Virtex-4 RocketIO Multi-Gigabit Transceiver

User Guide

UG076 (v1.3) July 1, 2005

XILINX®

Xilinx does not assume any liability arising out of the application or use of the Specification; nor does Xilinx convey any license under its patents, copyrights, or any rights of others. You are responsible for obtaining any rights you may require for your use or implementation of the Specification. Xilinx reserves the right to make changes, at any time, to the Specification as deemed desirable in the sole discretion of Xilinx. Xilinx assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. Xilinx will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Specification.

THE SPECIFICATION IS PROVIDED “AS IS" WITH ALL FAULTS, AND THE ENTIRE RISK AS TO ITS FUNCTION AND IMPLEMENTATION IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XILINX, OR ITS AGENTS OR EMPLOYEES. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE SPECIFICATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOST DATA AND LOST PROFITS, ARISING FROM OR RELATING TO YOUR USE OF THE SPECIFICATION, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE TOTAL CUMULATIVE LIABILITY OF XILINX IN CONNECTION WITH YOUR USE OF THE SPECIFICATION, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL IN NO EVENT EXCEED THE AMOUNT OF FEES PAID BY YOU TO XILINX HEREUNDER FOR USE OF THE SPECIFICATION. YOU ACKNOWLEDGE THAT THE FEES, IF ANY, REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XILINX WOULD NOT MAKE AVAILABLE THE SPECIFICATION TO YOU WITHOUT THESE LIMITATIONS OF LIABILITY.

The Specification is not designed or intended for use in the development of on-line control equipment in hazardous environments requiring fail-safe controls, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, or weapons systems (“High-Risk Applications”). Xilinx specifically disclaims any express or implied warranties of fitness for such High-Risk Applications. You represent that use of the Specification in such High-Risk Applications is fully at your risk.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
03/01/05	1.0	Xilinx Initial Release.
03/10/05	1.1	Modified “Power Supply Requirements” in Chapter 6 and Table 6-1, page 130.
04/07/05	1.2	General typographical edits. Revised Table 2-2, Table 2-7, Figure 2-8, Figure 2-9, Figure 2-11, Figure 2-13, Figure 6-2, Figure 6-6, and Figure D-2. Added “Resetting the Transceiver,” page 65 and Figure 2-16. Edited Table 4-1, Table 4-3, Table 4-5, Table 7-2, Table 7-3, Table 7-4, Table 7-5, Table 7-6, Table A-1, Table C-14, and Table C-28.
07/01/05	1.3	Changes in Figure 2-3, Figure 2-14, Figure 3-15, Figure 4-9, Figure 6-2 and Figure 6-6. Revised Table 3-23 and Table 3-24. Added Table 5-5, revised Table 5-5. Changes to Table 7-3, Table 7-4, Table 7-5, Table 7-6. For clarity, revised all the notes in the tables in Appendix C. Added a default value to DCDR_FILTER in Appendix F.

Table of Contents

Preface: About This Guide

Chapter 1: RocketIO MGT Overview

Chapter 2: Clocking and Timing Considerations

Chapter 3: PCS Digital Design Considerations

Chapter 4: PMA Analog Design Considerations

Chapter 5: Cyclic Redundancy Check (CRC)

Chapter 6: Analog and Board Design Considerations

Chapter 7: Simulation and Implementation

Appendix A: RocketIO MGT Timing Model

Appendix B: 8B/10B Valid Characters

Appendix C: Dynamic Configuration Bus

Appendix D: Virtex-II Pro/Virtex-II Pro X to Virtex-4 RocketIO Transceiver Design Migration

Appendix E: Related Online Documents

Appendix F: Restricted/Reserved Attributes

Schedule of Figures

Chapter 1: RocketIO MGT Overview

Chapter 2: Clocking and Timing Considerations

Chapter 3: PCS Digital Design Considerations

Chapter 4: PMA Analog Design Considerations

Chapter 5: Cyclic Redundancy Check (CRC)

Chapter 6: Analog and Board Design Considerations

Chapter 7: Simulation and Implementation

Appendix A: RocketIO MGT Timing Model

Appendix B: 8B/10B Valid Characters

Appendix C: Dynamic Configuration Bus

Appendix D: Virtex-II Pro/Virtex-II Pro X to Virtex-4 RocketIO Transceiver Design Migration

Appendix E: Related Online Documents

Appendix F: Restricted/Reserved Attributes

Schedule of Tables

Chapter 1: RocketIO MGT Overview

Chapter 2: Clocking and Timing Considerations

Chapter 3: PCS Digital Design Considerations

Chapter 4: PMA Analog Design Considerations

Chapter 5: Cyclic Redundancy Check (CRC)

Chapter 6: Analog and Board Design Considerations

Chapter 7: Simulation and Implementation

Appendix A: RocketIO MGT Timing Model

Appendix B: 8B/10B Valid Characters

Appendix C: Dynamic Configuration Bus

Appendix D: Virtex-II Pro/Virtex-II Pro X to Virtex-4 RocketIO Transceiver Design Migration

Appendix E: Related Online Documents

Appendix F: Restricted/Reserved Attributes

About This Guide

The *Virtex™-4 RocketIO™ MGT User Guide* provides the product designer with the detailed technical information needed to successfully implement the RocketIO MGT in Virtex-4 Platform FPGA designs. For information on the Virtex™-II Pro RocketIO and the Virtex-II Pro X RocketIO X transceivers, see the RocketIO Transceiver User Guide and the RocketIO X Transceiver User Guide.

MGT Features

RocketIO MGTs have flexible, programmable features that allow a multi-gigabit serial transceiver to be easily integrated into any Virtex-4 design:

- 622 Mb/s to 10.3125 Gb/s data rates
- 8 to 24 transceivers per FPGA
- 3-tap transmitter pre-emphasis (pre-equalization)
- Receiver continuous time equalization
- Decision Feedback Equalizer (DFE) equalization for legacy backplane applications
- Optional on-chip AC coupled receiver
- Digital oversampled receiver for data rates up to 1.25 Gb/s
- Receiver signal detect and loss of signal indicator and out-of-band (OOB) signal receiver
- Transmit driver idle state for OOB signaling (both outputs at V_{CM})
- 8B/10B or 64B/66B encoding, or no data encoding (pass-through mode)
- Channel bonding
- Flexible Cyclic Redundancy Check (CRC) generation and checking
- Pins for transmitter and receiver termination voltage
- User reconfiguration using the Dynamic Configuration Bus
- Multiple loopback paths, including PMA RX-TX path

User Guide Organization

This guide is organized as follows:

- Preface, "About This Guide" – Summary of MGT features that allow a multi-gigabit serial transceiver to be integrated easily into any Virtex-4 design.
- Chapter 1, "RocketIO MGT Overview" – MGT basic architecture and capabilities. Includes instantiations, available ports, primitive and modifiable attributes, and byte mapping.

- Chapter 2, "Clocking and Timing Considerations" – Ports and attributes for the provided communications protocol primitives, transceiver instantiation, 8B/10B encoding, 64B/66B encoding, and channel bonding.
- Chapter 3, "PCS Digital Design Considerations" – Clock domain architecture, clock ports, and examples for clocking and reset schemes.
- Chapter 4, "PMA Analog Design Considerations" – Out-of-band signals for power-down applications.
- Chapter 5, "Cyclic Redundancy Check (CRC)"– CRC functionality, latency, and timing.
- Chapter 6, "Analog and Board Design Considerations" – RocketIO serial overview, pre-emphasis, jitter, clock/data recovery, and PCB design requirements.
- Chapter 7, "Simulation and Implementation" – Simulation models and considerations, implementation tools, and debugging and diagnostics.
- Appendix A, "RocketIO MGT Timing Model" – Timing parameters associated with the MGT core.
- Appendix B, "8B/10B Valid Characters" – Valid data and K-character table.
- Appendix C, "Dynamic Configuration Bus" – Parallel programming bus for dynamically configuring the attribute settings. For *Advanced Users Only*.
- Appendix D, "Virtex-II Pro/Virtex-II Pro X to Virtex-4 RocketIO Transceiver Design Migration" – Important differences regarding migration from Virtex-II Pro/ Virtex-II Pro X to the Virtex-4 FPGAs. Highlights relevant PCB, power supply, and reference clock differences.
- Appendix E, "Related Online Documents" – Related documentation will be available on the Xilinx website.
- Appendix F, "Restricted/Reserved Attributes" – Restricted/Reserved attributes.
- "Index" – Standard index of key terms.

Related Information

For a complete menu of online information resources available on the Xilinx website, visit http://www.xilinx.com/virtex4/.
For a comprehensive listing of available tutorials and resources on network technologies and communications protocols, visit http://www.iol.unh.edu/knowledgebase/training/.

Additional Resources

For additional information, go to http://support.xilinx.com. The following table lists some of the resources available on this website. Use the URLs to access these resources directly.

Resource	Description/URL
Tutorials	Tutorials covering Xilinx design flows, from design entry to verification and debugging http://support.xilinx.com/support/techsup/tutorials/index.htm
Answer Browser	Database of Xilinx solution records http://support.xilinx.com/xlnx/xil_ans_browser.jsp

Resource	Description/URL
Application Notes	Descriptions of device-specific design techniques and approaches http://support.xilinx.com/apps/appsweb.htm
Data Sheets	Device-specific information on Xilinx device characteristics, including readback, boundary scan, configuration, length count, and debugging http://support.xilinx.com/xlnx/xweb/xil_publications_index.jsp
Problem Solvers	Interactive tools that allow you to troubleshoot your design issues http://support.xilinx.com/support/troubleshoot/psolvers.htm
Tech Tips	Latest news, design tips, and patch information for the Xilinx design environment http://www.support.xilinx.com/xlnx/xil_tt_home.jsp

User Guide Conventions

This document uses the following conventions.

Port and Attribute Names

All input and output ports of the RocketIO transceiver primitives are denoted in upper-case letters. Attributes of the RocketIO transceiver can be denoted in upper-case letters with underscores or all upper-case letters. When assumed to be the same frequency, RXUSRCLK and TXUSRCLK are referred to as USRCLK and can be used interchangeably. This also holds true for RXUSRCLK2, TXUSRCLK2, and USRCLK2.

Comma Definition

A comma is a "K" character used by the transceiver to align the serial data on a byte/half-word boundary (depending on the protocol used), so that the serial data is correctly decoded into parallel data.

Jitter Definition

Jitter is defined as the short-term variations of significant instants of a signal from their ideal positions in time (ITU). Jitter is typically expressed in a decimal fraction of Unit Interval (UI), for example, 0.3 UI.

Total Jitter (DJ + RJ) Definition

- Deterministic Jitter (DJ) – DJ is data pattern dependant jitter, attributed to a unique source (for example, Inter Symbol Interference (ISI) due to loss effects of the media). DJ is linearly additive.
- Random Jitter (RJ) – RJ is due to stochastic sources, such as thermal and flicker noise, and so on. RJ is additive as the sum of squares and follows a normal distribution.

MGT Definition

The term MGT refers to the Virtex-4 RocketIO Multi-Gigabit Transceiver. Previous generations will be explicitly called out: Virtex-II Pro RocketIO or Virtex-II Pro X RocketIO X.

Typographical

The following typographical conventions are used in this document:

Convention	Meaning or Use	Example
`Courier font`	Messages and prompts 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 syntax statements for which you must supply values	**`ngdbuild`** *`design_name`*
	References to other manuals	See the *Virtex-II Pro 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.
Square brackets [ ]	Optional entry / parameter; required in bus specifications, such as **`bus[7:0]`**	**`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`**`}`

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

Chapter 1

RocketIO MGT Overview

Basic Architecture and Capabilities

The RocketIO™ MGT block diagram is illustrated in Figure 1-1, page 20. Depending on the device, a Virtex-4 FPGA has between 8 and 24 transceiver modules, as shown in Table 1-1.

Table 1-1: **Number of MGT Cores per Device Type**

Device	RocketIO MGT Cores
XC4VFX20	8
XC4VFX40	12
XC4VFX60	12 or 16[1]
XC4VFX100	20
XC4VFX140	24

Notes:

1. Number of MGTs depends on the package.

The transceiver module is designed to operate at any serial bit rate in the range of 622 Mb/s to 10.3125 Gb/s per channel, including the specific bit rates used by the communications standards listed in Table 1-2. Data-rate-specific attribute settings are set appropriately in the Architecture Wizard.

Table 1-2: **Communications Standards Supported by the MGT**

Mode	Channels[1] (Lanes)	I/O Bit Rate (Gb/s)
SONET OC-12	1	0.622
Fibre Channel (1, 2, 4X)	1	1.06/2.12/4.25
Gigabit Ethernet	1	1.25
SONET OC-48	1	2.488
Infiniband	1/4/12	2.5
PCI Express	1/2/4/8/16	2.5
Serial Rapid IO	1/4	1.25/2.5/3.125
Serial ATA	1	1.5/3
XAUI (10 Gigabit Ethernet)	4	3.125
XAUI (10 Gigabit Fibre Channel)	4	3.1875
SONET OC-192[2]	1	9.95328
OIF 10G BP	1	9.96 - 10.3125
Aurora Protocol[3]	1/2/3/4...	0.622 - 10.3125

Notes:

1. One channel is considered to be one transceiver.
2. Payload compatible only.
3. See www.xilinx.com/aurora for details.

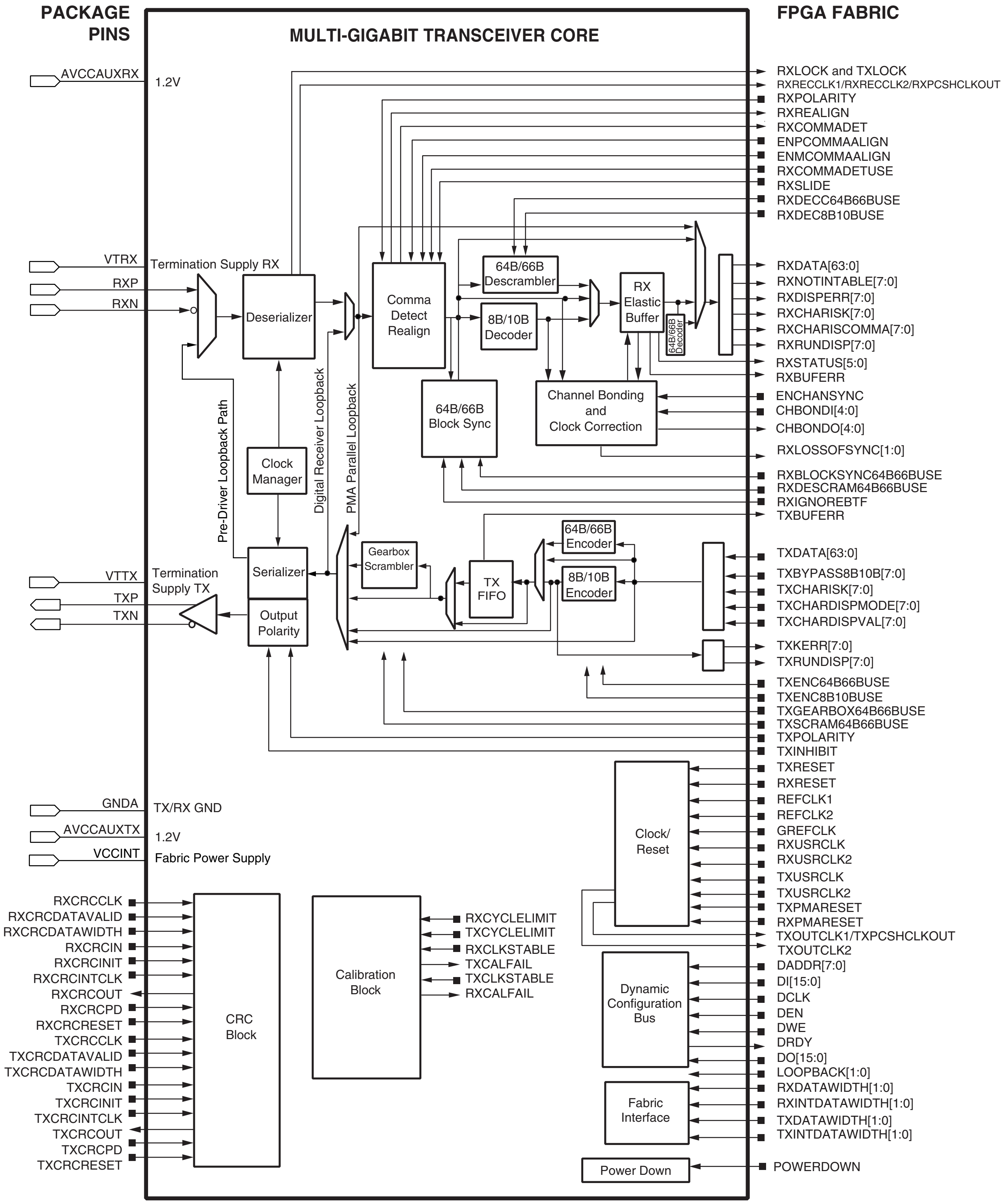


Figure 1-1: **RocketIO MGT Block Diagram**

Table 1-3 lists supported standards and certain values used to support that standard. Data widths of one, two, and four bytes (lower speeds) or four and eight bytes (higher speeds) are selectable for the various protocols. These standards will be supported by the GT11CUSTOM with the use of the Architecture Wizard in ISE 7.1 (or greater).

Table 1-3: **MGT Protocol Settings**[1]

Standard/ Application	Data Rate (Gb/s)	MGT Coding	Reference Clock Frequency (F_{in})	Parallel Data Width (bytes)	Parallel Data Frequency (MHz)
OIF 10G BP[2]	9.96 - 10.3125	64B/66B	622.5 - 644.53125	8	150.9 -156.25
				4	301.818 - 312.50
10G Ethernet	10.3125	64B/66B	644.53125	8	156.25
				4	312.50
OC-192[3]	9.95328	None	622.08	8	155.52
				4	311.04
Custom 6.25 Gb/s	6.25	8B/10B	156.25	8	78.13
				4	156.25
4X Fibre Channel	4.25	8B/10B	106.25	8	53.13
				4	106.25
10G Fibre Channel over 4 links	3.1875	8B/10B	159.375	8	39.84
				4	79.69
XAUI	3.125	8B/10B	156.25	8	39.06
				4	78.13
				2	156.25
Serial Rapid IO Type 3	3.125	8B/10B	156.25	8	39.06
				4	78.13
				2	156.25
Serial Rapid IO Type 2	2.5	8B/10b	125.0	2	125.00
				1	250.00
Serial Rapid IO Type 1	1.25	8B/10b	125.0	2	62.50
				1	125.00
Serial ATA Type 2	3.0	8B/10B	150.0	8	37.50
				4	75.00
				2	150
Serial ATA Type 1	1.5	8B/10B	150.0	2	75.00
				1	150.00
PCI Express	2.5	8B/10B	125.0	2	125.00
				1	250.00

Table 1-3: **MGT Protocol Settings[1]** ***(Continued)***

Standard/ Application	Data Rate (Gb/s)	MGT Coding	Reference Clock Frequency (F_{in})	Parallel Data Width (bytes)	Parallel Data Frequency (MHz)
Infiniband	2.5	8B/10B	125.0	2	125.00
				1	250.00
OIF SxI-5	2.488 - 3.125	None	155.5 - 195.3	8	38.88 - 48.83
				4	77.75 - 97.66
				2	155.5 - 195.32
OIF SFI-4.2	2.56608	64B/66B	160.38	2	155.52
OC-48	2.48832	None	155.52	2	155.52
				1	311.04
2X Fibre Channel	2.125	8B/10B	106.25	2	106.25
				1	212.50
1X Fibre Channel	1.0625	8B/10B	106.25	2	52.13
				1	106.25
1000 BaseX	1.25	8B/10B	125.0	2	62.50
				1	125.0
OC-12	0.622	None	155.52	2	38.88
				1	77.76
Aurora	0.622 - 10.3125	64B/66B, 8B/10B, None	106.25 - 693.75	1, 2, 4, 8	38.88 - 336.36

Notes:

1. Any fabric I/F is possible. However, these are the best settings for the overall system performance and clocking domains, including the 350 MHz maximum parallel speed of the MGT.
2. BP = Back Plane.
3. Payload compatible only.

The three ways to configure the MGT are:

1. Static properties can be set through attributes in the HDL code. Use of attributes are covered in detail in "Attributes," page 35.
2. Dynamic changes can be made to the attributes via the Dynamic Configuration Bus. See Appendix C, "Dynamic Configuration Bus"for details.
3. Dynamic changes can be made through the ports of the primitives.

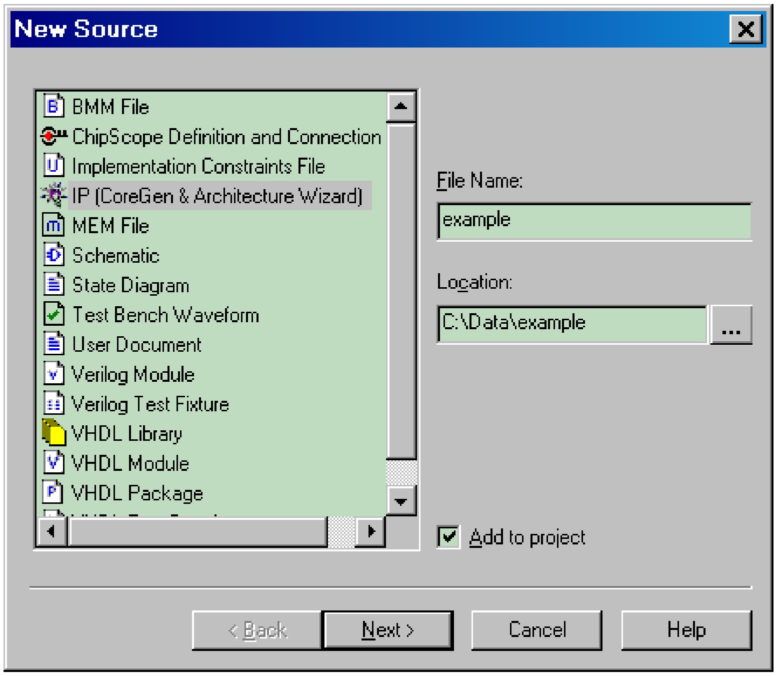
The RocketIO MGT transceiver consists of the Physical Media Attachment (PMA) and Physical Coding Sublayer (PCS). The PMA contains the serializer/deserializer (SERDES), TX and RX input/output buffers, clock generator, and clock recovery circuitry. The PCS contains the 8B/10B encoder/decoder, the 64B/66B encoder/decoder/scrambler/descrambler, and the elastic buffer supporting channel bonding and clock correction. Refer to Figure 1-1, page 20, showing the MGT top-level block diagram and FPGA interface signals.

RocketIO MGT Instantiations

The ISE 7.1 (or greater) Architecture Wizard should be used to create instantiation templates. This ensures the proper analog attribute settings. For those standards not specifically supported in the wizard, use a protocol with the same or similar serial rate and modify the PCS attributes that differ.

If the Architecture Wizard is not used, two GT11 primitives should be instantiated with the COMBUSOUT port connected to the COMBUSIN of the other GT11. **The GT11_CUSTOM and GT11_DUAL should not be used after ISE 7.1 is available.**

To use the wizard from the project navigator, select “add a new source.” The New Source Window (Figure 1-2) appears. Select “IP (CoreGen & Architecture Wizard)” and provide file name and location.



File Name: example

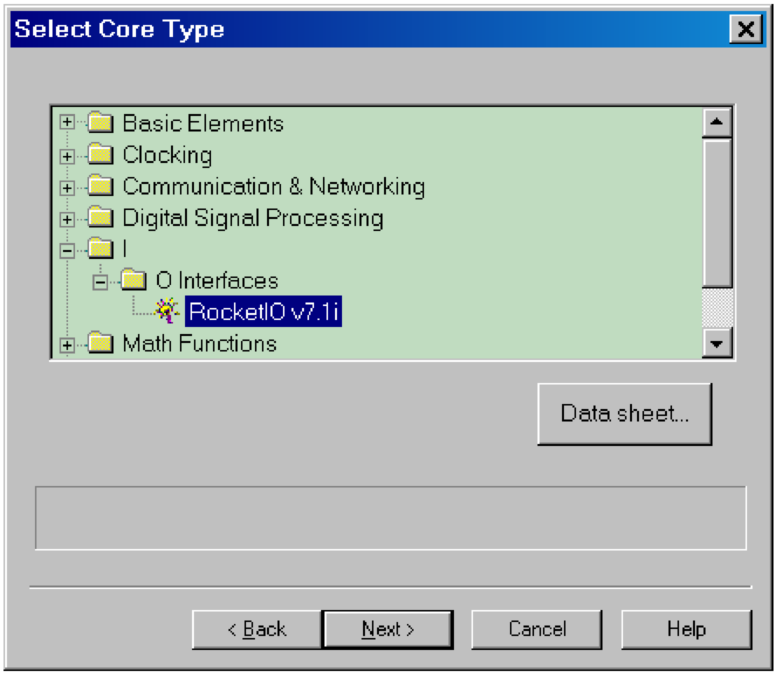
Location: C:\Data\example

☑ Add to project

ug076_ch1_051104

Figure 1-2: **New Source Window**

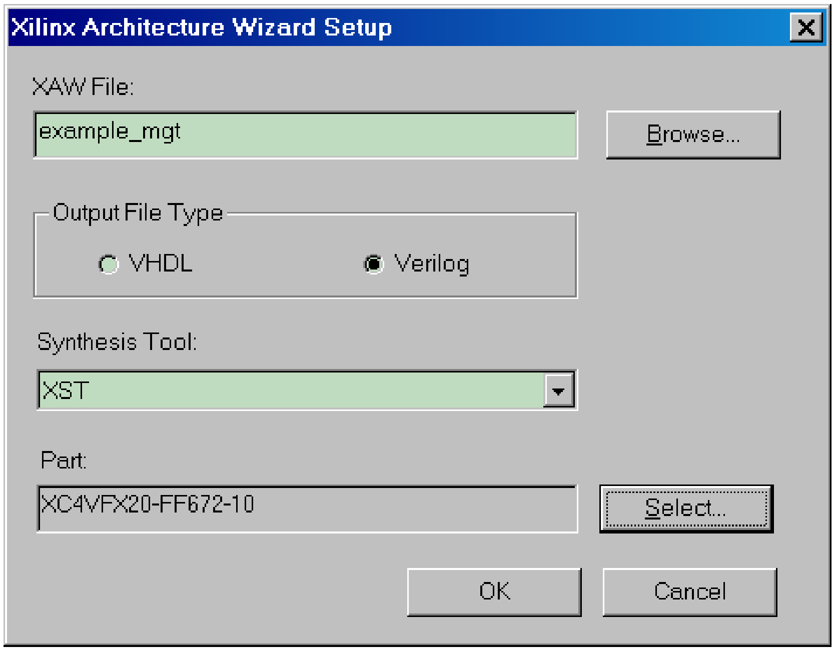
In the next window (Figure 1-3), select the “IO interfaces” and “RocketIO.”



ug076_ch1_02_051104

Figure 1-3: **Select Core Type Window**

Figure 1-4 shows the setup window. Select HDL type, synthesis tool, and targeted device.


Xilinx Architecture Wizard Setup

XAW File: example_mgt

Output File Type: ○ VHDL ● Verilog

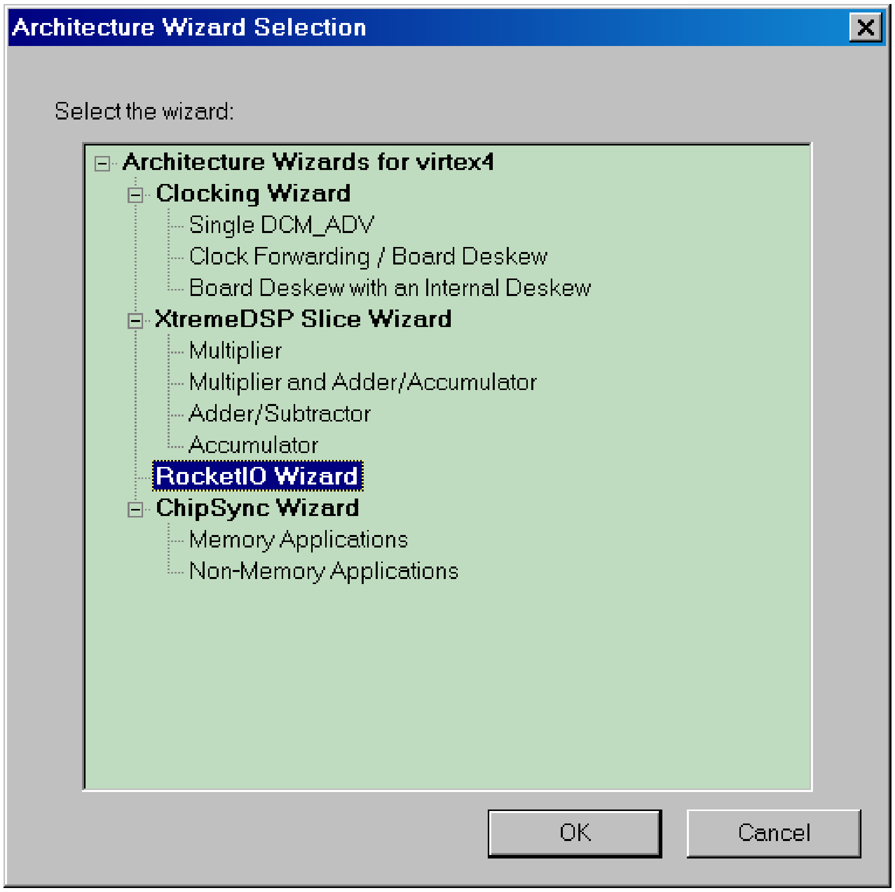
Synthesis Tool: XST

Part: XC4VFX20-FF672-10

ug076_ch1_03_051104

Figure 1-4: **Architecture Wizard Setup Window**

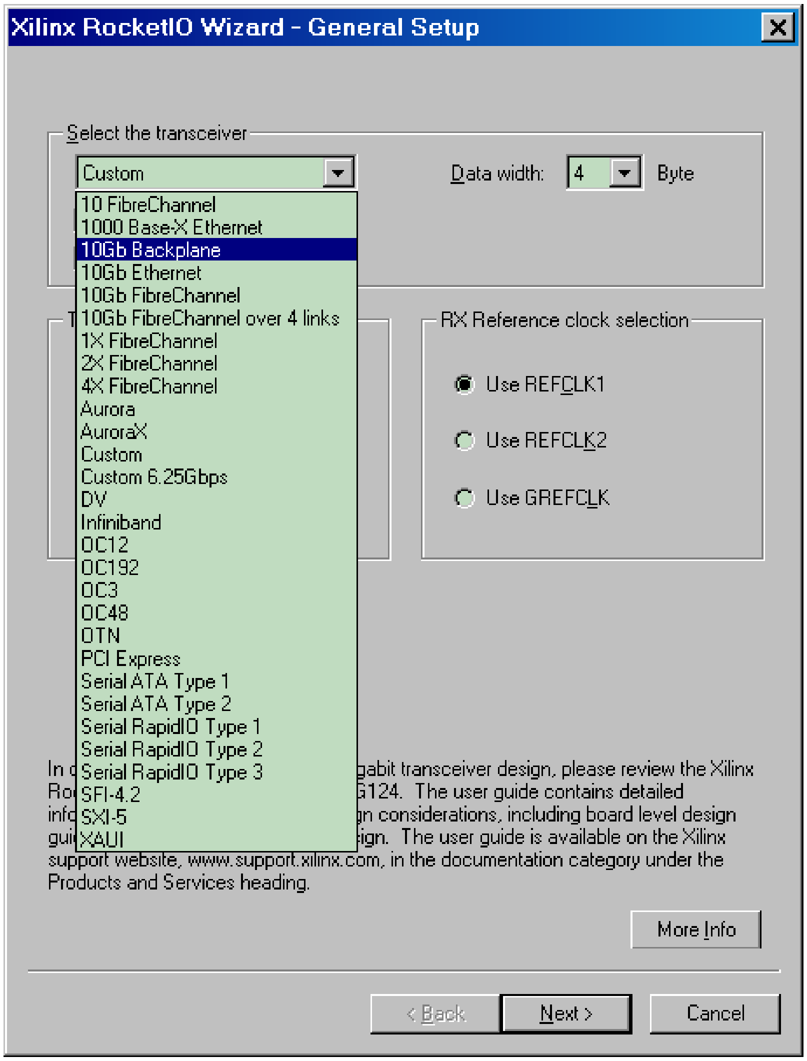
In the next window (Figure 1-5), select the "RocketIO Wizard."



ug076_ch1_04_051104

Figure 1-5: **Architect Wizard Selection Window**

Figure 1-6 shows the list of supported standards, which are preset with the appropriate attributes for the particular standard chosen. For more details, see the Architecture Wizard documentation.



Xilinx RocketIO Wizard - General Setup

Select the transceiver

Custom

- 10 FibreChannel
- 1000 Base-X Ethernet
- 10Gb Backplane
- 10Gb Ethernet
- 10Gb FibreChannel
- 10Gb FibreChannel over 4 links
- 1X FibreChannel
- 2X FibreChannel
- 4X FibreChannel
- Aurora
- AuroraX
- Custom
- Custom 6.25Gbps
- DV
- Infiniband
- OC12
- OC192
- OC3
- OC48
- OTN
- PCI Express
- Serial ATA Type 1
- Serial ATA Type 2
- Serial RapidIO Type 1
- Serial RapidIO Type 2
- Serial RapidIO Type 3
- SFI-4.2
- SXI-5
- XAUI

Data width: 4 Byte

RX Reference clock selection

● Use REFCLK1

○ Use REFCLK2

○ Use GREFCLK

In d gabit transceiver design, please review the Xilinx
Ro G124. The user guide contains detailed
info n considerations, including board level design
gui ign. The user guide is available on the Xilinx
support website, www.support.xilinx.com, in the documentation category under the
Products and Services heading.

ug076_ch1_05_051104

Figure 1-6: **RocketIO Wizard - General Setup Window**

Available Ports

Table 1-4 (CRC), Table 1-5 (PMA), Table 1-6 (PCS), Table 1-7 (Global Signals), and Table 1-8 (Configuration) contain all the primitive port descriptions. The RocketIO MGT primitives contain 99 ports. The differential serial data ports (RXN, RXP, TXN, and TXP) are connected directly to external pads; the remaining 95 ports are all accessible from the FPGA logic.

Table 1-4: **CRC Primitive Ports**

Port	I/O	Port Size	Definition
RXCRCCLK	I	1	Receiver CRC logic clock.
RXCRCDATAVALID	I	1	Signals that the RXCRCIN data is valid.
RXCRCDATAWIDTH	I	3	Determines the data width of the RXCRCIN.
RXCRCIN	I	64	Receiver CRC logic input data.
RXCRCINIT	I	1	When set to logic 1, CRC logic initializes to the RXCRCINITVAL.
RXCRCINTCLK	I	1	Receiver CRC/FPGA fabric interface clock.
RXCRCOUT	O	32	Receiver CRC output data.
RXCRCPD	I	1	Indicates the RX CRC logic to power down when set to logic 1.
RXCRCRESET	I	1	Resets the RX CRC logic when set to logic 1.
TXCRCCLK	I	1	Transmitter CRC logic clock.
TXCRCDATAVALID	I	1	Signals that the TXCRCIN data is valid when set to a logic 1.
TXCRCDATAWIDTH	I	3	Determines the data width of the TXCRCIN.
TXCRCIN	I	64	Transmitter CRC logic input data.
TXCRCINIT	I	1	When set to logic 1, CRC logic initializes to the TXCRCINITVAL.
TXCRCINTCLK	I	1	Transmitter CRC/FPGA fabric interface clock.
TXCRCOUT		32	Transmitter CRC output data.
TXCRCPD	I	1	Indicates the TX CRC logic to power down when set to a logic 1.
TXCRCRESET	I	1	Resets the TX CRC logic when set to a logic 1.

Table 1-5: **PMA Primitive Ports**

Port	I/O	Port Size	Definition
Calibration			
RXCALFAIL	O	1	Status output that when set to a logic 1 indicates receiver cannot lock to the reference clock. Possible reasons include no reference clock, incorrect reference clock frequency, incorrect attribute configuration, or PMA power not applied.
RXCLKSTABLE	I	1	An input that when set to a logic 1 indicates the clocks are stable and MGT RX calibration can start. See Chapter 2, "Clocking and Timing Considerations," for details.
RXCYCLELIMIT	O	1	A status output that when set to a logic 1 indicates that the receiver cannot lock to the reference clock. Possible reasons include an unstable or noisy reference clock, or incorrect attribute configuration.
TXCALFAIL	O	1	Status output that when set to a logic 1 indicates that the transmitter cannot lock to the reference clock. Possible reasons include no reference clock, incorrect reference clock frequency, incorrect attribute configuration or PMA power not applied.
TXCLKSTABLE	I	1	An input that when set to a logic 1 indicates that the clocks are stable and MGT TX calibration can start. See Chapter 2, "Clocking and Timing Considerations," for details.
TXCYCLELIMIT	O	1	A status output that when set to a logic 1 indicates that the transmitter cannot lock to the reference clock. Possible reasons include an unstable or noisy reference clock, or incorrect attribute configuration.
Driver/Buffers			
TXINHIBIT	I	1	If set to a logic 1, the TX differential pairs are forced to be a constant 1/0. TXN = 1, TXP = 0
RXPOLARITY	I	1	Sets the input polarity for the receiver inputs RXP and RXN. Normal polarity is used when set to logic 0. The receiver input is inverted when set to logic 1.
TXPOLARITY	I	1	Sets the output polarity for the transmitter outputs TXP and TXN. Normal polarity is used when set to logic 0. The transmitter output is inverted when set to logic 1.
RXN	I	1	Differential serial input (external package pin). See Chapter 7, "Simulation and Implementation," for package pin correlation to MGT location constraint.
RXP	I	1	Differential serial input (external package pin). See Chapter 7, "Simulation and Implementation," for package pin correlation to MGT location constraint.
TXN	O	1	Differential serial output (external package pin). See Chapter 7, "Simulation and Implementation," for package pin correlation to MGT location constraint.

Table 1-5: **PMA Primitive Ports *(Continued)***

Port	I/O	Port Size	Definition
TXP	O	1	Differential serial output (external package pin). See Chapter 7, "Simulation and Implementation," for package pin correlation to MGT location constraint.
Clocks/Clock Status			
RXLOCK	O	1	If set to logic 1, the receiver is locked to reference clock or locked to the input data. Logic 0 indicates the receiver is not locked. Possible reasons include no reference clock, incorrect reference clock frequency, incorrect attribute configuration, PMA power was not applied, or receiver was not able to lock to data and is in the process of locking to the local reference clock again.
RXPMARESET	I	1	Resets the receiver PMA when set to logic 1.
RXRECCLK1	O	1	Recovered clock from incoming data. See "PMA Receive Clocks" in Chapter 2.
RXRECCLK2	O	1	Recovered clock from incoming data. RXRECCLK1 or RXPCSHCLKOUT should always be used as an alternative for this port. See "PMA Receive Clocks" in Chapter 2.
RXMCLK	O	1	Recovered clock from the receiver PLL. This is an output with dedicated routing to a GT11CLK instance. It can only be connected to the RXBCLK input port on GT11CLK from an MGT B. This recovered clock can then be selected as the output of the GT11CLK instance.
TXLOCK	O	1	Status output that when set to logic 1 indicates that the transmitter PLL is locked to the reference clock. This output cycles from between logic 0 and logic 1 during lock acquisition. When the output maintains a logic 1 state, the transmitter PLL is locked. Failure to acquire or maintain lock can be due to no reference clock, incorrect reference clock frequency, incorrect attribute configuration, or PMA power was not applied.
TXOUTCLK1	O	1	Transmitter output clock derived from PLL based on transmitter reference clock. See "PMA Transmit Clocks" in Chapter 2.
TXOUTCLK2	O	1	Transmitter output clock derived from PLL based on transmitter reference clock. TXCLKOUT1 OR TXPCSHCLKOUT should always be used as an alternative for this port. See "PMA Transmit Clocks" in Chapter 2.
TXPMARESET	I	1	Resets the transmitter PMA when set to a logic 1.
RXPCSHCLKOUT	O	1	Recovered clock from receiver PLL. This output has dedicated routing to the FPGA global clock resources. It should be used when connecting a receiver recovered clock to a global clock.
TXPCSHCLKOUT	O	1	Clock from transmitter PLL. This output has dedicated routing to the FPGA global clock resources. It should be used when connecting a transmitter PLL clock to a global clock.

Table 1-5: **PMA Primitive Ports *(Continued)***

Port	I/O	Port Size	Definition
Special Signals			
RXSIGDET	O	1	Indicates that a signal at the receiver input has been detected. The signal detect function can also be used as a beacon or Out-of-Band (OOB) signal receiver. See Chapter 4, "PMA Analog Design Considerations," for details.
RXSYNC	I	1	When transitioned from low to high, synchronous clocks are aligned. See the *Reduced Latency* section.
TXENOOB	I	1	Enables the transmitter to send electric idle signals on the TXN/TXP pins when set to a logic 1. See Chapter 4, "PMA Analog Design Considerations," for details.
TXSYNC	I	1	Enables internal parallel clocks to sync to GREFCLK for channel bonding applications. See Chapter 2, "Clocking and Timing Considerations" for details.

Table 1-6: **PCS Primitive Ports**

Port	I/O	Port Size	Definition
Channel Bonding			
CHBONDI	I	5	The channel bonding control that is used only by "slaves" which are driven by a transceiver's CHBONDO port.
CHBONDO	O	5	Channel bonding control that passes channel bonding and clock correction control to other transceivers.
ENCHANSYNC	I	1	Control from the fabric to the transceiver which enables the transceiver to perform channel bonding.
64B/66B			
RXBLOCKSYNC64B66BUSE	I	1	If set to logic 1, the block sync is used. If set to a logic 0, the block sync logic is bypassed.
RXDEC64B66BUSE	I	1	If set to a logic 1, the 64/B66B decoder is used. If set to a logic 0, the 64B/66B decoder is bypassed.
RXDESCRAM64B66BUSE	I	1	If set to a logic 1, the descrambler is used. If set to a logic 0, the descrambler is bypassed.
RXIGNOREBTF	I	1	If set to a logic 1, the block type field (BTF) is ignored in the 64B/66B decoder. Instead of reporting an error, the block is passed on as is. If set to a logic 0, unrecognized BTFs are marked as error blocks.
TXENC64B66BUSE	I	1	If set to a logic 1, the 64B/66B encoder is used. If set to a logic 0, the 64B/66B encoder is bypassed.
TXGEARBOX64B66BUSE	I	1	If set to a logic 1, the 64B/66B gearbox is used. If set to a logic 0, the 64B/66B gearbox is bypassed. TXSCRAM64B66USE = TXGEARBOX64B66BUSE
TXSCRAM64B66BUSE	I	1	If set to a logic 1, the 64B/66B scrambler is used. If set to a logic 0, the 64B/66B scrambler is bypassed. TXSCRAM64B66USE = TXGEARBOX64B66BUSE

Table 1-6: **PCS Primitive Ports *(Continued)***

Port	I/O	Port Size	Definition
8B/10B			
RXCHARISK	O	8	If 8B/10B decoding is enabled, it indicates that the received data is a "K" character when asserted. Included in byte-mapping. If 8B/10B decoding is bypassed, it remains as the first bit received (Bit "a") of the 10-bit encoded data.
RXCHARISCOMMA	O	8	Indicates the reception of K28.0, K28.5, K28.7, and some out of band commas (depending on the setting of DEC_VALID_COMMA_ONLY by the 8B/10B decoder.
RXDEC8B10BUSE	I	1	If set to a logic 1, the 8B/10B decoder is used. If set to a logic 0, the 8B/10B decoder is bypassed.
RXDISPERR	O	8	If 8B/10B encoding is enabled, it indicates whether a disparity error has occurred on the serial line. Included in byte-mapping scheme.
RXNOTINTABLE	O	8	Status of encoded data when the data is not a valid character when set to a logic 1. Applies to the byte-mapping scheme.
RXRUNDISP	O	8	Signals the running disparity (0 = negative, 1 = positive) in the received serial data. If 8B/10B encoding is bypassed, it remains as the second bit received (Bit "b") of the 10-bit encoded data.
TXBYPASS8B10B	I	8	Determines which encoding is enabled or bypassed. See Chapter 3, "PCS Digital Design Considerations," for details.
TXCHARDISPMODE	I	8	If 8B/10B encoding is enabled, this bus determines what mode of disparity is to be sent. When 8B/10B is bypassed, this becomes the first bit transmitted (Bit "a") of the 10-bit encoded TXDATA bus section (see Figure 3-5, page 72) for each byte specified by the byte-mapping. The bits have no meaning if TXENC8B10BUSE is set to a logic 0.
TXCHARDISPVAL	I	8	If 8B/10B encoding is enabled, this bus determines what type of disparity is to be sent. When 8B/10B is bypassed, this becomes the second bit transmitted (Bit "b") of the 10-bit encoded TXDATA bus section (see Figure 3-5, page 72) for each byte specified by the byte-mapping section. The bits have no meaning if TXENC8B10BUSE is set to a logic 0.
TXCHARISK	I	8	If TXENC8B10BUSE = 1 (8B/10B encoder enable), then TXCHARISK[7:0] signals the K-definition of the TXDATA byte in the corresponding byte lane. (1 indicates that the byte is a K-character; 0 indicates that the byte is a data character) If TXENC64B66BUSE = 1 (64B/66B encoder enable), then TXCHARISK[3:0] signals the block-formatting definitions of TXDATA (1 indicates that the byte is a control character; 0 indicates that the byte is a data character). TXCHARISK[7:4] bits are not relevant in this particular configuration.

Table 1-6: **PCS Primitive Ports *(Continued)***

Port	I/O	Port Size	Definition
TXENC8B10BUSE	I	1	If set to a logic 1, the 8B/10B encoder is used. If set to a logic 0, the 8B/10B encoder is bypassed.
TXKERR	O	8	Indicates even boundary for bypassing in 64B/66B mode. In 8B/10B mode, indicates that an invalid K-character was transmitted.
TXRUNDISP	O	8	Signals the running disparity for its corresponding byte, after that byte is encoded. Zero equals negative disparity and positive disparity for a one.
Alignment			
ENMCOMMAALIGN	I	1	Selects realignment of incoming serial bitstream on minus-comma. When set to logic 1, realigns serial bitstream byte boundary to where minus-comma is detected.
ENPCOMMAALIGN	I	1	Selects realignment of incoming serial bitstream on plus-comma. When reasserted, realigns serial bitstream byte boundary to where plus-comma is detected.
RXCOMMADET	O	1	Indicates that the symbol defined by PCOMMA_10B_VALUE (IF PCOMMA_DETECT is asserted) and/or MCOMMA_10B_VALUE (if MCOMMA_DETECT is asserted) has been received.
RXCOMMADETUSE	I	1	If set to a logic 1, the comma detect is used. If set to a logic 0, the comma detect is bypassed.
RXLOSSOFSYNC	O	2	Bit 0 is always zero. Bit 1 indicates 64B/66B block lock when set to logic 0.
RXREALIGN	O	1	Signal from the PMA denoting that the byte alignment with the serial data stream changed due to a comma detection. Set to a logic 1 when alignment occurs.
RXSLIDE	I	1	Enables the "slip" of the detection block by 1 bit. To enable a slide of 1 bit, it increments from a lower bit to a higher bit. This signal must be set to logic 1 and then set to a logic 0 synchronous to RXUSRCKLK2. RXSLIDE must be held Low for at least two clock cycles before being set to a logic 1 again.
Data Path			
RXDATA	O	64	Up to eight bytes of decoded (8B/10B encoding) or encoded (8B/10B bypassed) received data at the user fabric.
RXDATAWIDTH	I	2	Indicates width of FPGA parallel bus.
RXINTDATAWIDTH	I	2	Sets the internal mode of the receive PCS, either 32 or 40 bit.
TXDATA	I	64	Transmit data from the FPGA user fabric that is 8 bytes wide. TXDATA[7:0] is always the first byte transmitted.
TXDATAWIDTH	I	2	Indicates width of FPGA parallel bus.
TXINTDATAWIDTH	I	2	Indicates internal data width.
Status/Clocks			

Table 1-6: **PCS Primitive Ports *(Continued)***

Port	I/O	Port Size	Definition
RXBUFERR	O	1	Provides status of the receiver FIFO. If set to a logic 1, an overflow/underflow has occurred. When this bit becomes set, it can only be reset by asserting RXRESET
RXRESET	I	1	Synchronous RX PCS reset that "recenters" the receive elastic buffer. It also resets 8B/10B decoder, comma detect, channel bonding, clock correction logic, and other internal receive registers. It does not reset the receiver PLL or transmit PCS.
RXSTATUS	O	6	RXSTATUS[5] indicates a receiver has successfully completed channel bonding when set to logic 1. RXSTATUS[4:0] receiver elastic buffer status. Indicates the status of the receive FIFO pointers, channel bonding skew, and clock correction events.
RXUSRCLK	I	1	Clock that is used for reading the RX elastic buffer. It also clocks CHBONDI and CHBONDO in and out of the transceiver. Typically, the same as TXUSRCLK.
RXUSRCLK2	I	1	Clock output that clocks the receive data and status between the transceiver and the FPGA core. Typically, the same as TXUSRCLK2.
TXBUFERR	O	1	Provides status of the transmission FIFO. If set to a logic 1, an overflow/underflow has occurred. When this bit becomes set, it can only be reset by setting TXRESET to logic 1.
TXRESET	I	1	Synchronous TX PCS reset that "recenters" the transmit buffer. It also resets 8B/10B encoder and other internal transmission registers. It does not reset the PMA including the PLL or receive PCS.
TXUSRCLK	I	1	Clock output that is clocked with the REFCLK (or other reference clock). This clock is used for writing the TX buffer and is frequency-locked to the REFCLK.
TXUSRCLK2	I	1	Clock input that clocks that clocks the receive data and status between the FPGA core and the transceiver. Typically, the same as RXUSRCLK2.

