LogiCORE™ IP Virtex®-5 FPGA RocketIO™ GTP Transceiver Wizard v2.1

Getting Started Guide

UG188 (v2.1) April 19, 2010

XILINX®

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice.

XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/10/06	1.0	Initial Xilinx release.
11/30/06	1.1	New Wizard software revision 1.3. New v1.3 screen captures. Revise Installation procedure.
03/01/07	1.2	Standardize Document name. New Wizard software revision 1.4. New v1.4 screen captures.
05/17/07	1.3	Wizard 1.5 release. New Wizard software revision 1.5. New v1.5 screen captures. Added list sort detail (new feature) to Reference Clock description in Table 3-3, page 22.
08/15/07	1.4	Wizard v1.6 release. Software and screen captures updated to v1.6.
10/10/07	1.5	Wizard v1.7 release. Software and screen captures updated to v1.7.
03/24/08	1.6	Wizard v1.8 release. Software and screen captures updated to v1.8. Major revision to Chapter 2, "Installation and Licensing"sections: "System Requirements," "Before You Begin," and "Installing the Wizard." Major revisions to Chapter 3, "Running the Wizard" in: "Creating a Directory," Table 3-9, page 28, Table 3-19, page 39, and "Functional Simulation of the Example Design."Added Appendix A, "References."
03/24/08	1.6.1	Minor edits: • Preface, first paragraph: Move trademark symbol. • Remove version numbers from referenced document titles. • Change "test bench" to two words.

Date	Version	Revision
06/26/08	1.9	Synchronize Document version with Wizard version. Revised parameter tables and shading to match current values. Replaced page headings with descriptive names in Configuring the Wrapper. Minor edits: • Revised version numbers. • Installing the Wizard: Revised section. • Verifying Your Installation: Removed redundant support link. • Latency, Buffering, and Clocking: ♦ Revised TX PCS/PMA Phase Alignment section. ♦ Added Table 3-7. • Table 3-16: Add "Decode Valid Comma Only" entry. • Table 3-19: Removed "Enable Asynchronous and/or Spread Spectrum Clocking" entry. • Implementing the Example Design: ♦ Added source file details section. ♦ Added Table 3-21. ♦ Corrected directory name and added note in script examples.
06/27/08	1.9.1	Updated release date and trademarks.
06/24/09	1.10	Updated for v1.10 Wizard release. Miscellaneous edits throughout. Moved the example design and directory hierarchy into the new Chapter 5, "Detailed Example Design."
04/19/10	2.1	Updated for v2.1 Wizard release. Moved quick-start instructions to new Chapter 4, "Quick Start Example Design." The previous chapter 4 is now Chapter 5, "Detailed Example Design."

Table of Contents

Preface: About This Guide

Chapter 1: Introduction

Chapter 2: Installation and Licensing

Chapter 3: Running the Wizard

Chapter 4: Quick Start Example Design

Chapter 5: Detailed Example Design

Appendix A: References

About This Guide

The *LogiCORE IP Virtex-5 FPGA RocketIO GTP Transceiver Wizard Getting Started Guide* describes the function and operation of the LogiCORE™ IP RocketIO™ GTP Wizard for the Virtex®-5 LXT and SXT sub-families.

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 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-5 FPGA RocketIO GTP Transceiver Wizard.
- Chapter 3, "Running the Wizard" provides an overview of the Virtex-5 FPGA RocketIO GTP Transceiver Wizard, and gives a step-by-step tutorial on how to generate a sample RocketIO GTP transceiver wrapper with the Xilinx® CORE Generator™ tool.
- Chapter 4, "Quick Start Example Design" introduces the example design that is included with the Virtex-5 FPGA RocketIO GTP Transceiver Wizard. The quick start instructions are a step-by-step procedure for generating an Virtex-5 FPGA RocketIO GTP Transceiver Wizard, implementing the core in hardware using the accompanying example design, and simulating the core with the provided demonstration test bench (demo_tb).
- 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

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.
Red text	Cross-reference link to a location in another document	See Figure 2-5 in the *Virtex-II Platform FPGA User Guide.*
Blue, underlined text	Hyperlink to a website (URL)	Go to www.xilinx.com for the latest speed files.

Introduction

This chapter introduces the Virtex®-5 FPGA RocketIO™ GTP Transceiver Wizard core and provides related information, including additional resources, technical support, and submitting feedback to Xilinx.

The Virtex-5 FPGA RocketIO GTP 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 either Verilog or VHDL.

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

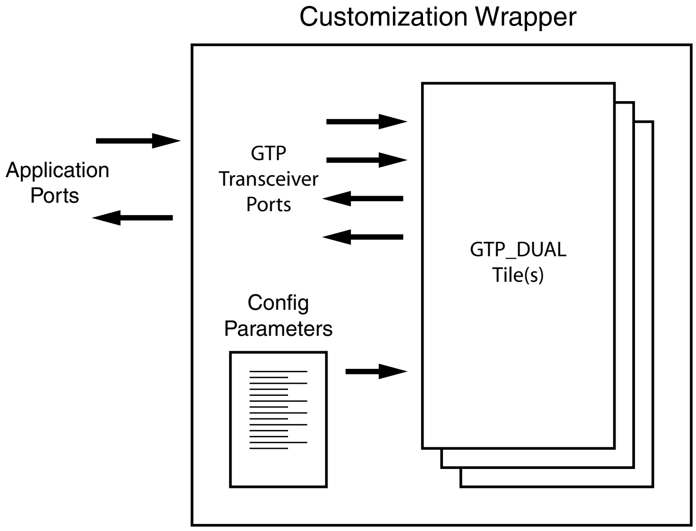


Figure 1-1: **GTP Transceiver Wizard Wrapper**

About the Wizard

The Virtex-5 FPGA RocketIO GTP 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."

Additional Wizard Resources

For detailed information and updates about the Virtex-5 FPGA RocketIO GTP Transceiver Wizard, see the following documents located at the Architecture Wizard page.

- DS590: *Virtex-5 FPGA RocketIO GTP Transceiver Wizard Data Sheet*, [Ref 3]
- Virtex-5 FPGA RocketIO GTP Transceiver Wizard Release Notes

Technical Support

For technical support, go to www.xilinx.com/support. Questions are routed to a team with expertise using the Virtex-5 FPGA RocketIO GTP Transceiver Wizard.

Xilinx provides technical support for use of this product as described in the *LogiCORE IP Virtex-5 FPGA RocketIO GTP Transceiver Wizard 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-5 FPGA RocketIO GTP Transceiver Wizard and the accompanying documentation.

Virtex-5 FPGA RocketIO GTP Transceiver Wizard

For comments or suggestions about the Virtex-5 FPGA RocketIO GTP Transceiver Wizard, please submit a WebCase from www.xilinx.com/support. 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. Be sure to include the following information:

- Document title
- Document number
- Document revision 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).

Chapter 2

Installation and Licensing

This chapter provides instructions for installing the Virtex®-5 FPGA RocketIO™ GTP Transceiver Wizard in the Xilinx® CORE Generator™ tool. It is not necessary to obtain a license to use the Wizard.

System Requirements

Windows

- Windows XP Professional SP1, SP2, 32-bit, 64-bit
- Windows Vista Business 32-bit

Linux

- Red Hat Enterprise 3.0/4.0/5.0 32-bit/64-bit

Software

- ISE® software v12.1
- Mentor Graphics ModelSim v6.5c and above

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 12.1 software installed on your system. If you already have an account and have the software installed, go to "Verifying Your Installation," page 14, 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 12.1 software. For the software installation instructions, refer to the ISE Design Suite Release Notes and Installation Guide available in ISE software Documentation [Ref 4].

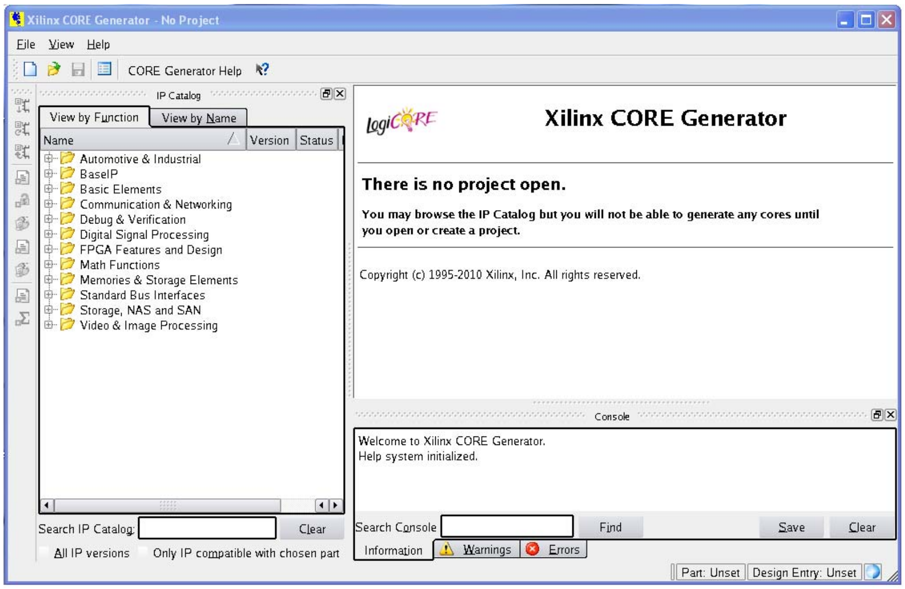
Installing the Wizard

The Virtex-5 FPGA RocketIO GTP Transceiver Wizard is included with the ISE 12.1 software. See *ISE CORE Generator IP Updates - Installation Instructions* for details on the installation of ISE 12.1.

Verifying Your Installation

Use the following procedure to verify that you have successfully installed the Virtex-5 FPGA RocketIO GTP Transceiver Wizard in the CORE Generator tool.

1. Start the CORE Generator tool.
2. After creating a new Virtex®-5 LXT/SXT family project, or opening an existing one, the IP core functional categories appear at the left side of the window, as shown in Figure 2-1.



There is no project open.

You may browse the IP Catalog but you will not be able to generate any cores until you open or create a project.

Copyright (c) 1995-2010 Xilinx, Inc. All rights reserved.

```
Welcome to Xilinx CORE Generator.
Help system initialized.
```

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 bottom 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-5 FPGA RocketIO GTP Transceiver Wizard 2.1 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®-5 FPGA RocketIO™ GTP 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 RocketIO GTP transceivers for use in a XAUI application. Guidelines are also given for incorporating the wrapper in a design and for the expected behavior in operation.

The XAUI example consists of the following components:

- A single RocketIO GTP transceiver wrapper implementing a four-lane XAUI port using four RocketIO GTP transceivers on two GTP_DUAL tiles
- A demonstration test bench to drive the example design in simulation
- An example design providing clock signals and connecting an instance of the XAUI wrapper with modules to drive and monitor the wrapper in hardware, including optional ChipScope™ Pro modules
- Scripts to synthesize and simulate the example design

The Virtex-5 FPGA RocketIO GTP Transceiver Wizard example design has been tested with XST 12.1 for synthesis and ModelSim 6.5c for simulation.

