

Intel® Stratix® 10 H-Tile Hard IP for Ethernet IP Core User Guide

Updated for Intel® Quartus® Prime Design Suite: **17.1**

Subscribe
Send Feedback

UG-20121 | 2018.01.10
Latest document on the web: **PDF** | **HTML**

Contents

1 About the Intel® Stratix® 10 H-tile HIP for Ethernet IP Core..... 3
1.1 Intel Stratix 10 H-tile HIP for Ethernet IP Core Supported Features..... 4
1.2 IP Core Device Family and Speed Grade Support..... 7
1.2.1 Intel Stratix 10 H-tile HIP for Ethernet IP Core Device Family Support..... 7
1.2.2 Intel Stratix 10 H-tile HIP for Ethernet IP Core Device Speed Grade Support..... 8
1.3 IP Core Verification..... 8
1.3.1 Simulation Environment..... 8
1.3.2 Compilation Checking..... 8
1.4 Resource Utilization..... 8
1.5 Release Information..... 9

2 Getting Started..... 10
2.1 Installing and Licensing Intel FPGA IP Cores..... 10
2.2 Specifying the IP Core Parameters and Options..... 11
2.3 Generated File Structure..... 12
2.4 Integrating Your IP Core in Your Design..... 14
2.4.1 Channel Placement..... 14
2.4.2 Pin Assignments..... 16
2.4.3 Adding the Transceiver PLLs..... 17
2.4.4 Clock Requirements..... 19
2.4.5 Placement Settings for the Intel Stratix 10 H-tile HIP for Ethernet IP Core..... 19
2.5 IP Core Testbenches..... 20
2.6 Compiling the Full Design..... 20

3 Intel Stratix 10 H-tile HIP for Ethernet Parameters..... 21
3.1 Parameter Editor Parameters..... 21
3.2 RTL Parameters..... 26

4 Functional Description..... 29

5 Reset..... 30

6 Interfaces and Signal Descriptions..... 32
6.1 TX MAC Interface to User Logic..... 33
6.2 RX MAC Interface to User Logic..... 36
6.3 TX PCS Interface to User Logic..... 38
6.4 RX PCS Interface to User Logic..... 40
6.5 Ethernet Link and Transceiver Signals..... 41
6.6 Transceiver Reconfiguration Signals..... 42
6.7 Ethernet Reconfiguration Interface..... 44
6.8 Miscellaneous Status and Debug Signals..... 44
6.9 Reset Signals..... 45
6.10 Clocks..... 46

7 Ethernet Reconfiguration and Status Register Descriptions..... 48

A Advanced RTL Parameters..... 49

B Additional Information..... 57
B.1 Intel Stratix 10 H-tile HIP for Ethernet User Guide Revision History..... 57

1 About the Intel® Stratix® 10 H-tile HIP for Ethernet IP Core

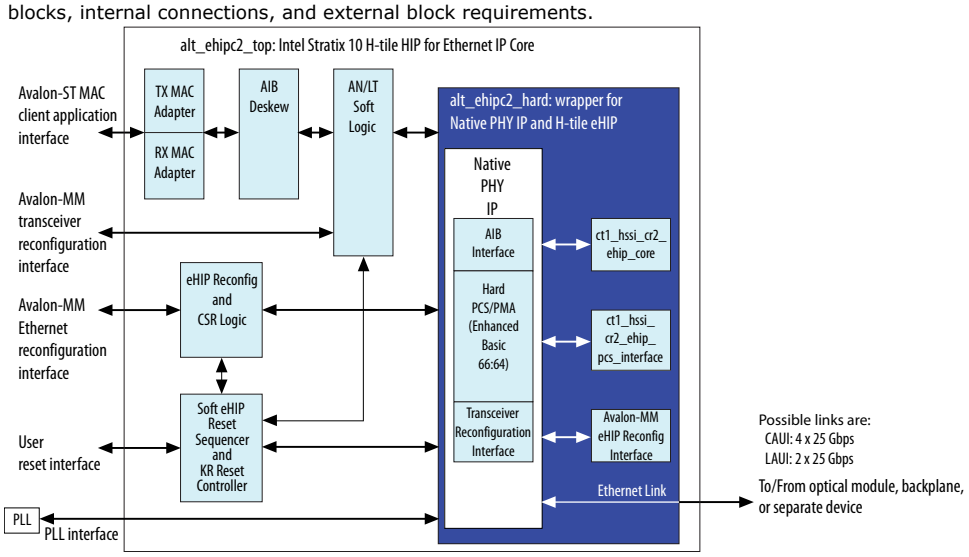
Intel Stratix 10 H-tile FPGA production devices include a configurable, hardened protocol stack for Ethernet that is compatible with the *IEEE 802.3 High Speed Ethernet Standard* and the *25G & 50G Ethernet Specification, Draft 1.6* from the 25G Ethernet Consortium.

The Intel Stratix 10 H-tile HIP for Ethernet 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.

Note: The full product name, Intel Stratix 10 H-Tile Hard IP for Ethernet, is shortened to Intel Stratix 10 H-tile HIP for Ethernet IP core in this document.

Figure 1. Intel Stratix 10 H-tile HIP for Ethernet IP Core

Main blocks, internal connections, and external block requirements.



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 variation or a PCS-only variation.

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 (MII)	128	256

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

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.

