LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.10

User Guide

UG691 (v1.10) January 18, 2012

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/16/09	1.1	Initial Xilinx release.
12/02/09	1.2	Updates to tools and Wizard. Added GTH0 Settings in Chapter 3 and GTH1, GTH2, and GTH3 Settings in Chapter 3. Deleted “Using the ISE Simulator” section.
04/19/10	1.3	Updates to the tools and Wizard. Added Using the ISE Simulator in Chapter 4.
07/23/10	1.4	Updates to the tools and Wizard.
09/21/10	1.5	Wizard v1.5 release.
12/14/10	1.6	Updates to the tools and Wizard. Replaced 10GBASE-KR with 10GBASE-R. Incorporated references into Chapter 1, Introduction and deleted Appendix A: References. Reordered the sections in Chapter 4, Quick Start Example Design.

Date	Version	Revision
03/01/11	1.7	Wizard v1.7 release. Integrated the *LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard Data Sheet* (DS738) and *Getting Started Guide* (UG691). The title of the integrated document is *LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.7 User Guide* (UG691). Removed the Conventions section from the Preface. Revised the following chapters, mainly to incorporate the integration: • Chapter 1, Introduction: Added Features, Supported Devices, Provided with the Wizard, and Ordering Information. Removed the Additional Wizard Resources section and moved its content to Related Xilinx Documents. • Chapter 2, Installing the Wizard: Expanded Design Tools. • Chapter 3, Running the Wizard: Added Functional Overview and Figure 3-1, Structure of the GTH Transceiver Wrapper, Example Design, and Testbench and Figure 3-2, and Example Design—10GBASE-R Configuration. Updated Figure 3-6, Figure 3-7, Figure 3-8, Figure 3-9, Figure 3-10, and Figure 3-11. Added RX Off and TX Off options to Table 3-3. • Chapter 4, Quick Start Example Design: Expanded introductory description of Functional Simulation of the Example Design. • Chapter 5, Detailed Example Design, modified name of Figure 5-1. Minor typographical edits.
04/08/11	1.7.1	Updated Legal disclaimer. Chapter 1, Introduction: Added Virtex-6 Silicon Revision Support for supported silicon revision CES (ES2.0). Chapter 3, Running the Wizard: In GTH Placement and Clocking, clarified the description of Page 1 of the IP GUI and updated Figure 3-7.
04/08/11	1.8	Chapter 1, Introduction: Updated Virtex-6 Silicon Revision Support. Chapter 3, Running the Wizard: In GTH Placement and Clocking, updated Figure 3-7.
06/22/11	1.8.1	Updated Wizard v1.8 for release. In Chapter 1, Introduction, revised About the Wizard, Features, Supported Devices, Provided with the Wizard, Recommended Design Experience, and Related Xilinx Documents. In Chapter 2, Installing the Wizard, revised Tools and System Requirements and Installing the Wizard. In Chapter 3, Running the Wizard, revised title and content of Structure of the GTH Transceiver Wrapper, Example Design, and Testbench. Renamed Figure 3-2, revised Example Design—10GBASE-R Configuration, and Transceiver to GTH Transceiver Placement and Clocking. In Chapter 4, Quick Start Example Design, revised source directories in Table 4-1. In Chapter 5, Detailed Example Design, added gth<n>_rom_init_tx and gth<n>_rom_init_rx to Table 5-4. Added Synplify, ChipScope Pro, and PlanAhead software support to Table 5-5. Added Window scripts for running Xilinx ISE software and Cadence Incisive Enterprise Simulator (IES). Added Linux script for running Xilinx ISE software to Table 5-8.

Date	Version	Revision
10/19/11	1.9	Updated Wizard v1.9 for release. Chapter 1: Updated About the Wizard. Chapter 3: Updated Functional Overview. Chapter 4: Added Netlist Simulation of the Example Design, including Table 4-3. Chapter 5: Added `<component_name>_pulse_synchronizer.v[hd]` to Table 5-1. Added `demo_tb_imp.v[hd]` to Table 5-7. Added simulation/netlist, including Table 5-9.
01/18/12	1.10	Updated Wizard v1.10 for release. Replaced 13.3 with 13.4 throughout. Chapter 2: Updated VCS MX version in Simulation.

Table of Contents

Chapter 4: Quick Start Example Design

Chapter 5: Detailed Example Design

About This Guide

This guide describes the Virtex®-6 FPGA GTH Transceiver Wizard (hereinafter called the Wizard).

Guide Contents

This guide contains the following chapters:

- Chapter 1, Introduction describes the Wizard and related information, including additional resources, technical support, and submitting feedback to Xilinx.
- Chapter 2, Installing the Wizard provides information about installing the Wizard.
- Chapter 3, Running the Wizard provides an overview of the Wizard and a step-by-step tutorial to generate a sample GTH transceiver wrapper with the CORE Generator™ tool.
- Chapter 4, Quick Start Example Design introduces the example design that is included with the GTH transceiver wrappers. The example design demonstrates how to use the wrappers and demonstrates some of the key features of the GTH transceiver.
- Chapter 5, Detailed Example Design provides detailed information about the example design, including a description of files and the directory structure generated by the CORE Generator tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration testbench.

Additional Resources

To find additional documentation, see the Xilinx website at:

http://www.xilinx.com/support/documentation/index.htm.

To search the Answer Database of silicon, software, and IP questions and answers, or to create a technical support WebCase, see the Xilinx website at:

http://www.xilinx.com/support/mysupport.htm.

Chapter 1

Introduction

This chapter introduces the Virtex®-6 FPGA GTH Transceiver Wizard and provides related information, including additional resources, technical support, and submitting feedback to Xilinx.

About the Wizard

The Wizard automates the task of creating HDL wrappers to configure the high-speed serial GTH transceivers in Virtex-6 devices.

The Virtex-6 FPGA GTH Wizard is a Xilinx® CORE Generator™ tool that generates Verilog or VHDL source code to configure the GTH transceiver primitives in Virtex-6 FPGAs.

The menu-driven interface allows one or more GTH transceivers to be configured using predefined templates for popular industry standards, or by using custom templates, to support a wide variety of custom protocols. The Wizard produces a wrapper, an example design, and a testbench for rapid integration and verification of the serial interface with your custom function.

The wrapper generated by the Wizard configures and instantiates one or more GTH transceivers for user applications (Figure 1-1).

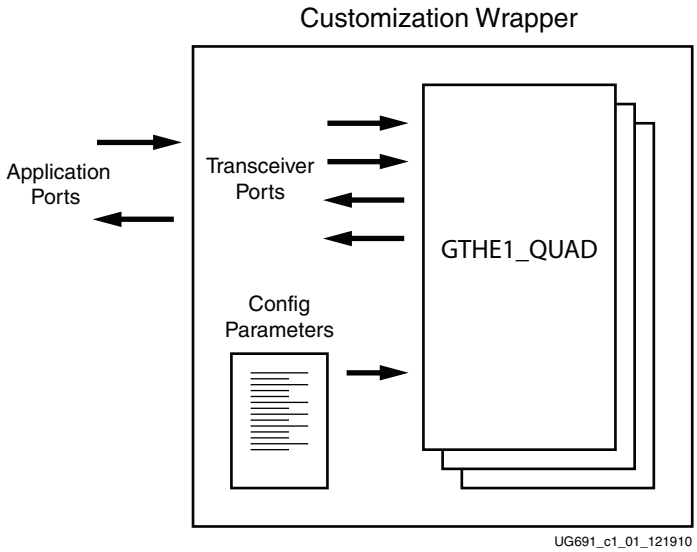


Figure 1-1: **GTH Transceiver Wizard Wrapper**

The Wizard can be accessed from the ISE® software CORE Generator tool. For information about system requirements and installation, see Chapter 2, Installing the Wizard.

For the latest information on the Wizard, refer to the Architecture Wizards product information page:
http://www.xilinx.com/products/design_resources/conn_central/solution_kits/wizards

For documentation, see the Virtex-6 FPGA GTH Transceiver Wizard page:

http://www.xilinx.com/support/documentation/ipfpgafeaturedesign_iointerface_v6gthwizard.htm

Virtex-6 Silicon Revision Support

Version 1.8 (and later versions) of this Wizard support production Virtex-6 HXT device silicon only. Please use version 1.7 of the Wizard to configure GTH transceiver settings for CES (ES 2.0) devices.