Figure 3-1, page 15 shows a block diagram of the default XAUI example design.

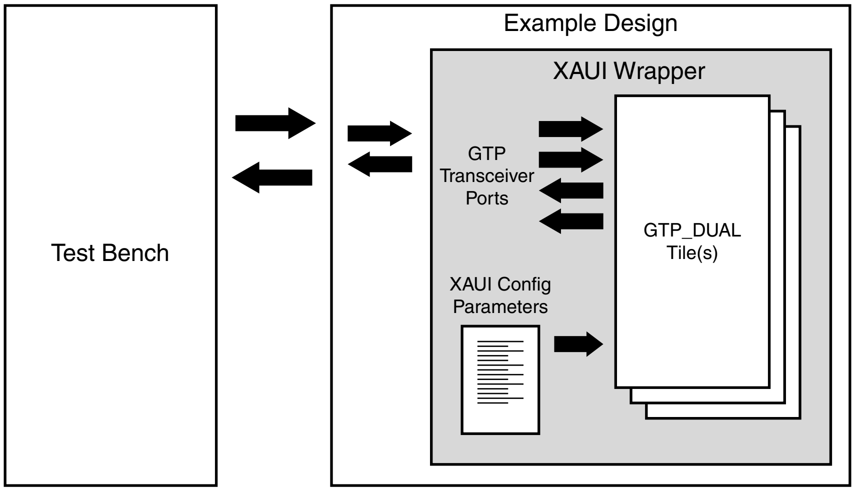


Figure 3-1: **Example Design**

Setting Up the Project

Before generating the example design, set up the project as shown 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:

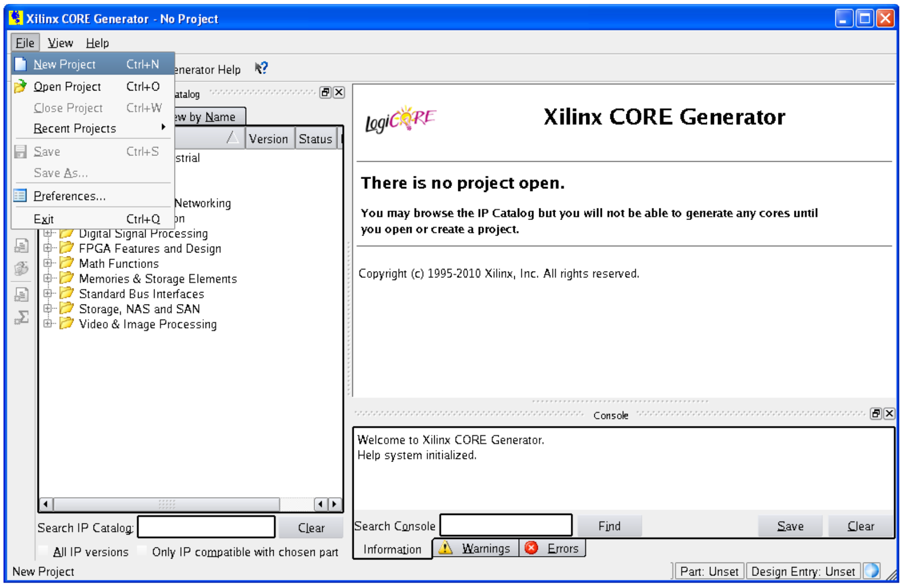
1. Change directory to the desired location. This example uses the following location and directory name:

   `/Projects/xaui_example`

2. Start the CORE Generator™ tool.

   For help starting and using the CORE Generator tool, see *CORE Generator Help*, available in ISE® software documentation [Ref 4].

3. Choose **File** → **New Project** (Figure 3-2, page 16).
4. Optionally change the name of the `.cgp` file.
5. Click **Save**.



There is no project open.

You may browse the IP Catalog but you will not be able to generate any cores until you open or create a project.

Copyright (c) 1995-2010 Xilinx, Inc. All rights reserved.

Welcome to Xilinx CORE Generator.
Help system initialized.

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

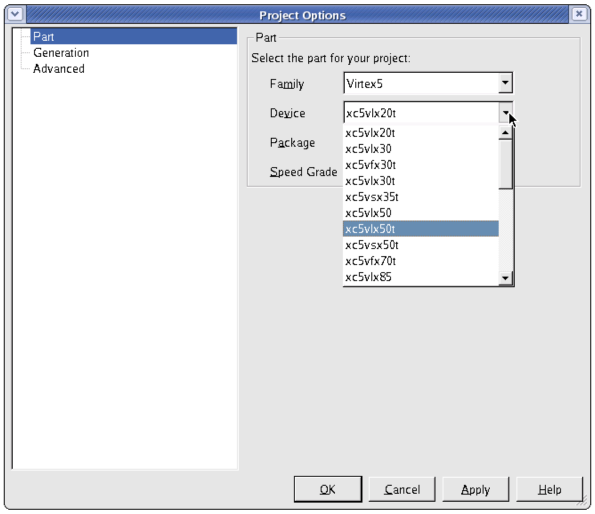
Setting the Project Options

Set the project options using the following steps:

1. Click **Part** in the option tree.
2. Select **Virtex5** from the Family list.
3. Select a device from the Device list which supports RocketIO GTP transceivers.
4. Select an appropriate package from the Package list. This example uses the XC5VLX50T device (see Figure 3-3).

   Note: If an unsupported silicon family is selected, the Virtex-5 FPGA RocketIO GTP Transceiver Wizard appears light grey in the taxonomy tree and cannot be customized. Only devices containing RocketIO GTP transceivers are supported by the Wizard. See the *Virtex-5 Family Overview* [Ref 1] for a list of devices containing RocketIO GTP transceivers.

5. Click **Generation** in the option tree and select either Verilog or VHDL as the output language.
6. Click **OK**.



Select the part for your project:

Family Virtex5

Device xc5vlx20t

Package

Speed Grade

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

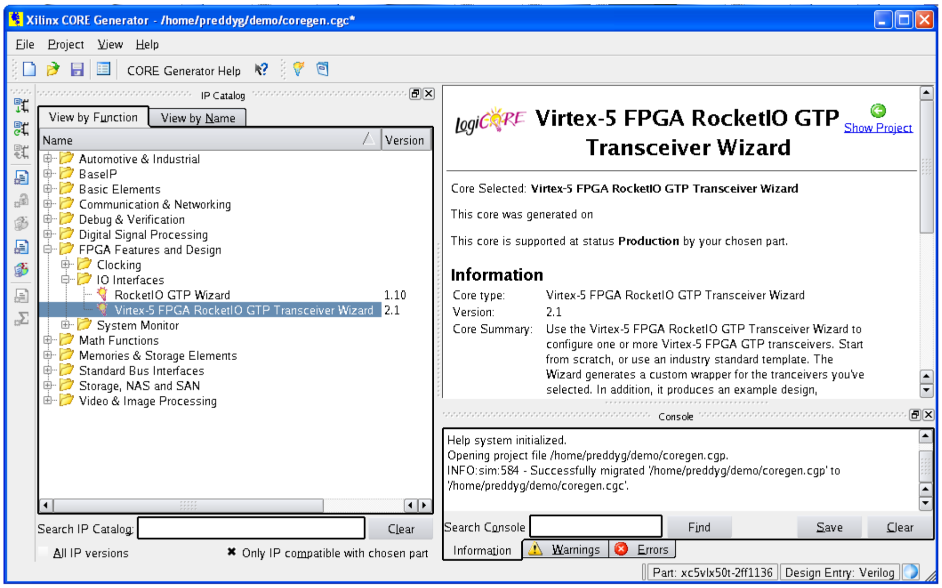
Configuring the Wrapper

This section shows how to generate an example RocketIO GTP transceiver wrapper 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 the RocketIO GTP Wizard Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1 in the taxonomy tree under:

   `/FPGA Features & Design/IO Interfaces`. (See Figure 3-3)

2. Double-click **Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1** to launch the Wizard.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Core Selected: **Virtex-5 FPGA RocketIO GTP Transceiver Wizard**

This core was generated on

This core is supported at status **Production** by your chosen part.

Information

Core type: Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Version: 2.1

Core Summary: Use the Virtex-5 FPGA RocketIO GTP Transceiver Wizard to configure one or more Virtex-5 FPGA GTP transceivers. Start from scratch, or use an industry standard template. The Wizard generates a custom wrapper for the tranceivers you've selected. In addition, it produces an example design,

```
Help system initialized.
Opening project file /home/preddyg/demo/coregen.cgp.
INFO:sim:584 - Successfully migrated '/home/preddyg/demo/coregen.cgp' to
'/home/preddyg/demo/coregen.cgc'.
```

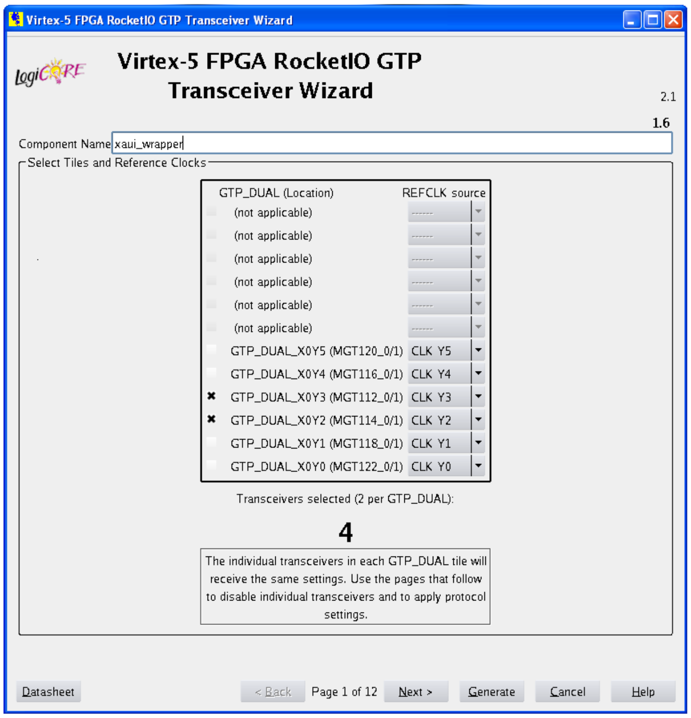
Figure 3-4: **Locating the RocketIO GTP Wizard**

Component Name and Tile Placement

Page 1 of the Wizard (Figure 3-5) is for selecting a component name, determining the placement of the GTP_DUAL tiles, and selecting the reference clock source.

1. In the Component Name field, enter a name for the core instance. This example uses the name **xaui_wrapper**.

The number of available GTP_DUAL tiles appearing on this page depends on the selected target device and package. The XAUI example design uses two tiles for a total of four GTP transceivers. Table 3-1, page 19 describes the GTP_DUAL tile selection and Reference Clock options.