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

1.1 Intel Stratix 10 H-tile HIP for Ethernet 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 Intel Stratix 10 H-tile HIP for Ethernet IP core variations are in full-duplex mode. These IP core variations offer the following features:

- PHY features:
  - 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
    - Auto-negotiation (AN) as defined in *IEEE Standard 802.3-2915 Clause 73* and the *25G Ethernet Consortium Schedule Draft 1.6*, and
    - Link training (LT) as defined in *IEEE Standard 802.3-2915 Clauses 92 and 93* and the *25G Ethernet Consortium Schedule Draft 1.6*
  - Optional deficit idle counter (DIC) options to maintain a finely controlled 8-byte, 10-byte, or 12-byte inter-packet gap (IPG) minimum average, or allow the user to drive the IPG from the client interface.
  - RX Skew Variation tolerance that exceeds the *IEEE 802.3-2015 High Speed Ethernet Standard* Clause 80.5 requirements.
- Frame structure control features:
  - Support for jumbo packets.
  - RX CRC pass-through control.
  - 1000 bits RX PCS lane skew tolerance, 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.

- 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.
  - 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 features:
  - Optional serial PMA loopback (TX to RX) at the serial transceiver for self-diagnostic testing.
  - Optional parallel loopback (TX to RX) at the MAC or at the PCS for self-diagnostic testing.
  - Bit-interleaved parity error counters to monitor bit errors per PCS lane.
  - RX PCS error block counters to monitor errors during and between frames.
  - Malformed and dropped packet counters.
  - High BER detection to monitor link bit error rates over all PCS lanes.
  - Optional scrambled Idle test pattern generation and checking.
  - Snapshot feature for precisely timed capture of statistics counter values.
  - TX error insertion capability supports test and debug.
- User system interfaces:
  - 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.

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

Related Links

- 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 Stratix 10 H-tile HIP for Ethernet IP core:

Intel Stratix 10 H-tile HIP for Ethernet IP Core Device Family Support on page 7

Intel Stratix 10 H-tile HIP for Ethernet IP Core Device Speed Grade Support on page 8

1.2.1 Intel Stratix 10 H-tile HIP for Ethernet IP Core Device Family Support

Table 2. 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 3. Intel Stratix 10 H-tile HIP for Ethernet IP Core Device Family Support

Shows the level of support offered by the Intel Stratix 10 H-tile HIP for Ethernet 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 Links

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 Intel Stratix 10 H-tile HIP for Ethernet IP Core Device Speed Grade Support

The Intel Stratix 10 H-tile HIP for Ethernet 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 Intel Stratix 10 H-tile HIP for Ethernet IP core, Intel performs extensive validation through both simulation and hardware testing. Before releasing a version of the Intel Stratix 10 H-tile HIP for Ethernet IP core, Intel runs comprehensive regression tests in the current version of the Intel Quartus Prime Pro Edition software.

Related Links

- Knowledge Base errata for the Intel Stratix 10 H-tile HIP for Ethernet IP core
  Exceptions to functional correctness that manifest in software releases 17.1 and later are documented in the Intel Stratix 10 H-tile HIP for Ethernet IP core errata.
- Intel Stratix 10 H-tile HIP for Ethernet IP Core Release Notes
  Changes to the Intel Stratix 10 H-tile HIP for Ethernet 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 Stratix 10 H-tile HIP for Ethernet IP core 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 Stratix 10 H-tile HIP for Ethernet IP core variations and designs that target different devices, to ensure the Intel Quartus Prime Pro Edition software places and routes the IP core ports correctly.

1.4 Resource Utilization

Resource utilization changes depending on the parameter settings you specify in the Intel Stratix 10 H-tile HIP for Ethernet 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 variation or a PCS Only variation, do affect footprint on the device. If you select a MAC+PCS varation, 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.

Table 4. IP Core FPGA Resource Utilization

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

These results were obtained using the Intel Quartus Prime Pro Edition v17.1 software. All parameters are at their default values except the parameters listed in the table:

- **Select Ethernet Rate**
- **Select 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.

IP Core Variation			ALMs	Dedicated Logic Registers	Memory M20K
Select Ethernet Rate	**Select Ethernet IP Layers**	**Enable AN/LT**			
100G	MAC+PCS	True			
		False	4900	7100	2
	PCS Only	True			
		False	2200	1700	0
50G	MAC+PCS	True			
		False	1400	1900	0
	PCS Only	True			
		False	1300	1200	0

Related Links

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

1.5 Release Information

Table 5. Intel Stratix 10 H-tile HIP for Ethernet IP Core Current Release Information

Item	Description
Version	17.1
Release Date	2017.11.06
Ordering Code	IP-ETH-HTILEHARDIP

2 Getting Started

The following sections explain how to install, parameterize, simulate, and initialize the Intel Stratix 10 H-tile HIP for Ethernet IP core:

Installing and Licensing Intel FPGA IP Cores on page 10

Specifying the IP Core Parameters and Options on page 11

Generated File Structure on page 12

Integrating Your IP Core in Your Design on page 14

IP Core Testbenches on page 20

Compiling the Full Design on page 20

Related Links

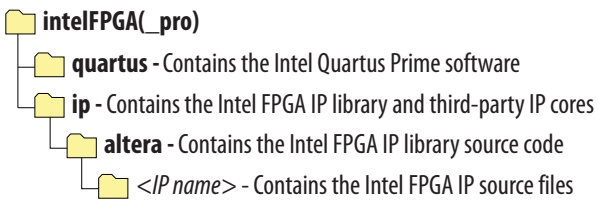
- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Qsys Simulation Scripts
  Create simulation scripts that do not require manual updates for 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



Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

Table 6. IP Core Installation Locations

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

2.2 Specifying the IP Core Parameters and Options

The Intel Stratix 10 H-tile HIP for Ethernet 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 Intel Stratix 10 H-tile HIP for Ethernet 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 21 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. Click **Finish**. The parameter editor adds the top-level `.ip` file to the current project automatically. If you are prompted to manually add the `.ip` file to the project, click **Project ➤ Add/Remove Files in Project** to add the file.
10. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports and set any appropriate per-instance RTL parameters.

