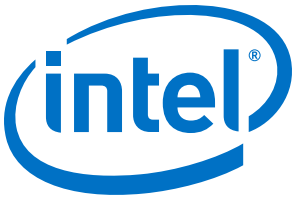
Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!


## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



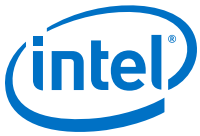
# 25G Ethernet Intel® Arria® 10 FPGA IP User Guide

Updated for Intel® Quartus® Prime Design Suite: **17.0**

**UG-20015 | 2018.07.17**
Latest document on the web: **PDF** | **HTML**

**Subscribe**
**Send Feedback**

# Contents

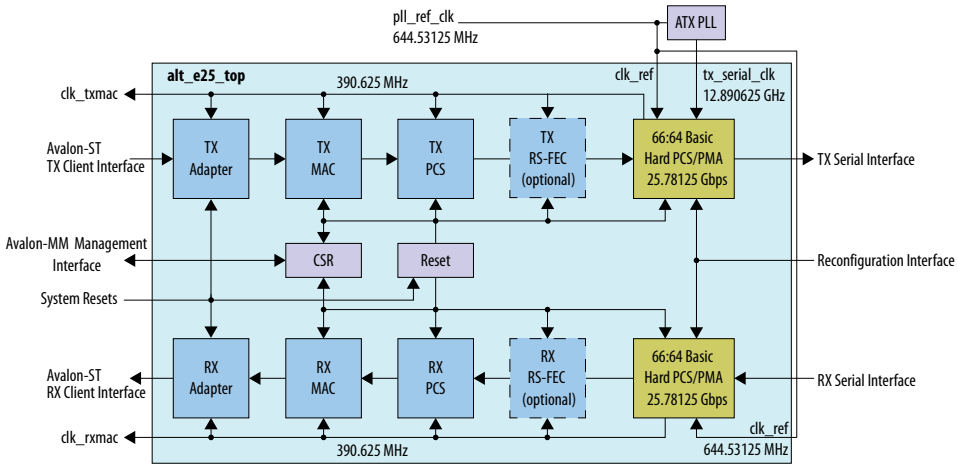


# 1. About 25G Ethernet Intel FPGA IP Core

The Intel® Arria® 10 25G Ethernet Intel FPGA IP core implements the *25G & 50G Ethernet Specification, Draft 1.6* from the 25 Gigabit Ethernet Consortium and the *IEEE 802.3by 25Gb Ethernet* specification. The IP core includes an option to support unidirectional transport as defined in *Clause 66* of the *IEEE 802.3-2012 Ethernet Standard*. The MAC client side interface for the 25G Ethernet Intel FPGA IP core is a 64-bit Avalon® Streaming (Avalon-ST) interface. It maps to one 25.78125 Gbps transceiver. The IP core optionally includes Reed-Solomon forward error correction (FEC) for support of direct attach copper (DAC) cable. *IEEE 802.3 Clause 74* KR-FEC is not supported.

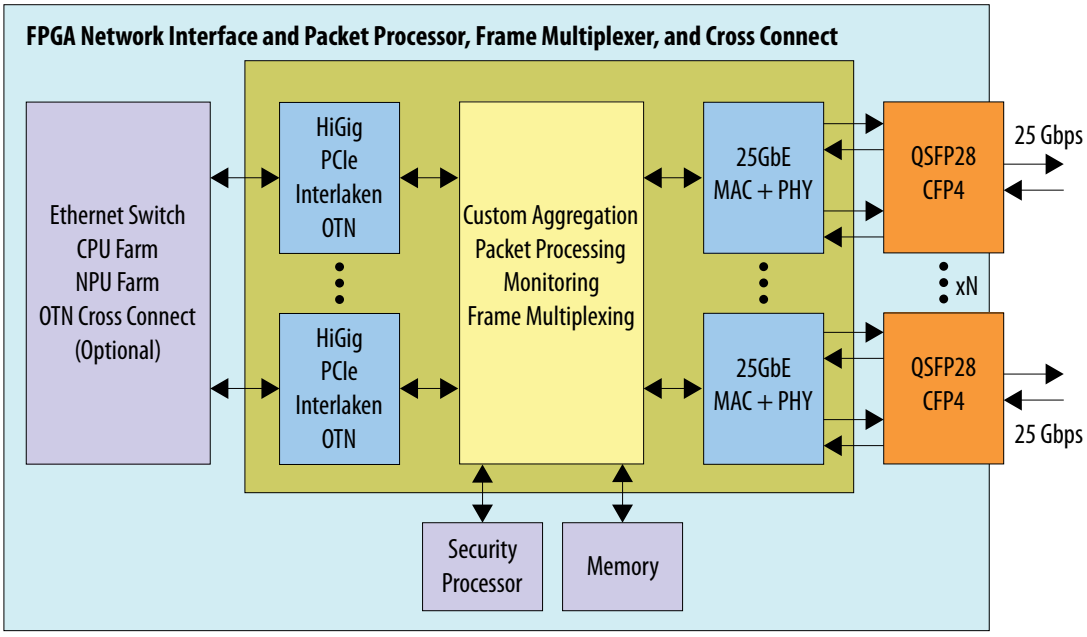
The IP core provides standard media access control (MAC) and physical coding sublayer (PCS), Reed-Solomon FEC, and PMA functions shown in the following block diagram. The PHY comprises the PCS, optional Reed-Solomon FEC, and PMA.

