1. 25G Ethernet Intel® FPGA IP Quick Start Guide .............................................................. 4
   1.1. Directory Structure ................................................................................................ 5
   1.2. Generating the Design Example ............................................................................... 6
       1.2.1. Design Example Parameters ........................................................................ 7
   1.3. Simulating the 25G Ethernet Intel FPGA IP Design Example Testbench ..................... 8
       1.3.1. Procedure ...................................................................................................... 8
   1.4. Compiling and Configuring the Design Example in Hardware ....................................... 8
       1.4.1. Procedure ...................................................................................................... 8
   1.5. Changing Target Device in Hardware Design Example ................................................. 9
       1.5.1. Procedure ...................................................................................................... 9
   1.6. Testing the 25G Ethernet Intel FPGA IP Design in Hardware ....................................... 10
       1.6.1. Procedure ...................................................................................................... 10

2. 10G/25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices ............. 12
   2.1. Features ................................................................................................................ 12
   2.2. Hardware and Software Requirements .................................................................... 12
   2.3. Functional Description .......................................................................................... 12
       2.3.1. Design Components ...................................................................................... 15
   2.4. Simulation ........................................................................................................... 16
       2.4.1. Testbench ...................................................................................................... 16
       2.4.2. Simulation Design Example Components ...................................................... 17
       2.4.3. Test Case—Design Example Without the IEEE 1588v2 Feature ....................... 17
       2.4.4. Test Case—Design Example with the IEEE 1588v2 Feature ............................. 19
   2.5. Compilation ........................................................................................................... 22
   2.6. Hardware Testing .................................................................................................. 22
       2.6.1. Test Procedure—Design Example Without the IEEE 1588v2 Feature ............... 23
       2.6.2. Test Procedure—Design Example with the IEEE 1588v2 Feature .................... 26

3. 25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices .................. 27
   3.1. Features ................................................................................................................ 27
   3.2. Hardware and Software Requirements .................................................................... 27
   3.3. Functional Description .......................................................................................... 27
       3.3.1. Design Components ...................................................................................... 30
   3.4. Simulation ........................................................................................................... 31
       3.4.1. Testbench ...................................................................................................... 31
       3.4.2. Simulation Design Example Components ...................................................... 32
       3.4.3. Test Case—Design Example Without the IEEE 1588v2 Feature ....................... 32
       3.4.4. Test Case—Design Example with the IEEE 1588v2 Feature ............................. 33
   3.5. Compilation ........................................................................................................... 36
   3.6. Hardware Testing .................................................................................................. 37
       3.6.1. Test Procedure—Design Example With and Without the IEEE 1588v2 Feature ... 37

4. 25G Ethernet Multi-Channel Design Example for Intel Stratix 10 Devices .................. 38
   4.1. Features ................................................................................................................ 38
   4.2. Hardware and Software Requirements .................................................................... 38
   4.3. Functional Description .......................................................................................... 38
       4.3.1. Design Components ...................................................................................... 39
   4.4. Simulation ........................................................................................................... 40
4.4.1. Testbench......................................................................................................................... 40
4.4.2. Simulation Design Example Components........................................................................... 41
4.4.3. Test Case.......................................................................................................................... 41
4.5. Compilation.......................................................................................................................... 44
4.6. Hardware Testing............................................................................................................... 45
    4.6.1. Test Procedure................................................................................................................ 45

5. 25G Ethernet Intel FPGA IP Design Example References...................................................... 46
    5.1. Design Example Interface Signals....................................................................................... 46
    5.2. Design Example Registers................................................................................................. 47
    5.3. Using Transceiver Toolkit on H-Tile Production Device...................................................... 48

6. 25G Ethernet Intel Stratix 10 FPGA IP Design Example User Guide Archives...................... 50

1. 25G Ethernet Intel® FPGA IP Quick Start Guide

The 25G Ethernet (25GbE) Intel® FPGA IP core for Intel Stratix® 10 devices provides the capability of generating design examples for selected configurations.

Figure 1. Development Stages for the Design Example

Related Information
- **10G/25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices** on page 12
  Provides details for the 10G/25G Ethernet single-channel design example.
- **25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices** on page 27
  Provides details for the 25G Ethernet single-channel design example.
- **25G Ethernet Multi-Channel Design Example for Intel Stratix 10 Devices** on page 38
  Provides details for the 25G Ethernet multi-channel design example.
1.1. Directory Structure

Figure 2. Directory Structure for the 25G and 10G/25G Ethernet Design Examples

- The simulation files (testbench for simulation only) are located in 
  <design_example_dir>/example_testbench.
- The compilation-only design example is located in <design_example_dir>/compilation_test_design.
- The hardware configuration and test files (the design example in hardware) are located in <design_example_dir>/hardware_test_design.

Table 1. Directory and File Descriptions

<table>
<thead>
<tr>
<th>File Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth_ex_25g.qpf</td>
<td>Intel Quartus® Prime project file.</td>
</tr>
<tr>
<td>eth_ex_25g.qsf</td>
<td>Intel Quartus Prime project settings file.</td>
</tr>
<tr>
<td>eth_ex_25g.sdc</td>
<td>Synopsys Design Constraints file. You can copy and modify this file for your own 25GbE Intel FPGA IP core design.</td>
</tr>
<tr>
<td>eth_ex_25g.v</td>
<td>Top-level Verilog HDL design example file.</td>
</tr>
<tr>
<td>common/</td>
<td>Hardware design example support files.</td>
</tr>
<tr>
<td>hwtest/main.tcl</td>
<td>Main file for accessing System Console.</td>
</tr>
</tbody>
</table>
1.2. Generating the Design Example

Follow these steps to generate the hardware design example and testbench:

1. In the Intel Quartus Prime Pro Edition software, 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.

2. In the IP Catalog, locate and select 25G Ethernet Intel FPGA IP. The New IP Variation window appears.

3. Specify a top-level name for your IP variation and click OK. 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.

4. In the Intel Quartus Prime Pro Edition software, you must select a specific Intel Stratix 10 device in the Device field, or keep the default device that the Intel Quartus Prime software proposes.

   Note: The hardware design example overwrites the selection with the device on the target board. You specify the target board from the menu of design example options in the Example Design tab (Step 8).

5. Click OK. The parameter editor appears.

6. On the IP tab, specify the parameters for your IP core variation.

7. On the Example Design tab, for Example Design Files, select the Simulation option to generate the testbench, and select the Synthesis option to generate the hardware design example. Only Verilog HDL files are generated.
Note: A functional VHDL IP core is not available. Specify Verilog HDL only, for your IP core design example.

8. For Target Development Kit, select the Stratix 10 GX Signal Integrity L-Tile (Prod) Development Kit.

Note: The target device of the generated hardware example design is for Intel Stratix 10 GX Signal Integrity L-Tile (Production) Development Kit (1SX280LU2F50E1VG) and may differ from your selected device. The target device can be changed after hardware design example generation has completed. For the procedure to change the target device, refer to Changing Target Device in Hardware Design Example.


10. If you want to modify the design example directory path or name from the defaults displayed (alt_e25s10_0_example_design), browse to the new path and type the new design example directory name (<design_example_dir>).

11. Click OK.

Related Information
Changing Target Device in Hardware Design Example on page 9

1.2.1. Design Example Parameters

Table 2. Parameters in the Example Design Tab

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example Design</td>
<td>Available example designs for the IP parameter settings.</td>
</tr>
<tr>
<td>Example Design Files</td>
<td>The files to generate for the different development phase.</td>
</tr>
<tr>
<td></td>
<td>• Simulation—generates the necessary files for simulating the example design.</td>
</tr>
<tr>
<td></td>
<td>• Synthesis—generates the synthesis files. Use these files to compile the</td>
</tr>
<tr>
<td></td>
<td>design in the Intel Quartus Prime Pro Edition software for hardware testing</td>
</tr>
<tr>
<td></td>
<td>and perform static timing analysis.</td>
</tr>
<tr>
<td>Generate File Format</td>
<td>The format of the RTL files for simulation—Verilog.</td>
</tr>
<tr>
<td>Select Board</td>
<td>Supported hardware for design implementation. When you select an Intel FPGA</td>
</tr>
<tr>
<td></td>
<td>development board, use device 1SX280LU2F50E1VG as the Target Device for</td>
</tr>
<tr>
<td></td>
<td>design example generation.</td>
</tr>
<tr>
<td></td>
<td>If this menu is not available, there is no supported board for the</td>
</tr>
<tr>
<td></td>
<td>options that you select.</td>
</tr>
<tr>
<td>Stratix 10 GX Signal Integrity L-Tile (Prod) Development Kit: This option</td>
<td>This option allows you to test the design example on the selected Intel FPGA</td>
</tr>
<tr>
<td></td>
<td>IP development kit. This option automatically selects the Target Device</td>
</tr>
<tr>
<td></td>
<td>of 1SX280LU2F50E1VG. If your board revision has a different device grade,</td>
</tr>
<tr>
<td></td>
<td>you can change the target device.</td>
</tr>
<tr>
<td>None: This option excludes</td>
<td>This option excludes the hardware aspects for the design example.</td>
</tr>
<tr>
<td>the hardware aspects for the design example.</td>
<td></td>
</tr>
</tbody>
</table>
1.3. Simulating the 25G Ethernet Intel FPGA IP Design Example Testbench

1.3.1. Procedure

You can compile and simulate the design by running a simulation script from the command prompt.

1. At the command prompt, change the working directory to `<design_example_dir>/example_testbench`.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator.

<table>
<thead>
<tr>
<th>Simulator</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModelSim*</td>
<td>In the command line, type <code>vsim -do run_vsim.do</code>. If you prefer to simulate without bringing up the ModelSim GUI, type <code>vsim -c -do run_vsim.do</code>. Note: The ModelSim-AE and ModelSim-ASE simulators cannot simulate this IP core. You must use another supported ModelSim simulator such as ModelSim SE.</td>
</tr>
<tr>
<td>VCS*</td>
<td>In the command line, type <code>sh run_vcs.sh</code></td>
</tr>
<tr>
<td>NCSim</td>
<td>In the command line, type <code>sh run_ncsim.sh</code></td>
</tr>
<tr>
<td>Xcelium*</td>
<td>In the command line, type <code>sh run_xcelium.sh</code></td>
</tr>
</tbody>
</table>

A successful simulation ends with the following message:

```
Simulation Passed.
```

or

```
Testbench complete.
```

After successful completion, you can analyze the results.