Table 1-7: **Global Signal Primitive Ports**

Port	I/O	Port Size	Definition
LOOPBACK	I	2	Selects the four loopback test modes. These modes are PCS parallel, serial, and PMA parallel (RX –> TX) loopback. See Chapter 3, "PCS Digital Design Considerations," for details.
POWERDOWN	I	1	Shuts down entire PCS transceiver when set to logic 1. This input is asynchronous. PMA powerdown via attributes.
GREFCLK	I	1	Reference clock (alternate clock not recommended for over 1G operation).
REFCLK1	I	1	Reference clock (low jitter input clock).
REFCLK2	I	1	Reference clock (low jitter input clock).

Table 1-8: **Configuration Primitive Ports**

Port	I/O	Port Size	Definition
DADDR	I	8	Dynamic configuration address bus. See Appendix C, "Dynamic Configuration Bus."
DCLK	I	1	Dynamic configuration bus clock.
DEN	I	1	Dynamic configuration bus enable when set to a logic 1.
DI	I	16	Dynamic configuration input data bus.
DO	O	16	Dynamic configuration output data bus.
DRDY	O	1	Indicates that the dynamic configuration output data is valid when set to a logic 1.
DWE	I	1	Dynamic configuration write enable when set to a logic 1.

Table 1-9: **MGT Tile Communication Ports**

Port	I/O	Port Size	Definition
COMBUSOUT	O	16	Connects to the COMBUSIN of the other instance to allow proper simulation of shared clocks and PLLs.
COMBUSIN	I	16	Connects to the COMBUSOUT of the other instance to allow proper simulation of shared clocks and PLLs.

Attributes

An attribute is a control parameter to configure the MGT. There are both primitive ports (traditional I/O ports for control and status) and attributes. Transceiver attributes are also controls to the transceiver that regulate data widths and encoding rules, but they are controls that are configured as a group in "soft" form through the invocation of a primitive.

The MGT also contains attributes set by default to specific values. Included are channel bonding settings and clock correction sequences. Table 1-10 through Table 1-14 show a brief description of each attribute. See the memory maps in Appendix C, "Dynamic Configuration Bus"for the default values.

Table 1-10 through Table 1-14 show all the modifiable attributes. Some reserved attributes (found in Appendix C, "Dynamic Configuration Bus") should not be changed from the default settings. These are shown in Appendix F, "Restricted/Reserved Attributes" and should never be changed from the defaults.

Note: **Xilinx recommends that all attributes be set using the Architecture Wizard. These descriptions are provided to allow modification of individual attributes from the basic settings provided by the Wizard.**

Table 1-10: **RocketIO MGT CRC Attributes**

Attribute	Type	Description
RXCRCCLOCKDOUBLE	Boolean	FALSE/TRUE. Select clock frequency ratio between RXCRCCLK and RXCRCINTCLK. TRUE indicates the fabric data interface clock RXCRCINTCLK is operating at half the frequency of the internal clock RXCRCCLK. FALSE indicates both clocks are at the same frequency. See Chapter 5, "Cyclic Redundancy Check (CRC)," for more details.
RXCRCINITVAL	32-bit Hex	Initial value for the CRC generation logic and allows custom CRC generation.
RXCRCSAMECLOCK	Boolean	TRUE/FALSE. Select single clock mode for RX CRC. TRUE indicates that the fabric interface clock rate is the same as the internal clock rate (RXCRCCLOCKDOUBLE is FALSE), the CRC fabric interface and internal logic are being clocked using the RXCRCINTCLK. FALSE indicates that the clocks are being supplied to both the RXCRCCLK and RXCRCINTCLK ports. This attribute is typically set to TRUE when RXCRCCLOCKDOUBLE is set to FALSE. See Chapter 5, "Cyclic Redundancy Check (CRC)," for more details.
TXCRCCLOCKDOUBLE	Boolean	FALSE/TRUE. Select clock frequency ratio between TXCRCCLK and RXCRCINTCLK. TRUE indicates the fabric data interface clock TXCRCINTCLK is operating at half the frequency the internal clock TXCRCCLK. FALSE indicates both clocks are a the same frequency. See Chapter 5, "Cyclic Redundancy Check (CRC)," for more details.
TXCRCINITVAL	32-bit Hex	Initial value for the CRC generation logic to create custom CRC seed.

Table 1-10: **RocketIO MGT CRC Attributes *(Continued)***

Attribute	Type	Description
TXCRCSAMECLOCK	Boolean	TRUE/FALSE. Select single clock mode for TX CRC. TRUE indicates that the fabric interface clock rate is the same as the internal clock rate (TXCRCCLOCKDOUBLE is FALSE), and the CRC fabric interface and internal logic are being clocked using the TXCRCINTCLK. FALSE indicates that the clocks are being supplied to both the TXCRCCLK and TXCRCINTCLK ports. This attribute is typically set to TRUE when TXCRCCLOCKDOUBLE is set to FALSE. See the Chapter 5, "Cyclic Redundancy Check (CRC)."

Table 1-11: **RocketIO MGT PMA Attributes**

Attribute	Type	Description
Calibration		
FDET_HYS_CAL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
FDET_HYS_SEL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
FDET_LCK_CAL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
FDET_LCK_SEL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
LOOPCAL_WAIT	2-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
RXFDET_LCK_CAL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
RXFDET_HYS_CAL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
RXFDET_HYS_SEL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4for more details.
RXFDET_LCK_CAL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4for more details.
RXFDET_LCK_SEL	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
RXLOOPCAL_WAIT	2-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
RXFDET_CLOCK_DIVIDE	3-bit Binary	Sets up the calibration circuitry. See "Calibration for the PLLs" in Chapter 4 for more details.
RXVCODAC_INIT	10-bit Binary	Affects the characteristics of the calibration logic. See "Calibration for the PLLs" in Chapter 4.
TXFDCAL_CLOCK_DIVIDE	String	None, two, four.
VCODAC_INIT	10-bit Binary	Affects the characteristics of the calibration logic. See "Calibration for the PLLs" in Chapter 4.

Table 1-11: **RocketIO MGT PMA Attributes *(Continued)***

Attribute	Type	Description
Drivers/Buffers		
RXAFEEQ	8-bit Binary	Receiver linear equalization control attributes. See "Receive Equalization" in Chapter 4.
RXDCCOUPLE	Boolean	FALSE - Internal RX AC coupling capacitors enabled. TRUE - Internal RX AC coupling capacitors bypassed.
RXEQ	64-bit Hex	Receiver DFE equalization control attributes. See "Receive Equalization" in Chapter 4.
TXDAT_TAP_DAC	5-bit Binary	Transmitter data amplitude control. See "Output Swing and Emphasis" in Chapter 4.
TXPOST_TAP_DAC	5-bit Binary	Transmitter post-cursor amplitude control. See "Output Swing and Emphasis" in Chapter 4.
TXHIGHSIGNALEN	Boolean	This attribute controls the line driver strength. This should always be set to TRUE.
TXPOST_TAP_PD	Boolean	Transmitter post-cursor amplitude control. See Chapter 4, "PMA Analog Design Considerations."
TXPRE_TAP_DAC	5-bit Binary	Transmitter pre-cursor amplitude control. See "Output Swing and Emphasis" in Chapter 4.
TXPRE_TAP_PD	Boolean	Transmitter pre-cursor pre-emphasis disable control: TRUE - Disable post-cursor pre-emphasis. FALSE - Enable post-cursor pre-emphasis. See "Emphasis" in Chapter 4.
TXSLEWRATE	Boolean	Select reduced slew rate on transmitter output: TRUE - Select reduced output slew rate. FALSE - Select standard output slew rate.
TXTERMTRIM	4-bit Binary	Resistive line driver termination trim.
Clocks		
RXCLKMODE	6-bit Binary	Sets the internal clocking modes. See Chapter 2, "Clocking and Timing Considerations."
RXOUTDIV2SEL	Integer	Frequency acquisition loop output divide. See Chapter 2, "Clocking and Timing Considerations" for setting the correct value.
RXPLLNDIVSEL	Integer	Frequency acquisition loop feedback divide for the RX PLL. See Chapter 2, "Clocking and Timing Considerations" for setting the correct value.

RXPMACLKSEL	String	REFCLK1, REFCLK2, GREFCLK. Selects reference clock input for receive PLL:
		REFCLK1 = Select REFCLK1 input (DRP value 00).
		REFCLK2 = Select REFCLK2 input (DRP value 01).
		GREFCLK = Select GREFCLK input (DRP value 10).
		See Chapter 2, "Clocking and Timing Considerations" for more details.

RXRECCLK1_USE_SYNC	Boolean	Selects the RXRECCLK1 frequency. See Chapter 2, "Clocking and Timing Considerations."

Table 1-11: **RocketIO MGT PMA Attributes *(Continued)***

Attribute	Type	Description

TXABPMACLKSEL	String	REFCLK1, REFCLK2, GREFCLK. Selects reference clock input for shared Tile transmit PLL:
		REFCLK1 = Select REFCLK1 input (DRP value 00).
		REFCLK2 = Select REFCLK2 input (DRP value 01).
		GREFCLK = Select GREFCLK input (DRP value 10).
		See Chapter 2, "Clocking and Timing Considerations" for more details.

TXCLKMODE	4-bit Binary	Sets the internal clocking mode. See Chapter 2, "Clocking and Timing Considerations."
TXOUTCLK1_USE_SYNC	Boolean	See Chapter 2, "Clocking and Timing Considerations" for more details.
TXOUTDIV2SEL	Integer	Frequency acquisition loop output divide. See Chapter 2, "Clocking and Timing Considerations" for setting the correct value.
TXPHASESEL	Boolean	TRUE - Selects PCS clock for synchronization clock. FALSE - Selects GREFCLK for synchronization clock.
TXPLLNDIVSEL	Integer	Frequency acquisition loop feedback divide for TX PLL. See Chapter 2, "Clocking and Timing Considerations" for setting the correct value.
Miscellaneous		
RXCDRLOS	6-bit Binary	These bits set the threshold value for the signal detector (OOB signal detect). See "Out-of-Band (OOB) Signals" in Chapter 4.
RXLKADJ	5-bit Binary	Adjusts the sample point.
RXPD	Boolean	Power down selector for the receiver. TRUE - Power down receiver. FALSE - Power up receiver
RXRSDPD	Boolean	FALSE -Power-up receiver signal detect logic. TRUE - Power-down receiver signal detect logic.
TXPD	Boolean	TRUE - Power-down transmitter. FALSE - Power-up transmitter.
PMACORENABLE	Boolean	TRUE: Powers up RXA, RXB, and TXAB. FALSE: Powers down RXA, RXB, and TXAB.
PMA_BIT_SLIP	Boolean	FALSE/TRUE TRUE: Performs 2 UI bit slip in the PMA on the rising edge of RXSYNC. FALSE: Performs PCS clock phase alignment when RXSYNC is set to a logic 1.

Table 1-12: **RocketIO MGT PCS Attributes**

Attribute	Type	Description
Channel Bonding		
CCCB_ARBITRATOR_DISABLE	Boolean	TRUE/FALSE determines if the clock correction/channel bonding arbitrator is disabled or not. When set to FALSE, the arbitrator is enabled, allowing clock correction and channel bonding sequences to occur adjacent without padding bytes. The clock correction always takes priority over the channel bonding.
CHAN_BOND_LIMIT	Integer	Integer 1-31 defines maximum number of bytes a slave receiver can read following a channel bonding sequence and still successfully align to that sequence.
CHAN_BOND_MODE	String	STRING: NONE, MASTER, SLAVE_1_HOP, SLAVE_2_HOPS NONE: No channel bonding involving this transceiver. MASTER: This transceiver is master for channel bonding. Its CHBONDO port directly drives CHBONDI ports on one or more SLAVE_1_HOP transceivers. SLAVE_1_HOP: This transceiver is a slave for channel bonding. SLAVE_1_HOP's CHBONDI is directly driven by a MASTER transceiver CHBONDO port. SLAVE_1_HOP's CHBONDO port can directly drive CHBONDI ports on one or more SLAVE_2_HOPS transceivers. SLAVE_2_HOPS: This transceiver is a slave for channel bonding. SLAVE_2_HOPS CHBONDI is directly driven by a SLAVE_1_HOP CHBONDO port.
CHAN_BOND_ONE_SHOT	Boolean	FALSE/TRUE that controls repeated execution of channel bonding. FALSE: Master transceiver initiates channel bonding whenever possible (whenever channel-bonding sequence is detected in the input) as long as input ENCHANSYNC is High and RXRESET is Low. TRUE: Slave transceiver initiates channel bonding only the first time it is possible (channel bonding sequence is detected in input) following negated RXRESET and asserted ENCHANSYNC. After channel-bonding alignment is done, it does not occur again until RXRESET is asserted and negated, or until ENCHANSYNC is negated and reasserted.
CHAN_BOND_SEQ_1_1, 2, 3, 4	11-bit Binary	These define the channel bonding sequence. The usage of these vectors also depends on CHAN_BOND_SEQ_LEN and CHAN_BOND_SEQ_2_USE. For details, see Table 3-15, "Definition of CCS or CBS bits 9-0."
CHAN_BOND_SEQ_1_MASK	4-bit Binary	Each bit of the mask determines if that particular sequence is detected regardless of its value. For example, if bit 0 is High, then CHAN_BOND_SEQ_1_1 is matched regardless of its value.
CHAN_BOND_SEQ_2_1, 2, 3, 4	11-bit Binary	These define the channel bonding sequence.The usage of these vectors also depends on CHAN_BOND_SEQ_LEN and CHAN_BOND_SEQ_2_USE. For details, see Table 3-15, "Definition of CCS or CBS bits 9-0."

Table 1-12: **RocketIO MGT PCS Attributes *(Continued)***

Attribute	Type	Description
CHAN_BOND_SEQ_2_MASK	4-bit Binary	Each bit of the mask determines if that particular sequence is detected regardless of its value. For example, if bit 0 is High, then CHAN_BOND_SEQ_2_1 is matched regardless of its value.
CHAN_BOND_SEQ_2_USE	Boolean	FALSE/TRUE that controls use of second channel bonding sequence. FALSE: Channel bonding uses only one channel bonding sequence defined by CHAN_BOND_SEQ_1_1... 4, **or** one 8-byte sequence defined by CHAN_BOND_SEQ_1...4 and CHAN_BOND_SEQ_2_1...4 in combination. TRUE: Channel bonding uses two channel bonding sequences defined by CHAN_BOND_SEQ_1_1... 4 and CHAN_BOND_SEQ_2_1... 4, as further constrained by CHAN_BOND_SEQ_LEN.
CHAN_BOND_SEQ_LEN	Integer	Integer (1, 2, 3, 4, 8) defines length in bytes of channel bonding sequence. This defines the length of the sequence the transceiver matches to detect opportunities for channel bonding.
Clock Correction		
CLK_CORRECT_USE	Boolean	TRUE/FALSE controls the use of clock correction logic. FALSE: Permanently disable execution of clock correction (rate matching). Clock RXUSRCLK must be frequency-locked with RXRECCLK in this case. TRUE: Enable clock correction (normal mode).
CLK_COR_8B10B_DE	Boolean	This signal selects if clock correction and channel bonding occur relative to the encoded or decoded version of the 8B/10B stream. If set to TRUE, the decoded version is used. If set to FALSE, the encoded version is used. Must be set in conjunction with RXDEC8B10USE. CLK_COR_8B10B_DE = RXDEC8B10BUSE
CLK_COR_MAX_LAT	Integer	(0-63) Integer defines the upper bound of the receive FIFO.
CLK_COR_MIN_LAT	Integer	(0-63) Integer defines the lower bound of the receive FIFO.
CLK_COR_SEQ_1_1, 2, 3, 4	11-bit Binary	These define the sequence for clock correction. The attribute used depends on the CLK_COR_SEQ_LEN and CLK_COR_SEQ_2_USE. For details, see Table 3-15, "Definition of CCS or CBS bits 9-0."
CLK_COR_SEQ_1_MASK	4-bit Binary	Each bit of the mask determines if that particular sequence is detected regardless of its value. For example, if bit 0 is High, then CLK_COR_SEQ_1_1 is matched regardless of its value.
CLK_COR_SEQ_2_1, 2, 3, 4	11-bit Binary	These define the sequence for clock correction. The attribute used depends on the CLK_COR_SEQ_LEN and CLK_COR_SEQ_2_USE. For details, See Table 3-15, "Definition of CCS or CBS bits 9-0."
CLK_COR_SEQ_2_MASK	4-bit Binary	Each bit of the mask determines if that particular sequence is detected regardless of its value. For example, if bit 0 is High, then CLK_COR_SEQ_2_1 is matched regardless of its value.

Table 1-12: **RocketIO MGT PCS Attributes *(Continued)***

Attribute	Type	Description
CLK_COR_SEQ_2_USE	Boolean	FALSE/TRUE Control use of second clock correction sequence. FALSE: Clock correction uses only one clock correction sequence defined by CLK_COR_SEQ_1_1... 4, **or** one 8-byte sequence defined by CLK_COR_SEQ_1_1... 4 and CLK_COR_SEQ_2_1... 4 in combination. TRUE: Clock correction uses two clock correction sequences defined by CLK_COR_SEQ_1_1... 4 and CLK_COR_SEQ_2_1... 4, as further constrained by CLK_COR_SEQ_LEN.
CLK_COR_SEQ_LEN	Integer	Integer (1, 2, 3, 4, 8) that defines the length of the sequence the transceiver matches to detect opportunities for clock correction. It also defines the size of the correction, because the transceiver executes clock correction by repeating or skipping entire clock correction sequences.
Alignment		
ALIGN_COMMA_WORD	Integer	Integer controls the alignment of detected commas within the transceiver's 4-byte-wide data path. 1: Aligns commas within a 10-bit alignment range. As a result, the comma is aligned to any byte in the transceivers internal data path. 2: Aligns commas to any 2-byte boundary. 4: Aligns commas to any 4-byte boundary.
COMMA32	Boolean	FALSE/TRUE. If set to TRUE, comma alignment is set for 32 bits. This is used for SONET alignment applications.
COMMA_10B_MASK	10-bit Hex	These define the mask that is ANDed with the incoming serial-bit stream before comparison against PCOMMA_32B_VALUE and MCOMMA_32B_VALUE.
DEC_MCOMMA_DETECT	Boolean	TRUE/FALSE controls the raising of per-byte flag RXCHARISCOMMA on minus-comma.
DEC_PCOMMA_DETECT	Boolean	TRUE/FALSE controls the raising of per-byte flag RXCHARISCOMMA on plus-comma.
DEC_VALID_COMMA_ONLY	Boolean	TRUE/FALSE controls the raising of RXCHARISCOMMA on an invalid comma. FALSE: Raise RXCHARISCOMMA on: `xxx1111100` (if DEC_PCOMMA_DETECT is TRUE) and/or on: `xxx0000011` (if DEC_MCOMMA_DETECT is TRUE) or on 8B/10B translation commas regardless of the settings of the `xxx` bits and the comma being a valid character. TRUE: Raise RXCHARISCOMMA only on valid characters that are in the 8B/10B translation.
MCOMMA_DETECT	Boolean	TRUE/FALSE indicates whether to raise or not raise the RXCOMMADET when minus-comma is detected.

Table 1-12: **RocketIO MGT PCS Attributes *(Continued)***

Attribute	Type	Description
MCOMMA_32B_VALUE	32-bit Hex	These define plus-comma for the purpose of raising RXCOMMADET and realigning the serial bit stream byte boundary. This definition does not affect 8B/10B encoding or decoding. Also see COMMA_32B_MASK. See "SYMBOL Alignment and Detection," page 76.
PCS_BIT_SLIP	Boolean	FALSE/TRUE TRUE: Perform 1 UI bit slip when RXSYNC is set to a logic 1. FALSE: PCS bit slip disabled. See RXSYNC in Chapter 3, "PCS Digital Design Considerations."
PCOMMA_DETECT	Boolean	TRUE/FALSE indicates whether to raise or not raise the RXCOMMADET when plus-comma is detected.
PCOMMA_32B_VALUE	32-bit Hex	These define plus-comma for the purpose of raising RXCOMMADET and realigning the serial bit stream byte boundary. This definition does not affect 8B/10B encoding or decoding. Also see COMMA_32B_MASK. See "SYMBOL Alignment and Detection," page 76.
64B/66B		
SH_CNT_MAX	Integer	8-bit binary. Controls when the 64B/66B synchronization state machine enters synchronization (max sync header count).
SH_INVALID_CNT_MAX	Integer	8-bit binary. Controls when the 64B/66B synchronization state machine leaves synchronization (max invalid sync header count).
Clocks		
RXCLK0_FORCE_PMACLK	Boolean	Controls the clock tree in the PCS (see the Clocking section for more details).
RX_CLOCK_DIVIDER	2-bit Binary	Controls the clock tree in the PCS (see the Clocking section for more details).
RXASYNCDIVIDE	2-bit Binary	Setup the internal clocks. Chapter 2, "Clocking and Timing Considerations" for setting to the correct value.
RXUSRDIVISOR	5-bit Binary	Selects the divide clock for clock received from the PMA and goes to RXRECCLK1 (see the Clocking section for more details).
TXCLK0_FORCE_PMACLK	Boolean	Controls the clock tree in the PCS (see the Clocking section for more details).
TX_CLOCK_DIVIDER	2-bit Binary	Controls the clock tree in the PCS (see the Clocking section for more details).
TXASYNCDIVIDE	2-bit Binary	Setup the internal clocks. Chapter 2, "Clocking and Timing Considerations" for setting to the correct value.
Buffers		
RX_BUFFER_USE	Boolean	TRUE/FALSE. Controls bypassing the RX elastic buffer. FALSE bypasses the buffer.
TX_BUFFER_USE	Boolean	TRUE/FALSE. Controls bypassing the TX FIFO. FALSE bypasses the buffer.

Table 1-13: **RocketIO MGT Digital Receiver Attributes**

Attribute	Type	Description
DCDR_FILTER	3-bit Binary	Attribute should be set to 000.
DIGRX_FWDCLK	2-bit Binary	Select receiver output clock when ENABLE_DCDR is TRUE. See Chapter 2, "Clocking and Timing Considerations" for more details.
DIGRX_SYNC_MODE	Boolean	Set to TRUE in low latency mode and FALSE if CC/CB or buffer are being used. See Chapter 2, "Clocking and Timing Considerations" for more details.
ENABLE_DCDR	Boolean	Select clock and data recovery (CDR) mode: TRUE - data rates of 1.25 Gb/s and below. FALSE - data rates greater than 1.25 Gb/s.
REPEATER	Boolean	TRUE/FALSE. TRUE enables repeater (RX → TX loopback). See Loopback section.
RXBY_32	Boolean	Determines if internal data path is 32 or 40 for the digital receiver.
RXDIGRX	Boolean	Select clock and data recovery (CDR) mode: TRUE - data rates of 1.25 Gb/s and below. FALSE - data rates greater than 1.25 Gb/s.
SAMPLE_8X	Boolean	Determines the type of oversampling that is implemented in the digital receiver. This should always be set to TRUE.
RXDIGRESET	Boolean	Resets the deserializer. TRUE: Reset receiver deserializer. FALSE: Enable receiver deserializer.

Table 1-14: **MGT Tile Communication Attributes**

Attribute	Type	Description
GT11_MODE	String	SINGLE, A, B, DONT CARE Determines which MGT in the MGT tile is simulated. See Chapter 7, "Simulation and Implementation," for details.

Byte Mapping

Most of the 8-bit wide status and control buses correlate to a specific byte of the TXDATA or RXDATA. This scheme is shown in Table 1-15. This creates a way to tie all the signals together regardless of the data path width needed for the GT11_CUSTOM. Byte-mapped signals always appear at the FPGA fabric interface on the same clock cycle.

Table 1-15: **Control/Status Bus Association to Data Bus Byte Paths**

Control/Status Bit	Data Bits
[0]	[7:0]
[1]	[15:8]
[2]	[23:16]
[3]	[31:24]
[4]	[39:32]
[5]	[47:40]
[6]	[55:48]
[7]	[64:56]

Chapter 2

Clocking and Timing Considerations

Clock Domain Architecture

There are eleven clock inputs into each Virtex-4 RocketIO MGT instantiation. Three differential reference clocks (GREFCLK, REFCLK1, and REFCLK2) clock the internal PLLs. It is recommended that for any application over 1 Gb/s, REFCLK1/REFCLK2 are used with the dedicated MGTCLK inputs. **These inputs should never be driven from a DCM, because the jitter control is optimized on these clock nets to go directly from package pins to the MGTs.** Only one of these reference clocks is needed to run the MGT. However, multiple clocks can be used to implement multi-rate designs. Most of the four user clocks (TXUSRCLK, TXUSRCLK2, RXUSRCLK, and RXUSRCLK2) can be created within the MGT block without the use of a DCM. However, the reference clocks or several MGT clock outputs can drive the DCMs that in turn create the necessary clocks. This chapter also includes several use models. While these are the most common configurations, other configurations are also possible. The clocking attributes and serial speed determine the reference clock speed, which is shown in Table 1-3, page 21.

Clock Ports

The MGT has four groups of clocks: reference, user, MGT output, and CRC logic clocks. The clock ports are shown in Table 2-1.

Table 2-1: **Clock Ports**

Clock	I/O	Description
GREFCLK	I	Alternate reference clock (only used for 1G or less applications).
REFCLK1	I	Reference clock for the TX and RX PLLs. The multiplication ratio for parallel-to-serial conversion is application dependent.
REFCLK2	I	Reference clock for the TX and RX PLLs. The multiplication ratio for parallel-to-serial conversion is application dependent.
RXCRCCLK	I	Clocks the receiver CRC logic.
RXCRCINTCLK	I	Clocks the receiver CRC and FPGA interfacing logic.
RXRECCLK1/ RXPCSHCLKOUT	O	Recovered clock from incoming data.
RXRECCLK2	O	Recovered clock from incoming data.
RXUSRCLK	I	Clocks the receiver PCS internal logic.
RXUSRCLK2	I	Clocks the receiver PCS/ FPGA fabric interface.
TXCRCCLK	I	Clocks the transmitter CRC logic.
TXCRCINTCLK	I	Clocks the transmitter CRC and FPGA interfacing logic.

Table 2-1: **Clock Ports *(Continued)***

Clock	I/O	Description
TXOUTCLK1/ TXPCSHCLKOUT	O	Transmitter output clock derived from PLL based on transmitter reference clock. Can be used to clock the FPGA.
TXOUTCLK2	O	Transmitter output clock derived from PLL based on transmitter reference clock. Can be used to clock the FPGA. TXOUTCLK1/TXPCSHCLKOUT should always be used as alternatives for this port.
TXUSRCLK	I	Clocks the transmitter PCS internal logic.
TXUSRCLK2	I	Clocks the transmitter PCS/FPGA fabric interface.

Clock Distribution

The MGT clock distribution has changed to support the column-based structure of -Virtex-4 devices. A column consists of multiple MGT tiles (which contain two MGTs each). The MGT tile contains routing for the PLL reference clocks and the clocks that are derived from the PLLs. The clocks derived from the PLLs are forwarded to the FPGA global clock resources. Two low-jitter reference clock trees (SYNCLK1 and SYNCLK2) run the entire length of the column. These SYNCLKs have the ability to route a clock completely up and down a column or to become tile local clocks. See "GT11CLK_MGT and Reference Clock Routing," page 51 for more details. In a tile, each PLL can select its own RX reference clock and a shared TX reference clock. There are three reference inputs to choose from: the two column SYNCLKs which drive the REFCLK1 and REFCLK2 inputs of the MGTs and the tile local GREFCLK (for applications below 1 Gb/s). The attributes in Figure 2-1 and Table 2-2 show how to select the reference clock for the three PLLs in a tile.

Table 2-2: **Clock Selection for Three PLLs in a Tile**

Attribute String Value	RXPMACLKSEL (MGT A)	RXPMACLKSEL (MGT B)	TXPMACLKSEL (MGT Tile)
REFCLK1	REFCLK1 vertical bus supplies MGT A RX PLL	REFCLK1 vertical bus supplies MGT B RX PLL	REFCLK1 vertical bus supplies TX PLL for both MGTs in the tile
REFCLK2	REFCLK2 vertical bus supplies MGT A RX PLL	REFCLK2 vertical bus supplies MGT B RX PLL	REFCLK2 vertical bus supplies TX PLL for both MGTs in the tile
GREFCLK[1]	GREFCLK supplies the MGT A RX PLL	GREFCLK supplies the MGT B RX PLL	GREFCLK supplies TX PLL for both MGTs in the tile

Notes:

1. Should only be used for 1 Gb/s or lower serial rates.

The MGTCLK inputs drive the reference clocks that are low jitter and must be used for the fastest data rates (over 1 Gb/s). The global clocks (GREFCLK) can be used as a reference clock at lower data rates (1 Gb/s or lower). It also can only drive one tile unless using other FPGA fabric routing resources outside of the MGT tiles. Clocks derived from the MGT or MGT clock input can be forwarded to the FPGA global clock resources. See "GT11CLK_MGT and Reference Clock Routing," page 51.

The recovered clock of MGT B can be fed to the MGTCLK (shown in Figure 2-1) to be used as a reference clock for that tile or other tiles in the column. This is accomplished by the SYNCLK vertical clock buses and the reference clock routing block (shown in Figure 2-1 and more details in Figure 2-5). See the "GT11CLK_MGT and Reference Clock Routing," page 51 for more information on this block.

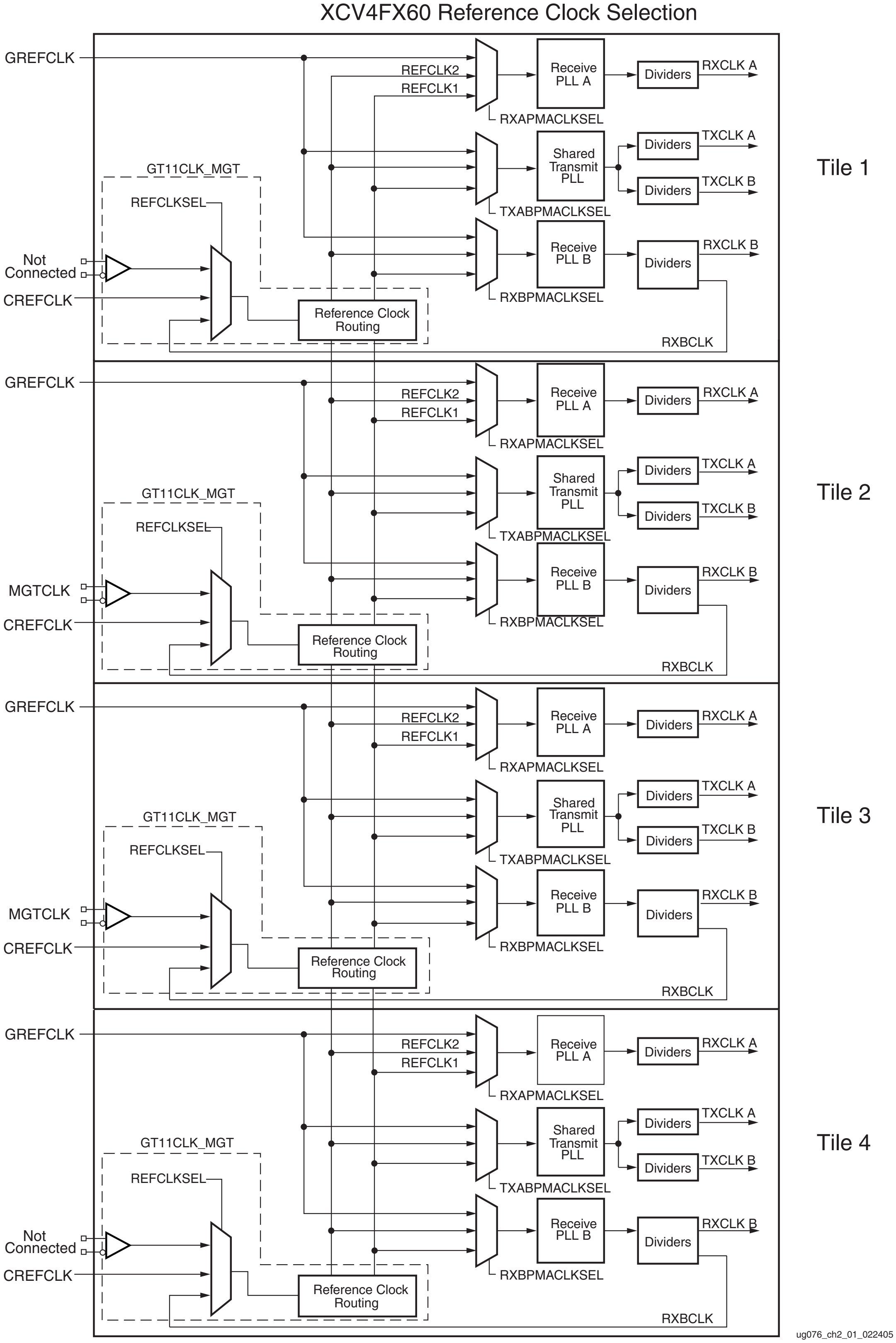


Figure 2-1: **MGT Column Clocking**

RXCLKSTABLE and TXCLKSTABLE

In some systems, there are cases where the reference clock is not available when the MGT and FPGA come out of configuration. The MGT PLL needs to know when its reference clock is stable and when to try to lock. One such case is when external cleanup PLLs are used, which is when RXCLKSTABLE and TXCLKSTABLE should be used. Figure 2-2 shows how these signals can be used. When the reference clock is known to be ready before the MGT, these signals should be set to a logic 1.

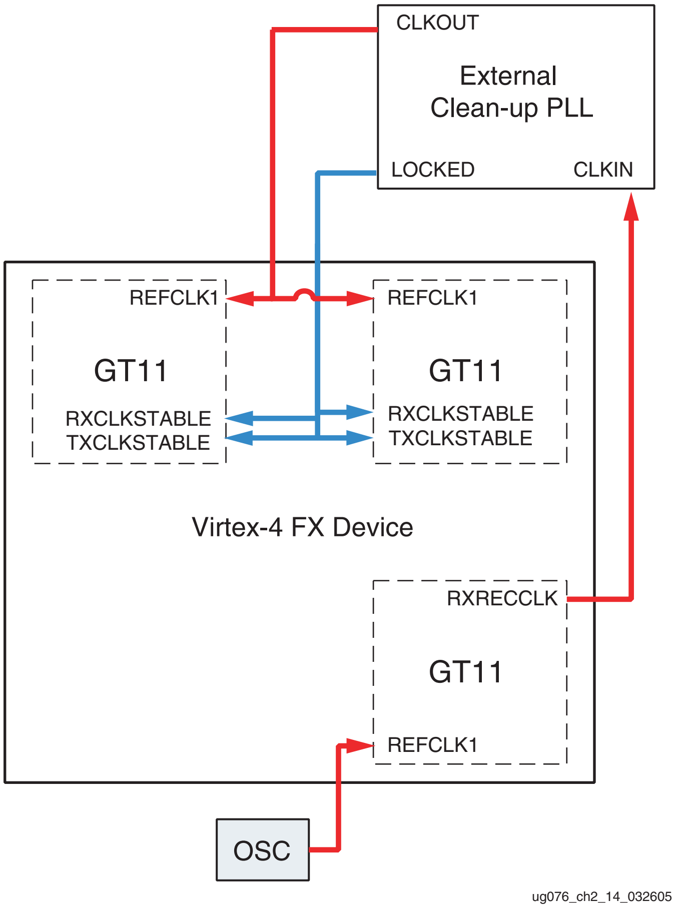


Figure 2-2: **External PLL Locked Signal for MGT**

PMA Transmit Clocks

The PMA transmit block has several clocking options. These options include several dividers that determine the relationship between the parallel clocks (TXOUTCLK1 and TXOUTCLK2) and the serial rates on TXP/TXN. These dividers are shown in Figure 2-3. Table 2-3 shows possible values of the TX PMA attributes that are used to create the clocking relationships.

Note: Always use TXOUTCLK1 or TXPCSHCLKOUT, not TXOUTCLK2.

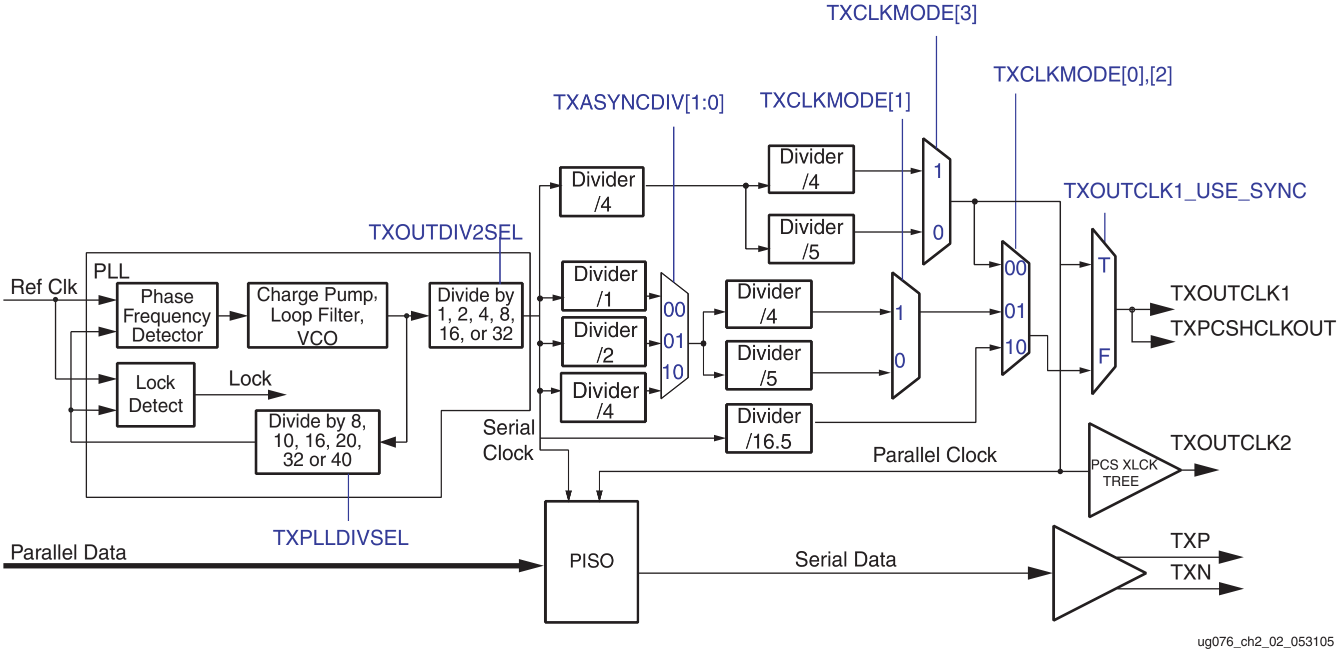


Figure 2-3: **MGT Transmit Clocking**

Table 2-3: **TX PMA Attribute Values**[1]

Attribute	Available Values	Definition
TXPLLDIVSEL	40, 32, 20, 16, 10, 8	Transmit PLL feedback divide. This value is also the PLL multiplication factor for the reference clock. Select divide by 40, multiply reference clock by 40 (DRP value 1010). Select divide by 32, multiply reference clock by 32 (DRP value 1000). Select divide by 20, multiply reference clock by 20 (DRP value 0110). Select divide by 16, multiply reference clock by 16 (DRP value 0100). Select divide by 10, multiply reference clock by 10 (DRP value 0010). Select divide by 8, multiply reference clock by 8 (DRP value 0000).
TXOUTDIV2SEL	1, 2, 4, 8, 16, 32	Transmitter PLL output divide. DRP values are: 1 = 0001 2 = 0010 4 = 0011 8 = 0100 16 = 0101 32 = 0110
TXASYNCDIVIDE	00, 01, 10, 11	Async Divide: 00 = Divide by 1 01 = Divide by 2 10 = Divide by 4 11 = Divide by 4
TXCLKMODE	0110, 0100, 1000, 1001, 0000, 1111	Divider Control for TXCLKSYNC and TXCLKASYNC
TXOUTCLK1_USE_SYNC	TRUE, FALSE	FALSE: TXOUTCLK1 = TXCLKASYNC TRUE: TXOUTCLK1 = TXCLKSYNC

Notes:

1. See Figure 2-12 and Figure 2-13 for application specific settings.

PMA Receive Clocks

The PMA receive block has several clocking options. These options include several dividers that determine the relationship between the parallel clocks (RXRECCLK1 and RXRECCLK2) and the serial rates on RXP/RXN. These dividers are shown in Figure 2-4. Table 2-4 shows possible values of these RX PMA attributes that are used to create the clocking relationships.

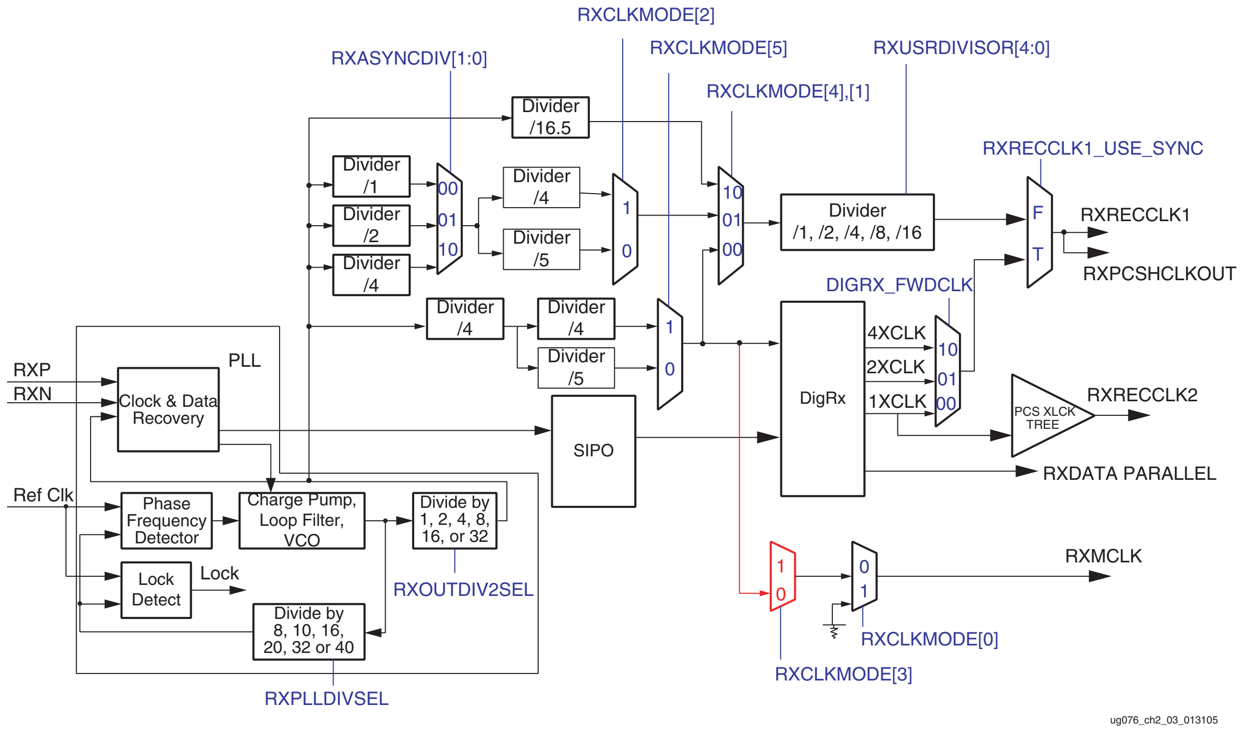


Figure 2-4: **MGT Receive Clocking**

Table 2-4: **RX PMA Attribute Values**(1)

Attribute	Available Values	Definition
RXPLLDIVSEL	8, 10, 16, 20, 16, 32, 40	Receiver PLL feedback divide. This value is also the PLL multiplication factor for the reference clock. Select divide by 40, multiply reference clock by 40 (DRP value 1010). Select divide by 32, multiply reference clock by 32 (DRP value 1000). Select divide by 20, multiply reference clock by 20 (DRP value 0110). Select divide by 16, multiply reference clock by 16 (DRP value 0100). Select divide by 10, multiply reference clock by 10 (DRP value 0010). Select divide by 8, multiply reference clock by 8 (DRP value 0000).
RXOUTDIV2SEL	1, 2, 4, 8, 16, 32	Receiver PLL output divide. DRP values are: 1 = 0001 2 = 0010 4 = 0011 8 = 0100 16 = 0101 32 = 0110

Table 2-4: **RX PMA Attribute Values**[1] ***(Continued)***

Attribute	Available Values	Definition
RXCLKMODE	See Figure 2-10 through Figure 2-13.	Selects receiver output clocks and 32- or 40-bit PMA output data path width. See "Setting the Clocking Options" for correct settings.
RXASYNCDIVIDE	00, 01, 10, 11	Async Divide: 00= Divide by 1 01= Divide by 2 10= Divide by 4 11= Divide by 4
RXDIG_FWDCLK	00, 01, 10, 11	00 = 1XCLK 01 = 2XCLK 10 = 4XCLK
RXRECCLK1_USE_SYNC	TRUE, FALSE	FALSE: RXRECCLK1 = synchronous clock TRUE: RXRECCLK1 = asynchronous clock

Notes:

1. See Figure 2-10 and Figure 2-11 for application specific settings.

GT11CLK_MGT and Reference Clock Routing

Each MGT tile contains a GT11CLK_MGT block which can be implemented by instantiating either GT11CLK_MGT or GT11CLK ports and attributes shown in Table 2-5. This block allows more clocking options for MGTs in a given column, including feeding the fabric clock trees via SYNCLK1 and SYNCLK2.

Table 2-5: **MGTCLK Ports and Attributes**

GT11CLK	GT11CLK_MGT	Description
Ports		
SYNCLK1OUT	SYNCLK1OUT	This output drives the REFCLK1 column bus and the FPGA clock trees.
SYNCLK2OUT	SYNCLK2OUT	This output drives the REFCLK2 column bus and the FPGA clock trees.
MGTCLKN	MGTCLKN	This is the differential package input for the MGT column.
MGTCLKP	MGTCLKP	
REFCLK	REFCLK	This input is from the FPGA fabric (indicated by CREFCLK in Figure 2-1). This reference clock should only be used in sub 1 Gb/s operation.
RXBCLK	n/a	This input is directly connected to the RXMCLK of the MGT instance.
SYNCLK1IN	n/a	These inputs should be connected to the REFCLK1 and REFCLK2, respectively.
SYNCLK2IN	n/a	

Table 2-5: **MGTCLK Ports and Attributes *(Continued)***

GT11CLK	GT11CLK_MGT	Description
Attributes		
REFCLKSEL	n/a	Determines which clock input is used for the reference clock (MGTCLK, RXBCLK).
SYNCLK1OUTEN	SYNCLK1OUTEN	Allows the SYNCLK1OUT to drive the REFCLK1 column bus.
SYNCLK2OUTEN	SYNCLK2OUTEN	Allows the SYNCLK2OUT to drive the REFCLK2 column bus.

High-speed dedicated clock pins are implemented through the GT11CLK_MGT module. There are two of these connections bonded out per column. These locations and package pins are shown in Chapter 7, "Simulation and Implementation." Table 7-6, page 146 shows which tiles allow this connection for a specific device.