**Figure 1. 25G Ethernet Intel FPGA IP MAC and PHY IP Clock Diagram**




The following block diagram shows an example of a network application with 25G Ethernet Intel FPGA IP MAC and PHY.

**Figure 2. Example Network Application**




## 1.1. 25G Ethernet Intel FPGA IP Core Supported Features

The 25G Ethernet Intel FPGA IP core is designed to the *25G & 50G Ethernet Specification, Draft 1.6* from the 25 Gigabit Ethernet Consortium and designed to the *IEEE 802.3by 25Gb Ethernet* specification, as well as the *IEEE 802.3ba-2012 High*

*Speed Ethernet Standard* available on the IEEE website (www.ieee.org). The MAC provides RX cut-through frame processing to optimize latency. The IP core supports the following features:

- PHY features:
  - Soft PCS logic that interfaces seamlessly to Intel Arria 10 FPGA 25.78125 gigabits per second (Gbps) serial transceivers.
  - Optional Reed-Solomon forward error correction (FEC).
- Frame structure control features:
  - Support for jumbo packets, defined as packets greater than 1500 bytes.
  - Receive (RX) CRC removal and pass-through control.
  - Transmit (TX) CRC generation and insertion.
  - RX and TX preamble pass-through option for applications that require proprietary user management information transfer.
  - TX automatic frame padding to meet the 64-byte minimum Ethernet frame length.
- Frame monitoring and statistics:
  - RX CRC checking and error reporting.
  - RX malformed packet checking per IEEE specification.
  - Optional statistics counters.
  - Optional fault signaling detects and reports local fault and generates remote fault, with *IEEE 802.3ba-2012 Ethernet Standard Clause 66* support.
  - Unidirectional transport as defined in *Clause 66 of the IEEE 802.3-2012 Ethernet Standard*.
- Flow control:
  - Standard *IEEE 802.3 Clause 31* and Priority-Based *IEEE 802.1Qbb* flow control.

- Precision Time Protocol support:
  - Optional support for the IEEE Standard 1588-2008 Precision Clock Synchronization Protocol (1588 PTP). This feature supports PHY operating speed with a constant timestamp accuracy of ± 3 ns and a dynamic timestamp accuracy of ± 1 ns.
- Debug and testability features:
  - Programmable serial PMA local loopback (TX to RX) at the serial transceiver for self-diagnostic testing.
  - TX error insertion capability.
  - Optional access to Altera Debug Master Endpoint (ADME) for serial link debugging or monitoring PHY signal integrity.
- User system interfaces:
  - Avalon Memory-Mapped (Avalon-MM) management interface to access the IP core control and status registers.
  - Avalon Streaming (Avalon-ST) data path interface connects to client logic.
  - Configurable ready latency of 0 or 3 clock cycles for Avalon-ST TX interface.
  - Hardware and software reset control.

For a detailed specification of the Ethernet protocol refer to the *IEEE 802.3 Ethernet Standard*.

**Related Information**

IEEE website
The *IEEE 802.3 Ethernet Standard* is available on the IEEE website.

## 1.2. 25G Ethernet Intel FPGA IP Core Device Family and Speed Grade Support

### 1.2.1. Device Family Support

**Table 1. Intel IP Core Device Support Levels**

| Device Support Level | Definition |
|---|---|
| **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 2. 25G Ethernet Intel FPGA IP Core Device Family Support**

Shows the level of support offered by the 25G Ethernet Intel FPGA IP core for each Intel device family.

| Device Family | Support |
|---|---|
| Intel Arria 10 GT | Default support level provided in the Intel Quartus® Prime software. Refer to the *Intel Quartus Prime Standard Edition Software and Device Support Release Notes* and the *Intel Quartus Prime Pro Edition Software and Device Support Release Notes*. |
| Other device families | Not supported. |

**Related Information**

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

### 1.2.2. 25G Ethernet Intel FPGA IP Core Device Speed Grade Support

**Table 3. Slowest Supported Device Speed Grades**

| IP Core | Device Family | Supported Speed Grades |
|---|---|---|
| 25G Ethernet Intel FPGA IP | Intel Arria 10 GT | E2 |

## 1.3. IP Core Verification

To ensure functional correctness of the 25G Ethernet Intel FPGA IP core, Intel performs extensive validation through both simulation and hardware testing. Before releasing a version of the 25G Ethernet Intel FPGA IP core, Intel runs comprehensive regression tests in the current version of the Intel Quartus Prime software.

Intel verifies that the current version of the Intel Quartus Prime software compiles the previous version of each IP core. Any exceptions to this verification are reported in the *Intel FPGA IP Release Notes*. Intel does not verify compilation with IP core versions older than the previous release.

**Related Information**

- Knowledge Base Issues for IP core
  Exceptions to functional correctness are documented in the 25G Ethernet Intel FPGA IP core errata.
- 25G Ethernet Intel FPGA IP Release Notes
- Intel Quartus Prime Design Suite Update Release Notes
  Includes changes in minor releases (updates).

### 1.3.1. Simulation Environment

Intel performs the following tests on the 25G Ethernet Intel FPGA 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.
- 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 25G Ethernet Intel FPGA IP core variations and designs to ensure the Intel Quartus Prime software places and routes the IP core ports correctly.

### 1.3.3. Hardware Testing

Intel performs hardware testing of the key functions of the 25G Ethernet Intel FPGA IP core using internal loopback and also with other 25G switches. The hardware tests also ensure reliable solution coverage for hardware related areas such as performance, link synchronization, and reset recovery.

