frame Contents for Lane 3 Fields

Bit	Name	Description	Access	Reset
30	lp_receiver_ready_ln3	**Link Partner Receiver Ready Status for Lane 3** 1: The link partner receiver has determined that training is complete and is prepared to receive data 0: The link partner receiver is requesting that training continue	RO	0x0
29:24	lp_coefficient_status_ln3	**TX EQ Coefficient Status from Link Partner for Lane 3** [5:4] Status of Link Partner (+1) TX EQ Coefficient [3:2] Status of Link Partner (0) TX EQ Coefficient [1:0] Status of Link Partner (-1) TX EQ Coefficient The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RO	0x0
23	lp_preset_coefficients_ln3	**PRESET Command from Link Partner on Lane 3** 1: Set local TX EQ to PRESET 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln3=1 to write the local values.	RW	0x0
22	lp_initialize_coefficients_ln3	**INIT Command from Link Partner on Lane 3** 1: Set local TX EQ to INIT 0: Normal Operation This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln3=1 to write the local values.	RW	0x0

continued...

Bit	Name	Description	Access	Reset
21:16	lp_coefficient_update_ln3	**TX EQ Coefficient Request from Link Partner on Lane 3** [5:4] Control for Local (+1) TX EQ Coefficient [3:2] Control for Local (0) TX EQ Coefficient [1:0] Control for Local (-1) TX EQ Coefficient This Field is normally Read-only, and the values are normally controlled by the Remote Partner Link Training SM. When ovride_local_coef_enable=1, this Field becomes writable, and is used to set the local values. When ovride_local_coef_enable=1, use updated_local_coef_ln3=1 to write the local values. The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RW	0x0
14	ld_receiver_ready_ln3	**Local Receiver Ready Status for Lane 3** 1: The local device receiver has determined that training is complete and is prepared to receive data 0: The local device receiver is requesting that training continue	RO	0x0
13:8	ld_coefficient_status_ln3	**Local TX EQ Coefficient Status for Lane 3** [5:4] Status of Local (+1) TX EQ Coefficient [3:2] Status of Local (0) TX EQ Coefficient [1:0] Status of Local (-1) TX EQ Coefficient The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RO	0x0
7	ld_preset_coefficients_ln3	**PRESET Coefficients command to Link Partner on Lane 3** 1: PRESET Coefficients 0: Normal Operation This field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, useupdated_lp_coef_ln3=1 to transmit the values to the Link Partner. The PRESET command is defined in IEEE 802.3 CL72.6.10.2.3.1.	RW	0x0
6	ld_initialize_coefficients_ln3	**INIT Coefficients command to Link Partner on Lane 3** 1: INIT Coefficients 0: Normal Operation This field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, useupdated_lp_coef_ln3=1 to transmit the values to the Link Partner.	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		The INIT command is defined in IEEE 802.3 CL72.6.10.2.3.2.		
5:0	ld_coefficient_update_ln3	**TX EQ Coefficient Request to Link Partner on Lane 3** [5:4] Control for Link Partner (+1) TX EQ Coefficient [3:2] Control for Link Partner (0) TX EQ Coefficient [1:0] Control for Link Partner (-1) TX EQ Coefficient This field is normally Read-only, and the values are normally controlled by the Link Training SM. When ovride_lp_coef_enable=1, this field becomes writable, and is used to set the values sent to the Link Partner. When ovride_lp_coef_enable=1, use updated_lp_coef_ln3=1 to transmit the values to the Link Partner. The Coefficient values are encoded as follows: • 2'b00: Hold • 2'b01: Increment • 2'b10: Decrement • 2'b11: Reserved	RW	0x0

B.1.35. Local Transceiver TX EQ 1 Settings for Lane 3

Provides the following Local TX EQ 1 Settings for Lane 3

- Local TX EQ VOD Setting for Lane 3
- Local TX EQ Post-Tap Setting for Lane 3
- Local TX EQ Pre-Tap Setting for Lane 3

Offset: 0xEA

Access: RO

Local Transceiver TX EQ 1 Settings for Lane 3 Fields

Bit	Name	Description	Access	Reset
20:16	lt_pretap_setting_ln3	**Local TX EQ Pre-tap Setting for Lane 3** This register returns the most recent Pre-tap setting that was written to the local transceiver.	RO	0x0
13:8	lt_posttap_setting_ln3	**Local TX EQ Post-tap Setting for Lane 3** This register returns the most recent Post-tap setting that was written to the local transceiver	RO	0x0
4:0	lt_vod_setting_ln3	**Local TX EQ VOD Setting for Lane 3** This register returns the most recent VOD setting that was written to the local transceiver	RO	0x0

B.1.36. Local Transceiver TX EQ 2 Settings for Lane 3

Provides the following Local TX EQ 2 Settings for Lane 3

- VMAXRULE Override value
- Enable VMAXRULE Override
- VODMINRULE Override value
- Enable VODMINRULE Override

- VPOSTRULE Override value
- Enable VPOSTRULE Override
- VPRERULE Override value
- Enable VPRERULE Override

Offset: 0xEB

Access: RW

Local Transceiver TX EQ 2 Settings for Lane 3 Fields

Bit	Name	Description	Access	Reset
29	lt_vpre_ovrd_en_ln3	**Enable VPRERULE Override for Lane 3** 1: Use the value of lt_vpre_ovrd to set VPRERULE 0: Use the value of VPRERULE set by the parameters that were used at compile time	RW	0x0
28:24	lt_vpre_ovrd_ln3	**VPRERULE Override value for Lane 3** When lt_vpre_ovrd_en=1, this CSR sets the maximum value of the Pre-tap on Lane 3. VPRERULE must be set to a value greater than INITPREVAL.	RW	0x0
22	lt_vpost_ovrd_en_ln3	**Enable VPOSTRULE Override for Lane 3** 1: Use the value of lt_vpost_ovrd to set VPOSTRULE. 0: Use the value of VPOSTRULE set by the parameters that were used at compile time.	RW	0x0
21:16	lt_vpost_ovrd_ln3	**VPOSTRULE Override value for Lane 3** When lt_vpost_ovrd_en=1, this CSR sets the maximum value of the Post-tap on Lane 3. VPOSTRULE must be set to a value greater than INITPOSTVAL.	RW	0x0
13	lt_vodmin_ovrd_en_ln3	**Enable VODMINRULE Override for Lane 3** 1: Use the value of lt_vodmin_ovrd to set VODMINRULE 0: Use the value of VODMINRULE set by the parameters that were used at compile time	RW	0x0
12:8	lt_vodmin_ovrd_ln3	**VODMINRULE Override value for Lane 3** When lt_vodmin_ovrd_en=1, this CSR sets the minimum setting of VOD allowed during link training for Lane 3. VODMINRULE must be set to a value less than INITMAINVAL. VODMINRULE must also be set to a value greater than VMINRULE.	RW	0x0
5	lt_vodmax_ovrd_en_ln3	**Enable VMAXRULE Override for Lane 3** 1: Use the value of lt_vodmax_ovrd to set VMAXRULE 0: Use the value of VMAXRULE set by the parameters that were used at compile time	RW	0x0
4:0	lt_vodmax_ovrd_ln3	**VMAXRULE Override Value for Lane 3** When lt_vodmax_ovrd_en=1, this CSR sets the maximum voltage allowed during link training for Lane 3. VMAXRULE must be set to a value greater than INITMAINVAL. Note that this value also changes PREMAINVAL.	RW	0x0

B.2. PHY Registers

B.2.1. PHY Module Revision ID

Returns a 4 byte value indicating the revision of this design.

Offset: 0x300

Access: RO

PHY Module Revision ID Fields

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module.	RO	0x11112015

B.2.2. PHY Scratch Register

32 bits of scratch register space for testing.

Offset: 0x301

Access: RW

PHY Scratch Register Fields

Bit	Name	Description	Access	Reset
31:0	scratch		RW	0x0

B.2.3. PHY Configuration

Offset: 0x310

PHY Configuration Fields

The reset values in this table represents register values after a reset has completed.

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

B.2.4. PMA Serial Loopback

Offset: 0x313

PMA Serial Loopback Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
3:0	sloop	**Activate Serial Loopback** 1: Force corresponding physical lane to receive serial data from its own transmitter instead of from its RX serial pins	RW	0x0

B.2.5. TX PLL Locked

Offset: 0x320

TX PLL Locked Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
3:0	tx_pll_locked	**TX PLL Locked** 1: TX PLL used by this physical lane is locked.	RO	0x0

B.2.6. RX CDR PLL Locked

Offset: 0x321

RX CDR PLL Locked Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
3:0	eio_freq_lock	**CDR PLL locked** 1: Corresponding physical lane's CDR has locked to reference for 10, 25, and 100G links.	RO	0x0

B.2.7. TX Datapath Ready

Offset: 0x322

TX Datapath Ready Fields

The reset values in this table represents register values after a reset has completed.

