50G Interlaken Design Example User Guide
Contents

Quick Start Guide.................................................................................................................................1-1
  Directory Structure.............................................................................................................................1-2
  Design Components............................................................................................................................1-3
  Generating the Design.........................................................................................................................1-3
  Simulating the Design.........................................................................................................................1-5
  Compiling and Testing the Design.......................................................................................................1-5

Design Example Description................................................................................................................2-1
  Features..............................................................................................................................................2-1
  Hardware and Software Requirements.................................................................................................2-1
  Functional Description........................................................................................................................2-1
  Example Design Behavior....................................................................................................................2-3
  Interface Signals................................................................................................................................2-3
  Register Map.....................................................................................................................................2-3

Document Revision History for 50G Interlaken Design Example User Guide
................................................................................................................................................................A-1
The Intel® Arria® 10 variations of the 50G Interlaken IP core feature a simulating testbench and a hardware example design that supports compilation and hardware testing, to help you understand usage. When you generate the example design, 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 Arria 10 GX Transceiver Signal Integrity Development Kit. The testbench and demonstration example design are available for a wide range of parameters. However, they do not cover all possible parameterizations of the 50G Interlaken IP Core.

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

**Figure 1-1: Development Steps**
Directory Structure

Figure 1-2: Directory Structure for the Generated Example Design

The hardware configuration, simulation, and test files are located in `<example_design_install_dir>/example_design_a10`. 
Design Components

Figure 1-3: Example Design Block Diagram

Generating the Design

Figure 1-4: Procedure

Follow these steps to generate the Intel Arria 10 hardware example design and testbench:

1. In the IP Catalog (Tools > IP Catalog), select the Intel Arria 10 target device family.
   
   **Note:** The Quick Start hardware example design is only supported in Intel Arria 10 devices. The testbench is available for variations that target Intel Arria 10 devices or Stratix® V devices.

2. In the IP Catalog, locate and double-click 50G Interlaken. The New IP Variation window appears.

3. Specify a top-level name for your custom IP variation. The parameter editor saves the IP variation settings in a file named <your_ip>.qsys.

4. You must select a specific Intel Arria 10 device in the Device field, or keep the default Intel Quartus® Prime software device selection.

5. Click OK. The parameter editor appears.
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, and select the **Synthesis** option to generate the hardware example design.

   **Note:** At least one of the **Simulation** and **Synthesis** check boxes from **Example Design Files** must be selected to allow generation of Example Design Files.

8. For **Generated HDL Format**, only **Verilog** is available.

9. For **Target Development Kit** select the **Intel Arria 10 GX Transceiver Signal Integrity Development Kit**. If you select a development kit, then the target device (selected in step 4) for Example Design is changed to match the device on target board.

10. Click the **Generate Example Design** button.

**Related Information**

**50G Interlaken IP Core Parameter Settings**
Provides information about the 50G Interlaken IP core customization by specifying parameters.
Simulating the Design

Figure 1-6: Procedure

Follow these steps to simulate the testbench:

1. Change to the testbench simulation directory <example_design_install_dir>/example_design_a10/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. Refer to the table Steps to Run Simulation.
3. Analyze the results.

Table 1-1: Steps to Run Simulation

<table>
<thead>
<tr>
<th>Simulator</th>
<th>Instructions</th>
</tr>
</thead>
</table>
| ModelSim* SE or Questa* or Questa-Intel FPGA Edition | In the command line, type
\verbatim
vsim -do vlog_pro.do
\endverbatim
If you prefer to simulate without bringing up the GUI, type
\verbatim
vsim -c -do vlog_pro.do
\endverbatim |
| VCS* | In the command line, type \verbatim
sh vcstest.sh
\endverbatim |

A successful simulation ends with the following message:

Test PASSED

Compiling and Testing the Design
To compile and run a demonstration test on the hardware example design, follow these steps:

1. Ensure hardware example design generation is complete.
2. Open the Intel Quartus Prime project `<user-specified location>/example_design_a10/quartus/example_design.qpf`, where `<user-specified location>` is the directory location you specified when you generated the testbench and hardware example design.
3. On the Processing menu, click Start Compilation.
4. After successful compilation, a .sof file will be generated in your specified directory. Follow these steps to program the hardware example design on the Intel Arria 10:
   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 Arria 10 GX Transceiver Signal Integrity Development Kit to which your Intel Quartus Prime session can connect.
   e. Ensure that Mode is set to JTAG.
   f. Select the Arria 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.
5. After the hardware example design is configured on the Intel Arria 10 device, in the Intel Quartus Prime software, on the Tools menu, click System Debugging Tools > System Console.
6. In the Tcl Console pane, type sysconsole_testbench.tcl.
7. Type run_example_design.
The example design demonstrate the functionalities of the 50G Interlaken IP core. You can generate the design from the Example Design tab of the 50G Interlaken graphical user interface (GUI) in the IP parameter editor.

Features

- Internal TX to RX serial loopback mode.
- Automatically generates fixed size packets.
- Basic packet checking capabilities.
- You can use system console to reset the design for re-testing purpose.

Hardware and Software Requirements

Altera uses the following hardware and software to test the example design:

- Quartus Prime software
- System Console
- VCS, ModelSim SE, Questa, or Questa-Intel FPGA Edition simulator
- Arria 10 GX Transceiver Signal Integrity Development Kit for hardware testing

Functional Description

The hardware example design connects system and PLL reference clocks and required design components. After you program the device on the Arria 10 GX transceiver signal integrity development board, 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 it receives on the IP core RX user data transfer interface are consistent with the packets sent in.
The hardware example design includes external PLLs. You can examine the clear text files to view sample code that implements one possible method to connect external PLLs to the 50G Interlaken IP core.

The hardware example design packs six Interlaken lanes in a transceiver block, and connects all of the channels in the same transceiver block to a single ATX PLL. The IP core connects ATX PLL to the 50G Interlaken IP core tx_pll_locked and tx_pll_powerdown ports. This simple connection model is only one of many options available to you for configuring and connecting the external PLLs in your 50G Interlaken design.

Related Information

- Arria 10 GX Transceiver Signal Integrity Development Kit product webpage
- 50G Interlaken MegaCore Function User Guide
Example Design Behavior

Immediately following configuration on the Arria 10 device, when you type `run_example_design` in system console, the 50G Interlaken IP core hardware example design performs the following actions:

1. Resets the 50G Interlaken IP core.
2. Configures the 50G Interlaken IP core in internal loopback mode.
3. Sends a sequence of 100 256-byte Interlaken packets with predefined data in the payload to the TX user data transfer interface of the IP core.
4. Checks the received packets and reports the status.

The packet checker included in the hardware example design provides the following basic packet checking capabilities:

- Checks that the transmitted packet sequence is not violated
- Checks that the received data matches expected values

Interface Signals

Table 2-1: Arria 10 50G Interlaken IP Core Hardware Example Design Signals

<table>
<thead>
<tr>
<th>Port Name</th>
<th>Direction</th>
<th>Width (Bits)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk50</td>
<td>Input</td>
<td>1</td>
<td>System clock input. Clock frequency must be 50 MHz.</td>
</tr>
<tr>
<td>pll_ref_clk</td>
<td>Input</td>
<td>1</td>
<td>Transceiver reference clock. Drives the RX CDR PLL.</td>
</tr>
<tr>
<td>rx_pin</td>
<td>Input</td>
<td>Number of lanes</td>
<td>Receiver SERDES data pin.</td>
</tr>
<tr>
<td>tx_pin</td>
<td>Output</td>
<td>Number of lanes</td>
<td>Transmit SERDES data pin.</td>
</tr>
<tr>
<td>sys_pll_reset_n</td>
<td>Input</td>
<td>1</td>
<td>System reset.</td>
</tr>
</tbody>
</table>

Register Map

Table 2-2: Example Design Register Map

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Access</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8'h00</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8'h01</td>
<td>Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Access</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>8’h02</td>
<td>System PLL reset</td>
<td>RO</td>
<td>Following bits indicates system PLL reset request and enable value:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Bit [0] - sys_pll_rst_req</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Bit [1] - sys_pll_rst_en</td>
</tr>
<tr>
<td>8’h03</td>
<td>RX lane aligned</td>
<td>RO</td>
<td>Indicates the RX lane alignment.</td>
</tr>
<tr>
<td>8’h04</td>
<td>WORD locked</td>
<td>RO</td>
<td>[NUM_LANES–1:0] – Word (block) boundaries identification.</td>
</tr>
<tr>
<td>8’h05</td>
<td>Sync locked</td>
<td>RO</td>
<td>[NUM_LANES–1:0] – Metaframe synchronization.</td>
</tr>
<tr>
<td>8’h06 - 8’h09</td>
<td>CRC32 error count</td>
<td>RO</td>
<td>Indicates the CRC32 error count.</td>
</tr>
<tr>
<td>8’h0A</td>
<td>CRC24 error count</td>
<td>RO</td>
<td>Indicates the CRC24 error count.</td>
</tr>
<tr>
<td>8’h0B</td>
<td>Overflow/Underflow signal</td>
<td>RO</td>
<td>Following bits indicate:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Bit [3] - TX underflow signal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Bit [2] - TX overflow signal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Bit [1] - RX overflow signal</td>
</tr>
<tr>
<td>8’h0C</td>
<td>SOP count</td>
<td>RO</td>
<td>Indicates the number of SOP.</td>
</tr>
<tr>
<td>8’h0D</td>
<td>EOP count</td>
<td>RO</td>
<td>Indicates the number of EOP</td>
</tr>
<tr>
<td>8’h0E</td>
<td>Error count</td>
<td>RO</td>
<td>Indicates the number of following errors:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Loss of lane alignment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Illegal control word</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Illegal framing pattern</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Missing SOP or EOP indicator</td>
</tr>
<tr>
<td>8’h0F</td>
<td>send_data_mm_clk</td>
<td>RW</td>
<td>Write 1 to enable the generator signal.</td>
</tr>
<tr>
<td>8’h10</td>
<td></td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>8’h11</td>
<td>System PLL lock</td>
<td>RO</td>
<td>PLL lock indication.</td>
</tr>
</tbody>
</table>

**Note:**
- Example design register address starts with 0x20** while the 50G Interlaken core register address starts with 0x10**.
- Access code: RO—Read Only, and RW—Read/Write.
- System console reads the example design registers and reports the test status on the screen.
<table>
<thead>
<tr>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021.10.04</td>
<td>Updated simulator information in section: <em>Simulating the Design</em> and <em>Hardware and Software Requirements</em>.</td>
</tr>
<tr>
<td>2018.03.22</td>
<td>Made following changes to <em>Simulating the Design</em> section:</td>
</tr>
<tr>
<td></td>
<td>• Corrected simulation directory location.</td>
</tr>
<tr>
<td></td>
<td>• Updated command to simulate the testbench in NCSim and VCS simulators.</td>
</tr>
<tr>
<td></td>
<td>• Updated simulation ends display message.</td>
</tr>
<tr>
<td>2016.10.31</td>
<td>• Changed the title to <em>50G Interlaken Design Example User Guide</em>.</td>
</tr>
<tr>
<td></td>
<td>• Corrected the Figure: <em>Arria 10 50G Interlaken IP Core Example Design Block Diagram</em>.</td>
</tr>
<tr>
<td></td>
<td>• Updated the Figure: <em>Directory Structure for the Generated Example Design</em>.</td>
</tr>
<tr>
<td>2016.05.02</td>
<td>Initial release</td>
</tr>
</tbody>
</table>