Features

The Wizard has these features:

- Creates customized HDL wrappers to configure Virtex-6 FPGA GTH transceivers
  - Industry standard protocols supported by predefined templates include 10GBASE-R, 100 Gb Attachment Unit Interface (CAUI), OC-48, OC-192, OTU-1, OTU-2, OTU-4, 40 Gb Attachment Unit Interface (XLAUI), Interlaken, and Aurora 64B/66B
  - Custom protocols can be specified using the **Start from Scratch** option in the GUI
- Automatically configures GTH transceiver analog settings for the Virtex-6 FPGA
- Option to specify 8B/10B, 64B/66B, or 64B/67B encoding/decoding
- Includes an example design with a companion testbench as well as implementation and simulation scripts

Supported Devices

The Wizard supports the Virtex-6 HXT family. For a complete listing of supported devices, see XTP025, *IP Release Notes Guide* for this Wizard. For more information on the Virtex-6 FPGAs, see DS150, *Virtex-6 Family Overview*.

Provided with the Wizard

The following are provided with the Wizard:

- Documentation: This user guide
- Design Files: Verilog and VHDL
- Example Design: Verilog and VHDL
- Testbench: Verilog and VHDL
- Constraints File: Synthesis constraints file
- Simulation Model: Verilog and VHDL

Recommended Design Experience

The Wizard is a fully verified solution that helps automate the task of defining parameter settings for Virtex-6 FPGA GTH transceivers. The additional challenge associated with implementing a complete design depends on the configuration and required functionality of the application. For best results, previous experience building high-performance, pipelined FPGA designs using Xilinx implementation software and user constraints files (UCF) is recommended.

For those with less experience, Xilinx offers various training classes to help with various aspects of designing with Xilinx FPGAs. These include classes on such topics as designing for performance and designing with multi-gigabit serial I/O. For more information, see http://www.xilinx.com/training.

Xilinx sales representatives can provide a closer review and estimation of specific design requirements.

Related Xilinx Documents

For detailed information and updates about the Wizard, see the following:

- UG691, *LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.10 User Guide*
- XTP025, *IP Release Notes Guide* for the Wizard

Prior to generating the Wizard, users should be familiar with the following:

- DS150: *Virtex-6 Family Overview*
- UG371: *Virtex-6 FPGA GTH Transceivers User Guide*
- ISE software documentation: www.xilinx.com/ise

Technical Support

For technical support, go to www.xilinx.com/support. Questions are routed to a team of engineers with expertise using this Wizard.

Xilinx provides technical support for use of this product as described in this guide. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow these guidelines.

Ordering Information

The Wizard is provided free of charge under the terms of the Xilinx End User License Agreement. The Wizard can be generated by the ISE software CORE Generator tool 13.4 or higher, which is a standard component of the ISE Design Suite. For more information, visit the Architecture Wizards web page. Information about additional LogiCORE modules is available at the IP Center. For pricing and availability of other LogiCORE modules and software, contact a local Xilinx sales representative.

Feedback

Xilinx welcomes comments and suggestions about the Wizard and the accompanying documentation.

Wizard

For comments or suggestions about the Wizard, submit a WebCase from www.xilinx.com/support. (Registration is required to log in to WebCase.) Be sure to include the following information:

- Product name
- Wizard version number
- List of parameter settings
- Explanation of any comments, including whether the case is requesting an *enhancement* (improvement) or reporting a *defect* (something is not working correctly)

Document

For comments or suggestions about this document, submit a WebCase from www.xilinx.com/support. (Registration is required to log in to WebCase.) Be sure to include the following information:

- Document title
- Document number
- Page number(s) to direct applicable comments
- Explanation of any comments, including whether the case is requesting an *enhancement* (improvement) or reporting a *defect* (something is not working correctly)

XILINX®

Chapter 2

Installing the Wizard

This chapter provides instructions to install the Virtex®-6 FPGA GTH Transceiver Wizard in the ISE® Design Suite CORE Generator™ tool.

Tools and System Requirements

Operating Systems

For a list of system requirements, see the *ISE Design Suite 13: Release Notes Guide*.

Design Tools

Design Entry

- ISE Design Suite CORE Generator software 13.4
- PlanAhead™ software 13.4

Simulation

- ISE Simulator (ISim) software 13.4
- Mentor Graphics ModelSim 6.6d
- Cadence Incisive Enterprise Simulator (IES) 10.2
- Synopsys Verilog Compiler Simulator (VCS) and VCS MX E-2011.03

See XTP025, *IP Release Notes Guide* for the Wizard for the required service pack. ISE software service packs can be downloaded from http://www.xilinx.com/support/download.htm.

Synthesis

- XST 13.4
- Synopsys Synplify Pro E-2011.03-SP2

Before You Begin

Before installing the Wizard, you must have a MySupport account and the ISE 13.4 software installed on your system. If you already have an account and have the software installed, go to Installing the Wizard, otherwise do the following:

1. Click **Login** at the top of the Xilinx home page then follow the onscreen instructions to create a MySupport account.
2. Install the ISE 13.4 software.

   For the software installation instructions, see the ISE Design Suite Release Notes and Installation Guide available in ISE Software Documentation.

Installing the Wizard

The Wizard is included with the ISE 13.4 software. Follow the ISE software 13.4 installation instructions in the ISE Installation and Release Notes available at http://www.xilinx.com/support/documentation under the Design Tools tab.

Verifying Your Installation

Use the following procedure to verify a successful installation of the Wizard in the CORE Generator tool.

1. Start the CORE Generator tool.
2. The IP core functional categories appear at the left side of the window (Figure 2-1).

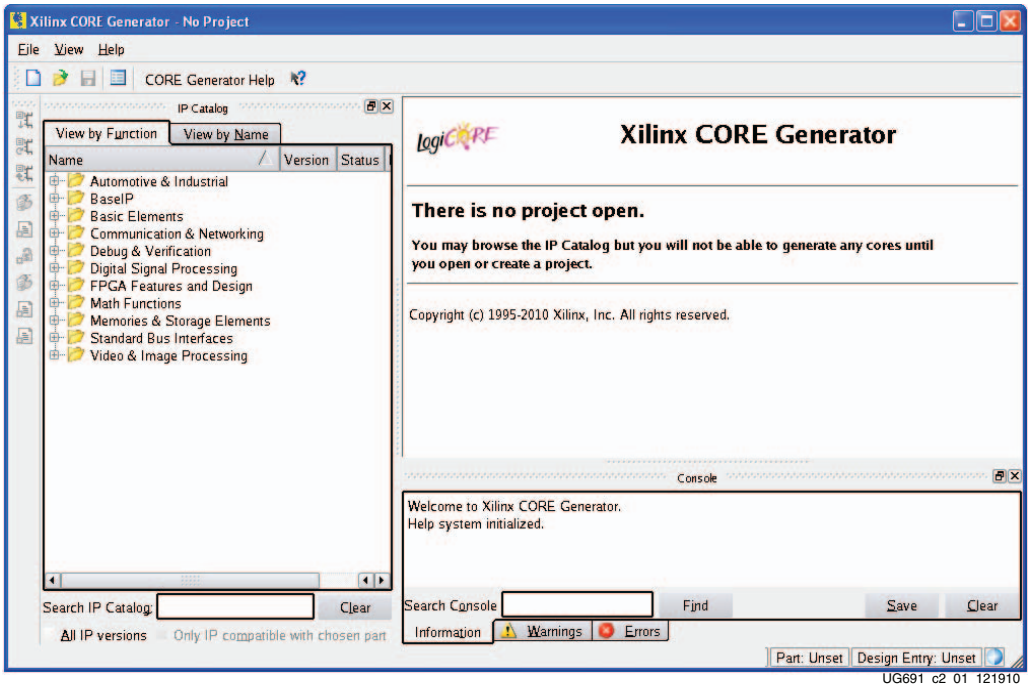


Figure 2-1: **CORE Generator Window**

3. Click to expand or collapse the view of individual functional categories, or click the **View by Name** tab at the top of the list to see an alphabetical list of all cores in all categories.
4. Determine if the installation was successful by verifying that Virtex-6 FPGA GTH Transceiver Wizard 1.10 appears at the following location in the Functional Categories list: `/FPGA Features and Design/IO Interfaces`

XILINX®

Chapter 3

Running the Wizard

Overview

This chapter provides a step-by-step procedure for generating a Virtex®-6 FPGA GTH transceiver wrapper, implementing the wrapper in hardware using the accompanying example design, and simulating the wrapper with the provided example testbench.

Note: The screen captures in this chapter are conceptual representatives of their subjects and provide general information only. For the latest information, see the CORE Generator™ tool.

Functional Overview

Figure 3-1, page 18 shows the steps required to configure GTH transceivers using the Wizard. Start the CORE Generator software and select the Virtex-6 FPGA GTH Transceiver Wizard, then follow the chart to configure the transceivers and generate a wrapper that includes an accompanying example design.