## 1.4. Performance and Resource Utilization

The following table shows the typical device resource utilization for selected configurations using the current version of the Intel Quartus Prime software. With the exception of M20K memory blocks, the numbers of ALMs and logic registers are rounded up to the nearest 100. The timing margin for this IP core is a minimum of 15%.

**Table 4. IP Core Variation Encoding for Resource Utilization Table**

"On" indicates the parameter is turned on. The symbol "—" indicates the parameter is turned off or not available.

| IP Core Variation / Parameter | A | B | C |
|---|---|---|---|
| **Ready Latency** | 0 | 0 | 3 |
| **Enable RS-FEC** | — | On | — |
| **Enable flow control** | — | Standard flow control, 1 queue | Standard flow control, 1 queue |
| **Enable link fault generation** | — | — | On |
| **Enable preamble passthrough** | — | — | On |
| **Enable TX CRC passthrough** | On | — | — |
| **Enable MAC statistics counters** | — | On | On |
| **Enable IEEE 1588** | — | — | On |

**Table 5. IP Core FPGA Resource Utilization for 25G Ethernet Intel FPGA IP Core for Intel Arria 10 Devices**

Lists the resources and expected performance for selected variations of the 25G Ethernet Intel FPGA IP core.

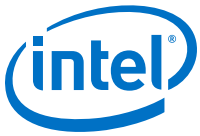
These results were obtained using the Intel Quartus Prime software v16.1.

- 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 Fitter Report.

| IP Core Variation | ALMs | Dedicated Logic Registers | M20K Memory Blocks |
|---|---|---|---|
| **A** | 3000 | 7200 | 0 |
| **B** | 13800 | 28700 | 29 |
| **C** | 11600 | 24900 | 34 |

## Related Information

- 25G Ethernet Intel FPGA IP Core Parameters on page 24
  Information about the parameters and values in the IP core variations.
- Fitter Resources Reports in the Quartus Prime Pro Edition Help



## 1.5. Release Information

### Table 6. 25G Ethernet Intel FPGA IP Core Current Release Information

| Item | Description |
|---|---|
| Version | 17.0 |
| Release Date | 2017.05.08 |
| Ordering Codes | IP-25GEUMACPHY (IPR-25GEUMACPHY for renewal) |

# 2. Getting Started

**Related Information**

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

## 2.1. Installing and Licensing Intel FPGA IP Cores

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

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

**Figure 3. IP Core Installation Path**

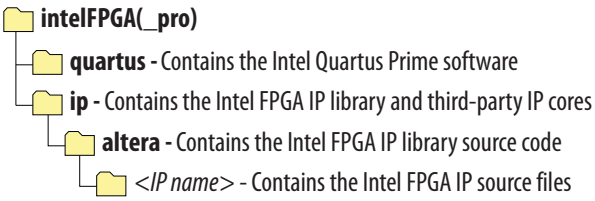
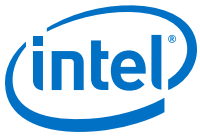



**Table 7. IP Core Installation Locations**

| Location | Software | Platform |
|---|---|---|
| *<drive>*:\intelFPGA_pro\quartus\ip\altera | Intel Quartus Prime Pro Edition | Windows* |
| *<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 |



### 2.1.1. Intel FPGA IP Evaluation Mode

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

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

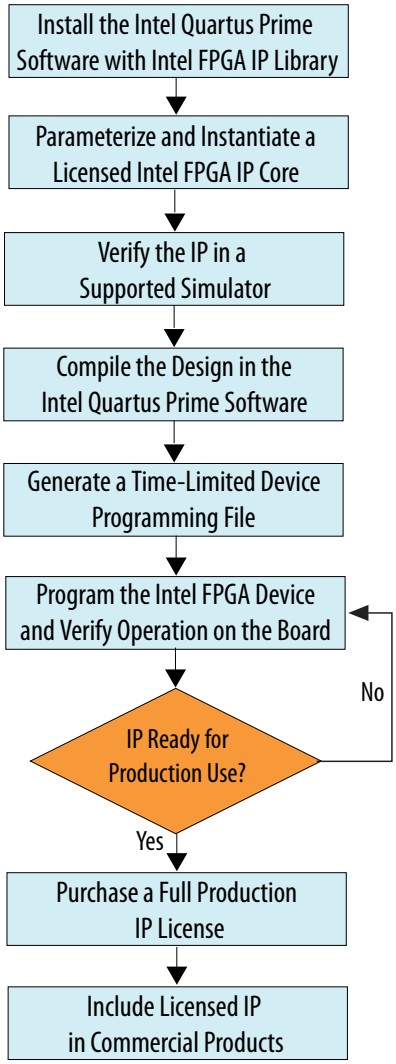
Intel FPGA IP Evaluation Mode supports the following operation modes:

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

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

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

**Figure 4. Intel FPGA IP Evaluation Mode Flow**




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

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

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

**Related Information**

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

## 2.2. Specifying the 25G Ethernet Intel FPGA IP Core Parameters and Options

The 25G Ethernet Intel FPGA IP parameter editor allows you to quickly configure your custom IP variation. Use the following steps to specify IP core options and parameters in the Intel Quartus Prime software.

