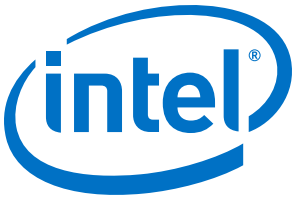


Low Latency E-Tile 40G Ethernet Intel® FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **20.2**

IP Version: **20.0.0**

UG-20273 | 2020.06.22

Latest document on the web: **PDF** | **HTML**

Subscribe

Send Feedback

Contents

1. Quick Start Guide....3
1.1. Generating the Design Example....3
1.1.1. Design Example Parameters....5
1.2. Directory Structure....6
1.3. Simulating the Design Example Testbench....7
1.4. Compiling and Configuring the Design Example in Hardware....8
1.5. Changing Target Device in Hardware Design Example....8
1.6. Testing the Low Latency E-Tile 40G Ethernet Intel FPGA IP Design in Hardware....9

2. Design Example Description....11
2.1. Features....11
2.2. Hardware and Software Requirements....11
2.3. Functional Description....11
2.4. Simulation....12
2.5. Hardware Testing....13
2.5.1. Internal Loopback Test....14
2.5.2. External Loopback Test....15
2.6. Low Latency E-Tile 40G Ethernet Design Example Registers....16
2.7. Design Example Interface Signals....17

A. Low Latency E-Tile 40G Ethernet Intel FPGA IP Archives....19

4. Document Revision History for Low Latency E-tile 40G Ethernet Design Example User Guide....20

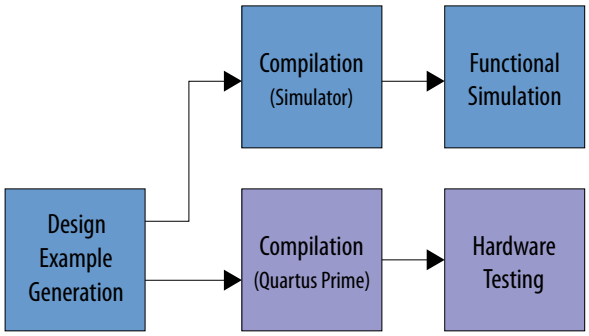
1. Quick Start Guide

The Low Latency E-Tile 40G Ethernet Intel® FPGA IP core provides a simulation testbench and a hardware design example that supports compilation and hardware testing. When you generate the design example, the Intel Quartus® Prime IP parameter editor automatically creates the files necessary to simulate, compile, and test the design in hardware.

In addition, you can download the compiled hardware design to the Intel device-specific development kit for interoperative testing. The Intel FPGA IP also includes a compilation-only example project that you can use to quickly estimate IP core area and timing.

The Low Latency E-Tile 40G Ethernet Intel FPGA IP supports design example generation with a wide range of parameters. However, the design examples do not cover all possible parameterizations of the Low Latency E-Tile 40G Ethernet Intel FPGA IP Core.

Figure 1. Development Steps for the Design Example



Related Information

- Low Latency E-Tile 40G Ethernet Intel FPGA IP User Guide
  For detailed information on Low Latency E-Tile 40G Ethernet IP.
- Low Latency E-Tile 40G Ethernet Intel FPGA IP Release Notes
  The IP Release Notes list IP changes in a particular release.

