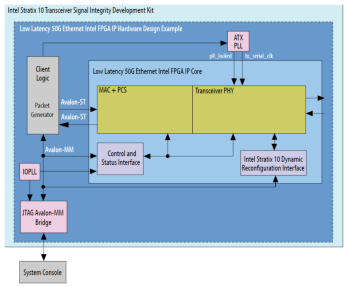
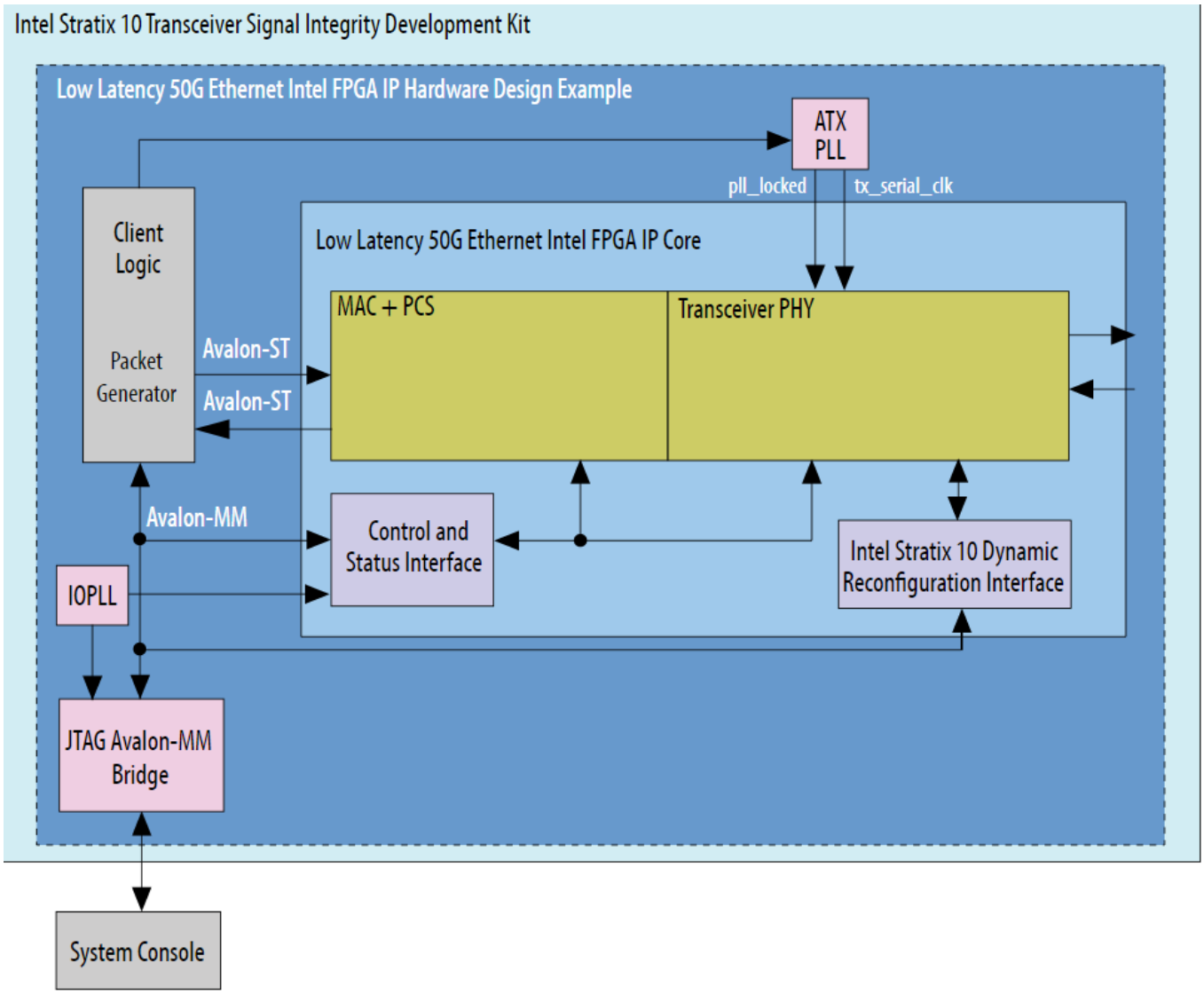