Virtex-5 FPGA RocketIO GTP Transceiver Wizard
LogiCORE
Virtex-5 FPGA RocketIO GTP Transceiver Wizard
2.1
1.6

Component Name xaui_wrapper

Select Tiles and Reference Clocks

GTP_DUAL (Location)	REFCLK source
□ (not applicable)	------
□ (not applicable)	------
□ (not applicable)	------
□ (not applicable)	------
□ (not applicable)	------
□ (not applicable)	------
□ GTP_DUAL_X0Y5 (MGT120_0/1)	CLK Y5
□ GTP_DUAL_X0Y4 (MGT116_0/1)	CLK Y4
☒ GTP_DUAL_X0Y3 (MGT112_0/1)	CLK Y3
☒ GTP_DUAL_X0Y2 (MGT114_0/1)	CLK Y2
□ GTP_DUAL_X0Y1 (MGT118_0/1)	CLK Y1
□ GTP_DUAL_X0Y0 (MGT122_0/1)	CLK Y0

Transceivers selected (2 per GTP_DUAL):

4

The individual transceivers in each GTP_DUAL tile will receive the same settings. Use the pages that follow to disable individual transceivers and to apply protocol settings.

Figure 3-5: **RocketIO GTP Wizard Page 1**

Table 3-1: **Select Tile and Reference Clocks**

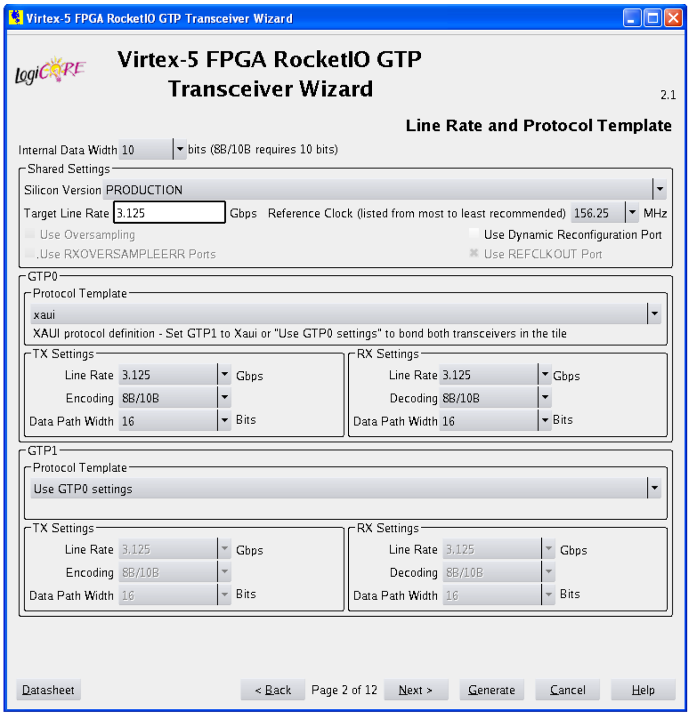
Option	Description
GTP_DUAL	Select the individual number of GTP_DUAL tiles by location to be used in the target design.
REFCLK Source	Determines the source for the reference clock signal provided to each selected GTP_DUAL tile (see Table 3-2, page 20). The XAUI example uses the reference clock from the differential input pins for the upper GTP_DUAL tile (CLK Y3 option) for both tiles.

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

REFCLK Sources	Description
GREFCLK	Reference clock driven by BUFG or BUFR. Lowest performance option.
CLK_Y0	Dedicated GTP reference clock for GTP_DUAL_X0Y0
CLK_Y1	Dedicated GTP reference clock for GTP_DUAL_X0Y1
CLK_Y2	Dedicated GTP reference clock for GTP_DUAL_X0Y2
CLK_Y3	Dedicated GTP reference clock for GTP_DUAL_X0Y3
CLK_Y4	Dedicated GTP reference clock for GTP_DUAL_X0Y4
CLK_Y5	Dedicated GTP reference clock for GTP_DUAL_X0Y5
CLK_Y6	Dedicated GTP reference clock for GTP_DUAL_X0Y6
CLK_Y7	Dedicated GTP reference clock for GTP_DUAL_X0Y7
CLK_Y8	Dedicated GTP reference clock for GTP_DUAL_X0Y8
CLK_Y9	Dedicated GTP reference clock for GTP_DUAL_X0Y9
CLK_Y10	Dedicated GTP reference clock for GTP_DUAL_X0Y10
CLK_Y11	Dedicated GTP reference clock for GTP_DUAL_X0Y11

Line Rate and Protocol Template

Page 2 of the Wizard (Figure 3-6) determines the line rate, reference clock frequency, encoding/decoding method, and data width. In addition, this page specifies a protocol template and a silicon version.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1

Line Rate and Protocol Template

Internal Data Width 10 bits (8B/10B requires 10 bits)

Shared Settings

Silicon Version PRODUCTION

Target Line Rate 3.125 Gbps Reference Clock (listed from most to least recommended) 156.25 MHz

□ Use Oversampling
□ Use RXOVERSAMPLEERR Ports
□ Use Dynamic Reconfiguration Port
☒ Use REFCLKOUT Port

GTP0

Protocol Template

xaui

XAUI protocol definition - Set GTP1 to Xaui or "Use GTP0 settings" to bond both transceivers in the tile

TX Settings

Line Rate 3.125 Gbps
Encoding 8B/10B
Data Path Width 16 Bits

RX Settings

Line Rate 3.125 Gbps
Decoding 8B/10B
Data Path Width 16 Bits

GTP1

Protocol Template

Use GTP0 settings

TX Settings

Line Rate 3.125 Gbps
Encoding 8B/10B
Data Path Width 16 Bits

RX Settings

Line Rate 3.125 Gbps
Decoding 8B/10B
Data Path Width 16 Bits

Figure 3-6: **RocketIO GTP Wizard Page 2**

1. Set the Internal Data Width to 8 or 10 as needed. If 8B/10B encoding/decoding is used, select 10-bit Internal Data Width to accommodate the encoded values. The application interface remains 8 or 16 bits. With no encoding, a 10-bit Internal Data Width yields a 10- or 20-bit application interface. The XAUI example requires 10 bits.

   Table 3-3, page 22 shows the options for the Shared Settings. These options establish the shared PMA PLL settings for both RocketIO GTP transceivers on each tile.

   Note: In all of the following tables, options not used by the XAUI example are shaded.

   The remaining options are divided into GTP0 and GTP1 groups with identical parameters. These apply to the two GTP transceivers present in each GTP_DUAL tile. The remaining discussion in this chapter describes only the GTP0 portion.

Table 3-3: **Shared Settings**

Option	Description
Silicon Version	Select from the list the version of the target device being used.
Target Line Rate	Line rate in Gb/s desired for the target design. The XAUI example uses 3.125 Gb/s.
Reference Clock	Select from the list the optimal reference clock frequency to be provided by the application. Frequencies are listed from the most recommended value to the least recommended. The XAUI example uses 156.25 MHz.
Use Oversampling	The GTP Wizard supports Oversampling for line rates between 100 Mb/s and 640 Mb/s. For line rates of 100 to 500 Mb/s, this option is automatically selected and the check box is disabled. For line rates of 500-640 Mb/s, the checkbox is enabled allowing optional selection of this feature. This option is not available for XAUI since the line rate exceeds the permissible range.
Use RXOVERSAMPLERR Ports	Select this option to have the RXOVERSAMPLEERR signals from both transceivers available to the application. The XAUI example does not use this signal.
Use Dynamic Reconfiguration Port	Select this option to have the Dynamic Reconfiguration Port signals available to the application.
Use REFCLKOUT Port	Select this option to have the REFCLKOUT signal available to the application. Any options selected on the following pages which require this signal will force selection and disable this check box. The XAUI example requires this signal.[1]

Notes:

1. See Figure 3-8, page 26

2. From the Protocol Template list, select **Start from scratch** if you wish to manually set all parameters. Select from the list one of the available protocols to begin your design with a pre-defined protocol template. For GTP1 only, select **Use GTP0 settings** to automatically copy the settings from GTP0.

   The XAUI example uses the XAUI protocol template. Because both GTP transceivers are configured identically, the protocol template option for GTP1 is set to **Use GTP0 settings**.

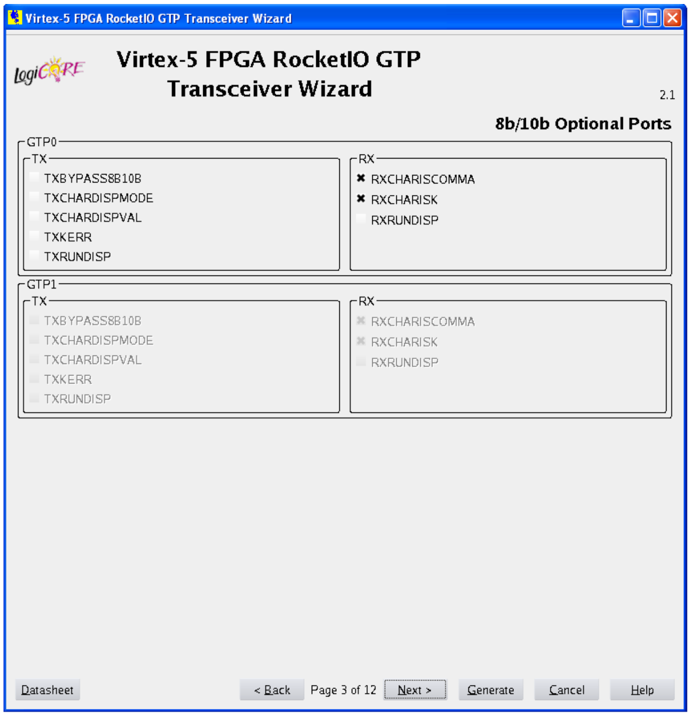
Table 3-4 details the TX Settings and RX Settings options.

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

Option		Description
Line Rate		Allows selection of the optimal line rate based on the shared PMA PLL settings divided by 1, 2, or 4. This is typically set to the value of the Target Line Rate. This option allows the transmit and receive line rates to differ as needed. The XAUI example uses 3.125 Gb/s.
Encoding / Decoding	None	Data stream is passed with no conversion.
	8B/10B	Data stream is passed to an internal 8B/10B encoder /decoder prior to transmission or after receiving.
Data Path Width	8	Sets the transmitter application interface data path width to a single 8-bit byte.
	16	Sets the transmitter application interface data path width to two 8-bit bytes (16 bits).
	10	Sets the transmitter application interface data path width to a single 10-bit byte. This option is only available if internal data width is 10 and no encoding is used.
	20	Sets the transmitter application interface data path width to two 10-bit bytes (20 bits). This option is only available if internal data width is 10 and no encoding is used.

