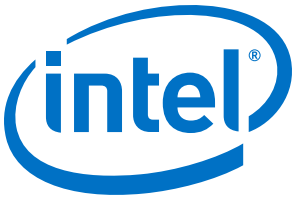


Intel® Stratix® 10 Low Latency 100G Ethernet Design Example User Guide

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

Subscribe
Send Feedback

UG-20086 | 2017.11.06
Latest document on the web: PDF | HTML

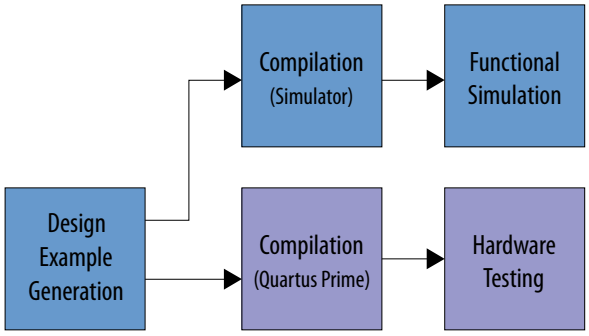
Contents

1 Quick Start Guide

The Intel® Stratix® 10 Low Latency (LL) 100-Gbps Ethernet (100GbE) IP core provides a simulation testbench and a hardware design example that supports compilation and hardware testing. When you generate the design example, the parameter editor automatically creates the files necessary to simulate, compile, and test the design in hardware. You can download the compiled hardware design to the Intel Stratix 10 GX Transceiver Signal Integrity Development Kit.

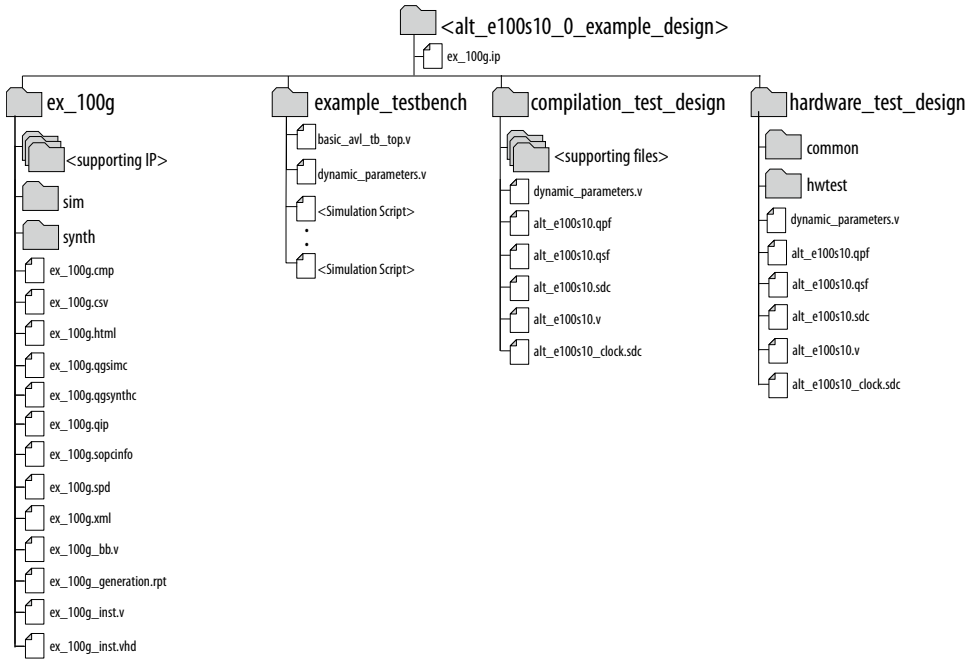
In addition, Intel provides a compilation-only example project that you can use to quickly estimate IP core area and timing.

Figure 1. Development Steps for the Design Example



1.1 Directory Structure

Figure 2. Intel Stratix 10 LL 100GbE Design Example Directory Structure



The hardware configuration and test files (the hardware design example) are located in <*design_example_dir*>/hardware_test_design. 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 LL 100GbE Simulation Design Example Block Diagram