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

B.2.8. Frame Errors Detected

Frame error(s) detected.

Offset: 0x323

Access: RO

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 50G links, only bits 3:0 are used, corresponding to Virtual lanes 0 to 3 • 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

B.2.9. Clear Frame Errors

Offset: 0x324

Clear Frame Errors Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
0	clr_frmerr	**Clear PHY frame error(s).** 1: Return all sticky frame error bits to 0.	RW	0x0

B.2.10. Reset Registers

Reset bits:

- TX MAC reset
- TX PCS reset
- RX MAC reset
- RX PCS reset

Offset: 0x325

Access: RW

Reset Register Fields

Bit	Name	Description	Access	Reset
19	rx_pcs_in_rst	**Reset RX PCS** • 1: Reset RX PCS • Defaults to 0 after power up and i_csr_rst_n	RW	0x1
18	rx_mac_in_rst	**Reset RX MAC** • 1: Reset RX MAC • Does not reset RX MAC statistics • Defaults to 0 after power up and i_csr_rst_n	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	RW	0x1
16	tx_mac_in_rst	**Reset TX MAC** • 1: Reset TX MAC • Does not reset TX MAC statistics • Defaults to 0 after power up and i_csr_rst_n	RW	0x1

B.2.11. RX PCS Status for AN/LT

Offset: 0x326

RX PCS Status for AN/LT Fields

The reset values in this table represents register values after a reset has completed.

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	RO	0x0

B.2.12. PCS Error Injection

Offset: 0x327

PCS Error Injection Fields

The reset values in this table represents register values after a reset has completed.

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

B.2.13. Alignment Marker Lock

Status register to indicate that alignment lock has been achieved.

Offset: 0x328

Access: R0

Alignment Marker Lock Fields

Bit	Name	Description	Access	Reset
0	am_lock	**AM Lock** 1: RX PCS has achieved Alignment Marker lock	RO	0x0

B.2.14. BER Count

Offset: 0x32A

ber_count Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	count	**BER Count**	RO	0x0

Bit	Name	Description	Access	Reset
		• 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		

B.2.15. PCS Virtual Lane 0

Offset: 0x330

PCS Virtual Lane 0 Fields

The reset values in this table represents register values after a reset has completed.

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			

B.2.16. PCS Virtual Lane 1

Offset: 0x331

PCS Virtual Lane 1 Fields

The reset values in this table represents register values after a reset has completed.

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			

B.2.17. PCS Virtual Lane 2

Offset: 0x332

PCS Virtual Lane 2 Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
29:25	vlane17	**Virtual lane mapping**	RO	0x1F

continued...

Bit	Name	Description	Access	Reset
24:20	vlane16	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		
19:15	vlane15			
14:10	vlane14			
9:5	vlane13			
4:0	vlane12			

B.2.18. PCS Virtual Lane 3

Offset: 0x333

PCS Virtual Lane 3 Fields

The reset values in this table represents register values after a reset has completed.

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			

B.2.19. Recovered Clock Frequency in KHz

Offset: 0x341

Recovered Clock Frequency in KHz Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	khz_rx	**Recovered clock frequency** Recovered clock frequency/100, in KHz.	RO	0x0

B.2.20. TX Clock Frequency in KHz

Offset: 0x342

TX Clock Frequency in KHz Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	khz_tx	**TX clock frequency** TX clock frequency/100, in KHz.	RO	0x0

B.2.21. Programmable Alignment Marker 0

Each of these registers defines the first 24 bits of the Alignment Marker Encoding for 1 of the 4 alignment markers used in 50G links. The Ethernet Standard (IEEE 802.3) defines the alignment markers required for 50G - these are the default values for the registers. The 24b value is combined with an 8b BIP value, and the combined 32b value is inverted and appended to make a 64b alignment marker

- AM0: 24'h907647
- AM1: 24'hF0C4E6
- AM2: 24'hC5659B
- AM3: 24'hA2793D

It is safe to change this value while the TX and RX PCS are operating, but 50G channels in operation while the change is made may lose alignment lock.

Offset: 0x376

Access: RW

Programmable Alignment Marker Fields

Bit	Name	Description	Access	Reset
23:0	am	**24b alignment marker encoding** This code and its inverse are combined with a BIP value to create an alignment marked for the corresponding PCS Virtual lane • After power-on, the reset value of this register is 0 • After configuration, asserting i_csr_rst_n causes the register to return to the value set by the am_encoding40g_*n* module parameter where *n* is the alignment marker index	RW	0x907647

B.2.22. Programmable Alignment Marker 1

Each of these registers defines the first 24 bits of the Alignment Marker Encoding for 1 of the 4 alignment markers used in 50G links. The Ethernet Standard (IEEE 802.3) defines the alignment markers required for 50G - these are the default values for the registers. The 24b value is combined with an 8b BIP value, and the combined 32b value is inverted and appended to make a 64b alignment marker

- AM0: 24'h907647
- AM1: 24'hF0C4E6
- AM2: 24'hC5659B
- AM3: 24'hA2793D

It is safe to change this value while the TX and RX PCS are operating, but 50G channels in operation while the change is made may lose alignment lock.

Offset: 0x377

Access: RW

Programmable Alignment Marker 1 Fields

Bit	Name	Description	Access	Reset
23:0	am	**24b alignment marker encoding** This code and its inverse are combined with a BIP value to create an alignment marked for the corresponding PCS Virtual lane • After power-on, the reset value of this register is 0 • After configuration, asserting i_csr_rst_n causes the register to return to the value set by the am_encoding40g_*n* module parameter where *n* is the alignment marker index	RW	0xF0C4E6

B.2.23. Programmable Alignment Marker 2

Each of these registers defines the first 24 bits of the Alignment Marker Encoding for 1 of the 4 alignment markers used in 50G links. The Ethernet Standard (IEEE 802.3) defines the alignment markers required for 50G - these are the default values for the registers. The 24b value is combined with an 8b BIP value, and the combined 32b value is inverted and appended to make a 64b alignment marker

- AM0: 24'h907647
- AM1: 24'hF0C4E6
- AM2: 24'hC5659B
- AM3: 24'hA2793D

It is safe to change this value while the TX and RX PCS are operating, but 50G channels in operation while the change is made may lose alignment lock.

Offset: 0x378

Access: RW

Programmable Alignment Marker 2 Fields

Bit	Name	Description	Access	Reset
23:0	am	**24b alignment marker encoding** This code and its inverse are combined with a BIP value to create an alignment marked for the corresponding PCS Virtual lane • After power-on, the reset value of this register is 0 • After configuration, asserting i_csr_rst_n causes the register to return to the value set by the am_encoding40g_*n* module parameter where *n* is the alignment marker index	RW	0xC5659B

B.2.24. Programmable Alignment Marker 3

Each of these registers defines the first 24 bits of the Alignment Marker Encoding for 1 of the 4 alignment markers used in50G links. The Ethernet Standard (IEEE 802.3) defines the alignment markers required for 50G - these are the default values for the registers. The 24b value is combined with an 8b BIP value, and the combined 32b value is inverted and appended to make a 64b alignment marker

- AM0: 24'h907647
- AM1: 24'hF0C4E6
- AM2: 24'hC5659B
- AM3: 24'hA2793D

It is safe to change this value while the TX and RX PCS are operating, but 50G channels in operation while the change is made may lose alignment lock.

Offset: 0x379

Access: RW

Programmable Alignment Marker 3 Fields

Bit	Name	Description	Access	Reset
23:0	am	**24b alignment marker encoding** This code and its inverse are combined with a BIP value to create an alignment marked for the corresponding PCS Virtual lane • After power-on, the reset value of this register is 0 • After configuration, asserting i_csr_rst_n causes the register to return to the value set by the am_encoding40g_*n* module parameter where *n* is the alignment marker index	RW	0xA2793D

B.3. TX MAC Registers

B.3.1. TX MAC Module Revision ID

Offset: 0x400

TX MAC Module Revision ID Fields

The reset values in this table represents register values after a reset has completed.

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

B.3.2. TX MAC Scratch Register

Offset: 0x401

TX MAC Scratch Register Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	scratch	32 bits of scratch register space for testing	RW	0x0

B.3.3. Reserved

Offset: 0x402

Reserved Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id	Returns 0, override with soft logic to indicate specific core name	RO	0x0

Offset: 0x403

Reserved Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

Offset: 0x404

Reserved Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.3.4. Link Fault Configuration

Offset: 0x405

Link Fault Configuration Fields

The reset values in this table represents register values after a reset has completed.