intel 50G Ethernet Design Example User Guide

Contents

intel 50G Ethernet Design Example

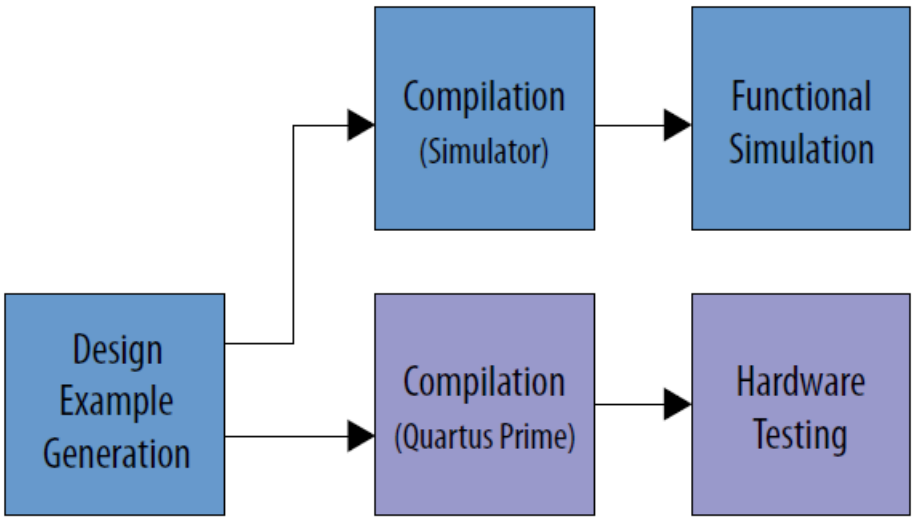


50GbE Quick Start Guide

The 50GbE IP core provides a simulation testbench and a hardware design example that supports compilation and hardware testing. When you generate the design example, the parameter editor automatically creates the files necessary to simulate, compile, and test the design in hardware. You can download the compiled hardware design to an Arria 10 GT device.

