LogiCORE™ IP Virtex®-6 FPGA GTH Transceiver Wizard v1.6

Getting Started Guide

UG691 (v1.6) December 14, 2010

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.

Table of Contents

Revision History 2

Preface: About This Guide

Guide Contents 5
Additional Resources 5
Conventions 6
Typographical 6
Online Document 7

Chapter 1: Introduction

About the Wizard 9
Recommended Design Experience 9
Related Xilinx Documents 10
Additional Wizard Resources 10
Technical Support 10
Feedback 10
GTH Transceiver Wizard 10
Document 11

Chapter 2: Installation and Licensing

Supported Tools and System Requirements 13
Operating Systems 13
Tools 13
Before You Begin 14
Installing the Wizard 14
Verifying Your Installation 14

Chapter 3: Running the Wizard

Overview 15
Setting Up the Project 16
Creating a Directory 16
Setting the Project Options 17
Configuring and Generating the Wrapper 18
GTH Placement and Clocking 19
GTH0 Settings 21
GTH1, GTH2, and GTH3 Settings 24
Summary 25

Chapter 4: Quick Start Example Design

Overview . . . 27
Functional Simulation of the Example Design . . . 27
Using ModelSim . . . 27
Using the ISE Simulator . . . 28
Implementing the Example Design . . . 28

Chapter 5: Detailed Example Design

Directory and File Structure . . . 29
Directory and File Contents . . . 30
<project directory> . . . 30
<project directory>/<component name> . . . 30
<component name>/doc . . . 31
<component name>/example design . . . 31
<component name>/implement . . . 32
/implement/results . . . 32
<component name>/simulation . . . 32
/simulation/functional . . . 33
Example Design Description . . . 34
Example Design Hierarchy . . . 35

About This Guide

This guide describes the function and operation of the LogiCORE™ IP GTH Transceiver Wizard for the Virtex®-6 HXT family.

Guide Contents

This guide contains the following chapters:

- Preface, "About this Guide" introduces the organization and purpose of this guide, a list of additional resources, and the conventions used in this document.
- Chapter 1, Introduction describes the wrapper core and related information, including additional resources, technical support, and submitting feedback to Xilinx.
- Chapter 2, Installation and Licensing provides information about installing and licensing the Virtex-6 FPGA GTH Transceiver Wizard.
- Chapter 3, Running the Wizard provides an overview of the Virtex-6 FPGA GTH Transceiver Wizard, and a step-by-step tutorial to generate a sample GTH transceiver wrapper with the Xilinx® 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 Xilinx CORE Generator tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration test bench.

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.

Conventions

This document uses the following conventions. An example illustrates each convention.

Typographical

This document uses the following typographical conventions. An example illustrates each convention.

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages, prompts, and program files that the system displays	`speed grade: - 100`
`Courier bold`	Literal commands that you enter in a syntactical statement	**`ngdbuild`** *design_name*
Helvetica bold	Commands that you select from a menu	**File → Open**
	Keyboard shortcuts	**Ctrl+C**
Italic font	Variables in a syntax statement for which you must supply values	**`ngdbuild`** *design_name*
	References to other manuals	See the *User Guide* for more information.
	Emphasis in text	If a wire is drawn so that it overlaps the pin of a symbol, the two nets are *not* connected.
Dark Shading	Items that are not supported or reserved	This feature is not supported
Square brackets [ ]	An optional entry or parameter. However, in bus specifications, such as **`bus[7:0]`**, they are required.	**`ngdbuild`** [*`option_name`*] *`design_name`*
Braces { }	A list of items from which you must choose one or more	**`lowpwr ={on|off}`**
Vertical bar |	Separates items in a list of choices	**`lowpwr ={on|off}`**
Angle brackets < >	User-defined variable or in code samples	<directory name>

Vertical ellipsis
.
.
.
	Repetitive material that has been omitted	`IOB #1: Name = QOUT'`
		`IOB #2: Name = CLKIN'`
		.
		.
		.

Horizontal ellipsis ...	Repetitive material that has been omitted	**`allow block`** *block_name loc1 loc2 ... locn;*

Convention	Meaning or Use	Example
Notations	The prefix '0x' or the suffix 'h' indicate hexadecimal notation	A read of address 0x00112975 returned 45524943h.
	An '_n' means the signal is active low	**usr_teof_n** is active low.