1.4. Compiling and Configuring the Design Example in Hardware

The 25G Ethernet Intel FPGA IP core parameter editor allows you to compile and configure the design example on a target development kit.

1.4.1. Procedure
To compile and configure a design example on hardware, follow these steps:

1. Launch the Intel Quartus Prime Pro Edition software and select **Processing > Start Compilation** to compile the design.

2. After you generate an SRAM object file `.sof`, follow these steps to program the hardware design example on the Intel Stratix 10 device:
   a. On the **Tools** menu, click **Programmer**.
   b. In the Programmer, click **Hardware Setup**.
   c. Select a programming device.
   d. Select and add the Intel Stratix 10 GX board to your Intel Quartus Prime Pro Edition session.
   e. Ensure that **Mode** is set to **JTAG**.
   f. Select the Intel Stratix 10 device and click **Add Device**. The Programmer displays a block diagram of the connections between the devices on your board.
   g. In the row with your `.sof`, check the box for the `.sof`.
   h. Check the box in the **Program/Configure** column.
   i. Click **Start**.

*Note:* This design targets the Intel Stratix 10 device. Please contact your Intel FPGA representative to inquire about a platform suitable to run this hardware example.

**Related Information**
- Incremental Compilation for Hierarchical and Team-Based Design
- Programming Intel FPGA Devices

### 1.5. Changing Target Device in Hardware Design Example

If you have selected **Intel Stratix 10 GX Signal Integrity L-Tile (Production) Development Kit** as your target device, the 25G Ethernet Intel FPGA IP core generates a hardware example design for target device 1SX280LU2F50E1VG. This device may differ from the device on your development kit.

#### 1.5.1. Procedure

To change the target device in your hardware design example, follow these steps:

1. Launch the Intel Quartus Prime Pro Edition software and open the hardware test project file `/hardware_test_design/eth_ex_25g.qpf`.
2. On the **Assignments** menu, click **Device**. The **Device** dialog box appears.
3. In the **Device** dialog box, select 1SG280LU2F50E2VG (L-tile) or 1SG280HU1F50E2VG (H-tile) in the target device table that matches the device part number on your development kit. Refer to the **Stratix 10 GX Signal Integrity Development Kit** link on the Intel website for more information.
4. A prompt appears when you select a device, as shown in the figure below. Select **No** to preserve the generated pin assignments and I/O assignments.
5. If you select 1SG280HU2F50E2VG (H-Tile GX) as your target device, click **Upgrade IP Components** in the **Project** menu, select **25G Ethernet FPGA IP** from the list of IP components, and click **Upgrade in Editor**. Regenerate this IP component.

   *Note:* If you select 1SG280LU1F50E2VG (L-Tile GX) as your target device, skip this step if you are using the same Quartus and IP version.

6. Modify the pin assignment of `cpu_resetn` port to pin `AW10` through Pin Planner or Assignment Editor in the Intel Quartus Prime Pro Edition. No other pin assignment modifications are required for the design example. When you generate the design example targeting other Intel Stratix 10 development kits, refer to the respective development kit user guides for pin assignment.

7. Perform full compilation of your design.

   You can now test the design on your hardware.

**Related Information**
- Stratix 10 GX Signal Integrity Development Kit
- Intel Stratix 10 GX Transceiver Signal Integrity Development Kit User Guide

### 1.6. Testing the 25G Ethernet Intel FPGA IP Design in Hardware

#### 1.6.1. Procedure

After you compile the 25G Ethernet Intel FPGA IP core design example and configure it on your Intel Stratix 10 device, you can use the System Console to program the IP core and its embedded Native PHY IP core registers.

To turn on the System Console and test the hardware design example, follow these steps:

1. In the Intel Quartus Prime Pro Edition software, select **Tools ➤ System Debugging Tools ➤ System Console** to launch the system console.

2. In the Tcl Console pane, type `cd hwtest` to change directory to `/hardware_test_design/hwtest`.

3. Type `source main.tcl` to open a connection to the JTAG master.

Follow the test procedure in the **Hardware Testing** section of the design example and observe the test results in the System Console.
Related Information

Analyzing and Debugging Designs with System Console
2. 10G/25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices

The 10G/25G Ethernet single-channel design example demonstrates an Ethernet solution for Intel Stratix 10 devices using the 25G Ethernet Intel FPGA IP core.

Generate the design example from the Example Design tab of the 25G Ethernet Intel FPGA IP parameter editor. You can choose to generate the design with or without the IEEE 1588v2 feature. You can also choose to generate the design with or without the Reed-Solomon Forward Error Correction (RS-FEC) feature.

2.1. Features

- Supports single Ethernet channel operating at either 10G or 25G.
- Generate design example with IEEE 1588v2 feature.
- Generate design example with RS-FEC feature.
- Generates design example separately from Intel Stratix 10 Transceiver Native PHY.
- Provides testbench and simulation script.

2.2. Hardware and Software Requirements

Intel uses the following hardware and software to test the design example in a Linux system:

- Intel Quartus Prime Pro Edition software.
- ModelSim-SE, NCSim (Verilog only), VCS, and Xcelium simulator.
- Intel Stratix 10 GX Signal Integrity L-Tile (Production) Development Kit (1SX280LU2F50E1VG) for hardware testing.

2.3. Functional Description

The 10G/25G Ethernet single-channel design example consists of two core variants—MAC+PCS+PMA and MAC+PCS. The following block diagrams show the design components and the top-level signals of the two core variants in the 10G/25G Ethernet single-channel design example.
Figure 6. Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC +PCS+PMA Core Variant) Without the IEEE 1588v2 Feature

Figure 7. Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC +PCS+PMA Core Variant) with the IEEE 1588v2 Feature
Figure 8. Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC +PCS Core Variant) Without the IEEE 1588v2 Feature

Note:
1. Components outside of this block are part of the design example only.

Figure 9. Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC +PCS Core Variant) with the IEEE 1588v2 Feature

Note:
1. Components outside of this block are part of the design example only.
### 2.3.1. Design Components

#### Table 4. Design Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>25G Ethernet Intel FPGA IP</strong></td>
<td>Consists of MAC, PCS, and Transceiver PHY, with the following configuration:</td>
</tr>
<tr>
<td></td>
<td>• Core Variant: MAC+PCS+PMA, MAC+PCS</td>
</tr>
<tr>
<td></td>
<td>• Enable flow control: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable link fault generation: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable preamble passthrough: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable statistics collection: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable MAC statistics counters: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable 10G/25G dynamic rate switching: Selected</td>
</tr>
<tr>
<td></td>
<td>• Enable Native PHY Debug Master Endpoint (NPDME): Optional</td>
</tr>
<tr>
<td></td>
<td>• Reference clock frequency: 644.531250/322.265625</td>
</tr>
<tr>
<td>For the design example with the IEEE 1588 feature, the following additional parameters are configured:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enable IEEE 1588: Selected</td>
</tr>
<tr>
<td></td>
<td>• Time of day format: Enable 96-bit timestamp format</td>
</tr>
<tr>
<td>For the design example with the RS-FEC feature, the following additional parameter is configured:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Enable RS-FEC: Selected</td>
</tr>
<tr>
<td><strong>Reconfiguration Sequencer</strong></td>
<td>Reconfigures the transceiver channel speed from 10 Gbps to 25 Gbps, and vice versa.</td>
</tr>
<tr>
<td><strong>ATX PLL</strong></td>
<td>Generates TX serial clocks for the 10G and 25G transceivers.</td>
</tr>
<tr>
<td><strong>Client logic</strong></td>
<td>Consists of:</td>
</tr>
<tr>
<td></td>
<td>• Traffic generator, which generates burst packets to the 25G Ethernet Intel FPGA IP core for transmission.</td>
</tr>
<tr>
<td></td>
<td>• Traffic monitor, which receives burst packets from the 25G Ethernet Intel FPGA IP core.</td>
</tr>
<tr>
<td><strong>Source and Probe</strong></td>
<td>Source and probe signals, including system reset input signal, which you can use for debugging.</td>
</tr>
<tr>
<td><strong>Design Components for the IEEE 1588v2 Feature</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Sampling PLL</strong></td>
<td>Generates the clocks for the IEEE 1588v2 design components.</td>
</tr>
<tr>
<td></td>
<td>• latency_sclk: 156.25 MHz for latency measurement.</td>
</tr>
<tr>
<td></td>
<td>• sampling_clk: 250 MHz for ToD synchronization</td>
</tr>
<tr>
<td><strong>Time-of-day (ToD) Sync</strong></td>
<td>Synchronizes the 10G and 25G ToDs.</td>
</tr>
<tr>
<td><strong>ToD Tx</strong></td>
<td>ToD for transmit paths for the 10G and 25G transceivers.</td>
</tr>
<tr>
<td><strong>ToD Rx</strong></td>
<td>ToD for receive paths for the 10G and 25G transceivers.</td>
</tr>
<tr>
<td><strong>Master Precision Time Protocol (PTP)</strong></td>
<td>Master PTP consists of a packet generator and a packet receiver.</td>
</tr>
<tr>
<td></td>
<td>• Packet generator: Obtains timestamp information from the 25G Ethernet Intel FPGA IP core and generates Avalon®-ST packets such as Sync packet and Delay Response packet.</td>
</tr>
<tr>
<td></td>
<td>• Packet receiver: Obtains the delay request packet information from the 25G Ethernet Intel FPGA IP core and produces timestamp values.</td>
</tr>
<tr>
<td><strong>Slave PTP</strong></td>
<td>Slave PTP consists of a packet generator, a packet receiver, and packet compute.</td>
</tr>
</tbody>
</table>

---

(1) The 10G/25G Ethernet single-channel design example with IEEE 1588v2 feature only supports 96-bit timestamp format.
### Component Description

- **Packet generator**: Obtains timestamp information from the 25G Ethernet Intel FPGA IP core and generates Avalon-ST packets such as Delay Request packet.
- **Packet receiver**: Obtains the Sync and Delay Response packets information from the 25G Ethernet Intel FPGA IP core and produces timestamp values.
- **Packet compute**: Calculates and produces the delay and offsets value based on the timestamp values.