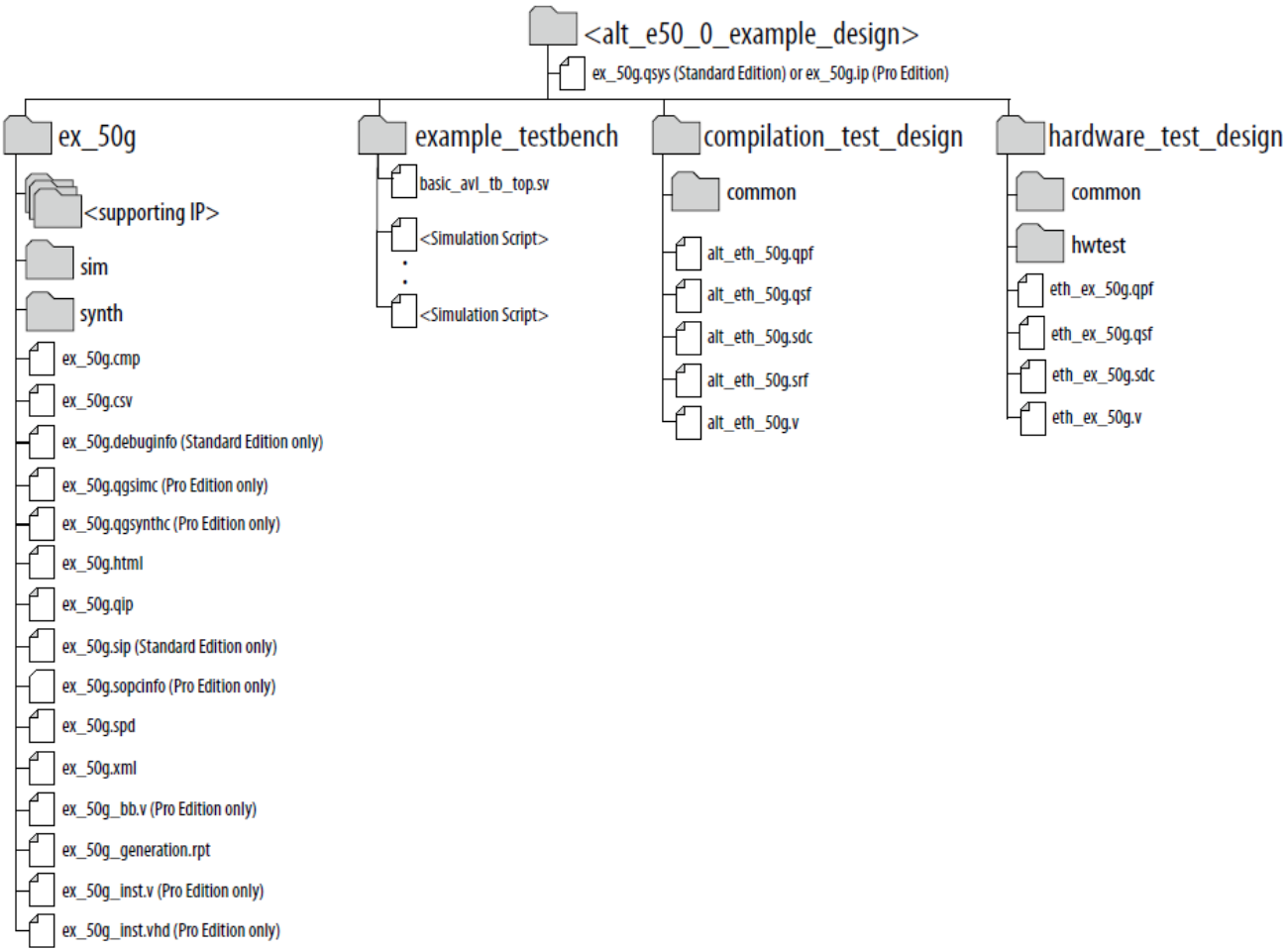
Note: This design example targets the Arria 10 GT device and requires a 25G retimer. Please contact your Intel FPGA representative to inquire about a platform suitable to run this hardware example. In some cases a loan of appropriate hardware may be