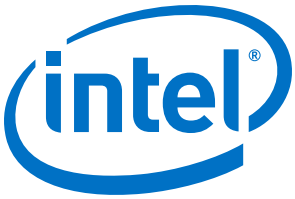


Intel® Stratix® 10 H-tile Hard IP for Ethernet Design Example User Guide

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

Subscribe
Send Feedback

UG-20122 | 2017.11.29
Latest document on the web: **PDF** | **HTML**

Contents

1 Quick Start Guide

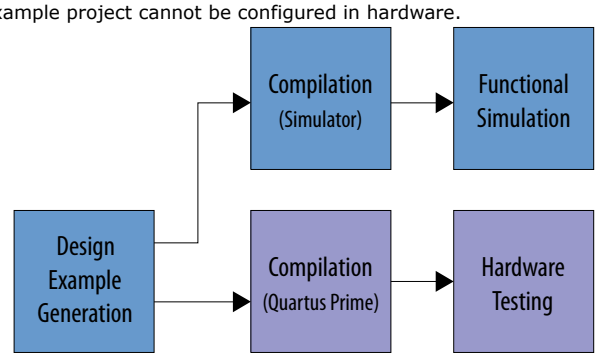
The Intel® Stratix® 10 H-tile Hard IP for Ethernet IP core provides a simulation testbench. When you generate the design example, the parameter editor automatically creates the files necessary to simulate the design.

In addition, Intel provides a compilation-only example project that you can use to quickly estimate IP core area and timing.

You can generate the simulation testbench and compilation-only example project for any Intel Stratix 10 H-tile HIP for Ethernet IP core variation: you can select MAC+PCS or PCS only, 100-Gbps or 50-Gbps Ethernet data rate, and any legal combination of parameter editor settings, and generate the design example for your IP core variation.

Figure 1. Development Steps for the Design Example

Future releases of the IP core also provide a hardware design example you can compile and test in hardware. The compilation-only example project cannot be configured in hardware.



1.1 Directory Structure

Figure 2. Intel Stratix 10 H-tile HIP for Ethernet Design Example Directory Structure

<*datarate*> is either "50" or "100", depending on your IP core variation.

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.

1.2 Simulation Design Example Components

Figure 3. Intel Stratix 10 H-tile HIP for Ethernet 100G Variation Simulation Design Example Block Diagram