1. Depending on whether you are using the Intel Quartus Prime Pro Edition software or the Intel Quartus Prime Standard Edition software, perform one of the following actions:
   - 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.
   - In the Intel Quartus Prime Standard Edition software, in the IP Catalog (**Tools ➤ IP Catalog**), select the Arria 10 target device family.
2. In the IP Catalog (**Tools ➤ IP Catalog**), locate and double-click the name of the IP core to customize. The New IP Variation window appears.
3. In the **New IP Variation** dialog box, specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named *<your_ip>*`.qsys` (in Intel Quartus Prime Standard Edition) or *<your_ip>*`.ip` (in Intel Quartus Prime Pro Edition).
4. In the Intel Quartus Prime Standard Edition software, you must select a specific Intel Arria 10 device in the **Device** field, or keep the default device the Quartus Prime software proposes.
5. Click **OK**. The parameter editor appears.
6. On the **IP** tab, specify the parameters for your IP core variation. Refer to 25G Ethernet Intel FPGA IP Core Parameters on page 24 for information about specific IP core parameters.
7. Optionally, to generate a simulation testbench or compilation and hardware design example, follow the instructions in the *Intel Arria 10 25G Ethernet Design Example User Guide*.
8. Click **Generate HDL**. The **Generation** dialog box appears.
9. Specify output file generation options, and then click **Generate**. The IP variation files generate according to your specifications.

   *Note:* A functional VHDL IP core is not available. Specify Verilog HDL only, for your IP core variation.
10. Click **Finish**. The parameter editor adds the top-level `.qsys` or `.ip` file to the current project automatically. If you are prompted to manually add the `.qsys` or `.ip` file to the project, click **Project ➤ Add/Remove Files in Project** to add the file.
11. After generating and instantiating your IP variation, make appropriate pin assignments to connect ports.

### Related Information

Intel Arria 10 25G Ethernet Design Example User Guide
Information about the **Example Design** tab in the 25G Ethernet Intel FPGA IP parameter editor.

## 2.3. Simulating the IP Core

You can simulate your 25G Ethernet Intel FPGA IP core variation with the functional simulation model and the testbench generated with the IP core. The functional simulation model is a cycle-accurate model that allows for fast functional simulation of your IP core instance using industry-standard Verilog HDL simulators. You can simulate the Intel-provided testbench or create your own testbench to exercise the IP core functional simulation model.

The functional simulation model and testbench files are generated in project subdirectories. These directories also include scripts to compile and run the design example.

*Note:* Use the simulation models only for simulation and not for synthesis or any other purposes. Using these models for synthesis creates a nonfunctional design.

In the top-level wrapper file for your simulation project, you can set the the following RTL parameters to enable simulation optimization. These optimizations significantly decrease the time to reach link initialization.

- SIM_SHORT_RST: Shortens the reset times to speed up simulation.
- SIM_SHORT_AM: Shortens the interval between alignment markers to accelerate alignment marker lock. Alignment markers are used when Reed-Solomon FEC is enabled.
- SIM_SIMPLE_RATE: Sets the PLL reference clock (`clk_ref`) to 625 MHz instead of 644.53125 MHz to optimize PLL simulation model behavior

In general, parameters are set through the IP core parameter editor and you should not change them manually. The only exceptions are these simulation optimization parameters.

To set these parameters on the PHY blocks, add the following lines to the top-level wrapper file:

```
defparam <dut instance>.SIM_SHORT_RST = 1'b1;
defparam <dut instance>.SIM_SHORT_AM = 1'b1;
defparam <dut instance>.SIM_SIMPLE_RATE = 1'b1;
```

*Note:* You can use the example testbench as a guide for setting the simulation parameters in your own simulation environment. These lines are already present in the Intel-provided testbench for the IP core.

### Related Information

- Simulating Intel FPGA Designs
  *Quartus Prime Standard Edition Handbook Volume 3: Verification* chapter that provides information about simulating Intel FPGA IP cores.