8B/10B Optional Ports

Page 3 of the Wizard (Figure 3-7) is for selecting the 8B/10B-specific optional ports. Placing a check next to one of the listed optional port names makes that port available in the wrapper for use by the application. Table 3-5, page 24 details the available TX and RX 8B/10B optional ports.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1

8b/10b Optional Ports

GTP0

TX
- ☐ TXBYPASS8B10B
- ☐ TXCHARDISPMODE
- ☐ TXCHARDISPVAL
- ☐ TXKERR
- ☐ TXRUNDISP

RX
- ☒ RXCHARISCOMMA
- ☒ RXCHARISK
- ☐ RXRUNDISP

GTP1

TX
- ☐ TXBYPASS8B10B
- ☐ TXCHARDISPMODE
- ☐ TXCHARDISPVAL
- ☐ TXKERR
- ☐ TXRUNDISP

RX
- ☒ RXCHARISCOMMA
- ☒ RXCHARISK
- ☐ RXRUNDISP

Figure 3-7: **RocketIO GTP Wizard Page 3**

Table 3-5: **8B/10B Optional Ports**

Option		Description
TX	TXBYPASS8B10B	Two-bit wide port disables 8B/10B encoder on a per-byte basis. High-order bit affects high-order byte of data path.
	TXCHARDISPMODE	Two-bit wide ports control disparity of outgoing 8B/10B data. High-order bit affects high-order byte of data path.
	TXCHARDISPVAL	
	TXKERR	Two-bit wide port flags invalid K character codes as they are encountered. High-order bit corresponds to high-order byte of data path.
	TXRUNDISP	Two-bit wide port indicates current running disparity of the 8B/10B encoder on a per-byte basis. High-order bit affects high-order byte of data path.

Table 3-5: **8B/10B Optional Ports *(Cont'd)***

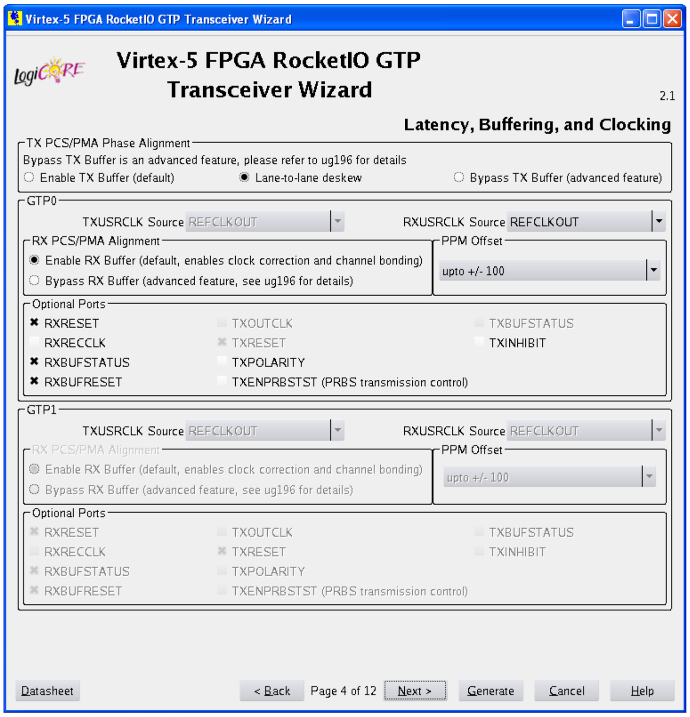
Option		Description
RX	RXCHARISCOMMA	Two-bit wide port flags valid 8B/10B comma characters as they are encountered. High-order bit corresponds to high-order byte of data path.
	RXCHARISK	Two-bit wide port flags valid 8B/10B K characters as they are encountered. High-order bit corresponds to high-order byte of data path.
	RXRUNDISP	Two-bit wide port indicates current running disparity of the 8B/10B decoder on a per-byte basis. High-order bit corresponds to high-order byte of data path.

Latency, Buffering, and Clocking

Page 4 of the Wizard (Figure 3-8) is for controlling latency, buffering, and clocking of the transmitter and receiver.

The **TX PCS/PMA Phase Alignment** setting controls whether the TX buffer is enabled or bypassed with a specific setting for lane-to-lane deskew. This setting affects both GTP transceivers on each tile. Please refer to the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2] for details on this setting.

The XAUI example bypasses the TX Buffer.


Virtex-5 FPGA RocketIO GTP Transceiver Wizard
LogiCORE
Virtex-5 FPGA RocketIO GTP Transceiver Wizard
2.1
Datasheet
< Back
Page 4 of 12
Next >
Generate
Cancel
Help

Latency, Buffering, and Clocking

TX PCS/PMA Phase Alignment
Bypass TX Buffer is an advanced feature, please refer to ug196 for details
○ Enable TX Buffer (default) ● Lane-to-lane deskew ○ Bypass TX Buffer (advanced feature)

GTP0
TXUSRCLK Source REFCLKOUT RXUSRCLK Source REFCLKOUT
RX PCS/PMA Alignment
● Enable RX Buffer (default, enables clock correction and channel bonding)
○ Bypass RX Buffer (advanced feature, see ug196 for details)
PPM Offset
upto +/- 100
Optional Ports
☒ RXRESET □ TXOUTCLK □ TXBUFSTATUS
□ RXRECCLK ☒ TXRESET □ TXINHIBIT
☒ RXBUFSTATUS □ TXPOLARITY
☒ RXBUFRESET □ TXENPRBSTST (PRBS transmission control)

GTP1
TXUSRCLK Source REFCLKOUT RXUSRCLK Source REFCLKOUT
RX PCS/PMA Alignment
● Enable RX Buffer (default, enables clock correction and channel bonding)
○ Bypass RX Buffer (advanced feature, see ug196 for details)
PPM Offset
upto +/- 100
Optional Ports
☒ RXRESET □ TXOUTCLK □ TXBUFSTATUS
□ RXRECCLK ☒ TXRESET □ TXINHIBIT
☒ RXBUFSTATUS □ TXPOLARITY
☒ RXBUFRESET □ TXENPRBSTST (PRBS transmission control)

Figure 3-8: **RocketIO GTP Wizard Page 4**

Table 3-6 details the TXUSRCLK and RXUSRCLK source signal options.

Table 3-6: **TXUSRCLK and RXUSRCLK Source**

Option		Description
TX	TXOUTCLK	TXUSRCLK is driven by TXOUTCLK. This option is not available if the TX Phase Alignment Circuit is used.
	REFCLKOUT	TXUSRCLK is driven by REFCLKOUT. This option is required if the TX Phase Alignment Circuit is used.
RX	TXOUTCLK	RXUSRCLK is driven by TXOUTCLK. This option is not available if the RX Phase Alignment Circuit is used.
	RXRECCLK	RXUSRCLK is driven by RXRECCLK. This option is required if the RX Phase Alignment Circuit is used.
	REFCLKOUT	RXUSRCLK is driven by REFCLKOUT. This option is not available if the RX Phase Alignment Circuit is used.

The RX PCS/PMA Alignment setting controls whether the RX Buffer is enabled or bypassed. Unlike the TX Buffer, this setting is available for both GTP transceivers independently.

The XAUI example enables the RX Buffer.

The PPM Offset setting optimizes the receiver CDR logic for the desired PPM tolerance range. Table 3-7 shows the available PPM Offset settings.

Table 3-7: **PPM Offset**

Option	Description
0 (Synchronous)	Use with synchronous applications (zero tolerance).
upto +/- 50	For applications where clock tolerance is below 50 PPM.
upto +/- 100	For applications where clock tolerance is below 100 PPM.
> +/- 100 or SSC	For applications where clock tolerance is greater than 100 PPM or spread-spectrum clocking is used.

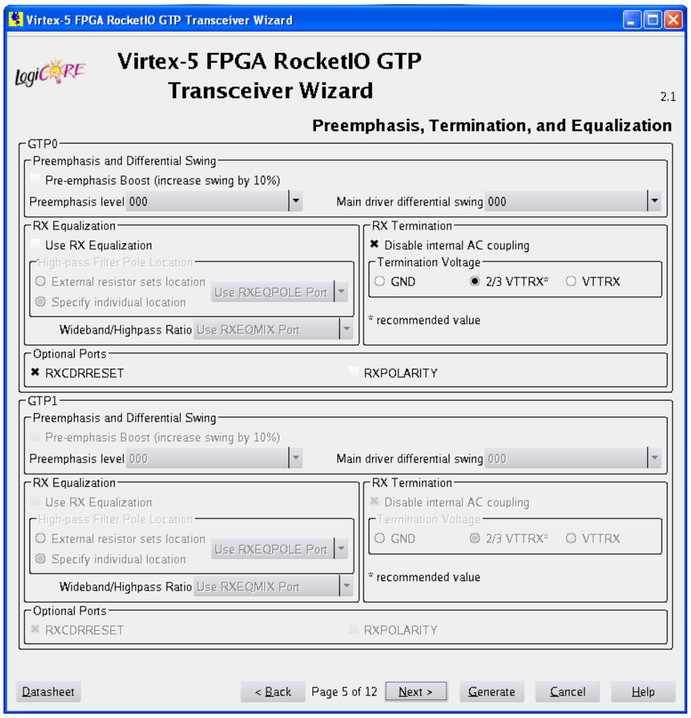
Table 3-8 shows the optional ports available on this page.

Table 3-8: **Latency, Buffering, and Clocking Optional Ports**

Option	Description
RXRESET	Active-High reset signal for the receiver PCS logic.
RXRECCLK	Recovered clock signal from the CDR logic. This option is required when selected as an input to RXUSRCLK.
RXBUFSTATUS	Indicates the condition of the RX elastic buffer. This option is not available when the RX Phase Alignment circuit is used.
RXBUFRESET	Active-High reset signal for the RX elastic buffer logic. This option is not available when the RX Phase Alignment circuit is used.
TXOUTCLK	Parallel clock signal generated by the GTP transceiver. This option is required when selected as an input to either TXUSRCLK or RXUSRCLK. This option is not available when the TX Phase Alignment circuit is used.
TXRESET	Active-High reset signal for the transmitter PCS logic.
TXPOLARITY	Active-High signal to invert the polarity of the transmitter output.
TXENPRBSTST	Two-bit active-High signal to enable the PRBS test pattern generator.
TXBUFSTATUS	Two-bit signal monitors the status of the TX elastic buffer. This option is not available when the TX Phase Alignment circuit is used.
TXINHIBIT	Active-High signal forces transmitter output to steady state.

Preemphasis, Termination, and Equalization

Page 5 of the Wizard (Figure 3-9) is for setting the Preemphasis, Termination, and Equalization options.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1

Preemphasis, Termination, and Equalization

GTP0

Preemphasis and Differential Swing
- □ Pre-emphasis Boost (increase swing by 10%)
- Preemphasis level 000
- Main driver differential swing 000

RX Equalization
- □ Use RX Equalization
- High-pass Filter Pole Location: ○ External resistor sets location ● Specify individual location — Use RXEQPOLE Port
- Wideband/Highpass Ratio Use RXEQMIX Port

RX Termination
- ☒ Disable internal AC coupling
- Termination Voltage: ○ GND ● 2/3 VTTRX* ○ VTTRX
- * recommended value

Optional Ports
- ☒ RXCDRRESET
- □ RXPOLARITY

GTP1

Preemphasis and Differential Swing
- □ Pre-emphasis Boost (increase swing by 10%)
- Preemphasis level 000
- Main driver differential swing 000

RX Equalization
- □ Use RX Equalization
- High-pass Filter Pole Location: ○ External resistor sets location ● Specify individual location — Use RXEQPOLE Port
- Wideband/Highpass Ratio Use RXEQMIX Port

RX Termination
- ☒ Disable internal AC coupling
- Termination Voltage: ○ GND ● 2/3 VTTRX* ○ VTTRX
- * recommended value

Optional Ports
- ☒ RXCDRRESET
- □ RXPOLARITY

Page 5 of 12

Figure 3-9: **RocketIO GTP Wizard Page 5**

Table 3-9 details the Preemphasis and Differential Swing settings.