- To use an existing template with no changes, simply select the desired template, then click **Generate**.
- To modify a standard template or start from scratch, proceed through the Wizard and adjust the settings as needed.

See Configuring and Generating the Wrapper, page 23 for details on the various transceiver features and parameters available.

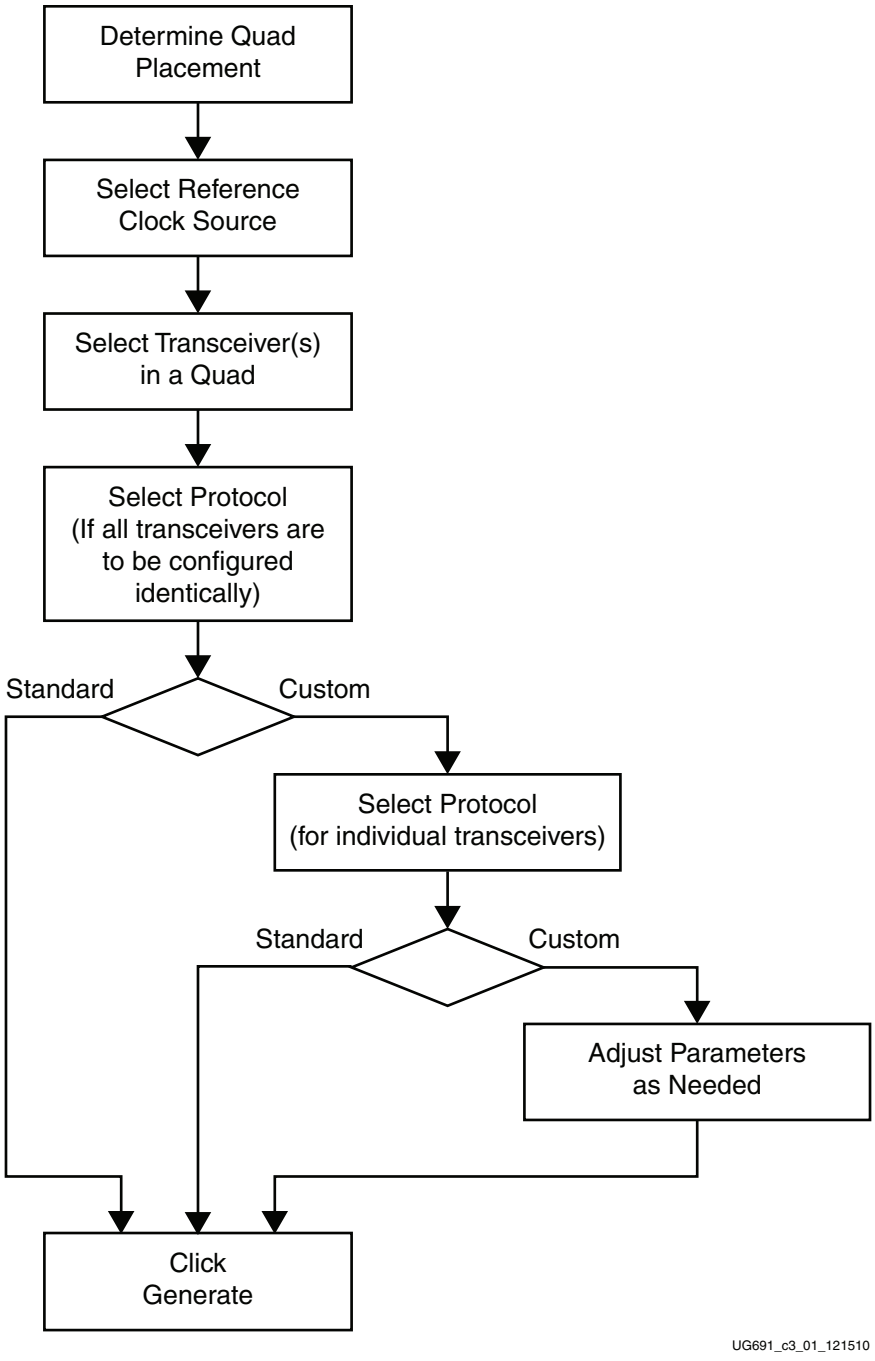


Figure 3-1: **Wizard Configuration Steps**

Structure of the GTH Transceiver Wrapper, Example Design, and Testbench

Figure 3-2 shows the relationship of the GTH transceiver wrapper, example design, and testbench files generated by the Wizard. For details, see Example Design Description, page 41.

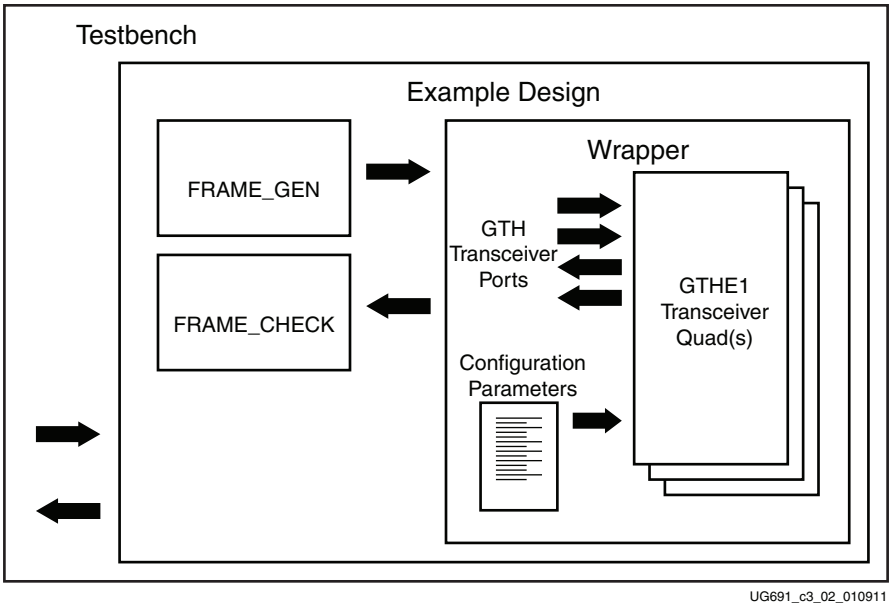


Figure 3-2: **Structure of the GTH Transceiver Wrapper, Example Design, and Testbench**

These files are generated by the Wizard to illustrate the components needed to simulate the configured transceiver:

- GTH transceiver wrapper, which includes:
  - Specific gigabit transceiver configuration parameters set using the Wizard.
  - GTHE1 transceiver Quad primitive(s) selected using the Wizard.
- Example design demonstrating the modules required to simulate the wrapper. These include:
  - FRAME_GEN module: Generates a user-definable data stream for simulation analysis.
  - FRAME_CHECK module: Tests for correct transmission of data stream for simulation analysis.
- Testbench:
  - Top-level testbench demonstrating how to stimulate the design.

Example Design—10GBASE-R Configuration

The example design covered in this section is a wrapper that configures a group of GTH transceivers for use in a 10GBASE-R application. Guidelines are also given for incorporating the wrapper in a design and for the expected behavior in operation. For detailed information, see Chapter 4, Quick Start Example Design.

The 10GBASE-R example consists of the following components:

- A single GTH transceiver wrapper implementing a 1-lane 10GBASE-R port using one GTH transceiver
- A demonstration testbench to drive the example design in simulation
- An example design providing clock signals and connecting an instance of the 10GBASE-R wrapper with modules to drive and monitor the wrapper in hardware, including optional ChipScope™ Pro tool support
- Scripts to synthesize and simulate the example design

The Wizard example design has been tested with Synplify Pro E-2011.03-SP2 and XST 13.4 for synthesis and ModelSim 6.6d for simulation.

Figure 3-3 shows a block diagram of the default 10GBASE-R example design.