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section "Additional Resources" for details. Refer to "Title Formats" in Chapter 1 for details.
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest speed files.

Chapter 1

Introduction

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

The Virtex-6 FPGA GTH Transceiver Wizard is a Xilinx® CORE Generator™ tool designed to support both Verilog and VHDL design environments. In addition, the example design delivered with the core is provided in Verilog or VHDL.

The Wizard produces a wrapper that instantiates one or more properly configured GTH transceivers for custom applications (Figure 1-1).

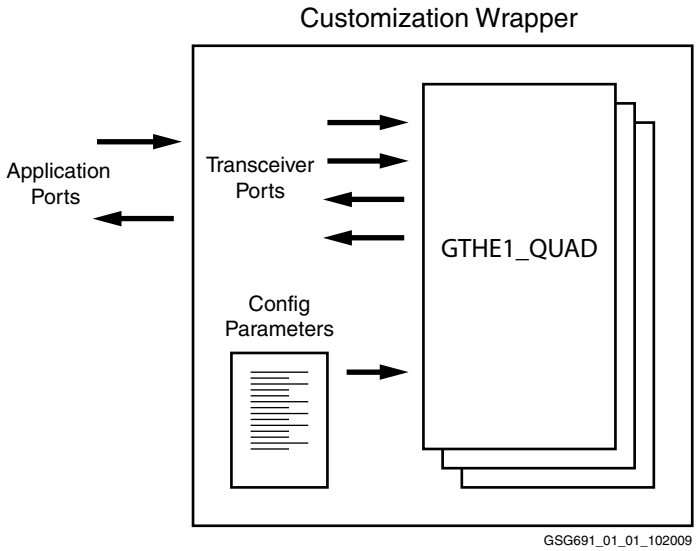

Figure 1-1: **GTH Transceiver Wizard Wrapper**

About the Wizard

The Virtex-6 FPGA GTH Transceiver Wizard is a Xilinx CORE Generator tool, available at the Xilinx IP Center. For information about system requirements, installation, and licensing options, see Chapter 2, Installation and Licensing.

Recommended Design Experience

Although the Virtex-6 FPGA GTH Transceiver Wizard core is a fully verified solution, the challenge associated with implementing a complete design varies depending on the configuration and 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.

Contact your local Xilinx representative for a closer review and estimation for your specific requirements.

Related Xilinx Documents

Prior to generating the Virtex-6 FPGA GTH Transceiver 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

Additional Wizard Resources

For detailed information and updates about the Virtex-6 FPGA GTH Transceiver Wizard, see the following documents located at the Architecture Wizards page.

- *LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.6 Data Sheet*
- Virtex-6 FPGA GTH Transceiver Wizard Release Notes

Technical Support

For technical support, go to www.xilinx.com/support. Questions are routed to a team of engineers with expertise using the Virtex-6 FPGA GTH Transceiver Wizard.

Xilinx provides technical support for use of this product as described in the *LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.6 Getting Started Guide*. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow these guidelines.

Feedback

Xilinx welcomes comments and suggestions about the Virtex-6 FPGA GTH Transceiver Wizard and the accompanying documentation.

GTH Transceiver Wizard

For comments or suggestions about the Virtex-6 FPGA GTH Transceiver Wizard, please 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 your comments, including whether the case is requesting an *enhancement* (you believe something could be improved) or reporting a *defect* (you believe something isn't working correctly).

Document

For comments or suggestions about this document, please 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 which your comments refer
- Explanation of your comments, including whether the case is requesting an *enhancement* (you believe something could be improved) or reporting a *defect* (you believe something isn't documented correctly).

XILINX®

Chapter 2

Installation and Licensing

This chapter provides instructions for installing the Virtex®-6 FPGA GTH Transceiver Wizard in the Xilinx® CORE Generator™ tool.

Note: It is not necessary to obtain a license to use the Wizard.

Supported Tools and System Requirements

Operating Systems

Windows

- Windows XP Professional 32-bit/64-bit
- Windows Vista Business 32-bit/64-bit

Linux

- Red Hat Enterprise Linux WS v4.0 32-bit/64-bit
- Red Hat Enterprise Desktop v5.0 32-bit/64-bit (with Workstation Option)
- SUSE Linux Enterprise (SLE) v10.1 32-bit/64-bit

Tools

- ISE® 12.4 software
- Mentor Graphics ModelSim 6.5c
- Cadence Incisive Enterprise Simulator (IES) 9.2
- Synopsys VCS and VCS MX 2009.12

Check the release notes for the required Service Pack; ISE Service Packs can be downloaded from www.xilinx.com/support/download.htm.