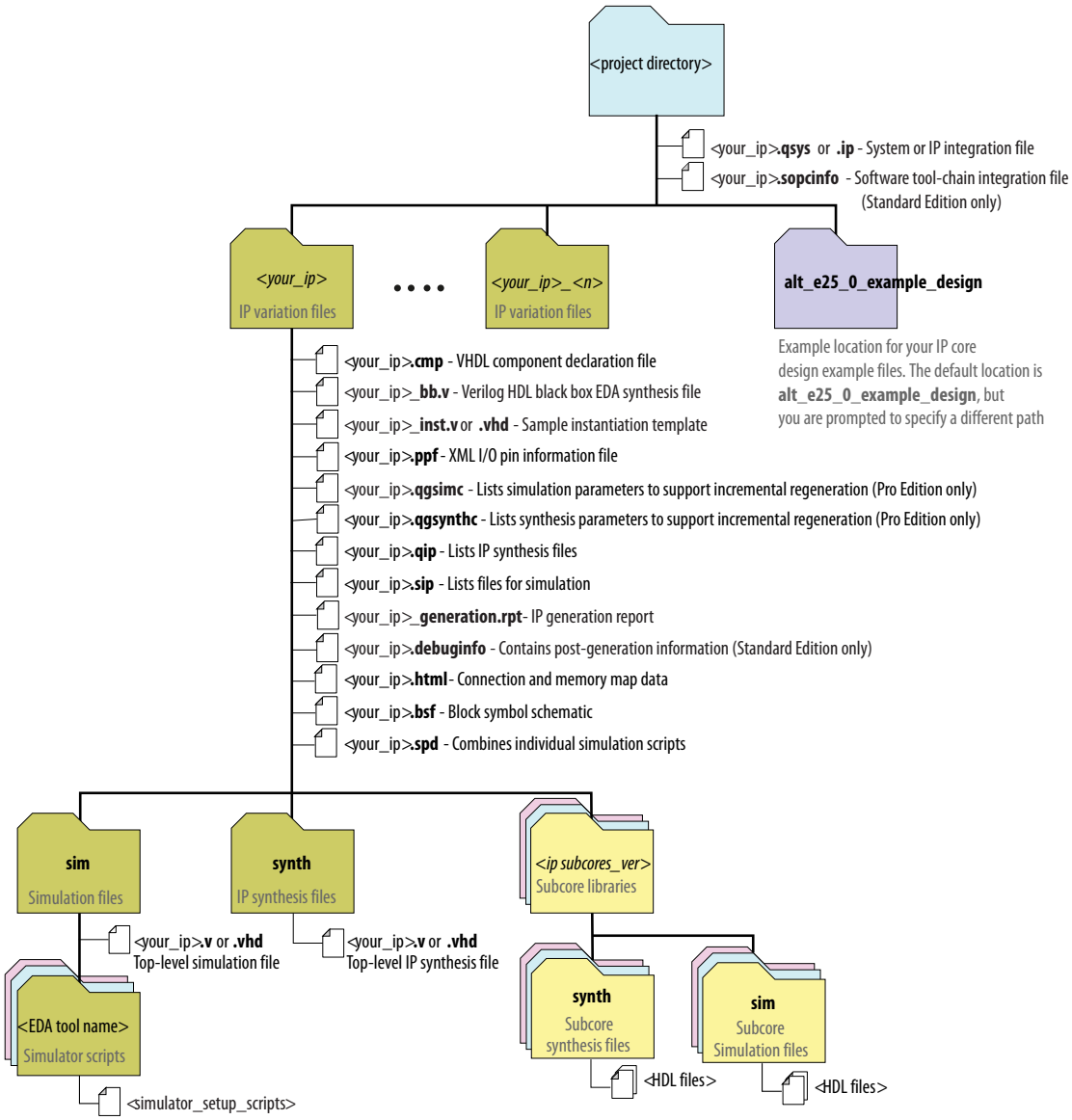
- Intel Arria 10 25G Ethernet Design Example User Guide
  Information about generating and simulating the Intel-provided 25G Ethernet Intel FPGA IP testbench. This testbench demonstrates a basic test of the IP core. It is not intended to be a substitute for a full verification environment.

## 2.4. Generated File Structure

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

For information about the file structure of the design example, refer to the *Arria 10 25G Ethernet Intel FPGA IP Design Example User Guide*.

**Figure 5. IP Core Generated Files**



## Table 8. IP Core Generated Files

| File Name | Description |
|---|---|
| <*your_ip*>.qsys (Intel Quartus Prime Standard Edition only) | The Platform Designer system or top-level IP variation file. <*your_ip*> is the name that you give your IP variation. |
| <*your_ip*>.ip (Intel Quartus Prime Pro Edition only) | |
| <*system*>.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. (Intel Quartus Prime Standard Edition only)<br>Downstream tools such as the Nios® II Gen 2 tool chain use this file. The .sopcinfo file and the system.h file generated for the Nios II Gen 2 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*>.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.<br>This IP core does not support VHDL. However, the Intel Quartus Prime software generates this file. |
| <*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*>.debuginfo | Contains post-generation information. Used to pass System Console and Bus Analyzer Toolkit information about the Platform Designer interconnect. The Bus Analysis Toolkit uses this file to identify debug components in the Platform Designer interconnect. (Intel Quartus Prime Standard Edition only) |
| <*your_ip*>.qgsimc | Lists simulation parameters to support incremental regeneration. (Intel Quartus Prime Pro Edition only) |
| <*your_ip*>.qgsynthc | Lists synthesis parameters to support incremental regeneration. (Intel Quartus Prime Pro Edition only) |
| <*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*>.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 Intel 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*>.sip | Contains information required for NativeLink simulation of IP components. You must add the **.sip** file to your Quartus Prime project. (Intel Quartus Prime Standard Edition only) |
| <*your_ip*>_inst.v and _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.<br>This IP core does not support VHDL. However, the Intel Quartus Prime software generates the _inst.vhd file. |

*continued...*

| File Name | Description |
|---|---|
| <*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 within a Platform Designer system.<br>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<br>and<br><*your_ip*>.vhd | HDL files that instantiate each submodule or child IP core for synthesis or simulation.<br>This IP core does not support VHDL. However, the Intel Quartus Prime software generates this file. |
| mentor/ | Contains a ModelSim script msim_setup.tcl to set up and run a simulation. |
| aldec/ | Contains a Riviera-PRO script rivierapro_setup.tcl to setup and run a simulation. |
| synopsys/vcs/<br>synopsys/vcsmx/ | Contains a shell script vcs_setup.sh to set up and run a VCS® simulation.<br>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 submodule. |
| <*child IP cores*>/ | For each generated child IP core directory, Platform Designer generates synth/ andsim/ sub-directories. |

**Related Information**

Intel Arria 10 25G Ethernet Design Example User Guide
Information about the 25G Ethernet Intel FPGA IP design example file structure.

## 2.5. Integrating Your IP Core in Your Design

### 2.5.1. Pin Assignments

When you integrate your 25G Ethernet Intel FPGA IP core instance in your design, you must make appropriate pin assignments. While compiling the IP core alone, you can create virtual pins to avoid making specific pin assignments for top-level signals. When you are ready to map the design to hardware, you can change to the correct pin assignments.