Table 3-9: **Preemphasis and Differential Swing**

Option	Description
Preemphasis Boost	Increases the output swing to approximately 1V to improve the signal quality on exceptionally attenuated transmission lines.
Preemphasis Level	Specifies the output pre-emphasis setting in 6.5% steps from 0% to approximately 45%. Selecting **Use TXPREEMPHASIS port** enables the optional TXPREEMPHASIS configuration port to dynamically set the pre-emphasis level. The XAUI example uses the default setting of `000` (0%). See the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2] for a table mapping TXPREEMPHASIS value settings to pre-emphasis levels.
Main Driver Differential Swing	Specifies the differential swing level for the transmitter main driver in 100 mV steps from approximately 800 mV to 200 mV. Can also be set to zero. Selecting Use TXDIFFCTRL port enables the optional TXDIFFCTRL configuration port to dynamically set the swing level. The XAUI example uses the default setting `000` (800 mV). See the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2] for a table mapping TXDIFFCTRL value settings to differential swing levels.

Table 3-10 describes the RX Equalization settings.

Table 3-10: **RX Equalization**

Option		Description
Use RX Equalization		Enables the receive equalization logic.
High-Pass Filter Pole Location	External Resistor Sets Location	Enables an external resistor connected via designated pins for use in trimming the high-pass filter.
	Specify Individual Location	Disables the external resistor network and allows selection of a fixed filter pole position from the drop down list. Pole position can be set in 12.5% increments from -37.5%, through zero, to +50%. Selecting **Use RXEQPOLE port** enables the optional **RXEQPOLE** configuration port to dynamically set the pole position.
Wide Band/High-Pass Ratio		Controls the proportion of signal derived from the high-pass filter and from the unfiltered receiver (wide band) when RX Equalization is active. Select a percentage ratio from the drop down list. This selection is disabled because the XAUI protocol file does not have RX Equalization enabled.

Table 3-11 describes the RX Termination settings.

Table 3-11: **RX Termination**

Option	Description
Disable Internal AC Coupling	Bypasses the internal AC coupling capacitor. Use this option for DC coupling applications or for external AC coupling. The XAUI example disables internal AC coupling.
Termination Voltage	Selecting **GND** grounds the internal termination network. Selecting either **2/3 VTTRX** or **VTTRX** applies an internal voltage reference source to the internal termination network at the level specified. The XAUI example uses the **2/3 VTTRX** setting.

The **TX/RX Termination Impedance** option configures the internal termination network to an impedance level of either 50 or 75Ω. The XAUI example uses 50Ω.

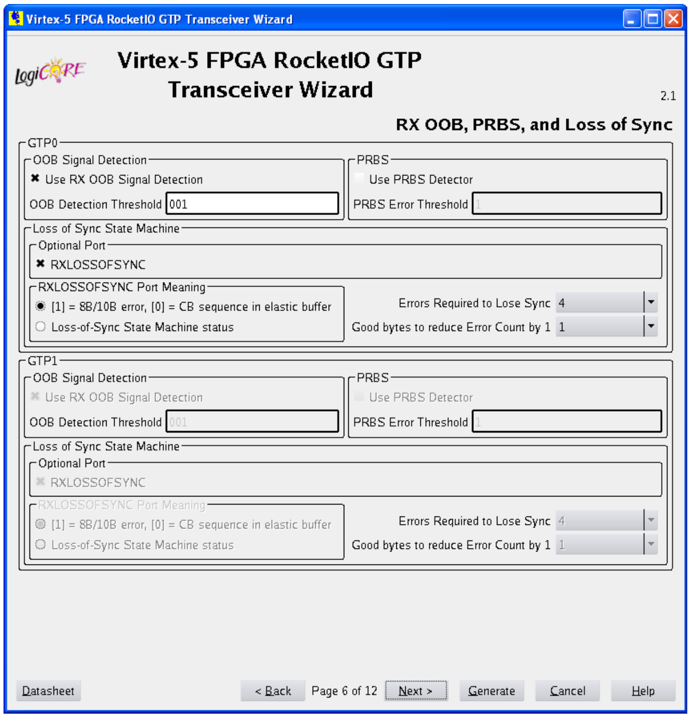
Table 3-12 describes the optional ports.

Table 3-12: **Optional Ports**

Option	Description
RXOVERSAMPLEERR	When oversampling is used, this port indicates when an error is detected by the oversampling logic. This option is available only if the selected line rate for the transceiver is below 500 Mb/s, causing the built-in digital oversampling to be enabled.
RXCDRRESET	Active-High reset signal causes the CDR logic to unlock and return to the shared PLL frequency.
RXPOLARITY	Active-High signal inverts the polarity of the receive data signal.

RX OOB, PRBS, and Loss of Sync

Page 6 of the Wizard (Figure 3-10) is for configuring the RX Out of Band signal (OOB) and PRBS Detector options. Also on this page are the Loss of Sync State Machine settings.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1

RX OOB, PRBS, and Loss of Sync

GTP0

OOB Signal Detection

☒ Use RX OOB Signal Detection

OOB Detection Threshold 001

PRBS

□ Use PRBS Detector

PRBS Error Threshold 1

Loss of Sync State Machine

Optional Port

☒ RXLOSSOFSYNC

RXLOSSOFSYNC Port Meaning

● [1] = 8B/10B error, [0] = CB sequence in elastic buffer

○ Loss-of-Sync State Machine status

Errors Required to Lose Sync 4

Good bytes to reduce Error Count by 1 1

GTP1

OOB Signal Detection

☒ Use RX OOB Signal Detection

OOB Detection Threshold 001

PRBS

□ Use PRBS Detector

PRBS Error Threshold 1

Loss of Sync State Machine

Optional Port

☒ RXLOSSOFSYNC

RXLOSSOFSYNC Port Meaning

● [1] = 8B/10B error, [0] = CB sequence in elastic buffer

○ Loss-of-Sync State Machine status

Errors Required to Lose Sync 4

Good bytes to reduce Error Count by 1 1

Page 6 of 12

Figure 3-10: **RocketIO GTP Wizard Page 6**

Table 3-13 shows the OOB signal detection options. Table 3-14, page 31 details the PRBS settings. The Loss-of-Sync State Machine settings are described in Table 3-15, page 31.

Table 3-13: **OOB Signal Detection**

Option	Description
Use RX OOB Signal Detection	Enables the internal Out-of-Band signal detector (OOB). OOB signal detection is used for SATA.
OOB Detection Threshold	Specifies a binary value between `000` and `111`, which represents a differential receive-signal voltage level. When the signal drops below this level it is determined to be an OOB signal. This option is not available if the Use RX OOB Signal Detection option is not selected. See the *Virtex-5 Family Overview* [Ref 1] for more information about the OOB Detection Threshold levels.

Table 3-14: **PRBS**

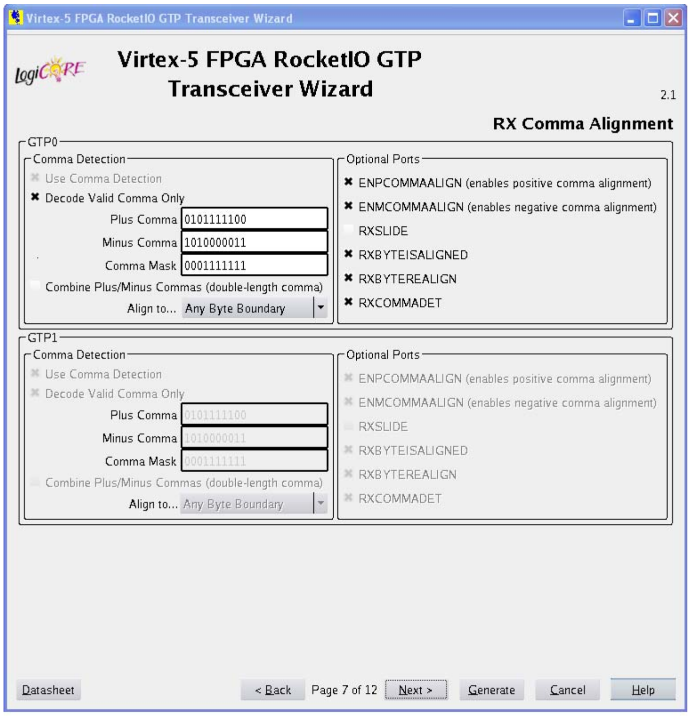
Option	Description
Use PRBS Detector	Enables the internal Pseudo Random Bitstream Sequence detector (PRBS). This can be used by an application to implement a built-in self-test feature.
PRBS Error Threshold	Specifies an integer value between `0` and `4294967295` (32-bit hex value), which represents an error count threshold. When the number of errors detected exceeds the specified threshold an error signal is asserted. This option is not available if the **Use PRBS Detector option** is not selected.

Table 3-15: **Loss of Sync State Machine**

Option		Description
RXLOSSOFSYNC Optional Port		Two-bit multi-purpose status port. The meaning of the bits is determined by the settings below.
RXLOSSOFSYNC Port Meaning	[1] = 8B/10B Error, [0] = CB Sequence in Elastic Buffer	Bit 1 of the **RXLOSSOFSYNC** status port indicates the detection of an 8B/10B coding error. Bit 0 indicates a Channel Bonding sequence is present in the receive elastic buffer.
	Loss-of-Sync State Machine Status	Bit 0 of the **RXLOSSOFSYNC** status port indicates sync state is active due to channel bonding or realignment. Bit 1 indicates sync lost due to invalid characters or reset.
Errors Required to Lose Sync		Integer value between 4 and 512 representing the count of invalid characters received, above which sync is determined to be lost. The XAUI example uses 128.
Good Bytes to Reduce Error Count by 1		Integer value between 1 and 128 representing the number of consecutive valid characters needed to cancel out the appearance of one invalid character. The XAUI example uses 8.

RX Comma Alignment

Page 7 of the Wizard (Figure 3-11) allows configuration of the RX comma detection and alignment logic. The settings are detailed in Table 3-16, page 33.