available. In addition, Intel provides a compilation-only example project that you can use to quickly estimate IP core area and timing.

Figure 1. Design Example Usage



Design Example Directory Structure

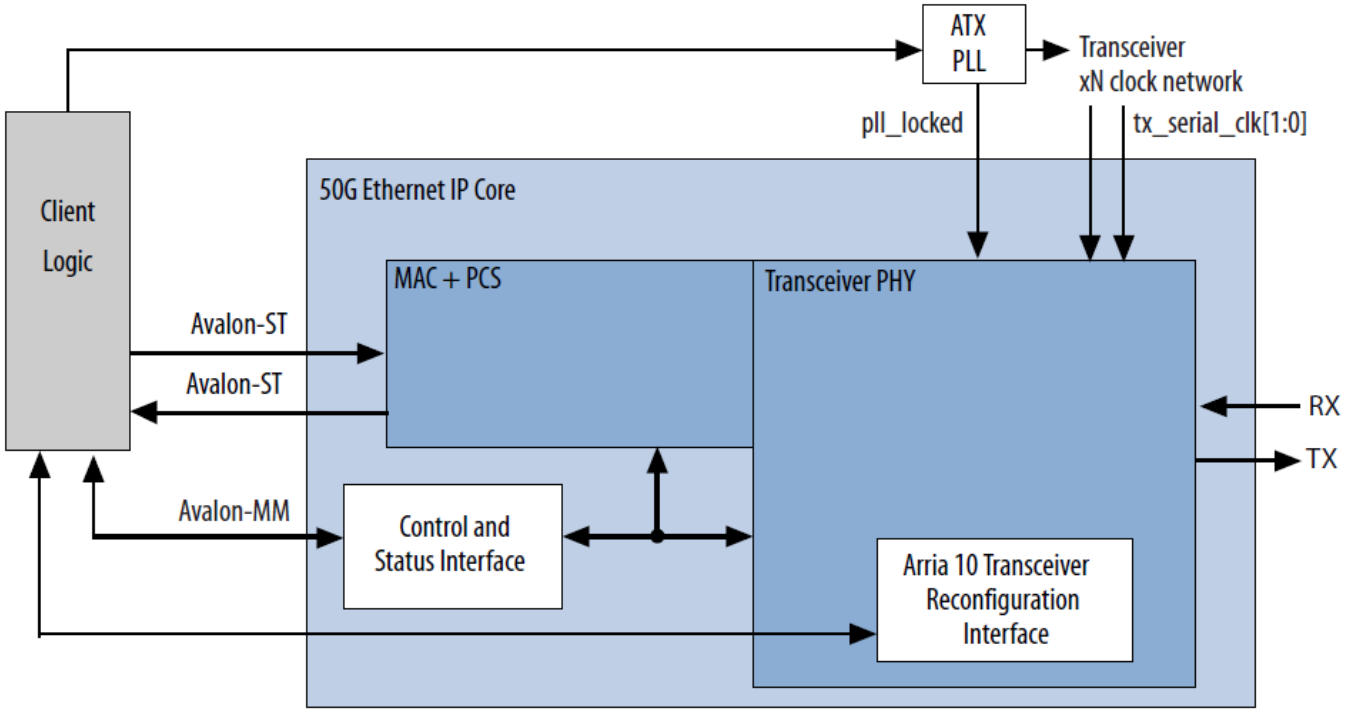
Figure 2. 50GbE Design Example Directory Structure