### 2.4. Simulation

The testbench sends traffic through the IP core, exercising the transmit side and receive side of the IP core.

#### Related Information

Simulating the 25G Ethernet Intel FPGA IP Design Example Testbench on page 8

### 2.4.1. Testbench

#### Figure 10.  Block Diagram of the 10G/25G Ethernet Single-Channel Design Example Simulation Testbench

#### Table 5.  Testbench Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device under test (DUT)</td>
<td>The 25G Ethernet Intel FPGA IP core.</td>
</tr>
<tr>
<td>Reconfiguration Sequencer</td>
<td>Reconfigures the transceiver channel speed from 10 Gbps to 25 Gbps, and vice versa.</td>
</tr>
</tbody>
</table>
| Ethernet Packet Generator and Packet Monitor | - Packet generator generates frames and transmit to the DUT.  
  - Packet Monitor monitors TX and RX datapaths and displays the frames in the simulator console. |
| ATX PLL                                       | Generates a TX serial clock for the Intel Stratix 10 10G/25G transceiver which is wrapped in the 25G Ethernet Intel FPGA IP core. |
Note: For the 10G/25G Ethernet single-channel design example with IEEE 1588v2 feature simulation testbench, refer to Figure 7 on page 13.

2.4.2. Simulation Design Example Components

Table 6. 10G/25G Ethernet Single-Channel Design Example Testbench File Descriptions

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic_avl_tb_top..sv</td>
<td>Top-level testbench file. The testbench instantiates the DUT, performs Avalon-MM configuration on design components and client logic, and sends and receives packet to or from.</td>
</tr>
</tbody>
</table>

Testbench Scripts

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>run_vsim.do</td>
<td>The ModelSim script to run the testbench.</td>
</tr>
<tr>
<td>run_vcs.sh</td>
<td>The Synopsys VCS script to run the testbench.</td>
</tr>
<tr>
<td>run_ncsim.sh</td>
<td>The Cadence NCSim script to run the testbench.</td>
</tr>
<tr>
<td>run_xcelium.sh</td>
<td>The Xcelium script to run the testbench.</td>
</tr>
</tbody>
</table>

2.4.3. Test Case—Design Example Without the IEEE 1588v2 Feature

The simulation test case performs the following actions:
1. Instantiates 25G Ethernet Intel FPGA IP and ATX PLL.
2. Starts up the design example with an operating speed of 25G.
3. Waits for RX clock and PHY status signal to settle.
4. Prints PHY status.
5. Sends and receives 10 valid data on 25G speed.
6. Performs channel reset and switches to 10G speed.
7. Waits for RX clock and PHY status signal to settle.
8. Prints PHY status.
9. Sends and receives another 10 valid data on 10G speed.
10. Performs channel reset and switches to 25G speed.
11. Waits for RX clock and PHY status signal to settle.
12. Prints PHY status.
13. Sends and receives another 10 valid data on 25G speed.
14. Analyzes the results. The successful testbench displays "Simulation PASSED."

The following sample output illustrates a successful simulation test run:
Waiting for RX alignment
RX deskew locked
RX lane alignment locked
TX enabled
** Sending Packet 1...
** Sending Packet 2...
** Sending Packet 3...
** Sending Packet 4...
** Sending Packet 5...
<table>
<thead>
<tr>
<th>Sending Packet</th>
<th>Received Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>6...</td>
<td>1...</td>
</tr>
<tr>
<td>7...</td>
<td>2...</td>
</tr>
<tr>
<td>8...</td>
<td>3...</td>
</tr>
<tr>
<td>9...</td>
<td>4...</td>
</tr>
<tr>
<td>10...</td>
<td>5...</td>
</tr>
<tr>
<td>1...</td>
<td>6...</td>
</tr>
<tr>
<td>2...</td>
<td>7...</td>
</tr>
<tr>
<td>3...</td>
<td>8...</td>
</tr>
<tr>
<td>4...</td>
<td>9...</td>
</tr>
<tr>
<td>5...</td>
<td>10...</td>
</tr>
</tbody>
</table>

Switching to 10G mode: 10G Reconfig start
Switching to 10G mode: 10G Reconfig End
Waiting for RX alignment
RX deskew locked
RX lane alignment locked
TX enabled

<table>
<thead>
<tr>
<th>Sending Packet</th>
<th>Received Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1...</td>
<td>1...</td>
</tr>
<tr>
<td>2...</td>
<td>2...</td>
</tr>
<tr>
<td>3...</td>
<td>3...</td>
</tr>
<tr>
<td>4...</td>
<td>4...</td>
</tr>
<tr>
<td>5...</td>
<td>5...</td>
</tr>
<tr>
<td>6...</td>
<td>6...</td>
</tr>
<tr>
<td>7...</td>
<td>7...</td>
</tr>
<tr>
<td>8...</td>
<td>8...</td>
</tr>
<tr>
<td>9...</td>
<td>9...</td>
</tr>
<tr>
<td>10...</td>
<td>10...</td>
</tr>
</tbody>
</table>

Switching to 25G mode: 25G Reconfig start
Switching to 25G mode: 25G Reconfig End
Waiting for RX alignment
RX deskew locked
RX lane alignment locked
TX enabled

<table>
<thead>
<tr>
<th>Sending Packet</th>
<th>Received Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>1...</td>
<td>1...</td>
</tr>
<tr>
<td>2...</td>
<td>2...</td>
</tr>
<tr>
<td>3...</td>
<td>3...</td>
</tr>
<tr>
<td>4...</td>
<td>4...</td>
</tr>
<tr>
<td>5...</td>
<td>5...</td>
</tr>
<tr>
<td>6...</td>
<td>6...</td>
</tr>
<tr>
<td>7...</td>
<td>7...</td>
</tr>
<tr>
<td>8...</td>
<td>8...</td>
</tr>
<tr>
<td>9...</td>
<td>9...</td>
</tr>
<tr>
<td>10...</td>
<td>10...</td>
</tr>
</tbody>
</table>

** Testbench complete.**
2.4.4. Test Case—Design Example with the IEEE 1588v2 Feature

Note: For 10G/25G Ethernet single-channel design example with IEEE 1588v2 feature simulation testbench, refer to Figure 7 on page 13.

The simulation test case performs the following actions:

1. Instantiates 25G Ethernet Intel FPGA IP, ATX PLL, and IO PLL (sampling PLL).
2. Starts up the design example with an operating speed of 25G.
3. Waits for RX clock and PHY status signal to settle.
4. Prints PHY status.
5. Checks for 10 valid data on 25G speed.
7. Waits for RX clock and PHY status signal to settle.
8. Prints PHY status.
9. Checks for another 10 valid data on 10G speed.
10. Switches to 25G speed once all 10 valid data passes.
11. Waits for RX clock and PHY status signal to settle.
12. Prints PHY status.
13. Checks for another 10 valid data on 25G speed.
14. Analyzes the results. The successful testbench displays "Simulation PASSED." when the PTP delay and offset data is within the threshold value.

The following sample output illustrates a successful simulation test run:

```
# Running at 25G mode...

# Waiting for RX alignment...
iatpg_pipeline_global_en is set
iatpg_pipeline_global_en is set
RX deskew locked.
RX lane alignment locked.
Sending packets...
Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000064457
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
  Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000064bb4
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000ffffff8a2
  Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000643b5
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000520
  Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000634fb
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
  Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063f3b
```
# Offset(sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063a1a
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000006445a
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063e95
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000648d5
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000643b5
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000520
Offset within tolerance range.

Finished sending packets.

Switching to 10G mode: 10G Reconfig starts...
Switching to 10G mode: 10G Reconfig End.

Waiting for RX alignment...
RX deskew locked.
RX lane alignment locked.

Configuring 1588 period...
Configuring 1588 period done.

Sending packets...

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e5a7d
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000002013
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e0764
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000002013
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e0764
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000002013
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000dfa97
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000006660
Offset within tolerance range.

# Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e1431
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000005993
Offset within tolerance range.
Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e2db7
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000fffffffccc0
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e1431
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000fffffff8006
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e60e4
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000fffffff320
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e60e4
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000fffffff8006
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000e60e4
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000fffffff320
Offset within tolerance range.

Finished sending packets.

Switching to 25G mode: 25G Reconfig start...
Switching to 25G mode: 25G Reconfig end.

Waiting for RX alignment...
RX deskew locked.
RX lane alignment locked.

Configuring 1588 period...
Configuring 1588 period done.

Sending packets...

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063c58
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063c58
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000006502f
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000006502f
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000006554d
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000664b10
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000000000000000000
Offset within tolerance range.
<table>
<thead>
<tr>
<th>Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000064b10</th>
<th>Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000</th>
<th>Offset within tolerance range.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000064bb4</td>
<td>Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000</td>
<td>Offset within tolerance range.</td>
</tr>
<tr>
<td>Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000065a6c</td>
<td>Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000</td>
<td>Offset within tolerance range.</td>
</tr>
</tbody>
</table>

** Finished sending packets. 
** Testbench complete.

### 2.5. Compilation

Follow the procedure in Compiling and Configuring the Design Example in Hardware on page 8 to compile and configure the design example in the selected hardware.

You can estimate resource utilization and Fmax using the compilation-only design example. You can compile your design using the Start Compilation command on the Processing menu in the Intel Quartus Prime Pro Edition software. A successful compilation generates the compilation report summary.


**Related Information**

Design Compilation  

### 2.6. Hardware Testing

In the hardware design example, you can program the IP core in internal serial loopback mode and generate traffic on the transmit side that loops back through the receive side.