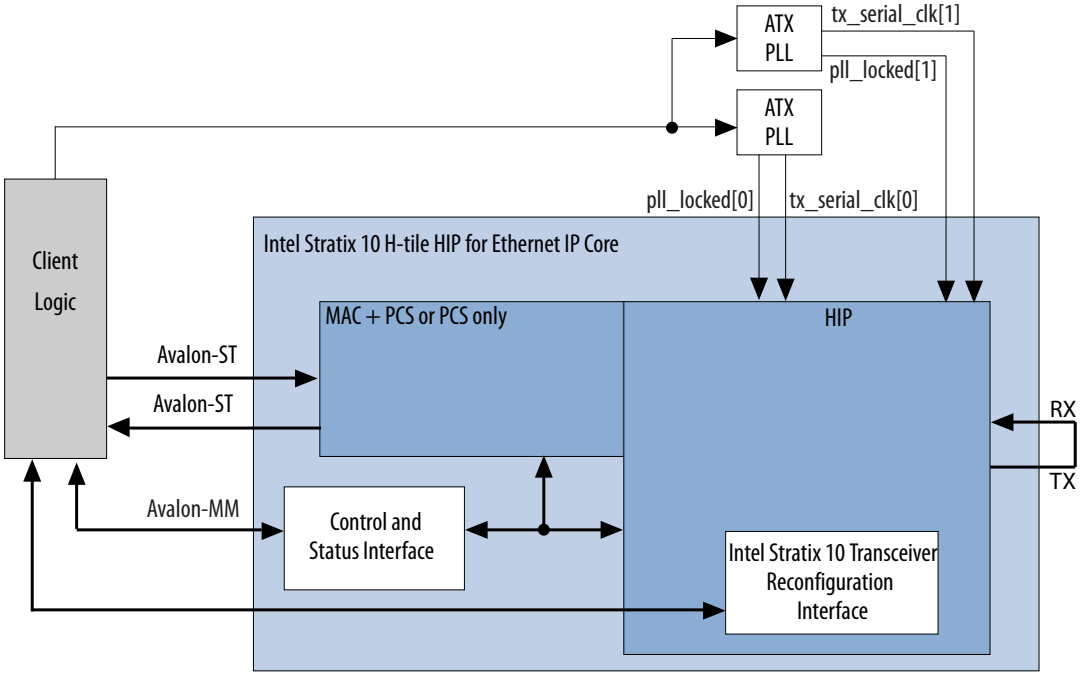
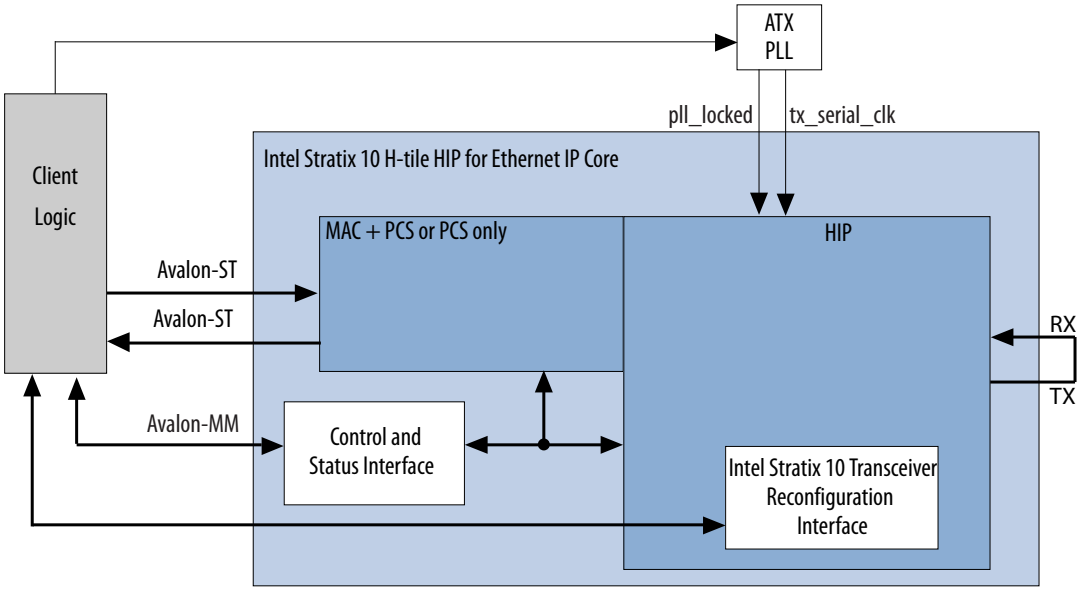


Figure 4. Intel Stratix 10 H-tile HIP for Ethernet 50G Variation Simulation Design Example Block Diagram



The simulation design example top-level test file is `basic_avl_tb_top.sv`. This file instantiates and connects one or two ATX PLLs, depending on the number of transceivers the IP core configures. It includes a task to send and receive packets. MAC+PCS variations send and receive ten packets. PCS Only variations send, receive, and check fourteen packets.

Table 1. Intel Stratix 10 H-tile HIP for Ethernet IP Core Testbench File Descriptions

File Names	Description
Key Testbench and Simulation Files	
`basic_avl_tb_top.sv`	Top-level testbench file. The testbench instantiates the DUT and runs Verilog HDL tasks to generate and accept packets.
Testbench Scripts	
`run_vsim.do`	The Mentor Graphics ModelSim* script to run the testbench.
`run_vcs.sh`	The Synopsys VCS* script to run the testbench.
`run_ncsim.sh`	The Cadence NCSim* script to run the testbench.