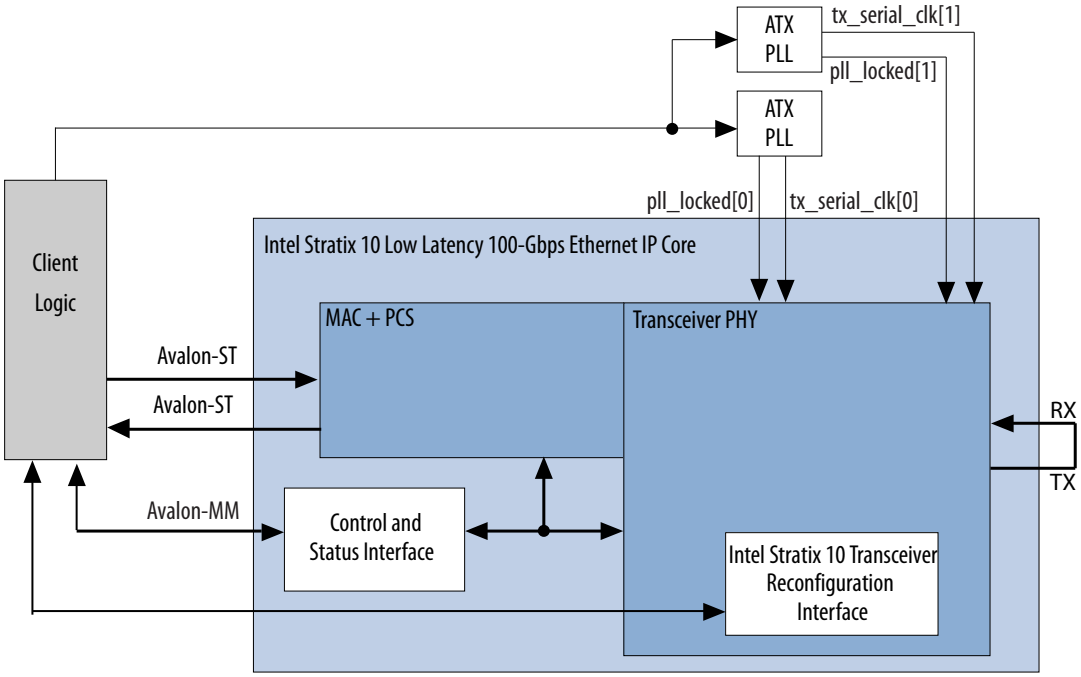
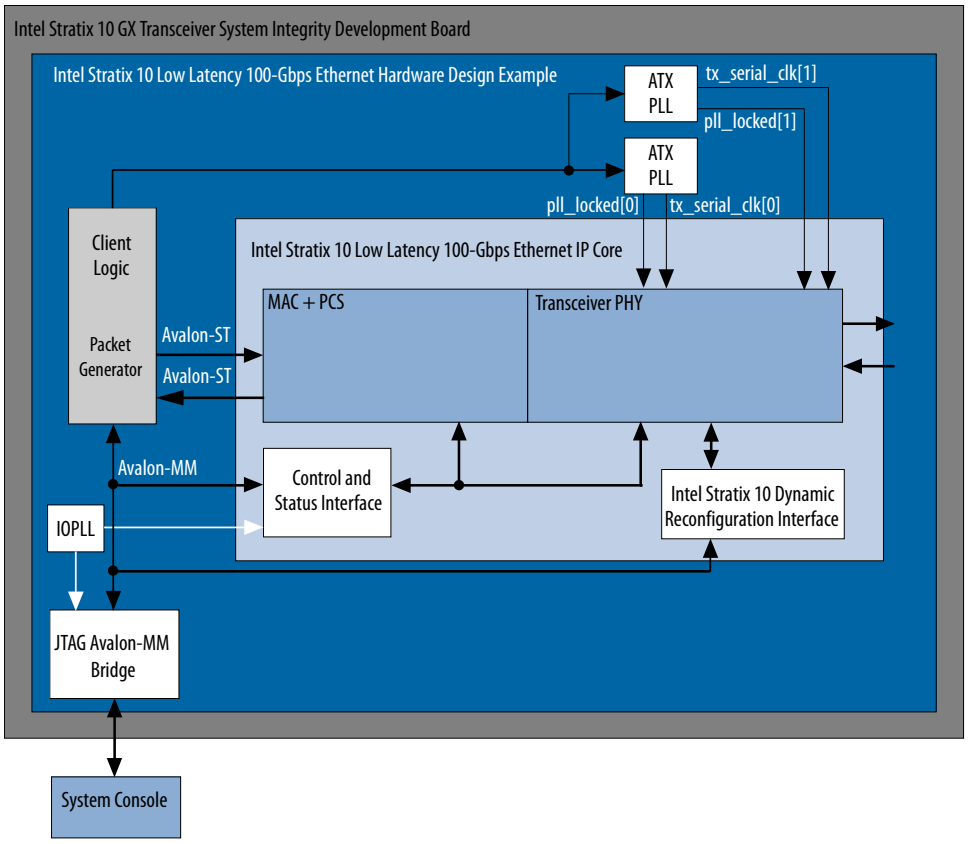


Table 1. Intel Stratix 10 LL 100GbE IP Core Testbench File Descriptions

File Names	Description
Key Testbench and Simulation Files	
`basic_avl_tb_top.v`	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 Hardware Design Example Components

Figure 4. Intel Stratix 10 LL 100GbE Hardware Design Example High Level Block Diagram



The Intel Stratix 10 LL 100GbE hardware design example includes the following components:

- Intel Stratix 10 LL 100GbE IP core.
- Client logic that coordinates the programming of the IP core and packet generation.
- Two ATX PLLs to drive the device transceiver channels.
- IOPLL to generate a 100 MHz clock from a 50 MHz input clock to the hardware design example.
- JTAG controller that communicates with the System Console. You communicate with the client logic through the System Console.