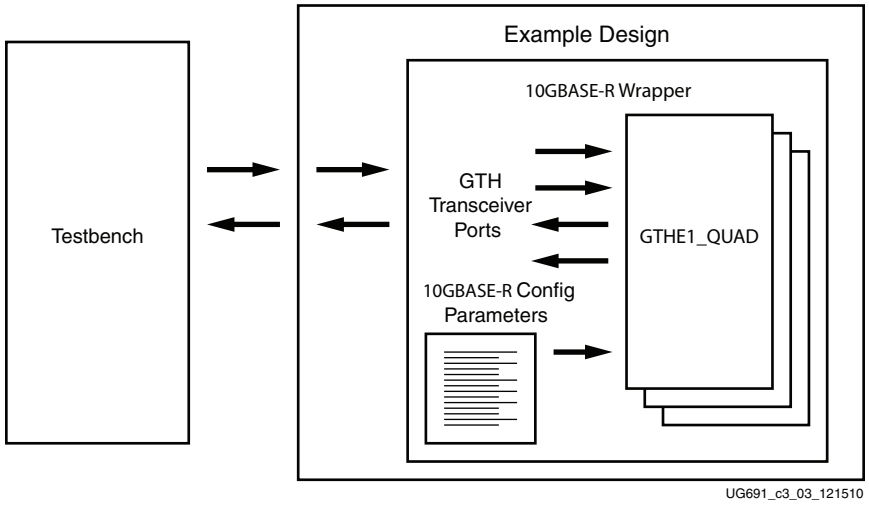


Figure 3-3: **Example Design and Testbench—10GBASE-R Configuration**

Setting Up the Project

Before generating the example design, set up the project as described in Creating a Directory and Setting the Project Options.

Creating a Directory

To set up the example project, first create a directory using the following steps:

1. Change directory to the desired location. This example uses the following location and directory name:

   `/Projects/10gbaser_example`

2. Start the CORE Generator™ software.

   For help starting and using the CORE Generator software, see *CORE Generator Help*, available in ISE® software documentation.

3. Choose **File > New Project** (Figure 3-4).
4. Change the name of the CGP file (optional).
5. Click **Save**.

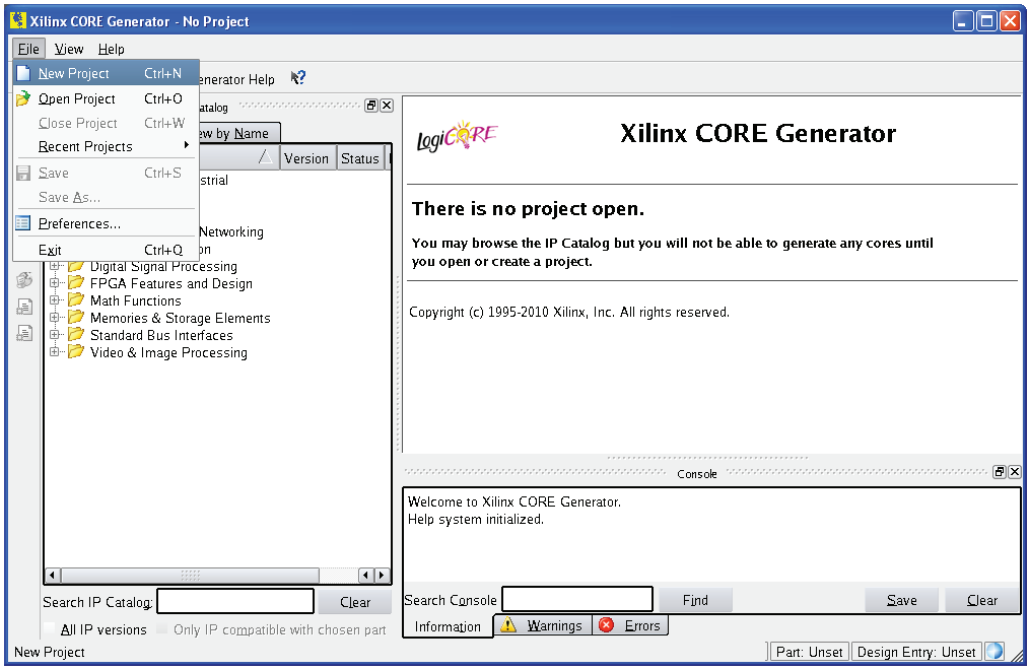


Figure 3-4: **Starting a New Project**

Setting the Project Options

Set the project options using the following steps:

1. Click **Part** in the option tree.
2. Select **Virtex6** from the Family list.
3. Select a device from the Device list that supports GTH transceivers.
4. Select an appropriate package from the Package list. This example uses the XC6VHX380T device (see Figure 3-5).

   Note: If an unsupported silicon family is selected, the Virtex-6 FPGA GTH Transceiver Wizard remains light grey in the taxonomy tree and cannot be customized. Only devices containing Virtex-6 GTH transceivers are supported by the Wizard. See DS150, *Virtex-6 Family Overview* for a list of devices containing GTH transceivers.
5. Click **Generation** in the option tree and select either Verilog or VHDL as the output language.
6. Click **OK**.

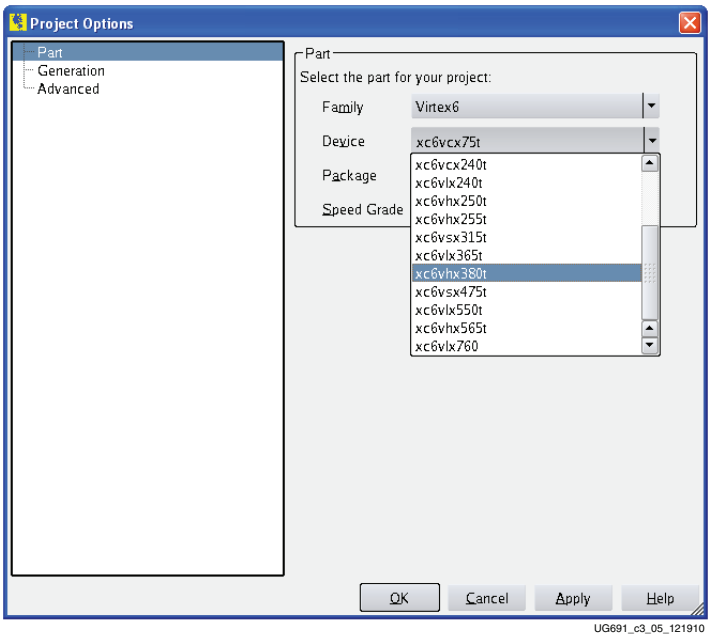


Figure 3-5: **Target Architecture Setting**

Configuring and Generating the Wrapper

This section provides instructions for generating an example GTH transceiver wrapper using the default values. The wrapper, associated example design, and supporting files are generated in the project directory. For additional details about the example design files and directories see Chapter 5, Detailed Example Design.

1. Locate the Virtex-6 FPGA GTH Transceiver Wizard 1.10 in the taxonomy tree under: `/FPGA Features & Design/IO Interfaces`. (See Figure 3-6)
2. Double-click **Virtex-6 FPGA GTH Transceiver Wizard 1.10** to launch the Wizard.

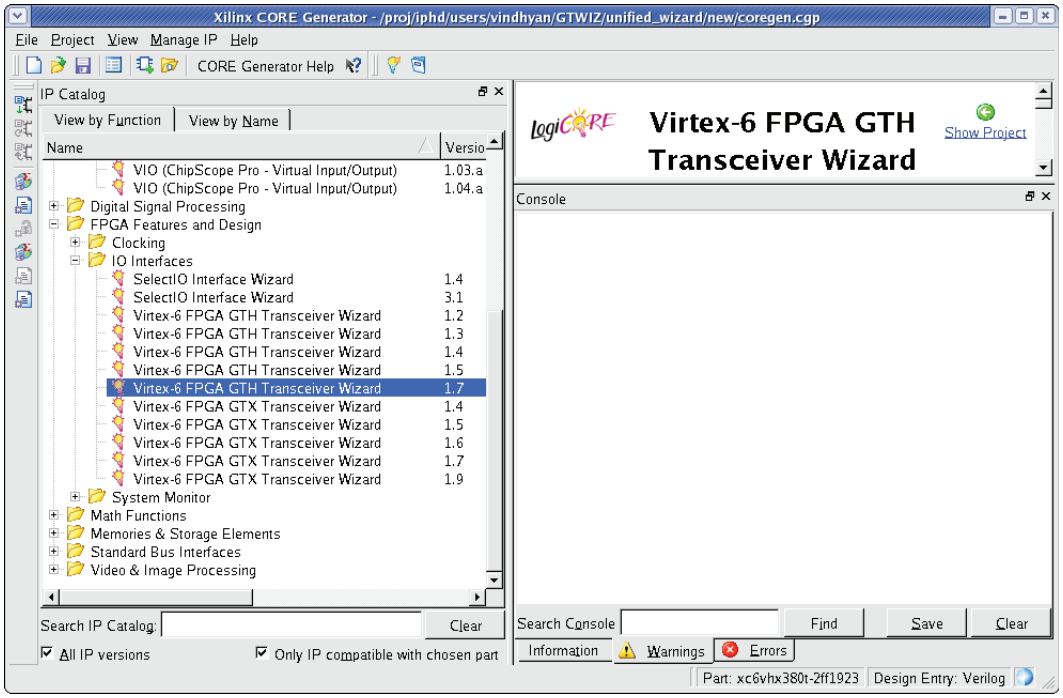


Figure 3-6: **Locating the GTH Transceiver Wizard**

GTH Transceiver Placement and Clocking

Page 1 of the Wizard (Figure 3-7, page 24) allows you to specify the component name, placement of the GTHE1_QUAD, reference clock source, target line rate, reference clock frequency, and DRP clock frequency. In addition, a drop-down menu on this page also provides you with the option of selecting a pre-configured protocol template to define the transceiver settings instead of stepping through the entire GUI and programming each transceiver setting individually

1. In the Component Name filed, enter a name for the wrapper instance. This example uses the name tengbaser_wrapper.
2. Select the GTHE1_QUAD and reference clock source required for the target design. This example uses GTHE1_QUAD_X0Y0 and enables only one GTH transceiver (GTH0).
3. From the Protocol Template list, select the desired protocol template. The 10GBASE-R example uses the 10GBASE-R protocol template.