1.3 Generating the Design

Figure 5. Procedure

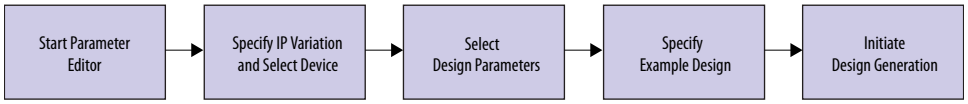
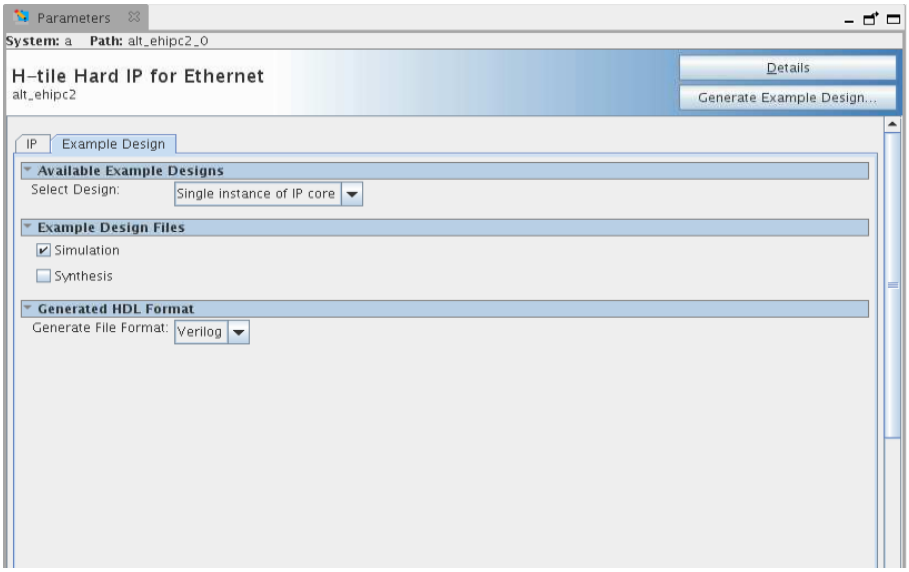


Figure 6. Example Design Tab in the Intel Stratix 10 H-tile HIP for Ethernet Parameter Editor



Follow these steps to generate the Intel Stratix 10 H-tile HIP for Ethernet hardware design example and testbench:

1. If you do not already have an Intel Quartus® Prime Pro Edition project in which to integrate your Intel Stratix 10 H-tile HIP for Ethernet IP core, you must create one.
   a. In the Intel Quartus Prime Pro Edition, click **File ➤ New Project Wizard** to create a new Quartus Prime project, or **File ➤ Open Project** to open an existing Intel Quartus Prime project. The wizard prompts you to specify a device.
   b. Specify the device family **Intel Stratix 10** and select a device that meets all of these requirements:
      - Transceiver tile is H-tile
      - Transceiver speed grade is 1 or 2
      - Core speed grade is 1 or 2
   c. Click **Finish**.
2. In the IP Catalog, locate and select **H-tile Hard IP for Ethernet**. The **New IP Variation** window appears.
3. Specify a top-level name <*your_ip*> for your custom IP variation. The parameter editor saves the IP variation settings in a file named <*your_ip*>`.ip`.
4. Click **OK**. The parameter editor appears.
5. On the **IP** tab, specify the parameters for your IP core variation.
6. On the **Example Design** tab, under **Example Design Files**, select the **Simulation** option to generate the testbench and the compilation-only project.

   Note: You must select the **Simulation** option to generate the testbench.
7. On the **Example Design** tab, under **Generated HDL Format**, select **Verilog** HDL or **VHDL**

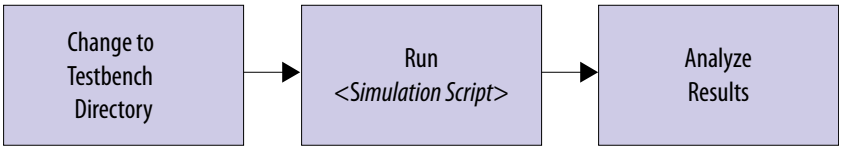
   Note: If you select **VHDL**, you must simulate the testbench with a mixed-language simulator. The device under test in the `ex_50G` or `ex_100G` directory is a VHDL model, but the main testbench file is a System Verilog file.
8. Click the **Generate Example Design** button. The **Select Example Design Directory** window appears.
9. If you wish to modify the design example directory path or name from the defaults displayed (`alt_ehipc2_0_example_design`), browse to the new path and type the new design example directory name (<*design_example_dir*>).

Related Links

IP Core Parameters
Provides more information about customizing your IP core.