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	RW	0x1

continued...

Bit	Name	Description	Access	Reset
		• 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		

B.3.5. IPG Words to remove per Alignment Marker Period

Offset: 0x406

IPG Words to remove per Alignment Marker Period Fields

The reset values in this table represents register values after a reset has completed.

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

B.3.6. Maximum TX Frame Size

Offset: 0x407

Maximum TX Frame Size Fields

The reset values in this table represents register values after a reset has completed.

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

B.3.7. TX MAC Configuration

Offset: 0x40A

TX MAC Configuration Fields

The reset values in this table represents register values after a reset has completed.

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	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		• 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		
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

B.3.8. EHIP TX MAC Feature Configuration

EHIP TX MAC Feature Configuration

Register for configuring EHIP specific TX MAC features

Offset: 0x40B

Access: RW

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 • 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	RW	0x13FFB
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: 50G	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	RW	0x5

continued...

Bit	Name	Description	Access	Reset
		• 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 1 for all other types of channels		
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

B.3.9. TX MAC Source Address Lower Bytes

Offset: 0x40C

TX MAC Source Address Lower Bytes Fields

The reset values in this table represents register values after a reset has completed.

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

B.3.10. TX MAC Source Address Higher Bytes

Offset: 0x40D

TX MAC Source Address Higher Bytes Fields

The reset values in this table represents register values after a reset has completed.

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

B.4. RX MAC Registers

B.4.1. RX MAC Module Revision ID

Offset: 0x500

RX MAC Module Revision ID Fields

The reset values in this table represents register values after a reset has completed.

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

B.4.2. RX MAC Scratch Register

Offset: 0x501

RX MAC Scratch Register Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	scratch	32 bits of scratch register space for testing	RW	0x0

B.4.3. Reserved

Offset: 0x502

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id	Returns 0, override with soft logic to indicate specific core name	RO	0x0

Offset: 0x503

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

Offset: 0x504

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.4.4. Maximum RX Frame Size

Offset: 0x506

Maximum RX Frame Size Fields

The reset values in this table represents register values after a reset has completed.

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

B.4.5. RX CRC Forwarding

Offset: 0x507

RX CRC Forwarding Fields

The reset values in this table represents register values after a reset has completed.

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

B.4.6. Link Fault Status

Offset: 0x508

Link Fault Status Fields

The reset values in this table represents register values after a reset has completed.

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

B.4.7. RX MAC Configuration

Offset: 0x50A

RX MAC Configuration Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
8	remove_rx_pad	**Remove PADs from padded frames**	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		0: Padded frames are not altered 1: Pads are removed from padded frames • 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

B.4.8. EHIP RX MAC Feature Configuration

Offset: 0x50B

EHIP RX MAC Feature Configuration Fields

The reset values in this table represents register values after a reset has completed.

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

B.5. Pause and Priority- Based Flow Control Registers

B.5.1. TXSFC Module Revision ID

Offset: 0x600

TXSFC Module Revision ID Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.2. TX SFC Scratch Register

Offset: 0x601

TX SFC Scratch Register Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	scratch	32 bits of scratch register space for testing	RW	0x0

B.5.3. Reserved

Offset: 0x602

txsfc_name_0 Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id	Returns 0, override with soft logic to indicate specific core name	RO	0x0

Offset: 0x603

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

Offset: 0x604

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.5.4. Enable TX Pause Ports

Offset: 0x605

Enable TX Pause Ports Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.5. TX Pause Request

Offset: 0x606

TX Pause Request Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.6. Enable Automatic TX Pause Retransmission

Offset: 0x607

Enable Automatic TX Pause Retransmission Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
8:0	en_holdoff	**Enable Holdoff timer.**	RW	0x1

Bit	Name	Description	Access	Reset
		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`		

B.5.7. Retransmit Holdoff Quanta

Retransmit Holdoff Quanta

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register bit is 1

Offset: `0x608`

Access: `RW`

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 • Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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`

B.5.8. Retransmit Pause Quanta

Retransmit Pause Quanta

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x609

Access: RW

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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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

B.5.9. Enable TX XOFF

Offset: 0x60A

Enable TX XOFF Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.10. Enable Uniform Holdoff

Offset: 0x60B

Enable Uniform Holdoff Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.11. Set Uniform Holdoff

Set uniform holdoff

16b value specifying the minimum holdoff time for all PFC queues when Enable Uniform Holdoff register=1

Offset: 0x60C

Access: RW

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 when cfg_retransmit_holdoff_en.en_holdoff_all=1. • Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: min(Pause Quanta register value) - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*min(Pause Quanta register value) - (15 + Maximum TX Frame Size register value/32) — 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	RW	0x0

B.5.12. Lower 4 bytes of the Destination address for Flow Control

Offset: 0x60D

Lower 4 bytes of the Destination address for Flow Control Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.13. Higher 2 bytes of the Destination address for Flow Control

Offset: 0x60E

Higher 2 bytes of the Destination address for Flow Control Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.14. 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

The reset values in this table represents register values after a reset has completed.

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

B.5.15. 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

The reset values in this table represents register values after a reset has completed.

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	RW	0xE100

Bit	Name	Description	Access	Reset
		• 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]		

B.5.16. TX Flow Control Feature Configuration

Offset: 0x611

txsfc_ehip_cfg Fields

The reset values in this table represents register values after a reset has completed.

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) • 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	RW	0x0

B.5.17. Pause Quanta 0

Pause Quanta 0

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x620

Access: RW

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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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

B.5.18. Pause Quanta 1

Pause Quanta 1

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x621

Access: RW

Pause Quanta 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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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

B.5.19. Pause Quanta 2

Pause Quanta 2

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x622

Access: RW

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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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

B.5.20. Pause Quanta 3

Pause Quanta 3

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x623

Access: RW

Pause Quanta 3 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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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		

B.5.21. Pause Quanta 4

Pause Quanta 4

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: `0x624`

Access: `RW`

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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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`

B.5.22. Pause Quanta 5

Pause Quanta 5

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x625

Access: RW

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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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

B.5.23. Pause Quanta 6

Pause Quanta 6

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x626

Access: RW

ehippfc_pause_quanta_6.xml 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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 100Gx4 link, 512 bit times is 2 valid clock cycles • Minimum allowed value: 1 • Maximum value: 16'hFFFF	RW	0xFFFF

Bit	Name	Description	Access	Reset
		• 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		

B.5.24. Pause Quanta 7

Pause Quanta 7

16b value specifying the pause quanta that will be transmitted in each XOFF frame sent

Offset: 0x627

Access: RW

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 50Gx2 link, 512 bit times is 4 valid clock cycles — On a 50Gx4, or 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

B.5.25. PFC Holdoff Quanta 0

PFC Holdoff Quanta 0

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register value bit is 1

Offset: 0x628

Access: RW

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 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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

B.5.26. PFC Holdoff Quanta 1

PFC Holdoff Quanta 1

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register value bit is 1

Offset: 0x629

Access: RW

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 value bit is 1	RW	0xFFFF

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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		

B.5.27. PFC Holdoff Quanta 2

PFC Holdoff Quanta 2

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding PFC Holdoff Quanta 2 bit is 1

Offset: `0x62A`

Access: `RW`

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 value bit is 1	`RW`	`0xFFFF`

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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		

B.5.28. PFC Holdoff Quanta 3

PFC Holdoff Quanta 3

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register value bit is 1

Offset: `0x62B`

Access: `RW`

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 value bit is 1	`RW`	`0xFFFF`

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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		

B.5.29. PFC Holdoff Quanta 4

PFC Holdoff Quanta 4

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register value bit is 1

Offset: `0x62C`

Access: `RW`

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 value bit is 1	`RW`	`0xFFFF`

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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		

B.5.30. PFC Holdoff Quanta 5

PFC Holdoff Quanta 5

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register value bit is 1

Offset: `0x62D`

Access: `RW`

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 value bit is 1	`RW`	`0xFFFF`

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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		

B.5.31. PFC Holdoff Quanta 6

PFC Holdoff Quanta 6

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register value bit is 1

Offset: `0x62E`

Access: `RW`

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 value bit is 1	`RW`	`0xFFFF`

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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		

B.5.32. PFC Holdoff Quanta 7

PFC Holdoff Quanta 7

16b value specifying the holdoff time before XOFF is retransmitted when the corresponding Enable Automatic TX Pause Retransmission register value bit is 1

Offset: `0x62F`

Access: `RW`

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 value bit is 1	`RW`	`0xFFFF`

Bit	Name	Description	Access	Reset
		• Times are programmed in holdoff quanta — For 50Gx4, and 100G links, 1 Holdoff Quanta = 2 clock cycles — For 50Gx2, 1 Holdoff Quanta = 2 clock cycles - this is different from PAUSE/PFC Quanta, which would have been 4 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 50Gx4 links: corresponding pause/pfc quanta - (15 + Maximum TX Frame Size register value/32) — For 50Gx2 links: 2*(corresponding pause/pfc quanta) - (15 + Maximum TX Frame Size register value/32) — 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		

B.5.33. RXSFC Module Revision ID

Offset: `0x700`

RXSFC Module Revision ID Fields

The reset values in this table represents register values after a reset has completed.