4. After reviewing the settings, click **Generate** to generate the wrapper or click **Next** to configure an individual transceiver.

The number of available GTHE1_QUAD appearing on this page depends on the selected target device and package. The 10GBASE-R example design uses one GTH transceiver from one GTHE1_QUAD. Table 3-1, page 25 describes the GTHE1_QUAD selection and reference clock options, Table 3-2, page 25 describes the reference clock source options, and Table 3-3, page 25 describes the shared settings options.

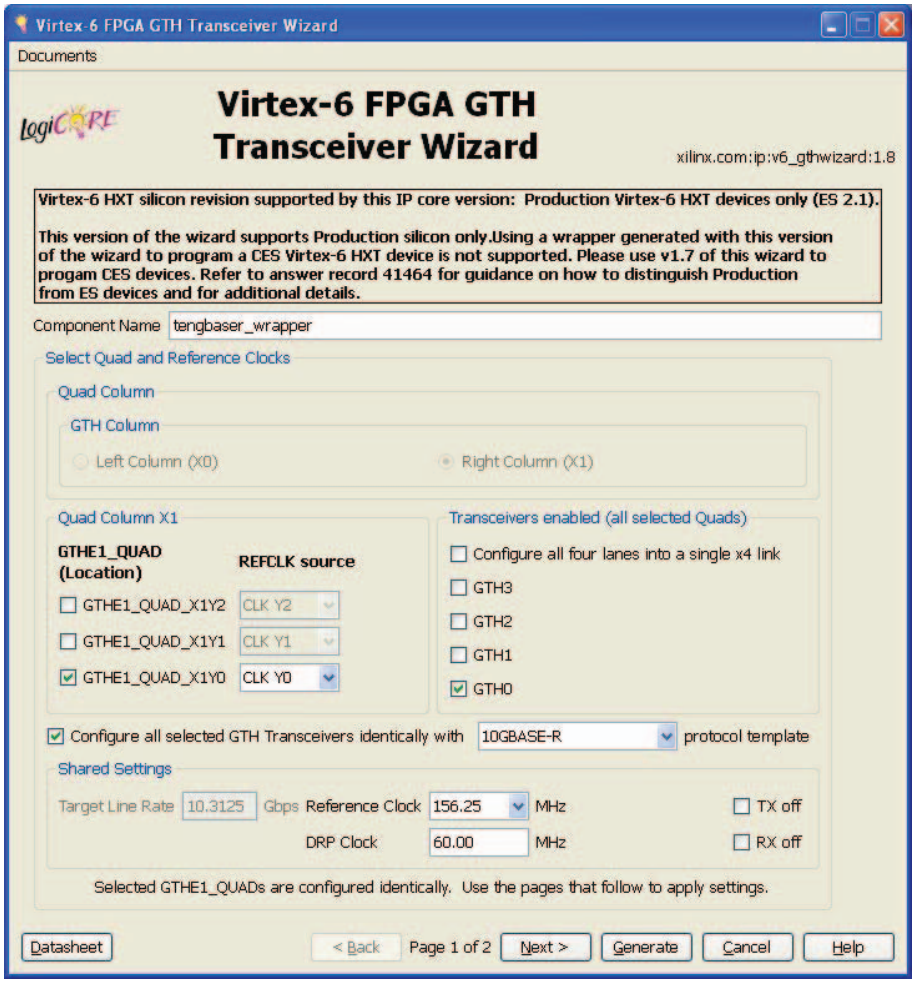


Figure 3-7: **GTH Transceiver Placement and Clocking—Page 1 of 2**

Table 3-1: **Select Quad and Reference Clock**

Option	Description
GTH Column	Toggles between displaying the GTHE1_QUAD on the left column (X0) and right column (X1).
GTHE1_QUAD	Select the individual number of GTHE1_QUAD by location to be used in the target design.
REFCLK Source	Determines the source for the reference clock signal provided to each selected GTHE1_QUAD. The 10GBASE-R example uses the reference clock from the differential input pins of GTHE1_QUAD_X0Y0 (CLK Y0).
GTH Transceivers	Select the individual GTH transceivers by location to be used in the target design. Each GTHE1_QUAD contains four GTH transceivers.
Configure All Four Lanes into Single X4	Select this option if lanes 0, 1, 2, and 3 need to be configured into a single X4 link.
Configure All Selected GTH Transceivers Identically	Check this box to configure selected GTH transceivers identically. Page 3, 4, and 5 of the GUI will not appear if this box is checked.

Table 3-2: **Reference Clock Source Options**

REFCLK Source	Description
CLK Y0	Dedicated GTH transceiver reference clock for GTHE1_QUAD_X[*m*]Y0 (1)
CLK Y1	Dedicated GTH transceiver reference clock for GTHE1_QUAD_X[*m*]Y1
CLK Y2	Dedicated GTH transceiver reference clock for GTHE1_QUAD_X[*m*]Y2

1. [*m*] = 0 if GTH column is set to left column and [*m*] = 1 if GTH column is set to right column.

Table 3-3: **Shared Settings**

Option	Description
Target Line Rate	Line rate in Gb/s desired for the target design. The 10GBASE-R example uses 10.3125 Gb/s.
Reference Clock	Select from the list the optimal reference clock frequency to be provided by the application. The 10GBASE-R example uses 156.25 MHz.
DRP Clock	DRP clock frequency in MHz desired for target design. The 10GBASE-R example uses 60 MHz.
TX Off	To use only receiver of the transceiver.
RX Off	To use only transmitter of the transceiver.

GTH0 Settings

Page 2 of the Wizard sets the line rate, encoding, and fabric data width for GTH0 along with TX and RX driver settings. Optional port selection is also provided.

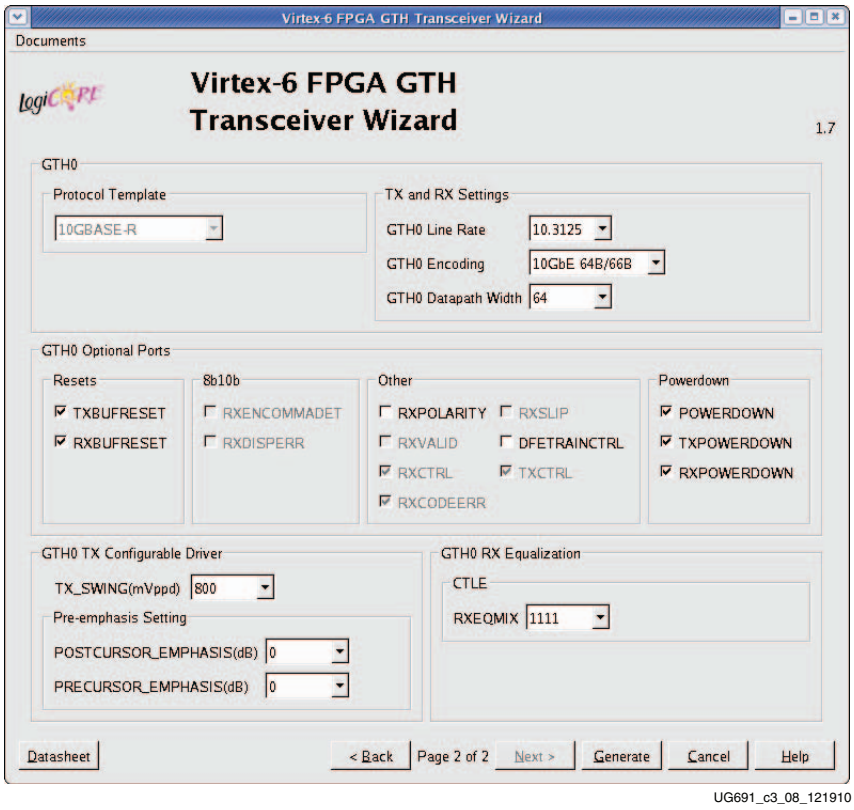


Figure 3-8: **GTH0 Settings—Page 2 of 2**

Table 3-4: **TX and RX Setting**

Option	Description
GTH0 Line Rate	Line rate in Gb/s for GTH0 in the targeted design. Value can be Target Line Rate, 1/2 of Target Line Rate, 1/4th of Target Line Rate or 1/8th of Target Line Rate selected on page 1. 10GBASE-R example uses 10.3125 Gb/s.
GTH0 Encoding	Encoding standard to be used for GTH0 transceiver. Value can be None, 8B/10B or 10GbE_64B/66B. The 10GBASE-R example uses 10GbE_64B/66B encoding.
GTH0 Datapath Width	Fabric data path width in bits. Value depends on encoding and line rate. 10GBASE-R example uses 64-bit data path.