Follow the procedure at the provided related information link to test the design example in the selected hardware.

**Related Information**

Testing the 25G Ethernet Intel FPGA IP Design in Hardware on page 10  
More information on the procedure and hardware setup.
2.6.1. Test Procedure—Design Example Without the IEEE 1588v2 Feature

Follow these steps to test the design examples in hardware using PMA serial loopback:

Note: The design example starts with default data rate of 25G.

1. Perform data rate switching to 10G:
   a. In Intel Quartus Prime Pro Edition software, go to Tools ➤ In-System Sources & Probes Editor tool to open the default source and probe GUI.

2. Perform data rate switching to 25G:
   a. In Intel Quartus Prime Pro Edition software, go to Tools ➤ In-System Sources & Probes Editor tool to open the default source and probe GUI.
   b. Set the source bit[1] in source and probe to 0.

3. Perform system reset release after executing the data rate reconfiguration:
   a. Click Tools ➤ In-System Sources & Probes Editor tool for the default Source and Probe GUI.
   b. Toggle the system reset signal (Source[0]) from 0 to 1 to apply the reset and return the system reset signal back to 0 to release the system from the reset state.
   c. Monitor the Probe signals and ensure that the status is valid.

4. Run the following commands in the system console to start the serial loopback test:

Table 7. Command Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>chkphy_status &lt;link num&gt;</td>
<td>Displays the clock frequencies and PHY lock status.</td>
<td>% chkphy_status 0 # Check status of link 0</td>
</tr>
<tr>
<td>chkmac_stats &lt;link num&gt;</td>
<td>Displays the values in the MAC statistics counters.</td>
<td>% chkmac_stats 1 # Checks mac statistics counter of link 1</td>
</tr>
<tr>
<td>clear_all_stats &lt;link num&gt;</td>
<td>Clears the IP core statistics counters.</td>
<td>% clear_all_stats 1 # Clears statistics counter of link 1</td>
</tr>
<tr>
<td>start_gen &lt;link num&gt;</td>
<td>Starts the packet generator.</td>
<td>% start_gen 1 # Begin packet generation on link 1</td>
</tr>
<tr>
<td>stop_gen &lt;link num&gt;</td>
<td>Stops the packet generator.</td>
<td>% stop_gen 1 # Stop packet generation on link 1</td>
</tr>
<tr>
<td>loop_on &lt;link num&gt;</td>
<td>Turns on internal serial loopback.</td>
<td>% loop_on 2 # Turn on internal loopback on link 2</td>
</tr>
<tr>
<td>loop_off &lt;link num&gt;</td>
<td>Turns off internal serial loopback.</td>
<td>% loop_off 2 # Turn off internal loopback on link 2</td>
</tr>
<tr>
<td>reg_read &lt;addr&gt;</td>
<td>Returns the IP core register value at &lt;addr&gt;.</td>
<td>% reg_read 0x302 # Read IP CSR register at address 302 of link 0</td>
</tr>
</tbody>
</table>

continued...
### Parameter Description Example Usage

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>% reg_read 0x4542 # Read transceiver reconfiguration register at address 4542 of link 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| reg_write <addr> <data> | Writes <data> to the IP core register at address <addr>. | % reg_write 0x30301 0x1 # Write 0x1 to IP CSR scratch register at address 301 of link 3  
|                  |                                                  | % reg_write 0x34542 0x0 # Write 0x0 to transceiver reconfiguration register at address 4542 of link 3 |

**Note:**

a. For single-channel design, `<link num>` is always 0.

b. For multi-channel design, `<link num>` is the channel number. The valid channel number range is 0 to 3.

a. Type `loop_on <link num>` to turn on the internal serial loopback mode.

b. Type `chkphy_status <link num>` to check the status of the PHY. The TXCLK, RXCLK, and RX status should have the same values shown below for a stable link:

**Figure 11. System Console Example Printout**

```
Link: 0  Addr: 300  
RX PHY Register Access: Checking Clock Frequencies (KHz)
  REFCLK :0 (KHz)  
  TXCLK :39066 (KHz)  
  RXCLK :39066 (KHz)  
Link: 0  Addr: 300  
RX PHY Status Polling  
  Rx Frequency Lock Status: 0x00000001  
  Mac Clock in OK Condition: 0x00000007  
  Rx Frame Error: 0x00000000  
  Rx PHY Fully Aligned?: 0x00000001  
```

c. Type `clear_all_stats <link num>` to clear TX and RX statistics registers.

d. Type `start_gen <link num>` to begin packet generation.

e. Type `stop_gen <link num>` to stop packet generation.

f. Type `chkmac_stats <link num>` to read the TX and RX statistics counters.

Make sure that:

i. The transmitted packet frames match the received packet frames.

ii. No error frames are received.

g. Type `loop_off <link num>` to turn off the internal serial loopback.
Note: The above configuration is applied to the default 25G mode for the first time.

**Figure 12. Sample Test Output—TX and RX Statistics Counters**

<table>
<thead>
<tr>
<th>Statistics for Base 25G (Rx)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragmented Frames: 0</td>
</tr>
<tr>
<td>Jabbered Frames: 0</td>
</tr>
<tr>
<td>Any Size with PCS Err Frame: 0</td>
</tr>
<tr>
<td>Right Size with PCS Err Pts: 0</td>
</tr>
<tr>
<td>Multicast Data Err Frames: 0</td>
</tr>
<tr>
<td>Broadcast Data Err Frames: 0</td>
</tr>
<tr>
<td>Unicast Data Err Frames: 0</td>
</tr>
<tr>
<td>Multicast Control Err Frame: 0</td>
</tr>
<tr>
<td>Broadcast Control Err Frame: 0</td>
</tr>
<tr>
<td>Unicast Control Err Frames: 0</td>
</tr>
<tr>
<td>Pause Control Err Frames: 0</td>
</tr>
<tr>
<td>64 Byte Frames: 374474</td>
</tr>
<tr>
<td>65 - 127 Byte Frames: 187734</td>
</tr>
<tr>
<td>128 - 255 Byte Frames: 0</td>
</tr>
<tr>
<td>256 - 511 Byte Frames: 0</td>
</tr>
<tr>
<td>512 - 1023 Byte Frames: 0</td>
</tr>
<tr>
<td>1024 - 1518 Byte Frames: 0</td>
</tr>
<tr>
<td>1519 - MAX Byte Frames: 0</td>
</tr>
<tr>
<td>Rx Frame Starts: 0</td>
</tr>
<tr>
<td>Multicast data OK Frame: 0</td>
</tr>
<tr>
<td>Broadcast data OK Frame: 0</td>
</tr>
<tr>
<td>Unicast data OK Frames: 563210</td>
</tr>
<tr>
<td>Multicast Control Frames: 0</td>
</tr>
<tr>
<td>Broadcast Control Frames: 0</td>
</tr>
<tr>
<td>Unicast Control Frames: 0</td>
</tr>
<tr>
<td>Pause Control Frames: 0</td>
</tr>
<tr>
<td>Payload Octets OK: 27499548</td>
</tr>
<tr>
<td>Frame Octets OK: 35747320</td>
</tr>
<tr>
<td>LMK: 0 Addr: 0x00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistics for Base 20G (Tx)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fragmented Frames: 0</td>
</tr>
<tr>
<td>Jabbered Frames: 0</td>
</tr>
<tr>
<td>Any Size with PCS Err Frame: 0</td>
</tr>
<tr>
<td>Right Size with PCS Err Pts: 0</td>
</tr>
<tr>
<td>Multicast Data Err Frames: 0</td>
</tr>
<tr>
<td>Broadcast Data Err Frames: 0</td>
</tr>
<tr>
<td>Unicast Data Err Frames: 0</td>
</tr>
<tr>
<td>Multicast Control Err Frame: 0</td>
</tr>
<tr>
<td>Broadcast Control Err Frame: 0</td>
</tr>
<tr>
<td>Unicast Control Err Frames: 0</td>
</tr>
<tr>
<td>Pause control Err Frames: 0</td>
</tr>
<tr>
<td>64 Byte Frames: 374474</td>
</tr>
<tr>
<td>65 - 127 Byte Frames: 187734</td>
</tr>
<tr>
<td>128 - 255 Byte Frames: 0</td>
</tr>
<tr>
<td>256 - 511 Byte Frames: 0</td>
</tr>
<tr>
<td>512 - 1023 Byte Frames: 0</td>
</tr>
<tr>
<td>1024 - 1518 Byte Frames: 0</td>
</tr>
<tr>
<td>1519 - MAX Byte Frames: 0</td>
</tr>
<tr>
<td>Tx Frame Starts: 0</td>
</tr>
<tr>
<td>Multicast data OK Frame: 0</td>
</tr>
<tr>
<td>Broadcast data OK Frame: 0</td>
</tr>
<tr>
<td>Unicast data OK Frames: 563210</td>
</tr>
<tr>
<td>Multicast Control Frames: 0</td>
</tr>
<tr>
<td>Broadcast Control Frames: 0</td>
</tr>
<tr>
<td>Unicast Control Frames: 0</td>
</tr>
<tr>
<td>Pause Control Frames: 0</td>
</tr>
<tr>
<td>Payload Octets OK: 27499548</td>
</tr>
<tr>
<td>Frame Octets OK: 35747320</td>
</tr>
</tbody>
</table>

**Related Information**

Design Example Registers on page 47
2.6.2. Test Procedure—Design Example with the IEEE 1588v2 Feature

Follow these steps to test the design examples in hardware using PMA serial loopback:

*Note:* The design example starts with default data rate of 25G.