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`

B.5.34. RXSFC Scratch Register

Offset: `0x701`

RXSFC Scratch Register Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`scratch`	32 bits of scratch register space for testing	RW	`0x0`

B.5.35. Reserved

Offset: `0x702`

rxsfc_name_0 Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id	Returns 0, override with soft logic to indicate specific core name	RO	0x0

Offset: 0x703

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

Offset: 0x704

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.5.36. Enable RX Pause Frame Processing

Offset: 0x705

Enable RX Pause Frame Processing Fields

The reset values in this table represents register values after a reset has completed.

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

B.5.37. Forward Flow Control Frames

Offset: 0x706

Forward Flow Control Frames Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
0	rx_pause_fwd	**Forward Flow Control Frames** Sets whether PAUSE and PFC frames are send 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	RW	0x0

Bit	Name	Description	Access	Reset
		• 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`		

B.5.38. 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

The reset values in this table represents register values after a reset has completed.

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. • 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'hC2000001 • When `i_csr_rst_n` is asserted, this register is set to the value given by the module parameter `rx_pause_daddr[31:0]`	RW	`0xC2000001`

B.5.39. 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

The reset values in this table represents register values after a reset has completed.

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	RW	`0x180`

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 16'h0180 • When `i_csr_rst_n` is asserted, this register is set to the value given by the module parameter `rx_pause_daddr[47:32]`		

B.6. TX Statistics Counter Registers

B.6.1. TX Frames less than 64 bytes with CRC error (lower 32 bits)

Offset: `0x800`

TX Frames less than 64 bytes with CRC error Fields (lower 32 bits)

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt6`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics Number of frames with less than 64 bytes that are malformed or have fewer than 18 bytes CRC is not checked, it is assumed correct at TX	RO	`0x0`

B.6.2. TX Frames less than 64 bytes with CRC error (upper 32 bits)

Offset: `0x801`

TX Frames less than 64 bytes with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt6`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics Number of frames with less than 64 bytes that are malformed or have fewer than 18 bytes CRC is not checked, it is assumed correct at TX	RO	`0x0`

B.6.3. Oversized TX frames with CRC error (lower 32 bits)

Offset: `0x802`

Oversized TX frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt2`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics Number of frames with more octets than the programmed max that are malformed or have fewer than 18 bytes CRC is not checked, it is assumed correct at TX	RO	`0x0`

B.6.4. Oversized TX frames with CRC error (upper 32 bits)

Offset: 0x803

Oversized TX frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics Number of frames with more octets than the programmed max that are malformed or have fewer than 18 bytes CRC is not checked, it is assumed correct at TX	RO	0x0

B.6.5. TX Frames of any size with a CRC error (lower 32 bits)

TX Frames of any size with a CRC error

Number of frames of any size that are malformed or have fewer than 18 bytes CRC is not checked, it is assumed correct at TX

Offset: 0x804

TX Frames of any size with a CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.6. TX Frames of any size with a CRC error (upper 32 bits)

TX Frames of any size with a CRC error

Number of frames of any size that are malformed or have fewer than 18 bytes CRC is not checked, it is assumed correct at TX

Offset: 0x805

TX Frames of any size with a CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.7. TX Frames of any size with a CRC error on OK packet (lower 32 bits)

TX Frames of any size with a CRC error on OK packet

Number of frames of any size that are malformed but are neither undersized or oversized CRC is not checked, it is assumed correct at TX

Offset: 0x806

TX Frames of any size with a CRC error on OK packet (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.8. TX Frames of any size with a CRC error on OK packet (upper 32 bits)

TX Frames of any size with a CRC error on OK packet

Number of frames of any size that are malformed but are neither undersized or oversized CRC is not checked, it is assumed correct at TX

Offset: 0x807

TX Frames of any size with a CRC error on OK packet (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.9. Multicast TX data frames with CRC error (lower 32 bits)

Multicast TX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a multicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x808

Multicast TX data frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.10. Multicast TX data frames with CRC error (upper 32 bits)

Multicast TX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a multicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x809

Multicast TX data frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.11. Broadcast TX data frames with CRC error (lower 32 bits)

Broadcast TX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a broadcast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x80A

Broadcast TX data frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.12. Broadcast TX data frames with CRC error (upper 32 bits)

Broadcast TX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a broadcast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x80B

Broadcast TX data frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.13. Unicast TX data frames with CRC error (lower 32 bits)

Unicast TX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a unicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x80C

Unicast TX data frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.14. Unicast TX data frames with CRC error (upper 32 bits)

Unicast TX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a unicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x80D

Unicast TX data frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.15. Multicast TX control frames with CRC error (lower 32 bits)

Multicast TX control frames with CRC error

Number of frames where L/T = 16'h8808 with a multicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x80E

Multicast TX control frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.16. Multicast TX control frames with CRC error (upper 32 bits)

Multicast TX control frames with CRC error

Number of frames where L/T = 16'h8808 with a multicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x80F

Multicast TX control frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word**	RO	0x0

Bit	Name	Description	Access	Reset
		4 bytes of an 8 byte EHIP Statistics		

B.6.17. Broadcast TX control frames with CRC error (lower 32 bits)

Broadcast TX control frames with CRC error

Number of frames where L/T = 16'h8808 with a broadcast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: `0x810`

Broadcast TX control frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt6`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	`0x0`

B.6.18. Broadcast TX control frames with CRC error (upper 32 bits)

Broadcast TX control frames with CRC error

Number of frames where L/T = 16'h8808 with a broadcast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: `0x811`

Broadcast TX control frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt6`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	`0x0`

B.6.19. Unicast TX control frames with CRC error (lower 32 bits)

Unicast TX control frames with CRC error

Number of frames where L/T = 16'h8808 with a unicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: `0x812`

Unicast TX control frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt6`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	`0x0`

B.6.20. Unicast TX control frames with CRC error (upper 32 bits)

Unicast TX control frames with CRC error

Number of frames where L/T = 16'h8808 with a unicast destination address that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x813

Unicast TX control frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.21. TX Pause frame with CRC error (lower 32 bits)

TX Pause frame with CRC error

Number of PAUSE (Standard Flow Control) frames that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x814

TX Pause frame with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.22. TX Pause frame with CRC error (upper 32 bits)

TX Pause frame with CRC error

Number of PAUSE (Standard Flow Control) frames that are malformed CRC is not checked, it is assumed correct at TX

Offset: 0x815

TX Pause frame with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.23. 64 byte TX frames (lower 32 bits)

64 byte TX frames

Number of frames of any type with a frame length of 64 bytes

Offset: 0x816

64 byte TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.24. 64 byte TX frames (upper 32 bits)

64 byte TX frames

Number of frames of any type with a frame length of 64 bytes

Offset: 0x817

64 byte TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.25. 65 to 127 byte TX frames (lower 32 bits)

65 to 127 byte TX frames

Number of frames of any type with a frame length of 65 to 127 bytes

Offset: 0x818

65 to 127 byte TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.26. 65 to 127 byte TX frames (upper 32 bits)

65 to 127 byte TX frames

Number of frames of any type with a frame length of 65 to 127 bytes

Offset: 0x819

65 to 127 byte TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.27. 128 to 257 byte TX frames (lower 32 bits)

128 to 257 byte TX frames

Number of frames of any type with a frame length of 128 to 257 bytes

Offset: 0x81A

128 to 257 byte TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.28. 128 to 257 byte TX frames (upper 32 bits)

128 to 257 byte TX frames

Number of frames of any type with a frame length of 128 to 257 bytes

Offset: 0x81B

128 to 257 byte TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.29. 256 to 511 byte TX frames (lower 32 bits)