**Related Information**

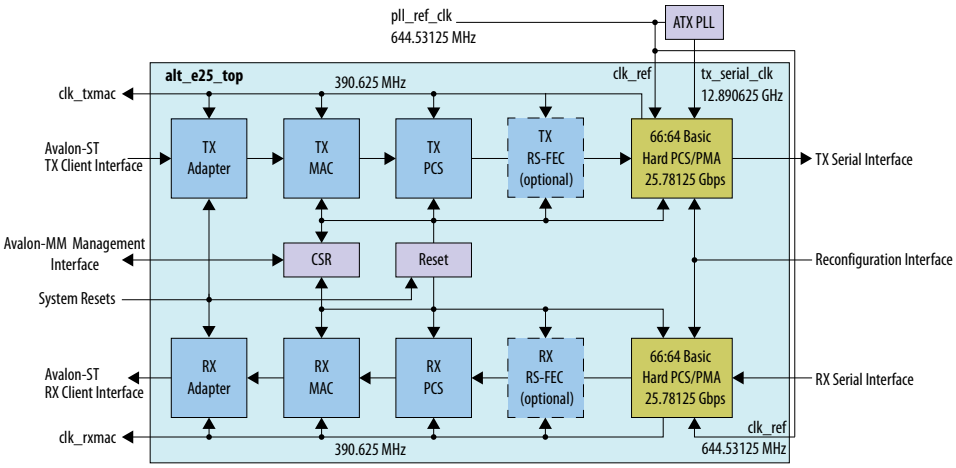
Intel Quartus Prime Help
For information about the Intel Quartus Prime software, including virtual pins.

## 2.5.2. Adding the Transceiver PLL

The 25G Ethernet Intel FPGA IP core targets Arria 10 GT devices. Intel Arria 10 GT devices require an external PLL to drive the TX transceiver serial clock, in order to compile and to function correctly in hardware. In many cases, the same PLL can be shared with an additional transceiver in your design.

**Figure 6. PLL Configuration Example**

The TX transceiver PLL is instantiated with an ATX PLL IP core. The TX transceiver PLL must always be instantiated outside the 25G Ethernet Intel FPGA IP core.




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

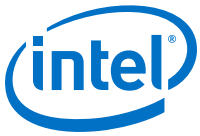
- Select **Intel Arria 10 Transceiver ATX PLL**.
- In the parameter editor, set the following parameter values:
  - **PLL output frequency** to **12890.625 MHz**. The transceiver performs dual edge clocking, using both the rising and falling edges of the input clock from the PLL. Therefore, this PLL output frequency setting supports a 25.78125 Gbps data rate through the transceiver.
  - **PLL reference clock frequency** to **644.53125** MHz.

You must connect the ATX PLL to the 25G Ethernet Intel FPGA IP core as follows:

- Connect the clock output port of the ATX PLL to the `tx_serial_clk` input port of the 25G Ethernet Intel FPGA IP core.
- Connect the `pll_locked` output port of the ATX PLL to the `tx_pll_locked` input port of the 25G Ethernet Intel FPGA IP core.
- Drive the ATX PLL reference clock port and the 25G Ethernet Intel FPGA IP core `clk_ref` input port with the same clock. The clock frequency must be the frequency you specify for the ATX PLL IP core **PLL reference clock frequency** parameter.

### Related Information

- Transceivers on page 56



- Intel Arria 10 Transceiver PHY User Guide
  Information about the correspondence between PLLs and transceiver channels, and information about how to configure an external transceiver PLL for your own design. You specify the clock network to which the PLL output connects by setting the clock network in the PLL parameter editor.

## 2.5.3. Handling Potential Jitter in Intel Arria 10 Devices

The RX path in the 25G Ethernet Intel FPGA IP core includes cascaded PLLs. Therefore, the IP core clocks might experience additional jitter in Intel Arria 10 devices.

Refer to the KDB Answer *How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Arria 10 PLL reference clock?* for a workaround you should apply to the IP core, in your design.

### Related Information

https://www.altera.com/support/support-resources/knowledge-base/tools/2017/fb470823.html
KDB Answer: How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Arria 10 PLL reference clock?

## 2.5.4. Adding the External Time-of-Day Module for Variations with 1588 PTP Feature

25G Ethernet Intel FPGA IP cores that include the 1588 PTP module require an external time-of-day (TOD) module to provide a continuous flow of current time-of-day information. The TOD module must update the time-of-day output value on every clock cycle, and must provide the TOD value in the V2 format (96 bits) or the 64-bit TOD format, or both.

Intel provides the following components that you can combine to create the TOD module the 25G Ethernet Intel FPGA IP core requires:

- A simple TOD clock module, available from the IP Catalog (**Interface Protocols > Ethernet > Reference Design Components > Ethernet IEEE 1588 Time of Day Clock Intel FPGA IP**). You can instantiate two of these clock modules and connect one to the TX MAC and the other to the RX MAC.
- A single-format TOD synchronizer, available from the IP Catalog (**Interface Protocols > Ethernet > Reference Design Components > Ethernet IEEE 1588 TOD Synchronizer Intel FPGA IP**). This component can handle only a single TOD format. Therefore, if you set the **Time of day format** parameter to the value of **Enable both formats**, you must instantiate and connect two TOD synchronizer modules. If your IP core supports only a single TOD format, your design requires only a single TOD synchronizer module.

