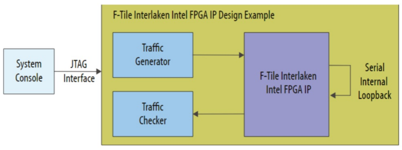
Manuals+ — User Manuals Simplified.



intel F-Tile Interlaken FPGA IPDesign Example User Guide

Home » Intel » intel F-Tile Interlaken FPGA IPDesign Example User Guide

Contents

intel F-Tile Interlaken FPGA IPDesign Example User Guide

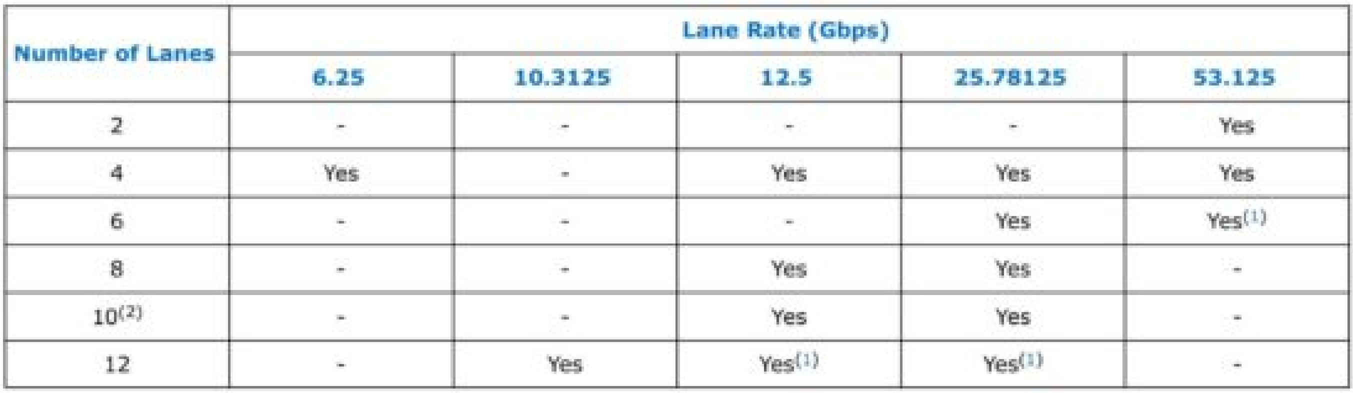
Updated for Intel® Quartus® Prime Design Suite: 21.4
IP Version: 3.1.0

1. Quick Start Guide

The F-Tile Interlaken 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 parameter editor automatically creates the files necessary to simulate, compile, and test the design.

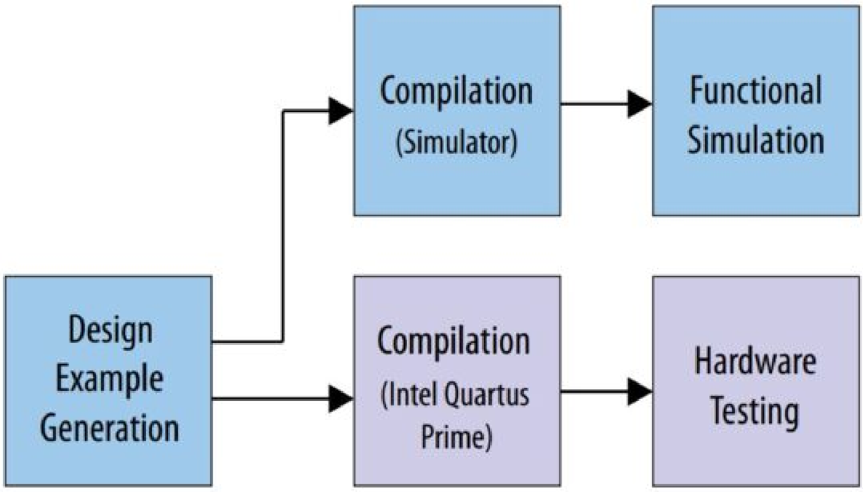
The testbench and design example supports NRZ and PAM4 mode for F-tile devices.
The F-Tile Interlaken Intel FPGA IP core generates design examples for the following supported combinations of number of lanes and data rates.

Table 1. IP Supported Combinations of Number of Lanes and Data Rates
The following combinations are supported in the Intel Quartus® Prime Pro Edition software version 21.4. All other combinations will be supported in a future version of the Intel Quartus Prime Pro Edition.



Number of Lanes	Lane Rate (Gbps)				
	6.25	10.3125	12.5	25.78125	53.125
2	-	-	-	-	Yes
4	Yes	-	Yes	Yes	Yes
6	-	-	-	Yes	Yes(1)
8	-	-	Yes	Yes	-
10(2)	-	-	Yes	Yes	-
12	-	Yes	Yes(1)	Yes(1)	-

Figure 1. Development Steps for the Design Example



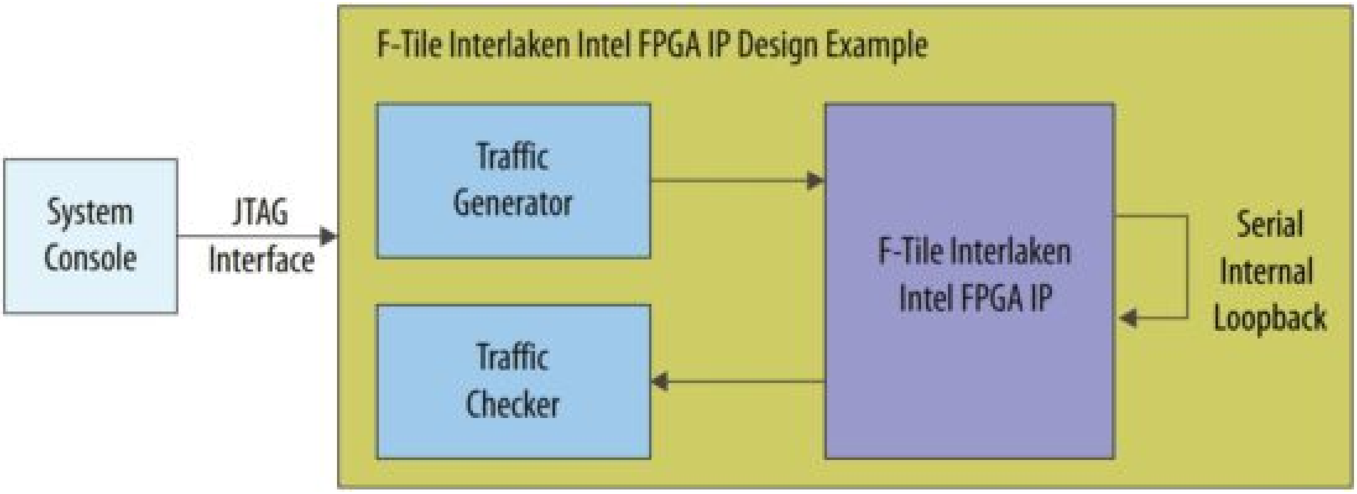
(1) This variant supports the Interlaken Look-aside Mode.
(2) For a 10-lane configuration design, the F-tile requires 12 lanes of TX PMA to enable bonded transceiver clocking for minimizing the channel skew.

*Other names and brands may be claimed as the property of others.

The F-Tile Interlaken Intel FPGA IP core design example supports the following features:

- Internal TX to RX serial loopback mode
- Automatically generates fixed size packets
- Basic packet checking capabilities
- Ability to use System Console to reset the design for re-testing purpose

Figure 2. High-level Block Diagram



Related Information

- F-Tile Interlaken Intel FPGA IP User Guide
- F-Tile Interlaken Intel FPGA IP Release Notes

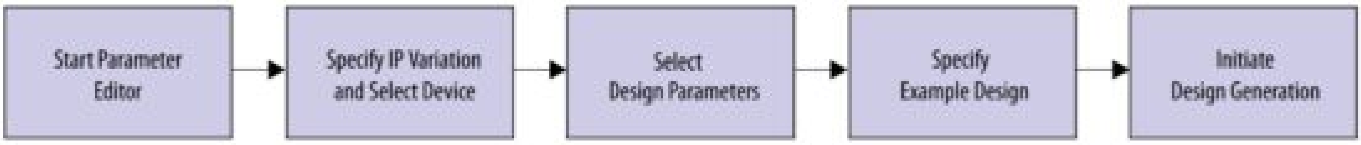
1.1. Hardware and Software Requirements

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

- Intel Quartus Prime Pro Edition software version 21.4
- System console available with the Intel Quartus Prime Pro Edition software
- A supported simulator:
  - — Synopsys* VCS*
  - — Synopsys VCS MX
  - — Siemens* EDA ModelSim* SE or Questa*
  - — Cadence* Xcelium*
- Intel Agilex™ I-Series Transceiver-SoC Development Kit

1.2. Generating the Design

Figure 3. Procedure

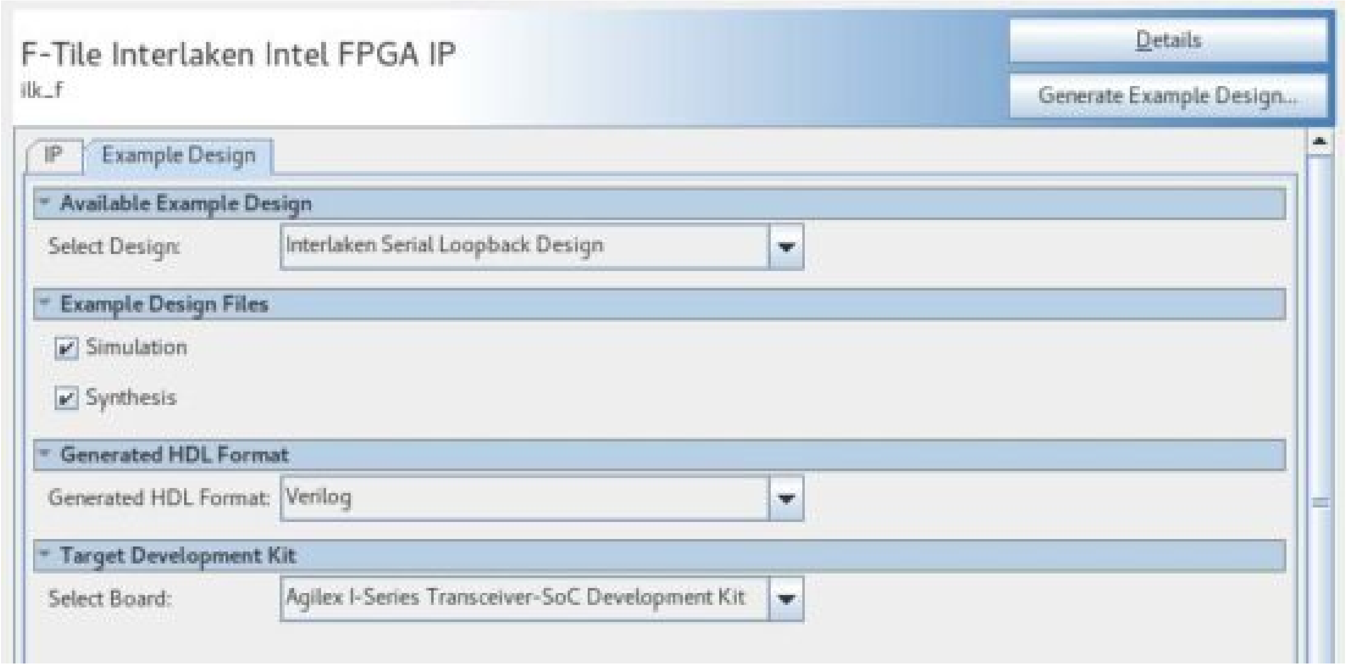


Follow these steps to generate the design example and testbench:

1. In the Intel Quartus Prime Pro Edition software, click File ➤ New Project Wizard to create a new Intel Quartus Prime project, or click 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 and select device with F-Tile for your design.
3. In the IP Catalog, locate and double-click F-Tile Interlaken Intel FPGA IP. The New IP Variant window appears.
4. 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.