Before You Begin

Before installing the Wizard, you must have a MySupport account and the ISE 12.4 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 12.4 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 Virtex-6 FPGA GTH Transceiver Wizard is included with the ISE 12.4 software. See *ISE CORE Generator IP Updates - Installation Instructions* for details about installing ISE 12.4.

Verifying Your Installation

Use the following procedure to verify that you have successfully installed the Virtex-6 FPGA GTH Transceiver 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, as shown in Figure 2-1.

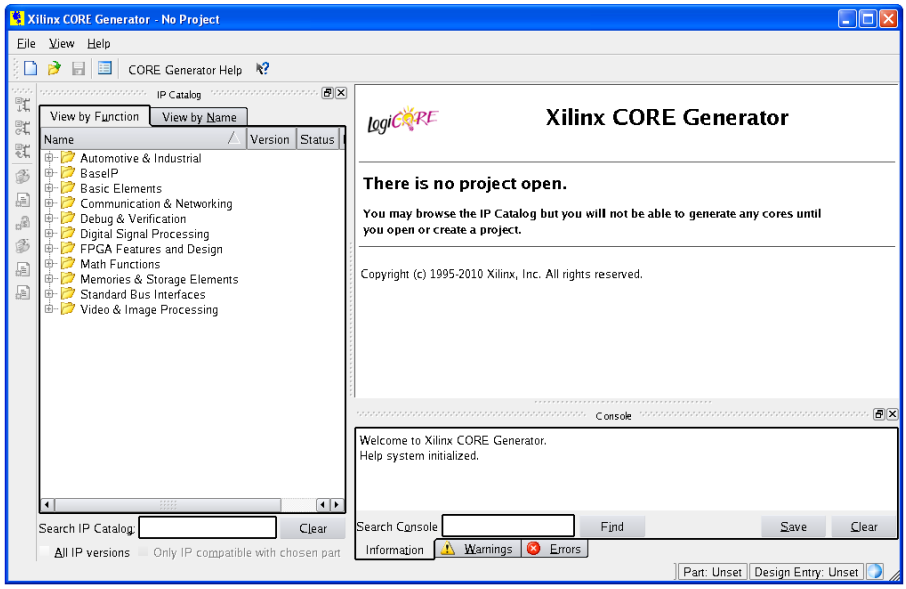

Figure 2-1: **CORE Generator Window**

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 1.6 appears at the following location in the Functional Categories list: `/FPGA Features and Design/IO Interfaces`

Chapter 3

Running the Wizard

Overview

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

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.

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

- A single GTH transceiver wrapper implementing a one-lane 10GBASE-R port using one GTH transceiver
- A demonstration test bench 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
- Scripts to synthesize and simulate the example design

The Virtex-6 FPGA GTH Transceiver Wizard example design has been tested with ModelSim 6.5c for simulation.

Figure 3-1 shows a block diagram of the default 10GBASE-R example design.

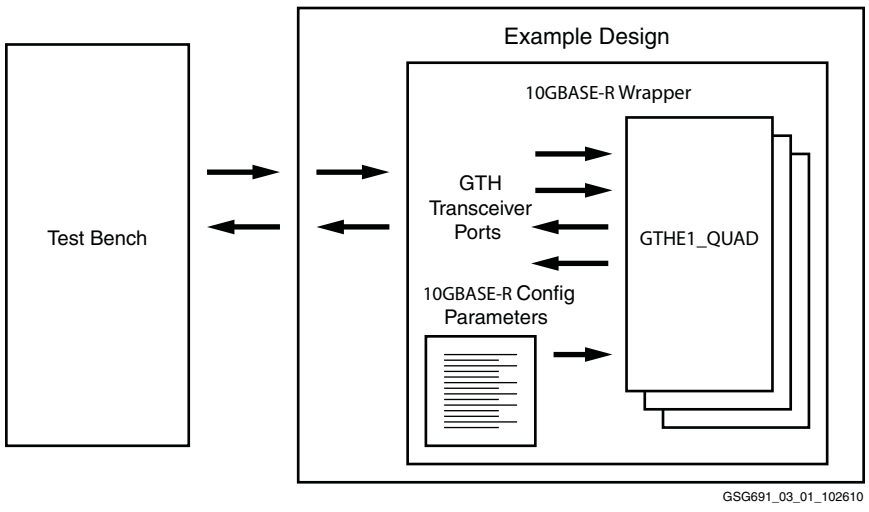

Figure 3-1: **Example Design**

Setting Up the Project

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

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 Xilinx 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 3-2).
4. Change the name of the .cgp file (optional).
5. Click **Save**.