1.1. Generating the Design Example

Figure 2. Procedure

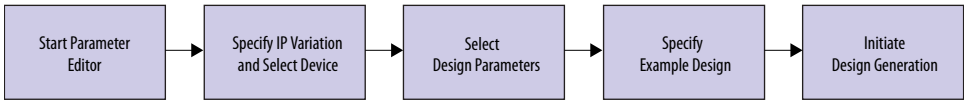
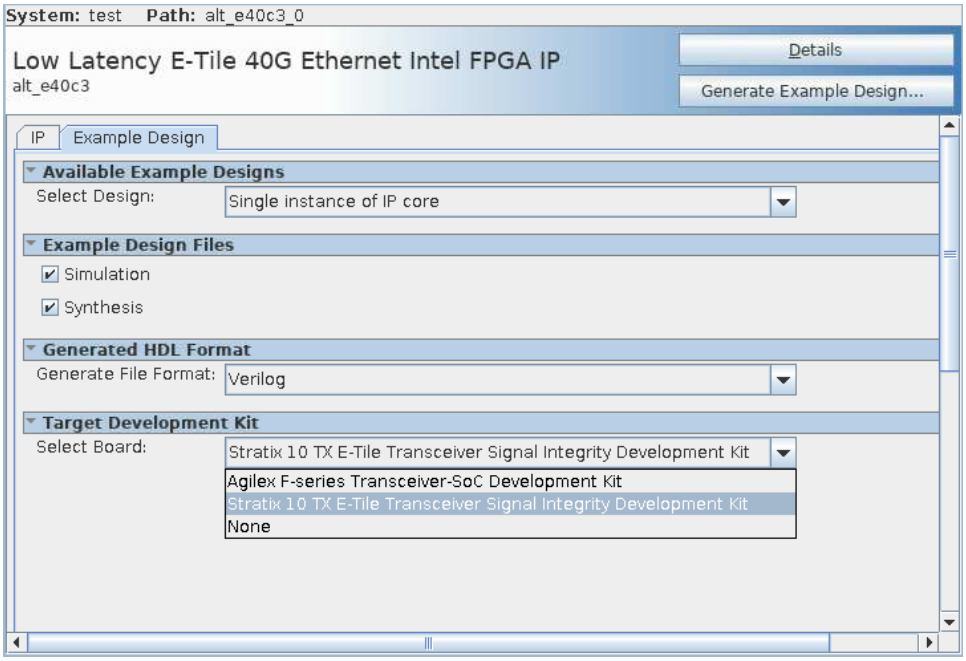


Figure 3. Example Design Tab in the Low Latency E-Tile 40G Ethernet Parameter Editor

Select **Stratix 10 TX E-Tile Transceiver Signal Integrity Development Kit** to generate design example for Intel Stratix® 10 devices. Select **Agilex F-series Transceiver-SoC Development Kit** to generate design example for Intel Agilex™ devices.



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 Intel Quartus Prime project, or **File ➤ Open Project** to open an existing Intel Quartus Prime software project. The wizard prompts you to specify a device family and device.

   Note: The 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).

2. In the IP Catalog, locate and select **Low Latency E-Tile 40G Ethernet Intel FPGA IP**. The **New IP Variation** window appears.
3. Specify a top-level name for your custom IP variation. The Intel Quartus Prime IP parameter editor saves the IP variation settings in a file named <*your_ip*>.ip.
4. Click **OK**. The IP parameter editor appears.
5. On the **IP** tab, specify the parameters for your IP core variation.

   Note: The Low Latency E-Tile 40G Ethernet Intel FPGA IP design example does not simulate correctly and does not function correctly if you specify any of the following parameters:

   - **Enable preamble pass-through** turned on
   - **Ready latency** set to the value of 3
   - **Enable TX CRC insertion** turned off

6. On the **Example Design** tab, under **Example Design Files**, enable the **Simulation** option to generate the testbench, and select the **Synthesis** option to generate the compilation-only and hardware design examples.

   Note: On the **Example Design** tab, under **Generated HDL Format**, only Verilog HDL is available. This IP core does not support VHDL.

7. Under **Target Development Kit** select the **Stratix 10 TX E-Tile Transceiver Signal Integrity Development Kit** or the **Agilex F-series Transceiver-SoC Development Kit**.

   Note: The development kit that you select overwrites the device selection in Step 1.

   - Intel Stratix 10 E-tile target device is 1SG280LU3F50E3VGS1.
   - Intel Agilex E-tile device target is AGFB014R24A2E2VR0.

8. Click the **Generate Example Design** button. The **Select Example Design Directory** window appears.
9. If you want to modify the design example directory path or name from the defaults displayed (`alt_e40c3_0_example_design`), browse to the new path and type the new design example directory name (<*design_example_dir*>).
10. Click **OK**.

Related Information

- IP Core Parameters
  Provides more information about customizing your IP core.
- Intel Stratix 10 E-Tile TX Signal Integrity Development Kit
- Intel Agilex F-Series FPGA Development Kit

1.1.1. Design Example Parameters

Table 1. Parameters in the Example Design Tab

Parameter	Description
Select Design	Available example designs for the IP parameter settings. When you select a design from the Preset library, this field shows the selected design.
Example Design Files	The files to generate for the different development phase. • **Simulation**—generates the necessary files for simulating the example design. • **Synthesis**—generates the synthesis files. Use these files to compile the design in the Intel Quartus Prime Pro Edition software for hardware testing and perform static timing analysis.
Generate File Format	The format of the RTL files for simulation—Verilog or VHDL.
Select Board	Supported hardware for design implementation. When you select an Intel development board, the *Target Device* is the one that matches the device on the Development Kit. If this menu is not available, there is no supported board for the options that you select. **Agilex F-series Transceiver-SoC Development Kit**: This option allows you to test the design example on the selected Intel FPGA IP development kit. This option automatically selects the **Target Device** of AGFB014R24A2E2VR0. If your board revision has a different device grade, you can change the target device.