Virtex-5 FPGA RocketIO GTP Transceiver Wizard
LogiCORE
Virtex-5 FPGA RocketIO GTP Transceiver Wizard
2.1
Datasheet
< Back
Page 7 of 12
Next >
Generate
Cancel
Help

RX Comma Alignment

GTP0

Comma Detection

- ☒ Use Comma Detection
- ☒ Decode Valid Comma Only
- Plus Comma: 0101111100
- Minus Comma: 1010000011
- Comma Mask: 0001111111
- □ Combine Plus/Minus Commas (double-length comma)
- Align to... Any Byte Boundary

Optional Ports

- ☒ ENPCOMMAALIGN (enables positive comma alignment)
- ☒ ENMCOMMAALIGN (enables negative comma alignment)
- □ RXSLIDE
- ☒ RXBYTEISALIGNED
- ☒ RXBYTEREALIGN
- ☒ RXCOMMADET

GTP1

Comma Detection

- ☒ Use Comma Detection
- ☒ Decode Valid Comma Only
- Plus Comma: 0101111100
- Minus Comma: 1010000011
- Comma Mask: 0001111111
- □ Combine Plus/Minus Commas (double-length comma)
- Align to... Any Byte Boundary

Optional Ports

- ☒ ENPCOMMAALIGN (enables positive comma alignment)
- ☒ ENMCOMMAALIGN (enables negative comma alignment)
- □ RXSLIDE
- ☒ RXBYTEISALIGNED
- ☒ RXBYTEREALIGN
- ☒ RXCOMMADET

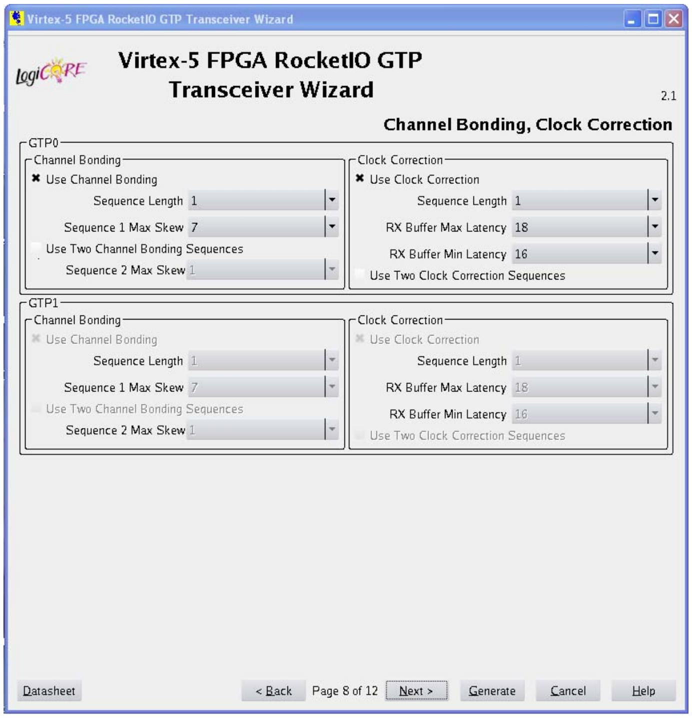
Figure 3-11: **RocketIO GTP Wizard Page 7**

Table 3-16: **Comma Detection**

Option		Description
Use Comma Detection		Enables receive comma detection. Used to identify comma characters and SONET framing characters in the data stream.
Decode Valid Comma Only		When receive comma detection is enabled, limits the detection to specific defined comma characters.
Plus Comma		10-bit binary pattern representing the positive-disparity comma character to match. The right-most bit of the pattern is the first bit to arrive serially. The XAUI example uses `0101111100`.
Minus Comma		10-bit binary pattern representing the negative-disparity comma character to match. The right-most bit of the pattern is the first bit to arrive serially. The XAUI example uses `1010000011`.
Comma Mask		10-bit binary pattern representing the mask for the comma match patterns. A `1` bit indicates the corresponding bit in the comma patterns is to be matched. A `0` bit indicates "don't care" for the corresponding bit in the comma patterns. The XAUI example matches the lower seven bits (`0001111111`).
Combine Plus/Minus Commas		Causes the two comma definition patterns to be combined into a single 20-bit pattern which must be contiguously matched in the data stream. The Mask value remains 10-bits and is duplicated for the upper and lower 10-bit portions of the extended pattern. This option can be used to search for SONET framing character patterns.
Align to...	Any Byte Boundary	When a comma is detected, the data stream is aligned using the comma pattern to the nearest byte boundary.
	Even Byte Boundaries	When a comma is detected, the data stream is aligned using the comma pattern to the nearest even byte boundary. This option is available only for 16 and 20 bit RX data interfaces.
Optional Ports	ENPCOMMAALIGN	Active-High signal which enables the byte boundary alignment process when Plus Comma pattern is detected.
	ENMCOMMAALIGN	Active-High signal which enables the byte boundary alignment process when Minus Comma pattern is detected.
	RXSLIDE	Active-High signal that causes the byte alignment to be adjusted by one bit with each assertion. Takes precedence over normal comma alignment.
	RXBYTEISALIGNED	Active-High signal indicating that the parallel data stream is aligned to byte boundaries.
	RXBYTEREALIGN	Active-High signal indicating that byte alignment has changed with a recent comma detection. Note that data errors can occur with this condition.
	RXCOMMADET	Active-High signal indicating the comma alignment logic has detected a comma pattern in the data stream.

Channel Bonding, Clock Correction

Page 8 of the Wizard (Figure 3-12) enables Channel Bonding and Clock Correction as detailed in Table 3-17, page 35.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Virtex-5 FPGA RocketIO GTP Transceiver Wizard

2.1

Channel Bonding, Clock Correction

GTP0

Channel Bonding

☒ Use Channel Bonding

Sequence Length: 1

Sequence 1 Max Skew: 7

□ Use Two Channel Bonding Sequences

Sequence 2 Max Skew: 1

Clock Correction

☒ Use Clock Correction

Sequence Length: 1

RX Buffer Max Latency: 18

RX Buffer Min Latency: 16

□ Use Two Clock Correction Sequences

GTP1

Channel Bonding

☒ Use Channel Bonding

Sequence Length: 1

Sequence 1 Max Skew: 7

□ Use Two Channel Bonding Sequences

Sequence 2 Max Skew: 1

Clock Correction

☒ Use Clock Correction

Sequence Length: 1

RX Buffer Max Latency: 18

RX Buffer Min Latency: 16

□ Use Two Clock Correction Sequences

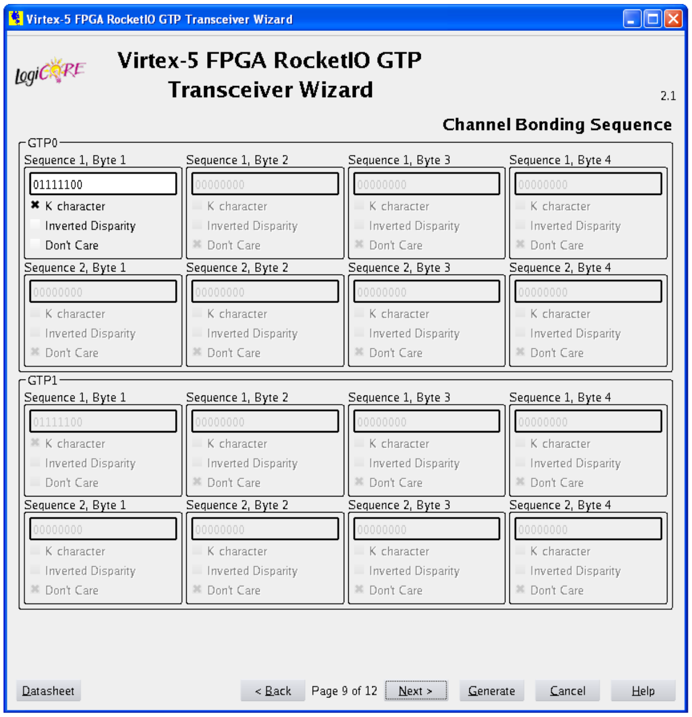
Figure 3-12: **RocketIO GTP Wizard Page 8**

Table 3-17: **Channel Bonding and Clock Correction**

Option	Description
Use Channel Bonding	Enables receiver channel bonding logic using unique character sequences. When recognized, these sequences allow for adding or deleting characters in the receive buffer to byte-align multiple data transceivers.
Sequence Length	Select from the drop down list the number of characters in the unique channel bonding sequence. The XAUI example uses **1**.
Sequence 1 Max Skew	Select from the drop down list the maximum skew in characters that can be handled by channel bonding. Must always be less than the minimum distance between channel bonding sequences. The XAUI example uses **7**.
Use Two Channel Bonding Sequences	Activates the optional second Channel Bonding sequence. Detection of either sequence triggers channel bonding.
Sequence 2 Max Skew	Same as Sequence 1 Max Skew.
Use Clock Correction	Enables receiver clock correction logic using unique character sequences. When recognized, these sequences allow for adding or deleting characters in the receive buffer to prevent buffer underflow/overflow due to small differences in the transmit/receive clock frequencies.
Sequence Length	Select from the drop down list the number of characters (subsequences) in the unique clock correction sequence. The XAUI example uses **1**.
RX Buffer Max Latency	Select from the drop down list the maximum number of characters to permit in the receive buffer before clock correction will attempt to delete incoming clock correction sequences. This also determines the maximum latency of the receive buffer in RXUSRCLK cycles. The XAUI example uses **18**.
RX Buffer Min Latency	Select from the drop down list the minimum number of characters to permit in the receive buffer before clock correction will attempt to add extra clock correction sequences to the receive buffer. This also determines the minimum latency of the receive buffer in RXUSRCLK cycles. The XAUI example uses **16**.
Use Two Clock Correction Sequences	Activates the optional second Clock Correction sequence. Detection of either sequence triggers clock correction.

Channel Bonding Sequence

Page 9 (Figure 3-13) defines the Channel Bonding sequence(s). Similarly, Page 10 (Figure 3-14, page 37) defines the Clock Correction sequence(s). Table 3-18 describes their use.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1

Channel Bonding Sequence

GTP0

Sequence 1, Byte 1	Sequence 1, Byte 2	Sequence 1, Byte 3	Sequence 1, Byte 4
01111100	00000000	00000000	00000000
☒ K character	□ K character	□ K character	□ K character
□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity
□ Don't Care	☒ Don't Care	☒ Don't Care	☒ Don't Care

Sequence 2, Byte 1	Sequence 2, Byte 2	Sequence 2, Byte 3	Sequence 2, Byte 4
00000000	00000000	00000000	00000000
□ K character	□ K character	□ K character	□ K character
□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity
☒ Don't Care	☒ Don't Care	☒ Don't Care	☒ Don't Care

GTP1

Sequence 1, Byte 1	Sequence 1, Byte 2	Sequence 1, Byte 3	Sequence 1, Byte 4
01111100	00000000	00000000	00000000
☒ K character	□ K character	□ K character	□ K character
□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity
□ Don't Care	☒ Don't Care	☒ Don't Care	☒ Don't Care

Sequence 2, Byte 1	Sequence 2, Byte 2	Sequence 2, Byte 3	Sequence 2, Byte 4
00000000	00000000	00000000	00000000
□ K character	□ K character	□ K character	□ K character
□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity	□ Inverted Disparity
☒ Don't Care	☒ Don't Care	☒ Don't Care	☒ Don't Care