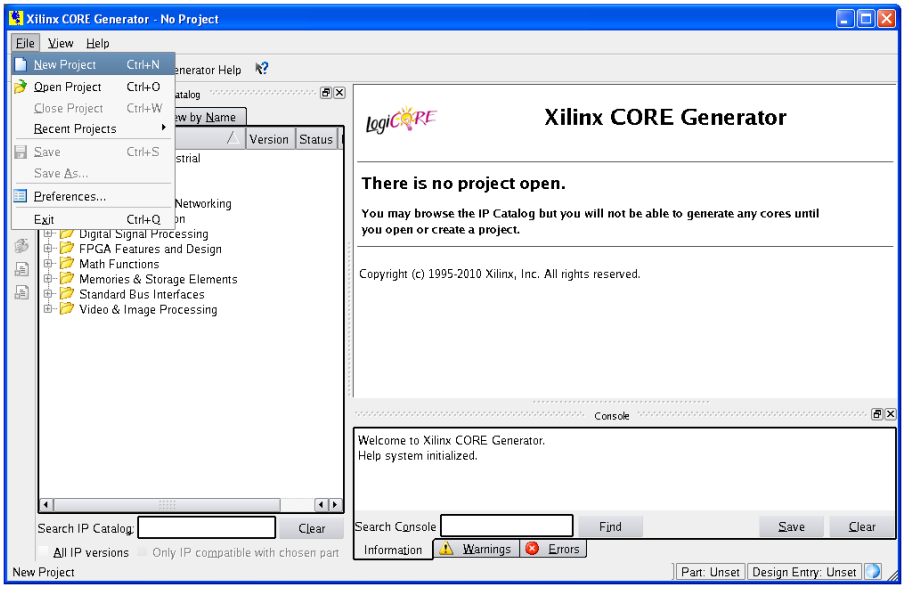

Figure 3-2: **Starting a New Project**

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 3-3).

   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 the *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**.

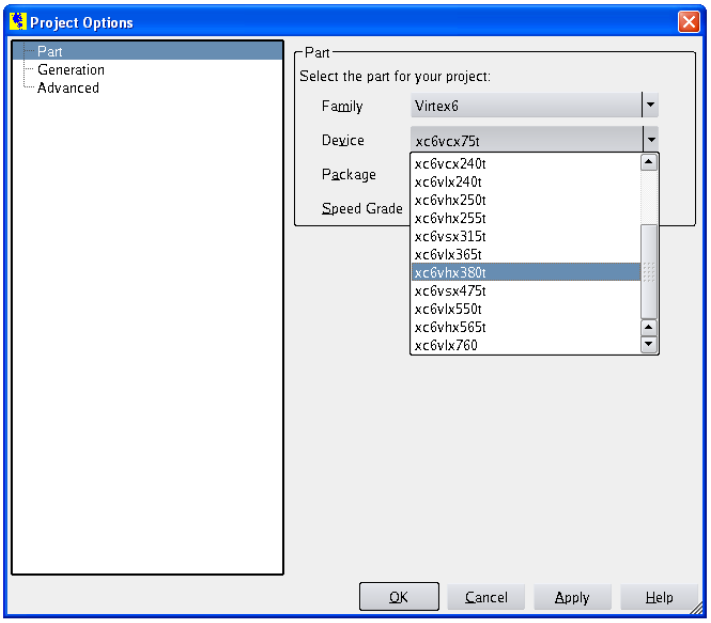

Figure 3-3: **Target Architecture Setting**

Configuring and Generating the Wrapper

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

1. Locate Virtex-6 FPGA GTH Transceiver Wizard 1.6 in the taxonomy tree under: `/FPGA Features & Design/IO Interfaces`. (See Figure 3-4)
2. Double-click **Virtex-6 FPGA GTH Transceiver Wizard 1.6** to launch the Wizard.