256 to 511 byte TX frames

Number of frames of any type with a frame length of 258 to 511 bytes

Offset: 0x81C

256 to 511 byte TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.30. 256 to 511 byte TX frames (upper 32 bits)

256 to 511 byte TX frames

Number of frames of any type with a frame length of 258 to 511 bytes

Offset: 0x81D

256 to 511 byte TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.31. 512 to 1023 byte TX frames (lower 32 bits)

512 to 1023 byte TX frames

Number of frames of any type with a frame length of 512 to 1023 bytes

Offset: 0x81E

512 to 1023 byte TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.32. 512 to 1023 byte TX frames (upper 32 bits)

512 to 1023 byte TX frames

Number of frames of any type with a frame length of 512 to 1023 bytes

Offset: 0x81F

512 to 1023 byte TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.33. 1024 to 1518 byte TX frames (lower 32 bits)

1024 to 1518 byte TX frames

Number of frames of any type with a frame length of 1024 to 1518 bytes

Offset: 0x820

1024 to 1518 byte TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt3	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.34. 1024 to 1518 byte TX frames (upper 32 bits)

1024 to 1518 byte TX frames

Number of frames of any type with a frame length of 1024 to 1518 bytes

Offset: 0x821

1024 to 1518 byte TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt3	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.35. 1519 to max size TX frames (lower 32 bits)

1519 to max size TX frames

Number of frames of any type with a frame length of 1519 to the programmed maximum size in bytes

Offset: 0x822

1519 to max size TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.36. 1519 to max size TX frames (upper 32 bits)

1519 to max size TX frames

Number of frames of any type with a frame length of 1519 to the programmed maximum size in bytes

Offset: 0x823

1519 to max size TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.37. Oversize TX frames (lower 32 bits)

Oversize TX frames

Number of frames of any type with a frame length greater than the programmed maximum size in bytes

Offset: 0x824

Oversize TX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.38. Oversize TX frames (upper 32 bits)

Oversize TX frames

Number of frames of any type with a frame length greater than the programmed maximum size in bytes

Offset: 0x825

Oversize TX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.39. Multicast TX data frames without error (lower 32 bits)

Multicast TX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a multicast destination address and no errors

Offset: 0x826

Multicast TX data frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.40. Multicast TX data frames without error (upper 32 bits)

Multicast TX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a multicast destination address and no errors

Offset: 0x827

Multicast TX data frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.41. Broadcast TX data frames without error (lower 32 bits)

Broadcast TX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a broadcast destination address and no errors

Offset: 0x828

Broadcast TX data frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.42. Broadcast TX data frames without error (upper 32 bits)

Broadcast TX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a broadcast destination address and no errors

Offset: 0x829

Broadcast TX data frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.43. Unicast TX data frames without error (lower 32 bits)

Unicast TX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a unicast destination address and no errors

Offset: 0x82A

Unicast TX data frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.44. Unicast TX data frames without error (upper 32 bits)

Unicast TX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808 with a unicast destination address and no errors

Offset: 0x82B

Unicast TX data frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.45. Multicast TX control frames without error (lower 32 bits)

Multicast TX control frames without error

Number of frames where L/T is 16'h8808 with a multicast destination address and no errors

Offset: 0x82C

Multicast TX control frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.46. Multicast TX control frames without error (upper 32 bits)

Multicast TX control frames without error

Number of frames where L/T is 16'h8808 with a multicast destination address and no errors

Offset: 0x82D

Multicast TX control frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.47. Broadcast TX control frames without error (lower 32 bits)

Broadcast TX control frames without error

Number of frames where L/T is 16'h8808 with a broadcast destination address and no errors

Offset: 0x82E

Broadcast TX control frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.48. Broadcast TX control frames without error (upper 32 bits)

Broadcast TX control frames without error

Number of frames where L/T is 16'h8808 with a broadcast destination address and no errors

Offset: 0x82F

Broadcast TX control frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.49. Unicast TX control frames without error (lower 32 bits)

Unicast TX control frames without error

Number of frames where L/T is 16'h8808 with unicast destination address and no errors

Offset: 0x830

Unicast TX control frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.50. Unicast TX control frames without error (upper 32 bits)

Unicast TX control frames without error

Number of frames where L/T is 16'h8808 with unicast destination address and no errors

Offset: 0x831

Unicast TX control frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.51. TX Pause frames without error (lower 32 bits)

TX Pause frames without error

Number of PAUSE (Standard Flow Control) frames without error

Offset: 0x832

TX Pause frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.52. TX Pause frames without error (upper 32 bits)

TX Pause frames without error

Number of PAUSE (Standard Flow Control) frames without error

Offset: 0x833

TX Pause frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.53. TX Frames with less than 64 bytes and a CRC error (lower 32 bits)

TX Frames with less than 64 bytes and a CRC error

Number of Frames with less than 64 bytes

Offset: 0x834

TX Frames with less than 64 bytes and a CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.54. TX Frames with less than 64 bytes and a CRC error (upper 32 bits)

TX Frames with less than 64 bytes and a CRC error

Number of Frames with less than 64 bytes

Offset: 0x835

TX Frames with less than 64 bytes and a CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.55. Number of TX frame starts (lower 32 bits)

Number of TX frame starts

Number of frame starts

Offset: 0x836

Number of TX frame starts (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.56. Number of TX frame starts (upper 32 bits)

Number of TX frame starts

Number of frame starts

Offset: 0x837

Number of TX frame starts (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.57. Number of TX length errors (lower 32 bits)

Number of TX length errors

Number of frames where the length of the frame advertised in the L/T field was larger than the frame that was received. Length checking must be enabled

Offset: 0x838

Number of TX length errors (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.58. Number of TX length errors (upper 32 bits)

Number of TX length errors

Number of frames where the length of the frame advertised in the L/T field was larger than the frame that was received. Length checking must be enabled

Offset: 0x839

Number of TX length errors (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.59. TX PFC frame with CRC error (lower 32 bits)

TX PFC frame with CRC error

Number of PFC frames that were malformed CRC is not checked, it is assumed correct at TX

Offset: 0x83A

TX PFC frame with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.60. TX PFC frame with CRC error (upper 32 bits)

TX PFC frame with CRC error

Number of PFC frames that were malformed CRC is not checked, it is assumed correct at TX

Offset: 0x83B

TX PFC frame with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.61. TX PFC frames without error (lower 32 bits)

TX PFC frames without error

Number of PFC frames without error

Offset: 0x83C

TX PFC frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.62. TX PFC frames without error (upper 32 bits)

TX PFC frames without error

Number of PFC frames without error

Offset: 0x83D

TX PFC frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.63. TXSTAT Module Revision ID

TXSTAT Module Revision ID

Returns a 4 byte value indicating the revision of this design

Offset: 0x840

TXSTAT Module Revision ID Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module	RO	0x11112015

B.6.64. TXSTAT Scratch Register

TXSTAT Scratch Register

32 bits of scratch register space for testing

Offset: 0x841

TXSTAT Scratch Register Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	scratch		RW	0x0

B.6.65. Reserved

Reserved

Returns 0, override with soft logic to indicate specific core name

Offset: 0x842

Reserved Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.6.66. Configure TX Statistics Counters

Configure TX Statistics Counters

Configuration bits to control the behavior of the TX Statistics counters

Offset: 0x845

Configure TX Statistics Counters Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
2	tx_shadow_req	**TX Shadow Request** 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	RW	0x0
1	rst_tx_parity	**Reset the TX Statistics Parity Error bit** 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	RW	0x0
0	rst_tx_stats	**Reset TX Statistics** 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	RW	0x0

B.6.67. TX Statistics Counter Status

TX Statistics Counter Status

Reports the status of the TX Stats counters

Offset: 0x846

TX Statistics Counter Status Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
1	tx_shadow_on	**TX Shadow Request in progress** 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	RO	0x0
0	tx_parity_err	**TX Statistics parity error detected** 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	RO	0x0

B.6.68. TX Payload bytes with no errors (lower 32 bits)

TX Payload bytes with no errors

Records the number of TX payload bytes in a frame with no FCS, undersized, oversized, or payload length errors.

- Packet bytes are all the bytes from an Ethernet packet except the preamble, the header, and the FCS; the count *does* include PAD bytes
- When TX VLAN/SVLAN detection is enabled, VLAN/SVLAN header bytes are also removed from the count
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum TX Frame Size value are not counted (oversize)
- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)
- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, Intel recommends using snapshot or shadow to freeze the count before reading it.

Offset: 0x860

Access: R0

TX Payload bytes with no errors (lower 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.69. TX Payload bytes with no errors (upper 32 bits)

TX Payload bytes with no errors

Records the number of TX payload bytes in a frame with no FCS, undersized, oversized, or payload length errors.