To implement such a connection, the GT11CLK_MGT should be instantiated. This is shown in Figure 2-5.

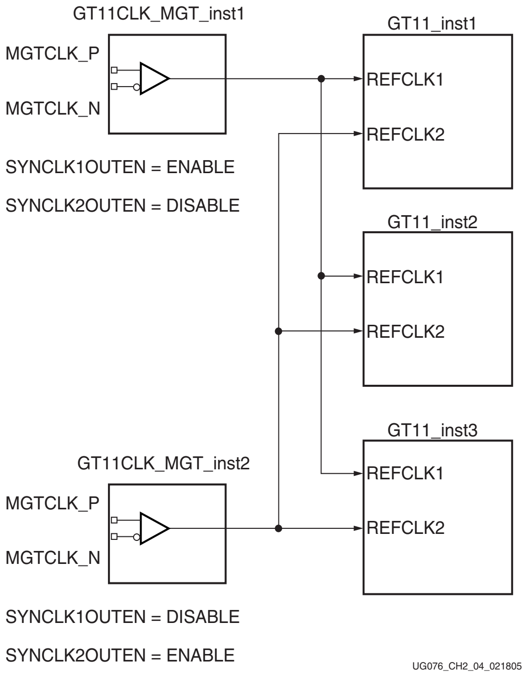


Figure 2-5: **High-Speed Dedicated Clocks (GT11CLK_MGT Instance)**

This MGTCLK allows a recovered clock from MGT B of the tile as a reference clock. Note that this is not sourced from the RXRECCLK, but another signal (RXMCLK) in the MGT tile. **This is a test mode only**. This signal frequency is determined by RXCLKMODE and shown in Figure 2-6. This clock can be used if the TX and RX need to be an exact frequency.

This is a higher jitter clock and should only be used for applications under 1 Gb/s serial rates.

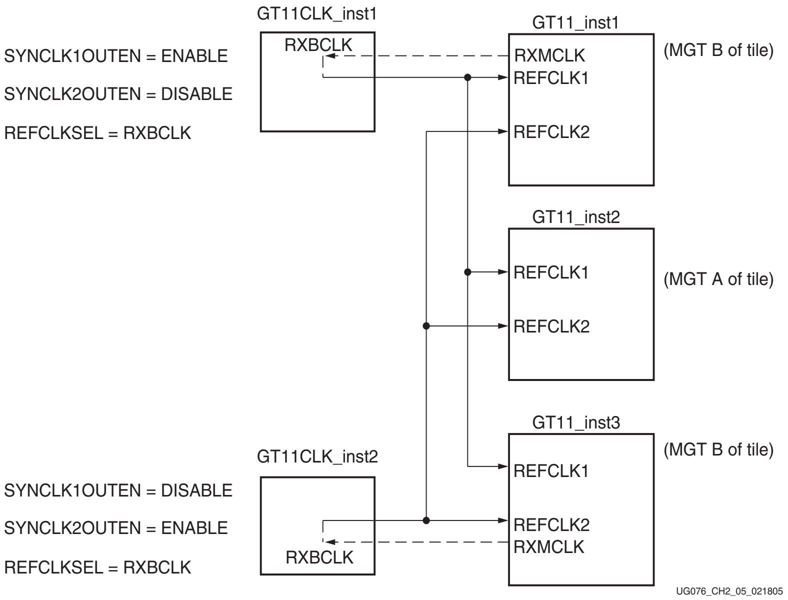


Figure 2-6: **Recovered Clock as REFCLK (1 Gb/s)(GT11CLK Instance)**

GREFCLK

GREFCLK is a clock that feeds the MGT PLLs of an MGT tile only. GREFCLK can supply all the MGTs in a column using fabric interconnect. For this application, the REFCLK input of the GT11CLK should be used to reduce the amount of fabric clocking resources. In this configuration, the reference clock is called CREFCLK (column GREFCLK). Figure 2-7 shows both of these options.

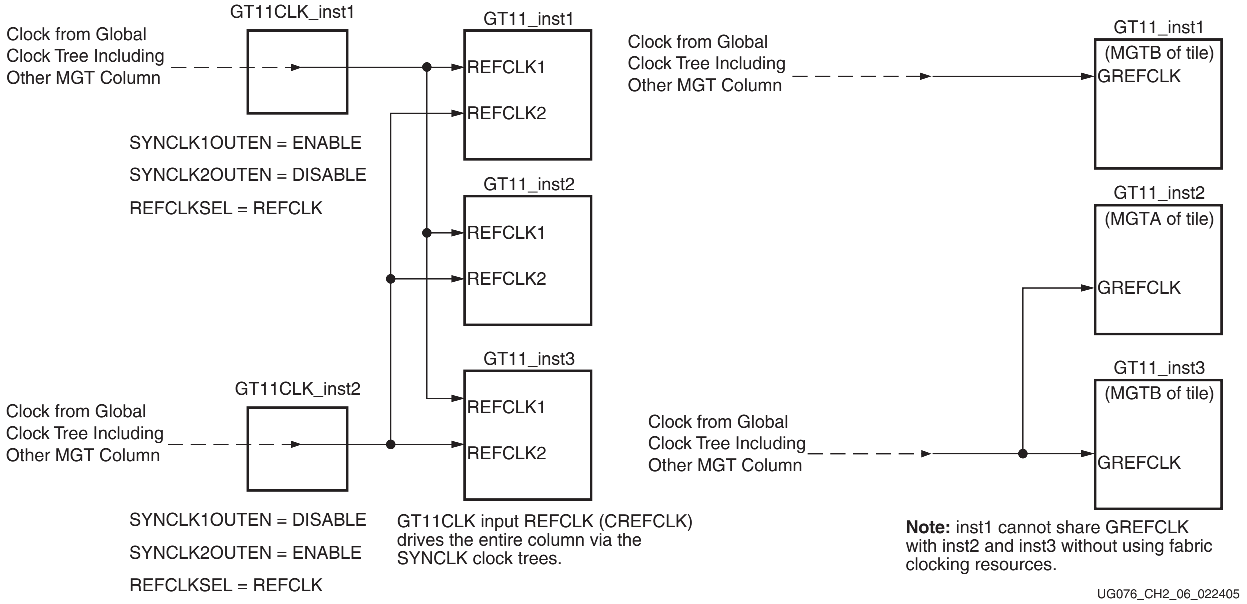


Figure 2-7: **CREFCLK and GREFCLK Options for an MGT Tile**

PCS Timing Considerations

The relationship between the USRCLK and USRCLK2 (both TX and RX) depends on several factors, including the parallel data width and serial standard used.

Another important timing consideration is the clock delay for data to pass through the entire MGT. Table 2-6 and Table 2-7 show approximate clock cycles for the main PCS blocks. The cycles depend on the external parallel data bus width (shown), plus the phase relationship between the different clocks, which adds a small uncertainty factor to this table.

Table 2-6: **Latency Through Various Transmitter Components/Processes**[1]

Transmit Blocks →	1 Byte	2 Byte	4 Byte	8 Byte
Fabric Interface[2]	4 TXUSRCLK2 + 1 TXUSRCLK	2 TXUSRCLK2 + 1 TXUSRCLK	1 TXUSRCLK2 + 1 TXUSRCLK	2 TXUSRCLK2 + 1 TXUSRCLK
Encoding[3]				
8B/10B	3 TXUSRCLK			
64B/66B	3 TXUSRCLK			
Bypass	1 TXUSRCLK			
TXBUFFER[3]				
TXFIFO	4 TXUSRCLK (±.5)			
64B/66B Format[3]				
64B/66B Scrambling & Gearbox	2 TXCLK0			
Bypass	1 TXCLK0			
PMA Convert	1 TXCLK0 (approx.)			
TXSERDES	1 TXCLK0 Input Register			

Notes:

1. See Chapter 5, "Cyclic Redundancy Check (CRC)" on CRC Latency.
2. Fabric interface has delays in both clock domains. Clock ratios are: 1-byte mode USR2:USR ratio = 4:1; 2-byte mode USR2:USR ratio = 2:1; 4-byte mode USR2:USR ratio = 1:1; and 8-byte mode USR2:USR ratio = 1:2.
3. These delays include registered data mux which counts for 1 clock delay.

Table 2-7: **Latency Through Various Receiver Components/Processes**[1][2]

Receive Blocks	1 Byte	2 Byte	4 Byte	8 Byte
RX SERDES	1/32 or 1/40 of serial clock			
PMA_PCS Interface[3]	2 RXCLK0			
RX Data Alignment[3]				
CommaDET/Align	3 RXCLK0			
Bypass	1 RXCLK0			
Decoding[4]				
8B/10B	2 RXCLK0			
64B/66B	2 RXCLK0			
Bypass	1 RXCLK0			

Table 2-7: **Latency Through Various Receiver Components/Processes**[1][2] ***(Continued)***

Receive Blocks	1 Byte	2 Byte	4 Byte	8 Byte
RXFIFO[5]				
No Clock Correction	3 RXCLK0 + RXUSR (phase diff) + 8 RXUSR Latency + 1 RXUSRMUX			
Clock Correction (Min/Max Used)	(3 RXCLK0 + 1 RXUSRCLK + MIN_LAT/4) < latency < (3 RXCLK0 + 1 RXUSRCLK + MAX_LAT/4)			
64B/66B Format				
Decode	2 RXUSRCLK			
Bypass	1 RXUSRCLK			
Fabric Interface[6]	1 USRCLK + 4 USRCLK2	1 USRCLK + 2USRCLK2	1 USRCLK + 1 USRCLK2	2 USRCLK + 1 USRCLK2

Notes:

1. See Chapter 5, "Cyclic Redundancy Check (CRC)" for CRC latency.
2. Neither linear nor DFE equalization affect the receiver latency.
3. These delays include registered data mux that counts for one clock of delay.
4. Bypass is when RXDEC8B10BUSE and RXDEC64B66BUSE are both deasserted, plus register data mux accounts for one of the clocks of delay.
5. Bypassed FIFO accounts for one registered data mux = 1 RXUSRCLK, but must be equal to 1 RXCLK0 to bypass buffer. (Only works reliably if RXCLK0 sync mode is implemented.)
6. Fabric interface has delays in both clock domains. Clock ratios are: 1-byte mode USR2:USR ratio = 4:1; 2-byte mode USR2:USR ratio = 2:1; 4-byte mode USR2:USR ratio = 1:1; and 8-byte mode USR2:USR ratio = 1:2.

Reduced Latency Mode

The RocketIO transceiver contains two buffers, one for the transmitter and the other an elastic receive buffer that supports clock correction and channel bonding and allows for phase differences between the RXCLK and the RXUSRCLK, as shown in Figure 2-8. However, some applications need low latency datapaths from the RXN/RXP pins and the RXDATA ports (a similar requirement exists on the transmit side). The MGT allows a reduced latency mode that synchronizes all the clock domains inside the transceiver. This allows bypassing the buffers to reduce the latency, however several other paths (shown in Figure 2-8 and Figure 2-9) are also available to bypass all the functionality of the PCS that essentially emulates a stand alone SERDES.

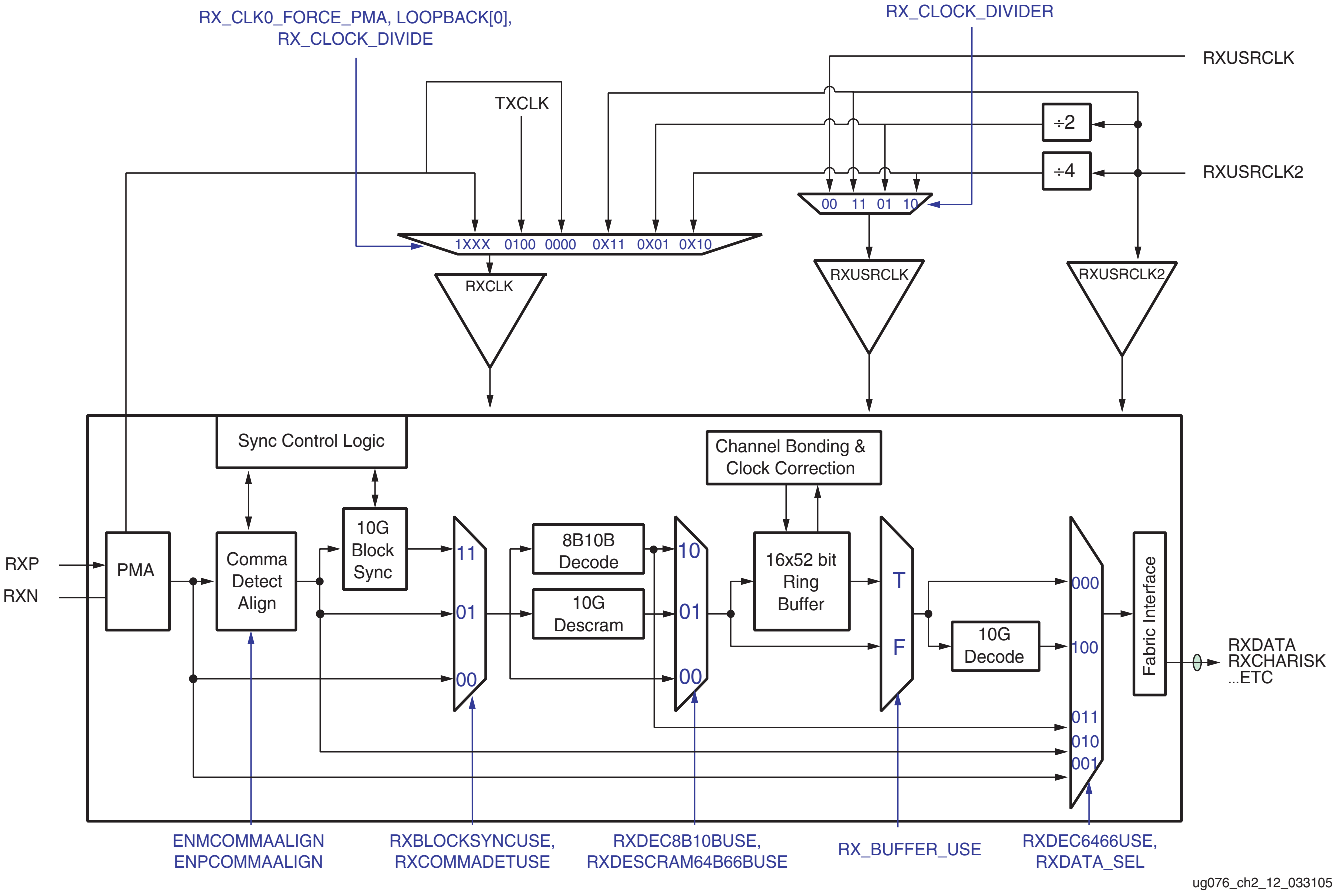


Figure 2-8: **PCS Receive Clocking Domains and Datapaths**

If the shortest path is taken, the RX latency is reduced to the latency of the RX SERDES, PMA_PCS interface, and fabric interface (shown in Table 2-6) ,which results in a latency of less than five USRCLK2 cycles in the 4-byte mode. On the transmit side, the only blocks to consider in the delay latency is the Fabric interface, PMA convert, and TX SERDES (shown in Table 2-7), which also results in less than five USRCLK2 cycles in the 4-byte mode.

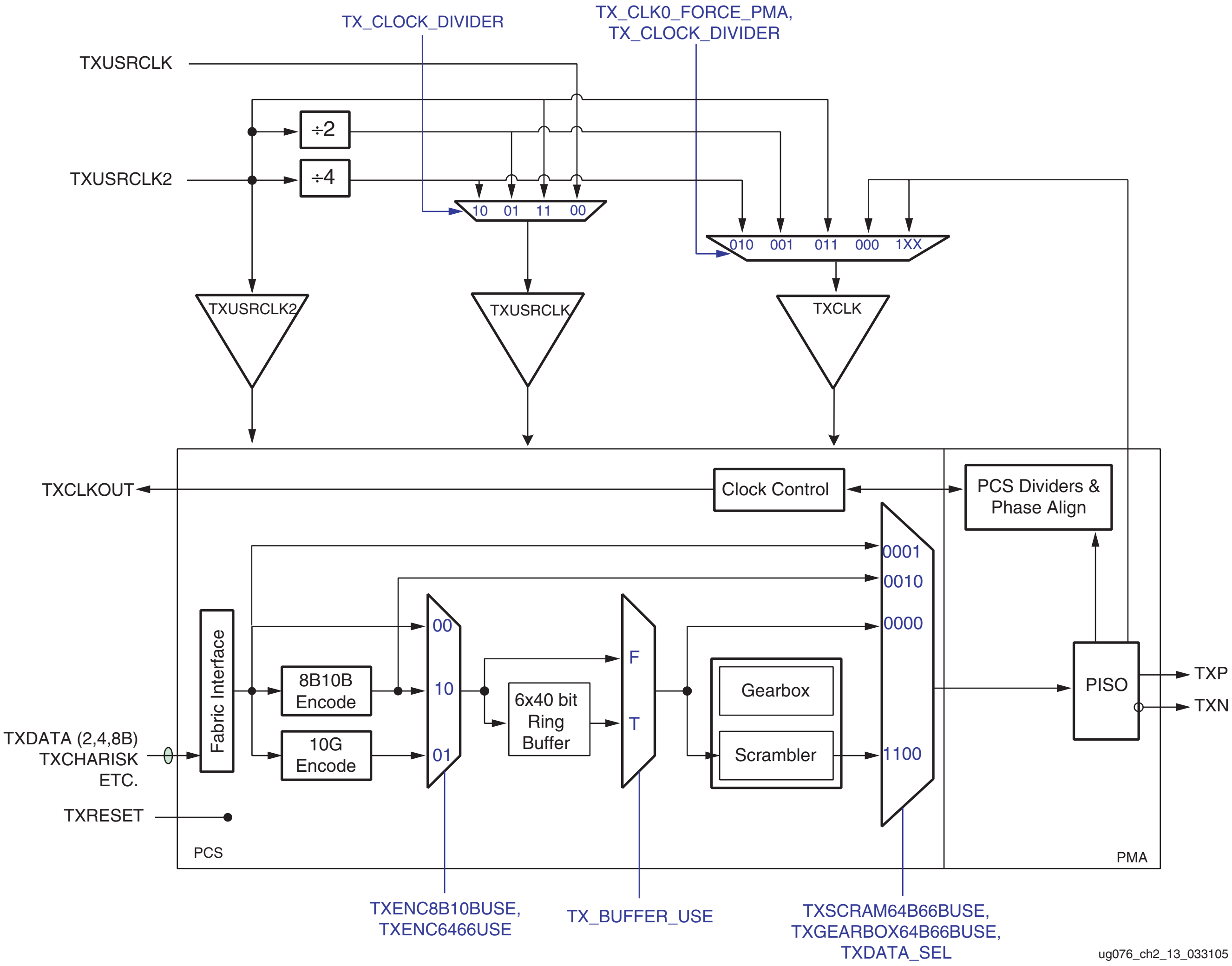


Figure 2-9: **PCS Transmit Clocking Domains and Datapaths**

There are several attributes that allow the reduced latency mode to operate properly. These are shown in Table 2-8. The clocking decision trees (Figure 2-10 through Figure 2-13) show how to set these and all the other clocking variables for the transmitter and receiver.

Table 2-8: **Attributes Supporting Reduced Latency Mode**

Attribute	Bypass importance
RXDATA_SEL	Select which bypass mode is enabled.
RXDEC64B66BUSE	Helps determine bypass mode in conjunction of RXDATA_SEL.
TXDATA_SEL	Select which bypass mode is enabled.
TXSCRAM66B64BUSE	Helps determine bypass mode in conjunction of TXDATA_SEL.
TXGEARBOX64B66BUSE	Helps determine bypass mode in conjunction of TXDATA_SEL.
RX_CLK0_FORCE_PMA	Determines if the PMA is synchronized to the RXCLK.
TX_CLK0_FORCE_PMA	Determines if the PMA is synchronized to the TXCLK.

Reduced latency mode has several restrictions. Because the RX elastic buffer is bypassed, clock correction and channel bonding are not supported, but can be implemented in the fabric. Also, the 8-byte mode is not supported along with the 64B/66B support, because the reduced latency does not work with the gearbox functionality.

Synchronizing the PMA and PCS clocks

It is recommended that the PLL be locked (indicated by TXLOCK or RXLOCK). At this point, TXSYNC or RXSYNC can be transitioned from a logic 0 to a logic 1, so that the PMA can synchronize the clocks. This synchronizing is required for the reduced latency mode.

Note: BIT_SLIP_MODE = 0

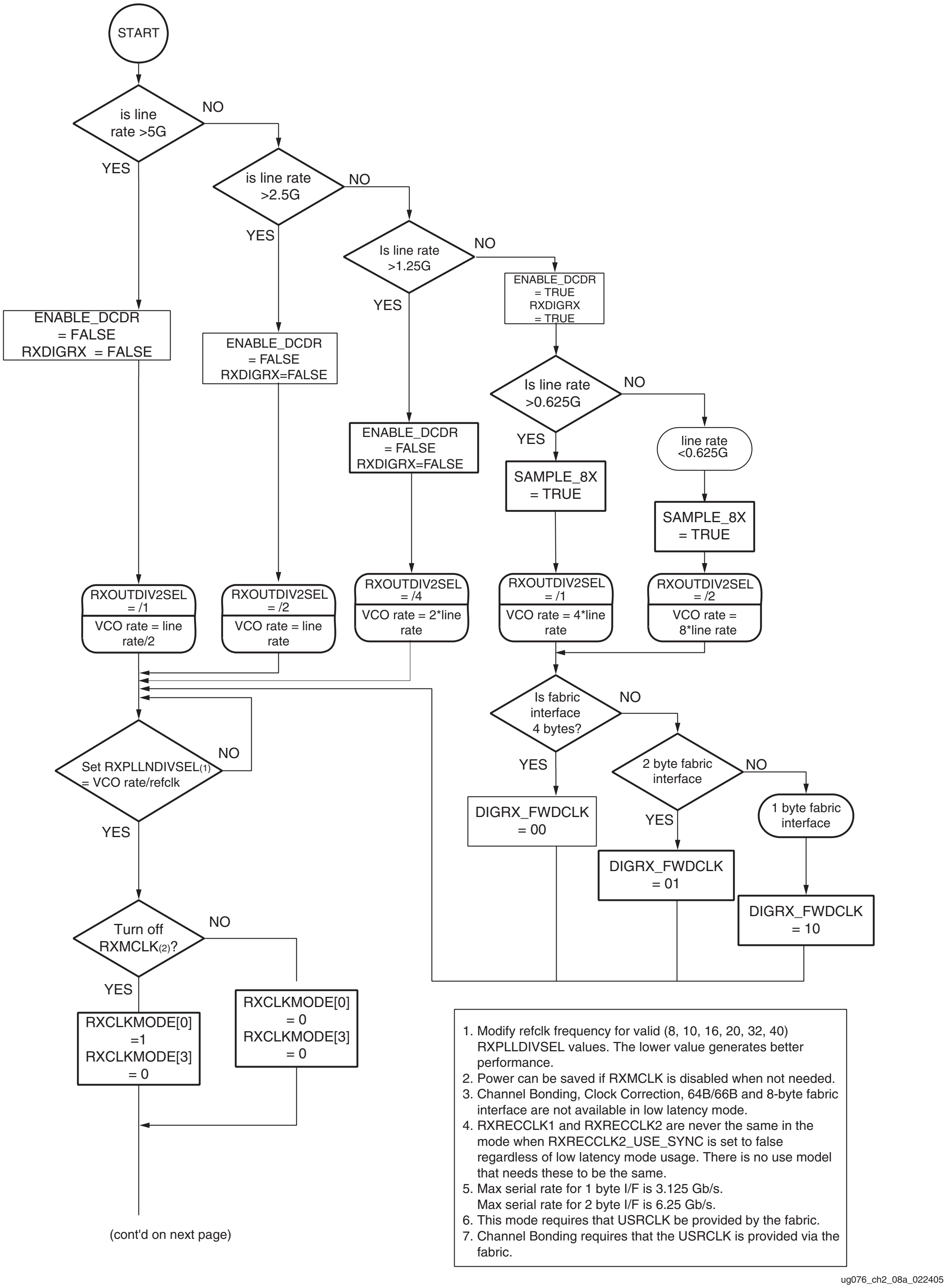
PMA Configurations

There are several configurations of the PMA that also affect serial speeds and clocking schemes. These configurations can be modified by the Dynamic Configuration Bus or with attributes. These settings are covered in Figure 2-10 through Figure 2-13.

Setting the Clocking Options

Because of the MGT's flexibility, there are many different clocking modes available. In most cases, the Architecture Wizard should be used to create the proper instantiation. However, some unique protocols might not be supported with protocol driven modules of the Architecture Wizard. In this case, several decision flow diagrams are available (Figure 2-10 through Figure 2-13) to determine how to set the clocking dividers for a specific application based on the fabric interface, serial rate, encoding, and other important considerations.

Note: The reference clock can directly drive the USRCLKs. Because most cases require multiple frequencies to clock data, it is recommended to use TXOUTCLK1 (or TXPCSHCLKOUT) and RXRECCLK1 (or RXPCSHCLKOUT) and the internal clock dividers.



1. Modify refclk frequency for valid (8, 10, 16, 20, 32, 40) RXPLLDIVSEL values. The lower value generates better performance.
2. Power can be saved if RXMCLK is disabled when not needed.
3. Channel Bonding, Clock Correction, 64B/66B and 8-byte fabric interface are not available in low latency mode.
4. RXRECCLK1 and RXRECCLK2 are never the same in the mode when RXRECCLK2_USE_SYNC is set to false regardless of low latency mode usage. There is no use model that needs these to be the same.
5. Max serial rate for 1 byte I/F is 3.125 Gb/s. Max serial rate for 2 byte I/F is 6.25 Gb/s.
6. This mode requires that USRCLK be provided by the fabric.
7. Channel Bonding requires that the USRCLK is provided via the fabric.

Figure 2-10: **Receive Clocking Decision Flow (Page 1 of 2)**

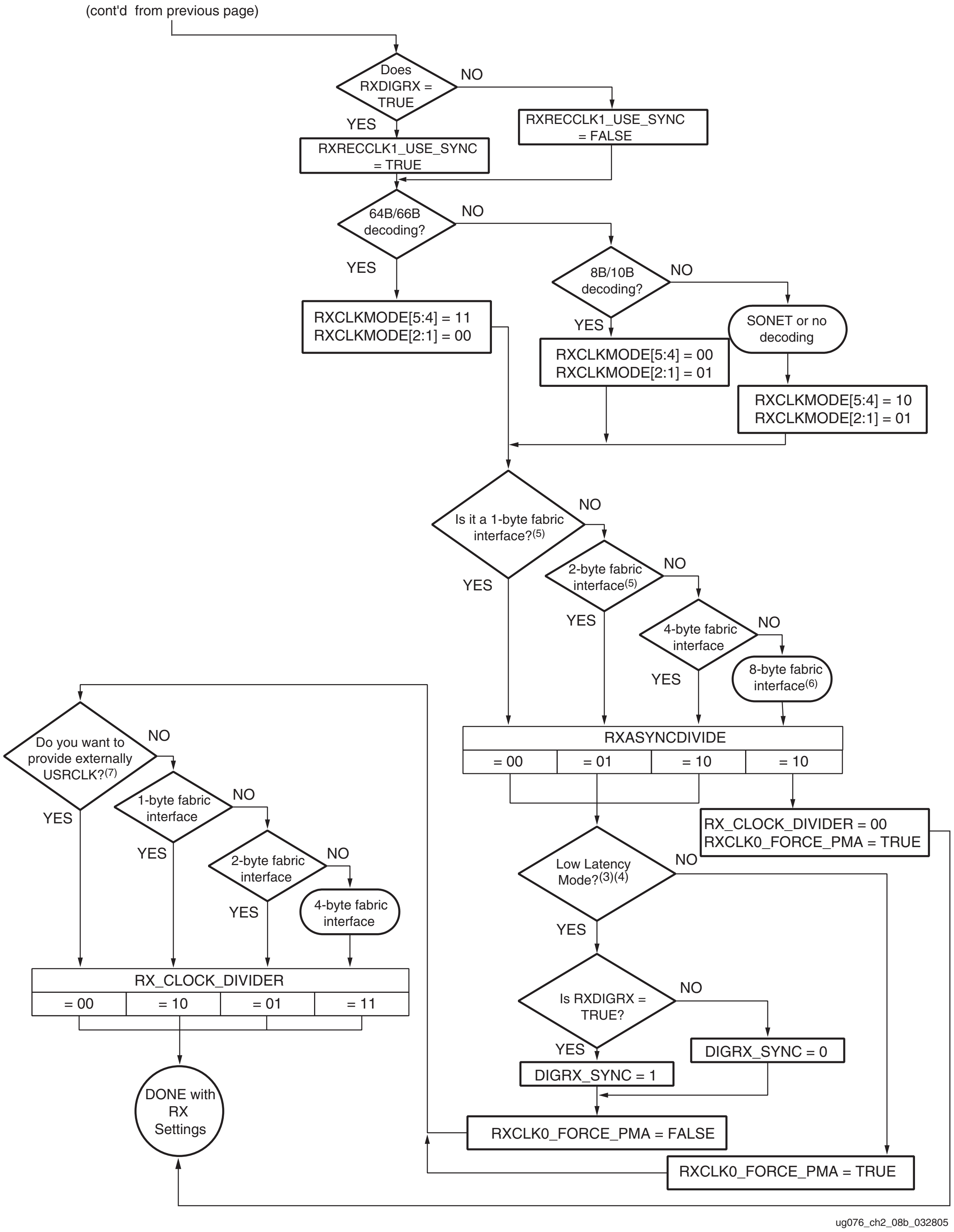
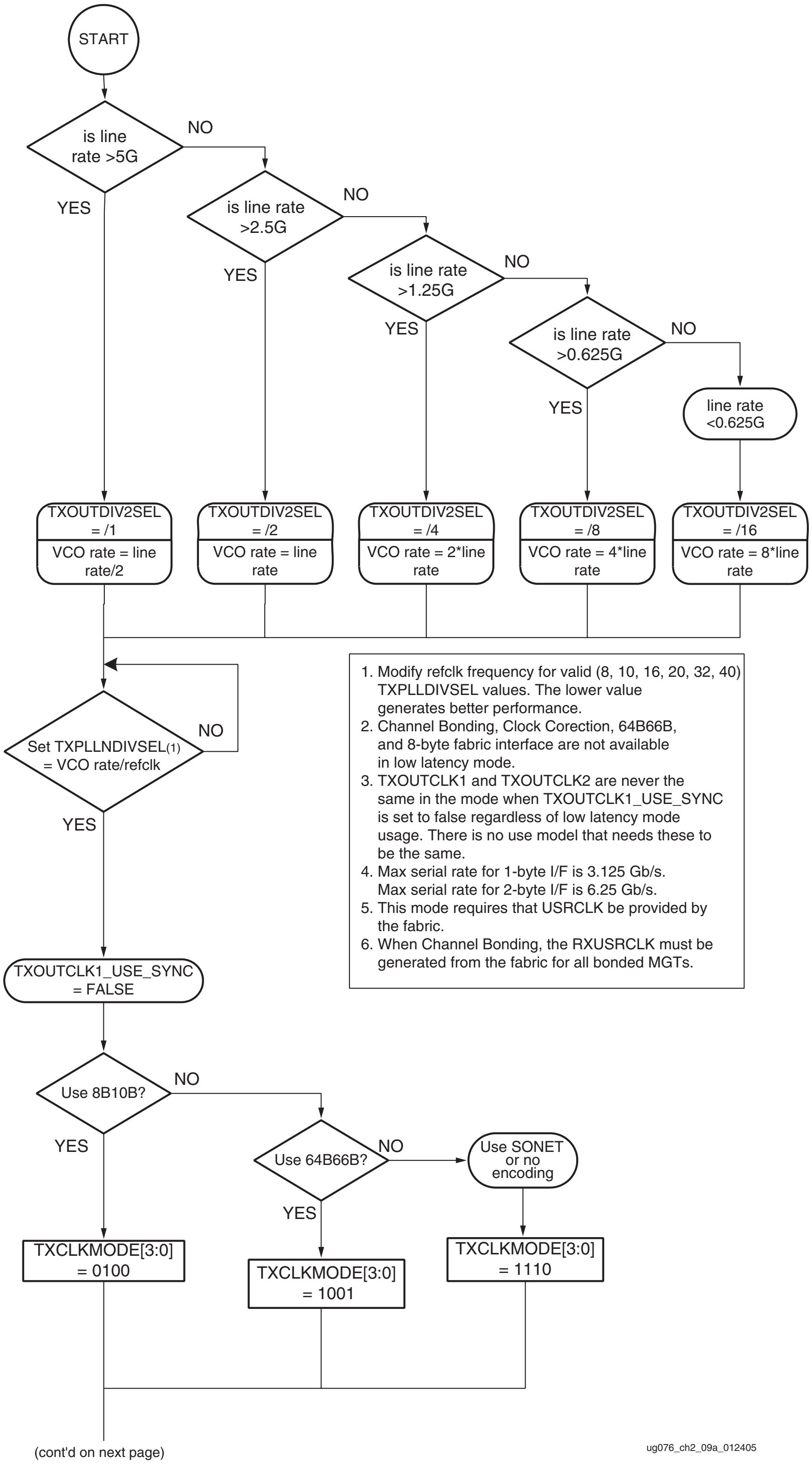


Figure 2-11: **Receive Clocking Decision Flow (Page 2 of 2)**



1. Modify refclk frequency for valid (8, 10, 16, 20, 32, 40) TXPLLDIVSEL values. The lower value generates better performance.
2. Channel Bonding, Clock Corection, 64B66B, and 8-byte fabric interface are not available in low latency mode.
3. TXOUTCLK1 and TXOUTCLK2 are never the same in the mode when TXOUTCLK1_USE_SYNC is set to false regardless of low latency mode usage. There is no use model that needs these to be the same.
4. Max serial rate for 1-byte I/F is 3.125 Gb/s. Max serial rate for 2-byte I/F is 6.25 Gb/s.
5. This mode requires that USRCLK be provided by the fabric.
6. When Channel Bonding, the RXUSRCLK must be generated from the fabric for all bonded MGTs.

Figure 2-12: **Transmit Clocking Decision Flow (Page 1 of 2)**

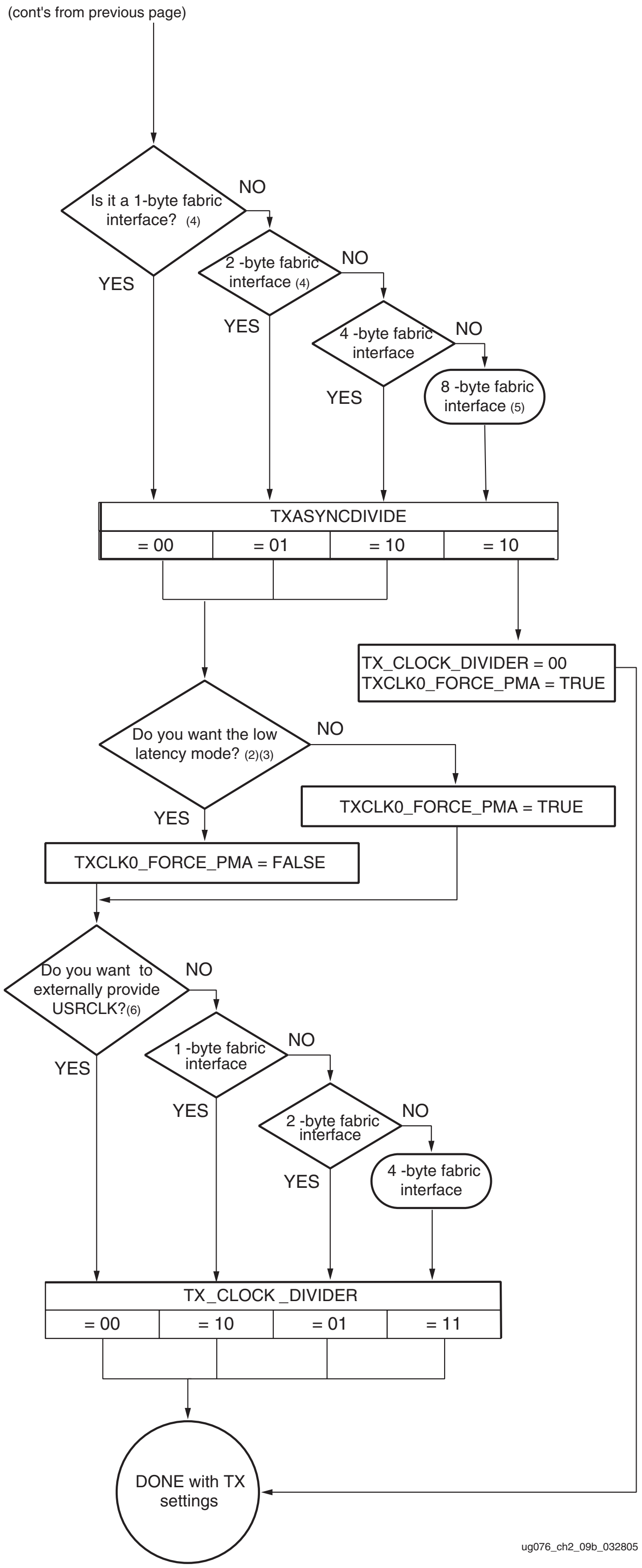
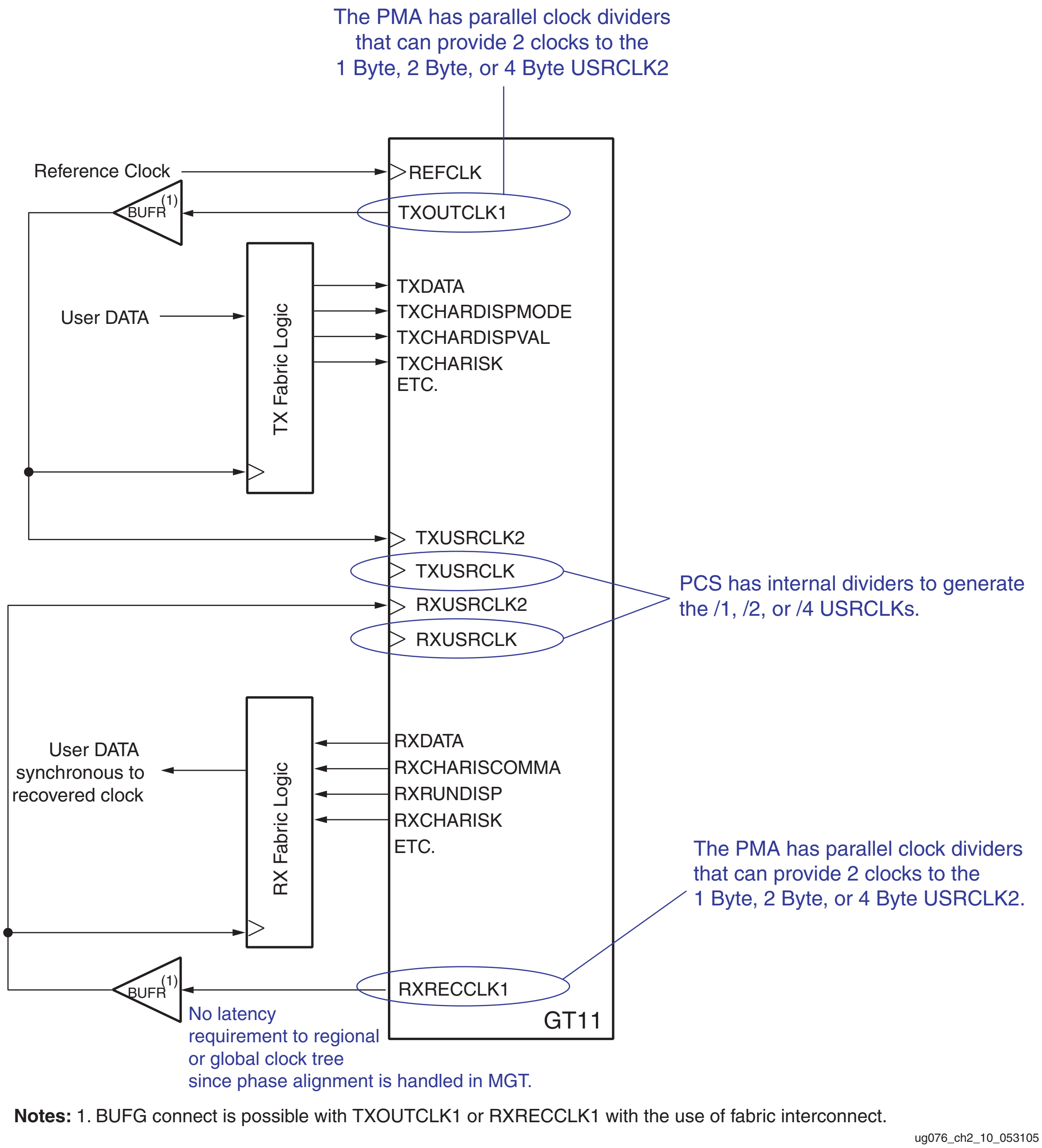


Figure 2-13: **Transmit Clocking Decision Flow (Page 2 of 2)**

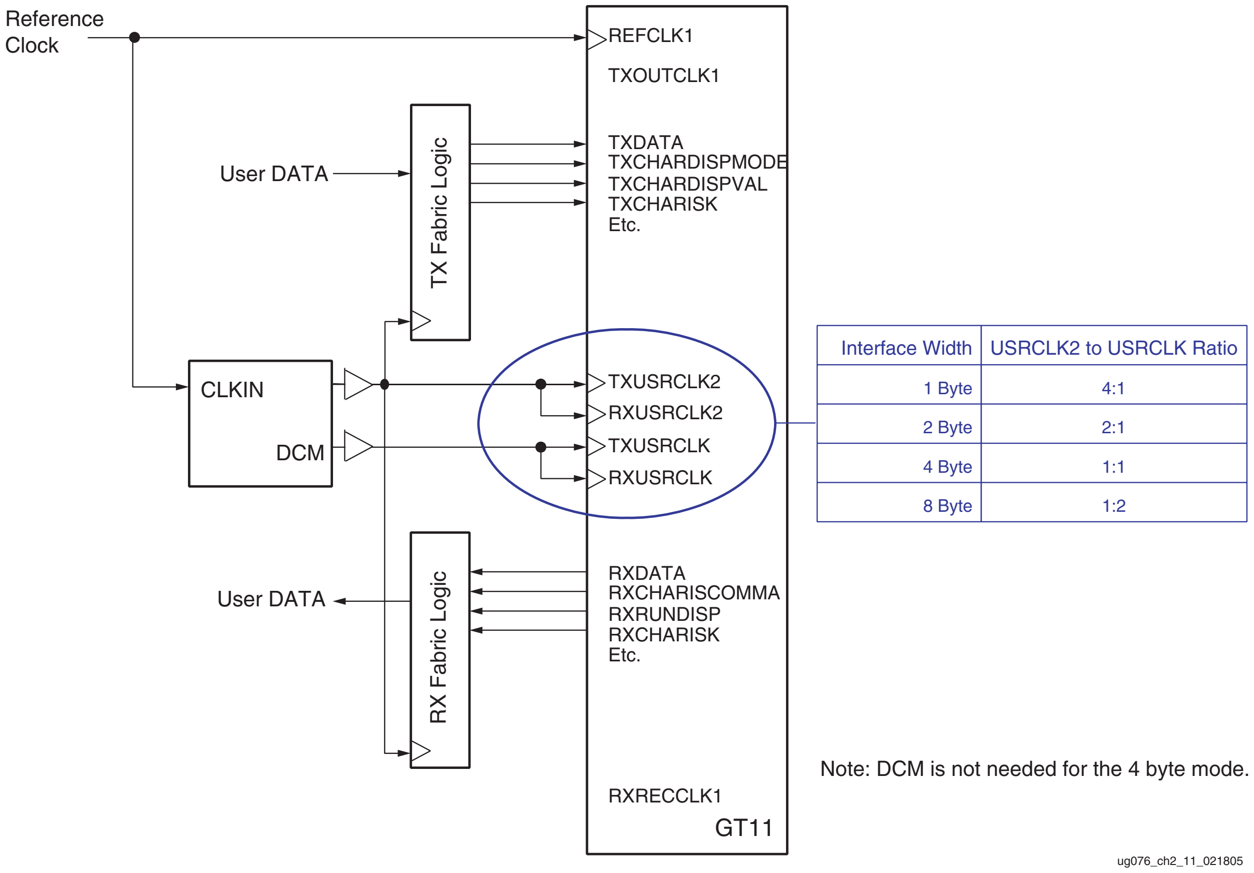
Figure 2-14 and Figure 2-15 show the common clocking use models that the MGT supports.

Note: TXOUTCLK/RXRECCLK connect to local and regional clocking. If connection is to a BUFG and global clocking resources, the TXPCSHCLKOUT/RXPCSHCLKOUT outputs of the MGT should be connected.



Notes: 1. BUFG connect is possible with TXOUTCLK1 or RXRECCLK1 with the use of fabric interconnect.

Figure 2-14: **Low Latency Clocking**



Interface Width	USRCLK2 to USRCLK Ratio
1 Byte	4:1
2 Byte	2:1
4 Byte	1:1
8 Byte	1:2

Note: DCM is not needed for the 4 byte mode.

Figure 2-15: **DCM Clocking**

Resets

The MGT has several different resets shown in Table 2-9. The resets affect different portions of the MGT. RXRESET and TXRESET reset the PCS portions of the transceiver. RXPMARESET and TXPMARESET reset the PMA portion of the transceiver. There are also two CRC logic resets (see Chapter 3, "PCS Digital Design Considerations").

Table 2-9: **MGT Reset Signals**

Reset	Description
RXCRCRESET	Resets the receiver CRC logic.
TXCRCRESET	Resets the transmitter CRC logic.
RXPMARESET	Resets the receiver PMA logic.
TXPMARESET	Resets the transmitter PMA logic.
TXRESET	Resets the transmitter PCS logic.
RXRESET	Resets the receiver PCS logic.

PCS Reset

It is possible to only reset the PCS, which brings every flip-flop in the PCS to a known value, but does not affect the PMA. When the signals TXRESET or RXRESET are set to a logic 1, the PCS is considered in reset. After the signal that caused the reset is asserted Low, the PCS takes five clocks for each clock domain to come out of reset.

The configuration bits are not affected during this reset, so the PMA speed, filter settings, and so on, all remain intact. Also, the PMA internal pipeline is not affected and continues to operate in normal fashion.

PMA Reset

Resetting the PMA and re-initializing the PMA function are also possible with the ports TXPMARESET and RXPMARESET. This resets the internal PMA dividers, but does not affect the PLLs or attribute values. It is recommended that these ports be set to a logic 1 upon startup for a minimum of two USRCLK cycles (based on fabric interface width).

Resetting the Transceiver

To reset the transceiver (Figure 2-16):

1. Assert the TX/RXPMARESET signal for at least three USRCLK cycles.
2. Wait for TX/RXLOCK to be asserted. In the absence of a serial data stream, the receiver will repeatedly lock and unlock.
3. Assert the TX/RXRESET for at least three USRCLK cycles.
4. Wait five USRCLK cycles after deasserting TX/RXRESET before sending packets.

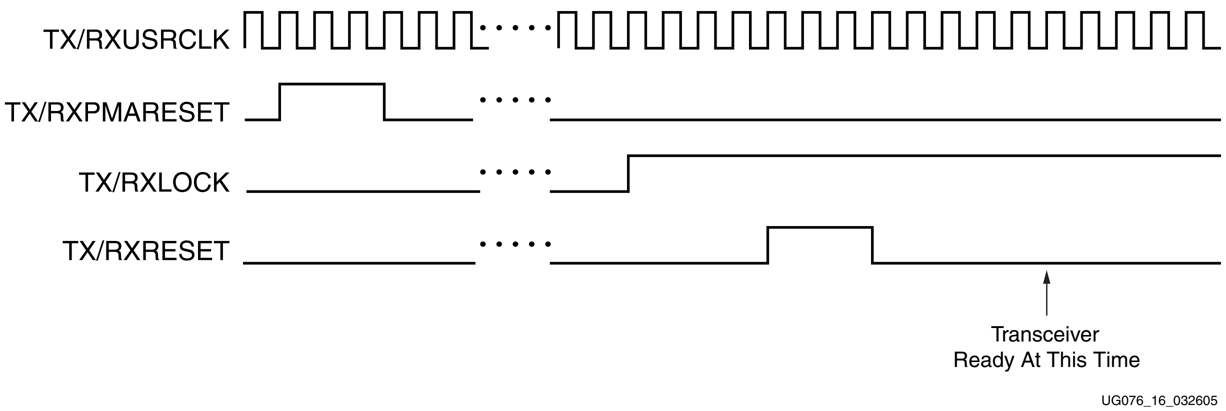


Figure 2-16: **Resetting the Transceiver**

Chapter 3

PCS Digital Design Considerations

The Virtex-4 RocketIO MGT PCS supports 8B/10B and 64B/66B encode/decode, SONET compatibility, and generic data modes. The MGT operates in two basic internal modes: 32 bit and 40 bit. The user has a large combination of protocols and data rates from which to choose in constructing the most advanced and easily configurable communication paths in the history of communication ICs. The MGT PCS continues to allow the user to change not only speeds of the PMA in real time, but also protocols within the PCS. Internal data width, external data width, and data routing can all be configured on a clock-by-clock basis.