Table 2. Intel Stratix 10 LL 100GbE IP Core Hardware Design Example File Descriptions

File Names	Description
`alt_e100s10.qpf`	Intel Quartus® Prime project file
`alt_e100s10.qsf`	Intel Quartus Prime project settings file

continued...

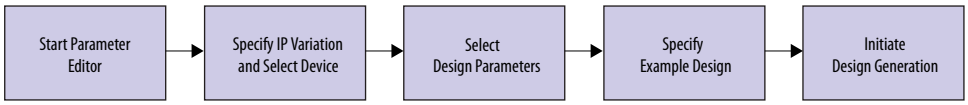
File Names	Description
`alt_e100s10.sdc`, `alt_e100s10_clock.sdc`	Synopsys Design Constraints files. You can copy and modify these files for your own Intel Stratix 10 LL 100GbE design.
`alt_e100s10.v`	Top-level Verilog HDL design example file
`common/`	Hardware design example support files
`hwtest/main.tcl`	Main file for accessing System Console

Related Links

Intel Stratix 10 GX Signal Integrity Development Kit Webpage

1.4 Generating the Design

Figure 5. Procedure



Follow these steps to generate the Intel Stratix 10 LL 100GbE 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 LL 100GbE 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 L-tile or H-tile (any transceiver tile)
      - Transceiver speed grade is 1 or 2
      - Core speed grade is 1 or 2
      - Device is not an 1SG280L ES1 device (part name 1SG280L...VGS1)
   c. Click **Finish**.
2. In the IP Catalog, locate and select **Low Latency 100G 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 select the **Synthesis** option to generate the compilation-only and hardware design examples.

   Note: You must select at least one of the **Simulation** and **Synthesis** options to generate the design example.

7. On the **Example Design** tab, under **Generated HDL Format**, only Verilog HDL is available. This IP core does not support VHDL.
8. Under **Target Development Kit** select the **Stratix 10 GX Transceiver Signal Integrity Development Kit** or select **None**.

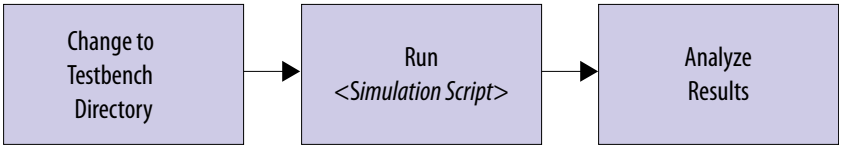
   Note: The compilation-only and hardware design examples target your project device. For correct hardware design functionality out of the box, you must ensure your project device is the device on your development kit.
9. Click the **Generate Example Design** button. The **Select Example Design Directory** window appears.
10. If you wish to modify the design example directory path or name from the defaults displayed (`alt_e100s10_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.
- Intel Stratix 10 GX Signal Integrity Development Kit Webpage

1.5 Simulating the Intel Stratix 10 LL 100GbE Design Example Testbench

Figure 6. 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 packets, receives ten packets, and displays `"Testbench complete."`

Table 3. 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 PLLs to lock.
2. Waiting for RX transceiver reset to complete.
3. Waiting for RX alignment.
4. Sending ten packets.
5. Receiving ten packets.
6. Displaying `Testbench complete.`

The following sample output illustrates a successful simulation test run:

```
ATX PLLs Locked
***************************************************
**************  Transmit Ready   *****************
***************************************************
Waiting for the receiver to be ready
Receive transceivers are out of reset
Waiting for RX alignment
Rx Alignment Status Update 1/4: Word/Block lock is acquired over all virtual
lanes
Rx Alignment Status Update 2/4: Virtual lanes Ordered
Rx Alignment Status Update 3/4: RX deskew lock acquired
Rx Alignment Status Update 4/4: RX alignment lock acquired
Rx is fully aligned with Tx
***************************************************
**************   Receive Ready   ******************
***************************************************
Transmitting test data
** Sending Packet           1...
** Sending Packet           2...
** Sending Packet           3...
** Sending Packet           4...
** Sending Packet           5...
** Sending Packet           6...
** Sending Packet           7...
** Sending Packet           8...
** Sending Packet           9...
** Sending Packet          10...
** Received Packet          1...
** Received Packet          2...
** Received Packet          3...
** Received Packet          4...
** Received Packet          5...
** Received Packet          6...
** Received Packet          7...
** Received Packet          8...
** Received Packet          9...
** Received Packet         10...
**
** Testbench complete.
**
****************************************
```