- Packet bytes are all the bytes from an Ethernet packet except the preamble, the header, and the FCS; the count *does* include PAD bytes
- When tx VLAN/SVLAN detection is enabled, VLAN/SVLAN header bytes are also removed from the count
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum TX Frame Size value are not counted (oversize)
- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)
- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, Intel recommends using snapshot or shadow to freeze the count before reading it.

Offset: 0x861

Access: R0

TX Payload bytes with no errors (upper 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.70. TX Frame bytes with no errors (lower 32 bits)

TX Frame bytes with no errors

Records the number of TX frame bytes from frames with no FCS, undersized, oversized, or payload length errors.

- Frame bytes are all the bytes from an Ethernet packet, except for the preamble bytes
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum TX Frame Size value are not counted (oversize)
- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)
- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, Intel recommends using snapshot or shadow to freeze the count before reading it.

Offset: 0x862

Access: R0

TX Frame bytes with no errors (lower 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	R0	0x0

B.6.71. TX Frame bytes with no errors (upper 32 bits)

TX Frame bytes with no errors

Records the number of TX frame bytes from frames with no FCS, undersized, oversized, or payload length errors.

- Frame bytes are all the bytes from an Ethernet packet, except for the preamble bytes
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum TX Frame Size value are not counted (oversize)
- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)

- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, Intel recommends using snapshot or shadow to freeze the count before reading it.

Offset: 0x863

Access: R0

TX Frame bytes with no errors (upper 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.72. TX Malformed frames (lower 32 bits)

TX Malformed frames

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 or ERROR
- Packets that have ERROR control bytes, but end with a TERM are not considered malformed; they are marked with fcs error

Offset: 0x864

Access: R0

TX Malformed frames (lower 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.73. TX Malformed frames (upper 32 bits)

TX Malformed frames

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 or ERROR
- Packets that have ERROR control bytes, but end with a TERM are not considered malformed; they are marked with fcs error

Offset: 0x865

Access: R0

TX Malformed frames (upper 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.74. TX Packets that were dropped due to error (lower 32 bits)

TX Packets that were dropped due to error

Records the number of TX packets that were dropped due to errors

- The TXMAC automatically pads short frames, except when `i_skip_crc` is asserted for the packet
- When CRC is skipped, if the packet is shorter than 21 bytes, it will be counted as a TX dropped packet

Offset: 0x866

Access: RO

TX Packets that were dropped due to error (lower 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.75. TX Packets that were dropped due to error (upper 32 bits)

TX Packets that were dropped due to error

Records the number of TX packets that were dropped due to errors

- The TXMAC automatically pads short frames, except when `i_skip_crc` is asserted for the packet
- When CRC is skipped, if the packet is shorter than 21 bytes, it will be counted as a TX dropped packet

Offset: 0x867

Access: RO

TX Packets that were dropped due to error (upper 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.76. TX Frames with bad length/type field (lower 32 bits)

TX Frames with bad length/type field

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 16'1536 or more
- If a packet has a L/T field where the value is between 16'd1501 to 16'd1535 (inclusive), the L/T field is considered bad, and this counter is incremented
- Note that if TX VLAN/SVLAN detection is turned on, it is the L/T field inside the VLAN/SVLAN header that is evaluated

Offset: 0x868

Access: RO

TX Frames with bad length/type field (lower 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.6.77. TX Frames with bad length/type field (upper 32 bits)

TX Frames with bad length/type field

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 16'1536 or more
- If a packet has a L/T field where the value is between 16'd1501 to 16'd1535 (inclusive), the L/T field is considered bad, and this counter is incremented
- Note that if TX VLAN/SVLAN detection is turned on, it is the L/T field inside the VLAN/SVLAN header that is evaluated

Offset: 0x869

Access: RO

TX Frames with bad length/type field (upper 32 bits) Fields

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7. RX Statistics Counter Registers

B.7.1. RX Frames less than 64 bytes with CRC error (lower 32 bits)

RX Frames less than 64 bytes with CRC error

Number of frames with less than 64 bytes and a CRC error

Offset: 0x900

RX Frames less than 64 bytes with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.2. RX Frames less than 64 bytes with CRC error (upper 32 bits)

RX Frames less than 64 bytes with CRC error

Number of frames with less than 64 bytes and a CRC error

Offset: 0x901

RX Frames less than 64 bytes with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.3. Oversized RX frames with CRC error (lower 32 bits)

Oversized RX frames with CRC error

Number of frames with more octets than the programmed maximum and a CRC error

Offset: 0x902

Oversized RX frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.4. Oversized RX frames with CRC error (upper 32 bits)

Oversized RX frames with CRC error

Number of frames with more octets than the programmed maximum and a CRC error

Offset: 0x903

Oversized RX frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.5. RX Frames of any size with a CRC error (lower 32 bits)

RX Frames of any size with a CRC error

Number of frames of any size with a CRC error

Offset: 0x904

RX Frames of any size with a CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.6. RX Frames of any size with a CRC error (upper 32 bits)

RX Frames of any size with a CRC error

Number of frames of any size with a CRC error

Offset: 0x905

RX Frames of any size with a CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.7. RX Frames of any size with a CRC error on OK packet (lower 32 bits)

RX Frames of any size with a CRC error on OK packet

Number of frames of any size with a CRC error on OK packet

Offset: 0x906

RX Frames of any size with a CRC error on OK packet (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.8. RX Frames of any size with a CRC error on OK packet (upper 32 bits)

RX Frames of any size with a CRC error on OK packet

Number of frames of any size with a CRC error on OK packet

Offset: 0x907

RX Frames of any size with a CRC error on OK packet (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.9. Multicast RX data frames with CRC error (lower 32 bits)

Multicast RX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a multicast destination address and a CRC error

Offset: 0x908

Multicast RX data frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.10. Multicast RX data frames with CRC error (upper 32 bits)

Multicast RX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a multicast destination address and a CRC error

Offset: 0x909

Multicast RX data frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.11. Broadcast RX data frames with CRC error (lower 32 bits)

Broadcast RX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a broadcast destination address and a CRC error

Offset: 0x90A

Broadcast RX data frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.12. Broadcast RX data frames with CRC error (upper 32 bits)

Broadcast RX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a broadcast destination address and a CRC error

Offset: 0x90B

Broadcast RX data frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.13. Unicast RX data frames with CRC error (lower 32 bits)

Unicast RX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a unicast destination address and a CRC error

Offset: 0x90C

Unicast RX data frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.14. Unicast RX data frames with CRC error (upper 32 bits)

Unicast RX data frames with CRC error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a unicast destination address and a CRC error

Offset: 0x90D

Unicast RX data frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.15. Multicast RX control frames with CRC error (lower 32 bits)

Multicast RX control frames with CRC error

Number of frames where L/T is 16'h8808, and with a multicast destination address and a CRC error

Offset: 0x90E

Multicast RX control frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.16. Multicast RX control frames with CRC error (upper 32 bits)

Multicast RX control frames with CRC error

Number of frames where L/T is 16'h8808, and with a multicast destination address and a CRC error

Offset: 0x90F

Multicast RX control frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.17. Broadcast RX control frames with CRC error (lower 32 bits)

Broadcast RX control frames with CRC error

Number of frames where L/T is 16'h8808, and with a broadcast destination address and a CRC error

Offset: 0x910

Broadcast RX control frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.18. Broadcast RX control frames with CRC error (upper 32 bits)

Broadcast RX control frames with CRC error

Number of frames where L/T is 16'h8808, and with a broadcast destination address and a CRC error

Offset: 0x911

Broadcast RX control frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.19. Unicast RX control frames with CRC error (lower 32 bits)

Unicast RX control frames with CRC error

Number of frames where L/T is 16'h8808, and with a unicast destination address and a CRC error

Offset: 0x912

Unicast RX control frames with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.20. Unicast RX control frames with CRC error (upper 32 bits)

Unicast RX control frames with CRC error

Number of frames where L/T is 16'h8808, and with a unicast destination address and a CRC error

Offset: 0x913

Unicast RX control frames with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.21. RX Pause frame with CRC error (lower 32 bits)

RX Pause frame with CRC error

Number of PAUSE (Standard Flow Control) frames with a CRC error

Offset: 0x914

RX Pause frame with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.22. RX Pause frame with CRC error (upper 32 bits)

RX Pause frame with CRC error

Number of PAUSE (Standard Flow Control) frames with a CRC error

Offset: 0x915

RX Pause frame with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.23. 64 byte RX frames (lower 32 bits)