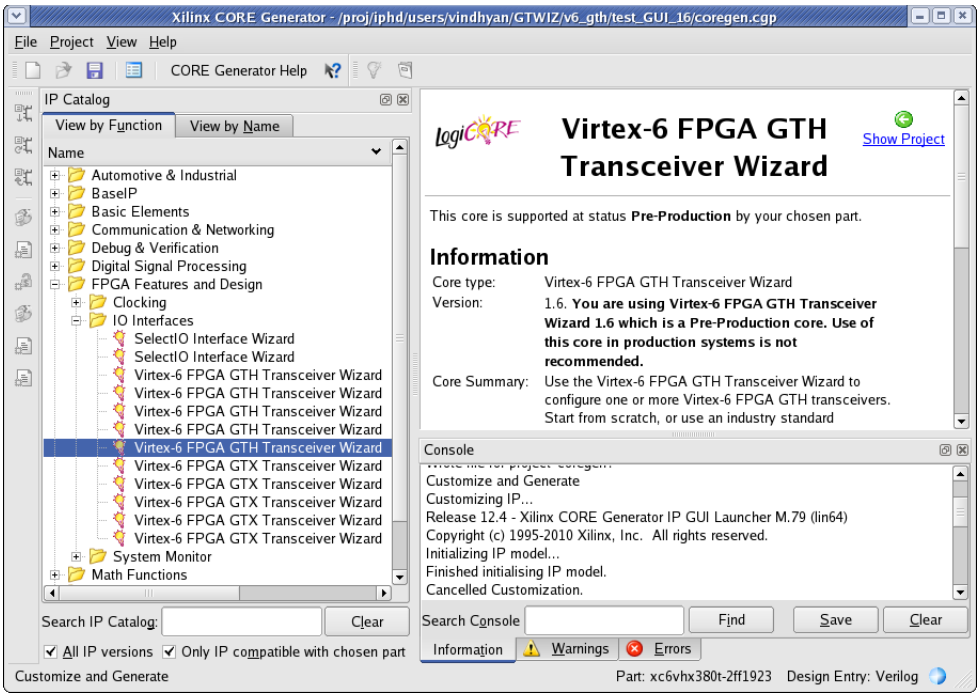

Figure 3-4: **Locating the GTH Transceiver Wizard**

GTH Placement and Clocking

Page 1 of the Wizard (Figure 3-5) determines the component name, placement of the GTHE1_QUAD, reference clock source, target line rate, reference clock frequency, and DRP clock frequency. In addition, this page specifies a protocol template.

1. In the Component Name filed, enter a name for the core 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 core 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 20 describes the GTHE1_QUAD selection and reference clock options.