With this advancement, users can initialize a communication channel at a low speed (for example, 2.5 Gb/s using 8B/10B (40-bit internal) and then auto-negotiate after the channel is stable to a 10.3125 Gb/s speed using 64B/66B (32-bit internal).

Note: The information in this chapter is provided to MGT users as a reference for understanding the individual attribute and control port settings within a primitive. Users have the choice of using the Architecture Wizard in ISE 7.1 (or greater) to generate the HDL code and ignoring this chapter, or using this chapter to better understand PCS configuration and/or to modify attribute and port values generated by the Architecture Wizard to create a user transceiver configuration.

Top-Level Architecture

Transmit Architecture

The transmit architecture for the PCS is shown in Figure 3-1. For information about bypassing particular blocks, consult the block function section for that particular block.

Note: The TX and RX CRC blocks can be run independently of the MGT. See Chapter 5, "Cyclic Redundancy Check (CRC)" for more information.

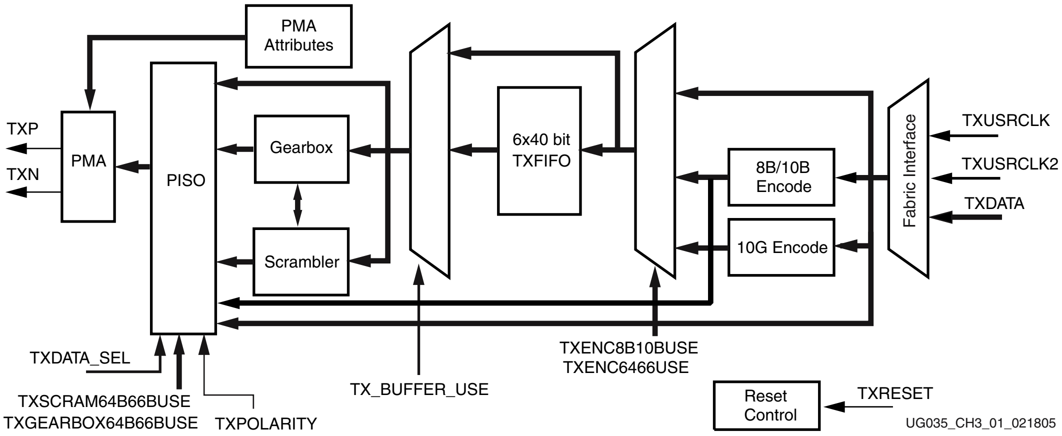


Figure 3-1: **Transmit Architecture**

Receive Architecture

The receive architecture for the PCS is shown in Figure 3-2. For information about bypassing particular blocks, consult the block function section for that particular block.

Note: The TX and RX CRC blocks can be run independently of the MGT. See Chapter 5, "Cyclic Redundancy Check (CRC)" for more information.

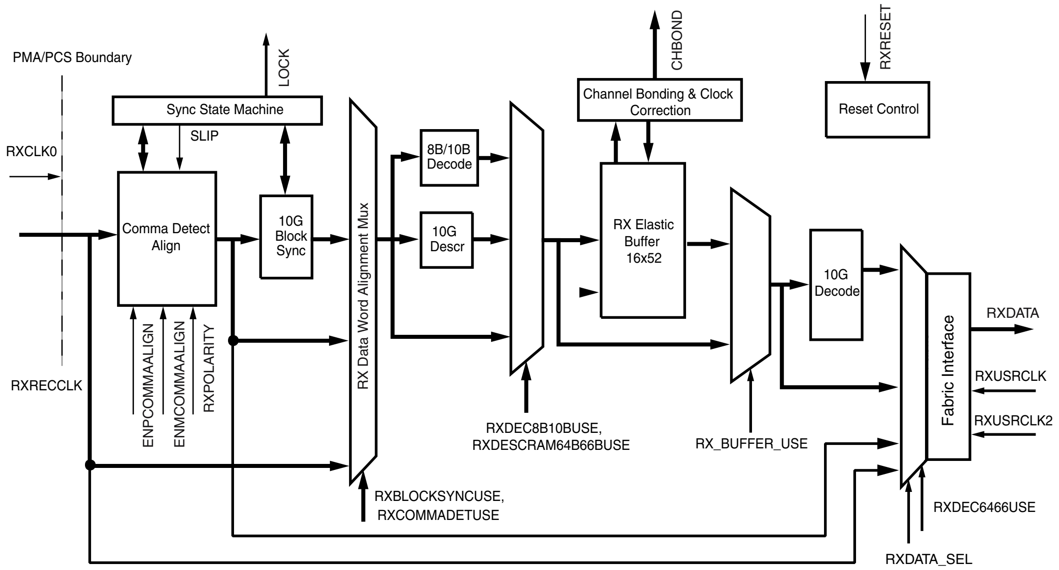


Figure 3-2: **Receive Architecture**

Block Level Functions

Bus Interface

External Bus Width Configuration (Fabric Interface)

By using the signals TXDATAWIDTH[1:0] and RXDATAWIDTH[1:0], the fabric interface can be determined. This also determines the USRCLK and USRCLK2 relationships. For slower serial speeds, 2-byte and 4-byte interfaces are preferred. For higher serial rates, especially 6.25 Gb/s and greater, 4-byte and 8-byte interfaces are recommended.

Table 3-1: **Selecting the External Configuration**

RXDATAWIDTH/TXDATAWIDTH	Data Width
`2'b00`	8/10 bit (1 byte)
`2'b01`	16/20 bit (2 byte)
`2'b10`	32/40 bit (4 byte)
`2'b11`	64/80 bit (8 byte)

Internal Bus Width Configuration

By using the signals TXINTDATAWIDTH[1:0] and RXINTDATAWIDTH[1:0], the internal bus width can be designated. This is usually determined by the encoding scheme implemented. For SONET and 64B/66B applications, the internal bus widths should be set to 32 bits. For 8B/10B and other encoding schemes that use alignment on 10-, 20-, or 40-bit boundaries, the 40-bit internal bus should be used.

Table 3-2 shows the available internal bus width setting.

Table 3-2: **Selecting the Internal Configuration**

RXINTDATAWIDTH/TXINTDATAWIDTH	Internal Data Width
`2'b10`	32 bit
`2'b11`	40 bit

Note:
The digital receiver must also have the same data bus width controlled with RXBY_32.

Encoding/Decoding

There are several supported encoding and decoding schemes. Among them are the 8B/10B and 64B/66B, plus A1, A2 detection logic built into the MGT. These blocks can be enabled or bypassed both statically and on a clock-by-clock basis. Bypassing these internal blocks and encoding in the fabric can support other encoding/decoding schemes. The following sections categorize the ports and attributes of the transceiver according to specific functionality, including 8B/10B encoding/decoding, 64B/66B encoding/decoding, SERDES alignment, clock correction, clock recovery, channel bonding, fabric interface, and other signals.

8B/10B

Note: In theVirtex-II Pro RocketIO transceiver, the most significant byte is sent first. In the Virtex-4 (and Virtex-II Pro X) RocketIO MGT, the least significant byte is sent first.

The 8B/10B encoding translates an 8-bit parallel data byte to be transmitted into a 10-bit serial data stream. This conversion and data alignment are shown in Figure 3-3. The serial port transmits the least significant bit of the 10-bit data, “a” first and proceeds to “j.” This allows data to be read and matched to the form shown in Appendix B, “8B/10B Valid Characters.”

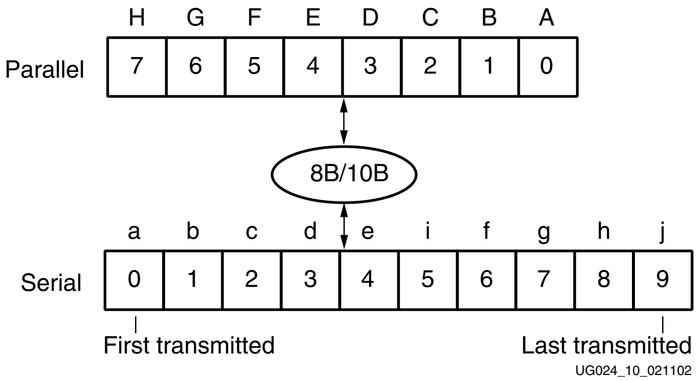


Figure 3-3: **8B/10B Parallel-to-Serial Conversion**

The serial data bit sequence is dependent on the width of the parallel data. The least significant byte is always sent first regardless of whether 1-, 2-, 4-, or 8-byte paths are used. The most significant byte is always last. Figure 3-4 shows a case when the serial data corresponds to each byte of the parallel data. TXDATA[7:0] is serialized and sent first, followed by TXDATA[15:8], TXDATA[23:16], and finally TXDATA[31:24]. The 2-byte path transmits TXDATA[7:0] and then TXDATA[15:8].

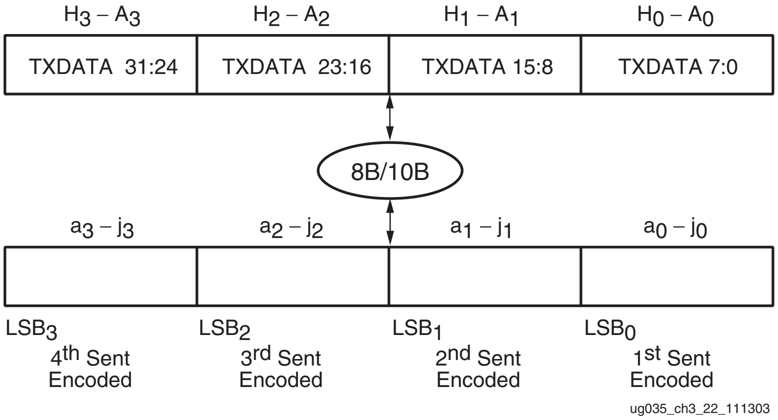


Figure 3-4: **4-Byte Serial Structure**

Encoder

A bypassable 8B/10B encoder is included in the transmitter. The encoder uses the same 256 data characters and 12 control characters (shown in Appendix B, "8B/10B Valid Characters") that are used for Gigabit Ethernet, XAUI, Fibre Channel, and InfiniBand.

The encoder accepts 8 bits of data along with a K-character signal for a total of 9 bits per character applied. If the K-character signal is set to a logic 1, the data is encoded into one of the 12 possible K-characters available in the 8B/10B code. If the K-character input is set to a logic 0, the 8 bits are encoded as standard data.

There are two ports that enable the 8B/10B encoding in the transceiver. The TXENC8B10BUSE controls whether the 8B/10B encoding block is used or not. When set to logic 1, the 8B/10B encoding block is used. When set to a logic 0, the 8B/10B encoding block is not used, allowing either the 64B/66B encoding block or complete encoding bypass. See Table 3-3. The TXBYPASS8B10B is a byte-mapped port that is 1, 2, 4 or 8 bits depending on the data width of the transceiver primitive being used. These bits correlate to each byte of the data path. This signal allows the data to bypass the 8B/10B encoding of the transmitter on a clock-by-clock basis. When set to a logic 1, the 8B/10B encoding is bypassed. In this mode, the extra bits are fed through the TXCHARDISPMODE and TXCHARDISPVAL buses. Otherwise, the normal 8-, 16-, 32-, or 64-bit fabric interfaces are used.

Table 3-3: **8B/10B Signal Definitions**

Signal		Function	
TXENC8B10BBUSE	1	8B/10B encoding is enabled.	
	0	8B/10B encoding is disabled. Encoding scheme depends on TXENC64B66BUSE.	
		8B/10B Encoding Block Enabled	**8B/10B Encoding Block Disabled**
TXBYPASS8B10B	1	8B/10B encoding is bypassed on a clock-by-clock basis.	Defined by 64B/66B functionality.
	0	8B/10B encoding block is enabled on a clock-by-clock basis.	
TXCHARDISPMODE, TXCHARDISPVAL	00	Maintain running disparity normally.	Part of 10-bit encoded byte (see Figure 3-5): TXCHARDISPMODE[0], (or: [1] /[2] /[3] /[4]/[5]/[6]/[7]) TXCHARDISPVAL[0], (or: [1] /[2] /[3] /[4]/[5]/[6]/[7]) TXDATA[7:0] (or: [15:8]/[23:16]/[31:24]/[39:32]/[47:40]/[55:48]/[63:56})
	01	Invert the normally generated running disparity before encoding this byte.	
	10	Set negative running disparity before encoding this byte.	
	11	Set positive running disparity before encoding this byte.	
TXCHARISK	1	Byte to transmit is a K-character.	Undefined. These bits should be set to a logic 0.
	0	Byte to transmit is a data character.	

Notes:

1. TXCHARDISPVAL and TXCHARDISPMODE become part of the 10-, 20-, 40-, or 80-bit data only if the internal data width is 40.

TXCHARDISPVAL and TXCHARDISPMODE

TXCHARDISPVAL and TXCHARDISPMODE are dual-purpose ports for the transmitter depending on whether 8B/10B encoding is done. Table 3-3 shows this dual functionality. When encoding is enabled, these ports function as byte-mapped control ports controlling the running disparity of the transmitted serial data (Table 3-4).

Table 3-4: **Running Disparity Control**

{TXCHARDISPMODE, TXCHARDISPVAL}	Function
00	Maintain running disparity normally.
01	Invert normally generated running disparity before encoding this byte.
10	Set negative running disparity before encoding this byte.
11	Set positive running disparity before encoding this byte.

In the encoding configuration, the disparity of the serial transmission can be controlled with the TXCHARDISPVAL and TXCHARDISPMODE ports. When TXCHARDISPMODE is set to a logic 1, the running disparity is set before encoding the specific byte. TXCHARDISPVAL determines if the disparity is negative (set to a logic 0) or positive (set to a logic 1).

When TXCHARDSIPMODE is set to a logic 0, the running disparity is maintained if TXCHARDISPVAL is also set to a logic 0. However, the disparity is inverted before encoding the byte when the TXCHARDISPVAL is set to a logic 1. Most applications use the mode where both TXCHARDISPMODE and TXCHARDISPVAL are set to logic 0. Some applications can use other settings if special running disparity configurations are required, such as in the "Non-Standard Running Disparity Example," page 74.

In the bypassed configuration, TXCHARDISPMODE[0] becomes bit 9 of the 10 bits of encoded data (TXCHARDISPMODE[1:7] are bits 19, 29, 39, 49, 59, 69, and 79 in the 20-bit, 40-bit, and 80-bit wide buses). TXCHARDISPVAL becomes bits 8, 18, 28, 38, 48, 58, 68, and 78 of the transmit data bus while the TXDATA bus completes the bus. See Table 3-3.

During transmit while 8B/10B encoding is enabled, the disparity of the serial transmission can be controlled with the TXCHARDISPVAL and TXCHARDISPMODE ports. When 8B/10B encoding is bypassed, these bits become Bits "h" and "j," respectively, of the 10-bit encoded data that the transceiver must transmit to the receiving terminal. Figure 3-5 illustrates the TX data map during 8B/10B bypass.

Note: When bypassing on a clock-by-clock basis, TXCHARDISPMODE is transmitted first, TXCHARDISPVAL is transmitted second, and TXDATA[0] is transmitted last.

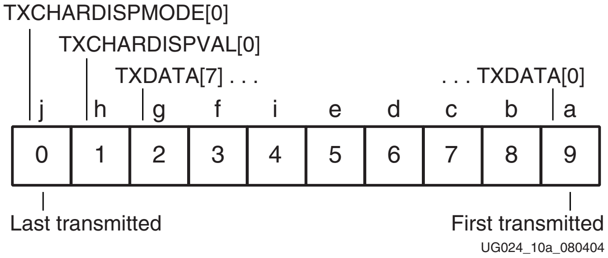


Figure 3-5: **10-Bit TX Data Map with 8B/10B Bypassed**

TXCHARISK

TXCHARISK is a byte-mapped control port that is only used when the 8B/10B encoder is implemented. This port indicates whether the byte of TXDATA is to be encoded as a control (K) character when set to a logic 1 and data character when set to a logic 0. When 8B/10B encoding is bypassed, this port is undefined. See Table 3-3.

TXRUNDISP

TXRUNDISP is a status port that is byte-mapped to the TXDATA. This port indicates the running disparity after this byte of TXDATA is encoded. When set to a logic 1, the disparity is positive. When set to a logic 0, the disparity is negative.

Decoder

An optional 8B/10B decoder is included in the receiver. A programmable option allows the decoder to be bypassed. When the 8B/10B decoder is bypassed, the 10-bit character order is shown in Figure 3-6 for a graphical representation of the received 10-bit character.

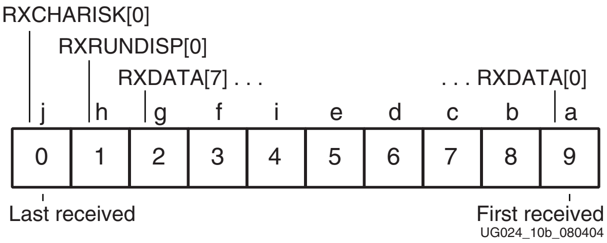


Figure 3-6: **10-Bit RX Data Map with 8B/10B Bypassed**

The decoder uses the same table (see Appendix B, "8B/10B Valid Characters") that is used for Gigabit Ethernet, Fibre Channel, and InfiniBand. In addition to decoding all data and K-characters, the decoder has several extra features. The decoder separately detects both "disparity errors" and "out-of-band" errors. A *disparity error* occurs when a 10-bit character is received that exists within the 8B/10B table but has an incorrect disparity. An *out-of-band error* occurs when a 10-bit character is received that does not exist within the 8B/10B table. It is possible to obtain an out-of-band error without having a disparity error. The proper disparity is always computed for both legal and illegal characters. The current running disparity is available at the RXRUNDISP signal. The 8B/10B decoder performs a unique operation if out-of-band data is detected. When out-of-band data is detected, the decoder signals the error and passes the illegal 10-bits through and places them on the outputs. This can be used for debugging purposes if desired.

The decoder also signals reception of one of the 12 valid K-characters. In addition, a programmable comma detect is included. The comma detect signal registers a comma on the receipt of any comma+, comma–, or both. Since the comma is defined as a 7-bit character, this includes several out-of-band characters. Another option allows the decoder to detect only the three defined commas (K28.1, K28.5, and K28.7) as comma+, comma–, or both. In total, six possible options exist: three for valid commas, three for "any comma."

Note that all bytes (1, 2, 4, or 8) at the RX FPGA interface each have their own individual 8B/10B indicators (K-character, disparity error, out-of-band error, current running disparity, and comma detect).

During receive and while 8B/10B decoding is enabled, the running disparity of the serial transmission can be read by the transceiver from the RXRUNDISP port, while the RXCHARISK port indicates presence of a K-character. When 8B/10B decoding is bypassed, these bits remain as Bits "b" and "a," respectively, of the 10-bit encoded data that the transceiver passes on to the user logic. Table 3-5 illustrates the RX data map during 8B/10B bypass.

Table 3-5: **8B/10B Bypassed Signal Significance**

Signal		Function	
RXCHARISK		Received byte is a K-character.	Part of 10-bit encoded byte (see Figure 3-6): RXCHARISK[0], (or: [1]/[2]/[3]/[4]/[5]/[6]/[7]) RXRUNDISP[0], (or: [1]/[2]/[3]/[4]/[5]/[6]/[7]) RXDATA[7:0] (or: [15:8]/[23:16]/[31:24]/[39:32]/[47:40]/[55:48]/[63:56})
RXRUNDISP	0	Indicates running disparity is NEGATIVE.	
	1	Indicates running disparity is POSITIVE.	
RXDISPERR		Disparity error occurred on current byte.	Unused.

RXCHARISK and RXRUNDISP

RXCHARISK and RXRUNDISP are dual-purpose ports for the receiver depending whether 8B/10B decoding is enabled. Table 3-5 shows this dual functionality. When decoding is enabled, these ports function as byte-mapped status ports of the received data.

In the encoding configuration, when RXCHARISK is asserted that byte of the received data is a control (K) character. Otherwise, the received byte of data is a data character. (See Appendix B, "8B/10B Valid Characters"). The RXRUNDISP port indicates that the disparity of the received byte is either negative or positive. RXRUNDISP is asserted to indicate positive disparity. This is used in cases like the "Non-Standard Running Disparity Example," page 74.

In the bypassed configuration, RXCHARISK and RXRUNDISP are additional data bits for the 10-, 20-, 40-, or 80-bit buses. This is similar to the transmit side. RXCHARISK[0:7] relates to bits 9, 19, 29, 39, 49, 59, 69, and 79 while RXRUNDISP pertains to bits 8, 18, 28, 38, 48, 58, 68, and 78 of the data bus. See Figure 3-12.

RXDISPERR

RXDISPERR is a status port for the receiver that is byte-mapped to the RXDATA. When a bit is asserted, a disparity error occurred on the received data. This usually indicates that the data is corrupt by bit errors, the transmission of an invalid control character, or cases when normal disparity is not required, such as shown in the "Non-Standard Running Disparity Example," page 74.

RXNOTINTABLE

RXNOTINTABLE is set to a logic 1 whenever the received data is not in the 8B/10B tables. The data received on bytes marked by RXNOTINTABLE is the raw data. This port is also byte-mapped to RXDATA and is only used when the 8B/10B decoder is enabled.

Non-Standard Running Disparity Example

To support other protocols, the transceiver can affect the disparity mode of the serial data transmitted. For example, Vitesse channel-to-channel alignment protocol sends out:

```
K28.5+ K28.5+ K28.5- K28.5-
```

or

```
K28.5- K28.5- K28.5+ K28.5+
```

Instead of:

```
K28.5+ K28.5- K28.5+ K28.5-
```

or

```
K28.5- K28.5+ K28.5- K28.5+
```

The logic must assert TXCHARDISPVAL to cause the serial data to send out two negative running disparity characters.

Transmitting Vitesse Channel Bonding Sequence

```
TXBYPASS8B10B
| TXCHARISK
| | TXCHARDISPMODE
| | | TXCHARDISPVAL
| | | | TXDATA
| | | | |
0 1 0 0 10111100    K28.5+ (or K28.5-)
0 1 0 1 10111100    K28.5+ (or K28.5-)
0 1 0 0 10111100    K28.5- (or K28.5+)
0 1 0 1 10111100    K28.5- (or K28.5+)
```

The MGT core receives this data but must have the CHAN_BOND_SEQ set with the `disp_err` bit set High for the cases when TXCHARDISPVAL is set High during data transmission.

Receiving Vitesse Channel Bonding Sequence

On the RX side, the definition of the channel bonding sequence uses the `disp_err` bit to specify the flipped disparity.

```
                       10-bit literal value
                       | disp_err
                       | | char_is_k
                       | | | 8-bit_byte_value
                       | | | |
CHAN_BOND_SEQ_1_1 = 0 0 1 10111100   matches K28.5+ (or K28.5-)
CHAN_BOND_SEQ_1_2 = 0 1 1 10111100   matches K28.5+ (or K28.5-)
CHAN_BOND_SEQ_1_3 = 0 0 1 10111100   matches K28.5- (or K28.5+)
CHAN_BOND_SEQ_1_4 = 0 1 1 10111100   matches K28.5- (or K28.5+)
CHAN_BOND_SEQ_LEN = 4
CHAN_BOND_SEQ_2_USE = FALSE
```

Comma Detection

Summary

Comma detection has been expanded to detect and align to 32-bit boundary to support A1, A2 sequences of SONET applications. The ability to detect symbols, and then align either to 1-, 2-, or 4-word boundaries is included. The RXSLIDE input allows the user to *slide* or *slip* the alignment by one bit in each 32- and 40-bit mode at any time when any applications cannot be supported with the 32-bit or 10-bit comma definitions.

The following ports and attributes affect the function of the comma detection block:

- RXCOMMADETUSE
- ENMCOMMAALIGN
- ENPCOMMAALIGN
- ALIGN_COMMA_WORD[1:0]
- MCOMMA_32B_VALUE[31:0]
- DEC_MCOMMA_DETECT
- PCOMMA_32B_VALUE[31:0]
- DEC_PCOMMA_DETECT
- COMMA_10B_MASK[10:0]

- RXSLIDE
- RXINTDATAWIDTH[1:0]
- COMMA32

Bypass

By deasserting RXCOMMADETUSE Low, symbol/comma detection is not enabled. If RXCOMMADETUSE is set to logic 1, symbol/comma detection takes place. See Table 3-6.

Table 3-6: **Deserializer Comma Detection Bypass**

Signal		Function
RXCOMMADETUSE	0	Comma detection is disabled.
	1	Comma detection is enabled. COMMA is detected based on Table 3-7.
ENPCOMMAALIGN ENMCOMMAALIGN	00	Comma alignment is disabled. RXDATA reflects how the data is output from the deserializer.
	01	Comma alignment is enabled. The comma is defined by COMMA_10B_MASK and MCOMMA _32B_VALUE.
	10	Comma alignment is enabled. The comma is defined by COMMA_10B_MASK and PCOMMA _32B_VALUE.
	11	Comma alignment is enabled. The comma is defined by COMMA_10B_MASK and MCOMMA _32B_VALUE, PCOMMA_32B_VALUE.

SYMBOL Alignment and Detection

The MCOMMA_32B_VALUE, PCOMMA_32B_VALUE, COMMA_10B_MASK and COMMA32 attributes are used by the comma detection block values to which the data should be aligned. Once set to a value, the comma detection block searches the data stream for these values and aligns the pipeline to the position where the value was detected in the data stream. Virtex-4 users should note that the 10-bit values (used in 8B/10B encoding scheme) are reversed relative to Virtex-II Pro devices, but are similar to Virtex-II Pro X devices. For other generation differences, see Appendix D, "Virtex-II Pro/Virtex-II Pro X to Virtex-4 RocketIO Transceiver Design Migration." The reason for this is that while Virtex-II Pro devices support mainly 8B/10B applications, Virtex-4 devices can support many applications including SONET and use a more general approach. See "SONET Alignment."

8B/10B Alignment

For 8-or 10-bit alignment (as in 8B10B encoding), the attribute COMMA32 must be set to FALSE.

Figure 3-7 shows a Virtex-4 8B/10B comma detection example relative to the data stream received at the PCS/PMA interface on the receive side. By using the signals MCOMMA_32B_VALUE (lower 10 bits), DEC_MCOMMA_DETECT, PCOMMA_32B_VALUE (lower 10 bits), DEC_PCOMMA_DETECT, and

COMMA_10B_MASK, any 10 bits of a 8-, 16-, 32-, 10-, 20-, or 40-bit symbol detection can take place for two different symbol values.

The DEC_MCOMMA_DETECT and DEC_PCOMMA_DETECT indicate which symbol should be compared to the incoming data for alignment to be indicated by RXCHARISCOMMA. See Table 3-7.

DEC_VALID_COMMA_ONLY determines if only valid 8B/10B K-characters should be indicated on RXCHARISCOMMA.

Note: MCOMMA_32B_VALUE bits 31-10 and PCOMMA_32B_VALUE bits 31-10 are undefined in this mode of operation.

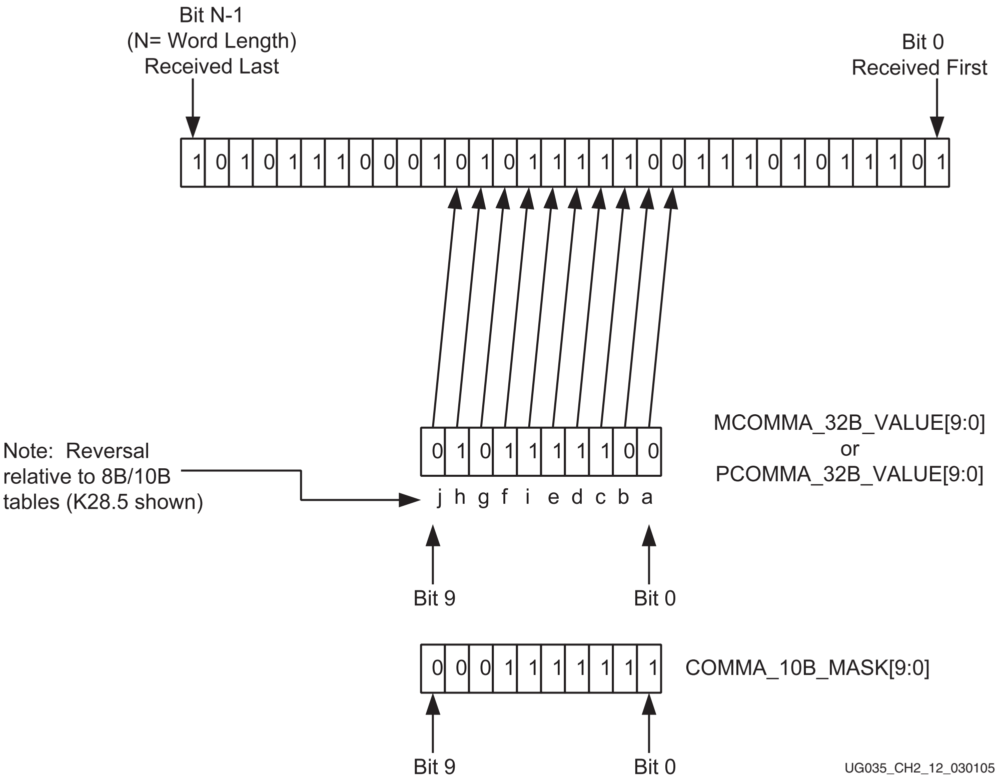


Figure 3-7: **8B/10B Comma Detection Example**

Table 3-7: **8B/10B Decoder Comma Detection**

DEC_MCOMMA_DETECT	DEC_PCOMMA_DETECT	Function
FALSE	FALSE	No symbol detection takes place.
FALSE	TRUE	RXCOMMADET is asserted if the incoming data is compared and aligned to the symbol defined by PCOMMA_32B_VALUE.
TRUE	FALSE	RXCOMMADET is asserted if the incoming data is compared and aligned to the symbol defined by MCOMMA_32B_VALUE.

Table 3-7: **8B/10B Decoder Comma Detection *(Continued)***

DEC_MCOMMA_DETECT	DEC_PCOMMA_DETECT	Function
TRUE	TRUE	RXCOMMADET is asserted if the incoming data is compared and aligned to the symbol defined by PCOMMA_32B_VALUE or MCOMMA_32B_VALUE.
DEC_VALID_COMMA_ONLY	FALSE	Raise RXCHARISCOMMA whether or not valid characters are in the 8B10B translation.
	TRUE	Raise RXCHARISCOMMA only when valid characters are in the 8B10B translation.

SONET Alignment

Virtex-4 MGTs support bit and byte alignment for SONET A1A2 transition. For this type of alignment, the attribute COMMA32 alignment must be set to TRUE. Also MCOMMA_32B_VALUE and PCOMMA_32B_VALUE bits 31-0 must be defined. Table 3-8 shows how to set up these attributes to align to the A1A2 transition. Notice that the MCOMMA sets up the bit alignment and the PCOMMA is the byte alignment, and that there is not a true 32-bit compare done in the alignment logic.

Table 3-8: **SONET Bit/Byte Alignment Attribute Settings**

Attribute	Setting	Definition
ALIGN_COMMA_WORD	`1`	Sets alignment to 1 byte mode.
COMMA32	TRUE	Enables byte alignment logic.
COMMA_10B_MASK	`0011111111`	Enables bit alignment of 8 bits.
MCOMMA_32B_VALUE	`1111 1111 1111 1111 1111 1111 1111 0110`	Sets the bit alignment to align to an A1 sequence (`0xF6`). Note that only the last 8 bits are matched based on the COMMA_10B_MASK.
MCOMMA_DETECT	TRUE	Enables the MCOMMA detections or A1 detection.
PCOMMA_32B_VALUE(1)	`1111 0110 1111 1011 0001 0100 0001 0100 00`	Defines the A1A1A2A2 (`0xF6F62828`) transition for the byte alignment.
PCOMMA_DETECT	TRUE	Enables the PCOMMA detection or A1A1A2A2 detection.
ENMCOMMAALIGN	`1`	Enables the bit alignment (A1).
ENPCOMMAALIGN	`1`	Enable the byte alignment (A1A1A2A2).

Notes:

1. This alignment is only a 10-bit value. The byte alignment is addition logic which checks each byte to determine where the A1A2 transition occurs and adjusts it accordingly.

Byte Alignment

After the positive or the negative symbol is detected, the data is aligned to that symbol. By using the signals ENMCOMMAALIGN, ENPCOMMAALIGN (see Table 3-6), ALIGN_COMMA_WORD, and RXSLIDE, alignment can be completely controlled for all data pipeline configurations.

ALIGN_COMMA_WORD

The attribute ALIGN_COMMA_WORD controls when realignment takes place and when the difference between symbols is on a byte-by-byte basis. If the current position of the symbol detected is some fraction of a byte different than the previous symbol position, alignment takes place regardless of the setting of ALIGN_COMMA_WORD.

There are three options for ALIGN_COMMA_WORD: 1 byte, 2 byte, and 4 byte. When ALIGN_COMMA_WORD is set to a 1, the detection circuit allows detection symbols in contiguous bytes. When ALIGN_COMMA_WORD is set to a 2, the detection circuit allows detection symbols every other byte. When ALIGN_COMMA_WORD is set to a 4, the detection circuit allows detection symbols every fourth byte. See Table 3-9.

Table 3-9: **ALIGN_COMMA_WORD Functionality**

ALIGN_COMMA_WORD	Function
1	Byte alignment (aligns comma on any byte boundary).
2	2-byte alignment (aligns on every other byte).
4	4-byte alignment (aligns on every fourth byte).

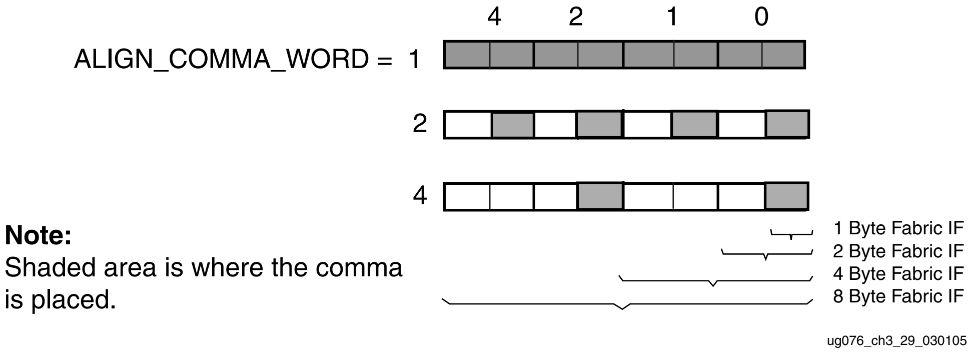


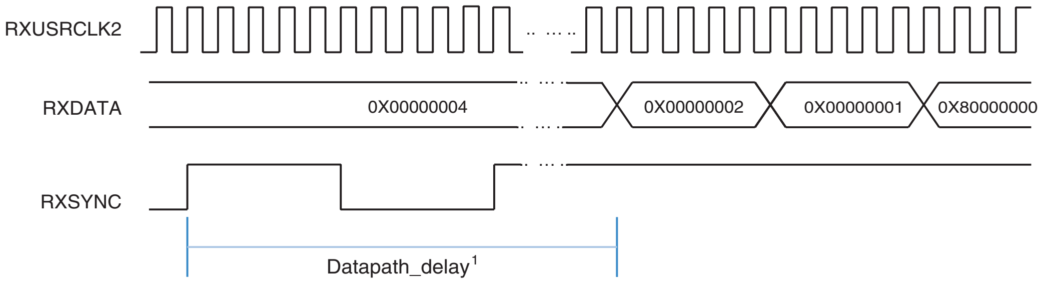
Figure 3-8: **Comma Placement**

RXSLIDE and RXSYNC

Both of these signals have similar functionality but are used for different applications.

RXSLIDE allows manual control of the PCS barrel shifter for applications that require an alignment function that cannot be defined with the P_COMMA_32_VALUE/M_COMMA_32_VALUE. When RXSLIDE is set to a logic 1, the position of the barrel shifter can be adjusted by one bit. This data increment can continue until it reaches the most significant bit, equal to the maximum word length minus 1 where it rolls over. When RXSLIDE is set to a logic 1, it must be set to a logic 0 for two clock periods before it can be set to a logic 1 again. RXSLIDE is the recommended manual bit slip method. RXSYNC should only be used for applications that cannot tolerate the latency uncertainty imposed by the barrel shifter.

RXSYNC allows adjustment of the received data 1 UI intervals by dropping data in the PMA. This function should only be used by applications that cannot tolerate the data uncertainty of the PCS barrel shifter in the comma alignment block as the use model. RXSYNC alignment in conjunction with low latency synchronous clock mode must take precautions not to violate the PCS/PMA timing interface. This functionality can be used for standards, such as SFI4.2, where multiple channels have a maximum skew value, but no channel bonding functionality is built into the protocol. To enable this functionality, PMA_BIT_SLIP and PCS_BIT_SLIP attributes need to be set to TRUE. Every rising or falling edge of the RXSYNC port *slips* the data by one serial bit. This functionality is shown in Figure 3-9. With the use of FPGA fabric logic, multiple channels can be aligned within ±1 serial UI.



Notes:

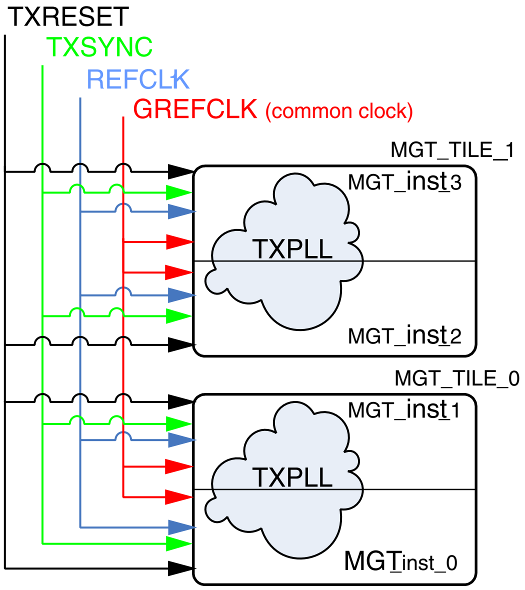
1. Datapath_delay depends on what receiver functions are used.
2. The receive data is continuously 0X00000004The data changes are caused by the rising and falling edge of RXSYNC.
3. PMA_BIT_SLIP and PCS_BIT_SLIP are set to TRUE.

ug076_ch3_9a_030105

Figure 3-9: **RXSYNC Timing Waveform**

TXSYNC

TXSYNC adjusts the possible TX skew upon multiple transceivers. This adjustment is recommended for standards, such as SFI4.2, where there is a maximum channel skew and there is no channel bonding functionality built into the protocol. TXSYNC uses the GREFCLK as a common clock among the common transceivers. Figure 3-10 and Figure 3-11 show how to implement. Since TXSYNC synchronizes the phases between the multiple PLLs, the TX skew uncertainty can be as small as ±1 UI at slower data rates. At faster data rates (10 Gb/s), it may be ±5 UI. Finally the PCS reset, TXRESET, needs to be asserted to reset all of the TX FIFO pointers.



ug076_ch3_10a_013105

Figure 3-10: **TXSYNC Common Clock**

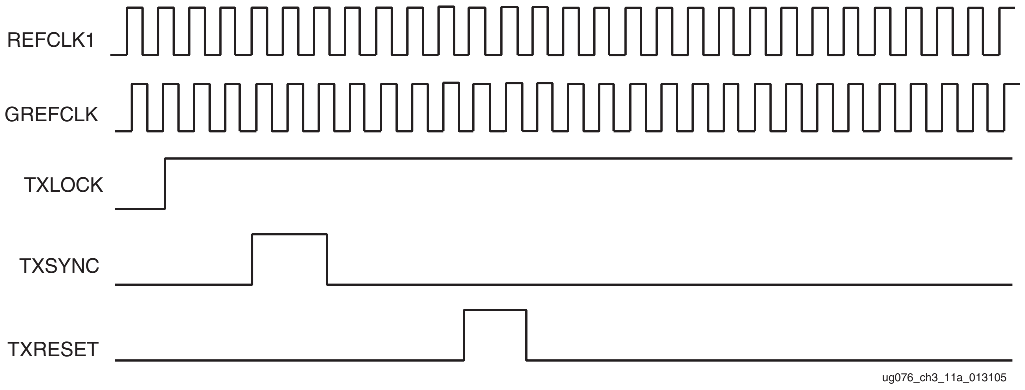


Figure 3-11: **TXSYNC Timing Waveforms**

64B/66B Encoder

Bypassing

There are two types of bypassing regarding the 64B/66B encoder. The encoder block can either be entirely bypassed, or the 64B/66B encoder can be used and bypassed on a clock-by-clock basis.

STATIC BYPASS

If TXENC64B66BUSE is set to a logic 0, the entire 64B/66B encoder is not used. If encoding is done in the fabric, the sync header [0:1] must be placed at TXCHARDISPVAL[0] and TXCHARDISPMODE[0]. In the following configurations:

- {TXCHARDISPVAL[0], TXCHARDISPMODE[0]} = 10 = Control Block
- {TXCHARDISPVAL[0], TXCHARDISPMODE[0]} = 01 = Data Block
- {TXCHARDISPVAL[0], TXCHARDISPMODE[0]} = 00 = other 32 bits of encoded 64 bits when fabric data bus = 32. This is shown in Figure 3-12.

DYNAMIC BYPASS

If TXENC64B66BUSE is set to a logic 1, the TXBYPASS8B10B bit 0 signal bypasses the 64B/66B encoder on a block basis, which means that two clock cycles are needed to do a full bypass of a block. The Sync Header is taken from the TXCHARDISPMODE[0:1]. To bypass on a block basis, the even boundary needs to be indicated at the fabric interface, which is contained in TXKERR bit 0 set to a logic 1. Figure 3-13 shows this operation. Note that TXBYPASS8B10B bit 0 must remain a logic 1 for the entire 64 bits of bypassed data (two clock cycles when the fabric data width is 32). The TXCHARISK signal performs the function of TXC. The functionality is shown in Table 3-10.

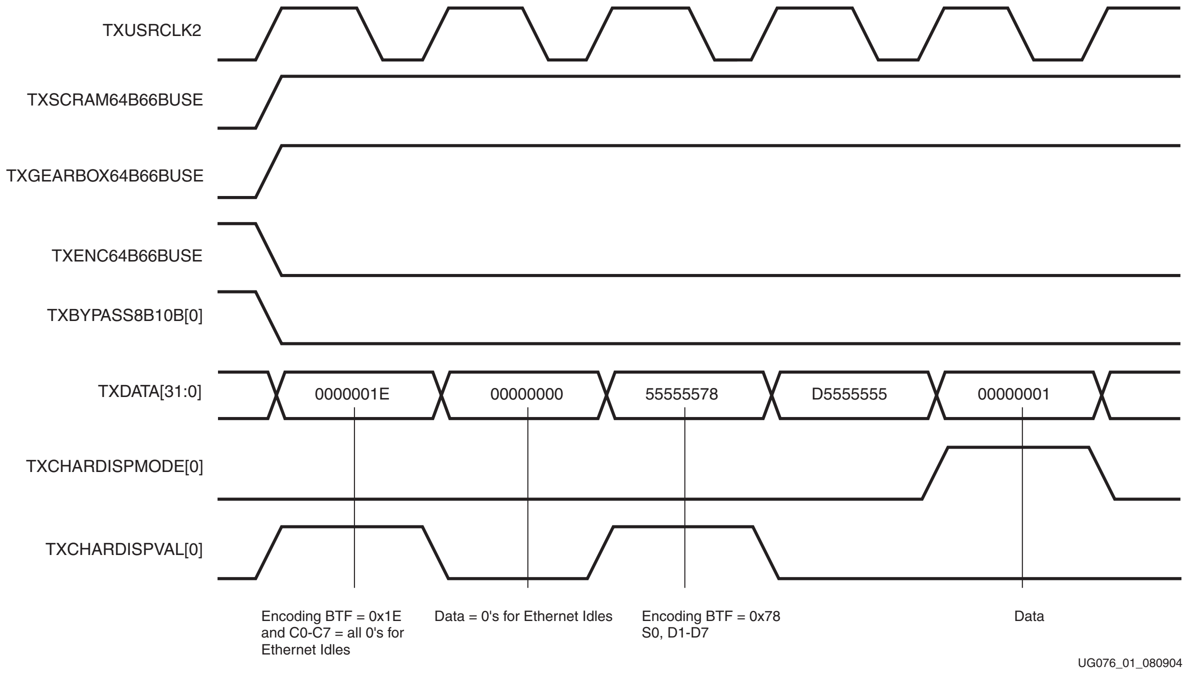


Figure 3-12: **64B/66B Encoder Static Bypass Waveform**

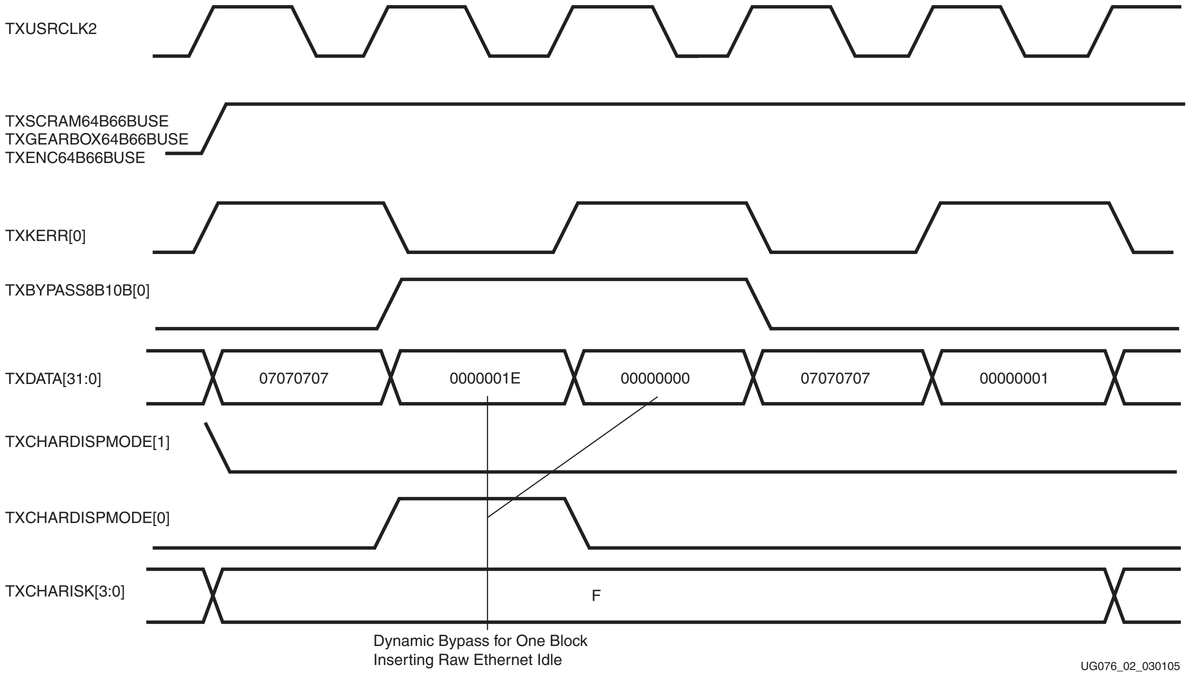


Figure 3-13: **64B/66B Encoder Dynamic Bypass Waveform**

Table 3-10: **64B/66B Bypassing**

Signal	Function	
TXENC64B66BUSE	**1 = Bypass 64B/66B Encoder on a Clock-to-Clock Basis:**	**0 = Bypass 64B/66B Encoder Entirely:**
TXBYPASS8B10B[0]	0 = no bypass. 1 = bypass this block	Defined by Table 3-3
TXCHARDISPMODE[0:1]	Sync Header shown in Figure 3-16 (same as SH[0:1])	
TXKERR[0]	Indicates even boundary for bypassing on block basis	
TXCHARISK[3:0]	Performs function of TXC	Indicates character is a (K) control character

The transmit 64B/66B encoder borrows four bits of the TXCHARISK bus (bits [3:0]) to convey the control signaling to the 64B/66B encoder (see Table 3-11). The eight TXC bits track with the eight bytes of TXDATA_IN (TXC[0] with TXDATA_IN[7:0], and so on) to signal data block formatting. The transmit fabric interface logic (which first monitors transmit data as it travels from the fabric interface to the PMA) drives the encoder with the four TXC bits as follows:

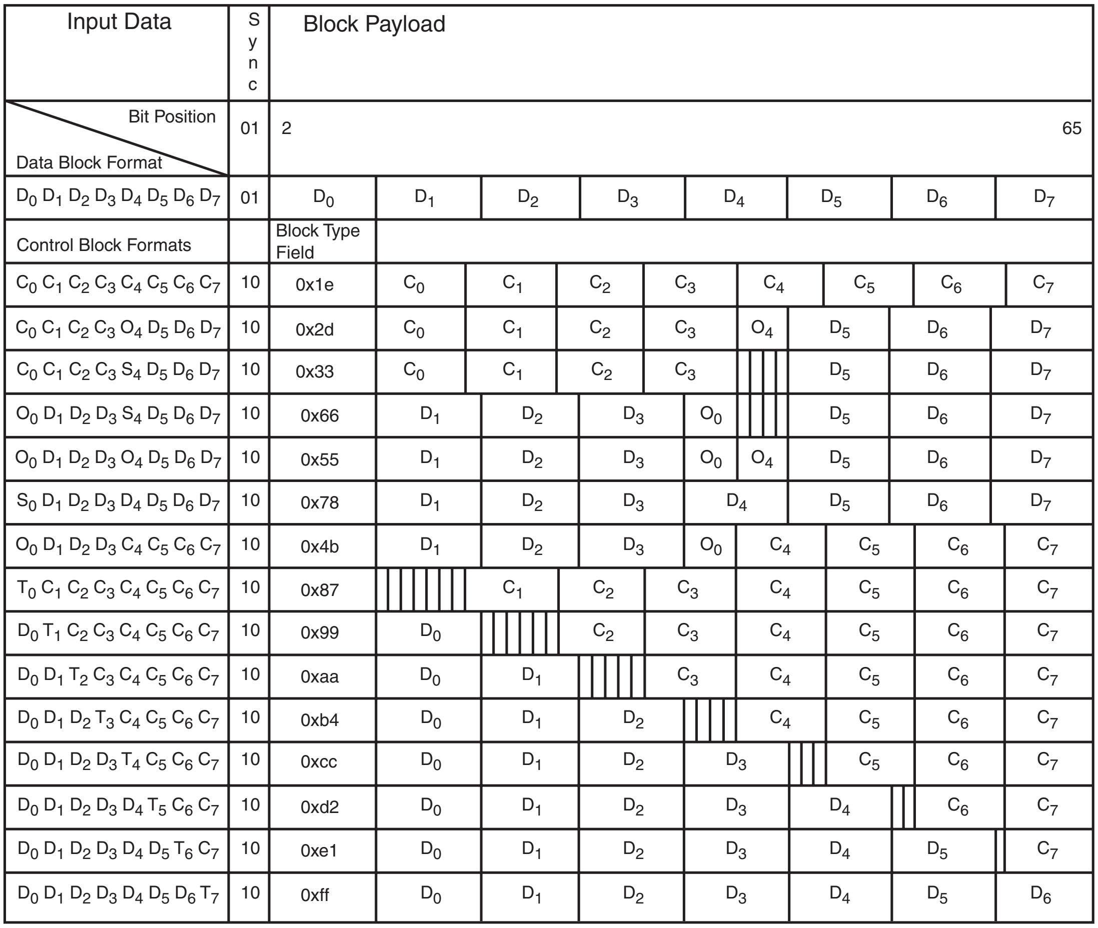
Table 3-11: **Transmit 64B/66B Encoder Control Mapping**

TXC[3:0] (TXCHARISK[3:0])	Block Formatting
1111	Idles OR terminate-with-idles
0001	Start-of-frame OR ordered-set
1110	Terminate in second position
1100	Terminate in third position
1000	Terminate in fourth position
0000	Data OR error (no K-chars)

Each "1" in the TXC span represents a control-character-match, that is, recognition that the associated byte is a special control character of some type (idle, start, terminate, or ordered set).