1.4 Simulating the Intel Stratix 10 H-tile HIP for Ethernet Design Example Testbench

Figure 7. Procedure



Follow these steps to simulate the testbench:

1. Change to the testbench simulation directory *<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. Refer to the table *Steps to Simulate the Testbench*.
3. Analyze the results. The successful testbench sends ten or fourteen packets, receives the same number of packets, and displays `"Testbench complete."`

Table 2. Steps to Simulate the Testbench

Simulator	Instructions
Mentor Graphics ModelSim	In the command line, type `vsim -do run_vsim.do` If you prefer to simulate without bringing up the ModelSim GUI, type `vsim -c -do run_vsim.do` *Note:* The ModelSim - Intel FPGA Edition simulator does not have the capacity to simulate this IP core. You must use another supported ModelSim simulator such as ModelSim SE.
Cadence NCSim	In the command line, type `sh run_ncsim.sh`
Synopsys VCS	In the command line, type `sh run_vcs.sh`

The successful test run displays output confirming the following behavior:

1. Waiting for the ATX PLL or ATX PLLs to lock.
2. Waiting for RX transceiver reset to complete.
3. Waiting for RX alignment.
4. Sending ten (MAC+PCS) or fourteen (PCS Only) packets.
5. Receiving those packets (MAC+PCS) or receiving and checking those packets (PCS Only).
6. Displaying `Testbench complete.`

The following sample output illustrates a successful simulation test run for a 100-Gbps, MAC+PCS IP core variation. Times are in picoseconds. The times required for simulation of 100-Gbps variations are slightly longer than the times required for 50-Gbps variations.

```
Ref clock is 644.53125 MHz

================================================
Module ct1_hssi_cr2_ehip_pcs_interface
================================================
silicon_rev = 14nm5bcr2ea
```

```
waiting for o_tx_lanes_stable...
o_tx_lanes_stable is 1 at time               525000
waiting for tx_dll_lock....
TX DLL LOCK is 1 at time             25332363
waiting for tx_transfer_ready....
TX transfer ready is 1 at time             25652235
waiting for rx_transfer_ready....
RX transfer ready is 1 at time             31979703
EHIP PLD Ready out is 1 at time             32040000
EHIP reset out is 0 at time            32304000
EHIP reset ack is 0 at time            32612783
EHIP TX reset out is 0 at time             32928000
EHIP TX reset ack is 0 at time             82562795
waiting for EHIP Ready....
EHIP READY is 1 at time            82629435
EHIP RX reset out is 0 at time             82968000
waiting for rx reset ack....
EHIP RX reset ack is 0 at time             83029275
Waiting for RX Block Lock
EHIP RX Block Lock  is high at time             90223063
Waiting for AM lock
EHIP RX AM Lock  is high at time             91172683
Waiting for RX alignment
RX deskew locked
RX lane aligmnent locked
TX enabled
** Sending Packet           1...
** Sending Packet           2...
** Sending Packet           3...
** Sending Packet           4...
** Sending Packet           5...
** Sending Packet           6...
** Received Packet          1...
** Sending Packet           7...
** Received Packet          2...
** Sending Packet           8...
** Received Packet          3...
** Sending Packet           9...
** Received Packet          4...
** Sending Packet          10...
** Received Packet          5...
** Received Packet          6...
** Received Packet          7...
** Received Packet          8...
** Received Packet          9...
** Received Packet         10...
**
** Testbench complete.
**
****************************************
```