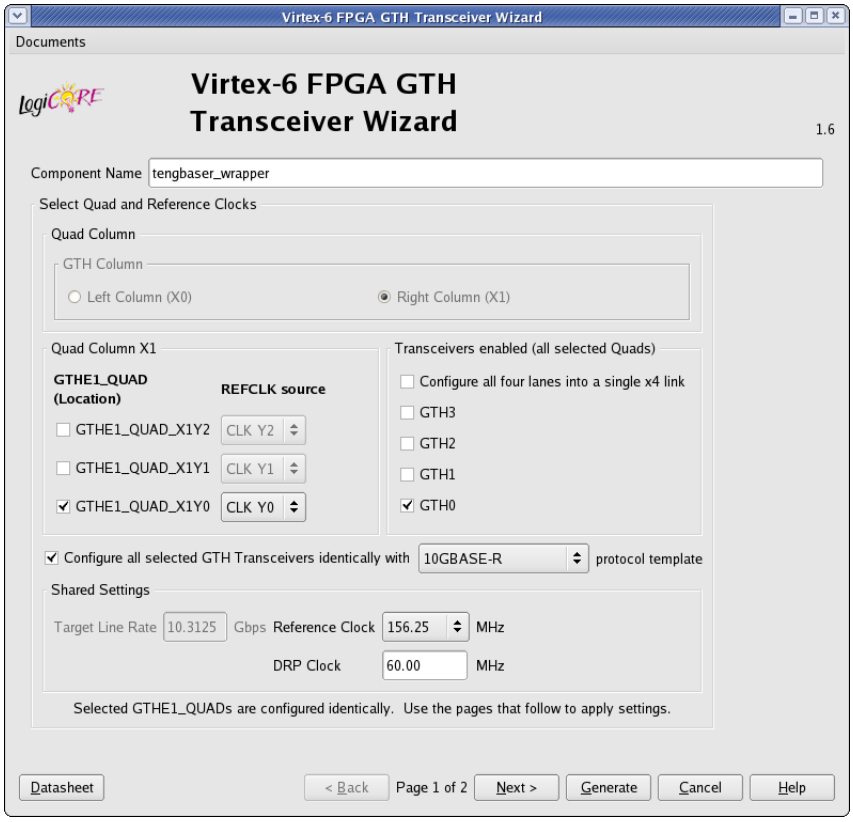

Figure 3-5: **GTH Transceiver Wizard Page 1 of 2**

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 Gbps desired for the target design. The 10GBASE-R example uses 10.3125 Gbps.
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.

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 3-6: **GTH0 Settings - Wizard Page 2 of 2**

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

Option	Description
GTH0 Line Rate	Line rate in Gbps 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 Gbps.
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 single 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 multi-lane 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 multi-lane 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	Pre-cursor 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 3-7, Figure 3-8, and Figure 3-9, page 25 are visible based on the GTH1, GTH2 and GTH3 selections on Page 1 (Figure 3-5, page 19). For the description of the options, see GTH0 Settings, page 21.

Figure 3-7: **GTH1 Settings - Wizard Page 3 of 5**

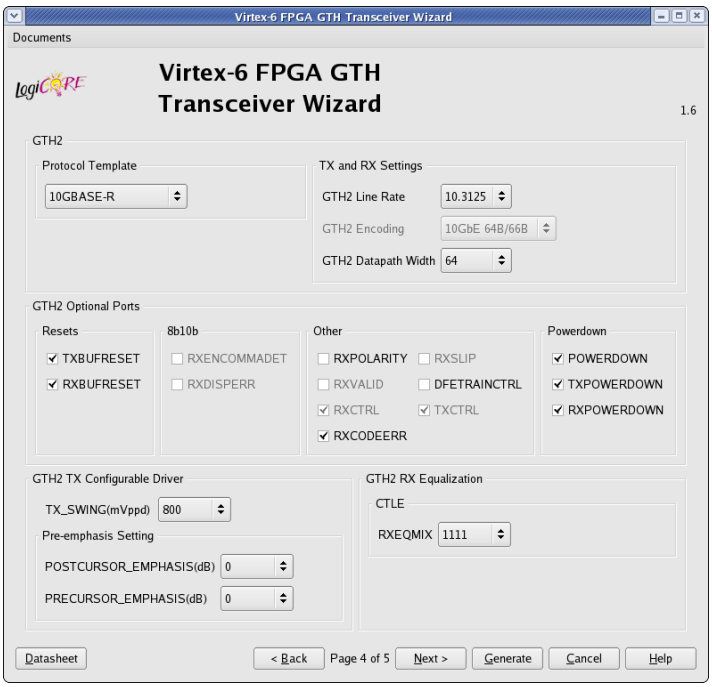

Figure 3-8: **GTH2 Settings - Wizard Page 4 of 5**

Figure 3-9: **GTH3 Settings - Wizard Page 5 of 5**

Summary

Not Applicable.

XILINX®

Chapter 4

Quick Start Example Design

Overview

This chapter 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. 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 provides a quick way to simulate and observe the behavior of the wrapper using the provided example design and script files.

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 12.4, 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*>/virtex6/verilog/src/unisims <*Xilinx dir*>/virtex6/secureip/mti
VHDL	UNISIM	<*Xilinx dir*>/virtex6/vhdl/src/unisims/primitive <*Xilinx dir*>/virtex6/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 test bench, 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 test bench, 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 Xilinx 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.

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 Xilinx® CORE Generator™ tool, the purpose and contents of the provided scripts, the contents of the example HDL wrappers, and the operation of the demonstration test bench.

Directory and File Structure

<project directory>
Top-level project directory; name is user-defined

<project directory>/<component name>
Core 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

Directory and File Contents

The Virtex®-6 FPGA GTH Transceiver Wizard core 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 wrapper files instantiation templates. Includes templates for the GTH wrapper module, and the IBUFDS_GTHE1.
<component_name>.xco	Log file from CORE Generator tool describing which options were used to generate the GTH wrapper. An XCO file is generated by CORE Generator tool for each core 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.