1. Perform data rate switching to 10G:
   a. In Intel Quartus Prime Pro Edition software, go to **Tools ➤ In-System Sources & Probes Editor** tool to open the default source and probe GUI.
   c. In the System Console panel, type the following commands as below to set the correct clock period for the required TX and RX MAC clock frequency in 10G speed mode:

   ```
   reg_write 0xA05 0x66666
   reg_write 0xB05 0x66666
   ```

2. Perform data rate switching to 25G:
   a. In Intel Quartus Prime Pro Edition software, go to **Tools ➤ In-System Sources & Probes Editor** tool to open the default source and probe GUI.
   b. Set the source bit[1] in source and probe to 0.
   c. In the System Console panel, type the following commands as below to set the correct clock period for the required TX and RX MAC clock frequency in 25G speed mode:

   ```
   reg_write 0xA05 0x28F5C
   reg_write 0xB05 0x28F5C
   ```

   *Note:* 0xA05 is register that configure TX_PTP_CLK_PERIOD. 0xB05 is register that configure RX_PTP_CLK_PERIOD.

3. Perform system reset release after executing the data rate reconfiguration:
   a. Click **Tools ➤ In-System Sources & Probes Editor** tool for the default Source and Probe GUI.
   b. Toggle the system reset signal (Source[0]) from 0 to 1 to apply the reset and return the system reset signal back to 0 to release the system from the reset state.
   c. Monitor the Probe signals and ensure that the status is valid.

4. To perform internal serial loopback test, refer to the **Test Procedure—Design Example Without the IEEE 1588v2 Feature** section of this chapter.

**Related Information**
- [Design Example Registers](#) on page 47
- [Test Procedure—Design Example Without the IEEE 1588v2 Feature](#) on page 23
3. 25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices

The 25G Ethernet single-channel design example demonstrates an Ethernet solution for Intel Stratix 10 devices using the 25G Ethernet Intel FPGA IP core.

Generate the design example from the Example Design tab of the 25G Ethernet Intel FPGA IP parameter editor. You can choose to generate the design with or without the IEEE 1588v2 feature. You can also choose to generate the design with or without the Reed-Solomon Forward Error Correction (RS-FEC) feature.

3.1. Features

- Supports single Ethernet channel operating at 25G.
- Generates design example with IEEE 1588v2 feature.
- Generates design example with RS-FEC feature.
- Generates design example separately from Intel Stratix 10 Transceiver Native PHY.
- Provides testbench and simulation script.

3.2. Hardware and Software Requirements

Intel uses the following hardware and software to test the design example in a Linux system:

- Intel Quartus Prime Pro Edition software.
- ModelSim-SE, NCSim (Verilog only), VCS, and Xcelium simulator.
- Intel Stratix 10 GX Signal Integrity L-Tile (Production) Development Kit (1SX280LU2F50E1VG) for hardware testing.

3.3. Functional Description

The 25G Ethernet single-channel design example consists of two core variants—MAC +PCS+PMA and MAC+PCS. The following block diagrams show the design components and the top-level signals of the two core variants in the 25G Ethernet single-channel design example.
Figure 13. Block Diagram—25G Ethernet Single-Channel Design Example (MAC+PCS +PMA Core Variant) Without the IEEE 1588v2 Feature

Figure 14. Block Diagram—25G Ethernet Single-Channel Design Example (MAC+PCS +PMA Core Variant) with the IEEE 1588v2 Feature
Figure 15. Block Diagram—25G Ethernet Single-Channel Design Example (MAC+PCS Core Variant) Without the IEEE 1588v2 Feature

25G Ethernet Single-Channel Design Example

Client Logic (Packet Generator and Monitor)

Control and Status Interface

Control Logic

Avalon-MM Combine

Avalon-MM Read Data

Transceiver PHY

MAC+PCS

RS-FEC (optional)

JTAG to Avalon-MM Bridge

Intel Stratix 10 Dynamic Reconfiguration Interface

Avalon-MM Read Data

System Console

ATX PLL (25G)

Sampling PLL

Latency_sclk

Sampling_clk (1)

Control and Status Interface

Control Logic

Avalon-MM

Avalon-ST TX [63:0]

Avalon-MM Read Data

Avalon-MM Client Logic

Packet Generator and Monitor

JTAG

ATX PLL (25G)

Source and Probe

Note:
1. Components outside of this block are part of the design example only.

Figure 16. Block Diagram—25G Ethernet Single-Channel Design Example (MAC+PCS Core Variant) with the IEEE 1588v2 Feature

Intel Stratix 10 Signal Integrity L-Tile (Production) Development Kit

25G Ethernet Single-Channel Design Example

Client Logic (Packet Generator and Monitor)

Control and Status Interface

Control Logic

Avalon-MM Combine

Avalon-MM Read Data

Transceiver PHY

MAC+PCS

RS-FEC (optional)

JTAG to Avalon-MM Bridge

Intel Stratix 10 Dynamic Reconfiguration Interface

Avalon-MM Read Data

System Console

ATX PLL (25G)

Sampling PLL

Latency_sclk

Sampling_clk (1)

Control and Status Interface

Control Logic

Avalon-MM

Avalon-ST RX [63:0]

Avalon-MM Read Data

Avalon-MM Client Logic

Packet Generator and Monitor

JTAG

ATX PLL (25G)

Source and Probe

Note:
1. Components outside of this block are part of the design example only.
### 3.3.1. Design Components

#### Table 8. Design Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
</table>
| 25G Ethernet Intel FPGA IP | Consists of MAC, PCS, and Transceiver PHY, with the following configuration:  
  - **Core Variant**: MAC+PCS+PMA, MAC+PCS  
  - **Enable flow control**: Optional  
  - **Enable link fault generation**: Optional  
  - **Enable preamble passthrough**: Optional  
  - **Enable statistics collection**: Optional  
  - **Enable MAC statistics counters**: Optional  
  - **Enable 10G/25G dynamic rate switching**: Not selected  
  - **Enable Native PHY Debug Master Endpoint (NPDME)**: Optional  
  - **Reference clock frequency**: 644.531250/322.265625  
  For the design example with the IEEE 1588 feature, the following additional parameters are configured:  
  - **Enable IEEE 1588**: Selected  
  - **Time of day format**: Enable 96-bit timestamp format \(^{(2)}\)  
  For the design example with the RS-FEC feature, the following additional parameter is configured:  
  - **Enable RS-FEC**: Selected |

| ATX PLL | Generates TX serial clocks for the 25G transceiver. |

| Client logic | Consists of:  
  - Traffic generator, which generates burst packets to the 25G Ethernet Intel FPGA IP core for transmission.  
  - Traffic monitor, which monitors burst packets that are coming from the 25G Ethernet Intel FPGA IP core. |

| Source and Probe | Source and probe signals, including system reset input signal, which you can use for debugging. |

#### Design Components for the IEEE 1588v2 Feature

| Sampling PLL | Generates the clocks for the IEEE 1588v2 design components.  
  - **latency_sclk**: 156.25 MHz for latency measurement.  
  - **sampling_clk**: 250 MHz for ToD synchronization |

| Time-of-day (ToD) Sync | Synchronizes the 25G ToD. |

| ToD Tx | ToD for transmit paths for the 25G transceiver. |

| ToD Rx | ToD for receive paths for the 25G transceiver. |

| Master Precision Time Protocol (PTP) | Master PTP consists of a packet generator and a packet receiver.  
  - Packet generator: Obtains timestamp information from the 25G Ethernet Intel FPGA IP core and generates Avalon-ST packets such as Sync packet and Delay Response packet.  
  - Packet receiver: Obtains the delay request packet information from the 25G Ethernet Intel FPGA IP core and produces timestamp values. |

| Slave PTP | Slave PTP consists of a packet generator, a packet receiver, and a packet compute.  
  - Packet generator: Obtains timestamp information from the 25G Ethernet Intel FPGA IP core and generates Avalon-ST packets such as Delay Request packet.  
  - Packet receiver: Obtains the Sync and Delay Response packets information from the 25G Ethernet Intel FPGA IP core and produces timestamp values.  
  - Packet compute: Calculates and produces the delay and offsets value based on the timestamp values. |

---

\(^{(2)}\) The 25G Ethernet single-channel design example with IEEE 1588v2 feature only supports 96-bit timestamp format.
3.4. Simulation

The testbench sends traffic through the IP core, exercising the transmit side and receive side of the IP core.

Related Information
Simulating the 25G Ethernet Intel FPGA IP Design Example Testbench on page 8

3.4.1. Testbench

Figure 17. Block Diagram of the 25G Ethernet Single-Channel Design Example Simulation Testbench

Table 9. Testbench Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device under test (DUT)</td>
<td>The 25G Ethernet Intel FPGA IP core.</td>
</tr>
<tr>
<td>Reconfiguration Sequencer</td>
<td>Reconfigures the transceiver channel speed from 10 Gbps to 25 Gbps, and vice versa.</td>
</tr>
</tbody>
</table>
| Ethernet Packet Generator and Packet Monitor | • Packet generator generates frames and transmit to the DUT.  
                                          | • Packet Monitor monitors TX and RX datapaths and displays the frames in the simulator console. |
| ATX PLL                                  | Generates a TX serial clock for the Intel Stratix 10 10G/25G transceiver which is wrapped in the 25G Ethernet Intel FPGA IP core. |

Note: For the 25G Ethernet single-channel design example with IEEE 1588v2 feature simulation testbench, refer to Figure 14 on page 28.
3.4.2. Simulation Design Example Components

Table 10. 25G Ethernet Single-Channel Design Example Testbench File Descriptions

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Testbench and Simulation Files</strong></td>
<td></td>
</tr>
<tr>
<td>basic_avl_tb_top.v</td>
<td>Top-level testbench file. The testbench instantiates the DUT, performs Avalon-MM configuration on design components and client logic, and sends and receives packet to or from.</td>
</tr>
<tr>
<td><strong>Testbench Scripts</strong></td>
<td></td>
</tr>
<tr>
<td>run_vsim.do</td>
<td>The ModelSim script to run the testbench.</td>
</tr>
<tr>
<td>run_vcs.sh</td>
<td>The Synopsys VCS script to run the testbench.</td>
</tr>
<tr>
<td>run_ncsim.sh</td>
<td>The Cadence NCSim script to run the testbench.</td>
</tr>
<tr>
<td>run_xcelium.sh</td>
<td>The Xcelium script to run the testbench.</td>
</tr>
</tbody>
</table>

3.4.3. Test Case—Design Example Without the IEEE 1588v2 Feature

The simulation test case performs the following actions:

1. Instantiates and ATX PLL.
2. Waits for RX clock and PHY status signal to settle.
3. Prints PHY status.
4. Analyzes the results. The successful testbench sends ten packets, receives ten packets, and displays "Testbench complete."
3.4.4. Test Case—Design Example with the IEEE 1588v2 Feature

Note: For 25G Ethernet single-channel design example with IEEE 1588v2 feature simulation testbench, refer to Figure 14 on page 28.