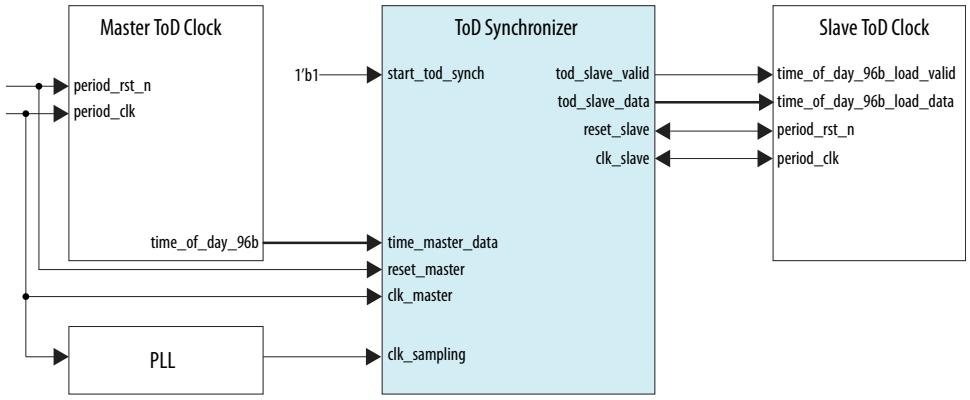
Each TOD synchronizer connects a master TOD clock and a slave TOD clock.

- If you create your TOD module with a single TOD synchronizer, the master TOD clock connects to the TX MAC of the 25G Ethernet Intel FPGA IP core and the slave TOD clock connects to the RX MAC of the 25G Ethernet Intel FPGA IP core.
- Alternatively, you can drive both the TX and RX TOD clocks from a single master TOD clock. In that case, your design must include two TOD synchronizers, one to connect the master TOD clock and the slave TX TOD clock and one to connect the master TOD clock and the slave RX TOD clock.

If your IP core supports both TOD formats, double the number of TOD synchronizers in your TOD module. The configuration you implement depends on your system design requirements for 1588 PTP functionality.

**Figure 7. TOD Synchronizer and TOD Clocks in 96-Bit TOD Format Design**

Shows the required connections between two TOD clock components and a TOD synchronizer component in a single TOD format design. In a simple TOD module, the master TOD clock connects to the TX MAC of the IP core, and the slave TOD clock connects to the RX MAC of the IP core. If your 25G Ethernet Intel FPGA IP core supports both TOD formats, a second TOD synchronizer connects to the corresponding 64-bit time-of-day signals of the same master and slave TOD clocks.




For information about the Ethernet IEEE 1588 Time of Day Clock and Ethernet IEEE 1588 TOD Synchronizer components, and the requirements for the PLL that connects to the TOD synchronizer, refer to the *Ethernet Design Example Components User Guide*.

**Table 9. TOD Module Required Connections to 25G Ethernet Intel FPGA IP Core**

Lists the required connections between the TOD module and the 25G Ethernet Intel FPGA IP core, using signal names for TOD modules that provide both a 96-bit TOD and a 64-bit TOD. If you create your own TOD module it must have the output signals required by the 25G Ethernet Intel FPGA IP core. However, its signal names could be different than the TOD module signal names in the table. The signals that the IP core includes depend on the value you set for **Time of day format** in the parameter editor. For example, an RX TOD module might require only a 96-bit TOD out signal. This table does not list required connections between the TOD module and additional parts of your design.

| TOD Module Signal | 25GbE IP Core Signal |
|---|---|
| `rst_n` (input to TX and RX TOD clocks) | Drive this signal from the same source as the `csr_rst_n` input signal to the 25G Ethernet Intel FPGA IP core. |
| `period_rst_n` (input to RX TOD clock)<br>`reset_slave` (input to Synchronizer) | Drive these signals from the same source as the `rx_rst_n` input signal to the 25G Ethernet Intel FPGA IP core. |
| `period_rst_n` (input to TX TOD clock)<br>`reset_master` (input to Synchronizer) | Drive these signals from the same source as the `tx_rst_n` input signal to the 25G Ethernet Intel FPGA IP core. |
| `time_of_day_96b[95:0]` (output from TX TOD clock) | `tx_time_of_day_96b_data[95:0]` (input) |
| `time_of_day_64b[63:0]` (output from TX TOD clock) | `tx_time_of_day_64b_data[63:0]` (input) |
| `time_of_day_96b[95:0]` (output from RX TOD clock) | `rx_time_of_day_96b_data[95:0]` (input) |
| `time_of_day_64b[63:0]` (output from RX TOD clock) | `rx_time_of_day_64b_data[63:0]` (input) |
| `period_clk` (input to TX TOD clock) | `clk_txmac` (output) |

*continued...*

| TOD Module Signal | 25GbE IP Core Signal |
|---|---|
| clk_master (input to Synchronizer) | |
| period_clk (input to RX TOD clock)<br>clk_slave (input to Synchronizer) | clk_rxmac (output) |

**Related Information**

- External Time-of-Day Module for 1588 PTP Variations on page 47
- Ethernet Design Example Components User Guide
  Describes the Ethernet IEEE 1588 Time of Day Clock component and the Ethernet IEEE 1588 TOD Synchronizer component available in the Intel Quartus Prime software from the IP Catalog.

## 2.5.5. Placement Settings for the 25G Ethernet Intel FPGA IP Core

The Quartus Prime software provides the options to specify design partitions and Logic Lock (Standard) or Logic Lock regions for incremental compilation, 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.

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

**Related Information**

- Quartus Prime Standard Edition Handbook Volume 2: Design Implementation and Optimization
  Describes incremental compilation, design partitions, and Logic Lock (Standard) regions.
- Intel Quartus Prime Pro Edition Handbook Volume 2: Design Implementation and Optimization
  Describes incremental compilation, design partitions, and Logic Lock regions.