Normal Operation

The 64B/66B encoder implements the Encoding Block Format function shown in Figure 3-14.



Input Data	Sync	Block Payload								
Bit Position / Data Block Format	01	2								65
$D_0\ D_1\ D_2\ D_3\ D_4\ D_5\ D_6\ D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	
Control Block Formats		Block Type Field								
$C_0\ C_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0x1e	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$C_0\ C_1\ C_2\ C_3\ O_4\ D_5\ D_6\ D_7$	10	0x2d	C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
$C_0\ C_1\ C_2\ C_3\ S_4\ D_5\ D_6\ D_7$	10	0x33	C_0	C_1	C_2	C_3		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3\ S_4\ D_5\ D_6\ D_7$	10	0x66	D_1	D_2	D_3	O_0		D_5	D_6	D_7
$O_0\ D_1\ D_2\ D_3\ O_4\ D_5\ D_6\ D_7$	10	0x55	D_1	D_2	D_3	O_0	O_4	D_5	D_6	D_7
$S_0\ D_1\ D_2\ D_3\ D_4\ D_5\ D_6\ D_7$	10	0x78	D_1	D_2	D_3	D_4	D_5	D_6	D_7	
$O_0\ D_1\ D_2\ D_3\ C_4\ C_5\ C_6\ C_7$	10	0x4b	D_1	D_2	D_3	O_0	C_4	C_5	C_6	C_7
$T_0\ C_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0\ T_1\ C_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0\ D_1\ T_2\ C_3\ C_4\ C_5\ C_6\ C_7$	10	0xaa	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0\ D_1\ D_2\ T_3\ C_4\ C_5\ C_6\ C_7$	10	0xb4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0\ D_1\ D_2\ D_3\ T_4\ C_5\ C_6\ C_7$	10	0xcc	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0\ D_1\ D_2\ D_3\ D_4\ T_5\ C_6\ C_7$	10	0xd2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0\ D_1\ D_2\ D_3\ D_4\ D_5\ T_6\ C_7$	10	0xe1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0\ D_1\ D_2\ D_3\ D_4\ D_5\ D_6\ T_7$	10	0xff	D_0	D_1	D_2	D_3	D_4	D_5	D_6	

UG035_ch3_23_091103

Figure 3-14: **Block Format Function**

The control codes are specified as follows in Table 3-12:

Table 3-12: **Control Codes**

Control Character	Notation	XGMII Control Code	10GBASE-R Control Code	10GBASE-R 0 Code	8B/10B Code
idle	/I/	0x07	0x00		K28.0 or K28.3 or K28.5
start	/S/	0xFB	encoded by block type field		K27.7
terminate	/T/	0xFD	encoded by block type field		K29.7
error	/E/	0xFE	0x1E		K30.7
Sequence ordered_set	/Q/	0x9C	encoded by block type field plus O mode	0x0	K28.4
reserved0[1]	/R/	0x1C	0x2D		K28.0
reserved1[1]		0x3C	0x33		K28.1
reserved2[1]	/N/	0x7C	0x4B		K28.3
reserved3[1]	/K/	0xBC	0x55		K28.5
reserved4[1]		0xDC	0x66		K28.6
reserved5[1]		0xF7	0x78		K23.7
Signal ordered_set	/Fsig/	0x5C	encoded by block type field plus O mode	0xF	K28.2

Notes:

1. Reserved if for Ethernet. The MGT supports these control characters.

64B/66B Scrambler

Bypassing

If the signal TXSCRAM64B66BUSE is set to a logic 0, the scrambler is not used.

Normal Operation

If the signal TXSCRAM64B66BUSE is set to a logic 1, the scrambler is enabled for use. The scrambler uses the polynomial:

```
G(x) = 1 + x39 + x58
```

to scramble 64B/66B payload data. The scrambler works in conjunction with the gearbox to scramble and format data correctly.

See Table 3-13 for TXSCRAM64B66BUSE and TXGEARBOX64B66BUSE configurations.

Table 3-13: **TXSCRAM64B66BUSE and TXGEARBOX64B66BUSE Configurations**

Signal	Function	
TXSCRAM64B66BUSE, TXGEARBOX64B66BUSE	`00`	Scrambler and gearbox are disabled. 64B/66B encoding is not used in this configuration.
	`11`	Scrambler and gearbox are enabled to implement the 64B/66B encoding.
	`01, 10`	Invalid configurations. TXSCRAM64B66BUSE must equal TXGEARBOX64B66BUSE.

64B/66B Gearbox

Normal Operation

If the signal TXGEARBOX64B66BUSE is set to logic 1, the gearbox is used. The gearbox should always be enabled when using the 64B/66B protocol, because the gearbox frames 64B/66B data for the PMA. The gearbox should also be equal to TXSCRAM64B66BUSE. See Table 3-13.

64B/66B Decoder

Bypassing

If RXDEC64B66BUSE is set to logic 0, the entire 64B/66B decoder is not used. In this mode, RXLOSSOFSYNC[1] = 0 indicates that block synchronization has been established. The sync header bits are overloaded to RXRUNDISP[0] and RXCHARISK[0] in the following configurations:

- {RXRUNDISP[0], RXCHARISK[0]} = `10` = Control Block
- {RXRUNDISP[0], RXCHARISK[0]} = `01` = Data Block
- {RXRUNDISP[0], RXCHARISK[0]} = `00` = Second 32 bits of 64-bit encoded bits (when fabric data width = 32).

Figure 3-15 shows the waveform for this situation.

Note: The signal RXIGNOREBTF can be set to logic 1 for applications that need to use the integrated decoder in conjunction with additional decoding logic in the fabric. This will cause /E/ block types to not be generated.

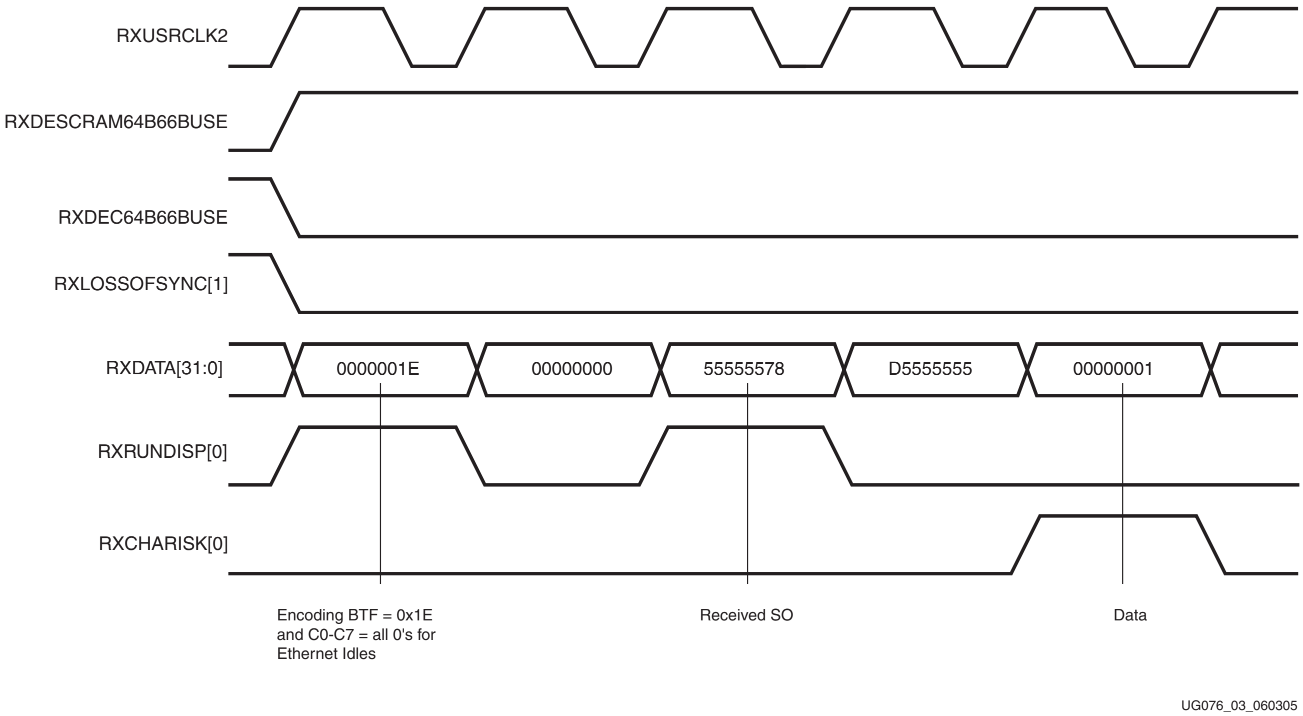


Figure 3-15: **64B/66B Decoder Bypass Waveform**

Normal Operation

If RXDEC64B66BUSE is set to logic 1, the 64B/66B decoder decodes according to the 64B/66B block format table shown in Figure 3-14.

If the signal RXIGNOREBTF is set to logic 1, block type fields not recognized are passed on, whereas if the signal is set to logic 0, the error block /E/ is passed on. RXCHARISK is equivalent to RXC when the decoder is enabled. See Table 3-14.

Table 3-14: **RXDEC64B66BUSE, RXIGNOREBTF, and RXCHARISK Configurations**

RXDEC64B66BUSE	0 indicates the decoder is not used.	
	1 indicates the decoder is used.	
RXIGNOREBTF	Function 64B/66B Decoder Used	Function 64B/66B Decoder Bypassed
	0 unrecognized field types cause /E/ passed on.	Undefined.
	1 unrecognized field types passed on.	
RXCHARISK	Equivalent to RXC.	Defined by 8B/10B decoder used.
RXLOSSOFSYNC[1]	1 indicates block synchronization.	

64B/66B Descrambler

Bypassing

If the signal RXDESCRAM64B66BUSE is set to logic 0, the descrambler is not used.

Normal Operation

If the signal RXDESCRAM64B66BUSE is set to logic 1, the descrambler is enabled for use. The descrambler uses the polynomial:

```
G(x) = 1 + x^39 + x^58
```

64B/66B Block Sync

Normal Operation

This block sync design works hand-in-hand with the commaDet block. The commaDet takes as input 32 bits of scrambled and unaligned data from the PMA. It then sends to the block sync the 2-bit sync header, or what it thinks is the sync header based on the current tag value. It asserts `test_sh` which tells the block sync to test the value of the sync header. The block sync analyzes the sync header and if it is valid, increments the `sh_cnt` counter. If the sync header is not a legal value, `sh_cnt` is incremented as well as the counter `sh_invalid_cnt`, and then `bit_slip` is asserted for one clock. The bit slip signal feeds back to the commaDet block and tells it to shift the barrel shifter by one bit. This process of slipping and testing the sync header repeats until block lock is achieved. See Figure 3-16.

The state machine works by keeping track of valid and invalid sync headers. Upon reset, block lock is deasserted, and the state is `LOCK_INIT`. The next state is `RESET_CNT` where all counters are zeroed out. When `test_sh` is asserted, the next state is `TEST_SH`, which checks the validity of the sync header. If it is valid, the next state is `VALID_SH`, if not, the state changes to `INVALID_SH`.

From `VALID_SH`, if `sh_cnt` is less than the attribute value `sh_cnt_max` and `test_sh` is High, the next state is `TEST_SH`. If `sh_cnt` is equal to `sh_cnt_max` and `sh_invalid_cnt` equals 0, the next state is `GOOD_64` and from there `block_lock` is asserted. Then the process repeats again and the counters are zeroed.

If at `TEST_SH` `sh_cnt` equals `sh_cnt_max,` but `sh_invalid_cnt` is greater than zero, then the next state is `RESET_CNT`. From `INVALID_SH`, if `sh_invalid_cnt` equals `sh_invalid_cnt_max`, or if `block_lock` is not asserted, the next state is `SLIP`, where `bit_slip` is asserted, and then on to `RESET_CNT`. If `sh_cnt` equals `sh_cnt_max` and `sh_invalid_cnt` is less than `sh_invalid_cnt_max` and `block_lock` is asserted, then goes back to `RESET_CNT` without changing `block_lock` or `bit_slip`.

Finally, if `test_sh` is High and `sh_cnt` is less than `sh_cnt_max,` and `sh_invalid_cnt` is less than `sh_invalid_cnt_max` and `block_lock` is asserted, goes back to the `TEST_SH` state. **The main thing to note with this state machine is that to achieve block lock, it must receive `sh_cnt_max` number of valid sync headers in a row without getting an invalid sync header.** However, once block lock is achieved, `sh_invalid_cnt_max -1` number of invalid sync headers can be received within `sh_cnt_max` number of valid sync headers. Thus, once locked, it is harder to break lock.

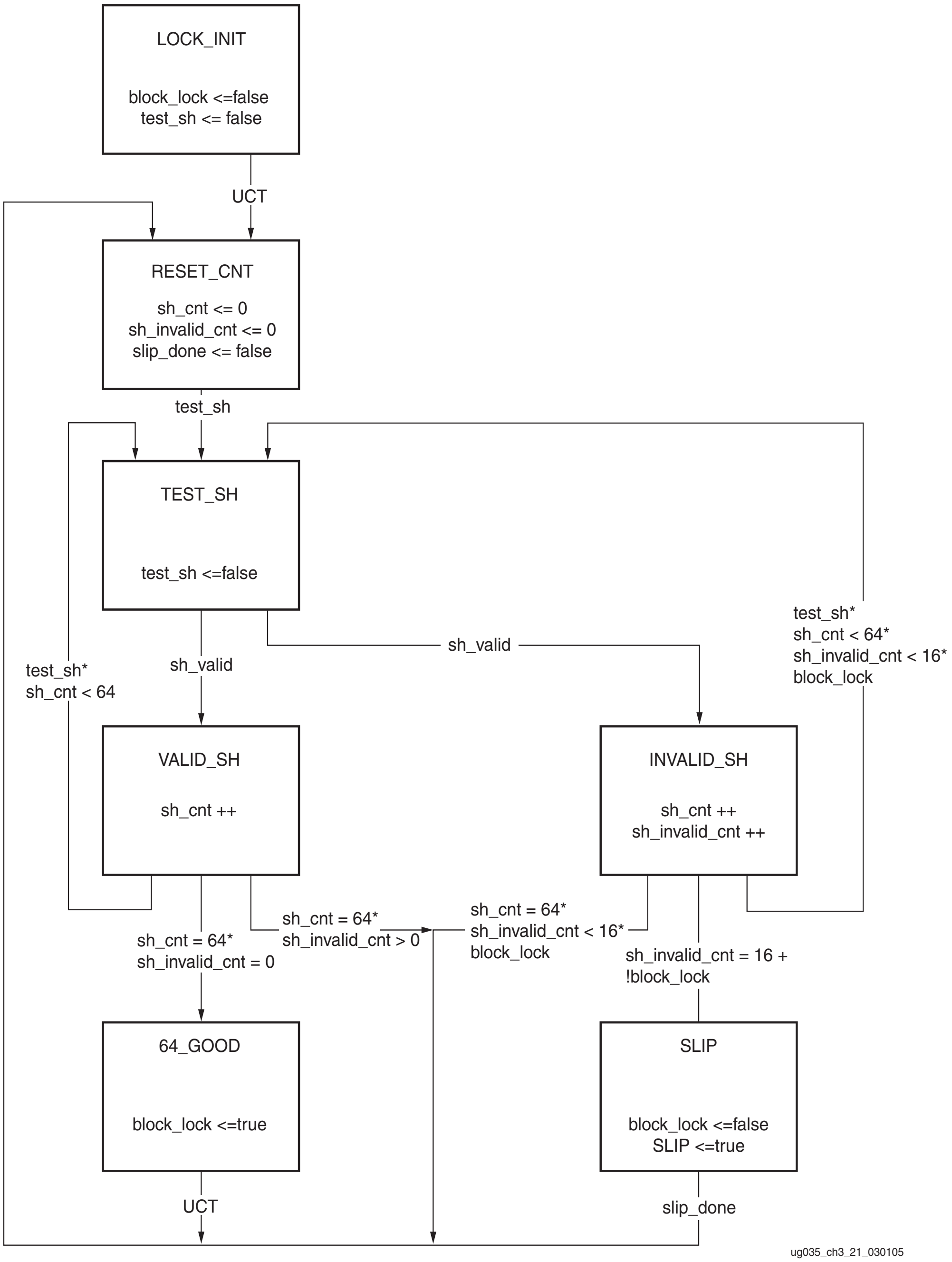


Figure 3-16: **Block Sync State Machine**

Functions Common to Most Protocols

Clock Correction

Clock correction is needed when the rate that data is fed into the write side of the receive FIFO is either slower or faster than the rate that data is retrieved from the read side of the receive FIFO. The rate of write data entering the FIFO is determined by the frequency of RXRECCLK. The rate of read data retrieved from the read side of the FIFO is determined by the frequency of RXUSRCLK.

Append/Remove Idle Clock Correction

When the attribute CLK_COR_SEQ_DROP is asserted Low and CLK_CORRECT_USE is set to logic 1, the Append/remove Idle Clock Correction mode is enabled. The Append/remove Idle Clock Correction mode corrects for differing clock rates by finding idles in the bitstream, and then either appending or removing idles at the point where the idles were found.

There are a few attributes that need to be set by the user so that the append/remove function can be used correctly. The attribute CLK_COR_MAX_LAT sets the maximum latency through the receive FIFO. If the latency through the receive FIFO exceeds this value, idles are removed so that latency through the receive FIFO is less than CLK_COR_MAX_LAT.

The attribute CLK_COR_MIN_LAT sets the minimum latency through the receive FIFO. If the latency through the receive FIFO is less than this value, idles are inserted so that the latency through the receive FIFO are greater than CLK_COR_MIN_LAT. A correction to the latency due to a CLK_COR_MAX_LAT violation is never less than CLK_COR_MIN_LAT. This is also true for a correction to the latency due to a CLK_COR_MIN_LAT violation; the resulting latency after the correction is greater than CLK_COR_MAX_LAT.

Clock Correction Sequences

Searching within the bitstream for an idle is the core function of the clock correction circuit. The detection of idles starts the correction procedure.

Idles the clock correction circuit should detect are specified by the lower 10 bits of the attributes:

- CLK_COR_SEQ_1_1
- CLK_COR_SEQ_1_2
- CLK_COR_SEQ_1_3
- CLK_COR_SEQ_1_4
- CLK_COR_SEQ_2_1
- CLK_COR_SEQ_2_2
- CLK_COR_SEQ_2_3
- CLK_COR_SEQ_2_4

The 11th bit of each clock correction sequence attribute determines either an 8- or 10-bit compare.

Detection of the clock correction sequence in the bitstream is specified by eight words consisting of 10 bits each. Clock correction sequences can have lengths of 1, 2, 3, 4 or 8 bytes. When the length specified by the user is between 1 and 4, CLK_COR_SEQ_1_* holds the first pattern to be searched for. CLK_COR_SEQ_1_1 is the least significant byte, which is transmitted first from the transmitter and detected first in the receiver. If CLK_COR_SEQ_2_USE is set to TRUE when the length is between 1 and 4, the sequence specified by CLK_COR_SEQ_2_* is specified as a second pattern to match. In that case, the pattern specified by sequence 1 *or* sequence 2 matches as a clock correction sequence.

When the length specified by the user is eight, CLK_COR_SEQ_1_* holds the first four bytes, while CLK_COR_SEQ_2_* holds the last four bytes. CLK_COR_SEQ_1_1 is the least significant byte, which is transmitted first from the transmitter and detected first in the receiver. CLK_COR__SEQ_2_USE must be set to FALSE.

The clock correction sequence is a special sequence to accommodate frequency differences between the received data (as reflected in RXRECCLK) and RXUSRCLK. Most of the primitives have these defaulted to the respective protocols. Only the GT11_CUSTOM allows this sequence to be set to any specific protocol. The sequence contains 11 bits including the 10 bits of serial data. The 11th bit has two different formats. The typical usage is shown in Table 3-15.

Table 3-15: **Definition of CCS or CBS Bits 9-0**

CCS or CBS bit [10]	Definition of CCS or CBS Bits 9-0			
	CLK_COR_8B10B_DE		64B66B Decoders	
	= TRUE	= FALSE	SYNC Header	Data
0	DISPERR, CHARISK, 8-bit decoded value	10-bit decoded data		
1			XX, sync character	XX, 8-bit encoded data

Table 3-16 is an example of data 11-bit attribute setting, the character value, CHARISK value, and the parallel data interface, and how each corresponds with the other.

Table 3-16: **Clock Correction Sequence/Data Correlation for 16-Bit Data Port**

Attribute Setting	Character	CHARISK	TXDATA (hex)
CLK_COR_SEQ_1_1 = 00110111100	K28.5	*1*	*BC*
CLK_COR_SEQ_1_2 = 00010010101	D21.4	*0*	*95*
CLK_COR_SEQ_1_3 = 00010110101	D21.5	*0*	*B5*
CLK_COR_SEQ_1_4 = 00010110101	D21.5	*0*	*B5*

64B/66B Clock Correction Settings Example

Table 3-17: **64B/66B Clock Correction Settings Example**

Attribute Setting	Notation	10GBASE_R Control Code
CLK_COR_SEQ_1_1 = 11000011110	Block Field	Block Field
CLK_COR_SEQ_1_2 = 10000000000	/I/	0x00
CLK_COR_SEQ_1_3 = 10000000000	/I/	0x00
CLK_COR_SEQ_1_4 = 10000000000	/I/	0x00
CLK_COR_SEQ_2_1 = 10000000000	/I/	0x00
CLK_COR_SEQ_2_2 = 10000000000	/I/	0x00
CLK_COR_SEQ_2_3 = 10000000000	/I/	0x00
CLK_COR_SEQ_2_4 = 10000000000	/I/	0x00

Note: The clock correction block works with the encoded data to determine when and what to perform clock correction on. Because of this fact, the clock correction must be done on an 8-byte block of data. This is the control form of byte sync = 10, block type = 1e, and data = 0(encoded idles).

CLK_COR_SEQ_1_MASK, CLK_COR_SEQ_2_MASK, CLK_COR_SEQ_LEN

These attributes are used to correctly define the clock correction sequences. The CLK_COR_SEQ_LEN attribute defines how many bytes long the clock correction sequence is. The possible values are 1, 2, 3, 4, and 8. In the case of 8 bytes, the CLK_COR_SEQ_2_USE must be set to FALSE. Table 3-18 shows the settings for a 2-byte clock correction sequence. Note that the CLK_COR_SEQ_1_MASK is set to indicate a 2-byte sequence as well.

Table 3-18: **Clock Correction Mask Example Settings (No Mask)**

Attribute	Setting	Definition
CLK_COR_SEQ_1_1	`00110111100`	Defines a K28.5.
CLK_COR_SEQ_1_2	`00010010101`	Defines a D21.4.
CLK_COR_SEQ_1_MASK	`1100`	Check compare first 2 bytes.
CLK_COR_SEQ_LEN	`2`	Complete sequence is 2 bytes.

In some cases, more than two clock correction sequences can be defined in a protocol. For this case, the CLK_COR_SEQ_MASK attributes can "mask" portions of the sequence. For example, there are three clock correction sequences:

1. K28.5, D.21.4, D21.5, D25.1;
2. K28.5, D.21.4, D21.5, D25.2;
3. K28.5, D.21.4, D21.5, D25.3.

Any of these sequences should be treated as a clock correction sequence. However, the CLK_COR_SEQ_*_* only allow two sequences. Since 3 of the 4 bytes are always the same, the CLK_COR_SEQ_MASK can be set to a logic 1, which corresponds to the last byte. Table 3-19 shows the setting for this case. Note that bit 3 corresponds to SEQ_*_4.

Table 3-19: **Clock Correction Mask Example Settings (Mask Enabled)**

Attribute	Setting	Definition
CLK_COR_SEQ_1_1	`00110111100`	Defines a K28.5.
CLK_COR_SEQ_1_2	`00010010101`	Defines a D21.4.
CLK_COR_SEQ_1_3	`00010110101`	Defines a D21.5.
CLK_COR_SEQ_1_4	`00010110101`	Defines a D21.5.
CLK_COR_SEQ_1_MASK	`1000`	Check compare first 3 bytes. If a 4-byte sequence contains the first 3 bytes, it is determined to be a clock correction sequence. The fourth byte can be any character.
CLK_COR_SEQ_LEN	`4`	Complete sequence is 4 bytes.

Notes:

1. Ensure that sequences that are not clock correction do not fall into this mask definition.

Determining Correct CLK_COR_MIN_LAT

To determine the correct CLK_COR_MIN_LAT value, several requirements must be met.

- CLK_COR_MIN_LAT must be greater than or equal to 12 (16 for 8 byte characters).
- CLK_COR_MIN_LAT and CLK_COR_MAX_LAT must be multiples of CCS/CBS lengths and ALIGN_COMMA_WORD.
- For symbols less than 8 bytes, (CLK_COR_MIN_LAT – CHAN_BOND_LIMIT) > 20. For symbols of 8 bytes, (CLK_COR_MIN_LAT – CHAN_BOND_LIMIT) > 24.

The defaults of 36 and 44 meet the above requirements.

Channel Bonding

Channel bonding is the technique of tying several serial channels together to create one aggregate channel. Several channels are fed on the transmit side by one parallel bus and reproduced on the receive side as the identical parallel bus. The maximum number of serial differential pairs that can be bonded is 24. The bonded channels consist of one master transceiver and one to 23 slave transceivers. The CHBONDI/CHBONDO buses of the transceiver are daisy-chained together as shown in Figure 3-17. When the master transceiver detects a channel bond alignment sequence in its data stream, it signals the slaves to perform channel bonding by driving its CHBONDO bus as follows in Table 3-20.

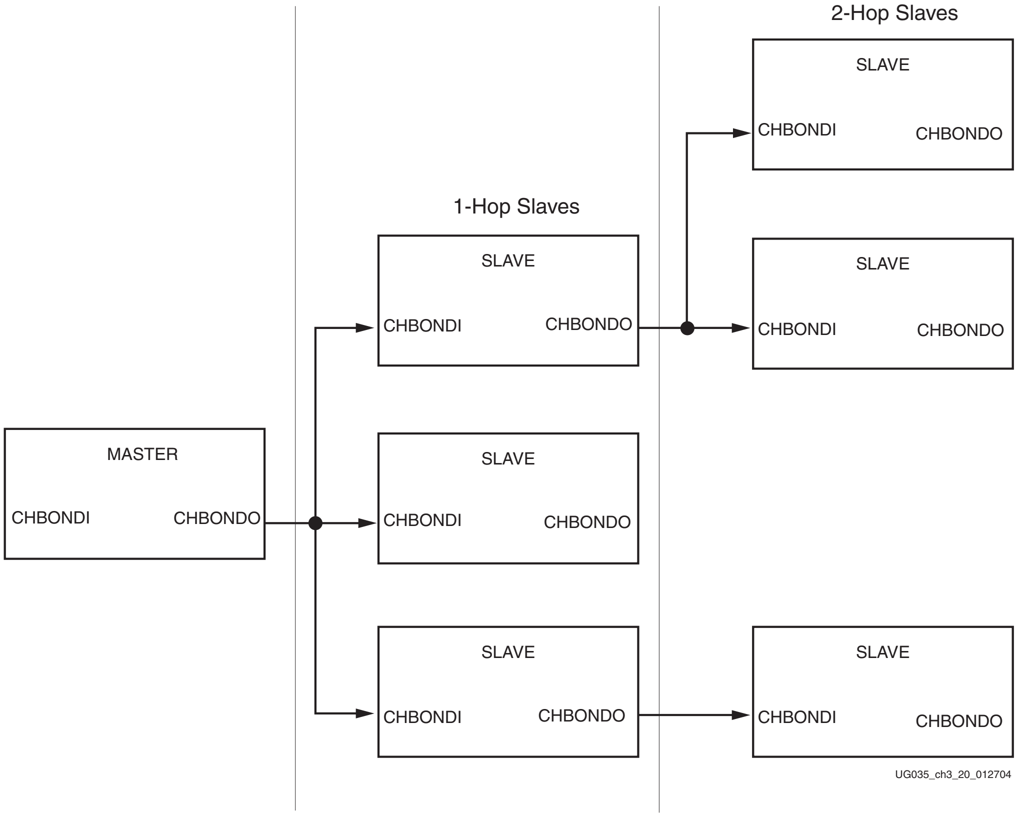


Figure 3-17: **Daisy-Chained Transceiver CHBONDI/CHBONDO Buses**

Table 3-20: **Channel Bond Alignment Sequence**

Detected	CHBONDO Bus
No Channel Bond	$XX000_2$
Channel Bond - Byte 0	$XX100_2$
Channel Bond - Byte 1	$XX101_2$
Channel Bond - Byte 2	$XX110_2$
Channel Bond - Byte 3	$XX111_2$

Whether a slave is a 1-hop or 2-hop slave, internal logic causes the data driven on the CHBONDO bus from the master to be recognized by the slaves at the same time and must be deterministic. Therefore, it is important that the interconnect of CHBONDO-to-CHBONDI not contain any pipeline stages. The data must transfer from CHBONDO to CHBONDI in one clock.

The data streams input to the channel bonded transceivers can be skewed in time from each other. The maximum byte skew that the channel bond logic should allow is set by the attribute CHAN_BOND_LIMIT. During the channel bond operation, the slave receives notification of the master's alignment code location via the CHBONDO bus. If a slave detects the position of its alignment code to be outside the window of CHAN_BOND_LIMIT from the master, then the slave does not perform the channel bond and sets a channel bond error flag. If the channel bond is successful, the slave outputs its skew relative to the master. The skew and channel bond error flag are available on the RXSTATUS bus.

For place and route, the transceiver has one restriction. This is required when channel bonding is implemented. Because of the delay limitations on the CHBONDO to CHBONDI ports, linking of the Master to a Slave_1_hop must run either in the X or Y direction, but not both.

In Figure 3-18, the two Slave_1_hops are linked to the master in only one direction. To navigate to the other slave (a Slave_2_hops), both X and Y displacement is needed. This slave needs one level of daisy-chaining, which is the basis of the Slave_2_hops setting.

Figure 3-18 and Table 3-13 show the channel bonding mode and linking for an XC4VFX20 and XC4VFX70 devices, which contain eight transceivers per chip.

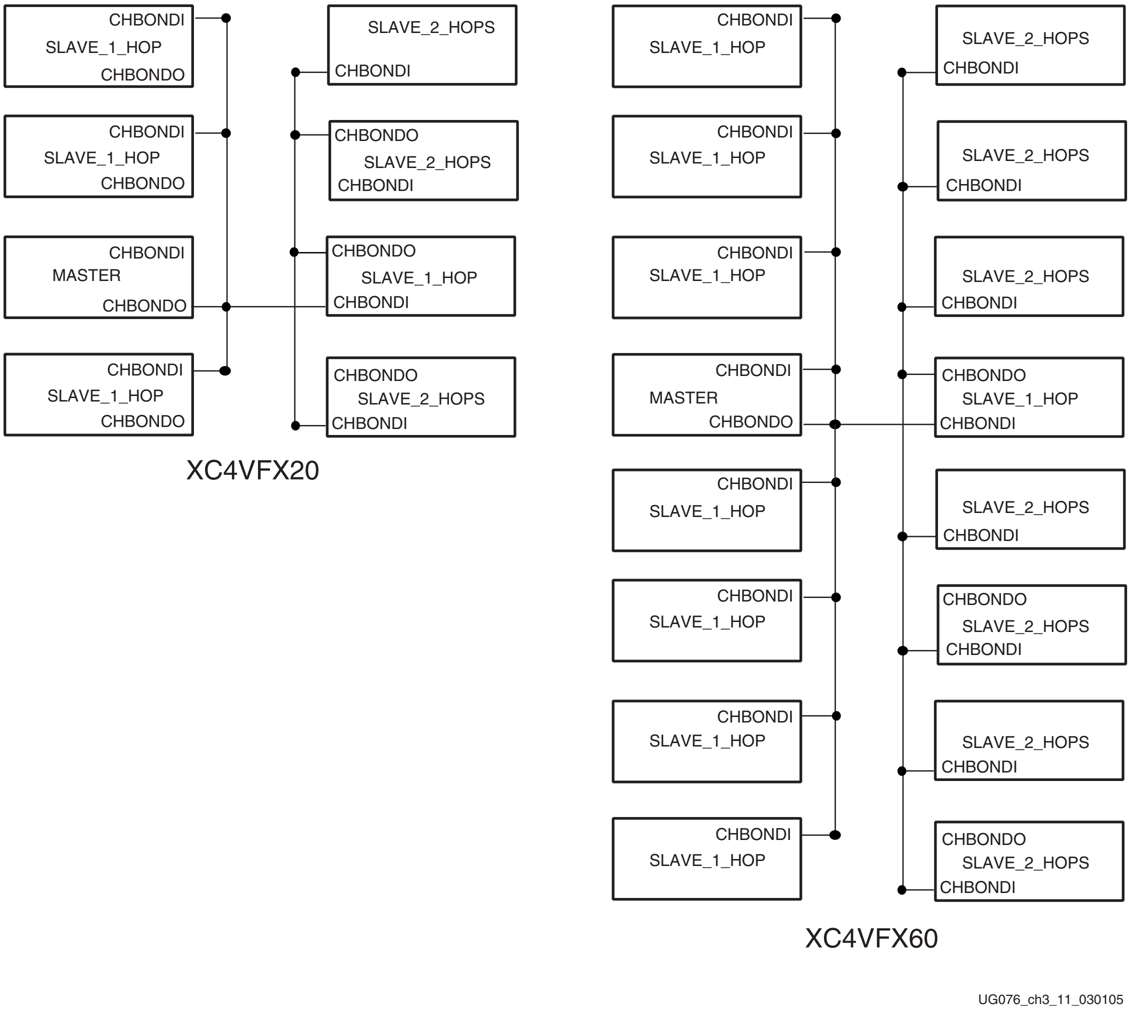


Figure 3-18: **XC4VFX20/XC4VFX60 Device Implementation**

CCCB_ARBITRATOR_DISABLE

This attribute (always defaults to FALSE) allows clock correction and channel bonding sequences to appear consecutively in the data stream. There are no requirements on how close or far apart the clock correction and channel bonding sequences are when this attribute is set to FALSE. However, when in this mode, the clock correction always takes priority over the channel bonding sequence. See Figure 3-19. Most protocols and applications should use this attribute in the default setting.

If a channel bonding sequence needs to be recognized every time, CCCB_ARBITRATOR_DISABLE should be set to TRUE. While in this mode, channel bonding always occurs (providing all the other channel bonding attributes are set accordingly). However, for proper operation of clock correction and channel bonding, there must be a minimum of 32 bytes between clock correction and channel bonding sequences at all times. Also, there must be a 32-byte gap between channel bonding sequences as shown in Figure 3-19.

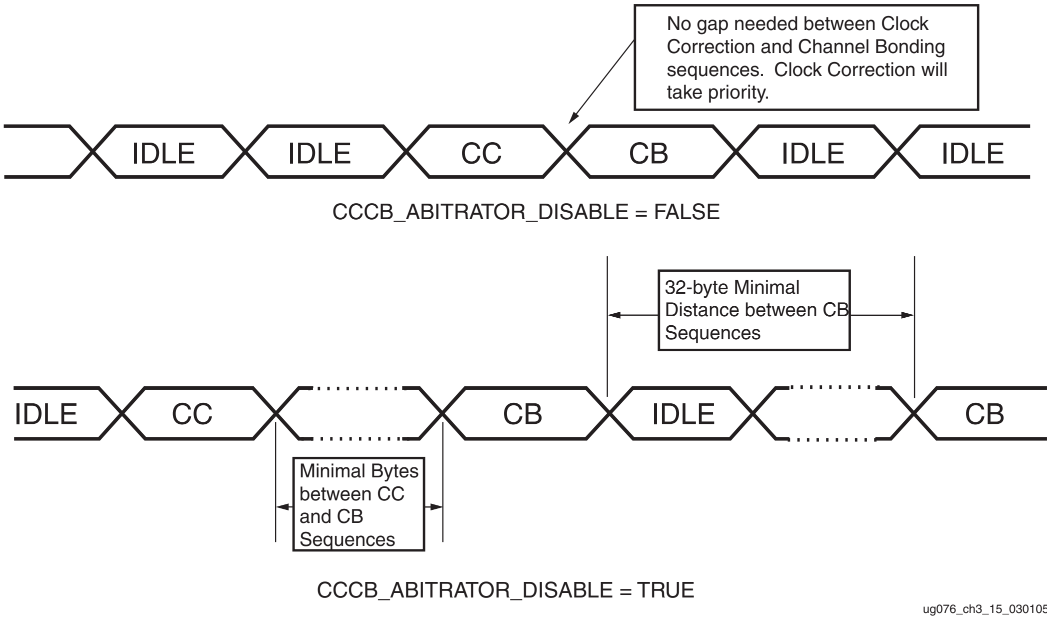

No gap needed between Clock Correction and Channel Bonding sequences. Clock Correction will take priority.
IDLE
IDLE
CC
CB
IDLE
IDLE
CCCB_ABITRATOR_DISABLE = FALSE
32-byte Minimal Distance between CB Sequences
IDLE
CC
CB
IDLE
CB
Minimal Bytes between CC and CB Sequences
CCCB_ABITRATOR_DISABLE = TRUE
ug076_ch3_15_030105

Figure 3-19: **Effects of CCCB_ARBITRATOR_DISABLE = TRUE**

CHAN_BOND_SEQ_1_MASK, CHAN_BOND_SEQ_2_MASK, CHAN_BOND_SEQ_LEN, CHAN_BOND_SEQ_*_*

These attributes operate exactly the same as their clock correction counterparts. See CLK_COR_SEQ_1_MASK for details. See Table 3-15 for the definitions for CHAN_BOND_SEQ_*_* bits.

Disable Channel Bonding

The MGT allows disabling the channel bonding by using the CHAN_BOND_MODE set to OFF.

Status and Event Bus

The Virtex-4 design has merged several signals together to provide extra functionality over the Virtex-II Pro design. The signals CHBONDDONE, RXBUFSTATUS, and RXCLKCORCNT were previously used independently of each other to indicate status. In the Virtex-4 design, these signals are concatenated together to provide a status and event bus.

There are two modes of this concatenated bus, status mode and event mode. In status mode, the bus indicates either the difference between the read and write pointers of the receive side FIFO or the skew of the last channel bond event.

Status Indication

In status mode, the RXSTATUS pins alternate between the buffer pointer difference and channel bonding skew. The protocol is described by three sequential clocks (STATUS and DATA are one clock in duration) when operating with a 32-bit or 40-bit internal data-width, or six sequential clocks (STATUS and DATA are two clocks in duration) when operating with a 16-bit or 32-bit internal data width:

```
<STATUS INDICATOR> <DATA0><DATA1>
```

where

STATUS INDICATOR can indicate either pointer difference or channel bonding skew,
DATA0 indicates status data 5:3, and
DATA1 indicates status data 2:0.

Table 3-21 shows the signal values for a pointer difference status where the variable pointerDiff[5:0] holds the pointer difference between the receive write and read pointers:

Table 3-21: **Signal Values for a Pointer Difference Status**

Status	RXSTATUS[5]	RXSTATUS[4:3]	RXSTATUS[2:0]
STATUS INDICATOR	1'b0	2’b01	3’b000
DATA0	1'b0	2’b00	pointerDiff[5:3]
DATA1	1'b0	2’b00	pointerDiff[2:0]

Table 3-22 shows the signal values for a channel bonding skew where the variable cbSkew[5:0] holds the pointer difference between the receive write and read pointers:

Table 3-22: **Signal Values for a Channel Bonding Skew**

Status	RXSTATUS[5]	RXSTATUS[4:3]	RXSTATUS[2:0]
STATUS INDICATOR	1'b0	2’b01	3’b001
DATA0	1'b0	2’b00	cbSkew[5:3]
DATA1	1'b0	2’b00	cbSkew[2:0]

Event Indication

Two types of events can occur. See Table 3-23. When an event occurs, it can override a status indication. An event can only last for one clock and can be signaled by RXSTATUS[5] being set to logic 1, or RXSTATUS[4:3] equating to 2'b10.

Table 3-23: **Signal Values for Event Indication**

Event	RXSTATUS[5]	RXSTATUS[4:3]	RXSTATUS[2:0]
Channel Bond Load	1'b1	2’b00	3’b111
Clock Correction	1'b0	2’b10	3’bXX1 (insertion)
			3’bXX0 (deletion)

Note: An event always overrides status, but after an event is completed, status continues to alternate between the pointer difference and the channel bond skew.

RXBUFERR

RXBUFERR uses the pointerDiff value to determine an underflow/overflow. When set to logic 1, and underflow or overflow has occurred. Once this bit is set, it can only be reset by RXRESET.

LOOPBACK

LOOPBACK allows the user to send data that is being transmitted directly to the receiver of the transceiver. This is used to isolate where a link fail, whether it is the physical link (such as a missing line card, termination issues, or SI issues), PHY layer, or Link layer. The MGT allows three loopback modes and a repeater mode. Figure 3-20 shows the four modes available while Table 3-24 shows the four settings of LOOPBACK to create the different modes. The repeater attribute must be set to TRUE to enable the repeater mode, which allows the received data to be directly transmitted.

Note: Fabric loopback is not a mode of the MGT, but is supported in the fabric.

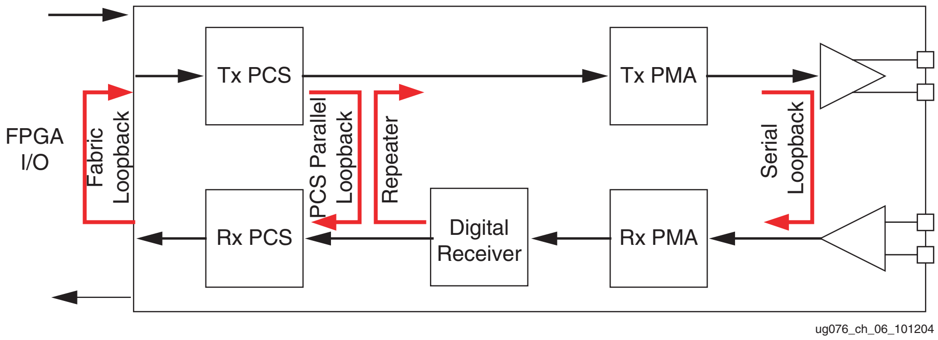


Figure 3-20: **Loopback Modes**

Table 3-24: **Loopback Modes**

Repeater[1]	LOOPBACK[1:0]	Function
FALSE	`00`	Normal Operation (No Loopback)
FALSE	`01`	PCS Parallel
FALSE	`10`	Reserved
FALSE	`11`	Pre-driver Serial Loopback
TRUE	xx	Repeater mode RX → TX path (received data is automatically transmitted)

Notes:

1. Repeater mode requires that the recovered clock is the TX reference clock. (The RXBCLK must drive REFCLK.)

Digital Receiver

The MGT is equipped with a digital receiver that *oversamples* the incoming data for rates at or below 1.25 Gb/s. This mode is available with an 8X mode. The line rate and the maximum VCO rate of the transceiver determine which mode is implemented. These settings are shown in the clocking settings in Figure 2-10 and Figure 2-11. During oversampling modes, the RX PLL must be locked to the reference clock and not the incoming data.

Note: The digital receiver generated clock should not be used to drive the DCM.

Figure 3-21 shows how the 1-Gigabit Ethernet design uses the digital receiver and the consequent clocks. In this mode, the SIPO actually creates 8 bits for each 1.25 Gb/s bit. The digital receiver then sends out a one-eighth clock that is locked to the data, of which only 0 of the 8 incoming bits is sampled.