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

Simulation Design Example Components

Figure 3. 50GbE Simulation Design Example Block Diagram



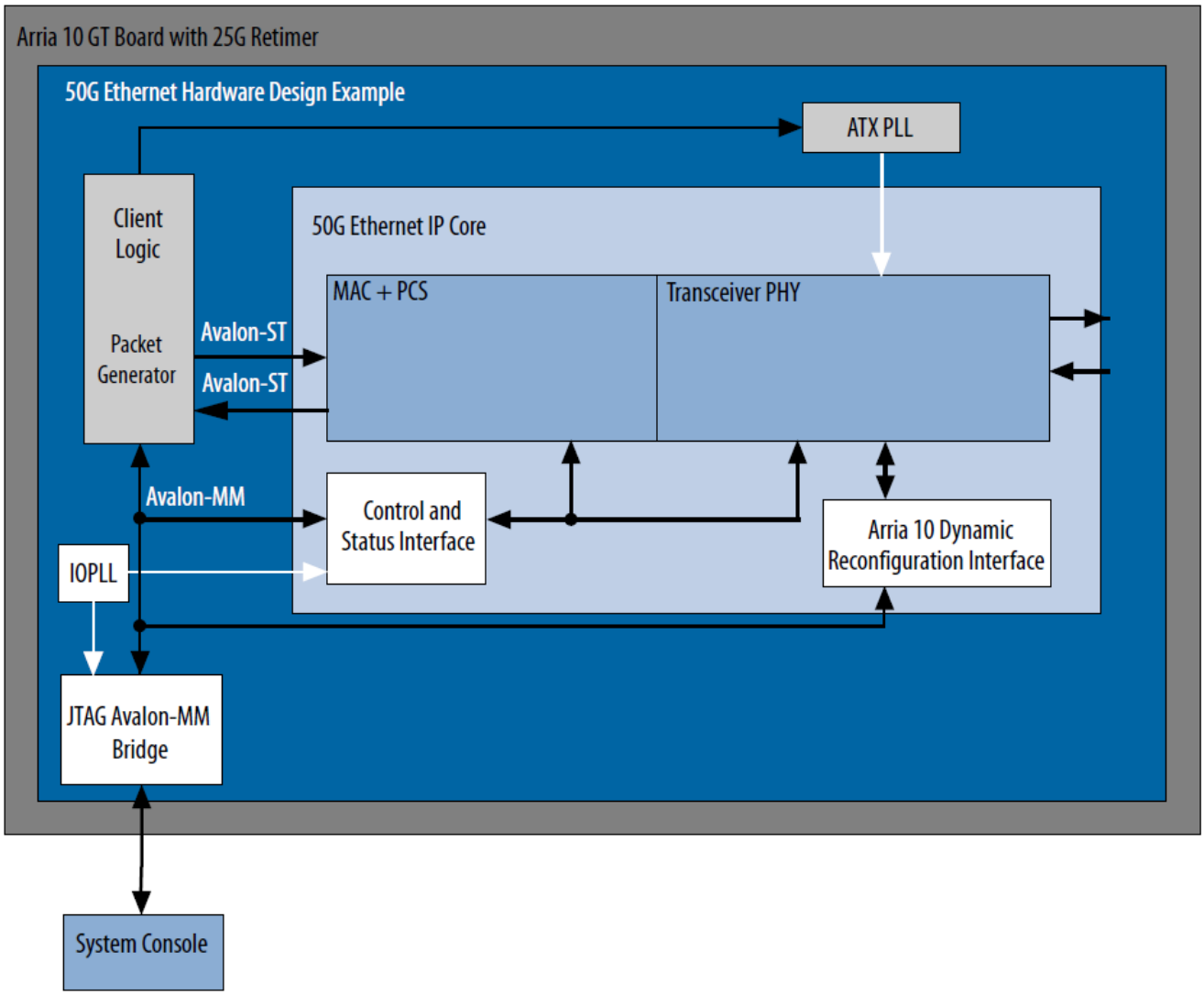
The simulation example design top-level test file is basic_avl_tb_top.sv This file instantiates and connects an ATX PLL. It includes a task, send_packets_50g_avl, to send and receive 10 packets.

Table 1. 50GbE IP Core Testbench File Descriptions

File Name	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 acce pt packets.
Testbench Scripts	
run_vsim.do	The ModelSim script to run the testbench.
run_vcs.sh	The Synopsys VCS script to run the testbench.
run_ncsim.sh	The Cadence NCSim script to run the testbench.
run_xcelium.sh	The Cadence Xcelium* script to run the testbench.

rdware Design Example Components

Figure 4. 50GbE Hardware Design Example High Level Block Diagram



The 50GbE hardware design example includes the following components

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

Table 2. 50GbE IP Core Hardware Design Example File Descriptions

File Names	Description
eth_ex_50g.qpf	Quartus Prime project file
eth_ex_50g.qsf	Quartus project settings file
eth_ex_50g.sdc	Synopsys Design Constraints file. You can copy and modify this file for your own 50GbE d esign.
continued...	

File Names	Description
eth_ex_50g.v	Top-level Verilog HDL design example file
common/	Hardware design example support files
hwtest/main.tcl	Main file for accessing System Console

Generating the Design Example

Figure 5. Procedure

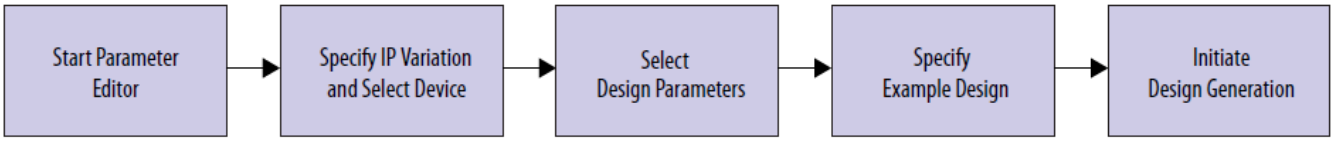
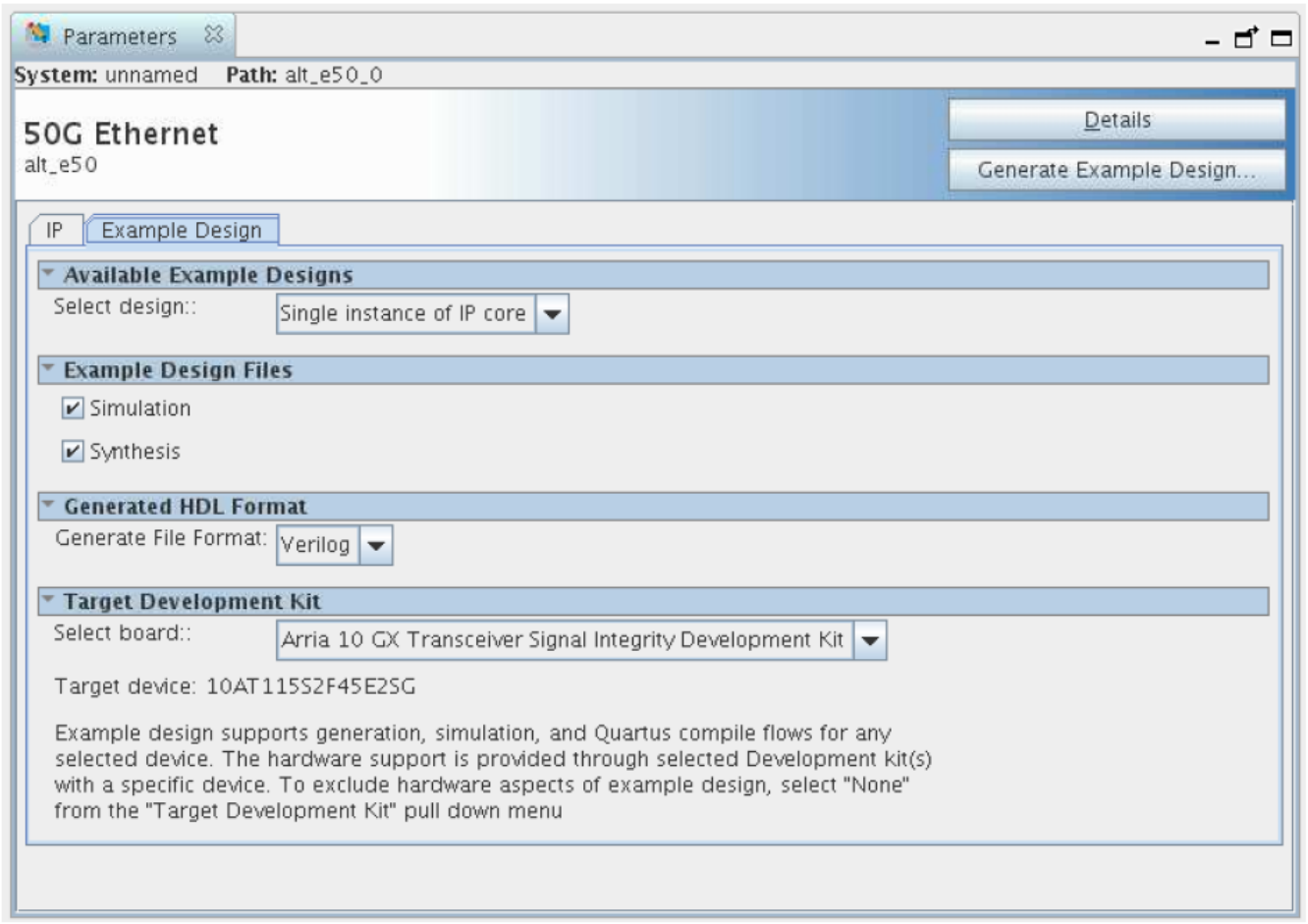


Figure 6. Example Design Tab in the 50GbE Parameter Editor



Follow these steps to generate the hardware design example and testbench

1. Depending on whether you are using the Intel Quartus® Prime Pro Edition software or the Intel Quartus Prime Standard Edition software, perform one of the following actions: 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 Quartus Prime project. The wizard prompts you to specify a device. In the Intel Quartus Prime Standard

Edition software, in the IP Catalog (Tools IP Catalog), select the Arria 10 target device family.

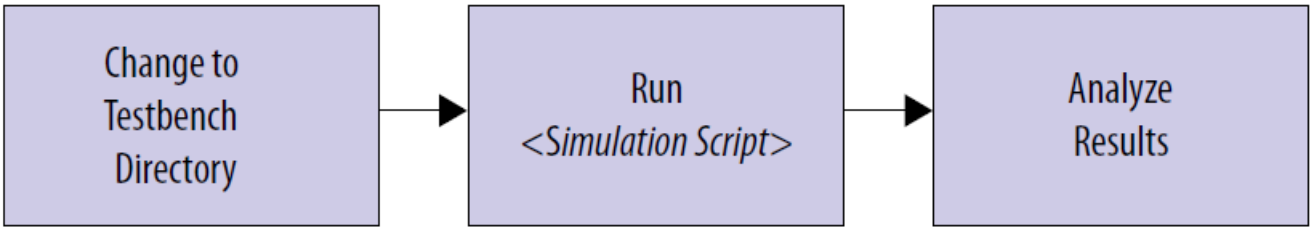
2. In the IP Catalog, locate and select 50G Ethernet. The New IP Variation window appears.
3. Specify a top-level name for your IP variation and click OK. The parameter editor adds the top-level .qsys (in Intel Quartus Prime Standard Edition) or .ip (in Intel Quartus Prime Pro Edition) file to the current project automatically. If you are prompted to manually add the .qsys or .ip file to the project, click Project ➤ Add/Remove Files in Project to add the file.
4. In the Intel Quartus Prime Standard Edition software, you must select a specific Arria 10 device in the Device field, or keep the default device the Quartus Prime software proposes.
   Note: The hardware 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).
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, for Example Design Files, select the Simulation option to generate the testbench, and select the Synthesis option to generate the hardware design example. Only Verilog HDL files are generated.
   Note: A functional VHDL IP core is not available. Specify Verilog HDL only, for your IP core design example.
8. For Hardware Board select the Arria 10 GX Transceiver Signal Integrity Development Kit.
   Note: Contact your Intel FPGA representative for information about a platform suitable to run this hardware example.
9. Click the Generate Example Design button. The Select Example Design Directory window appears.
10. If you wish to modify the design example directory path or name from the defaults displayed (alt_e50_0_example_design), browse to the new path and type the new design example directory name (<design_example_dir>).
11. Click OK.
12. Refer to the KDB Answer How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Arria 10 PLL reference clock? for a workaround you should apply in the hardware_test_design directory in the .sdc file.

Note: You must consult this KDB Answer because the RX path in the 50GbE IP core includes cascaded PLLs. Therefore, the IP core clocks might experience additional jitter in Arria 10 devices. This KDB Answer clarifies the software releases in which the workaround is necessary.

Related Information
KDB Answer: How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Arria 10 PLL reference clock?

Simulating the 50GbE Design Example Testbench

Figure 7. Procedure



Follow these steps to simulate the testbench

1. Change to the testbench simulation directory <design_example_dir>/ example_testbench.
2. Run the simulation script for the supported simulator of your choice. The script compiles and runs the testbench in the simulator. Refer to the table “Steps to Simulate the Testbench”.
3. Analyze the results. The successful testbench sends ten packets, receives ten packets, and displays “Testbench complete.”

Table 3. Steps to Simulate the Testbench

Simulator	Instructions
ModelSim	In the command line, type vsim -do run_vsim.do If you prefer to simulate without bringing up the ModelSim GUI, type vsim -c -do run_vsim.do *Note:* The ModelSim* – Intel FPGA Edition simulator does not have the capacity to simulate this IP core. You must use another supported ModelSim simulator such as ModelSim SE.
NCSim	In the command line, type sh run_ncsim.sh
VCS	In the command line, type sh run_vcs.sh
Xcelium	In the command line, type sh run_xcelium.sh

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

- #Ref clock is run at 625 MHz so whole numbers can used for all clock periods.
- #Multiply reported frequencies by 33/32 to get actual clock frequencies.
- #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.
- #**
- #**

Compiling and Configuring the Design Example in Hardware

To compile the hardware design example and configure it on your Arria 10 GT device,follow these steps

1. Ensure hardware design example generation is complete.
2. In the Intel Quartus Prime software, open the Intel Quartus Prime project <design_example_dir>/hardware_test_design/eth_ex_50g.qpf.
3. Before compiling, ensure you have implemented the workaround from the KDB Answer How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Arria 10 PLL reference clock? if relevant for your software release.
4. On the Processing menu, click Start Compilation.
5. After you generate a SRAM object file .sof, follow these steps to program the hardware design example on the Arria 10 device:

- On the Tools menu, click Programmer.
- In the Programmer, click Hardware Setup.
- Select a programming device.
- Select and add the Arria 10 GT board with 25G retimer to your Intel Quartus Prime session.
- Ensure that Mode is set to JTAG.
- Select the Arria 10 device and click Add Device. The Programmer displays a block diagram of the connections between the devices on your board.
- In the row with your .sof, check the box for the .sof.
- Check the box in the Program/Configure column.
- Click Start

Note: This design example targets the Arria 10 GT device. Please contact your Intel FPGA representative to inquire about a platform suitable to run this hardware example

Related Information

- KDB Answer: How do I compensate for the jitter of PLL cascading or nondedicated clock path for Arria 10 PLL reference clock?
- Incremental Compilation for Hierarchical and Team-Based Design
- Programming Intel FPGA Devices

Testing the 50GbE Hardware Design Example

After you compile the 50GbE IP core design example and configure it on your Arria 10 GT device, you can use the System Console to program the IP core and its embedded Native PHY IP core registers. To turn on the System Console and test the hardware design example, follow these steps:

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

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

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

Related Information

- 50GbE Design Example Registers on page 13 Register map for hardware design example.
- Analyzing and Debugging Designs with System Console

Design Example Description

The design example demonstrates the functions of the 50GbE core with transceiver interface compliant with the IEEE 802.3ba standard CAUI-4 specification. You can generate the design from the Example Design tab in the 50GbE 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 50GbE design in simulation and in hardware.

Related Information
Intel Arria® 10 50Gbps Ethernet IP Core User Guide

Design Example Behavior
The testbench sends traffic through the IP core, exercising the transmit side and receive side of the IP core. In the hardware design example, you can program the IP core in internal serial loopback mode and generate traffic on the transmit side that loops back through the receive side.

Design Example Interface Signals
The 50GbE testbench is self-contained and does not require you to drive any input signals.

Table 4. 50GbE Hardware Design Example Interface Signals

Signal	Direction	Comments
clk50	Input	Drive at 50 MHz. The intent is to dri ve this from a 50 Mhz oscillator on t he board.
clk_ref	Input	Drive at 644.53125 MHz.
cpu_resetn	Input	Resets the IP core. Active low. Driv es the global hard reset csr_reset_n to the IP core.
continued...		

Signal	Direction	Comments
tx_serial[1:0]	Output	Transceiver PHY output serial data.
rx_serial[1:0]	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 50GbE IP core signals and the interfaces to which they belong.

50GbE Design Example Registers

Table 5. 50GbE Hardware Design Example Register Map
Lists the memory mapped register ranges for the hardware design example. You access these registers with the reg_read and reg_write functions in the System Console.

Word Offset	Register Category
0x300–0x5FF	50GbE IP core registers.
0x4000–0x4C00	Arria 10 dynamic reconfiguration registers. Register base address is 0x4000 for Lane 0 and 0x4400 for Lane 1.

Related Information

- Testing the 50GbE Hardware Design Example on page 11 System Console commands to access the IP core and Native PHY registers.
- 50GbE Control and Status Register Descriptions Describes the 50GbE IP core registers.

Document Revision History

Table 6. 50G Ethernet Design Example User Guide Revision History

Date	Release	Changes
2019.04.03	17.0	Added the command to run Xcelium simulations.
2017.11.08	17.0	Added link to KDB Answer that provides workaround for potential jitter on Intel Arria® 10 devices due to cascading ATX PLLs in the IP core. Refer to **Generating the Design Example** on page 7 and **Compiling and Configuring the Design Example in Hardware** on page 10. This design example user guide has not been updated to reflect *Note:* minor changes in design generation in Intel Quartus Prime releases later than the Intel Quartus Prime software release v17.0.
2017.05.08	17.0	Initial public release.

Documents / Resources

	**intel 50G Ethernet Design Example** [pdf] User Guide 50G Ethernet Design Example, 50G, Ethernet Design Example, Design Example

References

- intel **How do I compensate for the jitter of PLL cascading or non-dedicated...**
- intel **1. 50GbE Quick Start Guide**
- intel **1. Datasheet**
- intel **1. Datasheet**
- intel **1. Datasheet**
- intel **Intel® Quartus® Prime Software User Guides**
- intel **Intel® Quartus® Prime Software User Guides**
- intel **Intel® Quartus® Prime Software User Guides**
- intel **Intel ISO 9001:2015 Registrations**