Table 3-5: **GTH0 Optional Ports**

Option	Description
TXBUFRESET	Active-High reset signal for TX buffer inside the TX data converter
RXBUFRESET	Active-High reset signal for RX buffer inside the RX data converter.
RXENCOMMADET	Active-High comma detection enable signal. This option is available only if 8B/10B encoding is selected.
RXDISPERR	Used only in 8B/10B mode. The 8-bit port indicates disparity error on RX data bus.
RXCODEERR	This is an 8-bit port. The output indicates an error occurred on RX data.
RXPOLARITY	The 1-bit port is used to invert polarity of RX data.
RXVALID	The status port indicates which bytes are valid in RX data. This option is available only in 8B/10B mode.
RXCTRL	This output either indicates status of RX data or used as an extension of RX data depending on encoding. This is an 8-bit port.
RXSLIP	This port is used in raw mode for the barrel shifter operation to advance the bit alignment position.
DFETRAINCTRL	This is a 1-bit port and controls DFE training sequence.
TXCTRL	This input either indicates control of TX data or they are used as an extension of TX data depending on the encoding selected. This is an 8-bit port.
POWERDOWN	This control signal powers off the corresponding lane. It is used to place individual lanes in a low power state. This port is used on a per-lane basis even when multiple lanes are configured as a single logical link.
TXPOWERDOWN	This control signal requests the transmitter power state: • `00`: Normal operation • `10`: Power off transmitter logic This port must always be set to `2'b10` during initialization and when GTHRESET is asserted. When the lanes within a Quad are configured as multilane links, the port from the lowest numbered lane of the link is valid.
RXPOWERDOWN	This control signal requests the receiver power state: • `00`: Normal operation. • `10`: Power off receiver logic This port must always be set to `2'b10` during initialization and when GTHRESET is asserted. When the lanes within a Quad are configured as multilane links, the port from the lowest numbered lane of the link is valid.

Table 3-6: **GTH0 TX Configurable Driver**

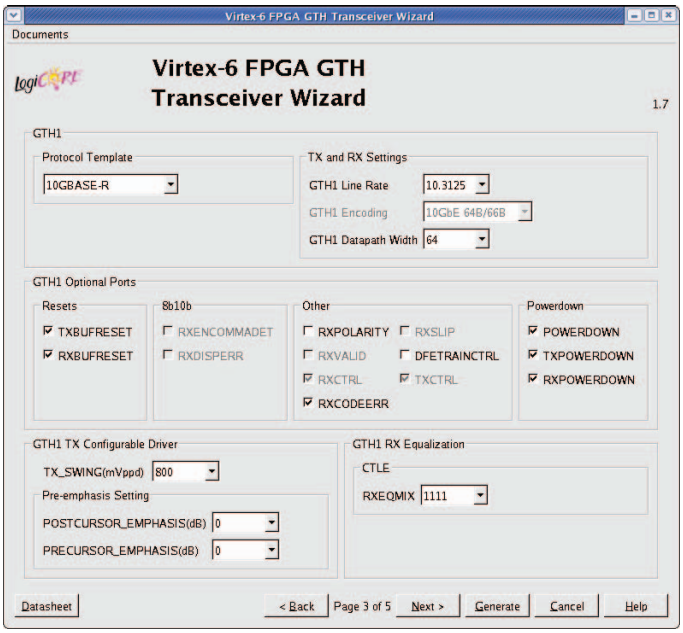
Option Description	
TX_SWING	Decimal value that controls the differential voltage swing.
POSTCURSOR_EMPHASIS	Post-cursor emphasis value in dB.
PRECURSOR_EMPHASIS	Precursor emphasis value in dB.

Table 3-7: **: GTH0 RX Equalization**

Option	Description
RXEQMIX	4-bit value that controls receive equalization.

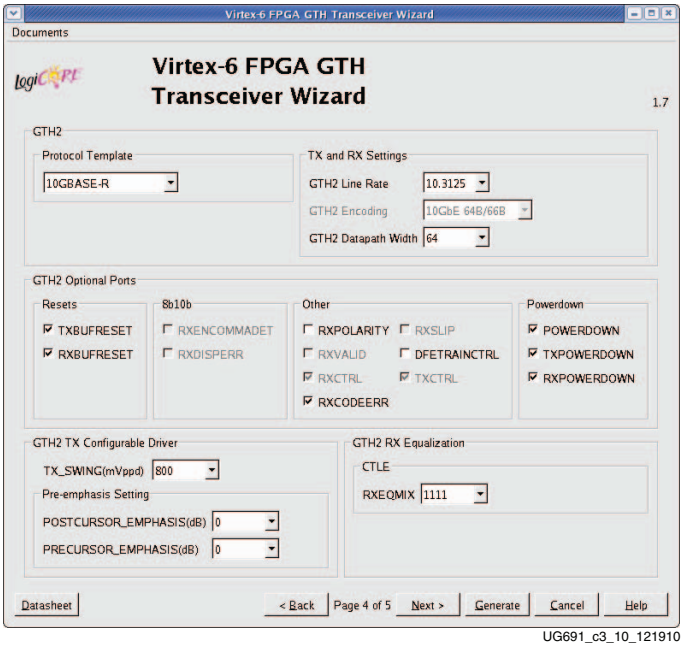
GTH1, GTH2, and GTH3 Settings

Figure 3-9, Figure 3-10, and Figure 3-11, page 29 are visible based on the GTH1, GTH2, and GTH3 selections on Page 1 (Figure 3-7, page 24). For the description of the options, see GTH0 Settings, page 26.



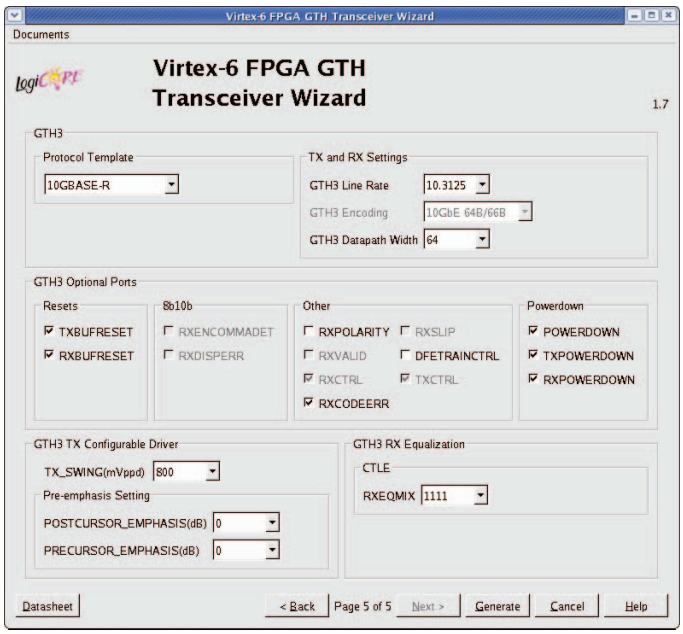
UG691_c3_09_121910

Figure 3-9: **GTH1 Settings—Page 3 of 5**



UG691_c3_10_121910

Figure 3-10: **GTH2 Settings—Page 4 of 5**



UG691_c3_11_121910

Figure 3-11: **GTH3 Settings—Page 5 of 5**

XILINX®

Chapter 4

Quick Start Example Design

Overview

This chapter introduces the example design that is included with the Virtex®-6 FPGA GTH transceiver wrappers. The example design demonstrates how to use the wrappers and demonstrates some of the key features of the GTH transceiver. For detailed information about the example design, see Chapter 5, Detailed Example Design.