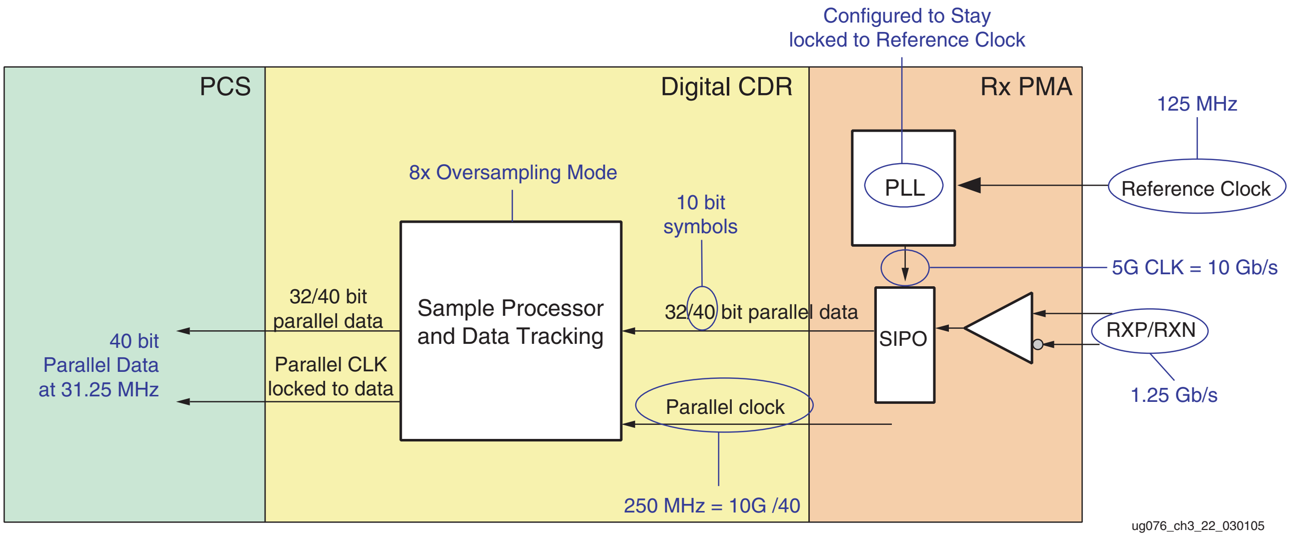


Figure 3-21: **Digital Receiver Example**

PMA Analog Design Considerations

Serial I/O Description

The MGT transmits and receives serial differential signals, using a nominal supply voltage of 1.5 VDC. A serial differential pair consists of a true (V_P) and a complement (V_N) set of signals. The voltage difference represents the transferred data. Thus: $V_P – V_N = V_{DATA}$. Differential switching is performed at the crossing of the two complementary signals so that no separate reference level is needed. A graphical representation of this concept is shown in Figure 4-1.

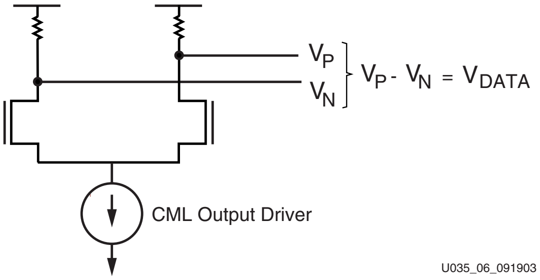


Figure 4-1: **Differential Amplifier**

Differential Transmitter

The MGT is implemented in Current Mode Logic (CML). A CML transmitter output consists of transistors configured as shown in Figure 4-1. The CML uses a positive supply and offers easy interface requirements. In this configuration, both legs of the driver, V_P and V_N, sink current, with one leg always sinking more current than its complement. The CML output consists of a differential pair with 50Ω source resistors. The signal swing is created by switching the current in a common-source differential pair.

The differential transmitter specifications are shown in the *RocketIO DC Specifications* and the *RocketIO Transmitter Switching Characteristics* in the Virtex-4 data sheet (DS302).

Output Swing and Emphasis

The output swing and emphasis levels of the MGTs are fully programmable. Each is controlled via attributes at configuration, but can be modified via the Dynamic Configuration Bus programming bus (Appendix C, "Dynamic Configuration Bus").

Emphasis

The MGT contains a 3-tap transmitter (shown in Figure 4-2). This allows a data driver, pre-cursor driver, and post-cursor driver to help create a cleaner signal at the receiver.

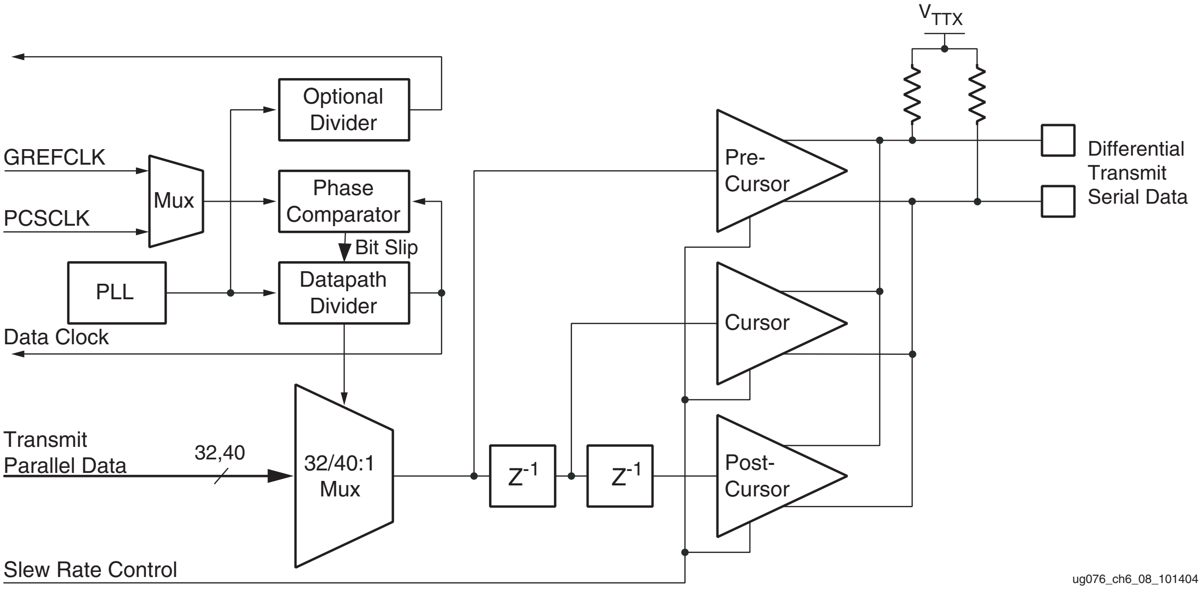


Figure 4-2: **3-Tap Pre-Emphasis**

There are several attributes that control the pre-emphasis characteristics. These are shown in Table 4-1.

Table 4-1: **Attributes Controlling Pre-Emphasis Characteristics**

Attribute	Definition
TXPOST_TAP_PD	Disables the POST driver which disables the pre-emphasis.
TXDAT_TAP_DAC	Controls the drive strength of the data driver. This value should be much larger (~4 times) than TXPOST_TAP_DAC and (~8 times) TXPRE_TAP_DAC.
TXPOST_TAP_DAC	Controls the drive strength of the post data driver. This value will usually be twice the value of TXPRE_TAP_DAC.
TXHIGHSIGNALEN	This doubles the current level of the driver when set to a logic 1. When set to a logic 0, the current level is set for applications interfacing to an XFP or similar device that requires low signal strength.
TXPRE_TAP_DAC	Controls the drive strength of the pre-data driver. This value should be very small.
TXPRE_TAP_PD	Disables the PRE driver used to disable the pre-emphasis.

The effect of these three taps are shown in Figure 4-3. The pre-cursor (pre-driver) helps the start of the pulse on the receiver. This degradation is small, which is why TXPRE_TAP_DAC is recommended to be small compared to the other driver settings. The post-cursor (post-driver) improves the "tail" of the pulse at the receiver. TXPOST_TAP_DAC should be less than TXDAT_TAP_DAC, so as not to destroy the pulse and also to preserve the subsequent pulse of the next bit being transmitted in an actual system.

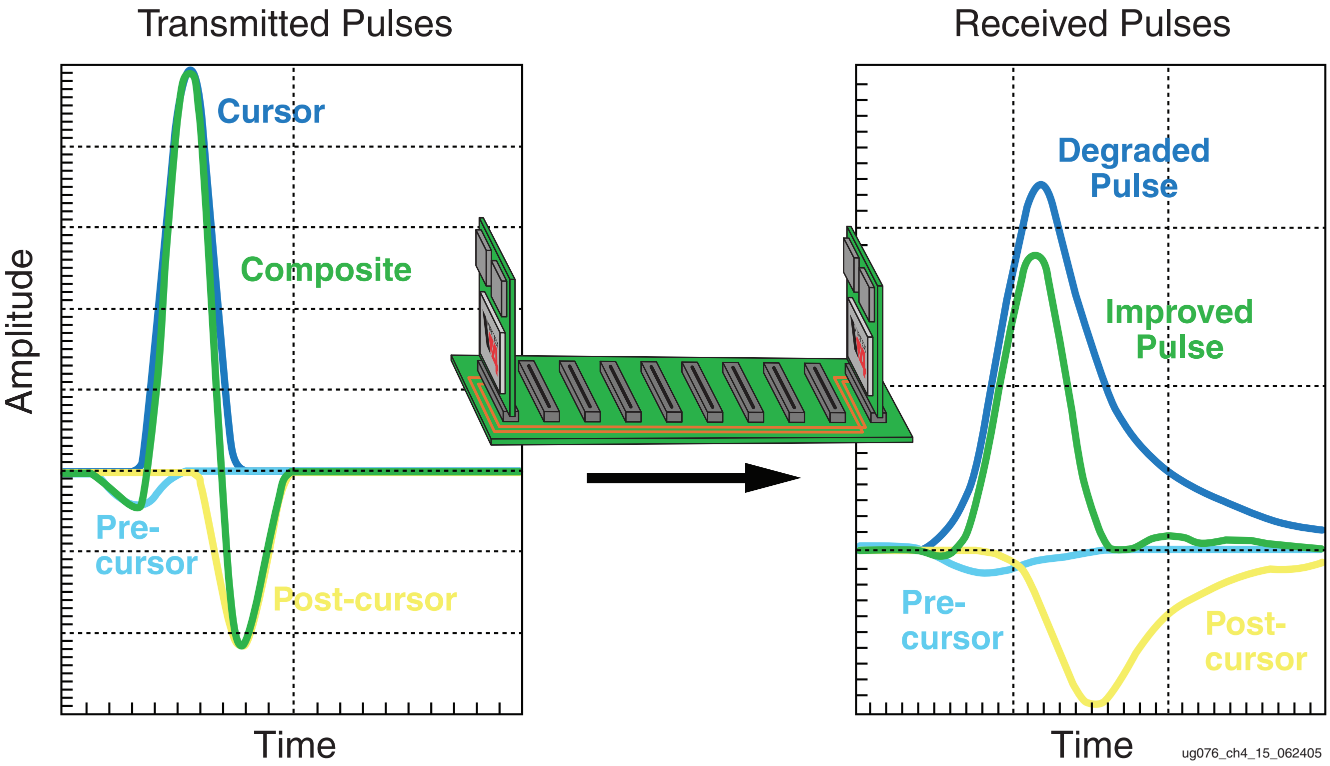


Figure 4-3: **Effect of 3-Tap Pre-Emphasis on a Pulse Signal**

Table 4-2 shows the recommended TXDAT_TAP_DAC and TXPOST_TAP_DAC settings for a range of differential swing settings and line loss conditions. 3.5 dB of line loss approximates the signal attenuation for a 10 Gb/s signal across 5 inches of FR4. For lower data rates, traces longer than 5 inches of FR4 can be traversed before the signal experiences 3.5 dB of attenuation.

Table 4-2: **Recommended TXDAT_TAP_DAC and TXPOST_TAP_DAC Settings**

Line Loss (dB)	Differential Swing (mV)	TXDAT_TAP_DAC	TXPOST_TAP_DAC
3.5	600	10011	00000
	800	11011	00010
7	400	01001	00001
	600	10001	00101
	800	11000	01001
10	400	01000	00100
	600	01111	01001
	800	10110	01111

Table 4-2: **Recommended TXDAT_TAP_DAC and TXPOST_TAP_DAC Settings *(Continued)***

Line Loss (dB)	Differential Swing (mV)	TXDAT_TAP_DAC	TXPOST_TAP_DAC
14	400	00110	00110
	600	01101	01101
	800	10100	10100
17	400	00110	01001
	600	01100	10001
	800	10010	11000

The TXPRE_TAP_DAC value can be set to some fraction of the TXPOST_TAP_DAC value. However, for most channels, TXPRE_TAP_DAC can be set to off, unless it is required to improve the performance of the linear receiver equalizer and decision feedback equalizer. It is recommended that all the attributes listed in this section are made accessible through the Dynamic Configuration Port (see Appendix C, "Dynamic Configuration Bus"). This allows transmitter parameters to be fine-tuned in the lab to match the characteristics of a particular channel.

With MGT emphasis, the initial differential voltage swing is boosted to create a stronger rising or falling waveform. This method compensates for high frequency loss in the transmission media that would otherwise limit the magnitude of this waveform. The effects of emphasis are shown in four scope screen captures, Figure 4-4 through Figure 4-7. (Also, see "Appendix C, "Dynamic Configuration Bus" for the signal or attribute Effects.) Note that Figure 4-4 through Figure 4-7 are for illustration purposes only and do not correspond to actual MGT data.

Figure 4-4 shows the transmit portion of a link with minimal pre-emphasis and Figure 4-5 corresponds to the signal after 24 inches of FR4. Figure 4-6 shows the transmit portion of a link with pre-emphasis over driving the rising and falling edges. This produces a less than optimal eye at the transmit side, but after the attenuation of the rising and falling edges of 24 inches of FR4 (Figure 4-7), the optimal eye is restored for the receiver. A second characteristic of MGT emphasis is that the overdriving level is reduced after some time to the normal driving level, thereby minimizing the voltage swing necessary to switch the differential pair into the opposite state.

Lossy transmission lines cause the dissipation of electrical energy. This emphasis technique extends the distance to which signals can be driven down lossy line media and increases the signal-to-noise ratio at the receiver.

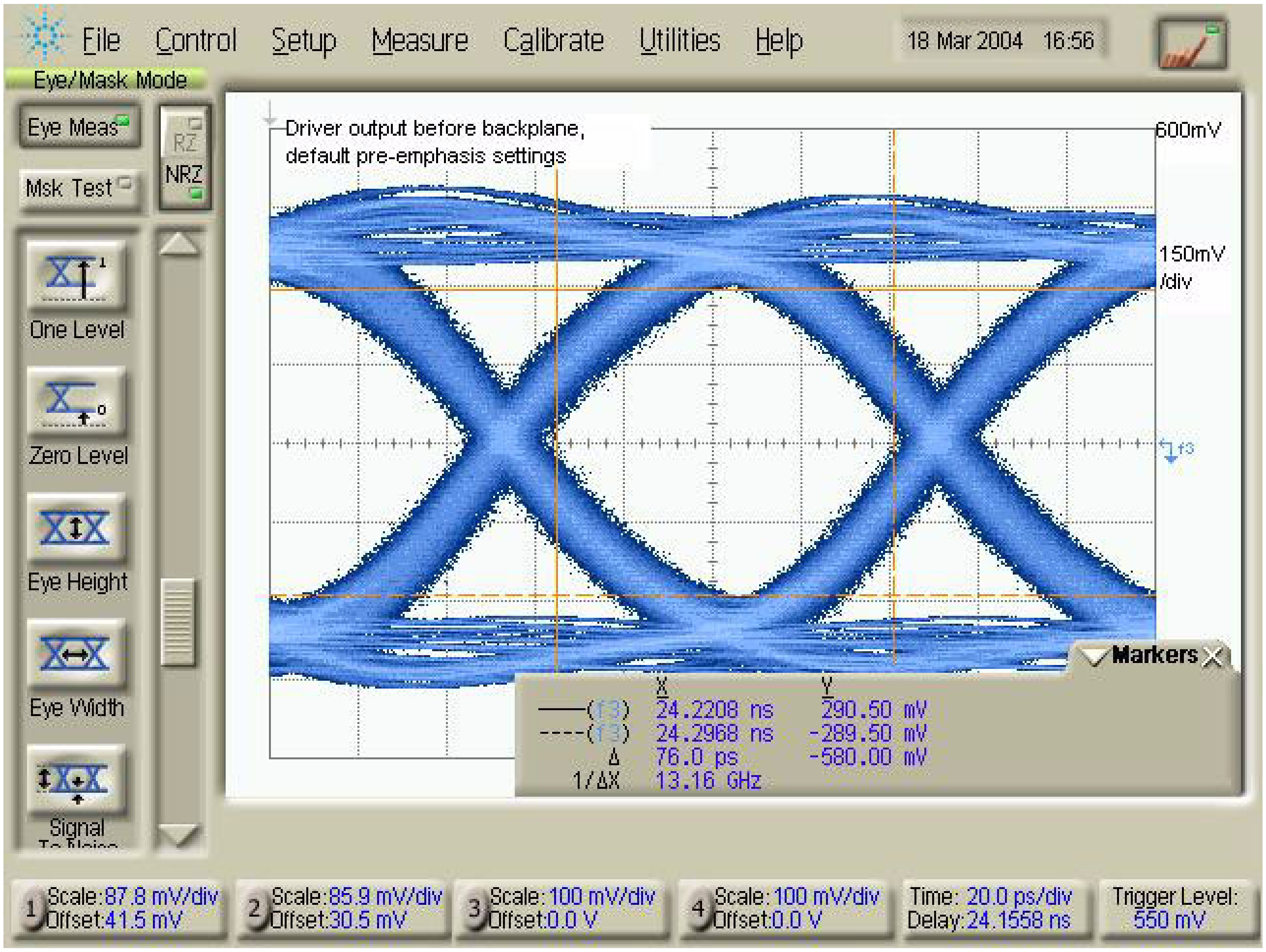
Emphasis can be described from two perspectives, additive to the smaller voltage (V_{SM}) (pre-emphasis) or subtractive from the larger voltage (V_{LG}) (de-emphasis). The resulting benefits in compensating for channel loss are identical. It is simply a relative way of specifying the effect at the transmitter.

The equations for calculating pre-emphasis as a percentage and dB are as follows:

$$\text{Pre-Emphasis\%} = ((V_{LG}-V_{SM})/V_{SM}) \times 100$$
$$\text{Pre-Emphasis dB} = 20\log(V_{LG}/V_{SM})$$

The equations for calculating de-emphasis as a percentage and dB are as follows:

$$\text{De-Emphasis\%} = (V_{LG} - V_{SM})/V_{LG}) \times 100$$
$$\text{De-Emphasized dB} = 20\log(V_{SM}/V_{LG})$$

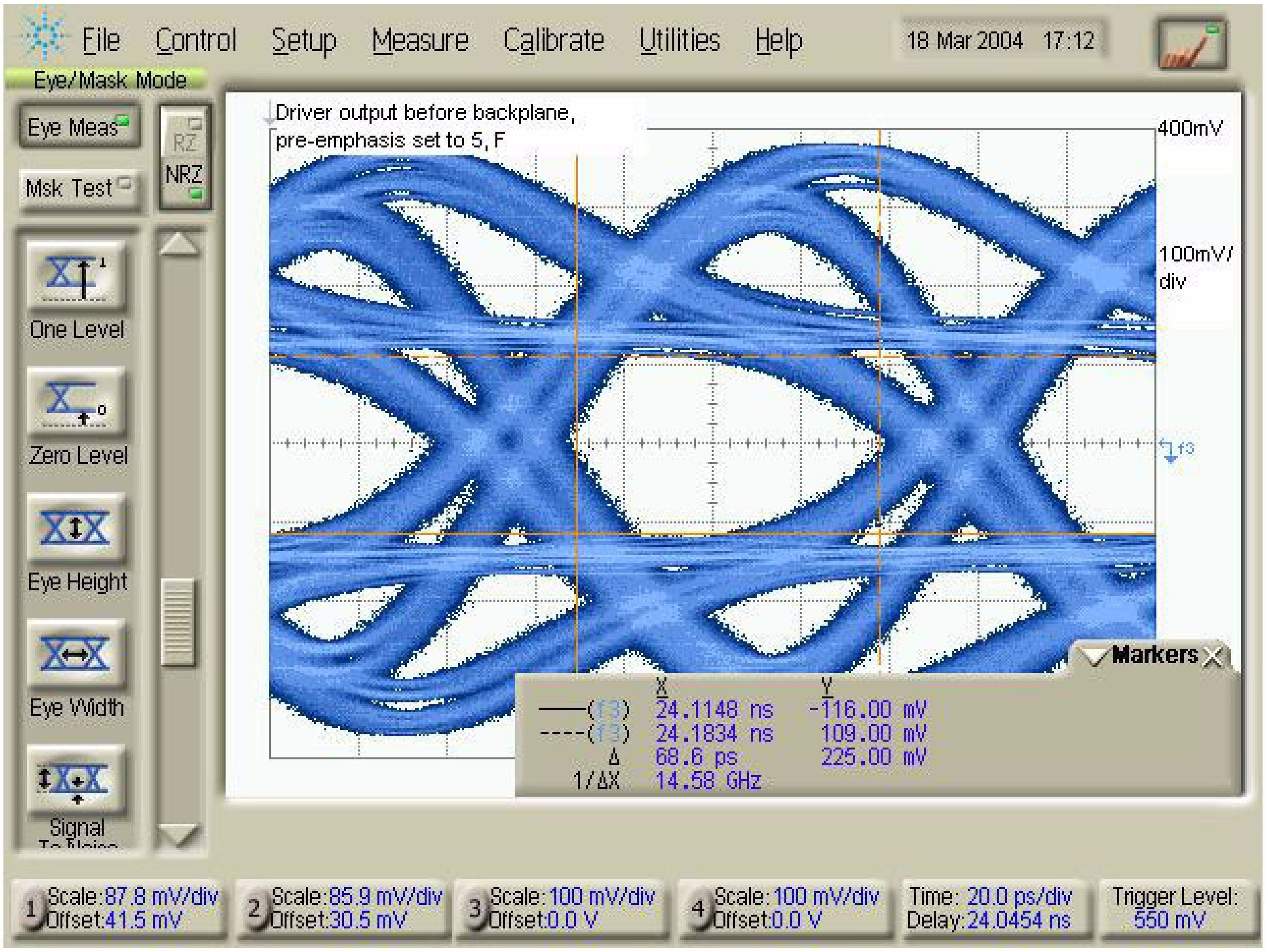


Driver output before backplane,
default pre-emphasis settings

```
            X               Y
——(f3)   24.2208 ns     290.50 mV
----(f3) 24.2968 ns    -289.50 mV
     Δ   76.0 ps       -580.00 mV
  1/ΔX   13.16 GHz
```

1 Scale:87.8 mV/div Offset:41.5 mV
2 Scale:85.9 mV/div Offset:30.5 mV
3 Scale: 100 mV/div Offset:0.0 V
4 Scale: 100 mV/div Offset:0.0 V
Time: 20.0 ps/div Delay:24.1558 ns
Trigger Level: 550 mV

Figure 4-4: **TX with Minimal Pre-Emphasis**



Driver output before backplane, pre-emphasis set to 5, F

```
            X              Y
——(f3)   24.1148 ns    -116.00 mV
----(f3)  24.1834 ns     109.00 mV
     Δ    68.6 ps        225.00 mV
  1/ΔX    14.58 GHz
```

```
1 Scale:87.8 mV/div   2 Scale:85.9 mV/div   3 Scale:100 mV/div   4 Scale:100 mV/div   Time: 20.0 ps/div    Trigger Level:
  Offset:41.5 mV        Offset:30.5 mV        Offset:0.0 V         Offset:0.0 V         Delay:24.0454 ns     550 mV
```

Figure 4-5: **TX with Maximal Pre-Emphasis**

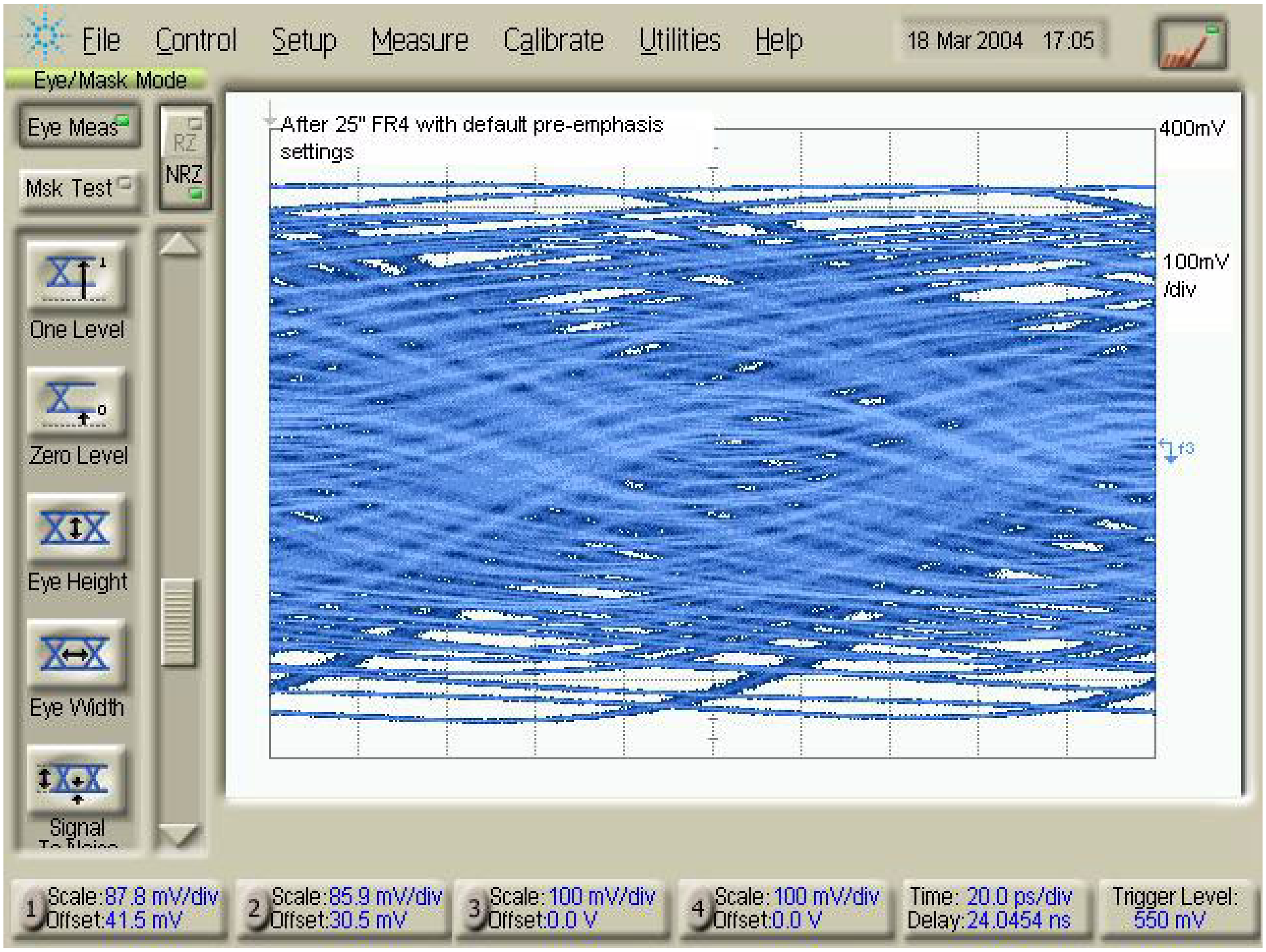
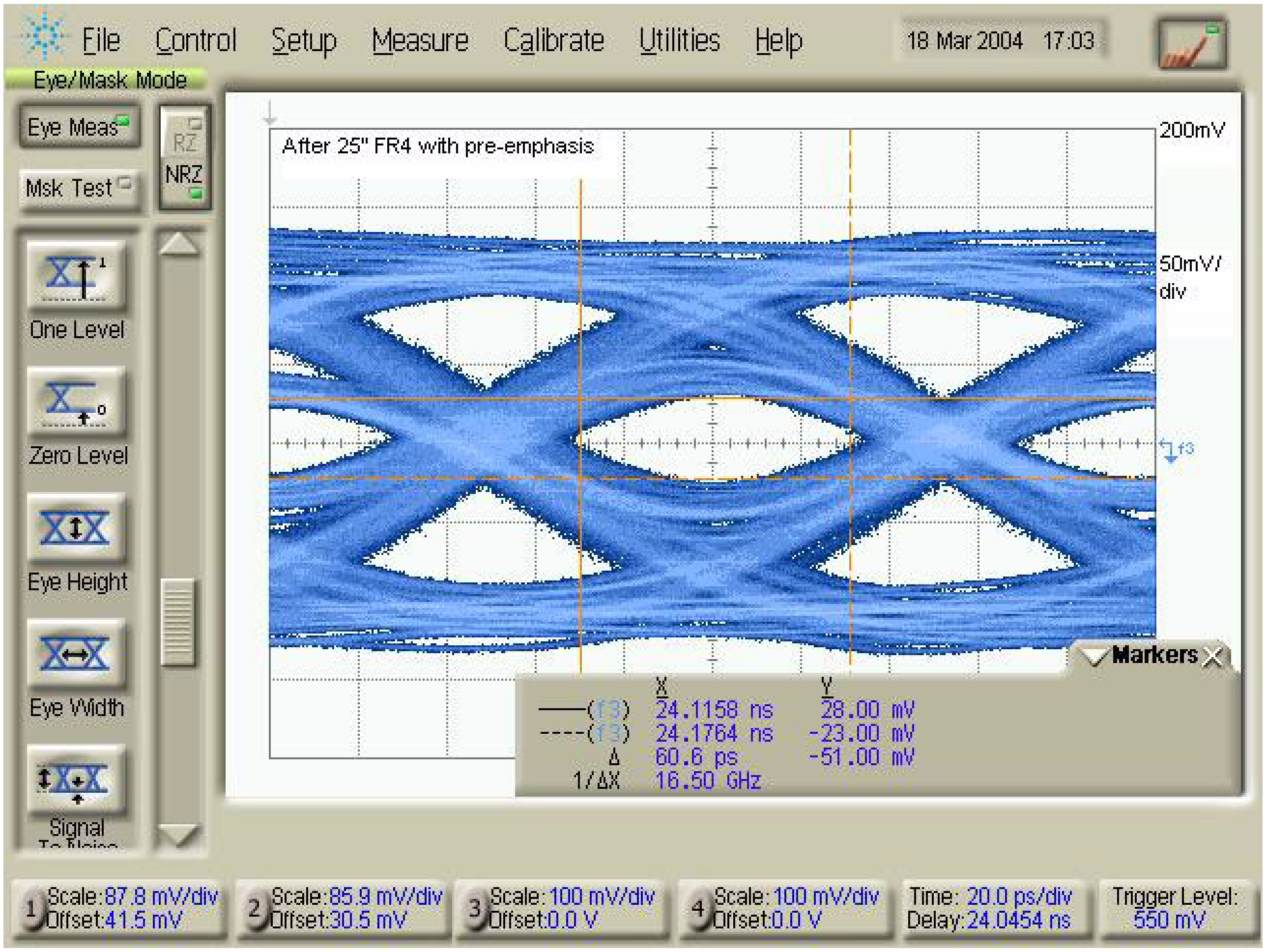

File
Control
Setup
Measure
Calibrate
Utilities
Help
18 Mar 2004 17:05
Eye/Mask Mode
Eye Meas
Msk Test
RZ
NRZ
One Level
Zero Level
Eye Height
Eye Width
Signal To Noise
After 25" FR4 with default pre-emphasis settings
400mV
100mV /div
f3
1 Scale:87.8 mV/div Offset:41.5 mV
2 Scale:85.9 mV/div Offset:30.5 mV
3 Scale:100 mV/div Offset:0.0 V
4 Scale:100 mV/div Offset:0.0 V
Time: 20.0 ps/div Delay:24.0454 ns
Trigger Level: 550 mV

Figure 4-6: **RX After 24 Inches FR4 and Minimal Pre-Emphasis**



	X	Y
——(f3)	24.1158 ns	28.00 mV
----(f3)	24.1764 ns	-23.00 mV
Δ	60.6 ps	-51.00 mV
1/ΔX	16.50 GHz	

1 Scale:87.8 mV/div Offset:41.5 mV
2 Scale:85.9 mV/div Offset:30.5 mV
3 Scale:100 mV/div Offset:0.0 V
4 Scale:100 mV/div Offset:0.0 V
Time: 20.0 ps/div Delay:24.0454 ns
Trigger Level: 550 mV

Figure 4-7: **RX After 24 Inches FR4 and Maximal Pre-Emphasis**

Differential Receiver

The differential receiver accepts the V_P and V_N signals, carrying out the difference calculation V_P -V_N electronically.

All input data must be differential and nominally biased to a common mode voltage of 0.25 V – 2.5V or AC coupled. Internal terminations provide for simple 50Ω transmission line connection.

The differential receiver parameters are shown in the *RocketIO DC Specifications* and the *RocketIO Receiver Switching Characteristics* in the Virtex-4 data sheet.

Clock and Data Recovery

The serial MGT input is locked to the input data stream through Clock and Data Recovery (CDR), a built-in feature of the MGT. CDR keys off of the rising and falling edges of incoming data and derives a clock that is representative of the incoming data rate.

The derived clock, RXRECCLK, is generated and locked to as long as it remains within the specified component range. This clock is presented to the FPGA fabric at 1/32nd and 1/40th the incoming data rate depending on mode.A sufficient number of transitions must be present in the data stream for CDR to work properly. The CDR circuit is guaranteed to work with 8B/10B and 64B/66B encoding. Further, CDR requires approximately 5,000 transitions upon powerup to guarantee locking to the incoming data rate.

Another feature of CDR is its ability to accept an external precision clock, REFCLK, which either acts to clock incoming data or to assist in synchronizing the derived RXRECCLK. For further clarity, the TXUSRCLK is used to clock data from the FPGA core to the TX FIFO. The FIFO depth accounts for the slight phase difference between these two clocks. If the clocks are locked in frequency, then the FIFO acts much like a pass-through buffer.

Receiver Lock Control

During normal operation, the receiver PLL automatically locks to incoming data (when present) or to the local reference clock (when data is not present). This is the default configuration for all primitives. This function can be overridden via the RXDIGRX port, as defined in Table 4-3.

Table 4-3: **RXDIGRX Definition**

RXDIGRX	Description
0	Automatic (Default)
1	Lock to local reference

When receive PLL lock is forced to the local reference, phase information from the incoming data stream is ignored. Data continues to be sampled, but synchronously to the local reference rather than relatively to the edges in the data stream.

Receive Equalization

In addition to transmit emphasis, the MGT provides a programmable receive equalization feature to further compensate the effects of channel attenuation at high frequencies. Copper traces on printed circuit boards, including backplanes, have low-pass filter characteristics. The impedance mismatch boundaries can also cause signal degradation. The MGT has an equalizer in the receiver which can essentially null the lossy attenuation effects of the PCB at GHz frequencies.

The receiver equalization circuit is comprised of three concatenated gain stages. Each stage is a peaking equalizer with a different center frequency and programmable gain. This allows varying amounts of gain to be applied depending on the overall frequency response of the channel loss. Channel loss is defined as the summation of all losses through the PCB traces, vias, connectors, and cables present in the physical link.

Ideally, the overall frequency response of the three gain stages should be the inverse of the overall channel loss response. Fine shaping of the gain response is possible by varying the gain for each stage and, hence, the gain at each of the three center frequencies.

The attribute RXAFEEQ[2:0] controls the gain settings for the receiver equalizer and can be adjusted by the Dynamic Configuration Bus.

Figure 4-7 shows the AC response of the 3-stage continuous-time linear equalizer.

From the graph (Figure 4-8), the four response curves correspond to their respective RXAFEEQ setting. Each curve shows the combined gain response of all three equalizer stages expressed in dB. This gain is applied to the incoming signal before the DFE block. See"Decision Feedback Equalizer." Using one of the four settings here provides sufficient receiver equalization flexibility for most applications. The choice of RX_AFE_EQ setting depends on the amount of high frequency loss in the transmission media. Links with more transmission loss should use the higher gain settings.

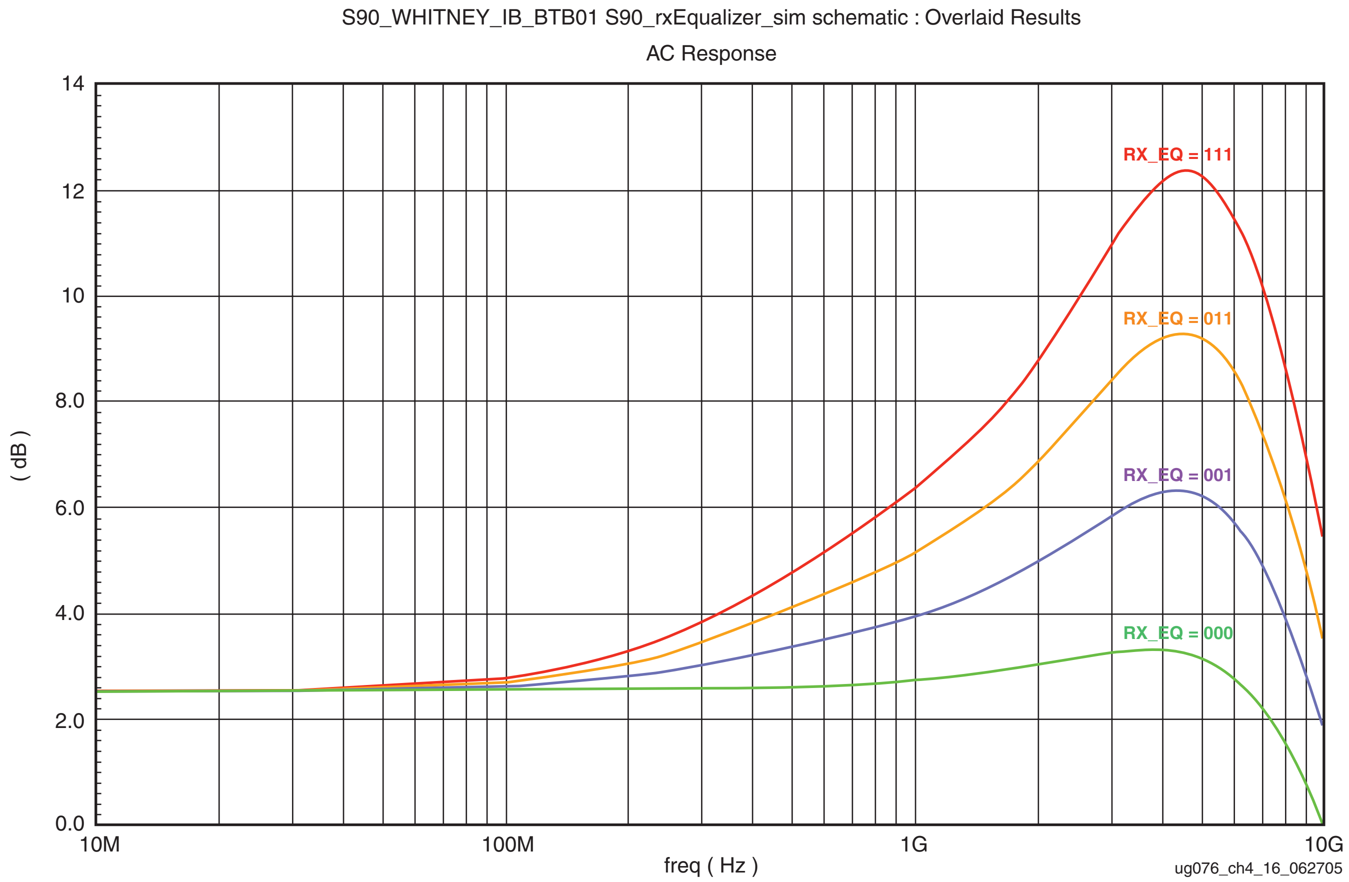


Figure 4-8: **AC Response of Continuous-Time Linear Receiver Equalizer**

Decision Feedback Equalizer

The Virtex-4 RocketIO receiver includes another sophisticated equalization technique known as Decision Feedback Equalization (DFE). DFE can be useful for extending the range in unique applications up to 6.25 Gb/s, particularly legacy channels with poor signal integrity. These are usually channels that were not initially designed to handle high serial data rates.

DFE works by actively shifting the incoming signal based on a history of the received data. Instead of amplifying attenuated signals, DFE directly removes signal energy that leaks from one bit to the following bits. By using the specific data history, the proper shifting occurs for any data pattern. The advantages of DFE include the ability to make bit-by-bit corrections to the pulse response, as well as its ability to ignore incoming crosstalk. DFE can be used in conjunction with transmit pre-emphasis and receive linear equalization.

As shown in Figure 4-3, page 103, attenuation of high frequency components in a transmitted signal due to the low-pass characteristic of the transmission media translates to the widening of the pulse as seen at the receiver input. This can affect the signal integrity of neighboring pulses resulting in Inter-Symbol Interference (ISI). If ISI is severe, it can cause neighboring pulses to become indistinguishable, leading to bit errors. In terms of an eye diagram, ISI translates to eye closure.

Pre-emphasis can serve to shape the transmitted pulse so as to reduce ISI at the receiver, but it cannot account for ISI introduced by physical link impedance discontinuities and the resulting reflections. While pre-emphasis alone can be sufficient for lower speeds of operation, receiver equalization is needed in conjunction with pre-emphasis for robust performance at higher line rates. See "Receive Equalization."

DFE gives the Virtex-4 device the unique ability to recover the transmitted waveform in adverse conditions by mitigating the effects of ISI. Additional margin can be gained over the use of pre-emphasis with continuous-time linear receiver equalization (see "Receive Equalization"). For equalization flexibility, any combination of the three equalization techniques can be used simultaneously.

DFE operates in the discrete-time domain and counteracts the widening of the input pulse by eliminating smearing of the original pulse onto the post-cursors. The input to the DFE block shown in Figure 4-9 is connected to the output from the continuous-time linear receiver equalizer. The output from the DFE is taken right after the sampler block and goes to the CDR circuitry.

Note: The sampler is clocked creating a delay element.

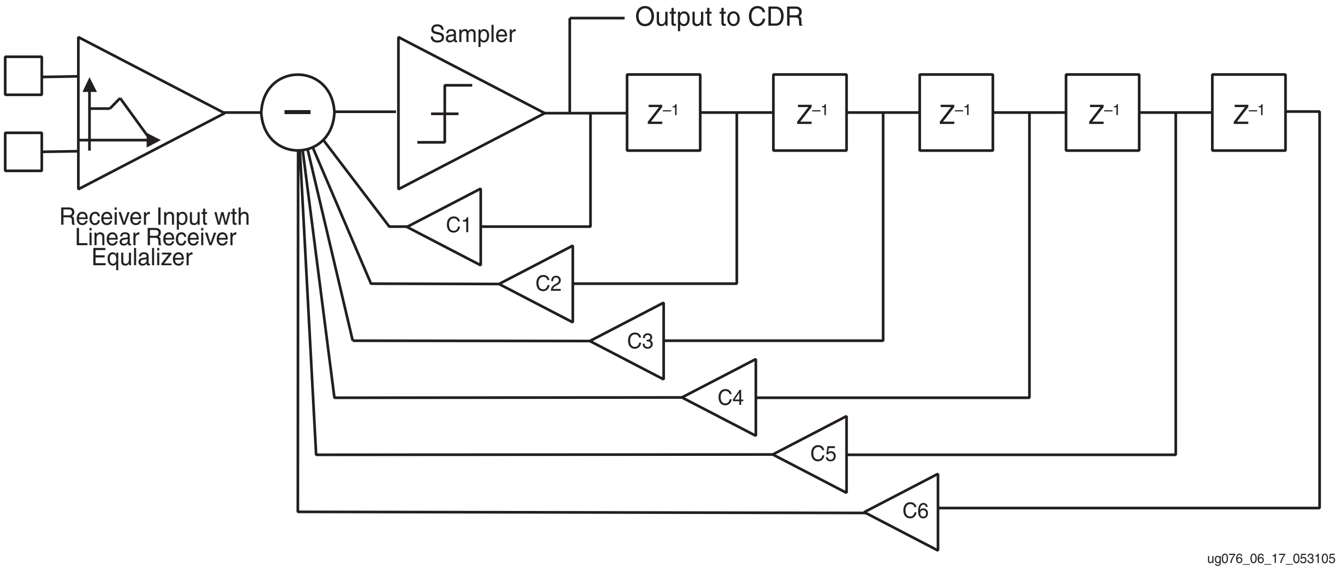


Figure 4-9: **DFE Block**

In Virtex-4 RocketIO receivers, a 12 dB limiter (non-equalizing gain stage) is added right before the sampler and is shown in Figure 4-8 as part of the sampler block. This offers significant performance improvement over the Virtex-II Pro X receivers, which do not have this limiter feature.

The sampler block and each of the delay blocks provide 1 UI of delay. The feedback paths all go through their respective programmable gain stage. Here, each tap value is multiplied by a coefficient, labeled C_1 through C_6. The outputs from the gain stages are fed into the subtraction block, which subtracts each of the weighted feedback values from the incoming analog signal.

At each time point T_0, the voltage value at the main cursor seen by the sampler is:

$$V(T_0) = V_{in}(T_0) - C_1{*}D(T_{-1}) - C_2{*}D(T_{-2}) - C_3{*}D(T_{-3}) - C_4{*}D(T_{-4}) - C_5{*}D(T_{-5}) - C_6{*}D(T_{-6})$$

where,
V_{in} is the output from the linear receiver equalizer,

C_n is the tap coefficient value,
$D(T_n)$ is the sampled data value at time point n.

As seen in Figure 4-10, a single pulse generates a negative pulse train for the next six successive UIs. This effectively cancels out the post-cursor smearing of the original pulse. By recovering the original pulse width in this manner, post-cursor ISI is reduced as a result.

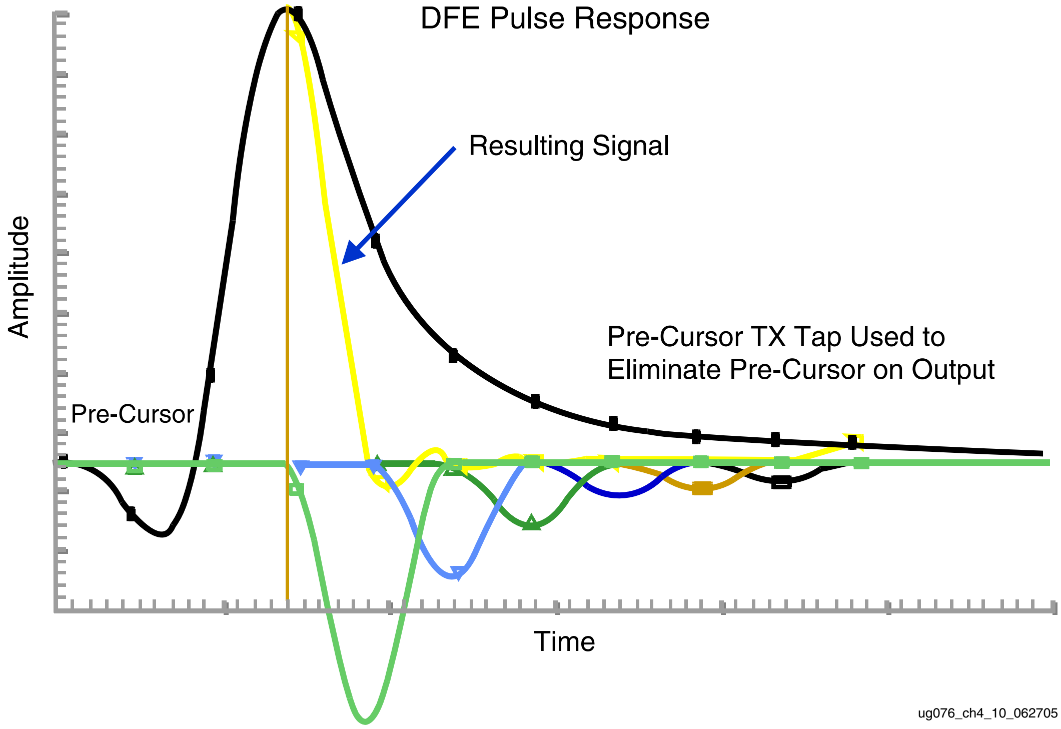


Figure 4-10: **Effect of Decision Feedback Equalization on a Received Pulse**

Another advantage of DFE is that it does not amplify signal crosstalk since the feedback is dependent only on the previously sampled data value. It also has the potential to address ripples in the pulse response due to reflections. However, this reflection ripple must occur within 6 UI of the original pulse.

The tap coefficients can be adjusted using the Dynamic Configuration Bus, allowing the DFE to optimally correct for the characteristics of a particular channel. The default values can be used as a starting point when searching for optimal values.

Tap coefficient values (shown in Table 4-4) are defined within the RXEQ[63:0] attribute.

Table 4-4: **DFE Tap Coefficients Defined**

Tap Coefficient	Location	Bit Width	Step Size	Default Value
DFE_TAP_C1	RXEQ[55:40]	16	150 µA	0000000011111111
DFE_TAP_C2	RXEQ[39:32]	8	150 µA	00000011
DFE_TAP_C3	RXEQ[31:24]	8	90 µA	00000011
DFE_TAP_C4	RXEQ[23:16]	8	90 µA	00000011

Table 4-4: **DFE Tap Coefficients Defined *(Continued)***

Tap Coefficient	Location	Bit Width	Step Size	Default Value
DFE_TAP_C5	RXEQ[15:8]	8	150 µA	`00000001`
DFE_TAP_C6	RXEQ[7:0]	8	150 µA	`00000001`

The tap coefficient values defined in the RXEQ attribute follow a thermometer code. With all tap coefficient bit positions set to zero, the corresponding DFE tap is turned off. When only the LSB is set to 1, the tap current is equal to the step size. Setting the two least significant bits to 1 brings the total tap current to two times the step size. The maximum tap current for each tap is thus defined as the bit width of the tap coefficient multiplied by the step size.