Related Links

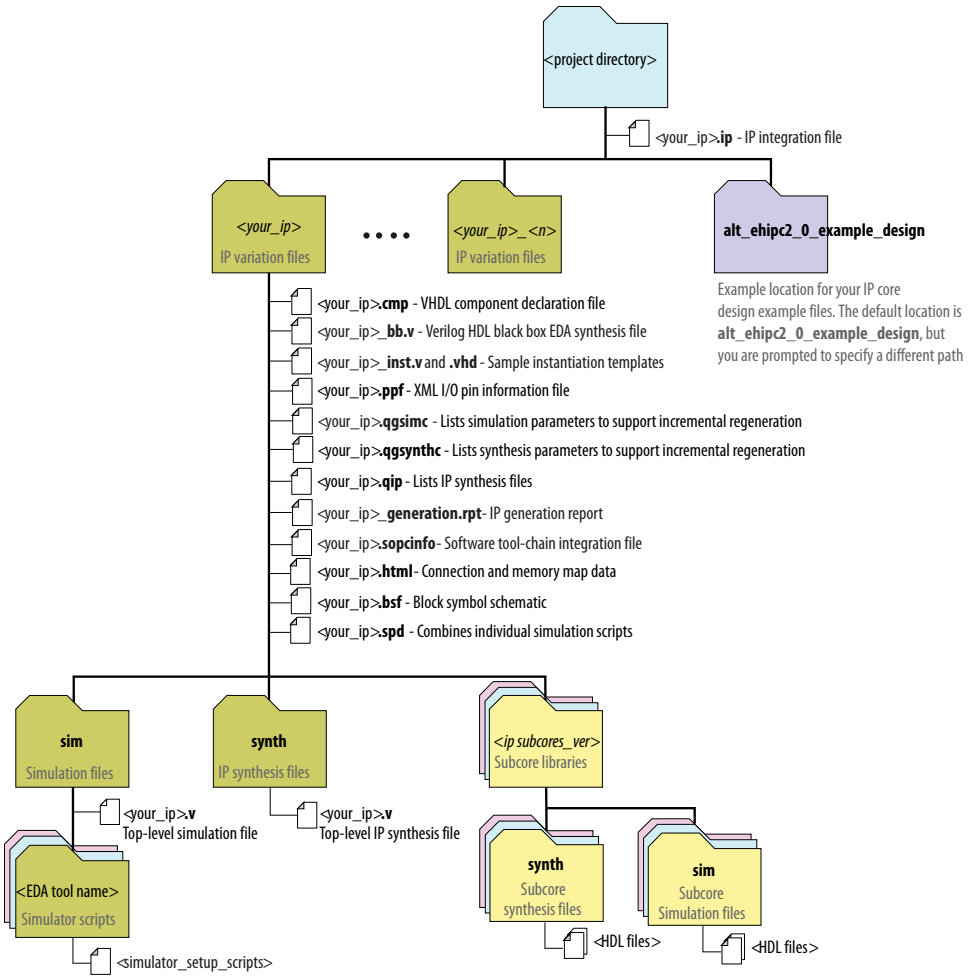
Intel Stratix 10 H-tile Hard IP for Ethernet Design Example User Guide
Information about generating the Intel Stratix 10 H-tile HIP for Ethernet design example.

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 *Intel Stratix 10 H-tile HIP for Ethernet Design Example User Guide*.

Figure 3. Intel Stratix 10 H-tile HIP for Ethernet IP Core Generated Files



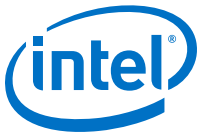


Table 7. IP Core Generated Files

File Name	Description
<*your_ip*>.ip	The Platform Designer system or top-level IP variation file. <*your_ip*> is the name that you give your IP variation.
<*your_ip*>.cmp	The VHDL Component Declaration (.cmp) file is a text file that contains local generic and port definitions that you can use in VHDL design files.
<*your_ip*>.html	A report that contains connection information, a memory map showing the address of each slave with respect to each master to which it is connected, and parameter assignments.
<*your_ip*>_generation.rpt	IP or Platform Designer generation log file. A summary of the messages during IP generation.
<*your_ip*>.qgsimc	Lists simulation parameters to support incremental regeneration.
<*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/	Contains a shell script vcs_setup.sh to set up and run a VCS* simulation.

continued...

File Name	Description
synopsys/vcsmx/	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 Links

Intel Stratix 10 H-tile Hard IP for Ethernet Design Example User Guide
Information about the Intel Stratix 10 H-tile HIP for Ethernet 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 14
Pin Assignments on page 16
Adding the Transceiver PLLs on page 17
Clock Requirements on page 19
Placement Settings for the Intel Stratix 10 H-tile HIP for Ethernet IP Core on page 19

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 Intel Stratix 10 H-tile HIP for Ethernet IP core instance in your design, you must make appropriate pin assignments. You can create a virtual pin to avoid making specific pin assignments for top-level signals until you are ready to map the design to hardware.

Intel Stratix 10 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 Links

- 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.