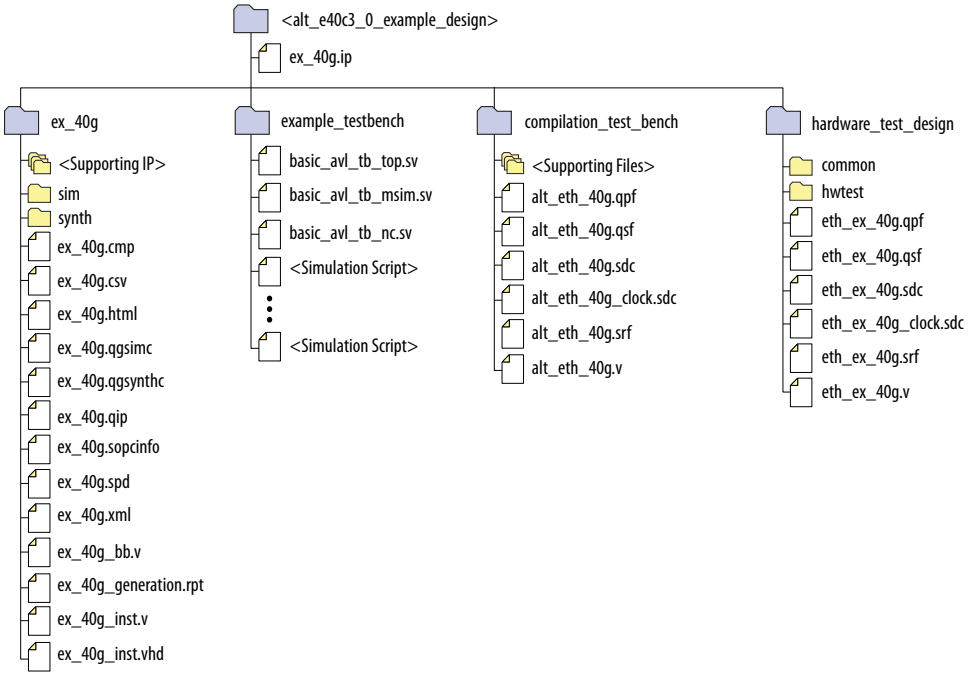
continued...

Parameter	Description
	Stratix 10 TX E-Tile Transceiver Signal Integrity Development Kit: This option allows you to test the design example on the selected Intel FPGA IP development kit. This option automatically selects the *Target Device* of 1ST280EY2F55E2VG. If your board revision has a different device grade, you can change the target device. **None**: This option excludes the hardware aspects for the design example.

1.2. Directory Structure

The Low Latency E-Tile 40G Ethernet IP core design example file directories contain the following generated files for the design example.

Figure 4. Directory Structure for the Generated Design Example



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

Table 2. Directory and File Descriptions

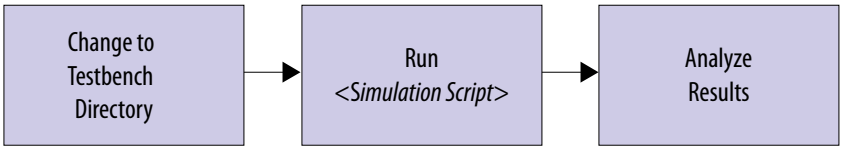
File Names	Description
eth_ex_40g.qpf	Intel Quartus Prime project file.
eth_ex_40g.qsf	Intel Quartus Prime project settings file.

continued...

File Names	Description
`eth_ex_40g.sdc`	Synopsys* Design Constraints file. You can copy and modify this file for your own Low Latency E-Tile 40G Ethernet Intel FPGA IP design.
`eth_ex_40g.srf`	Intel Quartus Prime project message suppression rule file.
`eth_ex_40g.v`	Top-level Verilog HDL design example file.
`eth_ex_40g_clock.sdc`	Synopsys Design Constraints file for clocks.
`common/`	Hardware design example support files.
`hwtest/main.tcl`	Main file for accessing System Console.