The output from the linear receive equalizer is a current-mode signal. This signal has a default value of 6.4 mA. The weight of each DFE tap fed to the subtraction block can be expressed as a percentage of this incoming current-mode signal as follows:

% Weight = Tap Current / Equalized Signal Current

= (Tap Coefficient Value x Step size) / 6.4 mA

Table 4-5 below uses the above equation to calculate the percentage weights for the default tap coefficients.

Table 4-5: **Default Tap Coefficient Expressed as a Percentage of Input Signal Current**

Tap Coefficient	Default Value	Tap Current (mA)	Weight Relative to Input Signal Current
DFE_TAP_C1	`0000000011111111`	1.20	18.75%
DFE_TAP_C2	`00000011`	0.30	4.69%
DFE_TAP_C3	`00000011`	0.18	2.81%
DFE_TAP_C4	`00000011`	0.18	2.81%
DFE_TAP_C5	`00000001`	0.15	2.34%
DFE_TAP_C6	`00000001`	0.15	2.34%

Special Analog Functions

Out-of-Band (OOB) Signals

Several protocols including Serial ATA and PCI Express implement OOB signals, which are essentially low frequency signals, compared to the normal data transmission. These are used mainly for powerdown applications and usually contain long run lengths of non-transitions. The MGT supports both the transmission and reception of these OOB signals. There are two fabric ports, RXSIGDET and TXENOOB.

TXENOOB causes the TXP/TXN pins to send an OOB signal that forces the differential output pins (TXP/TXN) to the common mode voltage resulting in a differential swing that is never more than 65mV pk-pk. This type of signal is illustrated in Figure 4-11. This signal has a direct connection to the transmit drivers which allows almost an instantaneous correlation between the assertion of TXENOOB and the TXP/TXN pins. Also the OOB signal stays on the TXP/TXN as long as the TXENOOB is set to a logic 1. When TXENOOB is set to a logic 0, the TXP/TXN pins quickly return back to normal operation.

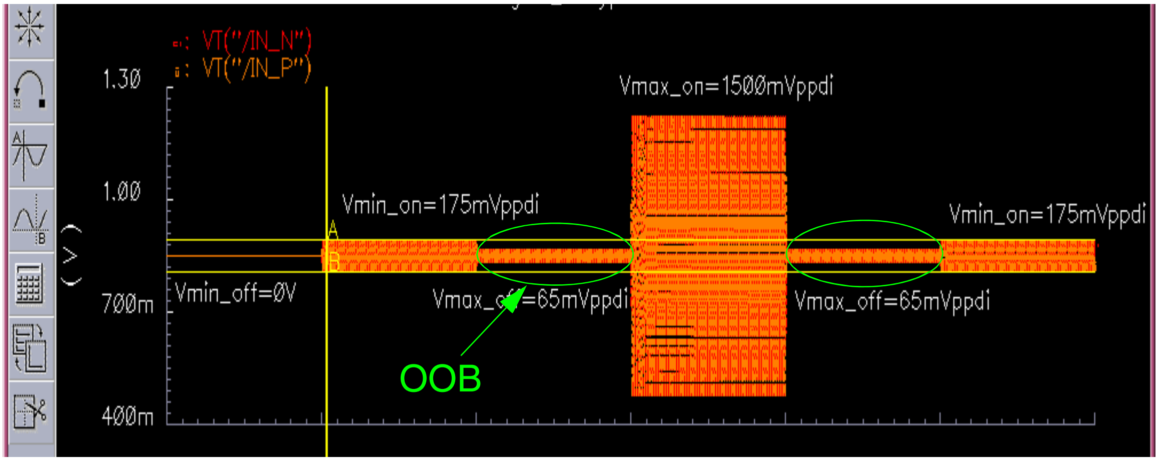


Figure 4-11: **OOB Signal**

Figure 4-12 shows how digital signals relate to the analog signal to create Out-of-Band signals, such as Beacons and Electric Idles for PCI Express and Serial ATA (SATA). Table 4-6 shows the OOB transmit and receive signal parameters.

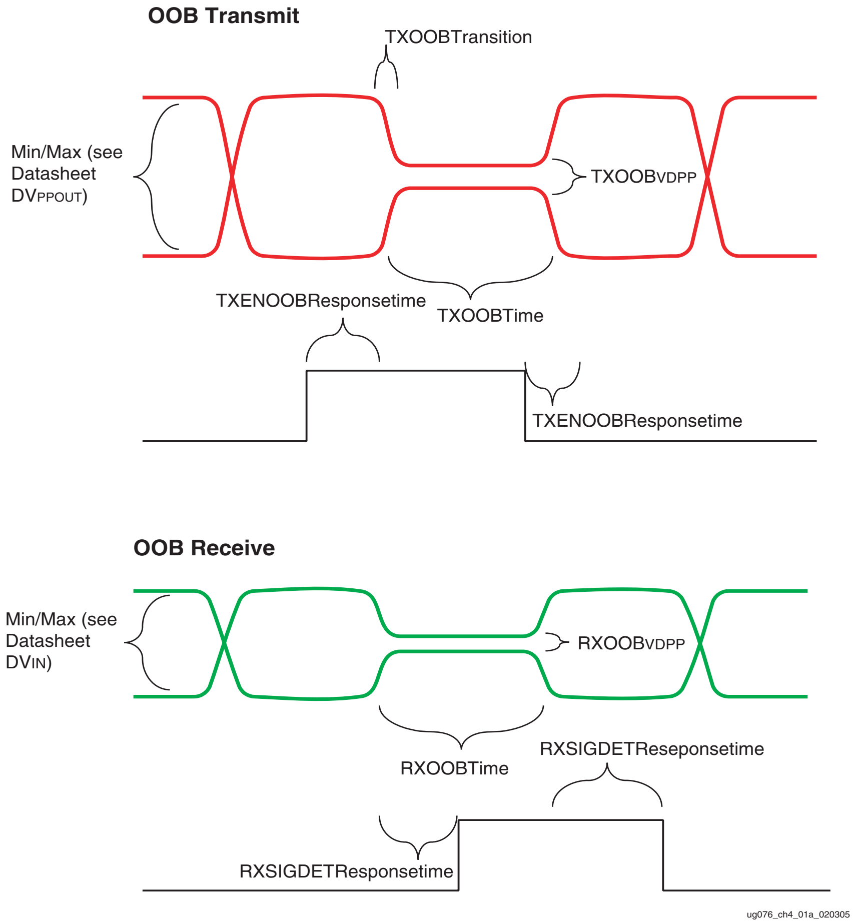


Figure 4-12: **OOB Signal Relationship**

Table 4-6: **OOB Transmit and Receive Signal Descriptions**

OOB Transmit/Receive	Min	Typ	Max	Description
TXENOOBResponsetime			10 ns	Time when TXENOOB is asserted and the signal is stable on TXP/TXN (dictates the transition from valid data to the OOB and from the OOB to valid data).
TXOOBTransition			10 ns	Time for valid data to settle to the OOB signal.
TXOOBTime				Minimum length of signal depends on how long TXOOB is asserted.[1]
TXOOBVDPP		50 mV		Differential voltage (peak-to-peak) of the TX OOB signal.
RXSIGDETResponsetime			5 ns	Time when signal is detected on RXP/RXN and the RXSIGDET is asserted (dictates the transition from valid date to the OOB and from the OOB to valid data).

Table 4-6: **OOB Transmit and Receive Signal Descriptions *(Continued)***

OOB Transmit/Receive	Min	Typ	Max	Description
RXOOBTime	100 ns			Minimum length of signal for RXSIGDET to be asserted.[1]
RXOOBVDPP	10 mV		630 mV	Threshold range which is programmable from attribute RXCDRLOS.
		10 mV		Threshold resolution.
		50 mV		Threshold accuracy.

Notes:

1. RXSIGDET and TXENOOB are asynchronous and have a 1-to-1 time relationship with the signals at RXP/RXN AND TXP/TXN.

RXSIGDET indicates when an OOB signal is detected at the RXP/RXN pins. This signal is not pipelined so there is very little latency between the OOB and the fabric indication.

RXCDRLOS is used to set the threshold level of the incoming OOB that will assert RXSIGDET. This can be set to increments of 10 mv in the range of 0-600 mV. See Table 4-7.

Table 4-7: **RXCDRLOS Values**

Binary Attribute Value	Threshold in mV
`000000`	0
`000001`	10
`000010`	20
. . .	. . .
`000110`	60
. . .	. . .
`111111`	630

Calibration for the PLLs

The MGT contains built-in circuitry to optimize the PLL performance. Table 4-8 and Table 4-9 show the settings that should be used for a specific system condition.

Table 4-8: **Transmit Calibration**

Attribute	Recommended Setting	System Conditions
FDET_HYS_CAL	010	
FDET_HYS_SEL	100	
FDET_LCK_CAL	100	
FDET_LCK_SEL	001	
VCODAC_INIT	0x000	2.488 GHz to < 3 GHz VCO frequency
	0x041	3 GHz to < 3.125 GHz VCO frequency
	0x051	3.125 GHz to < 4.25 GHz VCO frequency
	0x21F	4.25 GHz to < 5 GHz VCO frequency
	0x334	5 GHz and above VCO frequency
TXFDCAL_CLOCK_DIVIDE	NONE	Reference clock <= 250 MHz
	TWO	Reference clock > 250 MHz

Table 4-9: **Receive Calibration**

Attribute	Recommended Setting	System Conditions
RXFDET_HYS_CAL	010	
RXFDET_HYS_SEL	100	
RXFDET_LCK_CAL	100	
RXFDET_LCK_SEL	001	
RXVCODAC_INIT	0x000	2.488 GHz to < 3 GHz VCO frequency
	0x041	3 GHz to < 3.125 GHz VCO frequency
	0x051	3.125 GHz to < 4.25 GHz VCO frequency
	0x21F	4.25 GHz to < 5 GHz VCO frequency
	0x334	5 GHz and above VCO frequency
RXFDCAL_CLOCK_DIVIDE	NONE	Reference clock <= 250 MHz
	TWO	Reference clock > 250 MHz

Receiver Sample Phase Adjustment

There are situations based on the serial link that sampling the data at the center of the eye (based on the recovered clock) does not produce the best results. The MGT allows adjusting the phase of this sample point. This type of adjustment can be done dynamically, which requires that the Dynamic Reconfiguration Bus (DRB) be used in conjunction with the RXSELDACFIX[4:0] and RXSELDACTRAN[4:0] registers. These registers (whose DRB addresses are shown inTable 4-10) can control the DCR phase ±46.9%UI. Table 4-11 shows several settings and how this correlates to UI phase adjustment.

Table 4-10: **Register Address Location**

Register	MGT A Address and Bits	MGT B Address and Bits
RXSELDACFIX[4]	`0x56` bit [14]	`0x72` bit [14]
RXSELDACFIX[3:0]	`0x46` bit [14:11]	`0x62` bit [14:11]
RXSELDACTRAN[4:0]	`0x46` bit [10:6]	`0x62` bit [10:6]

Table 4-11: **Example RXSELDACTRAN and RXSELDACFIX Combinations**

RXSELDACFIX value	RXSELDACTRAN Value	Phase Adjustment (in UI%)
`11111`	`00001`	–46.875
`11110`	`00010`	–43.750
`10001`	`01111`	–3.125
`10000`	`10000`	0
`01111`	`10001`	+3.125
`00010`	`11110`	+43.750
`00001`	`11111`	+46.875

POWERDOWN

POWERDOWN, RXPD, and TXPD are used to reduce power. Table 4-12 through Table 4-14 show the definition of powerdown RXPD and TXPD.

Table 4-12: **MGT Power Control Description**

POWERDOWN	Function
`0`	PCS and PMA function normally.
`1`	PCS is in power down mode.

Table 4-13: **PMA Receiver Power Control Description**

RXPD	Function
TRUE	Powerdown the receiver.
FALSE	Normal receiver operation.

Table 4-14: **PMA Power Control Description**

TXPD	Function
TRUE	Powerdown the transmitter.
FALSE	Normal transmitter operation.

POWERDOWN is a single-bit primitive port (see Table 1-6) that allows shutting off the MGT, in case it is not needed for the design or will not be transmitting or receiving for a long period of time. When POWERDOWN is asserted, the PCS does not use any power. The PCS clocks are disabled and do not propagate through the core. During POWERDOWN, the current to TXP/TXN is turned off. The 50Ω connection to VTTX is always connected, creating the termination value. Any given MGT that is not instantiated in the design is automatically set to the POWERDOWN state by the Xilinx ISE development software and consumes no power. An instantiated MGT, however, consumes some power, even if it is not engaged in transmitting or receiving. Therefore, when an MGT is not to be used for an extended period of time, the POWERDOWN port should be asserted High to reduce overall power consumption by the Virtex-4 FPGA. Deasserting the POWERDOWN port restores the MGT to normal functional status.

RXDCCOUPLE

The attribute RXDCCOUPLE determines internal AC coupling. If set to TRUE, the internal AC coupling is bypassed. If set to FALSE, the internal AC coupling is enabled. The different coupling options are shown in Chapter 6, "Analog and Board Design Considerations."

RXPD

This attribute shuts off the clocks to the receiver to save power, which does not affect the PLL.

TXPD

This attribute shuts off the clocks to the transmitter to save power, which does not affect the PLL.

Chapter 5

Cyclic Redundancy Check (CRC)

Each MGT has two CRC-32 blocks. The input data path width can be 1 to 8 bytes and can be changed on each clock cycle to allow the computation of a CRC of any data length. The CRC initial value is the set with the attribute RXCRCINITVAL, which can be set with the Dynamic Configuration Bus. Figure 5-1 shows the basic concept of the CRC-32 block. Table 5-1 shows the CRC ports and attributes associated with the CRC-32 block.

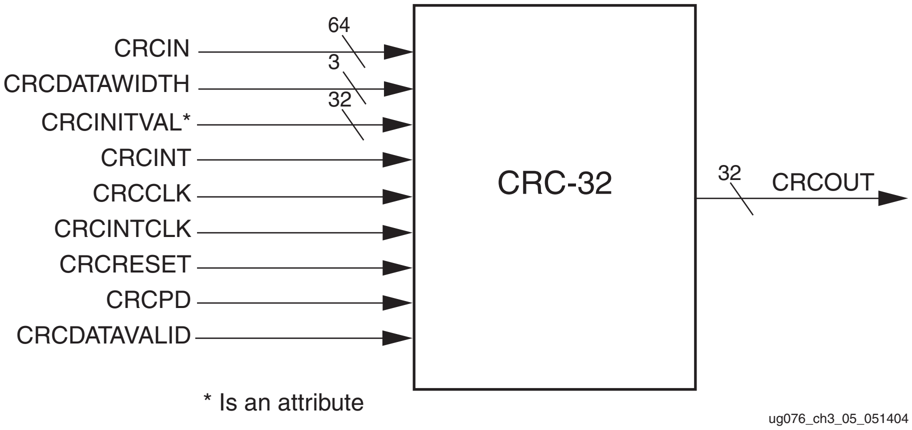


Figure 5-1: **CRC-32 Inputs and Outputs**

Table 5-1: **CRC Ports for the RX and TX CRC Blocks**

Port	I/O	Port Size	Function
RXCRCCLK	I	1	Receiver CRC logic clock.
RXCRCDATAVALID	I	1	Signals that the RXCRCIN data is valid.
RXCRCDATAWIDTH	I	3	Determines the data width of the RXCRCIN.
RXCRCIN	I	64	Receiver CRC logic input data.
RXCRCINIT	I	1	CRC initial data that the RX CRC logic uses to create the polynomial.
RXCRCINTCLK	I	1	Receiver CRC/FPGA fabric interface clock.
RXCRCOUT	O	32	Receiver CRC output data.
RXCRCPD	I	1	Indicates the RX CRC logic to powerdown when set to a logic 1.
RXCRCRESET	I	1	Resets the RX CRC logic when set to a logic 0.

Table 5-1: **CRC Ports for the RX and TX CRC Blocks *(Continued)***

Port	I/O	Port Size	Function
TXCRCCLK	I	1	Transmitter CRC logic clock.
TXCRCDATAVALID	I	1	Signals that the TXCRCIN data is valid when set to a logic 1.
TXCRCDATAWIDTH	I	3	Determines the data width of the TXCRCIN.
TXCRCIN	I	64	Transmitter CRC logic input data.
TXCRCINIT	I	1	CRC initial data that the TX CRC logic uses to create the polynomial.
TXCRCINTCLK	I	1	Transmitter CRC/FPGA fabric interface clock.
TXCRCOUT		32	Transmitter CRC output data.
TXCRCPD	I	1	Controls the TX CRC logic to power down when set to a logic 1.
TXCRCRESET	I	1	Resets the TX CRC logic when set to a logic 1.

Table 5-2: **CRC Attributes**

Attribute	Function
RXCRCINITVAL	Initializes the receiver CRC initial value.
TXCRCINITVAL	Initializes the transmitter CRC initial value.
RXCRCCLKDOUBLE	Allows ratio of 2:1 for RXCRCCLK and RXCRCINTCLK.
TXCRCCLKDOUBLE	Allows ratio of 2:1 for TXCRCCLK and TXCRCINTCLK.
RXCRCENABLE	Enables the RXCRC block. Set to FALSE to save power.
TXCRCENABLE	Enables the TXCRC block. Set to FALSE to save power.

Functionality

The CRC-32 block is a CRC generator using the following polynomial:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1.$$

An important feature of the block is its ability to work at twice the speed of the RX/TXCRCINTCLK when the fabric data width is twice the internal data width, as shown in Figure 5-2. When the attribute RXCRCCLKDOUBLE / TXCRCCLKDOUBLE is set to TRUE, the ratio between RXCRCINTCLK / TXCRCINTCLK and RXCRCCLK / TXCRCCLK frequencies are 1:2. The maximum interface data width is 64-bit. This allows a maximum supported data rate of 16 Gb/s at both 64-bit and 32-bit RXCRCIN / TXCRCIN data width, with RXCRCINTCLK / TXCRCINTCLK 250 MHz and 500 MHz, respectively. See Table 5-3 for all possible combinations of clock frequency and data widths. Data widths supported, shown in Table 5-4, are 8, 16, 24, 32, 40, 48, 56 and 64 bit. The data width can be changed by CRCDATAWIDTH at any time to support change in data rate and end of packet residue. For the different data widths, data in is always left aligned within the 64-bit input data in interface. For example, using 32-bit data width would indicate data is transmitted on data in bit range 63 to 32. See Table 5-4 for all combinations of data width field and used data in bits.

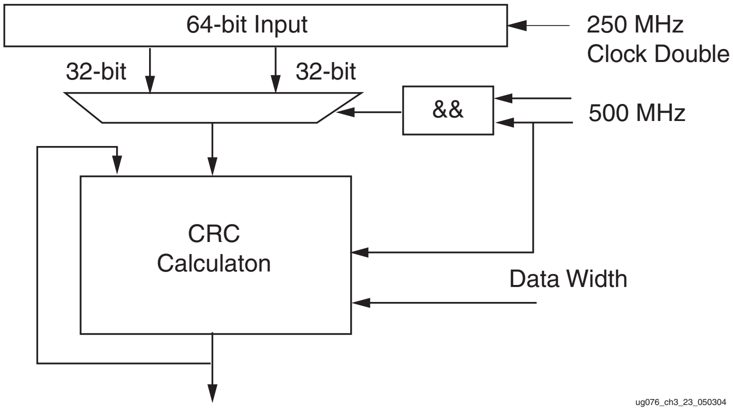


Figure 5-2: **64-Bit to 32-Bit Core Interface**

Table 5-3: **Examples of Data Rates for CRC Calculation**[1]

Data Width	CRCINTCLK	CRCCLK[3]	Data Rate	Remarks
64-bit	250 MHz	500 MHz	16 Gb/s	Max rate for CRC
56-bit	250 MHz	500 MHz	14 Gb/s	Max rate at 56-bit
48-bit	250 MHz	500 MHz	12 Gb/s	Max rate at 48-bit
40-bit	250 MHz	500 MHz	10 Gb/s	Max rate at 40-bit
32-bit	500 MHz[2]	500 MHz	16 Gb/s	Max rate for CRC
24-bit	500 MHz[2]	500 MHz	12 Gb/s	Max rate at 24-bit
16-bit	500 MHz[2]	500 MHz	8 Gb/s	Max rate at 16-bit
8-bit	500 MHz[2]	500 MHz	4 Gb/s	Max rate at 8-bit
32-bit	250 MHz	250 MHz	8 Gb/s	
24-bit	250 MHz	250 MHz	6 Gb/s	
16-bit	250 MHz	250 MHz	4 Gb/s	
8-bit	250 MHz	250 MHz	2 Gb/s	

Notes:

1. Other data rates can be achieved by changing the frequencies of operation and effective data widths.
2. The maximum speed of these configurations is determined by the fabric speed. The maximum speed is typically around 350 MHz.
3. The maximum frequency is speed grade specific.

Table 5-4: **All Combinations of Data Width and Active Data Bus Bits**

CRCDATAWIDTH	Data Width	Active Data Bus Bits
`000`	8-bit input	63:56
`001`	16-bit input	63:48
`010`	24-bit input	63:40
`011`	32-bit input	63:32
`100`	40-bit input	63:24
`101`	48-bit input	63:16
`110`	56-bit input	63:8
`111`	64-bit input	63:0

If the data width changes, as in the case of end-of-packet residue (shown in Figure 5-3), the CRCDATAWIDTH must be changed to correspond to the actual data transmitted data into the interface. Putting zeros at the unused data bits is interpreted as data, and the final CRC result will be different. The output of the CRC is transmitted to the fabric to deliver the CRC value for both the transmitter and the receiver CRC-32 blocks.

No data valid check is done. This must be done in the fabric to check the validity of the received packets. CRC is identical for both the transmitter and receiver operation. Intermediate values for the CRC field are correct CRC values, where they could be used in some standards that have multiple CRC fields.

The user must do the necessary input and output manipulation that is required by the different standard. For that purpose, the CRC data input for bit 63 is MSB and bit 0 is the LSB, and for the CRC output bit 31 is MSB and bit 0 is the LSB.

The attribute CRCINITVAL is used to initialize the CRC value for the beginning of CRC calculations; the default value is 0. CRCINIT is set to a logic 1 for one CRCINITCLK and is used to initialize the CRC with the CRCINITVAL at the beginning of each CRC calculation for every packet. CRCENABLE indicates the data on the CRCIN bus is valid. Disabling this input would keep the latest computed value of CRC output as long as it is disabled. Enabling it again would resume CRC computation. This is shown in Figure 5-3.

Table 5-5: **CRCINITVAL for Specific Protocols**

Protocol	CRCINITVAL Setting
Ethernet, PCI Express, Infiniband, Fibre Channel	`32’h FFFF_FFFF`
Serial ATA	`32’h 5232 5032`

Latency and Timing

Three CRCINTCLK cycles of latency are between the CRC and the corresponding CRCOUT. However, packets can be sent back-to-back without any separation. Both the CRCDATAVALID and CRCINIT are used to initialize the CRC for new packet calculations.

Figure 5-3, Figure 5-4, and Figure 5-5 show the cycle timing of all the ports of the CRC-32 block for 64-bit interface, 32-bit interface, and stopping a CRC that also has a residue.

64-Bit Example

Notice that valid CRCOUT data only occurs at 64-bit word boundary according to the CRCINTCLK, although internally all the CRC values at 32-bit boundary are according to the CRCCLK.

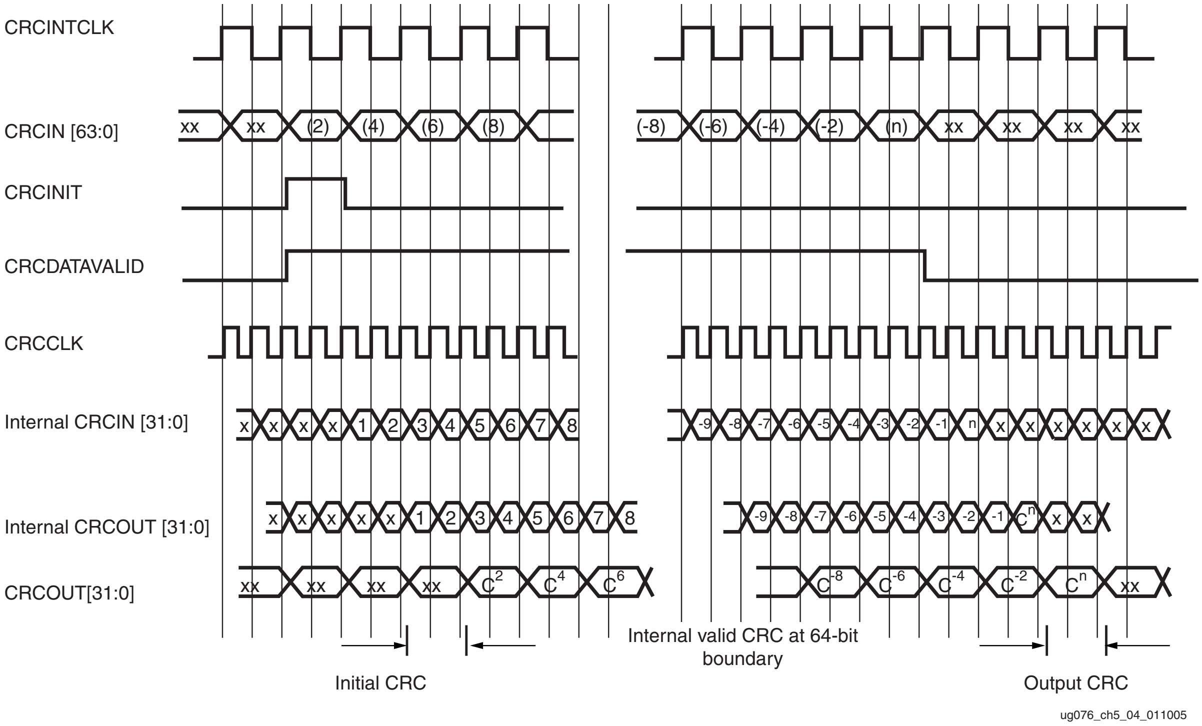


Figure 5-3: **Max Data Rate Example (64-Bit)**

32-Bit Example

The 32-bit input at 500 MHz CRCINTCLK and CRCCLK data is the same rate internally. CRCOUT rate is always the same as CRCIN rate. This is shown in Figure 5-4.

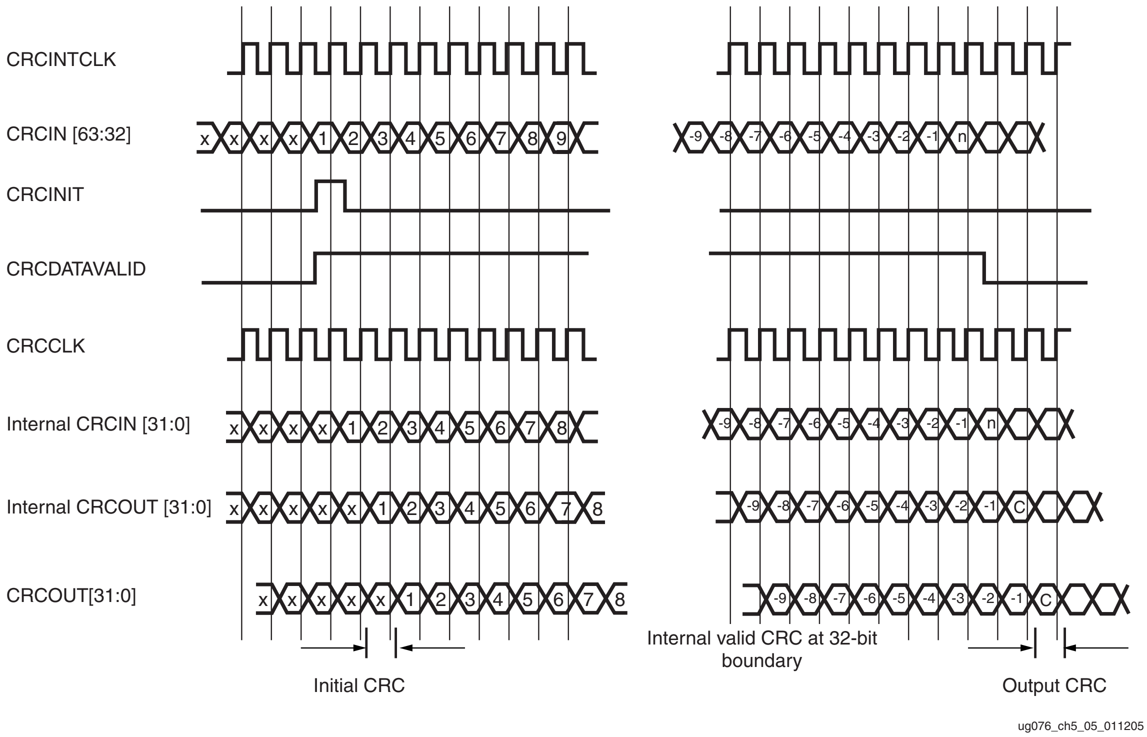


Figure 5-4: **Max Data Rate Example (32-Bit)**

16-Bit Transmission, Hold CRC, and Residue of 8-Bit Example

In this example (Figure 5-5), data is 16-bit wide. How to hold the CRC out value is shown. Also, a different residue data width is illustrated in this example. Notice that the internal data is used when there is correct data and is *don't care* otherwise. The internal CRC is the same for two consecutive CRCCLK cycles. CRCOUT is updated as always according to the interface clock. Packet length in bits has to be integer multiple of 8-bit word.

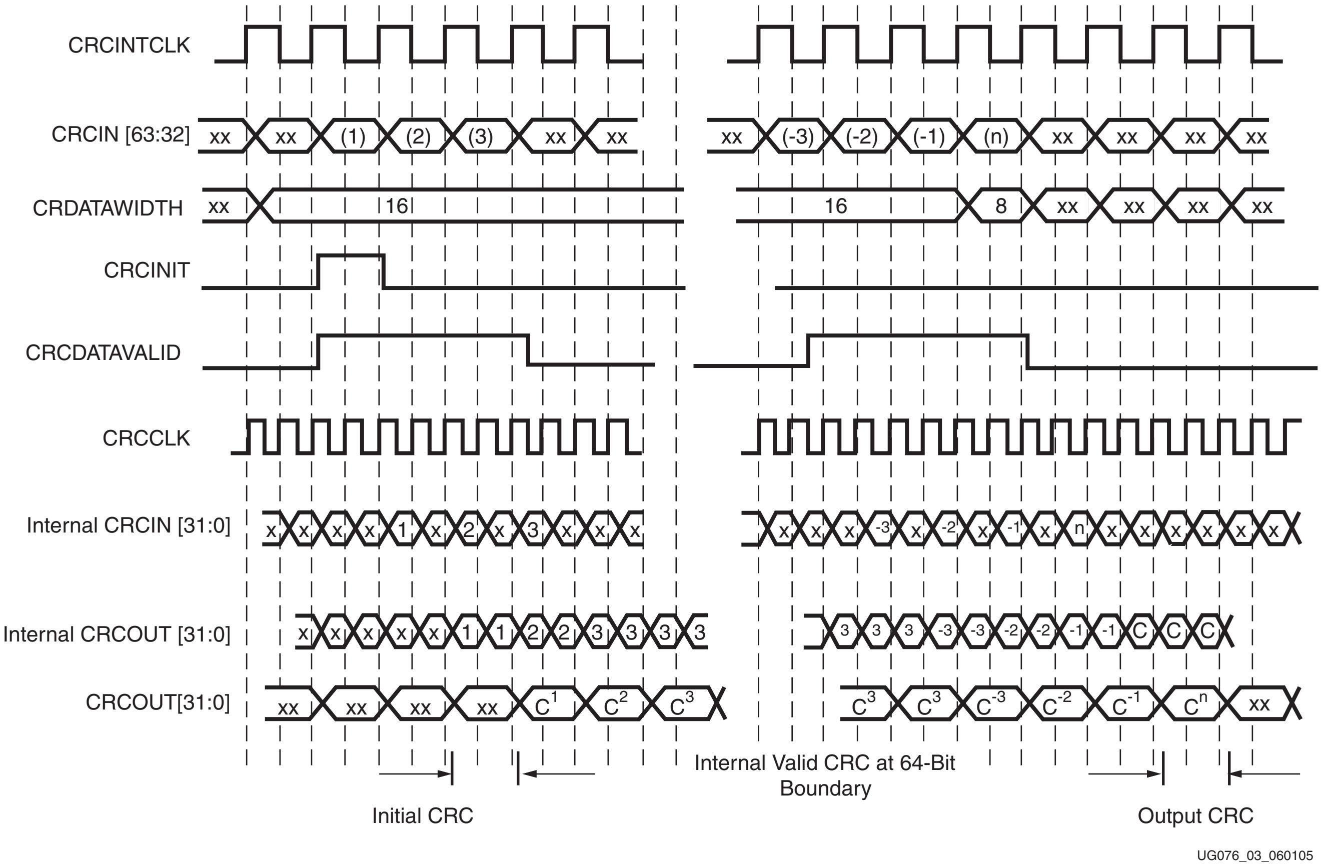


Figure 5-5: **16-Bit Transmission, Hold CRC, and Residue of 8-Bit Example**

Implementation

The CRC blocks of the MGT do not recognize start of frames (SOF/SOP) and end of frames (EOF/EOP), plus it does not indicate a CRC error. This functionality must be implemented in the FPGA fabric. Figure 5-6 shows how a state machine interacts with the MGT's CRC block to calculate a CRC value that is used to check for CRC errors.

Because of the inherent *endianess* of network protocols and CRC generation, when using the CRC blocks with the MGTs, the data must be byte swapped. This is shown in Figure 5-7 for a 2-byte interface.

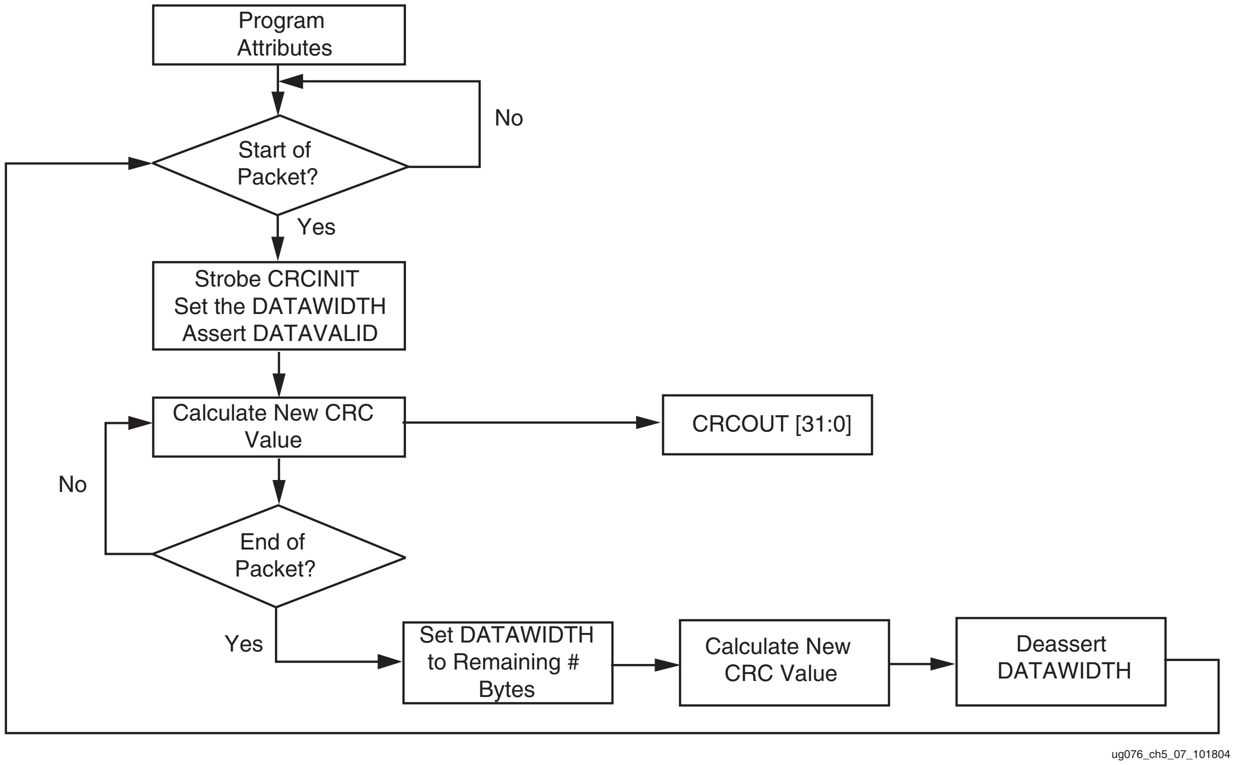


Figure 5-6: **CRC Checking Diagram**

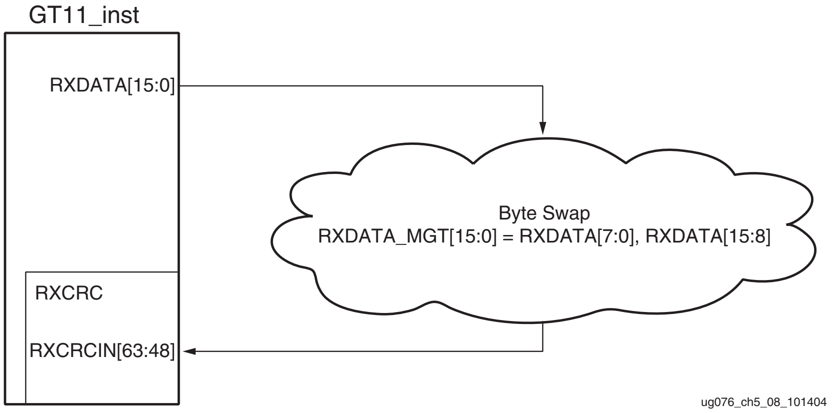


Figure 5-7: **RX CRC Byte Swap**

Chapter 6

Analog and Board Design Considerations

Physical Requirements

To ensure reliable operation of the Virtex-4 RocketIO MGT, the designer must meet certain requirements. This section outlines these requirements governing power filtering networks, reference, and clock high-speed differential signal traces. Any designs that do not adhere to these requirements are not supported by Xilinx, Inc.

Power Conditioning

Each MGT has five power supply pins (AVCCAUXTX is shared between two MGTs in a tile), all of which are extremely sensitive to noise. Table 6-1 summarizes the power supply pins, their names, associated voltages, and estimated power requirements.

To operate properly, MGTs require a certain level of noise isolation from surrounding noise sources. For this reason, it is required that both dedicated voltage regulators and passive high-frequency filtering be used to power the MGT circuitry.

Power Supply Requirements

The Virtex-4 data sheet (DS302) should be used for power consumption specifications for an MGT tile. Table 6-1 shows the typical power supply voltage and current ratings to operate an MGT tile. Each MGT tile contains two transceivers, MGT_A and MGT_B. Some applications use only a single transceiver in a tile, while others use both transceivers. The power requirements vary depending upon the end user application, baud rate, termination style, voltage level, voltage swing, and number of transceivers used in a tile. The footnotes in Table 6-1 indicate which current requirements are shared and which ones are not. Power required for AVCCAUXTX, AVCCAUXMGT, and VCCINT are shared between the two MGTs in a tile. For applications using a single transceiver in a tile, the power requirement is higher due to shared resource requirements.

- The current requirements for VCCINT should be divided by 2 to find the average current per MGT and the current for a single transceiver implementation.
- If using both transceivers in a tile, the current requirements for AVCCAUXTX and AVCCAUXMGT should be divided by 2 to find the average current per MGT.
- If using a single transceiver in a tile, 65% of the AVCCAUXTX current and 100% of the AVCCAUXMGT current must be allocated for single transceiver implementation.

The values shown in Table 6-1 do not include any design margins that would typically be used when determining voltage regulator sizing. Margins of 20 to 40 percent or higher could be used. Application specific margins should be provided depending upon operating environment and power system design above and beyond the basic requirements listed in Table 6-1.

Table 6-1: **Typical Supply Current Requirements for Two MGTs in a Tile**

Power Supply Pin Name	Typical Voltage[1] (V)	Typical Supply Current [2] (mA)			Description
		10.3 Gb/s	**6.25 Gb/s**	**622 Mb/s - 3.125 Gb/s**	
AVCCAUXTX[6]	1.2	260	225	190	Analog supply for non-I/O circuits in transmitter pair
AVCCAUXRXA	1.2	175	145	115	Analog supply for non-I/O circuits in receiver A
AVCCAUXRXB[6]	1.2	215	170	125	Analog supply for non-I/O circuits in receiver B, global bias, and reference clock circuits
VTTXA[3]	1.5	60[4]	50[4]	35[4]	Termination supply for transmitter A
VTTXB[3]	1.5	60[4]	50[4]	35[4]	Termination supply for transmitter B
VTRXA[3]	1.5	12[5]	10[5]	8[5]	Termination supply for receiver A
VTRXB[3]	1.5	12[5]	10[5]	8[5]	Termination supply for receiver B
GNDA	0.0	N/A	N/A	N/A	Analog ground for transmit and receive analog supplies
AVCCAUXMGT[6]	2.5	2	2	2	Analog supply for global bias
VCCINT[6]	1.2	225	170	110	PCS supply and Fabric Interface are connected to VCCINT in the package. There are no dedicated package pin for VCCINT in the MGT pinout.
		500 mW	420 mW	325 mW	PMA typical power
		635 mW	520 mW	390 mW	MGT typical power (PMA & PCS)

Notes:

1. See the Virtex-4 data sheet (DS302) for voltage variation specification.
2. These estimates are based on pre-silicon simulations. Does not include additional capacity and margins for DC power system design.
3. Each MGT has its own termination voltage.
4. Dependent on mode, voltage level, and swing and emphasis settings.
5. Dependent on termination style, voltage level, and swing. For floating 100Ω termination, VTRX is unconnected (0 mW). Refer to Figure 6-7.
6. AVCCAUXTX, AVCCAUXMGT, and VCCINT are shared between MGTs in a tile and should be cut in half to find approximate per MGT value. For a single transceiver implementation, use 50% of VCCINT, 65% of AVCCAUXTX, and 100% of AVCCAUXMGT. If only receiver A is used in the tile, use the difference between the AVCCAUXRXB and AVCCAUXRXA as the current for AVCCAUXRXB.

Voltage Regulation

The MGT voltage regulator circuits must not be shared with any other supplies (including FPGA supplies, V_{CCINT}, V_{CCO}, V_{CCAUX}, and V_{REF}). Voltage regulators can be shared among MGT power supplies of the same voltage; however, each supply pin must still have its own separate passive-filtering network. The regulator filtering network must be used. At this time, the Linear Technology LT1764 and LT1963 are the recommended regulators.

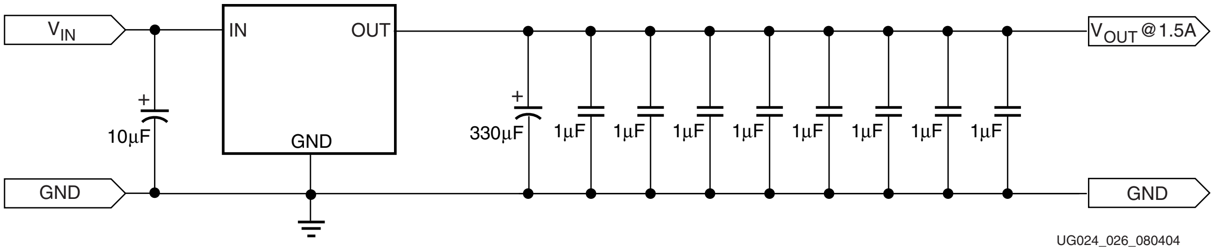


Figure 6-1: **Power Supply Circuit**

Note: Figure 6-1 shows the power supply circuit only. See Figure 6-2 for the MGT power filtering network. Figure 6-3 shows an example layout.

In most cases VTTX = VTRX, but receive termination voltage can be of any value in the range of 0.25V to 2.5V. In cases where the MGT is interfacing with a transceiver from another vendor, termination voltage can be dictated by the specifications of the other MGT. In cases where the MGT is interfacing with another MGT, a 1.5V termination voltage is recommended.