# 2.6. Compiling the Full Design and Programming the FPGA

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

*Note:* The 25G Ethernet Intel FPGA IP core design example synthesis directories include Synopsys Constraint (`.sdc`) files that you can copy and modify for your own design.

*Note:* For additional `.sdc` file requirements, please refer to the KDB Answer at https://www.altera.com/support/support-resources/knowledge-base/tools/2017/fb470823.html.

**Related Information**

- Incremental Compilation for Hierarchical and Team-Based Design
- Programming Intel Devices
- 25G Ethernet Intel Arria 10 FPGA IP Design Example User Guide
  Information about generating the design example and the design example directory structure.

# 3. 25G Ethernet Intel FPGA IP Core Parameters

The 25G Ethernet Intel FPGA IP parameter editor provides the parameters you can set to configure the 25G Ethernet Intel FPGA IP core and simulation testbenches.

The 25G Ethernet Intel FPGA IP parameter editor includes an **Example Design** tab. For information about that tab, refer to the *25G Ethernet Design Example User Guide*.

**Table 10. IP Core Parameters**

| Parameter | Range | Default Setting | Description |
|---|---|---|---|
| **General Options** | | | |
| **Device Family** | **Arria 10** | **Arria 10** | Selects the device family. |
| **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 `ll_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*.<br>Selecting a latency of 3 eases timing closure at the expense of increased latency for the datapath.<br>If you set the readyLatency to 3 and turn on standard flow control, data might be delayed in the IP core while the IP core is backpressured. |
| **PCS/PMA Options** | | | |
| **Enable RS-FEC** | **Enabled**, **Disabled** | **Disabled** | When enabled, the IP core implements Reed-Solomon forward error correction (FEC).<br>This parameter is not available if you turn on **Enable IEEE 1588**. |
| **Flow Control Options** | | | |
| **Enable flow control** | **Enabled**, **Disabled** | **Disabled** | When enabled, the IP core implements flow control. When either link partner experiences congestion, the respective transmit control sends pause frames. Register settings control flow control behavior, including whether the IP core implements standard flow control or priority-based flow control.<br>If you turn on standard flow control and set the readyLatency to 3, data might be delayed in the IP core while the IP core is backpressured. |
| **Number of queues** | **1**-**8** | **8** | Specifies the number of queues used in managing flow control. |
| **MAC Options** | | | |
| **Enable link fault generation** | **Enabled**, **Disabled** | **Disabled** | When enabled, the IP core implements link fault signaling as defined in the *IEEE 802.3-2012 IEEE Standard for Ethernet*. The MAC includes a Reconciliation Sublayer (RS) to manage local and remote faults. When enabled, the local RS TX logic can |

***continued...***

| Parameter | Range | Default Setting | Description |
|---|---|---|---|
| | | | transmit remote fault sequences in case of a local fault and can transmit IDLE control words in case of a remote fault. |
| **Enable preamble passthrough** | **Enabled**, **Disabled** | **Disabled** | When enabled, 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 Start Frame Delimiter (SFD) to the client instead of stripping them out of the Ethernet packet. In TX preamble pass-through mode, the client specifies the preamble and provides the SFD to be sent in the Ethernet frame. |
| **Enable TX CRC passthrough** | **Enabled**, **Disabled** | **Disabled** | When enabled, TX MAC does not insert the CRC-32 checksum in the out-going frame. In pass-through mode, the client must provide frames with at least 64 bytes, including the Frame Check Sequence (FCS). When disabled, the TX MAC computes and inserts a 32-bit FCS in the TX MAC frame.<br>This parameter is not available if you turn on **Enable IEEE 1588**. |
| **Enable MAC statistics counters** | **Enabled**, **Disabled** | **Enabled** | When enabled, the IP core includes statistics counters that characterize TX and RX traffic. |
| **IEEE 1588 Options** | | | |
| **Enable IEEE 1588** | **Enabled**, **Disabled** | **Disabled** | If enabled, the IP core supports the IEEE Standard 1588-2008 Precision Clock Synchronization Protocol, by providing the hooks to implement the Precise Timing Protocol (PTP).<br>This parameter is not available if you turn on **Enable TX CRC passthrough**.<br>This parameter is not available if you turn on **Enable RS-FEC**. |
| **Time of day format** | **Enable 96-bit timestamp format**, **Enable 64-bit timestamp format**, **Enable both formats** | **Enable both formats** | Specifies the interface to the Time of Day module. If you select **Enable both formats**, the IP core includes both the 64-bit interface and the 96-bit interface.<br>This parameter is available only in variations with **Enable IEEE 1588** turned on. The IP core provides the Time of Day interface; the IP core does not include Time of Day and synchronizer modules to connect to this interface. |
| **Fingerprint width** | **1–32** | **4** | Specifies the number of bits in the fingerprint that the IP core handles.<br>This parameter is available only in variations with **Enable IEEE 1588** turned on. |
| **Configuration, Debug and Extension Options** | | | |
| **Enable Altera Debug Master Endpoint (ADME)** | **Enabled**, **Disabled** | **Disabled** | If enabled, the IP core turns on the following features in the Arria 10 PHY IP core that is included in the 25G Ethernet Intel FPGA IP core:<br>• **Enable Altera Debug Master Endpoint (ADME)**<br>• **Enable capability registers**<br>If turned off, the IP core is configured without these features.<br>For information about these Arria 10 features, refer to the *Arria 10 Transceiver PHY User Guide*. . |

## Related Information

- 25G Ethernet Design Example User Guide
  Information about the **Example Design** tab in the 25G Ethernet Intel FPGA IP parameter editor.