1.3. Simulating the Design Example Testbench

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 3. Instructions to Simulate the Testbench

Simulator	Instructions
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-AE and ModelSim-ASE simulators cannot simulate this IP core. You must use another supported ModelSim simulator such as ModelSim SE.
VCS*	In the command line, type `sh run_vcs.sh`
VCS MX	In the command line, type `sh run_vcsmx.sh`. Use this script when the design contains Verilog HDL and System Verilog with VHDL.
NCSim	In the command line, type `sh run_ncsim.sh`
Xcelium*	In the command line, type `sh run_xcelium.sh`

A successful simulation ends with the following message:

```
Simulation Passed.
```

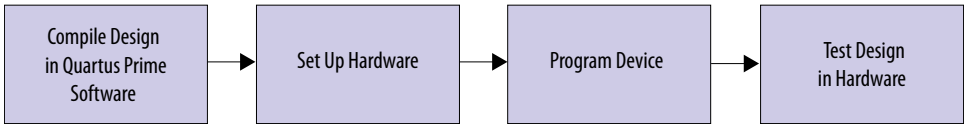
or

```
Testbench complete.
```

After successful completion, you can analyze the results.

1.4. Compiling and Configuring the Design Example in Hardware

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



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 device:
   a. Select **Tools ➤ Programmer**.
   b. In the Programmer, click **Hardware Setup**.
   c. Select a programming device.
   d. Select and add the Intel TX board to your Intel Quartus Prime Pro Edition session.
   e. Ensure that **Mode** is set to **JTAG**.
   f. Select the Intel 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. Turn on **Program/Configure** option for the `.sof`.
   i. Click **Start**.

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 **Stratix 10 TX E-Tile Transceiver Signal Integrity Development Kit** as your target device, the Low Latency E-Tile 40G Ethernet Intel FPGA IP core generates a hardware example design for target device `1ST280EY2F55E2VG`.

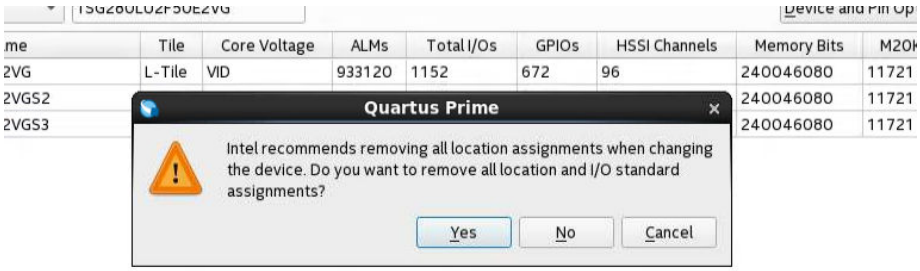
If you have selected **Agilex F-series Transceiver-SoC Development Kit** as your target device, the Low Latency E-Tile 40G Ethernet Intel FPGA IP core generates a hardware example design for target device `AGFB014R24A2E2VR0`.

The specified target device may differ from the device on your development kit.

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_40g.qpf`.
2. On the **Assignments** menu, click **Device**. The **Device** dialog box appears.
3. In the **Device** dialog box, select an E-tile based target device table that matches the device part number on your development kit. Refer to the 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.

Figure 5. Intel Quartus Prime Prompt for Device Selection



5. Perform full compilation of your design.

You can now test the design on your hardware.

Related Information

- Intel Stratix 10 E-Tile TX Signal Integrity Development Kit
- Intel Agilex F-Series FPGA Development Kit

1.6. Testing the Low Latency E-Tile 40G Ethernet Intel FPGA IP Design in Hardware

After you compile the Low Latency E-Tile 40G Ethernet Intel FPGA IP core design example and configure it on your Intel 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.

Additional design example commands are available to program the IP core:

- `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.
- `sys_reset_digital_analog`: System reset.
- `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*>.

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. Design Example Description

The E-tile based 40G Ethernet design example demonstrates the functions of the Low Latency E-Tile 40G Ethernet Intel FPGA IP core, with E-tile based transceiver interface compliant with the IEEE 802.3ba standard CAUI-4 specification. You can generate the design from the **Example Design** tab in the Low Latency E-Tile 40G Ethernet Intel FPGA IP 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 Low Latency E-Tile 40G Ethernet Intel FPGA IP design in simulation and in hardware.