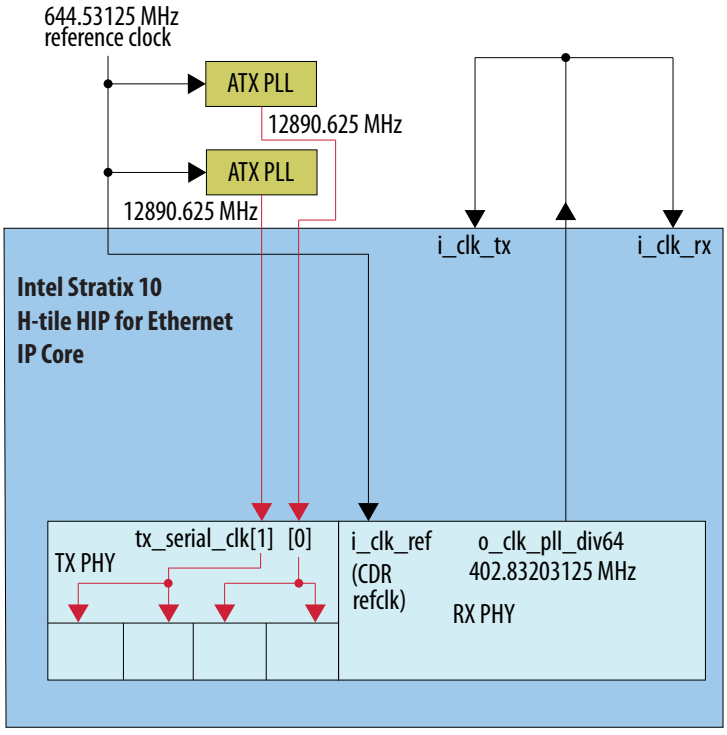
2.4.3 Adding the Transceiver PLLs

The Intel Stratix 10 H-tile HIP for Ethernet 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 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 Intel Stratix 10 H-tile HIP for Ethernet 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 IP Core Variation

The TX transceiver PLLs are instantiated with two Intel Stratix 10 ATX PLL IP cores. The TX transceiver PLLs must always be instantiated outside the Intel Stratix 10 H-tile HIP for Ethernet IP core.



You can use the IP Catalog to create a transceiver PLL.

- Select **Stratix 10 L-Tile/H-Tile Transceiver ATX PLL**.
- 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 local clock output port (tx_serial_clk_gxt)**.
  - Set **GXT output clock source** to **Local ATX PLL**.
  - **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 Intel Stratix 10 H-tile HIP for Ethernet 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_171/sim/*<variation_name>*_altera_xcvr_atx_pll_s10_htile_171_*<random_string>*.sv and the synthesis file *<variation_name>*/altera_xcvr_atx_pll_s10_htile_171/synth/*<variation_name>*_altera_xcvr_atx_pll_s10_htile_171_*<random_string>*.sv. However, the HDL code for the Intel Stratix 10 H-tile HIP for Ethernet IP core does not instantiate the ATX PLL. If you choose to use the ATX PLL provided with the Intel Stratix 10 H-tile HIP for Ethernet IP core, you must instantiate and connect the instances of the ATX PLL with the Intel Stratix 10 H-tile HIP for Ethernet IP core in user logic.

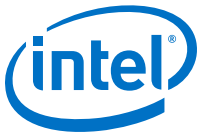
If you generate your own ATX PLL, you must ensure it has a different filename than the PLL provided with the IP core.

You must drive the reference clock input ports of the two PLLs with the same clock to minimize PPM differences. This clock can be but need not be the same as the clock that drives the Intel Stratix 10 H-tile HIP for Ethernet IP core reference clock.

Each PLL drives the `tx_serial_clk` input of two of the Intel Stratix 10 H-tile HIP for Ethernet IP core PHY links. You must connect the PLLs to the Intel Stratix 10 H-tile HIP for Ethernet IP core as follows:

PLL	PLL Signal	Intel Stratix 10 H-tile HIP for Ethernet IP Core Signal
A	`tx_serial_clk`	`i_tx_serial_clk[0]`
A	`pll_locked`	`i_tx_pll_locked[0]`
B	`tx_serial_clk`	`i_tx_serial_clk[1]`
B	`pll_locked`	`i_tx_pll_locked[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.



Related Links

- Intel Stratix 10 L- and H-Tile Transceiver PHY 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.
- Intel Stratix 10 H-tile Hard IP for Ethernet Design Example User Guide
  Information about the Intel Stratix 10 H-tile HIP for Ethernet design example, which connects the appropriate number of external PLLs to the IP core PHY links.

2.4.4 Clock Requirements

For normal 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`.

Related Links

Clocks on page 46

2.4.5 Placement Settings for the Intel Stratix 10 H-tile HIP for Ethernet IP Core

The Intel Quartus Prime Pro Edition software provides the options to specify design partitions and LogicLock® 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 Links

- Intel Quartus Prime Pro Edition Handbook Volume 2: Design Implementation and Optimization
  Describes design constraints and LogicLock Plus 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 compilation-only design example and a testbench that you can generate for the Intel Stratix 10 H-tile HIP for Ethernet IP core.

To generate the testbench, in the Intel Stratix 10 H-tile HIP for Ethernet 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 Links

Intel Stratix 10 H-tile Hard IP for Ethernet Design Example User Guide
Information about generating and running the design example and testbench files for the Intel Stratix 10 H-tile HIP for Ethernet 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

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

Related Links

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

3 Intel Stratix 10 H-tile HIP for Ethernet Parameters

3.1 Parameter Editor Parameters

The Intel Stratix 10 H-tile HIP for Ethernet parameter editor provides the parameters you can set to configure your Intel Stratix 10 H-tile HIP for Ethernet IP core variation and simulation and hardware design examples.