The simulation test case performs the following actions:
1. Instantiates ATX PLL, and IO PLL (sampling PLL).
2. Waits for RX clock and PHY status signal to settle.
3. Prints PHY status.
4. Checks for 10 valid data.
5. Analyzes the results. The successful testbench displays "Testbench complete." when the PTP delay and offset data are within the threshold values.

```
#rx_pcs_ready[ 0]
RX deskew locked
RX lane alignment locked
#TX enabled
*** Link 0 Sending Packet 1...
*** Link 0 Sending Packet 2...
*** Link 0 Sending Packet 3...
*** Link 0 Sending Packet 4...
*** Link 0 Sending Packet 5...
*** Link 0 Sending Packet 6...
*** Link 0 Sending Packet 7...
*** Link 0 Sending Packet 8...
*** Link 0 Received Packet 1...
*** Link 0 Received Packet 2...
*** Link 0 Received Packet 3...
*** Link 0 Received Packet 4...
*** Link 0 Received Packet 5...
*** Link 0 Received Packet 6...
*** Link 0 Received Packet 7...
*** Link 0 Received Packet 8...
*** Link 0 Received Packet 9...
*** Link 0 Received Packet 10...
***
*** Testbench complete.
***
```
The following sample output illustrates a successful simulation test run:

```
Waiting for RX alignment...
iatpg_pipeline_global_en is set
RX deskew locked.
RX lane alignment locked.

Sending packets...

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000064457
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000064bb4
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000ffffff8a2
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000643b5
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000520
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000634fb
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063f3b
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063a1a
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x00000000000000000006445a
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000063e95
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

Delay (sec[95:48],ns[47:16],fns[15:0]): 0x0000000000000000000648d5
Offset (sec[95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
Offset within tolerance range.

Finished sending packets.

**
** Testbench complete.
**```
Figure 19. Sample Simulation Output for Design Example with the IEEE 1588v2 Feature (Part 1 of 2)

† Waiting for RX alignment...
† istpy_pipeline_global_en is set
† istpy_pipeline_global_en is set
† RX desckew locked.
† RX lane alignment locked.
†
† Sending packets...
† Delay [sec95:48],ns[47:16],fns[15:0]): 0x0000000000000000033b7
† Offset [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
† Offset within tolerance range.
†
† Delay [sec95:48],ns[47:16],fns[15:0]): 0x0000000000000000032c2
† Offset [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
† Offset within tolerance range.
†
† Delay [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000003cd7
† Offset [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
† Offset within tolerance range.
†
† Delay [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000003d07
† Offset [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
† Offset within tolerance range.
†
† Delay [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000003d63
† Offset [sec95:48],ns[47:16],fns[15:0]): 0x000000000000000000000000
† Offset within tolerance range.
3.5. Compilation

Follow the procedure in Compiling and Configuring the Design Example in Hardware on page 8 to compile and configure the design example in the selected hardware.

You can estimate resource utilization and Fmax using the compilation-only design example. You can compile your design using the Start Compilation command on the Processing menu in the Intel Quartus Prime Pro Edition software. A successful compilation generates the compilation report summary.

3.6. Hardware Testing

In the hardware design example, you can program the IP core in internal serial loopback mode and generate traffic on the transmit side that loops back through the receive side.

Follow the procedure at the provided related information link to test the design example in the selected hardware.

3.6.1. Test Procedure—Design Example With and Without the IEEE 1588v2 Feature

Follow these steps to test the design example in hardware:

1. Before you run the hardware testing for this design example, you must reset the system:
   a. Click Tools ➤ In-System Sources & Probes Editor tool for the default Source and Probe GUI.
   b. Toggle the system reset signal (Source[0]) from 0 to 1 to apply the reset and return the system reset signal back to 0 to release the system from the reset state.
   c. Monitor the Probe signals and ensure that the status is valid.

2. To perform internal serial loopback test, refer to the Test Procedure—Design Example Without the IEEE 1588v2 Feature section of the 10G/25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices chapter.

Related Information
- Design Example Registers on page 47
- Test Procedure—Design Example Without the IEEE 1588v2 Feature on page 23
4. 25G Ethernet Multi-Channel Design Example for Intel Stratix 10 Devices

The 25G Ethernet multi-channel design example demonstrates an Ethernet solution for Intel Stratix 10 devices using the 25G Ethernet Intel FPGA IP core.

Generate the design example from the Example Design tab of the 25G Ethernet Intel FPGA IP parameter editor.

4.1. Features

- Supports up to four Ethernet channels operating at 25G.
- Provides testbench and simulation script.

4.2. Hardware and Software Requirements

Intel uses the following hardware and software to test the design example in a Linux system:

- Intel Quartus Prime Pro Edition software.
- ModelSim-SE, NCSim (Verilog only), VCS, and Xcelium simulator.
- Intel Stratix 10 GX Signal Integrity L-Tile (Production) Development Kit (1SX280LU2F50E1VG) for hardware testing.

4.3. Functional Description

The 25G Ethernet multi-channel design example consists of various components. The following block diagram shows the design components and the top-level signals of the design example.
4.3.1. Design Components

Table 11. Design Components

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>25G Ethernet Intel FPGA IP</td>
<td>Consists of MAC, PCS, and Transceiver PHY, with the following configuration:</td>
</tr>
<tr>
<td></td>
<td>• Core Variant: MAC+PCS+PMA</td>
</tr>
<tr>
<td></td>
<td>• Enable RS-FEC: Not selected</td>
</tr>
<tr>
<td></td>
<td>• Enable flow control: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable link fault generation: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable preamble passthrough: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable statistics collection: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable MAC statistics counters: Optional</td>
</tr>
<tr>
<td></td>
<td>• Enable IEEE 1588: Not selected</td>
</tr>
<tr>
<td></td>
<td>• Enable 10G/25G dynamic rate switching: Not selected</td>
</tr>
<tr>
<td></td>
<td>• Enable Native PHY Debug Master Endpoint (NPDE): Optional</td>
</tr>
<tr>
<td></td>
<td>• Reference clock frequency: 644.531250/322.265625</td>
</tr>
<tr>
<td>ATX PLL</td>
<td>Generates TX serial clocks for the 25G transceiver.</td>
</tr>
<tr>
<td>Client logic</td>
<td>Consists of:</td>
</tr>
<tr>
<td></td>
<td>• Traffic generator, which generates burst packets to the 25G Ethernet Intel FPGA IP core for transmission.</td>
</tr>
<tr>
<td></td>
<td>• Traffic monitor, which receives burst packets from the 25G Ethernet Intel FPGA IP core.</td>
</tr>
<tr>
<td>Source and Probe</td>
<td>Source and probe signals, including system reset input signal, which you can use for debugging.</td>
</tr>
</tbody>
</table>
4.4. Simulation

The testbench sends traffic through the IP core, exercising the transmit side and receive side of the IP core.

Related Information
Simulating the 25G Ethernet Intel FPGA IP Design Example Testbench on page 8

4.4.1. Testbench

Figure 22. Block Diagram of the 25G Ethernet Multi-Channel Design Example Simulation Testbench

<table>
<thead>
<tr>
<th>Table 12. Testbench Components</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Component</strong></td>
</tr>
<tr>
<td>Device under test (DUT)</td>
</tr>
<tr>
<td>Reconfiguration Sequencer</td>
</tr>
</tbody>
</table>
| Ethernet Packet Generator and Packet Monitor | • Packet generator generates frames and transmit to the DUT.  
• Packet Monitor monitors TX and RX datapaths and displays the frames in the simulator console. |
| ATX PLL | Generates a TX serial clock for the Intel Stratix 10 10G/25G transceiver which is wrapped in the 25G Ethernet Intel FPGA IP core. |
4.4.2. Simulation Design Example Components

Table 13. 25G Ethernet Multi-Channel Design Example Testbench File Descriptions

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>basic_avl_tb_top.v</td>
<td>Top-level testbench file. The testbench instantiates the DUT, performs Avalon-MM configuration on design components and client logic, and sends and receives packet to or from the 25G Ethernet Intel FPGA IP.</td>
</tr>
<tr>
<td>run_vsim.do</td>
<td>The ModelSim script to run the testbench.</td>
</tr>
<tr>
<td>run_vcs.sh</td>
<td>The Synopsys VCS script to run the testbench.</td>
</tr>
<tr>
<td>run_ncsim.sh</td>
<td>The Cadence NCSim script to run the testbench.</td>
</tr>
<tr>
<td>run_xcelium.sh</td>
<td>The Xcelium script to run the testbench.</td>
</tr>
</tbody>
</table>

4.4.3. Test Case

The simulation test case performs the following steps:
1. Instantiates 25G Ethernet Intel FPGA IP and ATX PLL.
2. Waits for PHY status signal to settle.
3. Prints PHY status.
4. Analyzes the results. The successful testbench sends and receives packets, and displays "Testbench complete."
Figure 23. Sample Simulation Output when Ethernet Channel is Configured to 1

This figure shows a successful simulation test run when the Ethernet channel (i.e., LINK) is configured to 1.