5. Click OK. The parameter editor appears.

Figure 4. Example Design Tab



F-Tile Interlaken Intel FPGA IP

ilk_f

Available Example Design

Select Design: Interlaken Serial Loopback Design

Example Design Files

☑ Simulation

☑ Synthesis

Generated HDL Format

Generated HDL Format: Verilog

Target Development Kit

Select Board: Agilex I-Series Transceiver-SoC Development Kit

6. On the IP tab, specify the parameters for your IP core variation.
7. On the Example Design tab, select the Simulation option to generate the testbench. 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.
8. For Generated HDL Format, both Verilog and VHDL option is available.
9. For Target Development Kit, select the Agilex I-Series Transceiver-SOC Development Kit.

Note: When you select the Development Kit option, the pin assignments are set according to the Intel Agilex I-Series Transceiver-SoC Development Kit device part number (AGIB027R31B1E2VR0) and may differ from your selected device. If you intend to test the design on hardware on a different PCB, select No development kit option and make the appropriate pin assignments in the .qsf file

10. Click Generate Example Design. The Select Example Design Directory window appears.
11. If you want to modify the design example directory path or name from the defaults displayed (ilk_f_0_example_design), browse to the new path and type the new design example directory name.
12. Click OK.

Note: In the F-Tile Interlaken Intel FPGA IP design example, a SystemPLL is instantiated automatically, and connected to F-Tile Interlaken Intel FPGA IP core. The SystemPLL hierarchy path in the design example is:

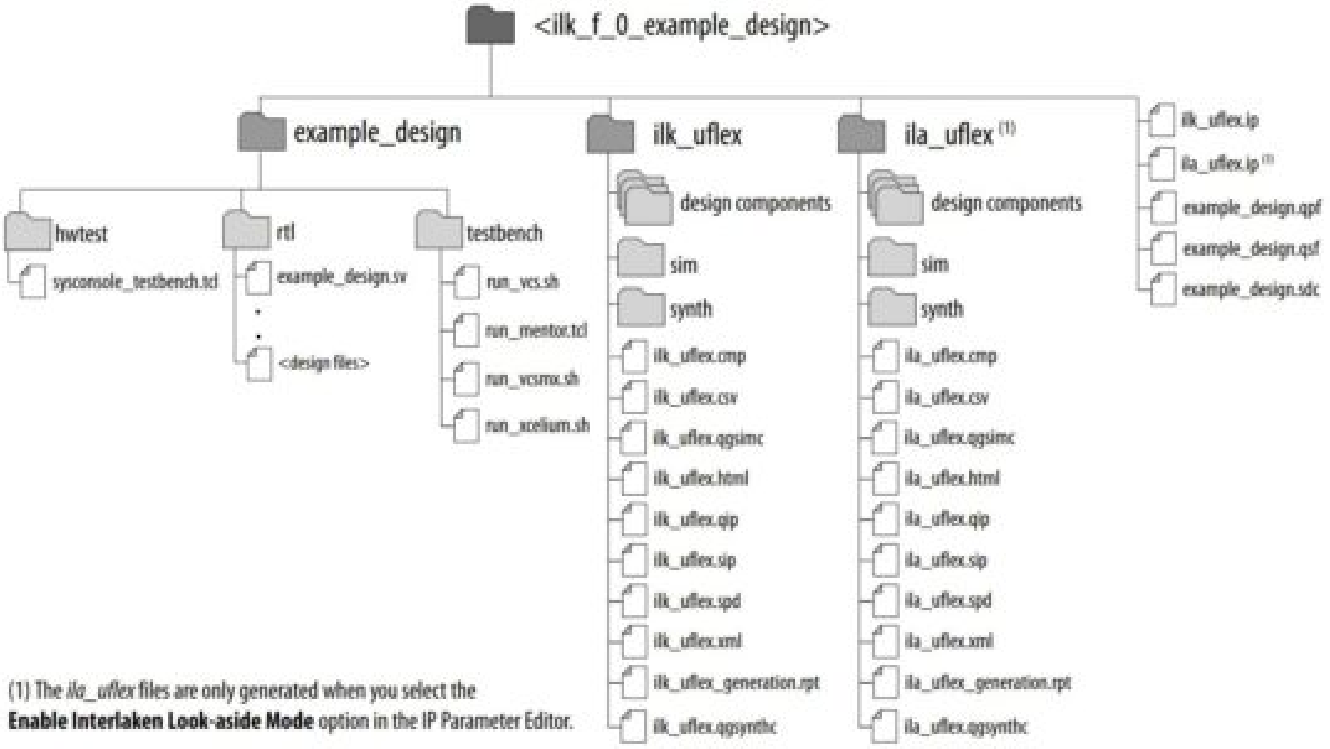
example_design.test_env_inst.test_dut.dut.pll

The SystemPLL in the design example shares the same 156.26 MHz reference clock as the Transceiver.

1.3. Directory Structure

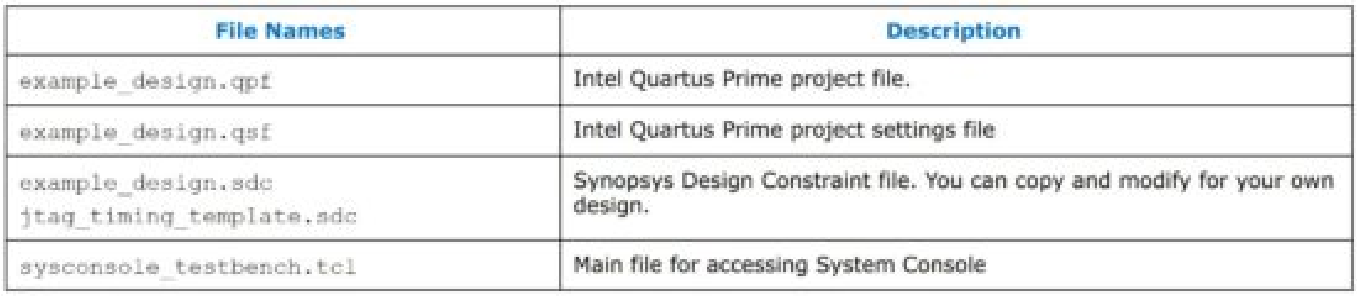
The F-Tile Interlaken Intel FPGA IP core generates the following files for the design example:

Figure 5. Directory Structure



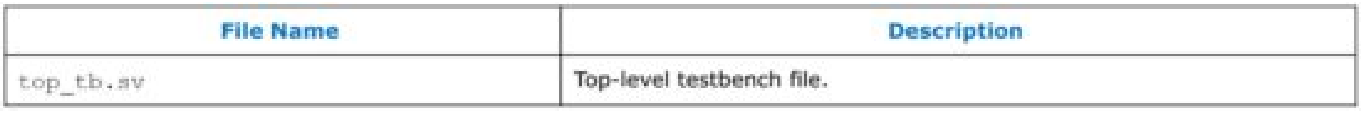
(1) The *ila_uflex* files are only generated when you select the **Enable Interlaken Look-aside Mode** option in the IP Parameter Editor.

Table 2. Hardware Design Example File Descriptions
These files are in the <design_example_installation_dir>/ilk_f_0_example_design directory.



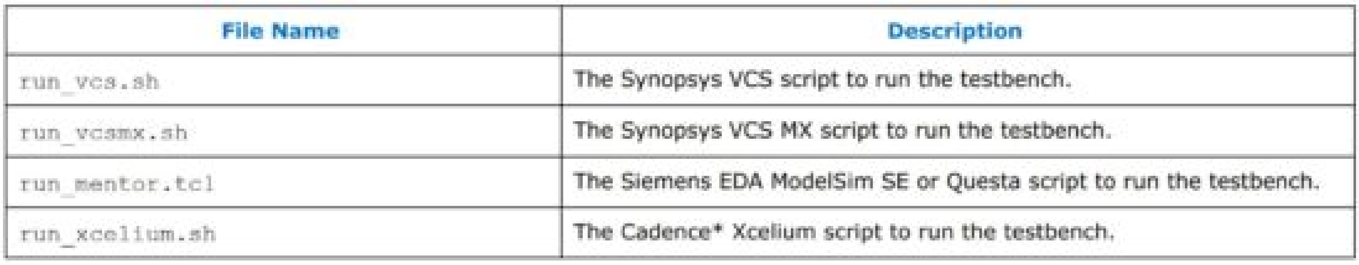
File Names	Description
example_design.qpf	Intel Quartus Prime project file.
example_design.qsf	Intel Quartus Prime project settings file
example_design.sdc jtag_timing_template.sdc	Synopsys Design Constraint file. You can copy and modify for your own design.
sysconsole_testbench.tcl	Main file for accessing System Console

Table 3. Testbench File Description
This file is in the <design_example_installation_dir>/ilk_f_0_example_design/example_design/rtl directory.



File Name	Description
top_tb.sv	Top-level testbench file.