The following sample output illustrates a successful simulation test run for a 100-Gbps, PCS Only IP core variation. Times are in picoseconds.

```
Ref clock is run at 625 MHz so whole numbers can used for all clock periods.
Multiply reported frequencies by 33/32 to get actual clock frequencies.

================================================
Module ct1_hssi_cr2_ehip_pcs_interface
================================================
silicon_rev = 14nm5bcr2ea

waiting for o_tx_lanes_stable...
o_tx_lanes_stable is 1 at time               525000
waiting for tx_dll_lock....
TX DLL LOCK is 1 at time             25332363
waiting for tx_transfer_ready....
TX transfer ready is 1 at time             25652235
waiting for rx_transfer_ready....
RX transfer ready is 1 at time             31979703
```

```
EHIP PLD Ready out is 1 at time              32040000
EHIP reset out is 0 at time             32304000
EHIP reset ack is 0 at time             32612783
EHIP TX reset out is 0 at time             32928000
EHIP TX reset ack is 0 at time             82562795
waiting for EHIP Ready....
EHIP READY is 1 at time             82669419
EHIP RX reset out is 0 at time             83008000
waiting for rx reset ack....
EHIP RX reset ack is 0 at time             83055931
Waiting for RX Block Lock
EHIP Rx Block Lock  is high at time              90697873
Waiting for AM lock
EHIP Rx am Lock  is high at time              92122303
Waiting for RX alignment
RX deskew locked
RX lane aligmnent locked
Sending Packets
Receiving Packets
** Received Packet
0101010101010101fefefefefefefefe0101010101010101fefefefefefefefe...
** Matched packet   1 ...
** Received Packet
0202020202020202fdfdfdfdfdfdfdfd0202020202020202fdfdfdfdfdfdfdfd...
** Matched packet   2 ...
** Received Packet
0303030303030303fcfcfcfcfcfcfcfc0303030303030303fcfcfcfcfcfcfcfc...
** Matched packet   3 ...
** Received Packet
0404040404040404fbfbfbfbfbfbfbfb0404040404040404fbfbfbfbfbfbfbfb...
** Matched packet   4 ...
** Received Packet
0505050505050505fafafafafafafafa0505050505050505fafafafafafafafa...
** Matched packet   5 ...
** Received Packet
0606060606060606f9f9f9f9f9f9f9f906060606060606060606f9f9f9f9f9f9f9f9...
** Matched packet   6 ...
** Received Packet
0707070707070707f8f8f8f8f8f8f8f807070707070707070707f8f8f8f8f8f8f8f8...
** Matched packet   7 ...
** Received Packet
0808080808080808f7f7f7f7f7f7f7f708080808080808080808f7f7f7f7f7f7f7f7...
** Matched packet   8 ...
** Received Packet
0909090909090909f6f6f6f6f6f6f6f609090909090909090909f6f6f6f6f6f6f6f6...
** Matched packet   9 ...
** Received Packet
0a0a0a0a0a0a0a0af5f5f5f5f5f5f5f50a0a0a0a0a0a0a0a0af5f5f5f5f5f5f5f5...
** Matched packet  10 ...
** Received Packet
0b0b0b0b0b0b0b0bf4f4f4f4f4f4f4f40b0b0b0b0b0b0b0b0bf4f4f4f4f4f4f4f4...
** Matched packet  11 ...
** Received Packet
0c0c0c0c0c0c0c0cf3f3f3f3f3f3f3f30c0c0c0c0c0c0c0c0cf3f3f3f3f3f3f3f3...
** Matched packet  12 ...
** Received Packet
0d0d0d0d0d0d0d0df2f2f2f2f2f2f2f20d0d0d0d0d0d0d0d0df2f2f2f2f2f2f2f2...
** Matched packet  13 ...
** Received Packet
0e0e0e0e0e0e0e0ef1f1f1f1f1f1f1f10e0e0e0e0e0e0e0e0ef1f1f1f1f1f1f1f1...
** Matched packet  14 ...
**
** Testbench complete.
**
****************************************
```

1.5 Compiling the Compilation-Only Project

To compile the compilation-only example project, follow these steps:

1. Ensure compilation design example generation is complete.
2. In the Intel Quartus Prime Pro Edition software, open the Intel Quartus Prime Pro Edition project `<design_example_dir>/compilation_test_design/alt_ehipc2.qpf`.
3. On the Processing menu, click **Start Compilation**.

After successful compilation, reports for timing and for resource utilization are available in your Intel Quartus Prime Pro Edition session.

Related Links

Block-Based Design Flows

2 Design Example Description

The design example demonstrates the basic functions of the Intel Stratix 10 H-tile HIP for Ethernet IP core. You can generate the design from the **Example Design** tab in the Intel Stratix 10 H-tile HIP for Ethernet parameter editor.

To generate the design example, you must first set the parameter values for the IP core variation you intend to generate in your end product. Generating the design example creates a copy of the IP core; the testbench and compilation-only design example use this variation as the DUT. If you do not set the parameter values for the DUT to match the parameter values in your end product, the design example you generate does not exercise the IP core variation you intend.

Note: The testbench demonstrates a basic test of the IP core. It is not intended to be a substitute for a full verification environment. You must perform more extensive verification of your own Intel Stratix 10 H-tile HIP for Ethernet design in simulation and in hardware.

Related Links

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

2.1 Design Example Behavior

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

To speed up simulation, the IP core simulation model sends alignment marker tags at shorter intervals than required by the IEEE Ethernet standard. The standard specifies an alignment marker interval of 16384 words in each virtual lane. The simulation model with the testbench implements an alignment marker interval of 512 words.

2.2 Design Example Interface Signals

The Intel Stratix 10 H-tile HIP for Ethernet testbench is self-contained and does not require you to drive any input signals.

Related Links

Interfaces and Signal Descriptions
Provides detailed descriptions of the Intel Stratix 10 H-tile HIP for Ethernet IP core signals and the interfaces to which they belong.

A Document Revision History

Table 3. Revision History

Date	Changes
2017.11.29	Initial release