```
#rx_pcs_ready[ 0]
#RX deskew locked
#RX lane alignment locked
#TX enabled

*** Link 0 Sending Packet 1...
*** Link 0 Sending Packet 2...
*** Link 0 Sending Packet 3...
*** Link 0 Sending Packet 4...
*** Link 0 Sending Packet 5...
*** Link 0 Sending Packet 6...
*** Link 0 Sending Packet 7...
*** Link 0 Sending Packet 8...
*** Link 0 Received Packet 1...
*** Link 0 Received Packet 2...
*** Link 0 Received Packet 3...
*** Link 0 Sending Packet 10...
*** Link 0 Received Packet 4...
*** Link 0 Received Packet 5...
*** Link 0 Received Packet 6...
*** Link 0 Received Packet 7...
*** Link 0 Received Packet 8...
*** Link 0 Received Packet 9...
*** Link 0 Received Packet 10...

***
*** Testbench complete.
***
```
This figure shows a successful simulation test run when the Ethernet channel (i.e., LINK) is configured to 4.

```plaintext
rx_pcs_ready[ 6]
rx_pcs_ready[ 1]
rx_pcs_ready[ 2]
rx_pcs_ready[ 3]
RX descru locked
RX lane alignment locked
TX enabled
** Link 0 Sending Packet 1...
** Link 0 Sending Packet 2...
** Link 0 Sending Packet 3...
** Link 0 Sending Packet 4...
** Link 0 Sending Packet 5...
** Link 0 Sending Packet 6...
** Link 0 Sending Packet 7...
** Link 0 Sending Packet 8...
** Link 0 Received Packet 1...
** Link 0 Received Packet 2...
** Link 0 Received Packet 3...
** Link 0 Received Packet 4...
** Link 0 Received Packet 5...
** Link 0 Received Packet 6...
** Link 0 Received Packet 7...
** Link 0 Received Packet 8...
** Link 0 Received Packet 9...
** Link 0 Received Packet 10...
** Link 1 Sending Packet 1...
** Link 1 Sending Packet 2...
** Link 1 Sending Packet 3...
** Link 1 Sending Packet 4...
** Link 1 Sending Packet 5...
** Link 1 Sending Packet 6...
** Link 1 Sending Packet 7...
** Link 1 Sending Packet 8...
** Link 1 Received Packet 1...
** Link 1 Received Packet 2...
** Link 1 Received Packet 3...
** Link 1 Received Packet 4...
** Link 1 Received Packet 5...
** Link 1 Received Packet 6...
** Link 1 Received Packet 7...
** Link 1 Received Packet 8...
** Link 1 Received Packet 9...
** Link 1 Received Packet 10...
```
Figure 25. Sample Simulation Output when Ethernet Channel is Configured to 4 (Part 2 of 2)

This figure shows a successful simulation test run when the Ethernet channel (i.e., LINK) is configured to 4.