Page 9 of 12

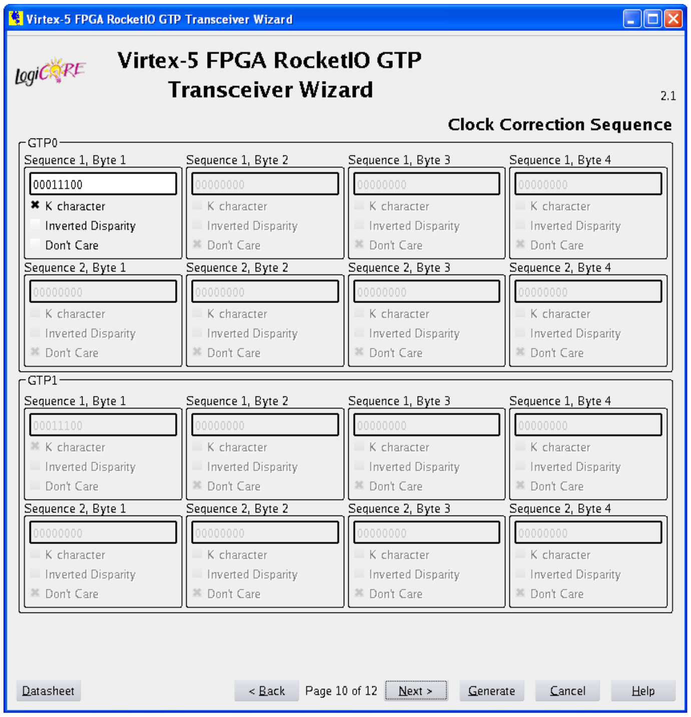
Figure 3-13: **RocketIO GTP Wizard Page 9**

Table 3-18: **Channel Bonding and Clock Correction Sequences**

Option	Description
Byte (Symbol)	Set each symbol to match the pattern the protocol requires. The XAUI sequence length is 8 bits. `01111100` is used for Channel Bonding. `00011100` is used for Clock Correction. The other symbols are disabled because the Sequence Length is set to 1.
K Character	This option is available when 8B/10B decoding is selected. When checked, as is the case for XAUI, the symbol is an 8B/10B K character.
Inverted Disparity	Some protocols with 8B/10B decoding use symbols with deliberately inverted disparity. This option should be checked when such symbols are expected in the sequence.
Don't Care	Multiple-byte sequences can have wild card symbols by checking this option. Unused bytes in the sequence automatically have this option set.

Clock Correction Sequence

Page 10 (Figure 3-14) defines the Clock Correction sequence. See Table 3-18, page 36 for details.


Virtex-5 FPGA RocketIO GTP Transceiver Wizard
LogiCORE
Virtex-5 FPGA RocketIO GTP Transceiver Wizard
2.1
Datasheet
< Back
Page 10 of 12
Next >
Generate
Cancel
Help

Clock Correction Sequence

GTP0

Sequence 1, Byte 1	Sequence 1, Byte 2	Sequence 1, Byte 3	Sequence 1, Byte 4
00011100 ☒ K character ☐ Inverted Disparity ☐ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care

Sequence 2, Byte 1	Sequence 2, Byte 2	Sequence 2, Byte 3	Sequence 2, Byte 4
00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care

GTP1

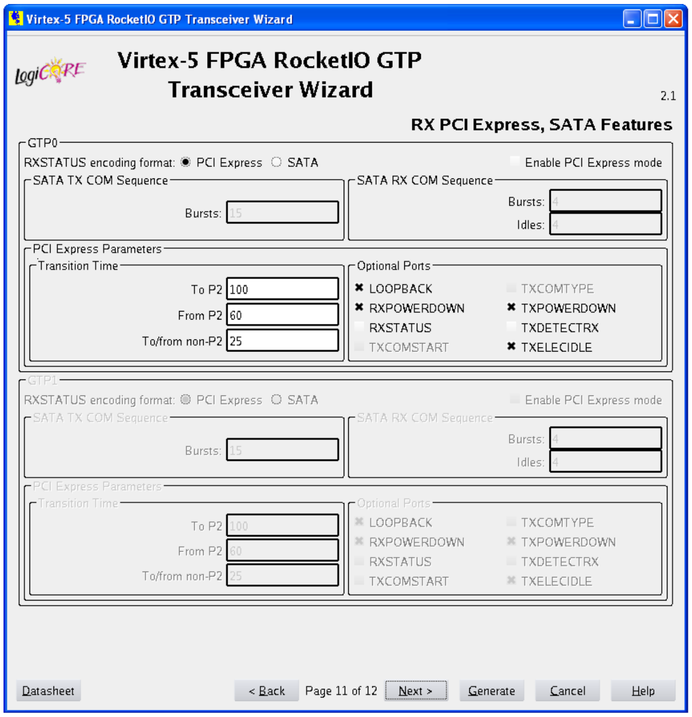
Sequence 1, Byte 1	Sequence 1, Byte 2	Sequence 1, Byte 3	Sequence 1, Byte 4
00011100 ☒ K character ☐ Inverted Disparity ☐ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care

Sequence 2, Byte 1	Sequence 2, Byte 2	Sequence 2, Byte 3	Sequence 2, Byte 4
00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care	00000000 ☐ K character ☐ Inverted Disparity ☒ Don't Care

Figure 3-14: **RocketIO GTP Wizard Page 10**

RX PCI Express, SATA Features

Page 11 (Figure 3-15) configures the receiver PCI Express® and Serial ATA as detailed in Table 3-19, page 39 and Table 3-20, page 40.



Virtex-5 FPGA RocketIO GTP Transceiver Wizard

Virtex-5 FPGA RocketIO GTP Transceiver Wizard 2.1

RX PCI Express, SATA Features

GTP0

RXSTATUS encoding format: ● PCI Express ○ SATA   □ Enable PCI Express mode

SATA TX COM Sequence
Bursts: 15

SATA RX COM Sequence
Bursts: 4
Idles: 4

PCI Express Parameters

Transition Time
To P2: 100
From P2: 60
To/from non-P2: 25

Optional Ports
☒ LOOPBACK   □ TXCOMTYPE
☒ RXPOWERDOWN   ☒ TXPOWERDOWN
□ RXSTATUS   □ TXDETECTRX
□ TXCOMSTART   ☒ TXELECIDLE

GTP1

RXSTATUS encoding format: ● PCI Express ○ SATA   □ Enable PCI Express mode

SATA TX COM Sequence
Bursts: 15

SATA RX COM Sequence
Bursts: 4
Idles: 4

PCI Express Parameters

Transition Time
To P2: 100
From P2: 60
To/from non-P2: 25

Optional Ports
☒ LOOPBACK   □ TXCOMTYPE
☒ RXPOWERDOWN   ☒ TXPOWERDOWN
□ RXSTATUS   □ TXDETECTRX
□ TXCOMSTART   ☒ TXELECIDLE

Figure 3-15: **RocketIO GTP Wizard Page 11**

Table 3-19: **Receiver Serial ATA Options**

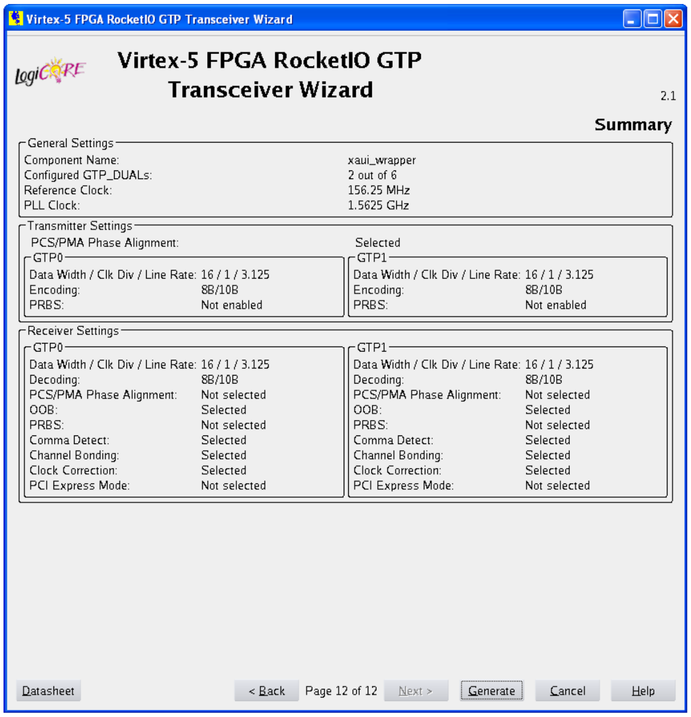
Option		Description
RXSTATUS Encoding Format	PCI Express	Default setting. The **RXSTATUS** optional port presents status information for the PIPE interface. See the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2] for more details.
	SATA	The **RXSTATUS** optional port presents codes for the SATA COM sequence status.
Enable PCI Express Mode		Selecting this option enables certain operations specific to PCI Express, including enabling options for PCI Express powerdown modes and PCI Express channel bonding. This option should be activated whenever the transceiver is used for PCI Express. This option is not available if **RXSTATUS encoding format** is set to **SATA**.
SATA TX COM Sequence	Bursts	Integer value between 0 and 15 indicating the number of busts to define a TX COM sequence. This option is not available if **RXSTATUS encoding format** is set to **PCI Express**.
SATA RX COM Sequence	Bursts	Integer value between 0 and 7 indicating the number of Burst sequences to declare a COM match. This value defaults to 4, which is the burst count specified in the SATA specification for COMINIT, COMRESET, and COMWAKE.
	Idles	Integer value between 0 and 7 indicating the number of Idle sequences to declare a COM match. Each Idle is an OOB signal with a length that matches COMINIT/COMRESET or COMWAKE. This value defaults to 3 per the SATA specification. This option is not available if **RXSTATUS encoding format** is set to **PCI Express**.

Table 3-20: **PCI Express Parameters**

Option		Description
Transition Time	To P2	Integer value between 0 and 65,535. Sets a counter to determine the transition time to the P2 power state for PCI Express. See the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2] for details on determining the time value for each count. The XAUI example does not require this feature and uses the default setting of **100**.
	From P2	Integer value between 0 and 65,535. Sets a counter to determine the transition time from the P2 power state for PCI Express. See the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2] for details on determining the time value for each count. The XAUI example does not require this feature and uses the default setting of **60**.
	To/From non-P2	Integer value between 0 and 65,535. Sets a counter to determine the transition time to or from power states other than P2 for PCI Express. See the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2] for details on determining the time value for each count. The XAUI example does not require this feature and uses the default setting of **25**. This option is not available if **RXSTATUS encoding format** is set to **SATA**.
Optional Ports	LOOPBACK	3-bit signal to enable the various data loopback modes for testing.
	RXPOWERDOWN	2-bit PCI Express-compliant receiver powerdown control signal.
	RXSTATUS	3-bit receiver status signal. The encoding of this signal is dependent on the setting of **RXSTATUS encoding format**.
	TXCOMSTART	Active-High signal initiates the transmission of the SATA COM sequence selected by the setting of **TXCOMTYPE**. This option is not available if **RXSTATUS encoding format** is set to **PCI Express**. Activate the **RXSTATUS** optional port when using this option.
	TXCOMTYPE	Active-High signal selects SATA COMWAKE sequence when asserted, otherwise selects COMINIT. The sequence is initiated upon assertion of **TXCOMSTART**. This option is not available if **RXSTATUS encoding format** is set to **PCI Express**.
	TXPOWERDOWN	2-bit is compliant with the PCI Express transmitter powerdown control signal.
	TXDETECTRX	PIPE interface for PCI Express specification-compliant control signal. Activates the PCI Express receiver detection feature. Function depends on the state of **TXPOWERDOWN**, **RXPOWERDOWN**, **TXELECIDLE**, **TXCHARDISPMODE**, and **TXCHARDISPVAL**. This port is not available if **RXSTATUS encoding format** is set to **SATA**.
	TXELECIDLE	Drives the transmitter to an electrical idle state (no differential voltage). In PCI Express mode this is used for electrical idle modes. Function depends on the state of **TXPOWERDOWN**, **RXPOWERDOWN**, **TXELECIDLE**, **TXCHARDISPMODE**, and **TXCHARDISPVAL**.