64 byte RX frames

Number of frames of any type with a frame length of 64 bytes

Offset: 0x916

64 byte RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.24. 64 byte RX frames (upper 32 bits)

64 byte RX frames

Number of frames of any type with a frame length of 64 bytes

Offset: 0x917

64 byte RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.25. 65 to 127 byte RX frames (lower 32 bits)

65 to 127 byte RX frames

Number of frames of any type with a frame length of 65 to 127 bytes

Offset: 0x918

65 to 127 byte RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.26. 65 to 127 byte RX frames (upper 32 bits)

65 to 127 byte RX frames

Number of frames of any type with a frame length of 65 to 127 bytes

Offset: 0x919

65 to 127 byte RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.27. 128 to 257 byte RX frames (lower 32 bits)

128 to 257 byte RX frames

Number of frames of any type with a frame length of 128 to 257 bytes

Offset: 0x91A

128 to 257 byte RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.28. 128 to 257 byte RX frames (upper 32 bits)

128 to 257 byte RX frames

Number of frames of any type with a frame length of 128 to 257 bytes

Offset: 0x91B

128 to 257 byte RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.29. 256 to 511 byte RX frames (lower 32 bits)

256 to 511 byte RX frames

Number of frames of any type with a frame length of 258 to 511 bytes

Offset: 0x91C

256 to 511 byte RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.30. 256 to 511 byte RX frames (upper 32 bits)

256 to 511 byte RX frames

Number of frames of any type with a frame length of 258 to 511 bytes

Offset: 0x91D

256 to 511 byte RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.31. 512 to 1023 byte RX frames (lower 32 bits)

512 to 1023 byte RX frames

Number of frames of any type with a frame length of 512 to 1023 bytes

Offset: 0x91E

512 to 1023 byte RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.32. 512 to 1023 byte RX frames (upper 32 bits)

512 to 1023 byte RX frames

Number of frames of any type with a frame length of 512 to 1023 bytes

Offset: 0x91F

512 to 1023 byte RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.33. 1024 to 1518 byte RX frames (lower 32 bits)

1024 to 1518 byte RX frames

Number of frames of any type with a frame length of 1024 to 1518 bytes

Offset: 0x920

1024 to 1518 byte RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt3	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.34. 1024 to 1518 byte RX frames (upper 32 bits)

1024 to 1518 byte RX frames

Number of frames of any type with a frame length of 1024 to 1518 bytes

Offset: 0x921

1024 to 1518 byte RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt3	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.35. 1519 to max size RX frames (lower 32 bits)

1519 to max size RX frames

Number of frames of any type with a frame length of 1519 to the programmed maximum size in bytes

Offset: 0x922

1519 to max size RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.36. 1519 to max size RX frames (upper 32 bits)

1519 to max size RX frames

Number of frames of any type with a frame length of 1519 to the programmed maximum size in bytes

Offset: 0x923

1519 to max size RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.37. Oversize RX frames (lower 32 bits)

Oversize RX frames

Number of frames of any type with a frame length greater than the programmed maximum size in bytes

Offset: 0x924

Oversize RX frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.38. Oversize RX frames (upper 32 bits)

Oversize RX frames

Number of frames of any type with a frame length greater than the programmed maximum size in bytes

Offset: 0x925

Oversize RX frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt2	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.39. Multicast RX data frames without error (lower 32 bits)

Multicast RX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a multicast destination address and no errors

Offset: 0x926

Multicast RX data frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.40. Multicast RX data frames without error (upper 32 bits)

Multicast RX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a multicast destination address and no errors

Offset: 0x927

Multicast RX data frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.41. Broadcast RX data frames without error (lower 32 bits)

Broadcast RX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a broadcast destination address and no errors

Offset: 0x928

Broadcast RX data frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.42. Broadcast RX data frames without error (upper 32 bits)

Broadcast RX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a broadcast destination address and no errors

Offset: 0x929

Broadcast RX data frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.43. Unicast RX data frames without error (lower 32 bits)

Unicast RX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a unicast destination address and no errors

Offset: 0x92A

Unicast RX data frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.44. Unicast RX data frames without error (upper 32 bits)

Unicast RX data frames without error

Number of frames where L/T is not between 16'd1500 and 16'd1536, and not 16'h8808, with a unicast destination address and no errors

Offset: 0x92B

Unicast RX data frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.45. Multicast RX control frames without error (lower 32 bits)

Multicast RX control frames without error

Number of frames where L/T is 16'h8808, with a multicast destination address and no errors

Offset: 0x92C

Multicast RX control frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.46. Multicast RX control frames without error (upper 32 bits)

Multicast RX control frames without error

Number of frames where L/T is 16'h8808, with a multicast destination address and no errors

Offset: 0x92D

Multicast RX control frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.47. Broadcast RX control frames without error (lower 32 bits)

Broadcast RX control frames without error

Number of frames where L/T is 16'h8808, with a broadcast destination address and no errors

Offset: 0x92E

Broadcast RX control frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.48. Broadcast RX control frames without error (upper 32 bits)

Broadcast RX control frames without error

Number of frames where L/T is 16'h8808, with a broadcast destination address and no errors

Offset: 0x92F

Broadcast RX control frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.49. Unicast RX control frames without error (lower 32 bits)

Unicast RX control frames without error

Number of frames where L/T is 16'h8808, with unicast destination address and no errors

Offset: 0x930

Unicast RX control frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.50. Unicast RX control frames without error (upper 32 bits)

Unicast RX control frames without error

Number of frames where L/T is 16'h8808, with unicast destination address and no errors

Offset: 0x931

Unicast RX control frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.51. RX Pause frames without error (lower 32 bits)

RX Pause frames without error

Number of PAUSE (Standard Flow Control) frames without error

Offset: 0x932

RX Pause frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.52. RX Pause frames without error (upper 32 bits)

RX Pause frames without error

Number of PAUSE (Standard Flow Control) frames without error

Offset: 0x933

RX Pause frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.53. RX Frames with less than 64 bytes and a CRC error (lower 32 bits)

RX Frames with less than 64 bytes and a CRC error

Number of Frames with less than 64 bytes

Offset: 0x934

RX Frames with less than 64 bytes and a CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.54. RX Frames with less than 64 bytes and a CRC error (upper 32 bits)

RX Frames with less than 64 bytes and a CRC error

Number of Frames with less than 64 bytes

Offset: 0x935

RX Frames with less than 64 bytes and a CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.55. Number of RX frame starts (lower 32 bits)

Number of RX frame starts

Number of frame starts

Offset: 0x936

Number of RX frame starts (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.56. Number of RX frame starts (upper 32 bits)

Number of RX frame starts

Number of frame starts

Offset: 0x937

Number of RX frame starts (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.57. Number of RX length errors (lower 32 bits)

Number of RX length errors

Number of frames where the L/T field had a larger length than the frame that was received. Length checking must be enabled

Offset: 0x938

Number of RX length errors (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.58. Number of RX length errors (upper 32 bits)

Number of RX length errors

Number of frames where the L/T field had a larger length than the frame that was received. Length checking must be enabled

Offset: 0x939

Number of RX length errors (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.59. RX PFC frame with CRC error (lower 32 bits)

RX PFC frame with CRC error

Number of PFC frames with a CRC error

Offset: 0x93A

RX PFC frame with CRC error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.60. RX PFC frame with CRC error (upper 32 bits)

RX PFC frame with CRC error

Number of PFC frames with a CRC error

Offset: 0x93B

RX PFC frame with CRC error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.61. RX PFC frames without error (lower 32 bits)

RX PFC frames without error

Number of PFC frames without error

Offset: 0x93C

RX PFC frames without error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.62. RX PFC frames without error (upper 32 bits)

RX PFC frames without error

Number of PFC frames without error

Offset: 0x93D

RX PFC frames without error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt6	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.63. RXSTAT Module Revision ID

RXSTAT Module Revision ID

Returns a 4 byte value indicating the revision of this design

Offset: 0x940

RXSTAT Module Revision ID Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id	**Revision ID** 32b Revision ID for the module	RO	0x11112015

B.7.64. RXSTAT Scratch Register

RXSTAT Scratch Register

32 bits of scratch register space for testing

Offset: 0x941

RXSTAT Scratch Register Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	scratch		RW	0x0

B.7.65. Reserved

Reserved

Returns 0, override with soft logic to indicate specific core name

Offset: 0x942

Reserved Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

Offset: 0x943

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

Offset: 0x944

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.7.66. Reserved

Reserved

Returns 0, override with soft logic to indicate specific core name

Offset: 0x943

Reserved Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.7.67. Reserved

Reserved

Returns 0, override with soft logic to indicate specific core name

Offset: 0x944

Reserved Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	id		RO	0x0

B.7.68. Configure RX Statistics Counters

Configure RX Statistics Counters

Configuration bits to control the behavior of the RX Statistics counters

Offset: 0x945

Configure RX Statistics Counters Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
2	rx_shadow_req	**RX Shadow Request** 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	RW	0x0
1	rst_rx_parity	**Reset the RX Statistics Parity Error bit** 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	RW	0x0
0	rst_rx_stats	**Reset RX Statistics** 1: Reset all RX Stats counters	RW	0x0

continued...