```
** Link 2 Sending Packet 1...
** Link 2 Sending Packet 2...
** Link 2 Sending Packet 3...
** Link 2 Sending Packet 4...
** Link 2 Sending Packet 5...
** Link 2 Sending Packet 6...
** Link 2 Sending Packet 7...
** Link 2 Sending Packet 8...
** Link 2 Received Packet 1...
** Link 2 Received Packet 2...
** Link 2 Sending Packet 9...
** Link 2 Sending Packet 10...
** Link 2 Received Packet 3...
** Link 2 Received Packet 4...
** Link 2 Received Packet 5...
** Link 2 Received Packet 6...
** Link 2 Received Packet 7...
** Link 2 Received Packet 8...
** Link 2 Received Packet 9...
** Link 2 Received Packet 10...
** Link 3 Sending Packet 1...
** Link 3 Sending Packet 2...
** Link 3 Sending Packet 3...
** Link 3 Sending Packet 4...
** Link 3 Sending Packet 5...
** Link 3 Sending Packet 6...
** Link 3 Sending Packet 7...
** Link 3 Sending Packet 8...
** Link 3 Received Packet 1...
** Link 3 Received Packet 2...
** Link 3 Sending Packet 9...
** Link 3 Sending Packet 10...
** Link 3 Received Packet 3...
** Link 3 Received Packet 4...
** Link 3 Received Packet 5...
** Link 3 Received Packet 6...
** Link 3 Received Packet 7...
** Link 3 Received Packet 8...
** Link 3 Received Packet 9...
** Link 3 Received Packet 10...
```

** Testbench complete.
```
```

4.5. Compilation

Follow the procedure in Compiling and Configuring the Design Example in Hardware on page 8 to compile and configure the design example in the selected hardware.
You can estimate resource utilization and Fmax using the compilation-only design example. You can compile your design using the **Start Compilation** command on the **Processing** menu in the Intel Quartus Prime Pro Edition software. A successful compilation generates the compilation report summary.

For more information, refer to *Design Compilation* in the *Compiler User Guide: Intel Quartus Prime Pro Edition*.

**Related Information**

- **Design Compilation**

### 4.6. Hardware Testing

In the hardware design example, you can program the IP core in internal serial loopback mode and generate traffic on the transmit side that loops back through the receive side.

Follow the procedure at the provided related information link to test the design example in the selected hardware.

**Related Information**

- *Testing the 25G Ethernet Intel FPGA IP Design in Hardware* on page 10
  - More information on the procedure and hardware setup.

#### 4.6.1. Test Procedure

Follow these steps to test the design example in hardware:

1. Before you run the hardware testing for this design example, you must reset the system:
   a. Click **Tools ➤ In-System Sources & Probes Editor** tool for the default Source and Probe GUI.
   b. Toggle the system reset signal (Source[0]) from 0 to 1 to apply the reset and return the system reset signal back to 0 to release the system from the reset state.
   c. Monitor the Probe signals and ensure that the status is valid.

2. To perform internal serial loopback test, refer to the *Test Procedure—Design Example Without the IEEE 1588v2 Feature* section of the *10G/25G Ethernet Single-Channel Design Example for Intel Stratix 10 Devices* chapter.

*Note:* link_num is valid for 0 to 3 only.

**Related Information**

- **Design Example Registers** on page 47
- **Test Procedure—Design Example Without the IEEE 1588v2 Feature** on page 23
5. 25G Ethernet Intel FPGA IP Design Example References

This section provides information about the 25G Ethernet Intel FPGA IP core interface signals and registers in the design examples.

5.1. Design Example Interface Signals

The 25G Ethernet Intel FPGA IP core testbench is self-contained and does not require you to drive any input signals.

Table 14. Hardware Design Example Interface Signals for 25G Ethernet Intel FPGA IP Core for Intel Stratix 10 Devices

<table>
<thead>
<tr>
<th>Signal</th>
<th>Direction</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk100</td>
<td>Input</td>
<td>Drive at 100 MHz. The intent is to drive this from a 100 MHz oscillator on the board.</td>
</tr>
<tr>
<td>clk_ref</td>
<td>Input</td>
<td>Drive at 644.53125 MHz or 322.265625 MHz from an oscillator on the board.</td>
</tr>
<tr>
<td>cpu_resetn</td>
<td>Input</td>
<td>Resets the IP core. Active low. Drives the global hard reset csr_reset_n to the IP core.</td>
</tr>
<tr>
<td>tx_serial</td>
<td>Output</td>
<td>Transceiver PHY output serial data.</td>
</tr>
<tr>
<td>rx_serial</td>
<td>Input</td>
<td>Transceiver PHY input serial data.</td>
</tr>
<tr>
<td>user_led[7:0]</td>
<td>Output</td>
<td>Status signals. The hardware design example connects these bits to drive LEDs on the target board. Individual bits reflect the following signal values and clock behavior:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [0]: Main reset signal to IP core</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [1]: Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [2]: Divided version of clk50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [3]: Divided version of 100 MHz status clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [4]: tx_lanes_stable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [5]: rx_block_lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [6]: rx_am_lock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [7]: rx_pcs_ready</td>
</tr>
</tbody>
</table>

Related Information

Interfaces and Signal Descriptions

Provides detailed descriptions of the Intel Stratix 10 core signals and the interfaces to which they belong.
### 5.2. Design Example Registers

#### Table 15. Hardware Design Example Register Map for 25G Ethernet Intel FPGA IP Core for Intel Stratix 10 Devices

You access these registers with the `reg_read` and `reg_write` functions in the System Console.

<table>
<thead>
<tr>
<th>Word Offset</th>
<th>Register Category</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variant: Single-Channel</strong></td>
<td></td>
</tr>
<tr>
<td>0X0000–0X0DFF</td>
<td>Register range to access the Status Registers.</td>
</tr>
<tr>
<td>0X4000–0X7FFF</td>
<td>Register range to access the Reconfiguration Registers.</td>
</tr>
<tr>
<td>0X10000–0X10001</td>
<td>Register range to access the Reconfiguration Registers module for 10G/25G switching.</td>
</tr>
<tr>
<td>0x1020</td>
<td>32-bit <code>average_offset_fnsec_r</code> register:</td>
</tr>
<tr>
<td></td>
<td>• Read this register to obtain average offset value [47:16] in fractional nanosecond, which is derived from the offset adjustment data [96:0] of the PTP slave.</td>
</tr>
<tr>
<td>0x1021</td>
<td>32-bit <code>average_offset_fnsec_to_mem</code> register:</td>
</tr>
<tr>
<td></td>
<td>• Read this register to obtain average offset value [15:0] in fractional nanosecond, which is derived from the offset adjustment data [96:0] of the PTP slave.</td>
</tr>
<tr>
<td></td>
<td>• Bit [31:16]: Reserved.</td>
</tr>
<tr>
<td>0x1030</td>
<td>32-bit <code>average_delay_fnsec_r</code> register:</td>
</tr>
<tr>
<td></td>
<td>• Read this register to obtain average delay value [47:16] in fractional nanosecond, which is derived from the offset adjustment data [191:96] of the PTP slave.</td>
</tr>
<tr>
<td>0x1031</td>
<td>32-bit <code>average_delay_fnsec_to_mem</code> register:</td>
</tr>
<tr>
<td></td>
<td>• Read this register to obtain average delay value [15:0] in fractional nanosecond, which is derived from the offset adjustment data [191:96] of the PTP slave.</td>
</tr>
<tr>
<td></td>
<td>• Bit [31:16]: Reserved.</td>
</tr>
<tr>
<td><strong>Variant: Multi-Channel</strong></td>
<td></td>
</tr>
<tr>
<td>0x00000–0x30DFF</td>
<td>For multi-channel design examples, the base address of all channels are incremented with 0x10000. This corresponds to:</td>
</tr>
<tr>
<td></td>
<td>• Channel 0 Range: 0x00300–00DFF</td>
</tr>
<tr>
<td></td>
<td>• Channel 1 Range: 0x10300–10DFF</td>
</tr>
<tr>
<td></td>
<td>• Channel 2 Range: 0x20300–20DFF</td>
</tr>
<tr>
<td></td>
<td>• Channel 3 Range: 0x30300–30DFF</td>
</tr>
<tr>
<td>0x04000–0x37FFF</td>
<td>For multi-channel design examples, the base address of all channels are incremented with 0x10000. This corresponds to:</td>
</tr>
<tr>
<td></td>
<td>• Channel 0 Range: 0x04000–0x07FFF</td>
</tr>
<tr>
<td></td>
<td>• Channel 1 Range: 0x14000–0x17FFF</td>
</tr>
<tr>
<td></td>
<td>• Channel 2 Range: 0x24000–0x27FFF</td>
</tr>
<tr>
<td></td>
<td>• Channel 3 Range: 0x34000–0x37FFF</td>
</tr>
</tbody>
</table>
Note: 1. For Intel Stratix 10 H-tile production device, disable the background calibration prior to accessing the transceiver core reconfiguration register, as described in the Disabling Background Calibration section of the 25G Ethernet Intel FPGA IP User Guide.

2. Dynamic reconfiguration switching for 10G/25G is not available for multi-channel designs.

3. For single-channel design example, 0x4000 is the base address of the PHY registers. For example, to read the background calibration register, type `reg_read 0x4542`.

4. For multi-channel design example, the base address of the PHY registers is `0x4000 + (0x10000 * <link num>)`. For example, to read the background calibration register at channel 2, type `reg_read 0x24542`.

Related Information

- Control, Status, and Statistics Register Descriptions section of the 25G Ethernet Intel FPGA IP User Guide
  
  Describes the 25G Ethernet Intel FPGA IP control, status, and statistics registers.

- Disabling Background Calibration section of the 25G Ethernet Intel FPGA IP User Guide

5.3. Using Transceiver Toolkit on H-Tile Production Device

If your design example targets the H-tile production device and Enable autoadaptation triggering for RX PMA CTLE/DFE mode option is turned on, you must perform additional steps to configure the register 0x343 bit[0] before you can use the Transceiver Toolkit. Refer to the description for register 0x343 in the 25G Ethernet Intel Stratix 10 FPGA IP User Guide for more information.

Follow these steps to use Transceiver Toolkit:

1. Follow the procedure in the Testing the 25G Ethernet Intel FPGA IP Design in Hardware section to load the main.tcl script.

2. For single-channel design example, type `reg_write 0x343 0x1` to hold the autoadaptation module FSM in idle state.

3. For multi-channel design example,

   a. type `reg_write 0x343 0x1` for channel 0
   b. type `reg_write 0x10343 0x1` for channel 1
   c. type `reg_write 0x20343 0x1` for channel 2
   d. type `reg_write 0x30343 0x1` for channel 3

4. Launch the Transceiver Toolkit.

Note: If the register 0x343 bit[0] is not set (1'b1), the transceiver channel is not visible in the Transceiver Toolkit.

Follow these steps after you already used the Transceiver Toolkit:
1. Close the Transceiver Toolkit.
2. For single-channel design example, type `reg_write 0x343 0x0` to re-start the auto adaptation module FSM.
3. For multi-channel design example,
   a. `type reg_write 0x343 0x0` for channel 0
   b. `type reg_write 0x10343 0x0` for channel 1
   c. `type reg_write 0x20343 0x0` for channel 2
   d. `type reg_write 0x30343 0x0` for channel 3

*Note:* If the register `0x343` bit[0] is cleared (1'b0) when you opened the Transceiver Toolkit, the System Console may hang.

**Related Information**
- [25G Ethernet Intel Stratix 10 FPGA IP User Guide](#)
- [Testing the 25G Ethernet Intel FPGA IP Design in Hardware](#) on page 10
### 6. 25G Ethernet Intel Stratix 10 FPGA IP Design Example User Guide Archives

IP versions are the same as the Intel Quartus Prime Design Suite software versions up to v19.1. From Intel Quartus Prime Design Suite software version 19.2 or later, IP cores have a new IP versioning scheme.

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

<table>
<thead>
<tr>
<th>IP Core Version</th>
<th>User Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.1</td>
<td>25G Ethernet Intel Stratix 10 FPGA IP Design Example User Guide</td>
</tr>
<tr>
<td>18.1</td>
<td>25G Ethernet Intel Stratix 10 FPGA IP Design Example User Guide</td>
</tr>
<tr>
<td>18.0</td>
<td>25G Ethernet Intel Stratix 10 FPGA IP Design Example User Guide</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Intel Quartus Prime Version</th>
<th>IP Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020.02.14</td>
<td>19.2</td>
<td>19.2.0</td>
<td>Updated the procedure steps in the Changing Target Device in Hardware Design Example section.</td>
</tr>
<tr>
<td>2019.12.13</td>
<td>19.2</td>
<td>19.2.0</td>
<td>Updated the procedure steps in the Changing Target Device in Hardware Design Example section.</td>
</tr>
<tr>
<td>2019.08.29</td>
<td>19.2</td>
<td>19.2.0</td>
<td>Updated the instruction for the ModelSim simulator in Table: Steps to Simulate the Testbench.</td>
</tr>
</tbody>
</table>
| 2019.07.01       | 19.2                        | 19.2.0     | • Added new topic—Changing Target Device in Hardware Design Example.  
• Updated references to Intel Stratix 10 L-Tile GX Transceiver Signal Integrity Development Kit (OPN: 1SX280LU2F50E2VG) as Intel Stratix 10 GX Signal Integrity L-Tile (Production) Development Kit (OPN: 1SX280LU2F50E1VG).  
• Updated the Generating the Design Example topic to add a note to Step 8.  
• Updated Figure: Example Design Tab in the 25G Ethernet Intel FPGA IP Parameter Editor.  
• Updated the Hardware and Software Requirements topics for all design example chapters. |

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 2019.05.10       | 19.1                        | • Renamed Altera Debug Master Endpoint (ADME) to Native PHY Debug Master Endpoint (NPDME).  
• Updated the hardware testing steps for the following topics:  
  — Test Procedure—Design Example Without the IEEE 1588v2 Feature for 10G/25G Ethernet single-channel design example.  
  — Test Procedure—Design Example with the IEEE 1588v2 Feature for 10G/25G Ethernet single-channel design example.  
  — Test Procedure—Design Example With and Without the IEEE 1588v2 Feature for 25G Ethernet single-channel design example.  
  — Test Procedure for 25G Ethernet multi-channel design example.  
• Updated the following Figures:  
  — Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC+PCS+PMA Core Variant) Without the IEEE 1588v2 Feature  
  — Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC+PCS+PMA Core Variant) with the IEEE 1588v2 Feature  
  — Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC+PCS Core Variant) Without the IEEE 1588v2 Feature  
  — Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC+PCS Core Variant) with the IEEE 1588v2 Feature |

*Other names and brands may be claimed as the property of others.*
<table>
<thead>
<tr>
<th>Document Version</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 2019.01.07       | 18.1                      | • Updated the Generating the Design Example topic to correct target development kit in Step 8 from Stratix 10 GX FPGA Development Kit to Intel Stratix 10 L-Tile GX Transceiver Signal Integrity Development Kit.  
  • Updated Figure: Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC+PCS+PMA Core Variant) with the IEEE 1588v2 Feature.  
  • Added a note to Design Example Registers topic. |
| 2018.10.03       | 18.1                      | • Updated Table: Parameters in the Example Design Tab to update the description for Select Board.  
  • Updated the Hardware and Software Requirements topics for all design example chapters.  
  • Updated the Design Components topics for all design example chapters.  
  • Added new Figures:  
    — Block Diagram—25G Ethernet Single-Channel Design Example (MAC +PCS Core Variant) Without the IEEE 1588v2 Feature  
    — Block Diagram—25G Ethernet Single-Channel Design Example (MAC +PCS Core Variant) Without the IEEE 1588v2 Feature  
    — Block Diagram—25G Ethernet Single-Channel Design Example (MAC +PCS Core Variant) Without the IEEE 1588v2 Feature  
    — Block Diagram—25G Ethernet Single-Channel Design Example (MAC +PCS Core Variant) with the IEEE 1588v2 Feature  
  • Updated Figures:  
    — Example Design Tab in the 25G Ethernet Intel FPGA IP Parameter Editor  
    — Block Diagram—25G Ethernet Single-Channel Design Example (MAC +PCS+PMA Core Variant) with the IEEE 1588v2 Feature  
    — Block Diagram—25G Ethernet Single-Channel Design Example (MAC +PCS+PMA Core Variant) Without the IEEE 1588v2 Feature  
    — Block Diagram—25G Ethernet Multi-Channel Design Example (MAC +PCS+PMA Core Variant)  
    — Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC+PCS+PMA) Without the IEEE 1588v2 Feature  
    — Block Diagram—10G/25G Ethernet Single-Channel Design Example (MAC+PCS+PMA) with the IEEE 1588v2 Feature  
  • Removed Figure: Sample Simulation Output for Design Example with the IEEE 1588v2 Feature (Part 1 of 2) and Sample Simulation Output for Design Example with the IEEE 1588v2 Feature (Part 2 of 2).  
  • Updated the simulation sample output of the Test Case—Design Example with the IEEE 1588v2 Feature topic for 25G Ethernet Single-Channel design example.  
  • Updated the simulation sample output of the Test Case—Design Example with the IEEE 1588v2 Feature topic for the 10G/25G Ethernet design example.  
  • Restructured descriptions for Features topics for all design example chapters.  
  • Streamlined the contents and document organization. |
| 2018.06.25       | 18.0                      | Initial release. |