Summary Page

Page 12 of the Wizard (Figure 3-16) is a summary of the selected configuration parameters. After reviewing the settings, click **Generate** to exit and generate the wrapper.



General Settings

Component Name:	xaui_wrapper
Configured GTP_DUALs:	2 out of 6
Reference Clock:	156.25 MHz
PLL Clock:	1.5625 GHz

Transmitter Settings

PCS/PMA Phase Alignment: Selected

GTP0		GTP1	
Data Width / Clk Div / Line Rate:	16 / 1 / 3.125	Data Width / Clk Div / Line Rate:	16 / 1 / 3.125
Encoding:	8B/10B	Encoding:	8B/10B
PRBS:	Not enabled	PRBS:	Not enabled

Receiver Settings

GTP0		GTP1	
Data Width / Clk Div / Line Rate:	16 / 1 / 3.125	Data Width / Clk Div / Line Rate:	16 / 1 / 3.125
Decoding:	8B/10B	Decoding:	8B/10B
PCS/PMA Phase Alignment:	Not selected	PCS/PMA Phase Alignment:	Not selected
OOB:	Selected	OOB:	Selected
PRBS:	Not selected	PRBS:	Not selected
Comma Detect:	Selected	Comma Detect:	Selected
Channel Bonding:	Selected	Channel Bonding:	Selected
Clock Correction:	Selected	Clock Correction:	Selected
PCI Express Mode:	Not selected	PCI Express Mode:	Not selected

Figure 3-16: **RocketIO GTP Wizard Page 12**

Chapter 4

Quick Start Example Design

Overview

This chapter introduces the example design that is included with the Virtex-5 FPGA RocketIO GTP Transceiver Wizard. The quick start instructions are a step-by-step procedure for generating an Virtex-5 FPGA RocketIO GTP Transceiver Wizard, implementing the core in hardware using the accompanying example design, and simulating the core with the provided demonstration test bench (demo_tb). For detailed information about the example design provided with the Virtex-5 FPGA RocketIO GTP Transceiver Wizard core, see Chapter 5, "Detailed Example Design."

Functional Simulation of the Example Design

Using ModelSim

The Virtex-5 FPGA RocketIO GTP Transceiver Wizard provides a quick way to simulate and observe the behavior of the wrapper using the provided example design. 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 12.1, available in the ISE Software Documentation [Ref 4], 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>/smartmodel/<OS>/wrappers/mtiverilog`

VHDL	UNISIM	`<Xilinx dir>/vhdl/src/unisims`
		`<Xilinx dir>/smartmodel/<OS>/wrappers/mtivhdl`

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

The Virtex-5 FPGA RocketIO GTP Transceiver 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 XAUI 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

```
ms-dos> cd xaui_wrapper\implement
ms-dos> .\implement.bat
```

For Linux

```
linux-shell% cd xaui_wrapper/implement
linux-shell% ./implement.sh
```

Note: Substitute *Component Name* string for "xaui_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.

Using ChipScope Pro Cores with the Virtex-5 FPGA RocketIO GTP Transceiver Wizard Wrapper

The ChipScope™ Pro ICON and VIO cores aid in debugging and validating the design in board. To assist with debugging, these cores are provided with the Virtex-5 FPGA RocketIO GTP Transceiver Wizard wrapper, which is enabled by setting USE_CHIPSCOPE as 1 in the <component_name>_top file.

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®-5 FPGA RocketIO™ GTP 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 project files.

Table 5-1: **Project Directory**

Name	Description
`<component_name>.v[hd]`	Main GTP transceiver wrapper. Instantiates individual GTP tile wrappers. For use in the target design.
`<component_name>.[veo	vho]`	GTP Wrapper files instantiation templates. Includes templates for the GTP Wrapper module, the IBUFDS_GTPE1, and essential GTP support modules (such as TX_SYNC).
`<component_name>.xco`	Log file from the CORE Generator tool describing which options were used to generate the GTP Wrapper. An XCO file is generated by the 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>_tile.v[hd]`	Individual GTPE1 transceiver wrapper to be instantiated in the main GTP transceiver wrapper. Instantiates the selected GTPE1 transceivers with settings for the selected protocol.

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: **GTP Wrapper Component Name**

Name	Description
<project_dir>/<component_name>	
`v5_gtpwizard_readme.txt`	Release notes for the GTP Wizard.
`<component_name>.pf`	Protocol description for the selected protocol from the GTP 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	
`v5_gtpwizard_ds590.pdf`	*Virtex-5 FPGA RocketIO GTP Transceiver Wizard Data Sheet*
`v5_gtpwizard_gsg188.pdf`	*LogiCORE IP Virtex-5 FPGA RocketIO GTP Transceiver Wizard 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	
`frame_check.v[hd]`	Frame-check logic to be instantiated in the example design.
`frame_gen.v[hd]`	Frame-generator logic to be instantiated in the example design.
`gtp_attributes.ucf`	Constraints file containing the GTP attributes generated by the GTP Wizard GUI settings.
`tx_sync.v[hd]`	TX sync logic module to be instantiated in the example design. Performs phase synchronization for all active TX data paths. Available for use in the target design.
`<component_name>_top.ucf`	Constraint file for mapping the GTP Wrapper example design onto a Virtex-5 device.
`<component_name>_top.v[hd]`	Top-level example design. Contains GTP transceiver wrapper, reset logic, and instantiations for frame generator, frame-checker, and TX sync logic. Also contains definitions for test frame data and ChipScope™ Pro module instantiation. See Figure 3-1, page 15.

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	
`chipscope_project.cpj`	Chipscope project file.
`data_vio.ngc`	ChipScope VIO netlist.
`icon.ngc`	ChipScope ICON netlist.
`ila.ngc`	ChipScope ILA netlist.
`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.
`implement_synplify.bat`	A Windows batch file that processes the example design through Synplify synthesis and the Xilinx tool flow.
`implement_synplify.sh`	A 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`	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 44.
`sim_reset_mgt_model.vhd`	Reset module for VHDL required for emulating the GSR pulse at the beginning of functional simulation in order to correctly reset the VHDL MGT smart model.

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_isim.sh`	ISim simulation script.
`simulate_mti.do`	ModelSim simulation script.
`wave_isim.tcl`	Script for adding GTP Wrapper signals to the ISim wave viewer.
`wave_mti.do`	Script for adding GTP Wrapper signals to the ModelSim wave viewer.

Back to Top

Example Design

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

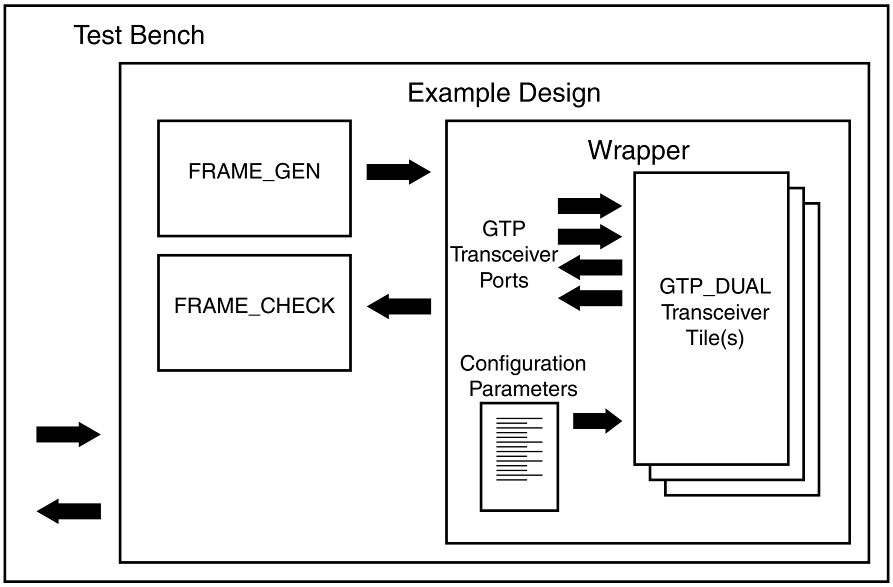


Figure 5-1: **Wrapper Block Diagram**

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 frame generator instantiation. 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.

If comma alignment is enabled, the comma character will be placed within the counting pattern. Similarly, if channel bonding is enabled, the channel bonding sequence would be interspersed within the counting pattern. The frame check works by first scanning the received data for the START_OF_PACKET_CHAR. In 8B/10B designs, this is the comma alignment character. 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 RXUSRCLK2 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.

For 64B/66B and 64B/67B example designs, the frame generator has scrambler logic while the frame checker has descrambler and block synchronization logic.

If the TX buffer is bypassed, the TX_SYNC module is instantiated in the example design and connected to the wrapper. The module performs the TX phase alignment procedure outlined in the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide*[Ref 2].

Similarly, if the X buffer is bypassed, the RX_SYNC module is instantiated in the example design and connected to the wrapper. The RX_SYNC module demonstrates the RX phase-alignment procedure outlined in the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* [Ref 2].

The example design also demonstrates how to properly connect clocks to GTP transceiver ports TXUSRCLK, TXUSRCLK2, RXUSRCLK and RXUSRCLK2. Properly configured DCM (Digital Clock manager), PLL (Phase lock loop) wrappers are also provided if they are required to generate user clocks for the instantiated GTP transceivers.

The example design may be synthesized using XST or Synplify Pro, implemented with ISE® software and then observed in hardware using the Chipscope Pro tools. RX output ports such as RXDATA can be observed on the Chipscope ILA core while input ports can be controlled from the Chipscope VIO core. A Chipscope project file is also included with each example design.

For the example design to work properly in simulation or in hardware, both the transmit and receive side need to be configured with the same line rate, encoding and datapath width in the GUI.

Example Design Hierarchy

The hierarchy for the design used in this example is as follows:

```
example_tb
|___example_mgt_top
    |___mgt_userclk_source_pll
    |___ibufds
    |___frame_gen
    |___frame_check
    |___tx_sync
    |___rocketio_wrapper
        |___rocketio_wrapper_tile
            |___gtp_dual
```

Appendix A

References

Documents specific to Virtex®-5 FPGAs:

1. DS100: *Virtex-5 Family Overview*

Documents specific to RocketIO™ transceivers:

2. UG196: *Virtex-5 FPGA RocketIO GTP Transceiver User Guide*
3. DS590: *Virtex-5 FPGA RocketIO GTP Transceiver Wizard Data Sheet*

Xilinx Tools and Solutions

4. ISE® Software Manuals