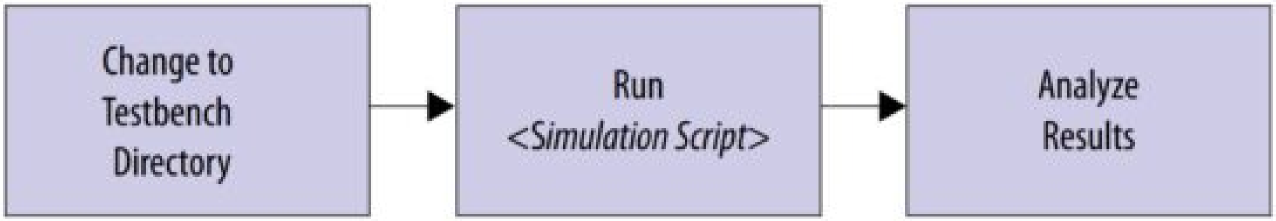
Table 4. Testbench Scripts
These files are in the <design_example_installation_dir>/ilk_f_0_example_design/example_design/testbench directory.



File Name	Description
run_vcs.sh	The Synopsys VCS script to run the testbench.
run_vcsmx.sh	The Synopsys VCS MX script to run the testbench.
run_mentor.tcl	The Siemens EDA ModelSim SE or Questa script to run the testbench.
run_xcelium.sh	The Cadence* Xcelium script to run the testbench.

1.4. Simulating the Design Example Testbench

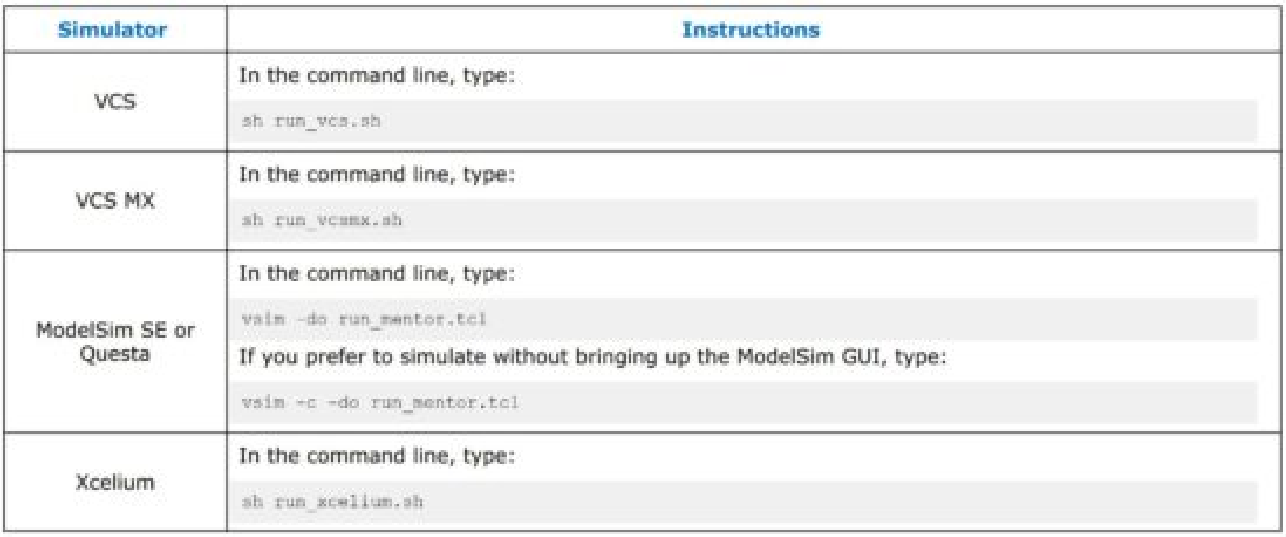
Figure 6. Procedure



Follow these steps to simulate the testbench:

1. At the command prompt, change to the testbench simulation directory. The directory path is <design_example_installation_dir>/example_design/testbench.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator. Your script should check that the SOP and EOP counts match after simulation is complete.

Table 5. Steps to Run Simulation



Simulator	Instructions
VCS	In the command line, type: `sh run_vcs.sh`
VCS MX	In the command line, type: `sh run_vcsmx.sh`

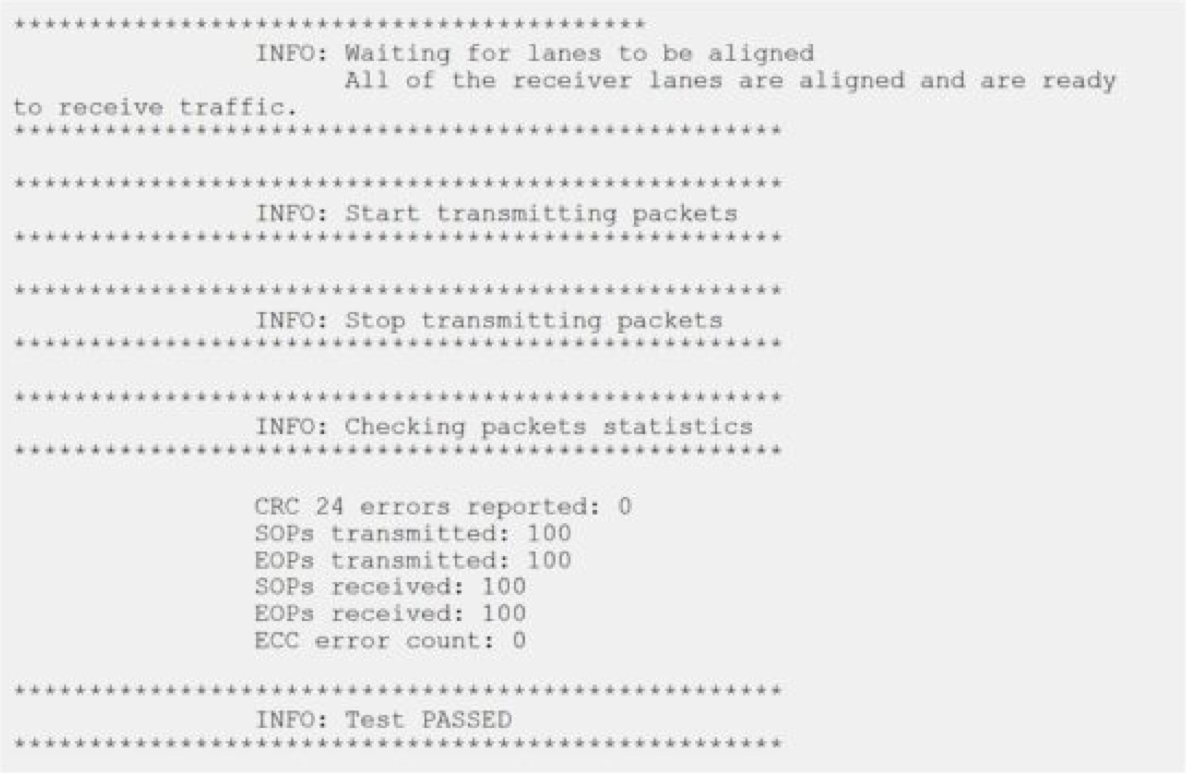
ModelSim SE or Questa	In the command line, type:
	`vsim -do run_mentor.tcl`
	If you prefer to simulate without bringing up the ModelSim GUI, type:
	`vsim -c -do run_mentor.tcl`

Xcelium	In the command line, type: `sh run_xcelium.sh`

3. Analyze the results. A successful simulation sends and receives packets, and displays “Test PASSED”.
The testbench for the design example completes the following tasks:

- Instantiates the F-Tile Interlaken Intel FPGA IP core.
- Prints PHY status.
- Checks metaframe synchronization (SYNC_LOCK) and word (block) boundaries (WORD_LOCK).
- Waits for individual lanes to be locked and aligned.
- Starts transmitting packets.
- Checks packet statistics:
  — CRC24 errors
  — SOPs
  — EOPs

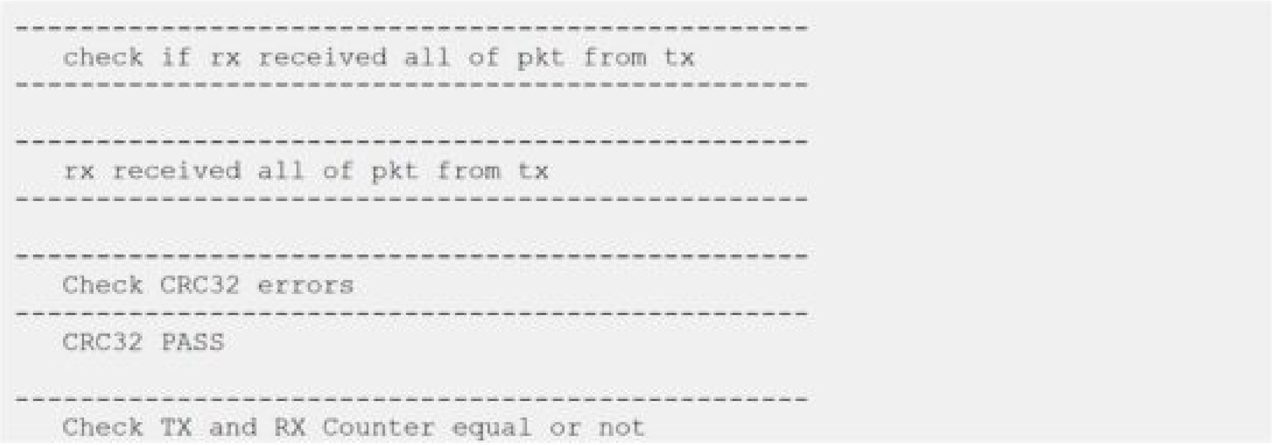
The following sample output illustrates a successful simulation test run:



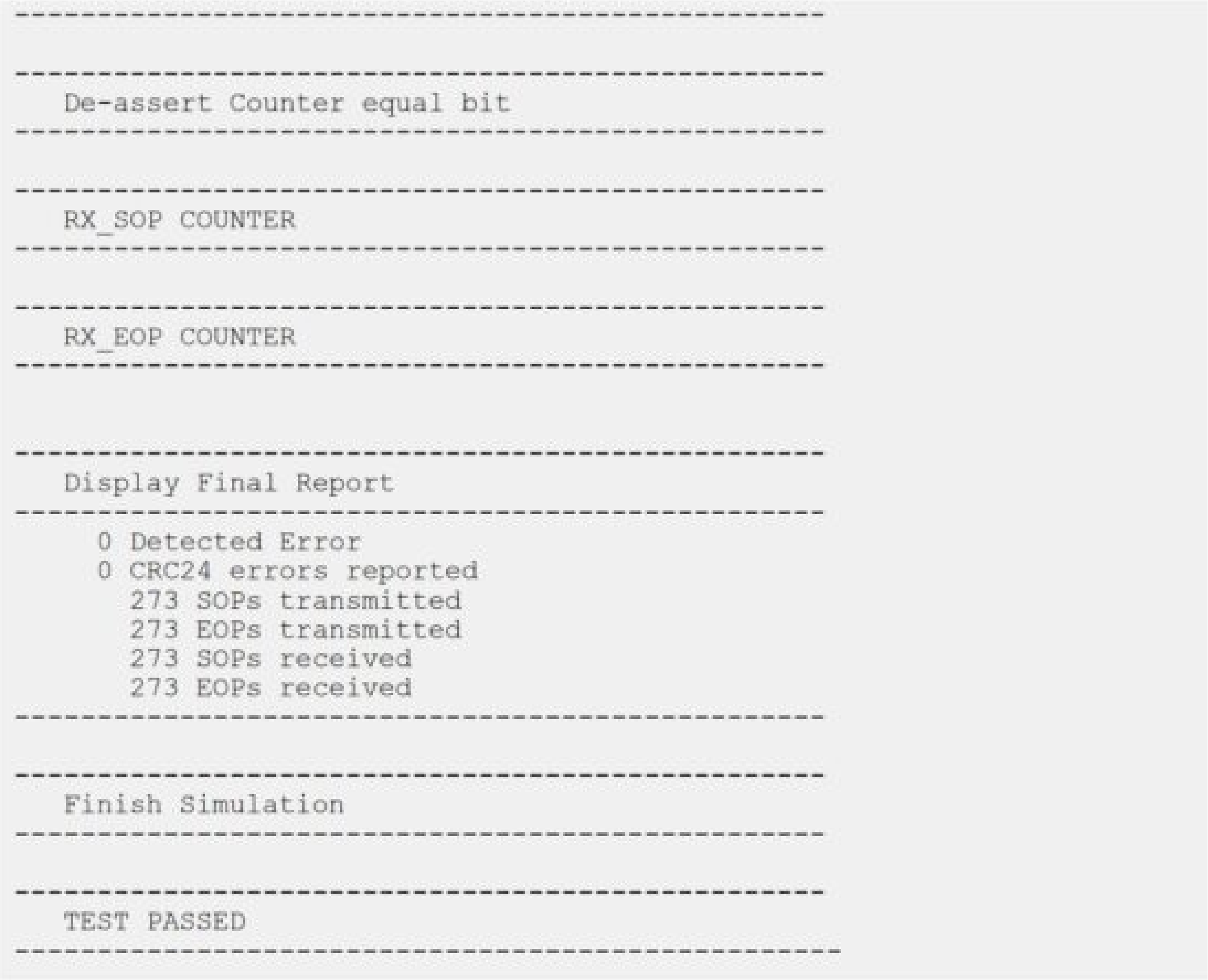
```
******************************************
                INFO: Waiting for lanes to be aligned
                      All of the receiver lanes are aligned and are ready
to receive traffic.
***************************************************

***************************************************
                INFO: Start transmitting packets
***************************************************

***************************************************
                INFO: Stop transmitting packets
***************************************************

***************************************************
                INFO: Checking packets statistics
***************************************************

                CRC 24 errors reported: 0
                SOPs transmitted: 100
                EOPs transmitted: 100
                SOPs received: 100
                EOPs received: 100
                ECC error count: 0

***************************************************
                INFO: Test PASSED
***************************************************
```

Note: The Interlaken design example simulation testbench sends 100 packets and receives 100 packets.

The following sample output illustrates a successful simulation test run for Interlaken Look-aside mode:



```
-------------------------------------------------
   check if rx received all of pkt from tx
-------------------------------------------------

-------------------------------------------------
   rx received all of pkt from tx
-------------------------------------------------

-------------------------------------------------
   Check CRC32 errors
-------------------------------------------------
   CRC32 PASS

-------------------------------------------------
   Check TX and RX Counter equal or not
```



```
-------------------------------------------------

-------------------------------------------------
   De-assert Counter equal bit
-------------------------------------------------

-------------------------------------------------
   RX_SOP COUNTER
-------------------------------------------------

-------------------------------------------------
   RX_EOP COUNTER
-------------------------------------------------


-------------------------------------------------
   Display Final Report
-------------------------------------------------
      0 Detected Error
      0 CRC24 errors reported
        273 SOPs transmitted
        273 EOPs transmitted
        273 SOPs received
        273 EOPs received
-------------------------------------------------

-------------------------------------------------
   Finish Simulation
-------------------------------------------------

-------------------------------------------------
   TEST PASSED
--------------------------------------------------
```

1.5. Compiling and Configuring the Hardware Design Example

1. Ensure the example design generation is complete.
2. In the Intel Quartus Prime Pro Edition software, open the Intel Quartus Prime project <design_example_installation_dir>/example_design.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 example design on the Intel Agilex device with F-tile:

   a. Connect the Development Kit to the host computer.

   b. Launch the Clock Control application, which is part of the development kit. Set new frequencies for the design example as following:

   • For NRZ mode:

   — Si5391 (U18), OUT0: Set to the value of pll_ref_clk(3) per your design requirement.

   • For PAM mode:

   — Si5391 (U45), OUT1: Set to the value of pll_ref_clk(3) per your design requirement.

   — Si5391 (U19), OUT1: Set to the value of mac_pll_ref_clk(3) per your design requirement. c. Click **Tools ➤ Programmer ➤ Hardware Setup.**

   d. Select a programming device. Add the Intel Agilex I-Series Transceiver-SoC Development Kit.

   e. Ensure that **Mode** is set to **JTAG**.

   f. Select the Intel Agilex I-Series device and click **Add Device**. The programmer displays a diagram of the connections between the devices on your board.

g. Check the box for the **.sof**.

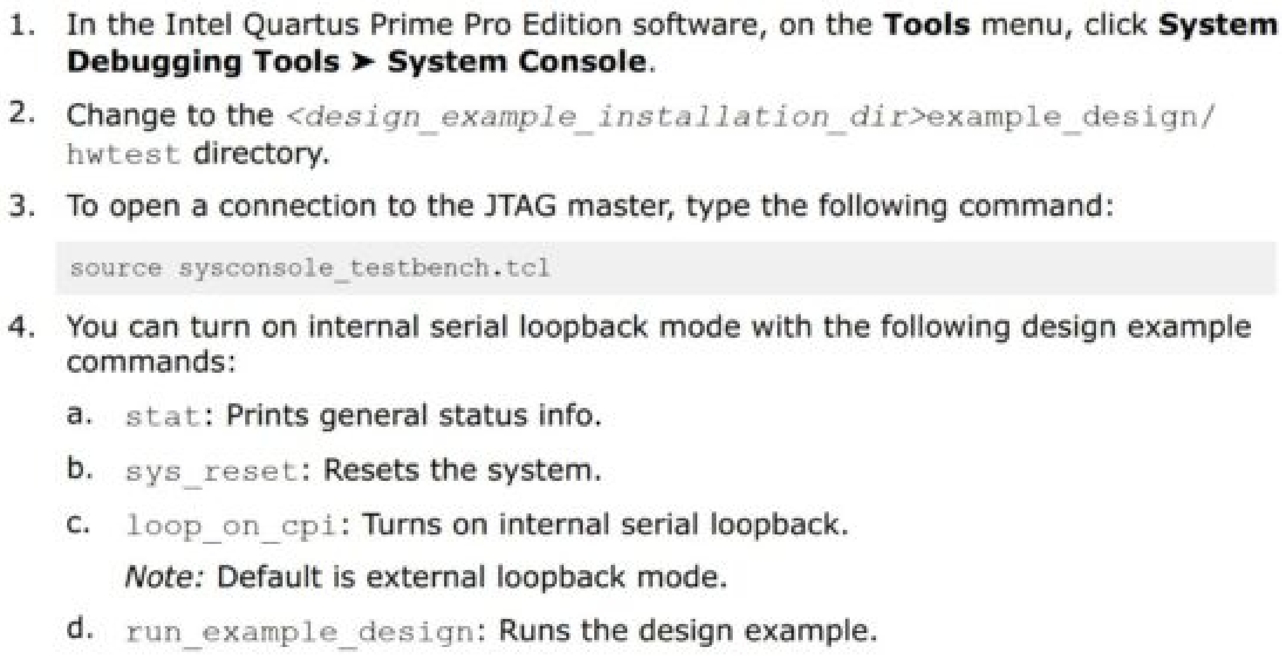
h. Check the box in the **Program**/**Configure** column.

i. Click **Start**.

1.6. Testing the Hardware Design Example

After you compile the F-tile Interlaken Intel FPGA IP design example and configure your device, you can use the System Console to program the IP core and its registers.

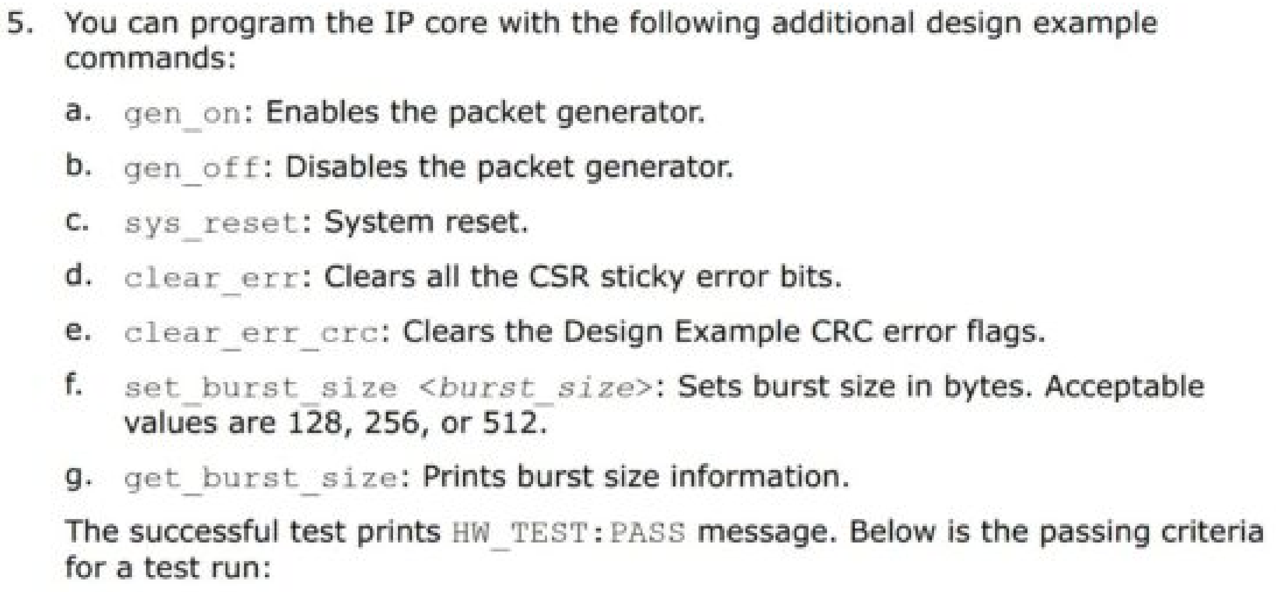
Follow these steps to bring up the System Console and test the hardware design example:



1. In the Intel Quartus Prime Pro Edition software, on the **Tools** menu, click **System Debugging Tools ➤ System Console**.
2. Change to the `<design_example_installation_dir>example_design/hwtest` directory.
3. To open a connection to the JTAG master, type the following command:

   ```
   source sysconsole_testbench.tcl
   ```

4. You can turn on internal serial loopback mode with the following design example commands:
   a. `stat`: Prints general status info.
   b. `sys_reset`: Resets the system.
   c. `loop_on_cpi`: Turns on internal serial loopback.
      Note: Default is external loopback mode.
   d. `run_example_design`: Runs the design example.

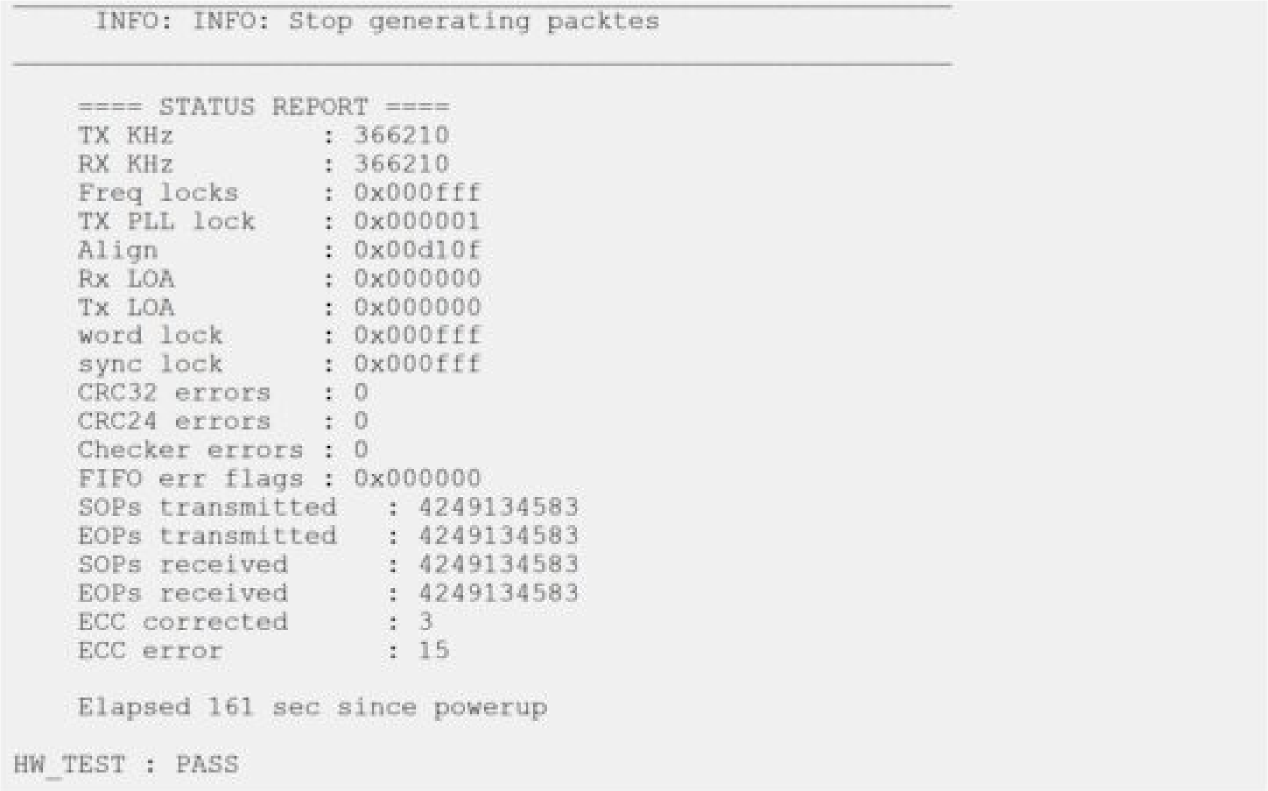


5. You can program the IP core with the following additional design example commands:
   a. `gen_on`: Enables the packet generator.
   b. `gen_off`: Disables the packet generator.
   c. `sys_reset`: System reset.
   d. `clear_err`: Clears all the CSR sticky error bits.
   e. `clear_err_crc`: Clears the Design Example CRC error flags.
   f. `set_burst_size <burst_size>`: Sets burst size in bytes. Acceptable values are 128, 256, or 512.
   g. `get_burst_size`: Prints burst size information.

   The successful test prints `HW_TEST:PASS` message. Below is the passing criteria for a test run:

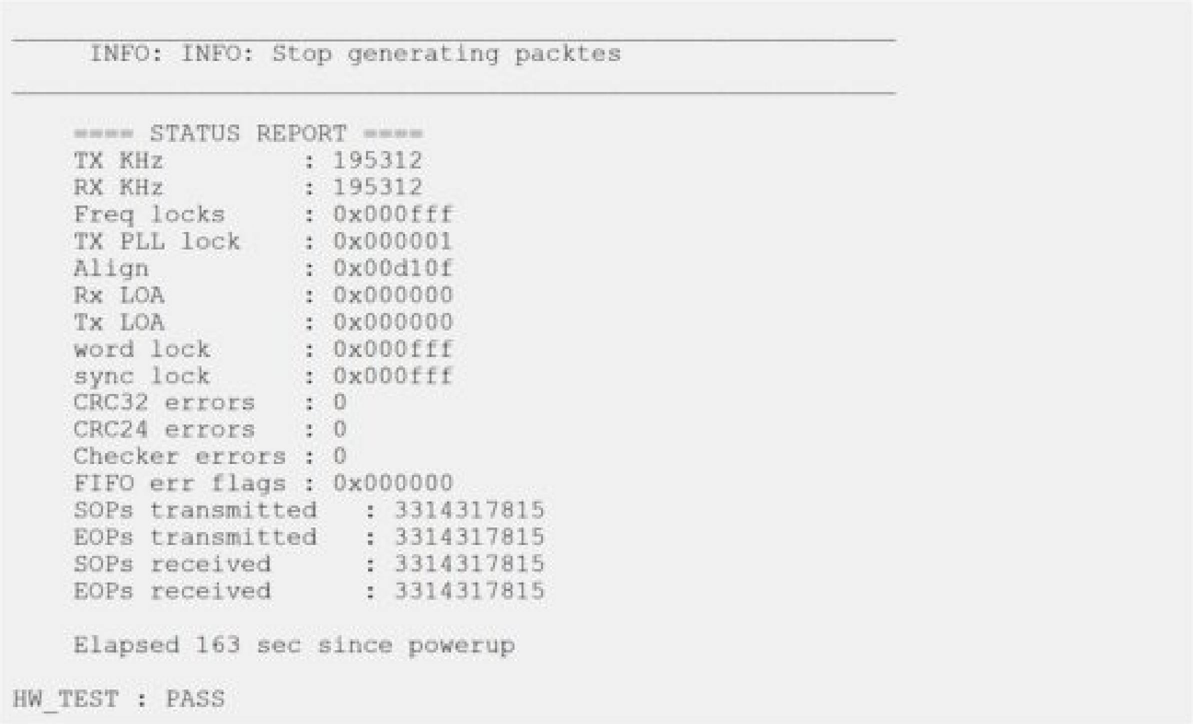
- No errors for CRC32, CRC24, and checker.
- Transmitted SOPs and EOPs should match with the received SOPs and EOPs.

The following sample output illustrates a successful test run in Interlaken mode:



```
________________________________________________________________
     INFO: INFO: Stop generating packtes
________________________________________________________________

    ==== STATUS REPORT ====
    TX KHz          : 366210
    RX KHz          : 366210
    Freq locks      : 0x000fff
    TX PLL lock     : 0x000001
    Align           : 0x00d10f
    Rx LOA          : 0x000000
    Tx LOA          : 0x000000
    word lock       : 0x000fff
    sync lock       : 0x000fff
    CRC32 errors    : 0
    CRC24 errors    : 0
    Checker errors : 0
    FIFO err flags : 0x000000
    SOPs transmitted    : 4249134583
    EOPs transmitted    : 4249134583
    SOPs received       : 4249134583
    EOPs received       : 4249134583
    ECC corrected       : 3
    ECC error           : 15

    Elapsed 161 sec since powerup

HW_TEST : PASS
```

The following sample output illustrates a successful test run in Interlaken Lookaside mode:



```
________________________________________________________________
     INFO: INFO: Stop generating packtes
________________________________________________________________

    ==== STATUS REPORT ====
    TX KHz          : 195312
    RX KHz          : 195312
    Freq locks      : 0x000fff
    TX PLL lock     : 0x000001
    Align           : 0x00d10f
    Rx LOA          : 0x000000
    Tx LOA          : 0x000000
    word lock       : 0x000fff
    sync lock       : 0x000fff
    CRC32 errors    : 0
    CRC24 errors    : 0
    Checker errors : 0
    FIFO err flags : 0x000000
    SOPs transmitted    : 3314317815
    EOPs transmitted    : 3314317815
    SOPs received       : 3314317815
    EOPs received       : 3314317815

    Elapsed 163 sec since powerup

HW_TEST : PASS
```

2. Design Example Description

The design example demonstrates the functionalities of the Interlaken IP core.

2.1. Design Example Components

The example design connects system and PLL reference clocks and required design components. The example design configures the IP core in internal loopback mode and generates packets on the IP core TX user data transfer interface. The IP core sends these packets on the internal loopback path through the transceiver.

After the IP core receiver receives the packets on the loopback path, it processes the Interlaken packets and transmits them on the RX user data transfer interface. The example design checks that the packets received and transmitted match.

The F-Tile Interlaken Intel FPGA IP design example includes the following components:

1. F-Tile Interlaken Intel FPGA IP core
2. Packet Generator and Packet Checker
3. F-Tile Reference and System PLL Clocks Intel FPGA IP core

2.2. Design Example Flow

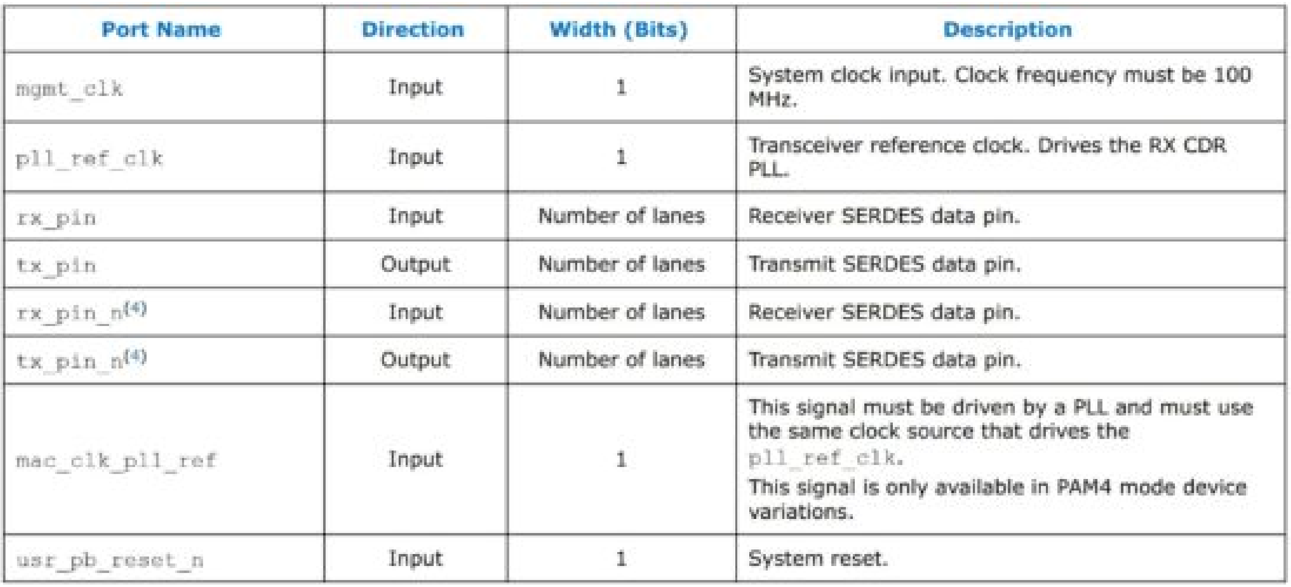
The F-Tile Interlaken Intel FPGA IP hardware design example completes the following steps:

1. Reset the the F-tile Interlaken Intel FPGA IP and F-Tile.
2. Release the reset on Interlaken IP (system reset) and F-tile TX (tile_tx_rst_n).
3. Configures the F-tile Interlaken Intel FPGA IP in the internal loopback mode.
4. Release the reset of F-tile RX (tile_rx_rst_n).
5. Sends a stream of Interlaken packets with predefined data in the payload to the TX user data transfer interface of the IP core.
6. Checks the received packets and reports the status. The packet checker included in the hardware design example provides the following basic packet checking capabilities:
   • Check that the transmitted packet sequence is correct.
   • Checks that the received data matches the expected values by ensuring both the start of packet (SOP) and end of packet (EOP) counts align while data is being transmitted and received.

*Other names and brands may be claimed as the property of others.

2.3. Interface Signals

Table 6. Design Example Interface Signals



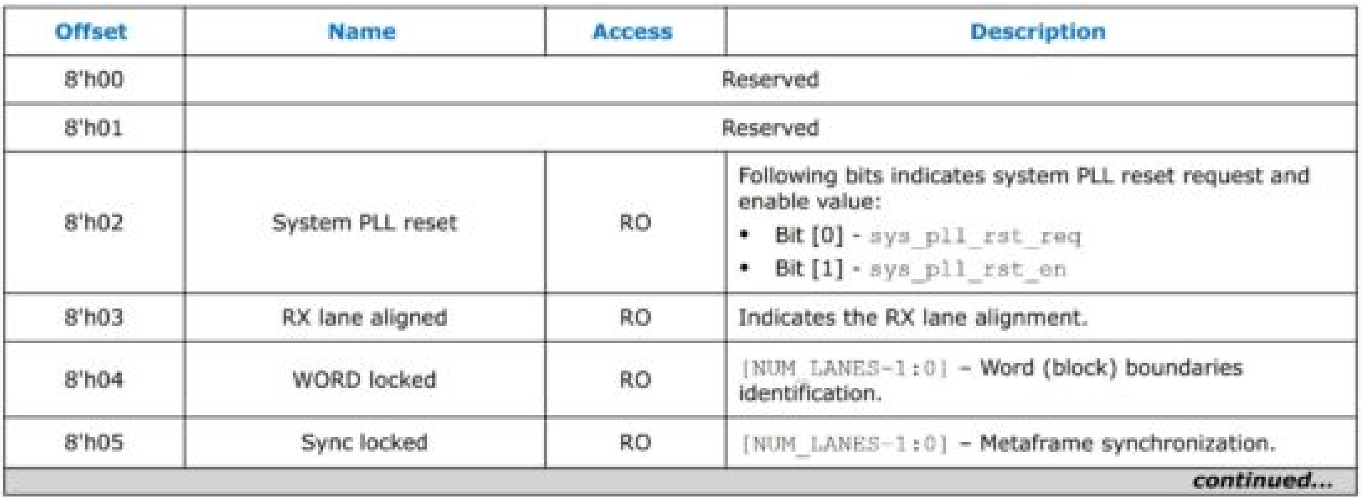
Port Name	Direction	Width (Bits)	Description
mgmt_clk	Input	1	System clock input. Clock frequency must be 100 MHz.
pll_ref_clk	Input	1	Transceiver reference clock. Drives the RX CDR PLL.
rx_pin	Input	Number of lanes	Receiver SERDES data pin.
tx_pin	Output	Number of lanes	Transmit SERDES data pin.
rx_pin_n[4]	Input	Number of lanes	Receiver SERDES data pin.
tx_pin_n[4]	Output	Number of lanes	Transmit SERDES data pin.
mac_clk_pll_ref	Input	1	This signal must be driven by a PLL and must use the same clock source that drives the pll_ref_clk. This signal is only available in PAM4 mode device variations.
usr_pb_reset_n	Input	1	System reset.

2.4. Register Map

Note:

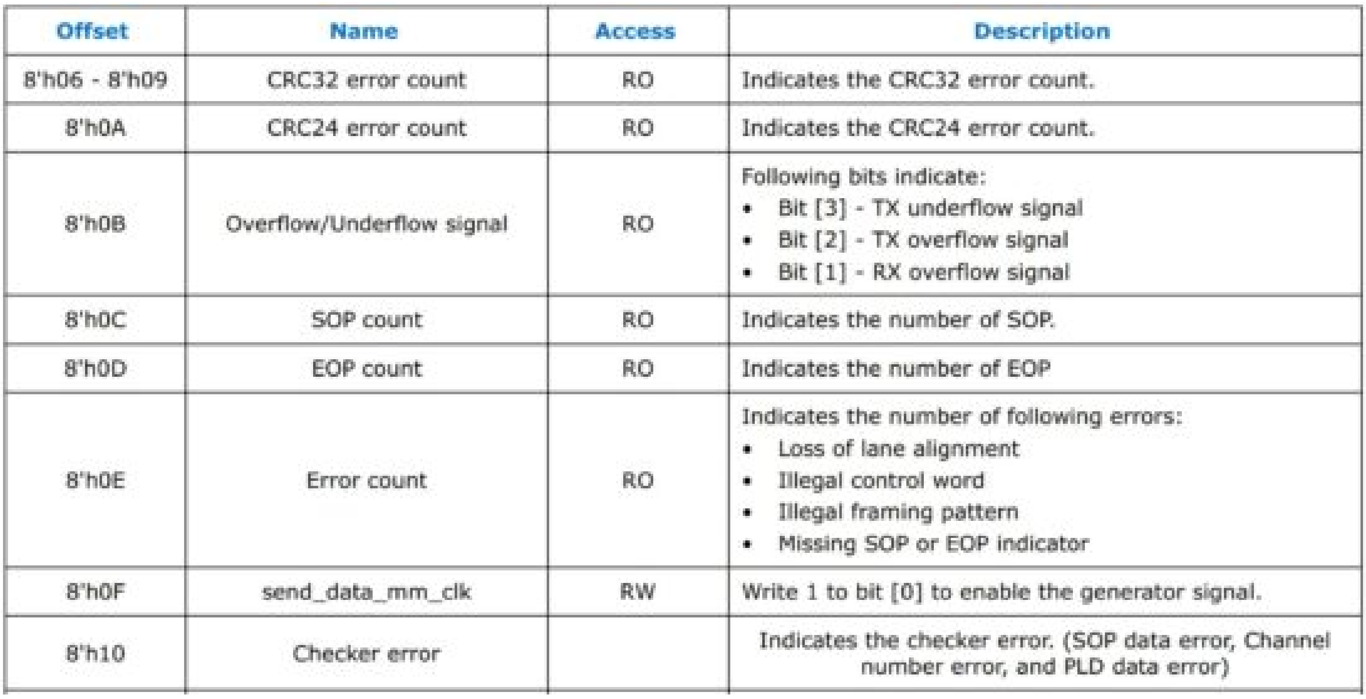
- Design Example register address starts with 0x20** while the Interlaken IP core register address starts with 0x10**.
- F-tile PHY register address starts with 0x30** while the F-tile FEC register address starts with 0x40**. FEC register is only available in PAM4 mode.
- Access code: RO—Read Only, and RW—Read/Write.
- System console reads the design example registers and reports the test status on the screen.

Table 7. Design Example Register Map

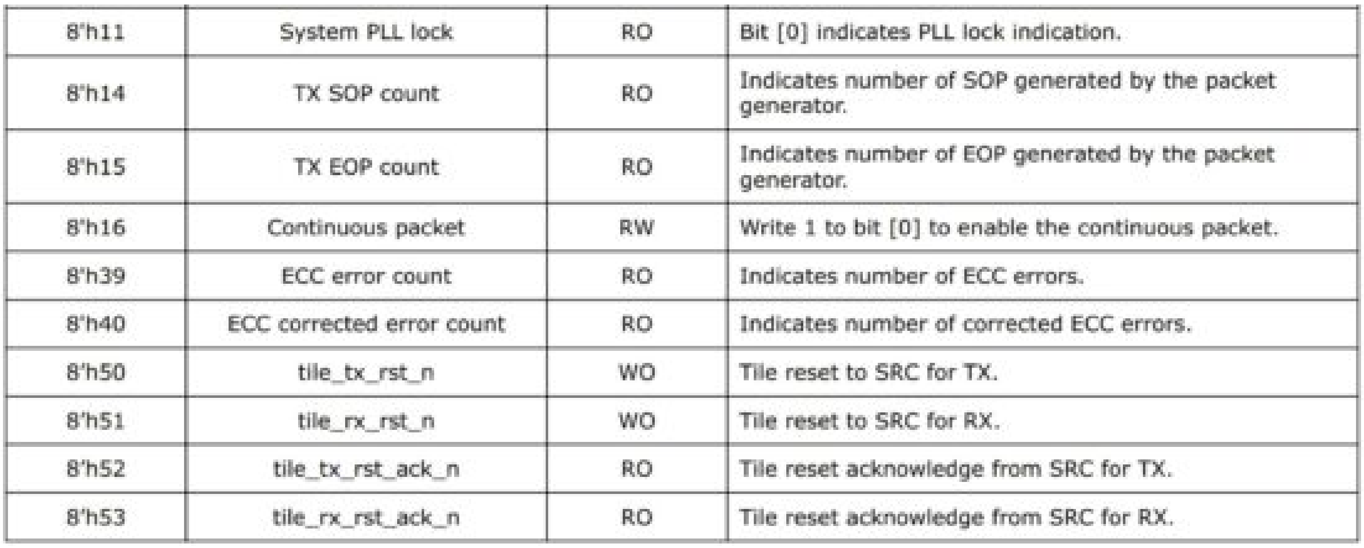


Offset	Name	Access	Description
8'h00	Reserved		
8'h01	Reserved		
8'h02	System PLL reset	RO	Following bits indicates system PLL reset request and enable value: • Bit [0] - sys_pll_rst_req • Bit [1] - sys_pll_rst_en
8'h03	RX lane aligned	RO	Indicates the RX lane alignment.
8'h04	WORD locked	RO	[NUM_LANES-1:0] – Word (block) boundaries identification.
8'h05	Sync locked	RO	[NUM_LANES-1:0] – Metaframe synchronization.

continued...



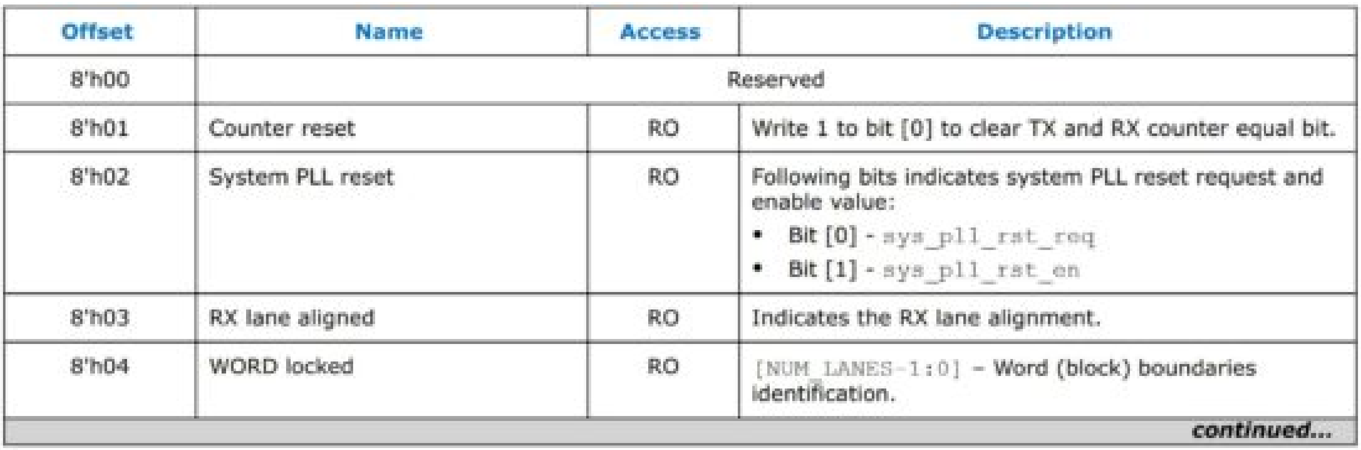
Offset	Name	Access	Description
8'h06 - 8'h09	CRC32 error count	RO	Indicates the CRC32 error count.
8'h0A	CRC24 error count	RO	Indicates the CRC24 error count.
8'h0B	Overflow/Underflow signal	RO	Following bits indicate: • Bit [3] - TX underflow signal • Bit [2] - TX overflow signal • Bit [1] - RX overflow signal
8'h0C	SOP count	RO	Indicates the number of SOP.
8'h0D	EOP count	RO	Indicates the number of EOP
8'h0E	Error count	RO	Indicates the number of following errors: • Loss of lane alignment • Illegal control word • Illegal framing pattern • Missing SOP or EOP indicator
8'h0F	send_data_mm_clk	RW	Write 1 to bit [0] to enable the generator signal.
8'h10	Checker error		Indicates the checker error. (SOP data error, Channel number error, and PLD data error)



8'h11	System PLL lock	RO	Bit [0] indicates PLL lock indication.
8'h14	TX SOP count	RO	Indicates number of SOP generated by the packet generator.
8'h15	TX EOP count	RO	Indicates number of EOP generated by the packet generator.
8'h16	Continuous packet	RW	Write 1 to bit [0] to enable the continuous packet.
8'h39	ECC error count	RO	Indicates number of ECC errors.
8'h40	ECC corrected error count	RO	Indicates number of corrected ECC errors.
8'h50	tile_tx_rst_n	WO	Tile reset to SRC for TX.
8'h51	tile_rx_rst_n	WO	Tile reset to SRC for RX.
8'h52	tile_tx_rst_ack_n	RO	Tile reset acknowledge from SRC for TX.
8'h53	tile_rx_rst_ack_n	RO	Tile reset acknowledge from SRC for RX.

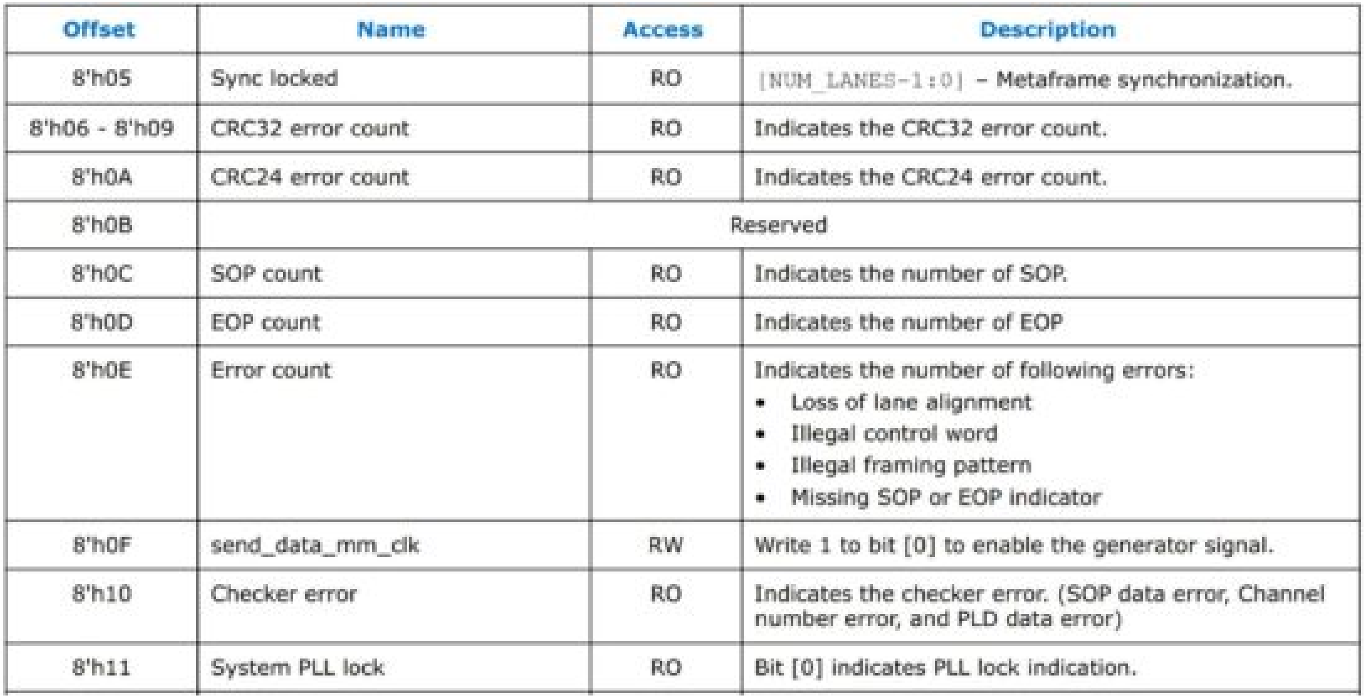
Table 8. Design Example Register Map for Interlaken Look-aside Design Example

Use this register map when you generate the design example with Enable Interlaken Look-aside Mode parameter turned on.

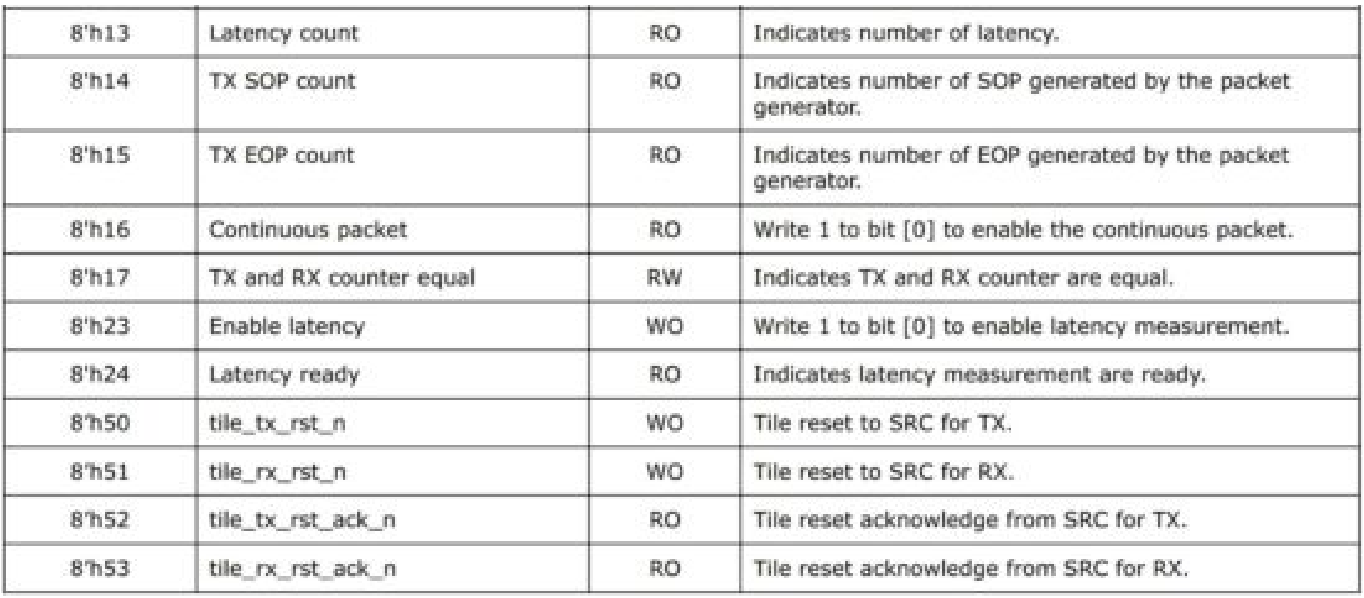


Offset	Name	Access	Description
8'h00	Reserved		
8'h01	Counter reset	RO	Write 1 to bit [0] to clear TX and RX counter equal bit.
8'h02	System PLL reset	RO	Following bits indicates system PLL reset request and enable value: • Bit [0] - `sys_pll_rst_req` • Bit [1] - `sys_pll_rst_en`
8'h03	RX lane aligned	RO	Indicates the RX lane alignment.
8'h04	WORD locked	RO	`[NUM_LANES-1:0]` – Word (block) boundaries identification.

continued...



Offset	Name	Access	Description
8'h05	Sync locked	RO	`[NUM_LANES-1:0]` – Metaframe synchronization.
8'h06 - 8'h09	CRC32 error count	RO	Indicates the CRC32 error count.
8'h0A	CRC24 error count	RO	Indicates the CRC24 error count.
8'h0B	Reserved		
8'h0C	SOP count	RO	Indicates the number of SOP.
8'h0D	EOP count	RO	Indicates the number of EOP
8'h0E	Error count	RO	Indicates the number of following errors: • Loss of lane alignment • Illegal control word • Illegal framing pattern • Missing SOP or EOP indicator
8'h0F	send_data_mm_clk	RW	Write 1 to bit [0] to enable the generator signal.
8'h10	Checker error	RO	Indicates the checker error. (SOP data error, Channel number error, and PLD data error)
8'h11	System PLL lock	RO	Bit [0] indicates PLL lock indication.



8'h13	Latency count	RO	Indicates number of latency.
8'h14	TX SOP count	RO	Indicates number of SOP generated by the packet generator.
8'h15	TX EOP count	RO	Indicates number of EOP generated by the packet generator.
8'h16	Continuous packet	RO	Write 1 to bit [0] to enable the continuous packet.
8'h17	TX and RX counter equal	RW	Indicates TX and RX counter are equal.
8'h23	Enable latency	WO	Write 1 to bit [0] to enable latency measurement.
8'h24	Latency ready	RO	Indicates latency measurement are ready.
8'h50	tile_tx_rst_n	WO	Tile reset to SRC for TX.
8'h51	tile_rx_rst_n	WO	Tile reset to SRC for RX.
8'h52	tile_tx_rst_ack_n	RO	Tile reset acknowledge from SRC for TX.
8'h53	tile_rx_rst_ack_n	RO	Tile reset acknowledge from SRC for RX.

2.5. Reset

In the F-Tile Interlaken Intel FPGA IP core, you initiate the reset (reset_n=0) and hold until the IP core returns a reset acknowledge (reset_ack_n=0). After the reset is removed (reset_n=1), the reset acknowledge returns to its initial state (reset_ack_n=1). In the design example, a rst_ack_sticky register holds the reset acknowledge assertion and then triggers the removal of the reset (reset_n=1). You can use alternative methods that fit your design needs.

Important: In any scenario where the internal serial loopback is required, you must release TX and RX of the F-tile separately in a specific order. Refer to the system console script for more information.

Figure 7. Reset Sequence in NRZ Mode

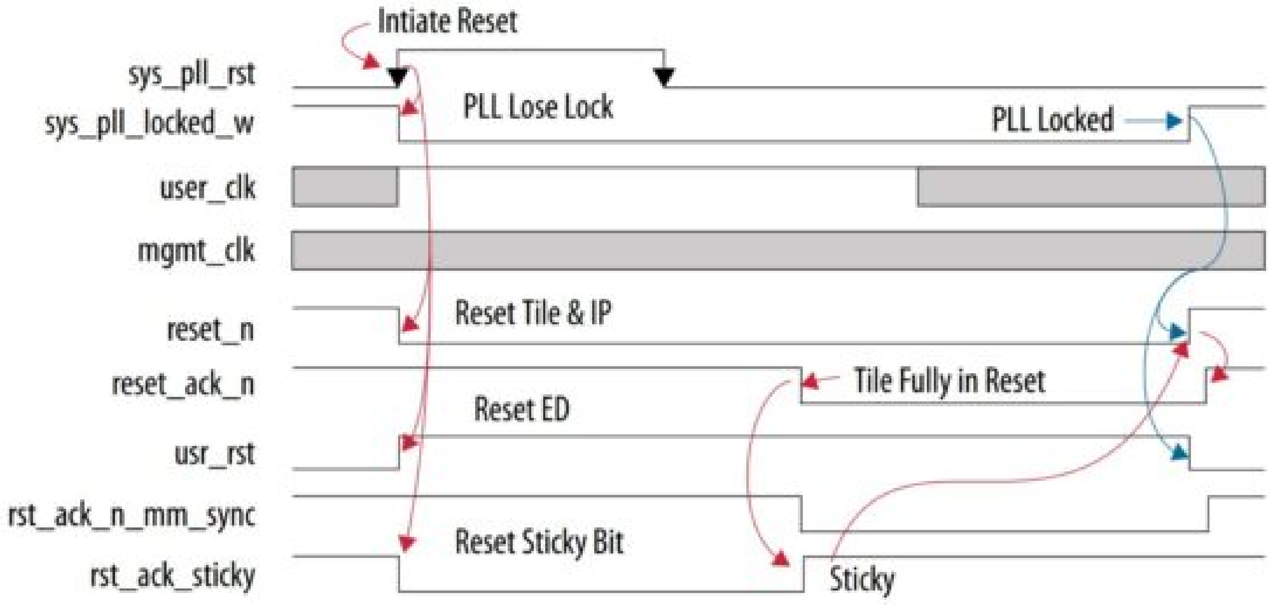
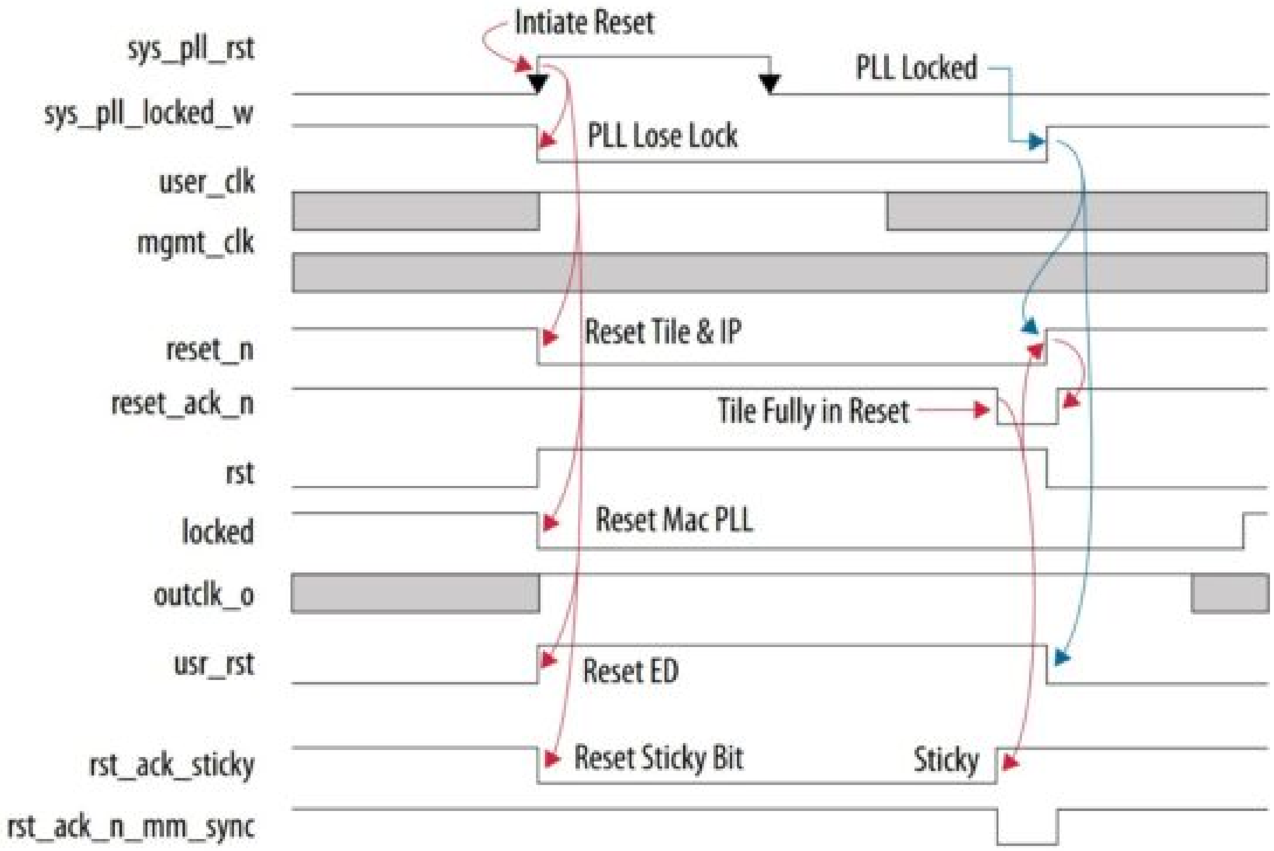


Figure 8. Reset Sequence in PAM4 Mode



3. F-Tile Interlaken Intel FPGA IP Design Example User Guide 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
21.3	3.0.0	F-Tile Interlaken Intel FPGA IP Design Example User Guide
21.2	2.0.0	F-Tile Interlaken Intel FPGA IP Design Example User Guide

4. Document Revision History for F-Tile Interlaken Intel FPGA IP Design Example User Guide

Document Version	Intel Quartus Prime Version	IP Version	Changes
2022.01.14	21.4	3.1.0	• Added support for the Cadence* Xcelium simulator. • Added support for the Interlaken Look-aside mode. • Added hardware support for the F-tile Interlaken Intel FPGA IP Design Example.
2021.10.04	21.3	3.0.0	• Added support for new lane rate combinations. For more information, refer to *Table: IP Supported Combinations of Number of Lanes and Data Rate*. • Updated the supported simulator list in section: *Hardware and Software Requirements*. • Added new reset registers in section: *Register Map*.
2021.06.21	21.2	2.0.0	Initial release.

Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Read More About This Manual & Download PDF:

Documents / Resources

 F-Tile Interlaken Intel® FPGA IP Design Example User Guide Updated for Intel® Quartus® Prime Design Suite: 21.4 IP Version: 3.1.0 Online Version Send Feedback UG-20334 ID: 683069 Version: 2022.01.14	intel F-Tile Interlaken FPGA IPDesign Example [pdf] User Guide F-Tile Interlaken FPGA IPDesign Example