Bit	Name	Description	Access	Reset
		• 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		

B.7.69. RX Statistics Counter Status

RX Statistics Counter Status

Reports the status of the RX Stats counters

Offset: `0x946`

RX Statistics Counter Status Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
1	`rx_shadow_on`	**RX Shadow Request in progress** 1: The CSRs for the RX Statistics are currently frozen, and holding the Stats values from the last time a shadow request was made • Shadow on is asserted for either a shadow request or a snapshot	RO	`0x0`
0	`rx_parity_err`	**RX Statistics parity error detected** 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 `rx_parity_err` is high, it means sometime in the past, a parity error was detected on the stats memory	RO	`0x0`

B.7.70. RX Payload bytes with no errors (lower 32 bits)

RX Payload bytes with no errors

Records the number of RX payload bytes in a frame with no FCS, undersized, oversized, or payload length errors.

- Packet bytes are all the bytes from an Ethernet packet except the preamble, the header, and the FCS; the count *does* include PAD bytes
- When RX VLAN/SVLAN detection is enabled, VLAN/SVLAN header bytes are also removed from the count
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum RX Frame Size value are not counted (oversize)

- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)
- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, Intel recommends using snapshot or shadow to freeze the count before reading it.

Offset: `0x960`

RX Payload bytes with no errors (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.71. RX Payload bytes with no errors (upper 32 bits)

RX Payload bytes with no errors

Records the number of RX payload bytes in a frame with no FCS, undersized, oversized, or payload length errors.

- Packet bytes are all the bytes from an Ethernet packet except the preamble, the header, and the FCS; the count *does* include PAD bytes
- When rx VLAN/SVLAN detection is enabled, VLAN/SVLAN header bytes are also removed from the count
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum RX Frame Size value are not counted (oversize)
- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)
- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, we recommend using snapshot or shadow to freeze the count before reading it.

Offset: `0x961`

RX Payload bytes with no errors (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.72. RX Frame bytes with no errors (lower 32 bits)

RX Frame bytes with no errors

Records the number of RX frame bytes from frames with no FCS, undersized, oversized, or payload length errors.

- Frame bytes are all the bytes from an Ethernet packet, except for the preamble bytes
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum RX Frame Size value are not counted (oversize)
- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)
- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, we recommend using snapshot or shadow to freeze the count before reading it.

Offset: 0x962

RX Frame bytes with no errors (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.73. RX Frame bytes with no errors (upper 32 bits)

RX Frame bytes with no errors

Records the number of RX frame bytes from frames with no FCS, undersized, oversized, or payload length errors.

- Frame bytes are all the bytes from an Ethernet packet, except for the preamble bytes
- Bytes from packets that are less than 72 bytes long are not counted (undersize)
- When length checking is turned on, bytes from frames where the L/T field was a Length, and the length was greater than the number of bytes in the packet are not counted (length error)
- Bytes from packets that are longer than Maximum RX Frame Size value are not counted (oversize)
- Bytes from packets that were interrupted by any kind of control frame are not counted (error or malformed)
- Bytes from packets with an FCS error are not counted (error)
- Bytes from packets with an L/T field that is between 1501 and 1535 inclusive are not counted (illegal type/length field)
- The total count is 64b, split into a lower and upper 32b chunk
- Because the count can change while the register is being read, we recommend using snapshot or shadow to freeze the count before reading it.

Offset: 0x963

RX Frame bytes with no errors (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.74. RX Malformed frames (lower 32 bits)

RX Malformed frames

Records the number of RX packets that were malformed

- A packet is malformed if it is interrupted by an MII Control byte other than TERM or ERROR
- Packets that have ERROR control bytes, but end with a TERM are not considered malformed; they are marked with fcs error

Offset: 0x964

RX Malformed frames (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.75. RX Malformed frames (upper 32 bits)

RX Malformed frames

Records the number of RX packets that were malformed

- A packet is malformed if it is interrupted by an MII Control byte other than TERM or ERROR
- Packets that have ERROR control bytes, but end with a TERM are not considered malformed; they are marked with fcs error

Offset: 0x965

RX Malformed frames (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.76. RX Packets that were dropped due to error (lower 32 bits)

RX Packets that were dropped due to error

Records the number of RX packets that were dropped due to errors

- When RX MAC Configuration register bit en_strict_preamble=1, RX packets where bytes 6:1 of the preamble do not match the Ethernet Standard Preamble byte (8'h55) are dropped
- When RX MAC Configuration register bit en_check_sfd=1, RX packets where the SFD byte does not match the Ethernet Standard SFD byte (8'hd5) are dropped
- When EHIP RX MAC Feature Configuration bit en_pp=0 and RX CRC Forwarding bit forward_rx_crc=0, RX packets shorter than 21 bytes are dropped
- When EHIP RX MAC Feature Configuration bit en_pp=0 and RX CRC Forwarding bit forward_rx_crc=1, RX packets shorter than 17 bytes are dropped
- When EHIP RX MAC Feature Configuration bit en_pp=1 and RX CRC Forwarding bit forward_rx_crc=0, RX packets shorter than 13 bytes are dropped
- When EHIP RX MAC Feature Configuration bit en_pp=1 and RX CRC Forwarding bit forward_rx_crc=1, RX packets shorter than 9 bytes are dropped

Offset: 0x966

RX Packets that were dropped due to error (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0

B.7.77. RX Packets that were dropped due to error (upper 32 bits)

RX Packets that were dropped due to error

Records the number of RX packets that were dropped due to errors

- When RX MAC Configuration register bit `en_strict_preamble=1`, RX packets where bytes 6:1 of the preamble do not match the Ethernet Standard Preamble byte (8'h55) are dropped
- When RX MAC Configuration register bit `en_check_sfd=1`, RX packets where the SFD byte does not match the Ethernet Standard SFD byte (8'hd5) are dropped
- When EHIP RX MAC Feature Configuration bit `en_pp=0` and RX CRC Forwarding bit `forward_rx_crc=0`, RX packets shorter than 21 bytes are dropped
- When EHIP RX MAC Feature Configuration bit `en_pp=0` and RX CRC Forwarding bit `forward_rx_crc=1`, RX packets shorter than 17 bytes are dropped
- When EHIP RX MAC Feature Configuration bit `en_pp=1` and RX CRC Forwarding bit `forward_rx_crc=0`, RX packets shorter than 13 bytes are dropped
- When EHIP RX MAC Feature Configuration bit `en_pp=1` and RX CRC Forwarding bit `forward_rx_crc=1`, RX packets shorter than 9 bytes are dropped

Offset: `0x967`

RX Packets that were dropped due to error (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt16`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	`0x0`

B.7.78. RX Frames with bad length/type field (lower 32 bits)

RX Frames with bad length/type field

Records the number of RX 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 16'1536 or more
- If a packet has a L/T field where the value is between 16'd1501 to 16'd1535 (inclusive), the L/T field is considered bad, and this counter is incremented
- Note that if RX VLAN/SVLAN detection is turned on, it is the L/T field inside the VLAN/SVLAN header that is evaluated

Offset: `0x968`

RX Frames with bad length/type field (lower 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	`stats_pcnt16`	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	`0x0`

B.7.79. RX Frames with bad length/type field (upper 32 bits)

RX Frames with bad length/type field

Records the number of RX 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 16'1536 or more
- If a packet has a L/T field where the value is between 16'd1501 to 16'd1535 (inclusive), the L/T field is considered bad, and this counter is incremented
- Note that if RX VLAN/SVLAN detection is turned on, it is the L/T field inside the VLAN/SVLAN header that is evaluated

Offset: `0x969`

RX Frames with bad length/type field (upper 32 bits) Fields

The reset values in this table represents register values after a reset has completed.

Bit	Name	Description	Access	Reset
31:0	stats_pcnt16	**Statistics word** 4 bytes of an 8 byte EHIP Statistics	RO	0x0