Functional Simulation of the Example Design

The Virtex-6 FPGA GTH Transceiver Wizard (hereinafter called the Wizard) provides a quick way to simulate and observe the behavior of the wrapper using the provided example design and script files.

To simulate simplex designs, the SIMPLEX_PARTNER environment variable should be set to the path of the complementary core generated to test the simplex design. For example, if a design is generated with RX OFF, a simplex partner design with RX enabled is needed to simulate the DUT. The SIMPLEX_PARTNER environment variable should be set to the path of the RX enabled design. The name of the simplex partner should be the same as the name of the DUT with a prefix of tx or rx as applicable. In the current example, the name of the simplex partner design would be prefixed with rx.

Using ModelSim

Prior to simulating the wrapper with ModelSim, the functional (gate-level) simulation models must be generated. All source files in the following directories must be compiled to a single library as shown in Table 4-1. See the *Synthesis and Simulation Design Guide* for ISE® 13.4, available in the ISE software documentation, for instructions on how to compile ISE simulation libraries.

Table 4-1: **Required ModelSim Simulation Libraries**

HDL	Library	Source Directories
Verilog	UNISIMS_VER	<*Xilinx dir*>/verilog/src/unisims <*Xilinx dir*>/secureip/mti
VHDL	UNISIM	<*Xilinx dir*>/vhdl/src/unisims/primitive <*Xilinx dir*>/secureip/mti

The Wizard provides a command line script for use within ModelSim. To run a VHDL or Verilog ModelSim simulation of the wrapper, use the following instructions:

1. Launch the ModelSim simulator and set the current directory to:

   `<project_directory>/<component_name>/simulation/functional`

2. Set the MTI_LIBS variable:

```
modelsim> setenv MTI_LIBS <path to compiled libraries>
```

3. Launch the simulation script:

```
modelsim> do simulate_mti.do
```

The ModelSim script compiles the example design and testbench and adds the relevant signals to the wave window.

Using the ISE Simulator

When using the ISE Simulator (ISim), the required Xilinx simulation device libraries are precompiled, and are updated automatically when service packs and IP updates are installed. There is no need to run CompXlib to compile libraries, or to manually download updated libraries.

Table 4-2: **Required ISim Simulation Libraries**

HDL	Library	Source Directories
Verilog	UNISIMS_VER	<*Xilinx dir*>/verilog/hdp/<*OS*>/unisims_ver
VHDL	UNISIM	<*Xilinx dir*>/vhdl/hdp/<*OS*>/unisim

Note: OS refers to the following operating systems: lin, lin64, nt, nt64.

The Wizard also generates a perl script for use with ISim. To run a VHDL or Verilog simulation of the wrapper, use the following instructions:

1. Set the current directory to

```
<project_directory>/<component_name>/simulation/functional
```

2. Launch the simulation script:

```
prompt> simulate_isim.sh
```

The ISim script compiles the example design and testbench, and adds the relevant signals to the wave window.

Implementing the Example Design

When all of the parameters are set as desired, clicking **Generate** creates a directory structure under the provided Component Name. Wrapper generation proceeds and the generated output populates the appropriate subdirectories.

The directory structure for the 10GBASE-R example is provided in Chapter 5, Detailed Example Design.

After wrapper generation is complete, the results can be tested in hardware. The provided example design incorporates the wrapper and additional blocks allowing the wrapper to be driven and monitored in hardware. The generated output also includes several scripts to assist in running the software.

From the command prompt, navigate to the project directory and type the following:

For Windows

```
> cd tengbaser_wrapper\implement
> implement.bat
```

For Linux

```
% cd tengbaser_wrapper/implement
% implement.sh
```

Note: Substitute *Component Name* string for `tengbaser_wrapper`.

These commands execute a script that synthesizes, builds, maps, places, and routes the example design and produces a bitmap file. The resulting files are placed in the implement/results directory.

Netlist Simulation of the Example Design

The Virtex-6 FPGA GTH Transceiver Wizard provides a script to perform simulations on the routed netlist of the example design.

Note: Timing checks are disabled.

Using ModelSim

Prior to performing netlist simulation with ModelSim, the generated design should successfully pass through implementation. All source files in the following directories must be compiled to a single library, as shown in Table 4-3. See the *Synthesis and Simulation Design Guide* for ISE 13.4 available in the ISE software documentation for instructions on how to compile ISE simulation libraries.

Table 4-3: **Required ModelSim Timing Simulation Libraries**

HDL	Library	Source Directories
Verilog	SIMPRIMS_VER	<*Xilinx dir*>/verilog/src/simprims <*Xilinx dir*>/secureip/mti
VHDL	SIMPRIM	<*Xilinx dir*>/vhdl/src/simprims/primitive <*Xilinx dir*>/secureip/mti

The Wizard provides a command line script for use within ModelSim. To run a VHDL or Verilog ModelSim simulation of the wrapper, use the following instructions:

1. Launch the ModelSim simulator and set the current directory to:

   ```
   <project_directory>/<component_name>/simulation/netlist
   ```

2. Set the MTI_LIBS variable:

   ```
   modelsim> setenv MTI_LIBS <path to compiled libraries>
   ```

3. Launch the simulation script:

   ```
   modelsim> do simulate_mti.do
   ```

The ModelSim script compiles and simulates the routed netlist of the example design and testbench.

XILINX®

Chapter 5

Detailed Example Design

This chapter provides detailed information about the example design, including a description of files and the directory structure generated by the CORE Generator™ tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration testbench.

Directory and File Structure

- **<project directory>**
  Top-level project directory; name is user-defined
  - <project directory>/<component name>
    Wizard release notes file
    - <component name>/doc
      Product documentation
    - <component name>/example design
      Verilog and VHDL design files
    - <component name>/implement
      Implementation script files
      - implement/results
        Results directory, created after implementation scripts are run, and contains implement script results
    - <component name>/simulation
      Simulation scripts
      - simulation/functional
        Functional simulation files
      - simulation/netlist
        Netlist simulation files

Directory and File Contents

The Virtex®-6 FPGA GTH Transceiver Wizard directories and their associated files are defined in the following sections.

<project directory>

The <project directory> contains all the CORE Generator tool's project files.

Table 5-1: **Project Directory**

Name	Description
<component_name>.v[hd]	Main GTH transceiver wrapper. Instantiates individual GTHE1_QUAD wrappers. For use in the target design.
<component_name>.[veo | vho]	GTH transceiver wrapper files instantiation templates. Includes templates for the GTH transceiver wrapper module, and the IBUFDS_GTHE1.
<component_name>.xco	Log file from CORE Generator tool describing which options were used to generate the GTH transceiver wrapper. An XCO file is generated by the CORE Generator tool for each Wizard wrapper that it creates in the current project directory. An XCO file can also be used as an input to the CORE Generator tool.
<component_name>_quad.v[hd]	Individual GTHE1_QUAD wrapper to be instantiated in the main GTH transceiver wrapper. Instantiates GTHE1_QUAD with settings for the selected protocol.
<component_name>_gth_init.v[hd]	GTH transceiver initialization module to be instantiated in the GTHE1_QUAD wrapper.
<component_name>_gth_reset.v[hd	GTH transceiver reset module to be instantiated in the GTHE1_QUAD wrapper.
<component_name>_gth_rx_pcs_cdr_reset.v[hd]	GTH transceiver receive PCS and CDR reset module to instantiated in GTHE1_QUAD wrapper.
<component_name>_gth_tx_pcs_reset.v[hd]	GTH transceiver transmit PCS reset module to instantiated in GTHE1_QUAD wrapper.
<component_name>_pulse_synchronizer.v[hd]	A pulse synchronizer module to instantiate whenever there a pulse crosses from one clock domain to another.

Back to Top

<project directory>/<component name>

The <component name> directory contains the README file provided with the Wizard, which might include last-minute changes and updates.

Table 5-2: **GTH Wrapper Component Name**

Name	Description
<project_dir>/<component_name>	
v6_gthwizard_readme.txt	README file for the Wizard.
<component_name>.pf	Protocol description for the selected protocol from the Wizard.

Back to Top

<component name>/doc

The doc directory contains the PDF documentation provided with the Wizard.

Table 5-3: **Doc Directory**

Name	Description
<project_dir>/<component_name>/doc	
ug691_v6_gthwizard.pdf	*LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.10 User Guide*

Back to Top

<component name>/example design

The example design directory contains the example design files provided with the Wizard wrapper.