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

Table 8. Intel Stratix 10 H-tile HIP for Ethernet 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			
Select Ethernet Rate	• 50G • 100G	50G	Selects the IP core Ethernet data rate.
Select Ethernet IP Layers	• MAC+PCS • PCS Only	MAC+PCS	Selects the inclusion or exclusion of a MAC layer in your IP core variation.
Enter 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*. In PCS Only 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 variations, this parameter has no effect and remains at the default value of 1518.

continued...

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

Parameter	Range	Default Setting	Parameter Description
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 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.
Choose 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.
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.

continued...

Parameter	Range	Default Setting	Parameter Description
			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. In future releases the parameter editor will display and accept input in hexidecimal 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.
Keep RX CRC	• True • False	False	Selects RX CRC forwarding. If turned on, the IP core maintains the CRC bits from the Ethernet packets and includes them in the data on the RX client interface. If turned off, the IP core strips the CRC bits before sending the data on the RX client interface.
Remove pads	• True • False	False	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.
TX VLAN Detection	• True • False	False	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	False	Specifies whether the IP core RX statistics block treats RX VLAN and Stacked VLAN Ethernet frames as regular control frames, or performs Length/Type field decoding, includes these frame in VLAN statistics, and counts the payload bytes instead of the full Ethernet frame in the `RxFrameOctetsOK` counter at offsets 0x962 and 0x963. If turned on, the IP core identifies these frames in RX statistics as VLAN or Stacked VLAN frames. If turned off, the IP core treats these frames as regular control frames.
MAC Options: Specialized Tab *Note:* In PCS 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

continued...

Parameter	Range	Default Setting	Parameter Description
			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).
Enable SyncE	• True • False	False	Exposes the RX recovered clocks `o_clk_rec_div64` and `o_clk_rec_div66` as output signals. This feature supports the Synchronous Ethernet standard described in the ITU-T G.8261, G.8262, and G.8264 recommendations. In fact these clocks are available to support the Synchronous Ethernet standard whether you turn on this parameter or turn off this parameter.
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*

continued...

Parameter	Range	Default Setting	Parameter Description
			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.
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.
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*.
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: General			
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*.

continued...

Parameter	Range	Default Setting	Parameter Description
Number of frames to send at end of training	• 127 • 255	127	Specifies the number of additional training frames the local link partner delivers 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 generates 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 only 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.
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.

Related Links

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

3.2 RTL Parameters

The Intel Stratix 10 H-tile HIP for Ethernet 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 9. Intel Stratix 10 H-tile HIP for Ethernet RTL Parameters

Parameter	Parameter Description
Parameters Available for all IP Core Variations	
`sim_mode`	Specifies whether the IP core is in simulation mode, in which alignment marker periods are shortened to decrease the time to RX PCS alignment. • Value `disable` (default value): The IP core MAC implements standard alignment marker periods as specified in the *IEEE Standard 802.3–2015*. Before compiling for synthesis, ensure this parameter has this value. • Value `enable`: The IP core implements shorter alignment marker periods to accelerate RX PCS alignment in simulation. The simulation link partner must have the same alignment marker periods. This mode is intended for simulation only. The value of this parameter determines the initial and reset values of these register fields: • `am_interval[13:0]` field (bits [13:0]) of the RXPCS_CONF register at 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 addresss 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 addresss 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 addresss 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 addresss 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 Links

Advanced RTL Parameters on page 49
Additional RTL parameters are available for advanced applications.

4 Functional Description

This chapter is pending.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

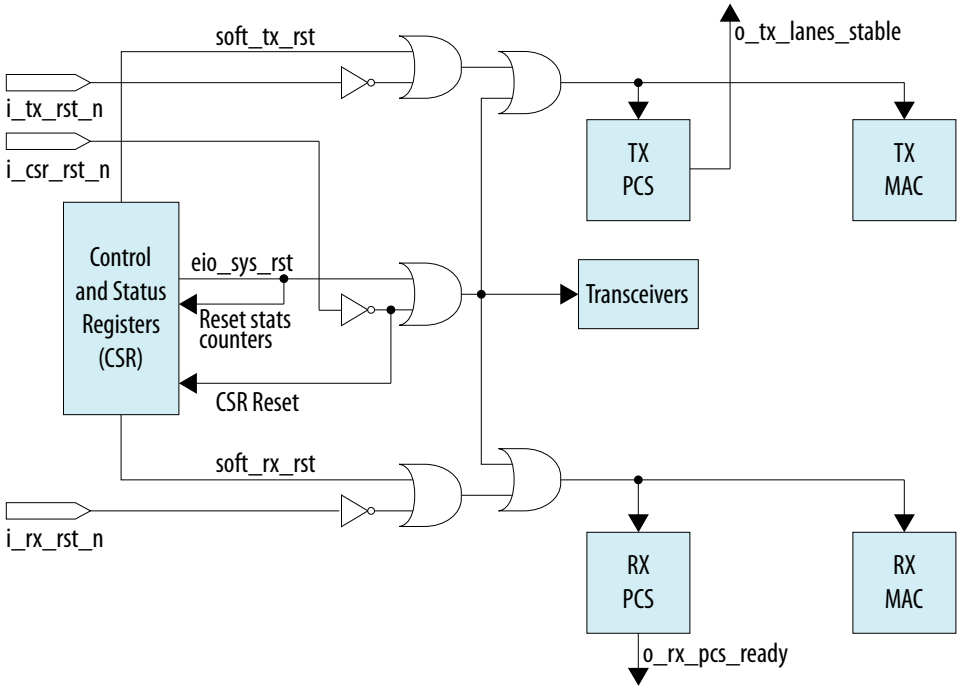
5 Reset

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 7. 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.



Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

The general reset signals reset the following functions:

- `soft_tx_rst`, `i_tx_rst_n`: Resets the IP core in the TX direction. 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 IP core in the RX direction. 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.

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, preferably by asserting the `i_csr_rst_n` signal. 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.

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.

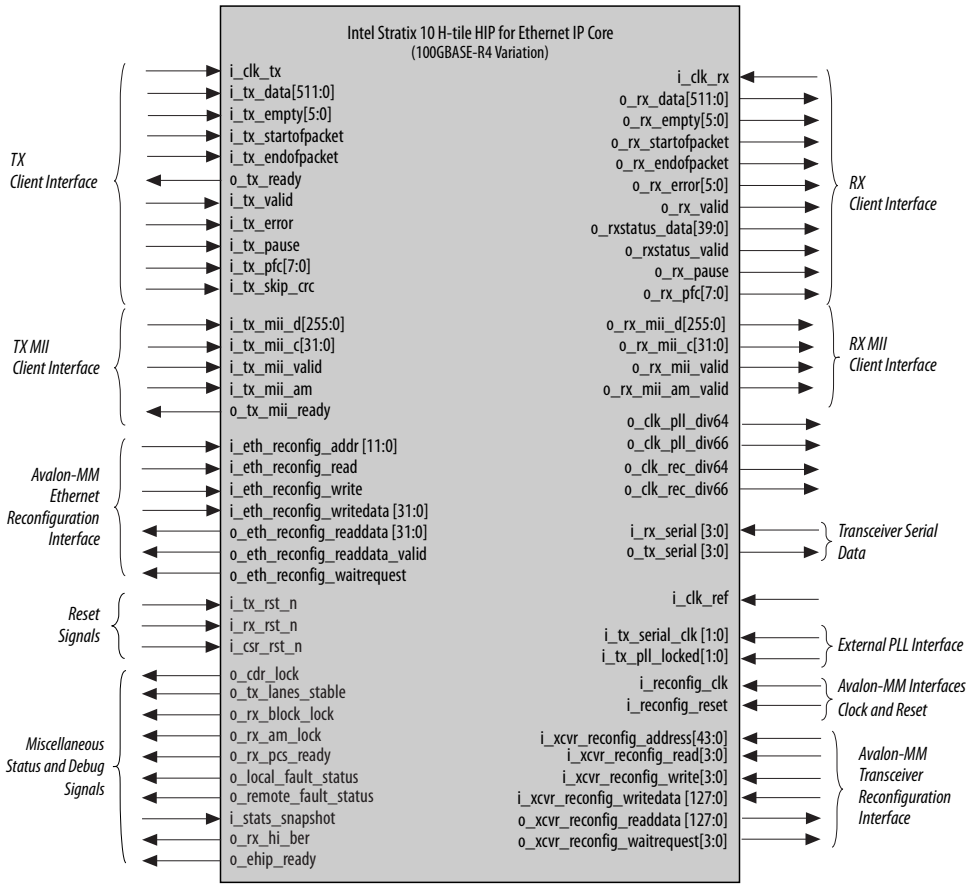
Related Links

- Reset Signals on page 45
- Ethernet Reconfiguration and Status Register Descriptions on page 48
  The soft reset bits are `PHY_CONFIG` register bits.

6 Interfaces and Signal Descriptions

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

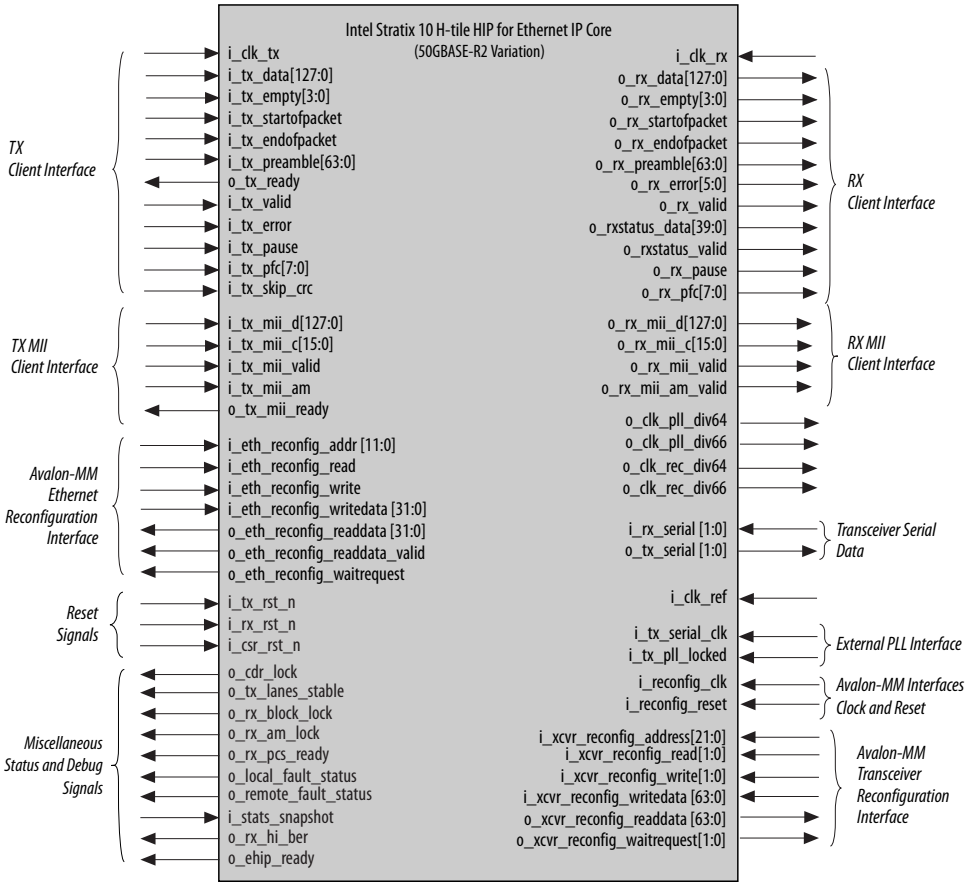
Figure 8. Intel Stratix 10 H-tile HIP for Ethernet 100GBASE-R4 Signals and Interfaces



Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

Figure 9. Intel Stratix 10 H-tile HIP for Ethernet 50GBASE-R2 Signals and Interfaces



6.1 TX MAC Interface to User Logic

The Intel Stratix 10 H-tile HIP for Ethernet 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 10. 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 Intel Stratix 10 H-tile HIP for Ethernet parameter editor is the Avalon-ST readyLatency value on this interface.

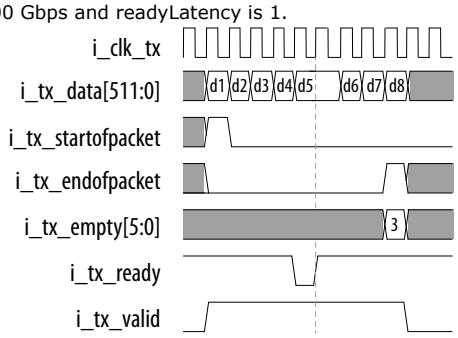
Signal Name	Description
i_clk_tx	The TX clock for the IP core is i_clk_tx. The frequency of this clock is 402.83203125 MHz.
i_tx_data[127:0] (in 50GBASE-R2 variations) i_tx_data[511:0] (in 100GBASE-R4 variations)	TX data. If the preamble pass-through feature is enabled, data in 100GBASE-R4 variations begins with the preamble. The Intel Stratix 10 H-tile HIP for Ethernet 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 errored, 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_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.
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.

continued...

Signal Name	Description
	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).

Figure 10. TX Avalon-ST MAC Client Interface for Intel Stratix 10 H-tile HIP for Ethernet IP Core

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



Related Links

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

6.2 RX MAC Interface to User Logic

The Intel Stratix 10 H-tile HIP for Ethernet 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 11. 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 402.83203125 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).

continued...

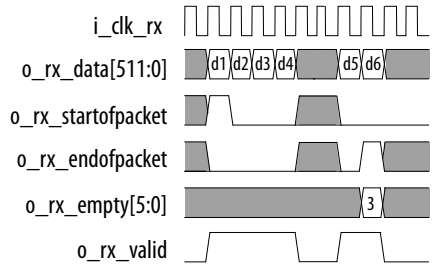
Name	Description
o_rx_startofpacket	When asserted, indicates that o_rx_data holds the first clock cycle of data in a packet (start of packet). The IP core asserts this signal for only a single clock cycle for each packet. When o_rx_startofpacket is asserted, the MSB of o_rx_data drives the start of packet.
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. • 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

continued...

Name	Description
	• 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 11. RX Avalon-ST MAC Client Interface for Intel Stratix 10 H-tile HIP for Ethernet IP Core

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



Related Links

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

6.3 TX PCS Interface to User Logic

The Intel Stratix 10 H-tile HIP for Ethernet 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 12. 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 402.83203125 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. The number of valid clock cycles from deassertion of `i_tx_mii_am` 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 in 100GBASE-R4 variations, and 32766 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 *Intel Stratix 10 H-tile Hard IP for Ethernet Design Example User Guide*. For information about the `sim_mode` RTL parameter, refer to the *RTL Parameters* section of this user guide. While you hold the value of this signal at 1, you must freeze the values of both `i_tx_mii_d` and `i_tx_mii_c`.

Figure 12. TX MII Client Interface for Intel Stratix 10 H-tile HIP for Ethernet IP Core

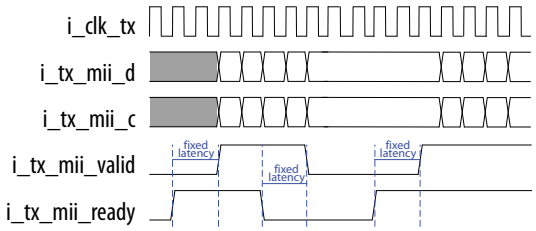
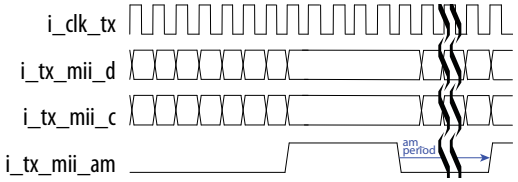


Figure 13. Alignment Marker Insertion on the TX MII Client Interface



Related Links

Intel Stratix 10 H-tile Hard IP for Ethernet Design Example User Guide
The Intel Stratix 10 H-tile HIP for Ethernet 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 Intel Stratix 10 H-tile HIP for Ethernet 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 13. Signals of the MII RX Client Interface

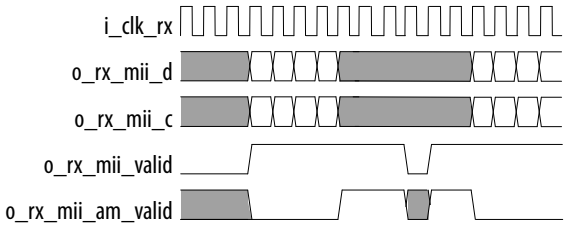
All interface signals are clocked by the `i_clk_rx` clock.