2.1. Features

- Supports 40G Ethernet MAC/PCS IP core for E-tile transceiver using Intel Stratix 10 or Intel Agilex device.
- Supports preamble pass-through and link training.
- Generates design example with MAC stats counters feature.
- Provides testbench and simulation script.

2.2. Hardware and Software Requirements

To test the example design, use the following hardware and software:

- Intel Quartus Prime Pro Edition software
- System Console
- ModelSim, VCS, VCS MX, NCSim, or Xcelium Simulator
- Intel Stratix 10 TX E-Tile Transceiver Signal Integrity Development Kit or Intel Agilex F-series Transceiver-SoC Development Kit

2.3. Functional Description

This section describes the 40G Ethernet MAC/PCS IP core using the Intel device in E-tile based transceiver.

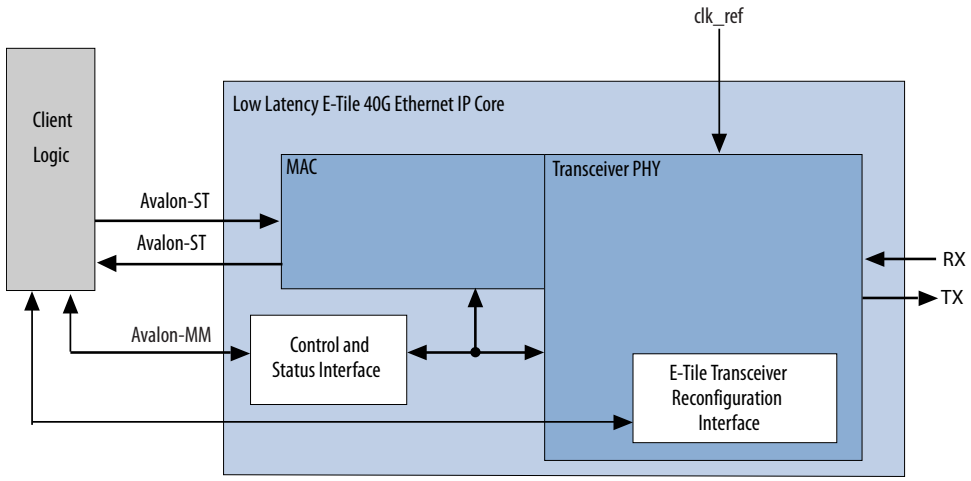
In the transmit direction, the MAC accepts client frames and inserts inter-packet gap (IPG), preamble, the start of frame delimiter (SFD), padding, and CRC bits before passing them to the PHY. The PHY encodes the MAC frame as required for reliable transmission over the media to the remote end.

In the receive direction, the PHY passes frames to the MAC. The MAC accepts frames from the PHY, performs checks, strips out the CRC, preamble, and SFD, and passes the rest of the frame to the client.

2.4. Simulation

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

Figure 6. Low Latency E-Tile 40G Ethernet Design Example Block Diagram



The simulation design example top-level test file is `basic_avl_tb_top.sv`. This file provides a clock reference `clk_ref` of 156.25 Mhz to the PHY. It includes a task to send and receive 10 packets.

Table 4. Low Latency E-Tile 40G Ethernet Core Testbench File Descriptions

File Names	Description
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.
`basic_avl_tb_top_nc.sv`	Top-level testbench file compatible with the NCSim simulator.
`basic_avl_tb_top_msim.sv`	Top-level testbench file compatible with the ModelSim simulator.
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.

continued...

File Names	Description
run_vcsmx.sh	The Synopsys VCS MX script (combined Verilog HDL and System Verilog with VHDL) to run the testbench.
run_ncsim.sh	The Cadence NCSim script to run the testbench.
run_xcelium.sh	The Cadence Xcelium script to run the testbench.

The successful test run displays output confirming the following behavior:

1. Waiting for RX clock to settle
2. Printing PHY status
3. Sending 10 packets
4. Receiving 10 packets
5. Displaying "Testbench complete."

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

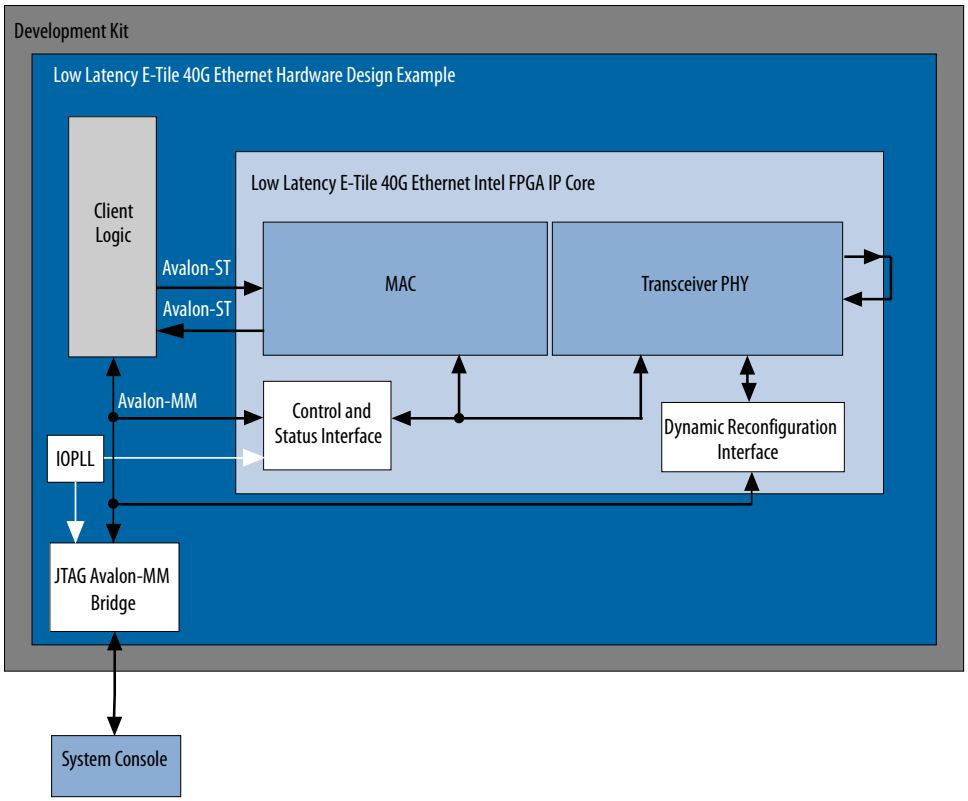
Related Information

Simulating the Design Example Testbench on page 7

2.5. 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.

Figure 7. Low Latency E-Tile 40G Ethernet IP Hardware Design Example High Level Block Diagram



The Low Latency E-Tile 40G Ethernet hardware design example includes the following components:

- Low Latency E-Tile 40G Ethernet Intel FPGA IP core.
- Client logic that coordinates the programming of the IP core, and packet generation and checking.
- IOPLL to generate a 100 MHz clock from a 50 MHz input clock to the hardware design example.
- JTAG controller that communicates with the Intel System Console. You communicate with the client logic through the System Console.

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

Related Information

- Testing the Low Latency E-Tile 40G Ethernet Intel FPGA IP Design in Hardware on page 9
- Analyzing and Debugging Designs with System Console

2.5.1. Internal Loopback Test

Run these steps to perform the internal loopback test:

1. Reset the system.

```
sys_reset_digital_analog
```

2. Display the clock frequency and PHY status.

```
chkphy_status
```

3. Turn on the internal loopback test.

```
loop_on
```

4. Display the clock frequency and PHY status. The `rx_clk` is set to 312.5 MHz and `rx_pcs_ready` is set to 1.

```
chkphy_status
```

5. Start the packet generator.

```
start_pkt_gen
```

6. Stop the packet generator.

```
stop_pkt_gen
```

7. Review the number of transmitted and received packets.

```
chkmac_stats
```

8. Tun off the internal loopback test.

```
loop_off
```

2.5.2. External Loopback Test

Run these steps to perform the external loopback test:

1. Reset the system.

```
sys_reset_digital_analog
```

2. Display the clock frequency and PHY status. The `rx_clk` is set to 312.5 MHz and `rx_pcs_ready` is set to 1.

```
chkphy_status
```

3. Start the packet generator.

```
start_pkt_gen
```

4. Stop the packet generator.

```
stop_pkt_gen
```

5. Review the number of transmitted and received packets.

```
chkmac_stats
```

2.6. Low Latency E-Tile 40G Ethernet Design Example Registers

Table 5. Low Latency E-Tile 40G Ethernet 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 Type
0x300-0x3FF	PHY registers
0x400-0x4FF	TX MAC registers
0x500-0x5FF	RX MAC registers
0x800-0x8FF	Statistics Counter registers - TX direction
0x900-0x9FF	Statistics Counter registers - RX direction
0x1000-1016	Packet Client registers

Table 6. Packet Client Registers

You can customize the Low Latency E-Tile 40G Ethernet hardware design example by programming the packet client registers.

Addr	Name	Bit	Description	HW Reset Value	Access
0x1008	`Packet Size Configure`	[29:0]	Specify the transmit packet size in bytes. These bits have dependencies to PKT_GEN_TX_CTRL register. • Bit [29:16]: Specify the upper limit of the packet size in bytes. This is only applicable to incremental mode. • Bit [13:0]: — For fixed mode, these bits specify the transmit packet size in bytes. — For incremental mode, these bits specify the incremental bytes for a packet.	0x25800040	RW
0x1009	`Packet Number Control`	[31:0]	Specify the number of packets to transmit from the packet generator.	0xA	RW
0x1010	`PKT_GEN_TX_CTRL`	[7:0]	• Bit [0]: Reserved. • Bit [1]: Packet generator disable bit. Set this bit to the value of 1 to turn off the packet generator, and reset it to the value of 0 to turn on the packet generator. • Bit [2]: Reserved. • Bit [3]: Has the value of 1 if the IP core is in MAC loopback mode; has the value of 0 if the packet client uses the packet generator.	0x6	RW

continued...

Addr	Name	Bit	Description	HW Reset Value	Access
			• Bit [5:4]: — 00: Random mode — 01: Fixed mode — 10: Incremental mode • Bit [6]: Set this bit to 1 to use 0x1009 register to turn off packet generator based on a fixed number of packets to transmit. Otherwise, bit [1] of PKT_GEN_TX_CTRL register is used to turn off the packet generator. • Bit [7]: — 1: For transmission without gap in between packets. — 0: For transmission with random gap in between packets.		
0x1011	Destination address lower 32 bits	[31:0]	Destination address (lower 32 bits)	0x56780ADD	RW
0x1012	Destination address upper 16 bits	[15:0]	Destination address (upper 16 bits)	0x1234	RW
0x1013	Source address lower 32 bits	[31:0]	Source address (lower 32 bits)	0x43210ADD	RW
0x1014	Source address upper 16 bits	[15:0]	Source address (upper 16 bits)	0x8765	RW
0x1016	PKT_CL_LOOPBACK_RESET	[0]	MAC loopback reset. Set to the value of 1 to reset the design example MAC loopback.	1'b0	RW

Related Information

Low Latency E-Tile 40G Ethernet Control and Status Register Descriptions
Describes the Low Latency E-Tile 40G Ethernet IP core registers.

2.7. Design Example Interface Signals

The Low Latency E-Tile 40G Ethernet testbench is self-contained and does not require you to drive any input signals.

Table 7. Low Latency E-Tile 40G Ethernet Hardware Design Example Interface Signals

Signal	Direction	Comments
clk50	Input	This clock is driven by the board oscillator. • Drive at 50 MHz on Intel Stratix 10 board. • Drive at 100 MHz on Intel Agilex board. The hardware design example routes this clock to the input of an IOPLL on the device and configures the IOPLL to drive a 100 MHz clock internally.
clk_ref	Input	Drive at 156.25 MHz.

continued...

Signal	Direction	Comments
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. The hardware design example connects these bits to drive LEDs on the target board. Individual bits reflect the following signal values and clock behavior: • [0]: Main reset signal to IP core • [1]: Divided version of clk_ref • [2]: Divided version of clk50 • [3]: Divided version of 100 MHz status clock • [4]: tx_lanes_stable • [5]: rx_block_lock • [6]: rx_am_lock • [7]: rx_pcs_ready

Related Information

Interfaces and Signal Descriptions

Provides detailed descriptions of the Low Latency E-Tile 40G Ethernet IP core signals and the interfaces to which they belong.

A. Low Latency E-Tile 40G Ethernet Intel FPGA IP Archives

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

Intel Quartus Prime Version	IP Core Version	User Guide
20.1	19.1.0	Low Latency E-Tile 40G Ethernet Design Example User Guide

4. Document Revision History for Low Latency E-tile 40G Ethernet Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2020.06.22	20.2	20.0.0	Added device support for Intel Agilex devices.
2020.04.13	20.1	19.1.0	Initial Release.