Table 5-4: **Example Design Directory**

Name	Description
<project_dir>/<component_name>/example_design	
gth<n>_frame_check.v[hd]	Frame-check logic to be instantiated in the example design. <n> ranges from 0 to 3 and corresponds to GTH transceivers 0 to 3 in a quad.
gth<n>_frame_gen.v[hd]	Frame-generator logic to be instantiated in the example design. <n> ranges from 0 to 3 and corresponds to GTH transceivers 0 to 3 in a quad.
gth_attributes.ucf	Constraints file containing the GTH transceiver attributes generated by the GTH transceiver Wizard GUI settings.
<component_name>_top.ucf	Constraint file for mapping the GTH transceiver wrapper example design onto a Virtex-6 HXT FPGA.
<component_name>_top.v[hd]	Top-level example design. Contains the GTH transceiver wrapper, reset logic, and instantiations for frame generator and frame checker.

Table 5-4: **Example Design Directory *(Cont'd)***

Name	Description
gth<n>_rom_init_tx	Data file containing the data pattern for the frame generator. <n> ranges from 0 to 3 and corresponds to GTH transceivers 0 to 3 in a quad.
gth<n>_rom_init_rx	Data file containing the data pattern for the frame checker. <n> ranges from 0 to 3 and corresponds to GTH transceivers 0 to 3 in a quad.

Back to Top

<component name>/implement

The implement directory contains the implementation script files provided with the Wizard wrapper.

Table 5-5: **Implement Directory**

Name	Description
<project_dir>/<component_name>/implement	
implement.bat	Windows batch file that processes the example design through the Xilinx® tool flow.
implement.sh	Linux shell script that processes the example design through the Xilinx tool flow.
implement_synplify.bat	Windows batch file that processes the example design through Synplify synthesis and the Xilinx tool flow.
implement_synplify.sh	Linux shell script that processes the example design through Synplify synthesis and the Xilinx tool flow.
synplify.prj	Synplify project file for the example design.
xst.prj	XST project file for the example design. The file lists all of the source files to be synthesized.
xst.scr	XST script file for the example design that is used to synthesize the Wizard. It is called from the implement script described above.
planAhead_ise.bat	Windows batch file that processes the example design through PlanAhead™ software-based ISE® design tools flow.
planAhead_ise.sh	Linux shell script that processes the example design through PlanAhead software-based ISE design tools flow.
planAhead_ise.tcl	TCL script which contains tool settings and file lists of the source files to be synthesized.
data_vio.ngc	Netlist of the design generated by the ChipScope™ Pro Virtual Input/Output (VIO) Wizard.
icon.ngc	Netlist of the design generated by the ChipScope Pro Integrated Controller (ICON) Wizard.

Table 5-5: **Implement Directory *(Cont'd)***

Name	Description
`ila.ngc`	Netlist of the design generated by the ChipScope Pro Integrated Logic Analyzer (ILA) Wizard.

Back to Top

implement/results

The results directory is created by the implement script, after which the implement script results are placed in the results directory.

Table 5-6: **UCF Directory**

Name	Description
<project_dir>/<component_name>/implement/results	
Implement script result files.	

Back to Top

<component name>/simulation

The simulation directory contains the simulation scripts provided with the Wizard wrapper.

Table 5-7: **Simulation Directory**

Name	Description
<project_dir>/<component_name>/simulation	
`demo_tb.v[hd]`	Testbench to perform functional simulation of the provided example design. See Functional Simulation of the Example Design, page 31.
`demo_tb_imp.v[hd]`	Testbench to perform netlist simulation of the provided example design. See Netlist Simulation of the Example Design, page 33.

Back to Top

simulation/functional

The functional directory contains functional simulation scripts provided with the Wizard wrapper.

Table 5-8: **Functional Directory**

Name	Description
<project_dir>/<component_name>/simulation/functional	
simulate_mti.do	ModelSim simulation script.
wave_mti.do	Script for adding GTH wrapper signals to the ModelSim wave viewer.
gth<n>_rom_init_tx	Data file containing the data pattern for the frame generator. <n> ranges from 0 to 3 and corresponds to GTH transceivers 0 to 3 in a quad.
gth<n>_rom_init_rx	Data file containing the data pattern for the frame checker. <n> ranges from 0 to 3 and corresponds to GTH transceivers 0 to 3 in a quad.
simulate_ncsim.sh	Linux script for running simulation using Cadence Incisive Enterprise Simulator (IES).
simulate_ncsim.bat	Window script for running simulation using Cadence IES.
simulate_vcs.sh	Linux script for running simulation using Synopsys VCS.
ucli_command.key	Command file for VCS simulator.
vcs_session.tcl	Script for adding GTH wrapper signals to VCS wave viewer.
wave_isim.tcl	Script for adding GTH wrapper signals to ISim wave viewer.
wave_mti.do	Script for adding GTH wrapper signals to ModelSim wave viewer.
wave_ncsim.sv	Script for adding GTH wrapper signals to Cadence IES wave viewer.
simulate_isim.sh	Linux script for running simulation using ISE simulator.
simulate_isim.bat	Windows script for running simulation using ISE simulator.

Back to Top

simulation/netlist

The netlist directory contains netlist simulation scripts provided with the Wizard wrapper.

Table 5-9: **Netlist Directory**

Name	Description
<project_dir>/<component_name>/simulation/netlist	
simulate_mti.do	ModelSim simulation script.

Back to Top

Example Design Description

The example design that is delivered with the wrappers helps Wizard designers understand how to use the wrappers and GTH transceivers in a design. The example design is shown in Figure 5-1.

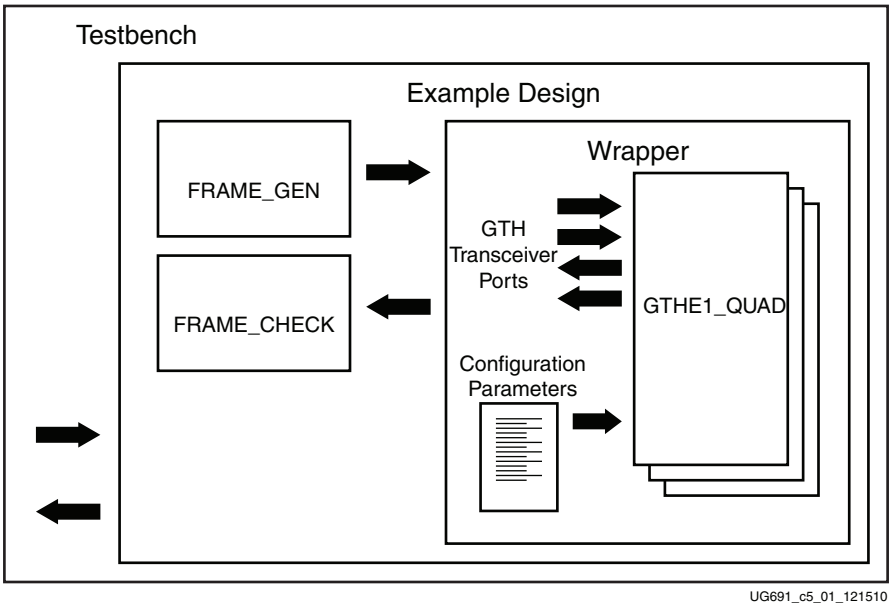


Figure 5-1: **Diagram of Example Design and Testbench**

The example design connects a frame generator and a frame checker to the wrapper. The frame generator transmits an incrementing counting pattern while the frame checker monitors the received data for correctness. The frame generator counting pattern is stored in block RAM. This pattern can be easily modified by altering the parameters in the `gth<n>_rom_init_tx.dat` file. The frame checker contains the same pattern in block RAM and compares it with the received data. An error counter in the frame checker keeps a track of how many errors have occurred.

The frame check works by first scanning the received data for the START_OF_PACKET_CHAR. Once the START_OF_PACKET_CHAR has been found, the received data will continuously be compared to the counting pattern stored in the block RAM at each RXUSERCLKIN cycle. Once comparison has begun, if the received data ever fails to match the data in the block RAM, checking of receive data will immediately stop, an error counter will be incremented and the frame checker will return to searching for the START_OF_PACKET_CHAR.

The example design also demonstrates how to properly connect clocks to GTH transceiver ports TXUSERCLKIN and RXUSERCLKIN.

Example Design Hierarchy

The hierarchy for the design used in this example is:

```
DEMO_TB
    |___TENGBASER_TOP
        |___TENGBASER_WRAPPER
        |    |___TENGBASER_WRAPPER_QUAD (1 per GTHE1_QUAD)
        |
        |___GTH<n>_FRAME_GEN (1 per transceiver, n ranges from 0 to 3 and
                              corresponds to GTH transceivers 0 to 3 in a
                              quad)
        |___GTH<n>_FRAME_CHECK (1 per transceiver, n ranges from 0 to 3
                              and corresponds to GTH transceivers 0 to 3
                              in a quad)
```