Signal Name	Description
`i_clk_rx`	The RX clock for the IP core is `i_clk_rx`. The frequency of this clock isk 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.

continued...

Signal Name	Description
	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 14. RX MII Client Interface for Intel Stratix 10 H-tile HIP for Ethernet IP Core



6.5 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 14. Transceiver Signals

Signal	Description
`i_tx_serial[1:0]` (in 50GBASE-R2 variations) `i_tx_serial[3:0]` (in 100GBASE-R4 variations)	TX transceiver data. Each `i_tx_serial` bit becomes two physical pins that form a differential pair.
`o_rx_serial[1:0]` (in 50GBASE-R2 variations) `o_rx_serial[3:0]` (in 100GBASE-R4 variations)	RX transceiver data. Each `o_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*. 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.

continued...

Signal	Description
	You must drive these clocks from the ATX PLL or ATX PLLs that you configure separately from the Intel Stratix 10 H-tile HIP for Ethernet 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 Intel Stratix 10 H-tile HIP for Ethernet 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 Links

- Adding the Transceiver PLLs on page 17
  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.

6.6 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 15. 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) `i_xcvr_reconfig_read[3:0]` (in 100GBASE-R4 variations)	Read request signal. Signal is active high. 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]`.

continued...

Port Name	Description
i_xcvr_reconfig_writedata[63:0] (in 50GBASE-R2 variations) i_xcvr_reconfig_writedata[127:0] (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[32(n+1)-1:32n]. 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[63:32].
o_xcvr_reconfig_readdata[63:0] (in 50GBASE-R2 variations) o_xcvr_reconfig_readdata[127:0] (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[32(n+1)-1:32n]. 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[63:32]. 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.

Related Links

- 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

6.7 Ethernet Reconfiguration Interface

You access Ethernet control and status registers of the Intel Stratix 10 H-tile HIP for Ethernet 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` signal. Asserting the `i_csr_rst_n` signal resets all Ethernet control and status registers, including the statistics counters; while this reset is in process, the Ethernet reconfiguration interface does not respond.

Table 16. 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[11:0]`	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 Links

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.

6.8 Miscellaneous Status and Debug Signals

The Intel Stratix 10 H-tile HIP for Ethernet 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 17. 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.
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.

6.9 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 18. Reset Signals

All of the IP core reset signals except the `i_reconfig_reset` signal are asynchronous.

Signal	Description
`i_tx_rst_n`	Active low hard reset signal. Resets the TX interface, including the TX PCS and TX MAC. This reset leads to the deassertion of the `o_tx_lanes_stable` output signal.
`i_rx_rst_n`	Active low hard reset signal. Resets the RX interface, including the RX PCS and RX MAC. This reset leads to the deassertion of the `o_rx_pcs_ready` output signal.
`i_csr_rst_n`	Active low hard 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 Intel Stratix 10 H-tile HIP for Ethernet 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 Links

Reset on page 30

6.10 Clocks

You must set the transceiver reference clock (`i_clk_ref`) frequency to a value that the IP core supports. The Intel Stratix 10 H-tile HIP for Ethernet 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 Intel Stratix 10 H-tile HIP for Ethernet 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 Intel Stratix 10 H-tile HIP for Ethernet IP core performs the filtering.

Table 19. 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.

continued...

Signal Name	Description
	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 Intel Stratix 10 H-tile HIP for Ethernet IP core. Refer to *Adding the Transceiver PLLs*.
`i_reconfig_clk`	Avalon clock for the Intel Stratix 10 H-tile HIP for Ethernet 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 20. 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 times 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 Links

- Clock Requirements on page 19
- Intel Stratix 10 Device Datasheet
  Provides transceiver reference clock phase noise specifications.

7 Ethernet Reconfiguration and Status Register Descriptions

This chapter provides information about the Ethernet registers for the Intel Stratix 10 H-tile HIP for Ethernet. You access these registers using the IP core Avalon-MM Ethernet reconfiguration interface. The 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 21. Register Base Addresses

Word Offset	Register Type
0xB0-0xFF	Stratix 10 LL 40GBASE-KR4/CR4 registers
0x300-0x3FF	PHY registers
0x400-0x4FF	TX MAC registers
0x500-0x5FF	RX MAC registers
0x600-0x708	Pause and Priority-Based Flow Control registers
0x800-0x8FF	Statistics Counter registers - TX direction
0x900-0x9FF	Statistics Counter registers - RX direction

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.

Note: Register content information is pending.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

A Advanced RTL Parameters

The Intel Stratix 10 H-tile HIP for Ethernet 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 22. Intel Stratix 10 H-tile HIP for Ethernet 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...

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

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 0ffset 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 0ffset 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 0ffset 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 0ffset 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 0ffset 0x50A.

continued...

Parameter	Parameter Description
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). • 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.

continued...

Parameter	Parameter Description
	• Default value is the value of the parameter editor **TX Maximum Frame Size** parameter. • Range is 65 through $2^{16}-1$. 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 Links

RTL Parameters on page 26
The basic RTL parameters suffice for most applications.

B Additional Information

B.1 Intel Stratix 10 H-tile HIP for Ethernet User Guide Revision History

Table 23. Document Revision History

Date	Version	Changes
2018.01.12	17.1	Initial public release. At this time, the *Registers* and *Functional Description* chapters are pending.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered