F-Tile CPRI PHY Intel® FPGA IP Design Example User Guide

Updated for Intel® Quartus® Prime Design Suite: **21.2**

IP Version: **2.0.0**

Subscribe

Send Feedback

UG-20328 | 2021.06.21

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

Contents

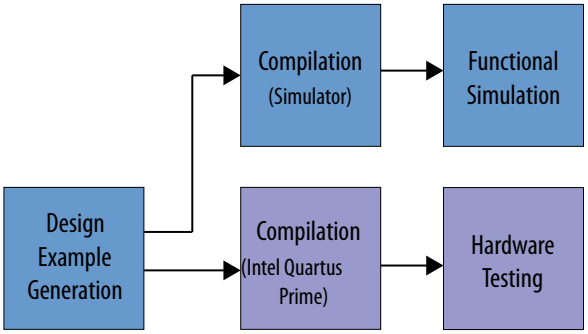
1. Quick Start Guide

The F-Tile CPRI PHY Intel® FPGA IP core provides a simulation testbench and 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.

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

The F-Tile CPRI PHY Intel FPGA IP core provides the capability of generating design examples for all supported combinations of number of CPRI channels and CPRI line bit rates. The testbench and design example support numerous parameter combinations of the F-Tile CPRI PHY Intel FPGA IP core.

Figure 1. Development Steps for the Design Example



Related Information

- F-Tile CPRI PHY Intel FPGA IP User Guide
  For detailed information on E-tile CPRI PHY IP.
- F-Tile CPRI PHY Intel FPGA IP Release Notes
  The IP Release Notes list IP changes in a particular release.

1.1. Hardware and Software Requirements

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

- Intel Quartus® Prime Pro Edition software
- System console
- VCS* or ModelSim* SE simulator

1.2. Generating the Design

Figure 2. Procedure

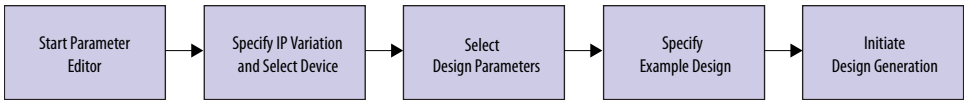
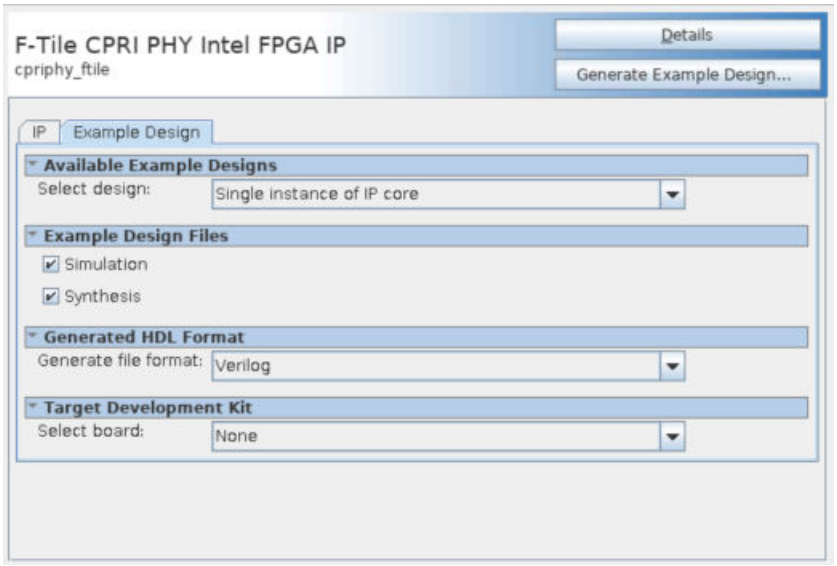


Figure 3. Example Design Tab in IP Parameter Editor



To create an Intel Quartus Prime Pro Edition project:

1. 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.
2. Specify the device family **Agilex** (I-series) and select a device that meets all of these requirements:
   - Transceiver tile is F-tile
   - Transceiver speed grade is -1 or -2
   - Core speed grade is -1 or -2
3. Click **Finish**.

Follow these steps to generate the F-Tile CPRI PHY Intel FPGA IP hardware design example and testbench:

1. In the IP Catalog, locate and select **F-Tile CPRI PHY Intel FPGA IP**. The **New IP Variation** window appears.
2. 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.
3. Click **OK**. The parameter editor appears.
4. On the **IP** tab, specify the parameters for your IP core variation.

5. On the **Example Design** tab, under **Example Design Files**, select the **Simulation** option to generate the testbench and the compilation-only project. Select the **Synthesis** option to generate the hardware design example. You must select at least one of the **Simulation** and **Synthesis** options to generate the design example.
6. On the **Example Design** tab, under **Generated HDL Format**, select **Verilog** HDL or **VHDL**. If you select **VHDL**, you must simulate the testbench with a mixed-language simulator. The device under test in the `ex_<datarate>` directory is a VHDL model, but the main testbench file is a System Verilog file.
7. Click the **Generate Example Design** button. The **Select Example Design Directory** window appears.
8. If you want to modify the design example directory path or name from the defaults displayed (`cpriphy_ftile_0_example_design`), browse to the new path and type the new design example directory name (*<design_example_dir>*).

1.3. Directory Structure

The F-Tile CPRI PHY Intel FPGA IP core design example file directories contain the following generated files for the design example.

Figure 4. Directory Structure of the Generated Example Design

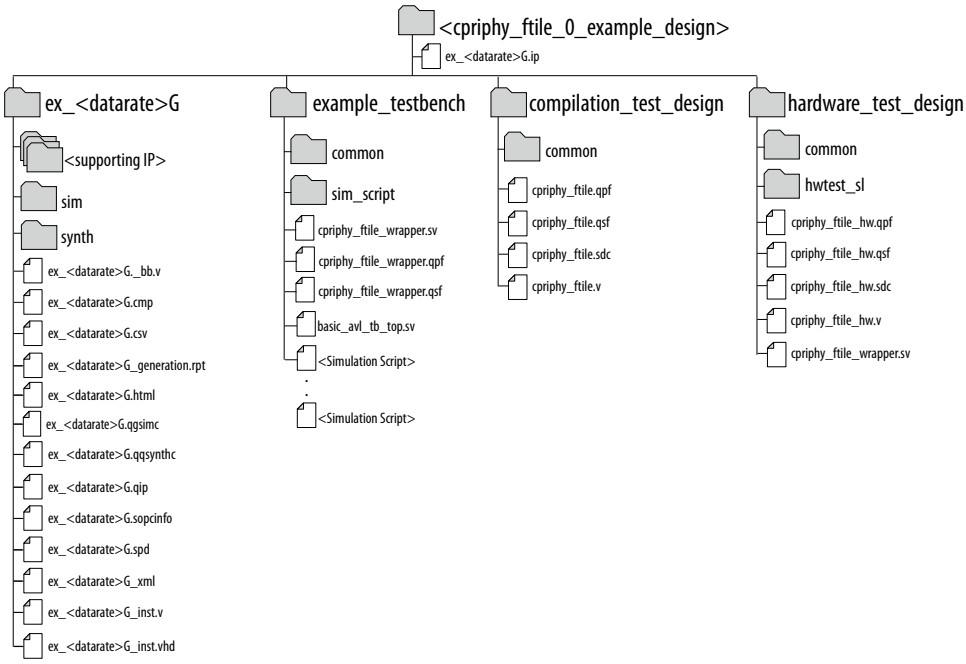


Table 1. Testbench File Descriptions

File Names	Description
Key Testbench and Simulation Files	
`<design_example_dir>/example_testbench/basic_avl_tb_top.sv`	Top-level testbench file. The testbench instantiates the DUT wrapper and runs Verilog HDL tasks to generate and accept packets.

continued...

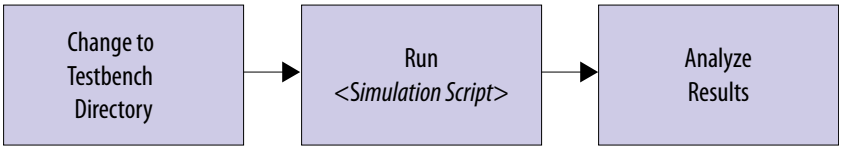
File Names	Description
<design_example_dir>/example_testbench/cpriphy_ftile_wrapper.sv	DUT wrapper that instantiates DUT and other testbench components.
Testbench Scripts	
<design_example_dir>/example_testbench/run_vsim.do	The Mentor Graphics ModelSim* script to run the testbench.
<design_example_dir>/example_testbench/run_vcs.sh	The Synopsys VCS* script to run the testbench.

Table 2. Hardware Design Example File Descriptions

File Names	Descriptions
<design_example_dir>/hardware_test_design/cpriphy_ftile_hw.qpf	Intel Quartus Prime project file.
<design_example_dir>/hardware_test_design/cpriphy_ftile_hw.qsf	Intel Quartus Prime project setting file.
<design_example_dir>/hardware_test_design/cpriphy_ftile_hw.sdc	Synopsys Design Constraints files. You can copy and modify these files for your own Intel Agilex™ design.
<design_example_dir>/hardware_test_design/cpriphy_ftile_hw.v	Top-level Verilog HDL design example file.
<design_example_dir>/hardware_test_design/cpriphy_ftile_wrapper.sv	DUT wrapper that instantiates DUT and other testbench components.
<design_example_dir>/hardware_test_design/hwtest_sl/main_script.tcl	Main file for accessing System Console.

1.4. Simulating the Design Example Testbench

Figure 5. Procedure



Follow these steps to simulate the testbench:

1. At the command prompt, change to the testbench simulation directory *<design_example_dir>*/example_testbench.

```
cd <my_design>/example_testbench
```

2. Run `quartus_tlg` on the generated project file:

```
quartus_tlg cpriphy_ftile_wrapper
```

3. Run `ip-setup-simulation`:

```
ip-setup-simulation --output-directory=./sim_script --use-relative-paths --
quartus-project=cpriphy_ftile_wrapper.qpf
```

4. 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*.
5. Analyze the results. The successful testbench received five hyperframes, and displays "PASSED".

Table 3. Steps to Simulate the Testbench in Synopsys VCS* Simulator

Simulator	Instructions
Synopsys VCS*	In the command line, type: `sh run_vcs.sh`

Mentor Graphics ModelSim* SE	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`

The following sample output illustrates a successful simulation test run for 24.33024 Gbps with 4 CPRI channels:

```
Ref clock is 184.32 MHz
Waiting for TX ready
TX is ready is high at time              75250000
Waiting for RX ready
RX is ready is high at time             120750000
*** configure DL ***
** Address offset = 0x185d, ReadData  = 0x00000a86
** Address offset = 0x3c00a, ReadData  = 0x000c0000
** Address offset = 0x3c00a, WriteData = 0x000c0a86
** Address offset = 0x3c00a, ReadData  = 0x000c0a86
** Address offset = 0x2, WriteData = 0xc0000000
** Address offset = 0x2, WriteData = 0x00000000
** Address offset = 0x185d, ReadData  = 0x000034a6
** Address offset = 0x3c008, ReadData  = 0x000c0000
** Address offset = 0x3c008, WriteData = 0x000c34a6
** Address offset = 0x3c008, ReadData  = 0x000c34a6
** Address offset = 0x2, WriteData = 0xc0000000
** Address offset = 0x2, WriteData = 0x00000000
** Address offset = 0x185d, ReadData  = 0x00003526
** Address offset = 0x3c006, ReadData  = 0x014c0000
** Address offset = 0x3c006, WriteData = 0x014c3526
** Address offset = 0x3c006, ReadData  = 0x014c3526
** Address offset = 0x2, WriteData = 0xc0000000
** Address offset = 0x2, WriteData = 0x00000000
** Address offset = 0x185d, ReadData  = 0x00001be6
** Address offset = 0x3c004, ReadData  = 0x014c0000
** Address offset = 0x3c004, WriteData = 0x014c1be6
** Address offset = 0x3c004, ReadData  = 0x014c1be6
** Address offset = 0x2, WriteData = 0xc0000000
** Address offset = 0x2, WriteData = 0x00000000
*** sending packets in progress, waiting for checker pass ***
*** waiting for measure_valid to assert...
** Address offset = 0x2, ReadData  = 0x00000003
** measure_valid is asserted.
** Address offset = 0x3, ReadData  = 0x00001ba0
** Address offset = 0x4, ReadData  = 0x000082bf
*** waiting for hyperframe sync to assert...
** hyperframe sync is asserted.
*** waiting for round trip measure...
-> 385120ns: Channel 0: Round trip measure done with count 5151
** Channel 0: RX checker has received packets correctly!
** PASSED
*** waiting for measure_valid to assert...
```

```
** Address offset = 0x2, ReadData  = 0x00000003
** measure_valid is asserted.
** Address offset = 0x3, ReadData  = 0x00001b44
** Address offset = 0x4, ReadData  = 0x00008516
*** waiting for hyperframe sync to assert...
** hyperframe sync is asserted.
*** waiting for round trip measure...
-> 385255ns: Channel 1: Round trip measure done with count 5216
** Channel 1: RX checker has received packets correctly!
** PASSED
*** waiting for measure_valid to assert...
** Address offset = 0x2, ReadData  = 0x00000003
** measure_valid is asserted.
** Address offset = 0x3, ReadData  = 0x00001b02
** Address offset = 0x4, ReadData  = 0x0000860c
*** waiting for hyperframe sync to assert...
** hyperframe sync is asserted.
*** waiting for round trip measure...
-> 385385ns: Channel 2: Round trip measure done with count 5240
** Channel 2: RX checker has received packets correctly!
** PASSED
*** waiting for measure_valid to assert...
** Address offset = 0x2, ReadData  = 0x00000003
** measure_valid is asserted.
** Address offset = 0x3, ReadData  = 0x00001a79
** Address offset = 0x4, ReadData  = 0x000083d7
*** waiting for hyperframe sync to assert...
** hyperframe sync is asserted.
*** waiting for round trip measure...
-> 387084ns: Channel 3: Round trip measure done with count 5152
** Channel 3: RX checker has received packets correctly!
** PASSED
**
**************************************
$finish called from file "basic_avl_tb_top.sv", line 352.
$finish at simulation time 387104ns
```

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/cpriphy_ftile.qpf.
3. On the Processing menu, click **Start Compilation**.
4. After successful compilation, reports for timing and for resource utilization are available in your Intel Quartus Prime Pro Edition session.

Related Information

Block-Based Design Flows

1.6. Compiling and Configuring the Design Example in Hardware

To compile the hardware design example and configure it on your Intel Agilex 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/cpriphy_ftile_hw.qpf.
3. Edit the `.qsf` file to assign pins based on your hardware.
4. On the Processing menu, click **Start Compilation**.
5. After successful compilation, a `.sof` file is available in <*design_example_dir*>/hardware_test_design/output_files directory. Follow these steps to program the hardware design example on the Intel Agilex device:
   a. Connect Intel Agilex I-series Transceiver Signal Integrity Development Kit to the host computer.

      Note: The development kit is preprogrammed with the correct clock frequencies by default. You do not need to use the Clock Control application to set the frequencies.

   b. On the **Tools** menu, click **Programmer**.
   c. In the Programmer, click **Hardware Setup**.
   d. Select a programming device.
   e. Ensure that **Mode** is set to **JTAG**.
   f. Select the Intel Agilex 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 Information

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

1.7. Testing the Hardware Design Example

After you compile the F-Tile CPRI PHY Intel FPGA IP core design example and configure it on your Intel Agilex device, you can use the System Console to program the IP core and its 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 Agilex 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_sl`.
3. Type `source main_script.tcl` to open a connection to the JTAG master and start the test.

2. Design Example Description

The design example demonstrates the basic functionality of the F-Tile CPRI PHY Intel FPGA IP core. You can generate the design from the Example Design tab in the F-Tile CPRI PHY 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. You can choose to generate the design example with or without the RS-FEC feature. The RS-FEC feature is available with 10.1376, 12.1651 and 24.33024 Gbps CPRI line bit rates.

Table 4. F-Tile CPRI PHY Intel FPGA IP Core Feature Matrix

The Intel Quartus Prime Pro Edition software version 21.2 supports the following combinations. All other combinations are planned to be supported in the future version of Intel Quartus Prime Pro Edition.

CPRI Line Bit Rate (Gbps)	RS-FEC Support	Reference Clock (MHz)	Deterministic Latency Support
2.4576	No	153.6	Yes
4.9152	No	153.6	Yes
9.8304	No	153.6	Yes
10.1376	With and Without	184.32	Yes
12.1651	With and Without	184.32	Yes
24.33024	With and Without	184.32	Yes

2.1. Features

- Generate the design example with RS-FEC feature
- Basic packet checking capabilities including round trip latency count

2.2. Simulation Design Example

The F-Tile CPRI PHY Intel FPGA IP design example generates a simulation testbench and simulation files that instantiates the F-Tile CPRI PHY Intel FPGA IP core when you select the **Simulation** option.

Figure 6. Block Diagram for 10.1316, 12.1651, and 24.33024 Gbps (with and without RS-FEC) Line Rates

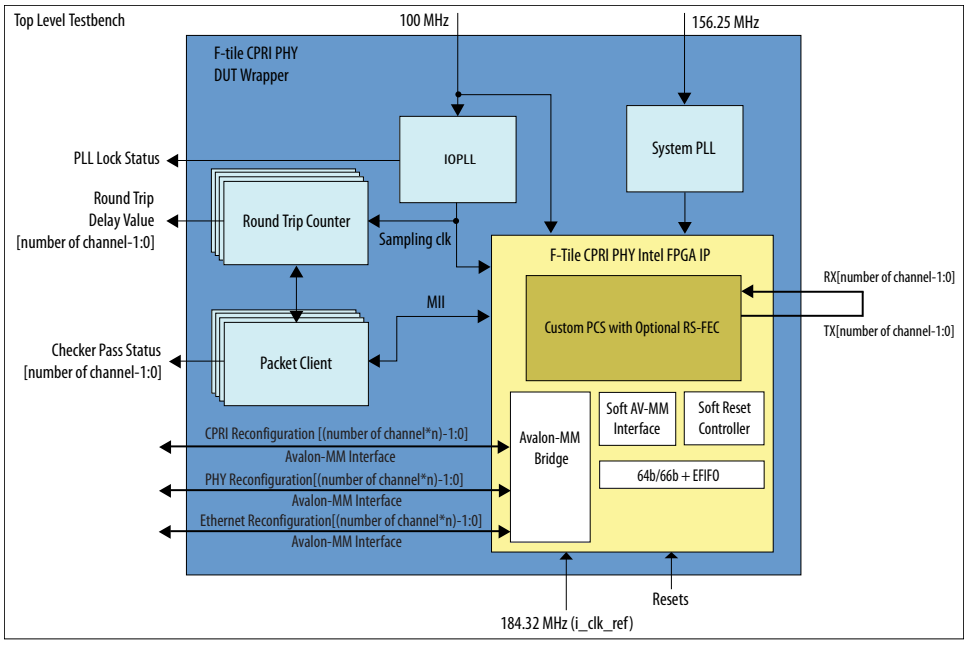
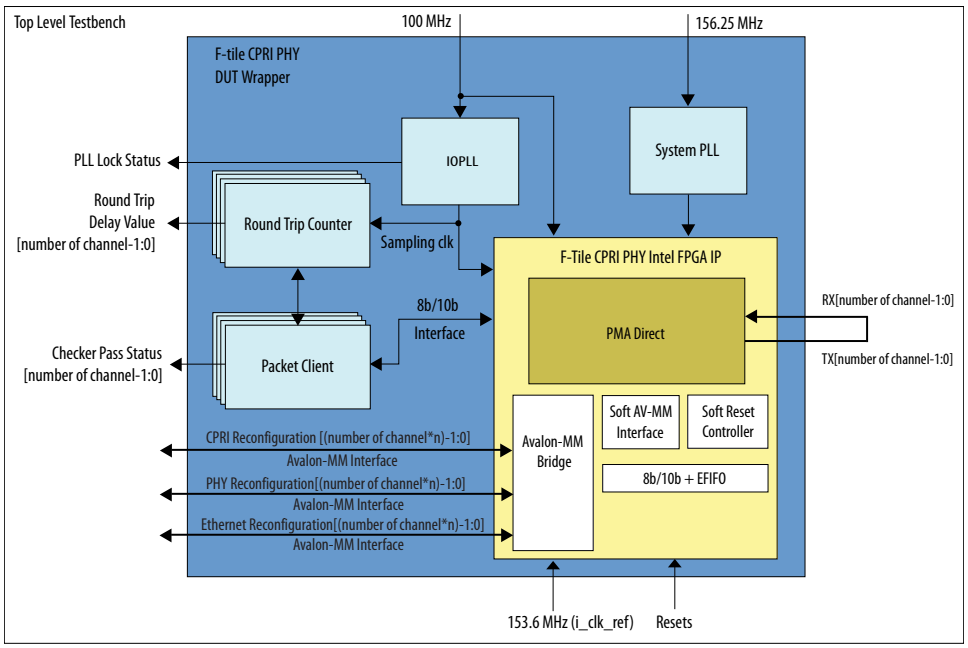


Figure 7. Block Diagram for 2.4576, 4.9152, and 9.8304 Gbps Line Rate



In this design example, the simulation testbench provides basic functionality such as startup and wait for lock, transmit and receive packets.

The successful test run displays output confirming the following behavior:

1. The client logic resets the IP core.
2. The client logic waits for the RX datapath alignment.
3. The client logic transmits hyperframes on the TX MII interface and waits for five hyperframes to be received on RX MII interface. Hyperframes are transmitted and received on MII interface according to the CPRI v7.0 specifications.

   Note: The CPRI designs that target 2.4, 4.9, and 9.8 Gbps line rate use 8b/10b interface and the designs that target 10.1, 12.1 and 24.3 Gbps (with and without RS-FEC) use MII interface.

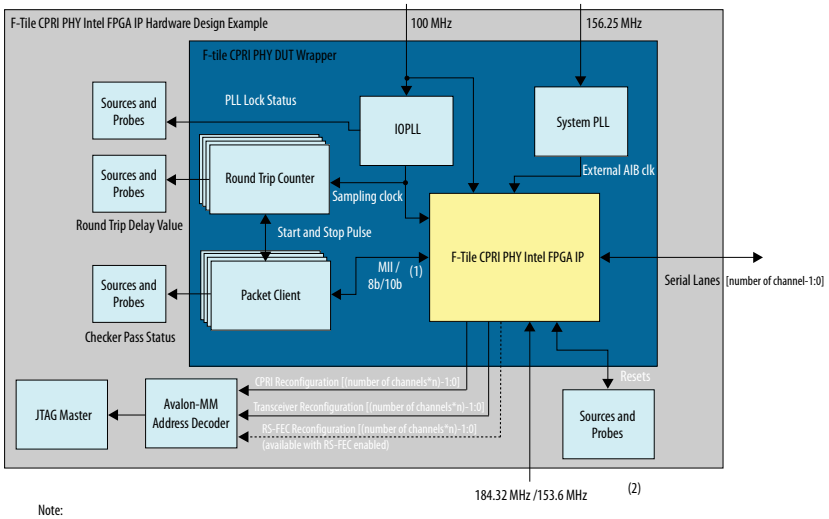
   Note: This design example includes a round trip counter to count the round trip latency from TX to RX.
4. The client logic reads the round trip latency value and checks for the content and correctness of the hyperframes data on the RX MII side once the counter completes the round trip latency count.

Related Information

CPRI Specifications

2.3. Hardware Design Example

Figure 8. Hardware Design Example Block Diagram



Note:
(1) The CPRI designs with 2.4/4.9/9.8 Gbps CPRI line rates use 8b/10b interface and all other CPRI line rates designs use MII interface.
(2) The CPRI designs with 2.4/4.9/9.8 Gbps CPRI line rates need 153.6 MHz transceiver reference clock and all other CPRI line rates need 184.32 MHz.

The F-Tile CPRI PHY Intel FPGA IP core hardware design example includes the following components:

- F-Tile CPRI PHY Intel FPGA IP core.
- Packet client logic block that generates and receives traffic.
- Round trip counter.
- IOPLL to generate sampling clock for deterministic latency logic inside the IP, and round trip counter component at testbench.
- System PLL to generate system clocks for the IP.

- Avalon®-MM address decoder to decode reconfiguration address space for CPRI, Transceiver, and Ethernet modules during reconfiguration accesses.
- Sources and probes for asserting resets and monitoring the clocks and a few status bits.
- JTAG controller that communicates with the System Console. You communicate with the client logic through System Console.

2.4. Interface Signals

Table 5. Design Example Interface Signals

Signal	Direction	Description
ref_clk100MHz	Input	Input clock for CSR access on all the reconfiguration interfaces. Drive at 100 MHz.
i_clk_ref[0]	Input	Reference clock for System PLL. Drive at 156.25 MHz.
i_clk_ref[1]	Input	Transceiver reference clock. Drive at • 153.6 MHz for CPRI line rate 2.4, 4.9, and 9.8 Gbps. • 184.32 MHz for CPRI line rates 10.1,12.1, and 24.3 Gbps with and without RS-FEC.
i_rx_serial[n]	Input	Transceiver PHY input serial data.
o_tx_serial[n]	Output	Transceiver PHY output serial data.

2.5. Design Example Registers

Table 6. Design Example Registers

Channel Number	Word Offset	Register Type
0	0x00000000	CPRI PHY Reconfiguration registers for Channel 0
	0x0C000000	Ethernet Reconfiguration registers for Channel 0
	0x04000000	Transceiver Reconfiguration registers for Channel 0
1(1)	0x01000000	CPRI PHY Reconfiguration registers for Channel 1
	0x0D000000	Ethernet Reconfiguration registers for Channel 1
	0x05000000	Transceiver Reconfiguration registers for Channel 1
2(1)	0x02000000	CPRI PHY Reconfiguration registers for Channel 2
	0x0E000000	Ethernet Reconfiguration registers for Channel 2
	0x06000000	Transceiver Reconfiguration registers for Channel 2
3(1)	0x03000000	CPRI PHY Reconfiguration registers for Channel 3
	0x0F000000	Ethernet Reconfiguration registers for Channel 3
	0x07000000	Transceiver Reconfiguration registers for Channel 3

(1) These registers are reserved if the channel is not used.

3. Document Revision History for F-Tile CPRI PHY Intel FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2021.06.21	21.2	2.0.0	Initial release.