Back to Top

<project directory>/<component name>

The <component name> directory contains the release notes file provided with the core, which may include last-minute changes and updates.

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

Name	Description
<project_dir>/<component_name>	
v6_gthwizard_readme.txt	Release notes for the GTH wizard
<component_name>.pf	Protocol description for the selected protocol from the GTH wizard

Back to Top

<component name>/doc

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

Table 5-3: **Doc Directory**

Name	Description
<project_dir>/<component_name>/doc	
v6_gthwizard_ds738.pdf	*LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.6 Data Sheet*
v6_gthwizard_gsg691.pdf	*LogiCORE IP Virtex-6 FPGA GTH Transceiver Wizard v1.6 Getting Started Guide*

Back to Top

<component name>/example design

The example design directory contains the example design files provided with the core.

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 attributes generated by the GTH Wizard GUI settings.
<component_name>_top.ucf	Constraint file for mapping the GTH wrapper example design onto a Virtex-6 HXT device.
<component_name>_top.v[hd]	Top-level example design. Contains GTH transceiver wrapper, reset logic, and instantiations for frame generator and frame-checker.

Back to Top

<component name>/implement

The implement directory contains the core implementation script files.

Table 5-5: **Implement Directory**

Name	Description
<project_dir>/<component_name>/implement	
`implement.bat`	A Windows batch file that processes the example design through the Xilinx tool flow.
`implement.sh`	A Linux shell script that processes the example design through the Xilinx tool flow.
`xst.prj`	The XST project file for the example design; it lists all of the source files to be synthesized.
`xst.scr`	The XST script file for the example design that is used to synthesize the core, called from the implement script described above.

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 core.

Table 5-7: **Simulation Directory**

Name	Description
<project_dir>/<component_name>/simulation	
`demo_tb.v`	Test bench to simulate the provided example design. See Functional Simulation of the Example Design, page 27.

Back to Top

/simulation/functional

The functional directory contains functional simulation scripts provided with the core.

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 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 GTX wrapper signals to VCS wave window
wave_isim.tcl	Script for adding GTX wrapper signals to the ISim wave viewer.
wave_mti.do	Script for adding GTX wrapper signals to the ModelSim wave viewer.
wave_ncsim.sv	Script for adding GTX wrapper signals to the Cadence IES wave viewer.

Back to Top

Example Design Description

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

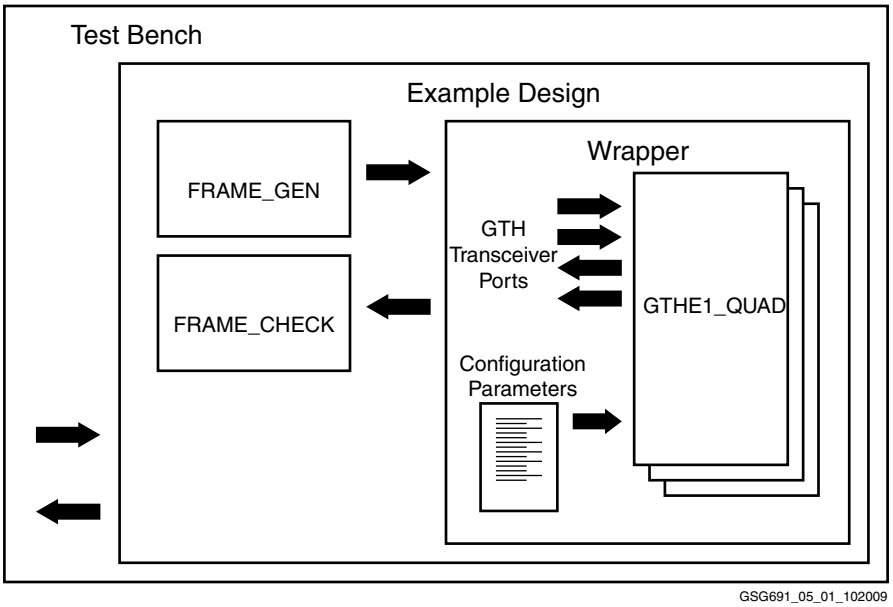

Figure 5-1: **Wrapper Block Diagram**

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 BRAM. 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 BRAM 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 BRAM at each RXUSERCLKIN cycle. Once comparison has begun, if the received data ever fails to match the data in the BRAM, 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 shown below.

```
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)
```