1.6 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 software, open the Intel Quartus Prime project `<design_example_dir>/compilation_test_design/alt_e100s10.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

1.7 Compiling and Configuring the Design Example in Hardware

To compile the hardware design example and configure it on your Intel Stratix 10 device, follow these steps:

1. Ensure hardware design example generation is complete.
2. In the Intel Quartus Prime Pro Edition software, open the Intel Quartus Prime project `<design_example_dir>/hardware_test_design/alt_e100s10.qpf`.
3. On the Processing menu, click **Start Compilation**.
4. After successful compilation, a `.sof` file is available in your specified directory. 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 Transceiver Signal Integrity Development Kit to which your Intel Quartus Prime Pro Edition session can connect.
   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**.

Related Links

- Block-Based Design Flows
- Programming Intel FPGA Devices
- Analyzing and Debugging Designs with System Console

1.8 Testing the Intel Stratix 10 LL 100GbE Hardware Design Example

After you compile the Intel Stratix 10 LL 100GbE 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. After the hardware design example is configured on the Intel Stratix 10 device, in the Intel Quartus Prime Pro Edition software, on the **Tools** menu, click **System Debugging Tools ➤ System Console**.
2. In the Tcl Console pane, type `cd hwtest` to change directory to `<design_example_dir>/hardware_test_design/hwtest`.
3. Type `source main.tcl` to open a connection to the JTAG master.

You can program the IP core with the following design example commands:

- `chkphy_status`: Displays the clock frequencies and PHY lock status.
- `chkmac_stats`: Displays the values in the MAC statistics counters.
- `clear_all_stats`: Clears the IP core statistics counters.
- `start_pkt_gen`: Starts the packet generator.
- `stop_pkt_gen`: Stops the packet generator.
- `loop_on`: Turns on internal serial loopback
- `loop_off`: Turns off internal serial loopback.
- `reg_read <addr>`: Returns the IP core register value at <*addr*>.
- `reg_write <addr> <data>`: Writes <*data*> to the IP core register at address <*addr*>.

Related Links

Analyzing and Debugging Designs with System Console

2 Design Example Description

The design example demonstrates the functions of the Intel Stratix 10 LL 100GbE IP core. You can generate the design from the **Example Design** tab in the Intel Stratix 10 LL 100GbE 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 hardware 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 LL 100GbE design in simulation and in hardware.

Related Links

Intel Stratix 10 Low Latency 100-Gbps 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. 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.

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 LL 100GbE testbench is self-contained and does not require you to drive any input signals.

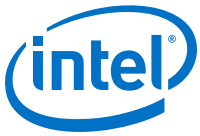


Table 4. Intel Stratix 10 LL 100GbE Hardware Design Example Interface Signals

Signal	Direction	Comments
`clk50`	Input	Drive at 50 MHz. The intent is to drive this from a 50 Mhz oscillator on the board.
`clk_ref_r`	Input	Drive at 644.53125 MHz.
`cpu_resetn`	Input	Resets the IP core. Active low. Drives the global hard reset `csr_reset_n` to the IP core.
`tx_serial[3:0]`	Output	Transceiver PHY output serial data.
`rx_serial[3:0]`	Input	Transceiver PHY input serial data.
`user_led[7:0]`	Output	Status signals. Currently the design example drives all of these signals to a constant value of 0.

Related Links

Interfaces and Signal Descriptions
Provides detailed descriptions of the Intel Stratix 10 LL 100GbE IP core signals and the interfaces to which they belong.

2.3 Intel Stratix 10 LL 100GbE Design Example Registers

Table 5. Intel Stratix 10 LL 100GbE Hardware Design Example Register Map

Lists the memory mapped register ranges for the hardware design example. You access these registers with the `reg_read` and `reg_write` functions in the System Console.

Word Offset	Register Category
0x300–0xDFF	Intel Stratix 10 LL 100GbE IP core registers.
0x4000–0x5FFF	Intel Stratix 10 dynamic reconfiguration registers. Register base address is 0x4000 for Lane 0, 0x4800 for Lane 1, 0x5000 for Lane 2, and 0x5800 for Lane 3.

Related Links

Intel Stratix 10 LL 100GbE IP core register descriptions

A Document Revision History

Table 6. Revision History

Date	Changes
2017.11.06	Initial release