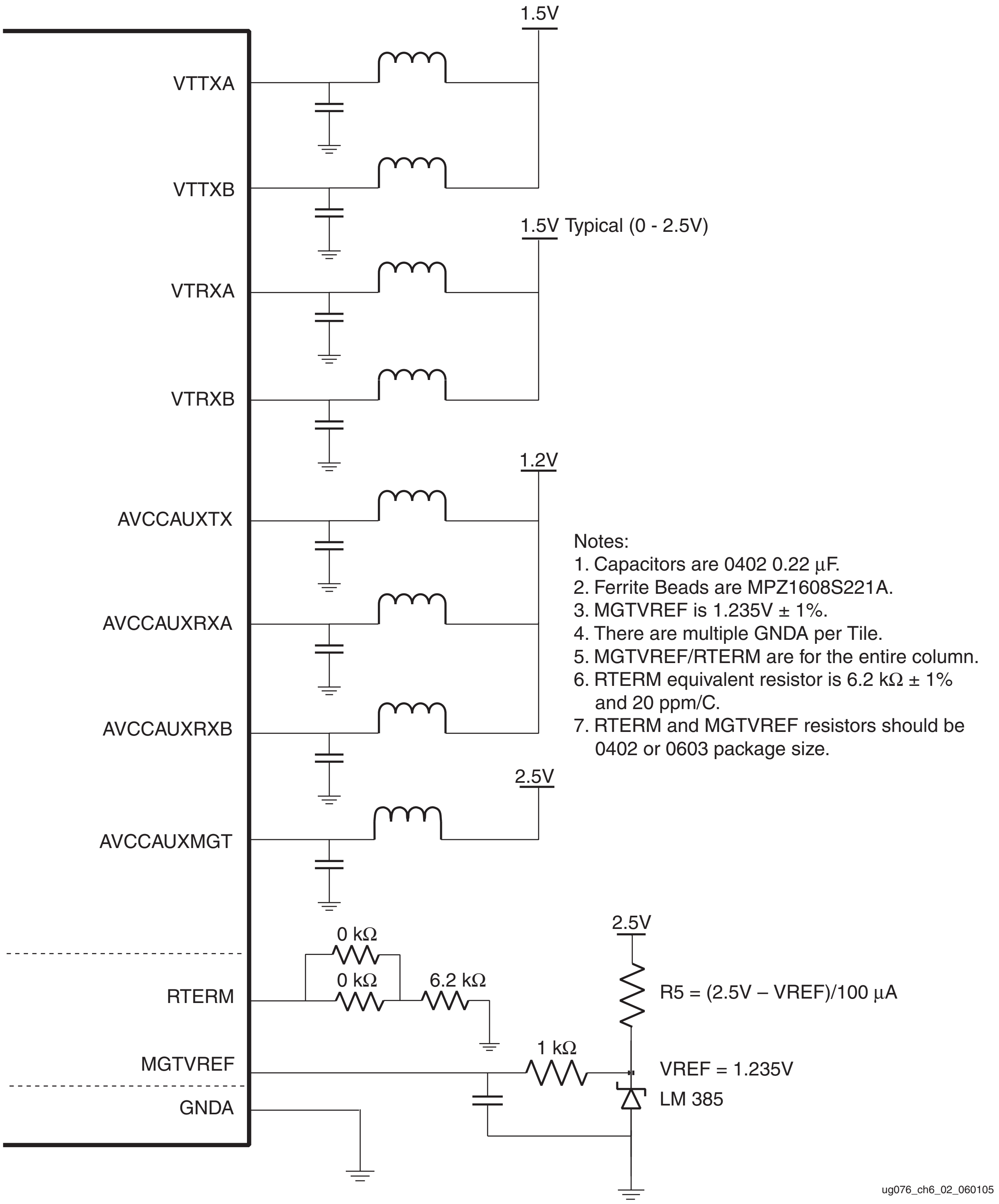
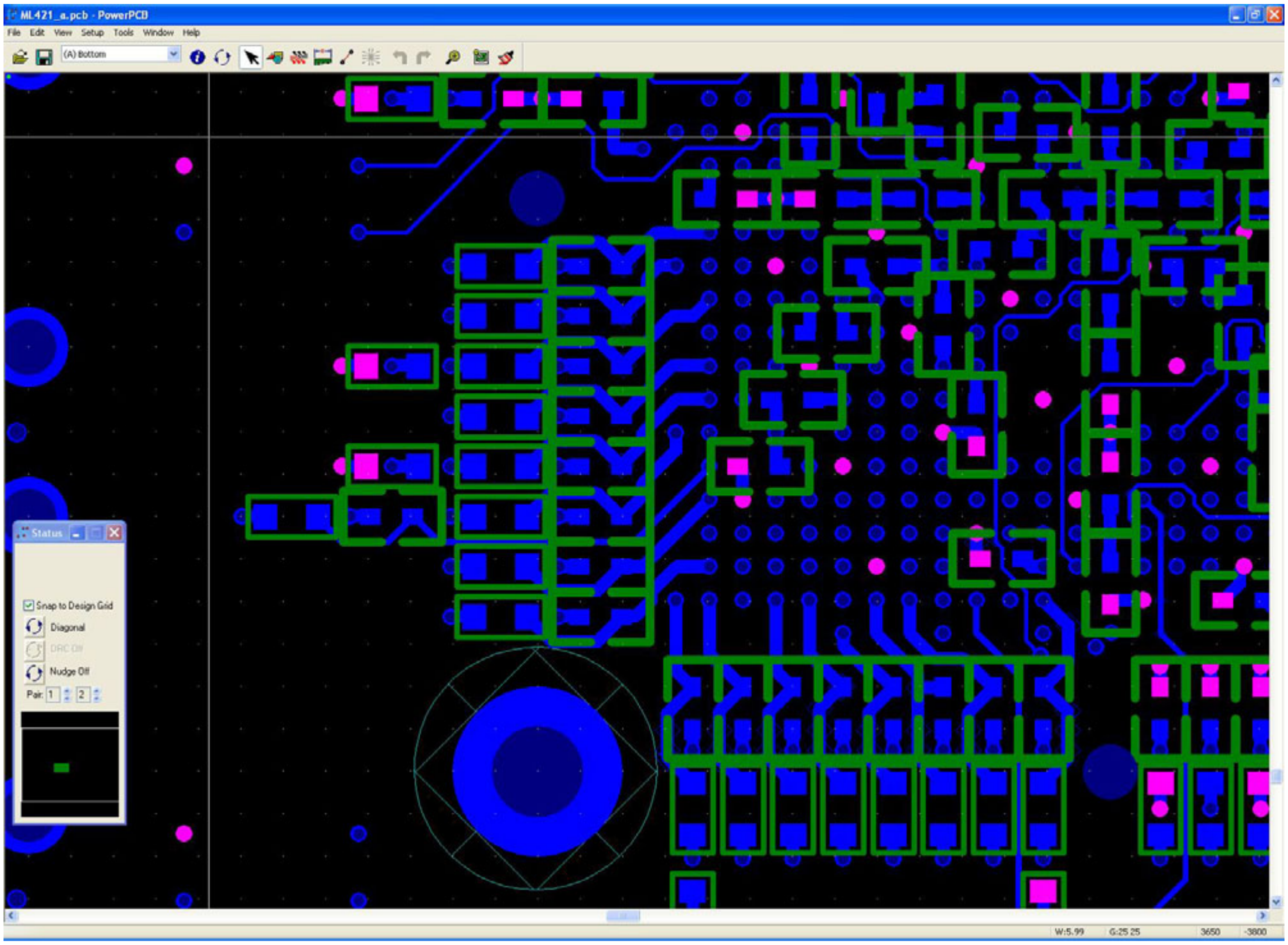

1.5V
VTTXA
VTTXB
1.5V Typical (0 - 2.5V)
VTRXA
VTRXB
1.2V
AVCCAUXTX
AVCCAUXRXA
AVCCAUXRXB
2.5V
AVCCAUXMGT
0 kΩ
0 kΩ
6.2 kΩ
RTERM
MGTVREF
GNDA
2.5V
R5 = (2.5V – VREF)/100 µA
1 kΩ
VREF = 1.235V
LM 385
Notes:
1. Capacitors are 0402 0.22 µF.
2. Ferrite Beads are MPZ1608S221A.
3. MGTVREF is 1.235V ± 1%.
4. There are multiple GNDA per Tile.
5. MGTVREF/RTERM are for the entire column.
6. RTERM equivalent resistor is 6.2 kΩ ± 1% and 20 ppm/C.
7. RTERM and MGTVREF resistors should be 0402 or 0603 package size.
ug076_ch6_02_060105

Figure 6-2: **Power Filtering Network for One MGT Tile**



ug076_ch6_03_020805

Figure 6-3: **Layout for Power Filtering Network**

Powering Unused MGTs

IMPORTANT! *All* MGTs in the FPGA, whether instantiated in the design or not, must be connected to power and ground. Unused MGTs can be powered by any 1.2V/1.5V/2.5V source, and passive filtering is not required. See Table 6-1 for more detail.

Powering Unused Transceivers

Since there are clocking resources for each MGT tile that are powered from the AVCCAUCRXB and AVCCAUXMGT, there must be careful placement of used/unused MGTs to reduce the amount of power pins that must be filtered. Several conditions require that the power supply be filtered:

- When the GT11CLK_MGT of a tile is used.
- When the SYNCLK1 or SYNCLK2 pass through a tile to reach another tile.

Figure 6-4 shows two scenarios of MGT utilization for an FX60 FF1152.

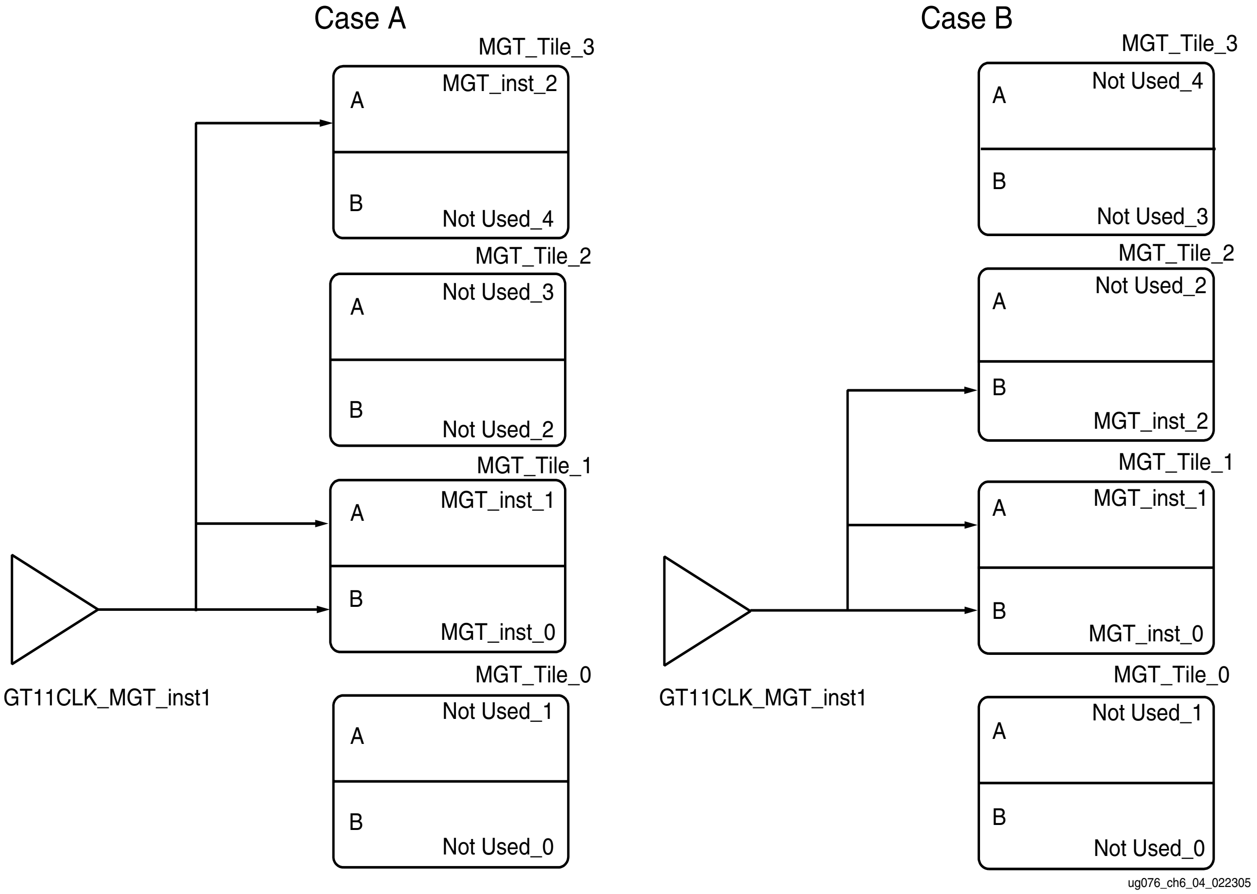


Figure 6-4: **Optimizing Filtering for an MGT Column**

Case A spreads the MGTs that are used among the column. In this case, the SYNCLK passes through MGT_TILE_2 to clock MGT_inst_2 at the top.

Case B condenses the MGTs that are used around the utilized GT11CLK_MGT_inst. This case does not have any unused tiles that have the clock passing through it, which ultimately reduces the numbers of required filtering networks.

Table 6-2 and Table 6-3 show which AVCCAUXRXB and AVCCAUXMGT must be filtered for each case shown in Figure 6-4.

Table 6-2: **Case A Filtering Requirement**

Case A Instance Name	AVCCAUCRXB	AVCCAUXMGT
MGT_inst_2	N/A	Filtering REQUIRED
not_used_4	Filtering REQUIRED	Filtering REQUIRED
not_used_3	N/A	Filtering REQUIRED
not_used_2	Filtering REQUIRED	Filtering REQUIRED
MGT_inst_1	N/A	Filtering REQUIRED
MGT_inst_0	Filtering REQUIRED	Filtering REQUIRED
not_used_1	N/A	Suggested but not required
not_used_0	Suggested but not required	Suggested but not required

Table 6-3: **Case B Filtering Requirement**

Case B Instance Name	AVCCAUCRXB	AVCCAUXMGT
not_used_4	N/A	Suggested but not required
not_used_3	Suggested but not required	Suggested but not required
not_used_2	N/A	Filtering REQUIRED
MGT_inst_2	Filtering REQUIRED	Filtering REQUIRED
MGT_inst_1	N/A	Filtering REQUIRED
MGT_inst_0	Filtering REQUIRED	Filtering REQUIRED
not_used_1	N/A	Suggested but not required
not_used_0	Suggested but not required	Suggested but not required

Reference Clock

Because a high degree of accuracy is required from the reference clock, an EPSON EG2121CA 2.5V oscillator should be used. (Visit the Epson Electronics America website for detailed information about this device.) The power supply circuit specified by the manufacturer must be used, and the circuit in Figure 6-5 must be used to interface the LVPECL outputs of the oscillator with the inputs of the transceiver reference clock.

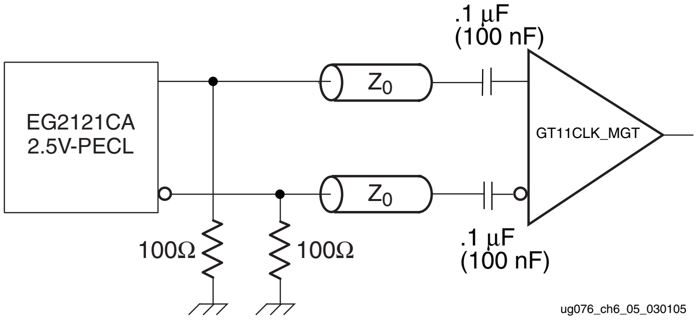


Figure 6-5: **Reference Clock Oscillator Interface (Up to 400 MHz)**

Note: Figure assumes that the Dedicated GT11CLK_MGT is used. For serial rates under 1 Gb/s, normal clock inputs can be used with appropriate termination circuitry.

The EG2121CA can be used for frequencies up to 400 MHz. For applications beyond 400 MHz (for example, 622.08 MHz, 644.53125 MHz), the EG2121CA has to be replaced with VS500, a Voltage Controlled Saw Oscillator (VCSO) (for more details, go to Vectron's website).

Note:

1. The VCSO requires a control voltage (not shown in Figure 6-6) in addition to the supply voltage.
2. The supply voltage is 3.3V and must be filtered as shown in Figure 6-6.

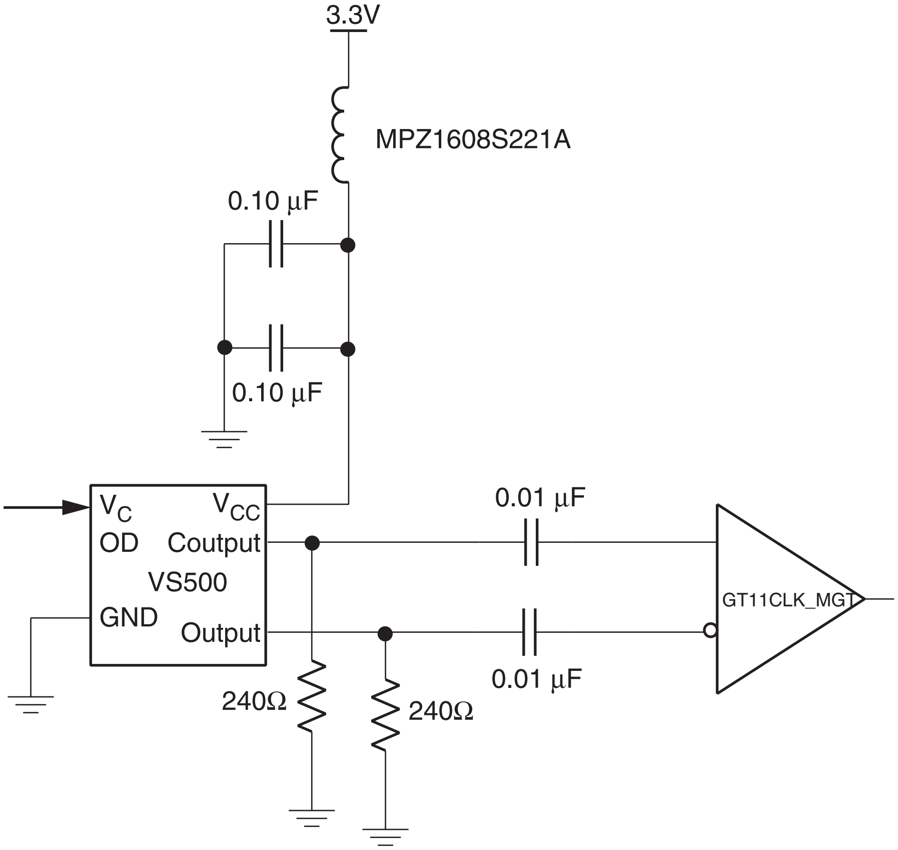


Figure 6-6: **Reference Clock VCSO**

Termination

The MGT implements on-chip 50Ω termination in both the transmitter (TXP/TXN) and receiver (RXP/RXN). The output driver and termination are powered by VTTX at 1.5V. This configuration uses a CML approach with 50Ω to TXP and TXN as shown in Figure 6-7.

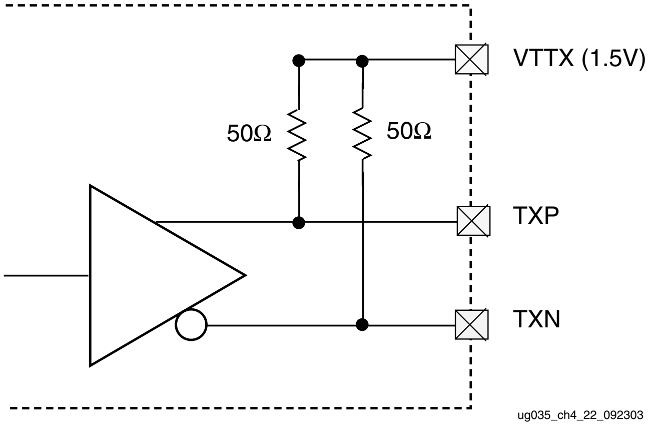


Figure 6-7: **Transmit Termination**

The receiver termination supply (VTRX) is the center tap of differential termination to RXP and RXN as shown in Figure 6-8. This supports multiple termination styles, including high side, low side, and differential (floating or active). This configuration supports receiver termination compatible to Virtex-II Pro RocketIO transceiver, using a CML (high-side) termination to an active supply of 1.8V – 2.5V. For DC coupling of two Virtex-4 devices or with a Virtex-II Pro X device, a 1.5V CML termination for VTRX is recommended.

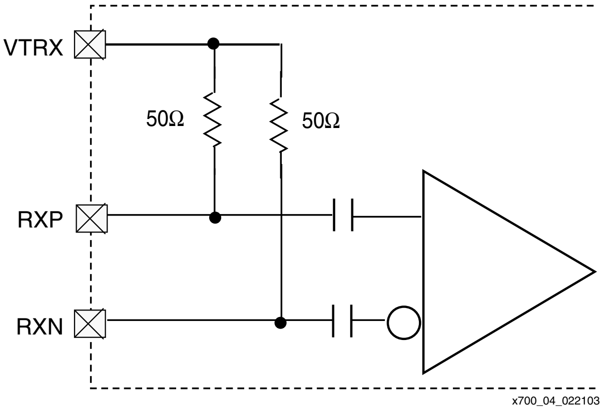


Figure 6-8: **Receive Termination**

AC and DC Coupling

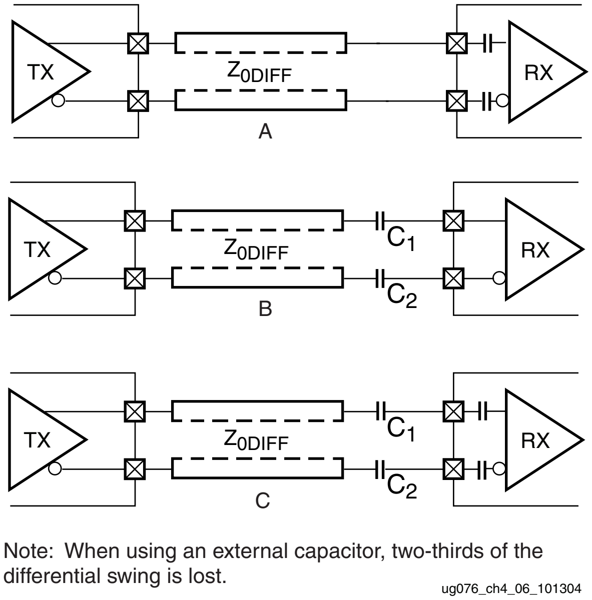
AC coupling (use of DC blocking capacitors in the signal path) should be used in cases where MGT differential voltages are compatible, but common mode voltages are not. Some designs require AC coupling to accommodate hot plug-in and/or differing power supply voltages at different MGTs or interfacing to other vendors. This is illustrated in Figure 6-9. The MGT has on-chip AC coupling caps in the receiver after the receive termination, thus supporting a wide common mode input range (0.25V – 2.5V). For common mode voltages outside of this range, external AC coupling should be used. The on-chip AC coupling supports DC-balanced data for the entire range of data rates (2.48 Gb/s - 10.3125 Gb/s). DC-balanced coding ensures that the coupling capacitor does not charge up or down, thus leaving the common mode voltage at an optimal value. This data can be coded or scrambled with a maximum run length of 72 bits to maintain on-chip PLL lock.

This AC coupling provides a high-pass filter with a corner frequency of 100 kHz. See the RXDCCOUPLE attribute in Chapter 4, “PMA Analog Design Considerations.”

Figure 6-9 shows the three possible configuration for AC coupling:

1. (Preferred for Virtex-4 transceivers) On-chip AC coupling caps used for a common mode input range of 0.25V - 2.5V.
2. External AC coupling caps used for a common mode input out of the 0.25V - 2.5V (internal caps bypassed).
3. External AC coupling caps used for a common mode input out of the 0.25V - 2.5V (internal caps enabled).

Capacitors of value 0.01 µF in a 0402 package are suitable for AC coupling up to 10.3125 Gb/s when 8B/10B or 64B/66B encoding is used. Different data rates and different encoding schemes can require a different value.



Note: When using an external capacitor, two-thirds of the differential swing is lost.

ug076_ch4_06_101304

Figure 6-9: **AC-Coupled Serial Link**

DC coupling (direct connection) should only be used when data is not DC-balanced. Passive components are not required when DC coupling is used. This is illustrated in Figure 6-10.

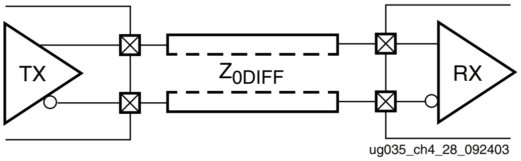


Figure 6-10: **DC-Coupled Serial Link**

High-Speed Serial Trace Design

Routing Serial Traces

All MGT I/Os are placed on the periphery of the BGA package to facilitate routing and inspection (since JTAG is not available on serial I/O pins). The MGTs have a 50Ω output/input impedance. Controlled impedance traces should be used to connect the MGT to other compatible transceivers.

When routing a differential pair, the complementary traces must be matched in length to the closest tolerance feasible. Length mismatches produce common mode noise and radiation. Severe length mismatches produce jitter and unpredictable timing problems at the receiver. Matching the differential traces to within 50 mils (1.27 mm) produces a robust design. Since signals propagate in FR4 PCB traces at approximately 180 ps per inch, a difference of 50 mils produces a timing skew of roughly 9 ps. Use SI CAD tools to confirm these assumptions on specific board designs.

All signal traces must have an intact reference plane beneath them. Stripline and microstrip geometries can be used. The reference plane should extend no less than five trace widths to either side of the trace to ensure predictable transmission line behavior.

Routing of a differential pair is optimally done in a point-to-point fashion, ideally remaining on the same PCB routing layer. As vias represent an impedance discontinuity, layer-to-layer changes should be avoided wherever possible. It is acceptable to traverse the PCB stackup to reach the transmitter and receiver package pins. If serial traces must change layers, care must be taken to ensure an intact current return path. For this reason, routing of high-speed serial traces should be on signal layers that share a reference plane. If both reference layers are DC coupled (if they are both ground), they can be connected with vias close to where the signals change layers.

To control crosstalk, serial differential traces should be spaced at least five trace separation widths from all other PCB routes, including other serial pairs. A larger spacing is required if other PCB routes carry noisy signals, such as TTL and other similarly noisy standards.

The MGT is designed to function up to 10.3125 Gb/s through 16 inches of FR4 with two high-bandwidth connectors. Longer trace lengths require use of a low-loss dielectric (for example, Rogers 4350 can essentially double the transmission distance compared to FR4). Longer traces are also supported at lower data rates; for example, data at both 3.125 and 6.25 Gb/s can be transmitted reliably over more than 46 inches of FR4 and two high-bandwidth connectors.

Differential Trace Design

The characteristic impedance of a pair of differential traces depends not only on the individual trace dimensions, but also on the spacing between them. The MGTs require a 100Ω differential trace impedance. A field solver should be used to determine the exact trace geometry suited to the specific application (Figure 6-11). This task should not be left up to the PCB vendor. Differential impedance of traces on the finished PCB should be verified with Time Domain Reflectometry (TDR) measurements.

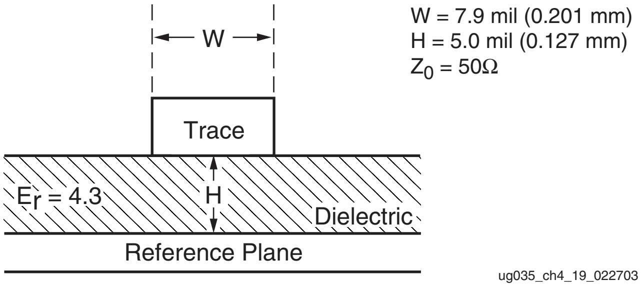


Figure 6-11: **Single-Ended Trace Geometry**

If it is necessary to separate the traces to connect to an AC coupling capacitor or connector, it can be helpful to modify the trace geometry in the vicinity of the obstacle (Figure 6-12) to correct for the impedance discontinuity (increase the individual trace width where trace separation occurs). Figure 6-13 and Figure 6-14 show examples of PCB geometries that result in 100Ω differential impedance.

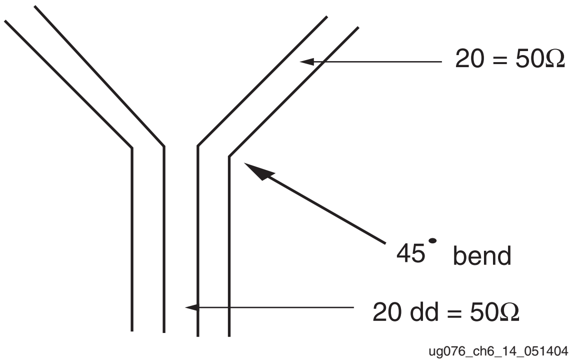


Figure 6-12: **Obstacle Route Geography**

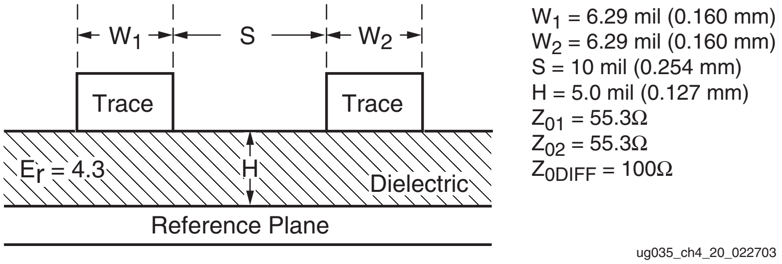


Figure 6-13: **Microstrip Edge-Coupled Differential Pair**

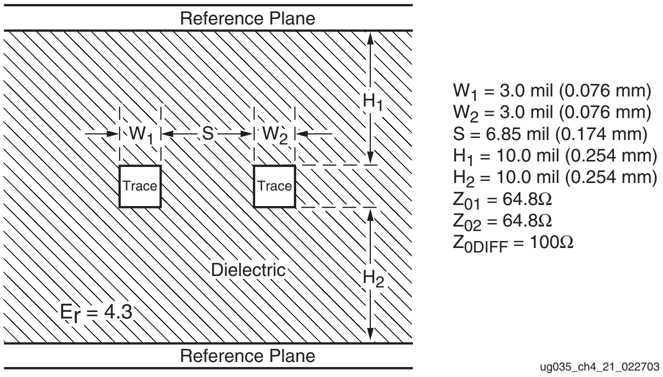


Figure 6-14: **Stripline Edge-Coupled Differential Pair**

Chapter 7

Simulation and Implementation

Model Considerations

The MGT SWIFT Model simulates the PLL in the same fashion as in the Virtex-II Pro device. To save simulation time, the PLL locks much faster than in hardware.

Since the MGT shares a TX PLL, to accurately model the functionality, two GT11 instances should be instantiated. (It is recommended this be done by the Architecture Wizard as described in "RocketIO MGT Instantiations" in Chapter 1.) The COMBUSIN/COMBUSOUT should be connected to each instance, plus the GT11_MODE should be set as shown in Table 7-1.

Table 7-1: **GT11_MODE Description**

Value	Description
SINGLE	Setting if only 1 MGT in the tile is used. (Other MGT to be set to DONT_CARE). It is suggested that the bottom MGT in a tile is used for this application.
A	Setting when both MGTs in a tile are used. (This is the top MGT) = LOC = X_x Y_{even+1}
B	Setting when both MGTs in a tile are used. (This is the bottom MGT) = LOC = X_x Y_{even}
DONT_CARE	Setting when the MGT is not used.

Simulation Models

SmartModels

SmartModels are encrypted versions of the actual HDL code. These models allow the user to simulate the actual functionality of the design without having access to the code itself. A simulator with SmartModel capability is required to use SmartModels. The models must be installed before they can be used. This can be accomplished similarly to compiling other Xilinx directories, such as UNISIMS and SIMPRIMS with the **compxlib** utility.

HSPICE

HSPICE is an analog design model that allows simulation of the RX and TX high-speed transceiver. To obtain these HSPICE models, go to the SPICE Suite Access web page at: http://support.xilinx.com/support/software/spice/spice-request.htm.

Transceiver Location and Package Pin Relation

MGT Package Pins

The MGT is a hard core placed in the FPGA fabric. All package pins for the MGTs are dedicated on the Virtex-4 FX device, as documented in the pinout tables and pinout diagrams in the ***Virtex-4 Packaging and Pinout Specification***, UG075. When creating a design, LOC constraints must be used to implement a specific MGT on the die and also determine which package pins are used. An MGT tile contains two transceivers. LOC GT11_XnYeven is associated with MGT B and LOC GT11_XnYeven+1 is associated with MGT A. Table 7-2 through Table 7-5 show the correlation between the LOC grid and the package pins themselves. The pin numbers are for TXNPAD, TXPPAD, RXNPAD, and RXPPAD, respectively. Other transceiver pins, including power, ground, and clocks, are documented in the *Virtex-4 Packaging and Pinout Specification*.

Table 7-2: **LOC Grid and Package Pins Correlation for FF672**

LOC Constraint	XC4VFX20 FF672					XC4VFX40 FF672					XC4VFX60 FF672				
	MGT	TXN	TXP	RXN	RXP	MGT	TXN	TXP	RXN	RXP	MGT	TXN	TXP	RXN	RXP
GT11_X0Y0	105B	AE26	AD26	AF23	AF24	105B	AE26	AD26	AF23	AF24	**Not Bonded Out**				
GT11_X1Y0	110B	AF5	AF4	AF8	AF7	110B	AF5	AF4	AF8	AF7	**Not Bonded Out**				
GT11_X0Y1	105A	AC26	AB26	Y26	W26	105A	AC26	AB26	Y26	W26	**Not Bonded Out**				
GT11_X1Y1	110A	AF3	AF2	AD1	AC1	110A	AF3	AF2	AD1	AC1	**Not Bonded Out**				
GT11_X0Y2	102B	A25	A24	D26	C26	103B	R26	P26	V26	U26	105B	AE26	AD26	AF23	AF24
GT11_X1Y2	113B	E1	D1	H1	G1	112B	W1	V1	AB1	AA1	110B	AF5	AF4	AF8	AF7
GT11_X0Y3	102A	A23	A22	A20	A19	103A	N26	M26	K26	J26	105A	AC26	AB26	Y26	W26
GT11_X1Y3	113A	C1	B1	A3	A4	112A	U1	T1	P1	N1	110A	AF3	AF2	AD1	AC1
GT11_X0Y4	–	–	–	–	–	102B	A25	A24	D26	C26	103B	R26	P26	V26	U26
GT11_X1Y4	–	–	–	–	–	113B	E1	D1	H1	G1	112B	W1	V1	AB1	AA1
GT11_X0Y5	–	–	–	–	–	102A	A23	A22	A20	A19	103A	N26	M26	K26	J26
GT11_X1Y5	–	–	–	–	–	113A	C1	B1	A3	A4	112A	U1	T1	P1	N1
GT11_X0Y6	–	–	–	–	–	–	–	–	–	–	102B	A25	A24	D26	C26
GT11_X1Y6	–	–	–	–	–	–	–	–	–	–	113B	E1	D1	H1	G1
GT11_X0Y7	–	–	–	–	–	–	–	–	–	–	102A	A23	A22	A20	A19
GT11_X1Y7	–	–	–	–	–	–	–	–	–	–	113A	C1	B1	A3	A4

Table 7-3: **LOC Grid and Package Pins Correlation for FF1152**

LOC Constraint	XCV4FX40 FF1152					XCV4FX60 FF1152					XCV4FX100 FF1152				
	MGT	TXN	TXP	RXN	RXP	MGT	TXN	TXP	RXN	RXP	MGT	TXN	TXP	RXN	RXP
GT11_X0Y0	105B	AM34	AL34	AP31	AP32	106B	AP20	AP21	AP17	AP18	106B	AP20	AP21	AP17	AP18
GT11_X1Y0	110B	AJ1	AH1	AM1	AL1	109B	AP12	AP11	AP15	AP14	109B	AP12	AP11	AP15	AP14
GT11_X0Y1	105A	AK34	AJ34	AG34	AF34	106A	AP22	AP23	AP25	AP26	106A	AP22	AP23	AP25	AP26
GT11_X1Y1	110A	AG1	AF1	AD1	AC1	109A	AP10	AP9	AP7	AP6	109A	AP10	AP9	AP7	AP6
GT11_X0Y2	103B	AA34	Y34	AD34	AC34	105B	AM34	AL34	AP31	AP32	105B	AM34	AL34	AP31	AP32
GT11_X1Y2	112B	V1	U1	AA1	Y1	110B	AJ1	AH1	AM1	AL1	110B	AJ1	AH1	AM1	AL1
GT11_X0Y3	103A	W34	V34	T34	R34	105A	AK34	AJ34	AG34	AF34	105A	AK34	AJ34	AG34	AF34
GT11_X1Y3	112A	T1	R1	N1	M1	110A	AG1	AF1	AD1	AC1	110A	AG1	AF1	AD1	AC1
GT11_X0Y4	102B	G34	F34	K34	J34	103B	AA34	Y34	AD34	AC34	103B	AA34	Y34	AD34	AC34
GT11_X1Y4	113B	D1	C1	G1	F1	112B	V1	U1	AA1	Y1	112B	V1	U1	AA1	Y1
GT11_X0Y5	102A	E34	D34	A32	A31	103A	W34	V34	T34	R34	103A	W34	V34	T34	R34
GT11_X1Y5	113A	A3	A4	A6	A7	112A	T1	R1	N1	M1	112A	T1	R1	N1	M1
GT11_X0Y6	–	–	–	–	–	102B	G34	F34	K34	J34	102B	G34	F34	K34	J34
GT11_X1Y6	–	–	–	–	–	113B	D1	C1	G1	F1	113B	D1	C1	G1	F1
GT11_X0Y7	–	–	–	–	–	102A	E34	D34	A32	A31	102A	E34	D34	A32	A31
GT11_X1Y7	–	–	–	–	–	113A	A3	A4	A6	A7	113A	A3	A4	A6	A7
GT11_X0Y8	–	–	–	–	–	–	–	–	–	–	101B	A26	A25	A29	A28
GT11_X1Y8	–	–	–	–	–	–	–	–	–	–	114B	A12	A13	A9	A10
GT11_X0Y9	–	–	–	–	–	–	–	–	–	–	101A	A24	A23	A21	A20
GT11_X1Y9	–	–	–	–	–	–	–	–	–	–	114A	A14	A15	A17	A18

Table 7-4: **LOC Grid and Package Pins Correlation for FF1517**

LOC Constraint	XC4VFX100 FF1517					XC4VFX140 FF1517				
	MGT	TXN	TXP	RXN	RXP	MGT	TXN	TXP	RXN	RXP
GT11_X0Y0	106B	AW24	AW25	AW21	AW22	106B	AW24	AW25	AW21	AW22
GT11_X1Y0	109B	AW16	AW15	AW19	AW18	109B	AW16	AW15	AW19	AW18
GT11_X0Y1	106A	AW27	AW28	AW30	AW31	106A	AW27	AW28	AW30	AW31
GT11_X1Y1	109A	AW13	AW12	AW10	AW9	109A	AW13	AW12	AW10	AW9
GT11_X0Y2	105B	AU39	AT39	AW36	AW37	105B	AU39	AT39	AW36	AW37
GT11_X1Y2	110B	AU1	AT1	AW4	AW3	110B	AU1	AT1	AW4	AW3
GT11_X0Y3	105A	AR39	AP39	AM39	AL39	105A	AR39	AP39	AM39	AL39
GT11_X1Y3	110A	AR1	AP1	AM1	AL1	110A	AR1	AP1	AM1	AL1
GT11_X0Y4	103B	R39	P39	V39	U39	104B	AF39	AE39	AJ39	AH39
GT11_X1Y4	112B	R1	P1	V1	U1	111B	AF1	AE1	AJ1	AH1
GT11_X0Y5	103A	N39	M39	K39	J39	104A	AD39	AC39	AA39	Y39
GT11_X1Y5	112A	N1	M1	K1	J1	111A	AD1	AC1	AA1	Y1
GT11_X0Y6	102B	A37	A36	D39	C39	103B	R39	P39	V39	U39
GT11_X1Y6	113B	A3	A4	D1	C1	112B	R1	P1	V1	U1
GT11_X0Y7	102A	A35	A34	A32	A31	103A	N39	M39	K39	J39
GT11_X1Y7	113A	A5	A6	A8	A9	112A	N1	M1	K1	J1
GT11_X0Y8	101B	A27	A26	A30	A29	102B	A37	A36	D39	C39
GT11_X1Y8	114B	A13	A14	A10	A11	113B	A3	A4	D1	C1
GT11_X0Y9	101A	A25	A24	A22	A21	102A	A35	A34	A32	A31
GT11_X1Y9	114A	A15	A16	A18	A19	113A	A5	A6	A8	A9
GT11_X0Y10	–	–	–	–	–	101B	A27	A26	A30	A29
GT11_X1Y10	–	–	–	–	–	114B	A13	A14	A10	A11
GT11_X0Y11	–	–	–	–	–	101A	A25	A24	A22	A21
GT11_X1Y11	–	–	–	–	–	114A	A15	A16	A18	A19

Table 7-5: **LOC Grid and Package Pins Correlation for FF1760**

LOC Constraint	XC4VFX140 FF1760				
	MGT	TXN	TXP	RXN	RXP
GT11_X0Y0	106B	BB36	BB37	BB33	BB34
GT11_X1Y0	109B	BB9	BB8	BB12	BB11
GT11_X0Y1	106A	BB38	BB39	AU42	AT42
GT11_X1Y1	109A	BB7	BB6	BB4	BB3
GT11_X0Y2	105B	AM42	AL42	AR42	AP42
GT11_X1Y2	110B	AP1	AN1	AU1	AT1
GT11_X0Y3	105A	AK42	AJ42	AG42	AF42
GT11_X1Y3	110A	AM1	AL1	AJ1	AH1
GT11_X0Y4	104B	AB42	AA42	AE42	AD42
GT11_X1Y4	111B	AD1	AC1	AG1	AF1
GT11_X0Y5	104A	Y42	W42	U42	T42
GT11_X1Y5	111A	AB1	AA1	W1	V1
GT11_X0Y6	103B	M42	L42	R42	P42
GT11_X1Y6	112B	P1	N1	U1	T1
GT11_X0Y7	103A	K42	J42	G42	F42
GT11_X1Y7	112A	M1	L1	J1	H1
GT11_X0Y8	102B	A37	A36	A40	A39
GT11_X1Y8	113B	A4	A5	G1	F1
GT11_X0Y9	102A	A35	A34	A32	A31
GT11_X1Y9	113A	A6	A7	A9	A10
GT11_X0Y10	101B	A27	A26	A30	A29
GT11_X1Y10	114B	A14	A15	A11	A12
GT11_X0Y11	101A	A25	A24	A22	A21
GT11_X1Y11	114A	A16	A17	A19	A20

Table 7-6: **Bonded Out MGTCLK Sources**

Device	Left Column				Right Column			
	Lower		Upper		Lower		Upper	
	CLKN	CLKP	CLKN	CLKP	CLKN	CLKP	CLKN	CLKP
XC4VFX20 FF672	AF20	AF21	G26	F26	AF11	AF10	L1	K1
XC4VFX40 FF672	AF20	AF21	G26	F26	AF11	AF10	L1	K1
XC4VFX40 FF1152	AP28	AP29	N34	M34	AP4	AP3	K1	J1
XC4VFX60 FF672	AF20	AF21	G26	F26	AF11	AF10	L1	K1
XC4VFX60 FF1152	AP28	AP29	N34	M34	AP4	AP3	K1	J1
XC4VFX100 FF1152	AP28	AP29	N34	M34	AP4	AP3	K1	J1
XC4VFX100 FF1517	AW33	AW34	G39	F39	AW7	AW6	G1	F1
XC4VFX140 FF1517	AW33	AW34	G39	F39	AW7	AW6	G1	F1
XC4VFX140 FF1760	AY42	AW42	D42	C42	AY1	AW1	D1	C1

Appendix A

RocketIO MGT Timing Model

This appendix explains the timing parameters associated with the RocketIO MGT core. It is intended to be used in conjunction with the Virtex-4 data sheet and the Timing Analyzer (TRCE) report from Xilinx software. For specific timing parameter values, refer to the data sheet.

There are many signals entering and exiting the MGT core. (Refer to Figure A-2.) The model presented in this section treats the MGT core as a "black box." Propagation delays internal to the MGT core logic are ignored. Signals are characterized with setup and hold times for inputs, and with clock to valid output times for outputs.

There are seven clocks associated with the MGT core, but only three of these clocks—RXUSRCLK, RXUSRCLK2, and TXUSRCLK2—have I/Os that are synchronous to them. The following table gives a brief description of all of these clocks. For an in-depth discussion of clocking the MGT core, refer to Chapter 2, "Clocking and Timing Considerations."

Table A-1: **MGT Clock Descriptions**

CLOCK SIGNAL	DESCRIPTION
GREFCLK	Reference clock for the MGT. Selected with attribute PMACLKBSEL. (Use only in 1 Gb/s or less applications.)
REFCLK1/ REFCLK2	High-quality reference clock driving transmission (reading TX FIFO, and multiplied for parallel/serial conversion) and clock recovery. This clock originates off the device, is routed through fabric interconnect, and is selected by the attribute PMACLKBSEL.
TXOUTCLK1	Synthesized clock from the PMA. This clock can be scaled (for example, for 64B/66B) relative to GREFCLK, depending upon the specific operating mode of the transmitter.
TXOUTCLK2	Synthesized clock from the PCS. This clock can be scaled (for example, for 64B/66B) relative to GREFCLK, depending upon the specific operating mode of the transmitter.
TXUSRCLK	Clock used for writing the TX buffer. Frequency-locked to REFCLK.
TXUSRCLK2	Clocks transmission data and status and reconfiguration data between the transceiver and the FPGA core. Relationship between TXUSRCLK2 and TXUSRCLK depends on width of transmission data path.

Table A-1: **MGT Clock Descriptions *(Continued)***

CLOCK SIGNAL	DESCRIPTION
RXRECCLK1/	Recovered clock from the CDR, locked to incoming data stream. This clock can be scaled (for example, 64B/66B) relative to incoming data rate, depending upon the specific operating mode of the receiver.
RXRECCLK1/ RXRECCLK2	Recovered clock from the PCS, locked to incoming data stream. This clock can be scaled (for example, 64B/66B) relative to incoming data rate, depending upon the specific operating mode of the receiver.
RXUSRCLK	Clock used for reading the RX elastic buffer. Clocks CHBONDI and CHBONO into and out of the transceiver. Typically the same as TXUSRCLK.
RXUSRCLK2	Clocks receiver data and status between the transceiver and the FPGA core. Typically the same as TXUSRCLK2. Relationship between RXUSRCLK2 and RXUSRCLK depends on width of receiver data path.

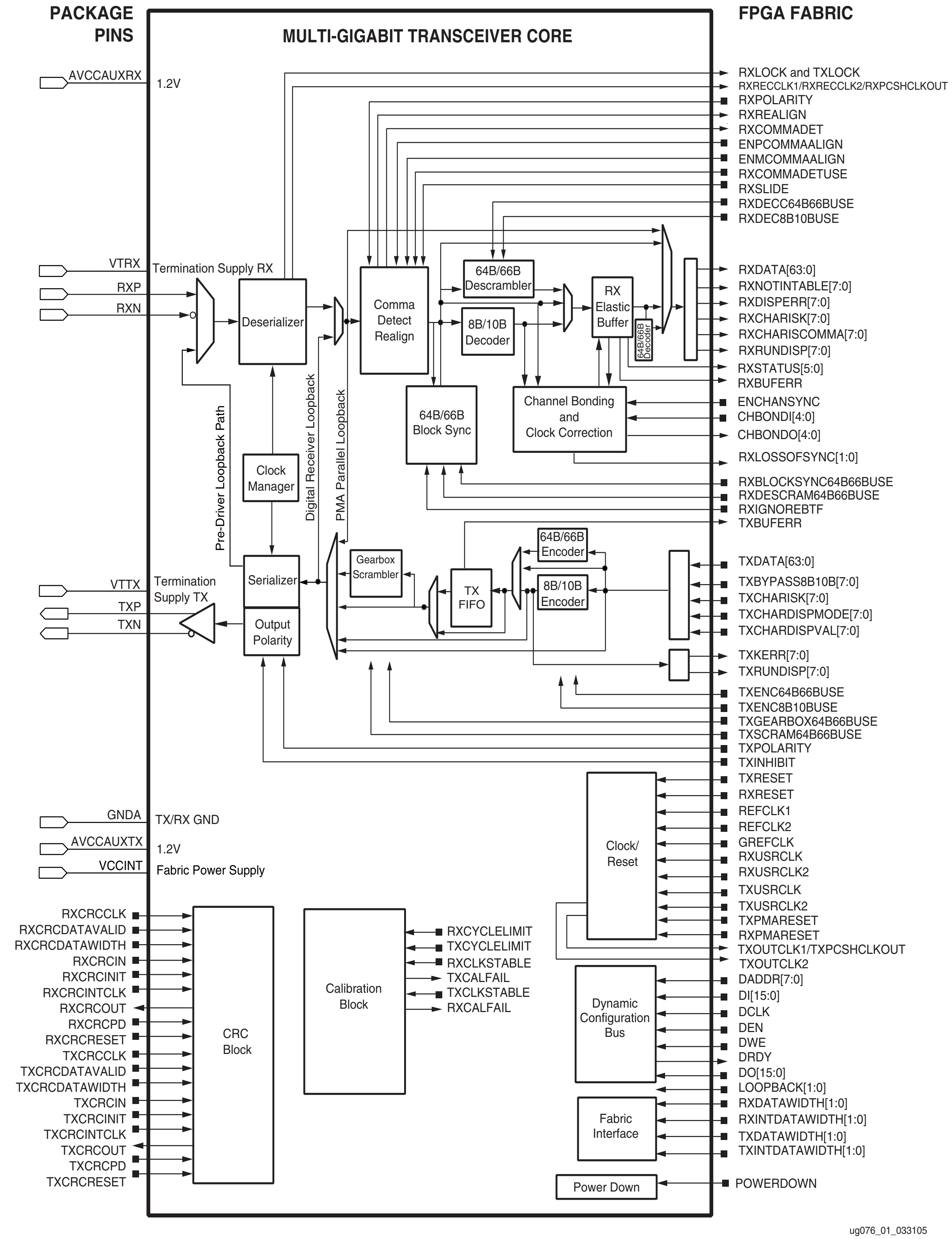


Figure A-1: **MGT Block Diagram**

Timing Parameters

Parameter designations are constructed to reflect the functions they perform, as well as the I/O signals to which they are synchronous. The following subsections explain the meaning of each of the basic timing parameter designations used in the tables.

Input Setup/Hold Times Relative to Clock

Basic Format:

ParameterName_SIGNAL
where
ParameterName = T with subscript string defining the timing relationship
SIGNAL = name of RocketIO signal synchronous to the clock
ParameterName Format:
T_{GxCK} = Setup time before clock edge
T_{GCKx} = Hold time after clock edge
where
x = C (Control inputs)
D (Data inputs)
Setup/Hold Time (Examples):
T_{GCCK}_RRST/T_{GCKC}_PLB Setup/hold times of RX Reset input relative to rising edge of RXUSRCLK2
T_{GDCK}_TDAT/T_{GCKD}_TDAT Setup/hold times of TX Data inputs relative to rising edge of TXUSRCLK2

Clock to Output Delays

Basic Format:

ParameterName_SIGNAL
where
ParameterName = T with subscript string defining the timing relationship
SIGNAL = name of RocketIO signal synchronous to the clock
ParameterName Format:
T_{GCKo} = Delay time from clock edge to output
Output Delay Time (Examples):
T_{GCKO}_CHBO Rising edge of RXUSRCLK to Channel Bond outputs
T_{GCKO}_RDAT Rising edge of RXUSRCLK2 to RX Data outputs
T_{GCKO}_TBERR Rising edge of TXUSRCLK2 to TX Buffer Err output

Clock Pulse Width

ParameterName Format:

T_{xPWH} = Minimum pulse width, High state
T_{xPWL} = Minimum pulse width, Low state
where
x = REF (REFCLK)
TX (TXUSRCLK)
TX2 (TXUSRCLK2)
RX (RXUSRCLK)
RX2 (RXUSRCLK2)

Pulse Width (Examples):

T_{TX2PWL} Minimum pulse width, TX2 clock, Low state
T_{REFPWH} Minimum pulse width, Reference clock, High state

Timing Diagram and Timing Parameter Tables

A timing diagram (Figure A-2) illustrates the timing relationships.

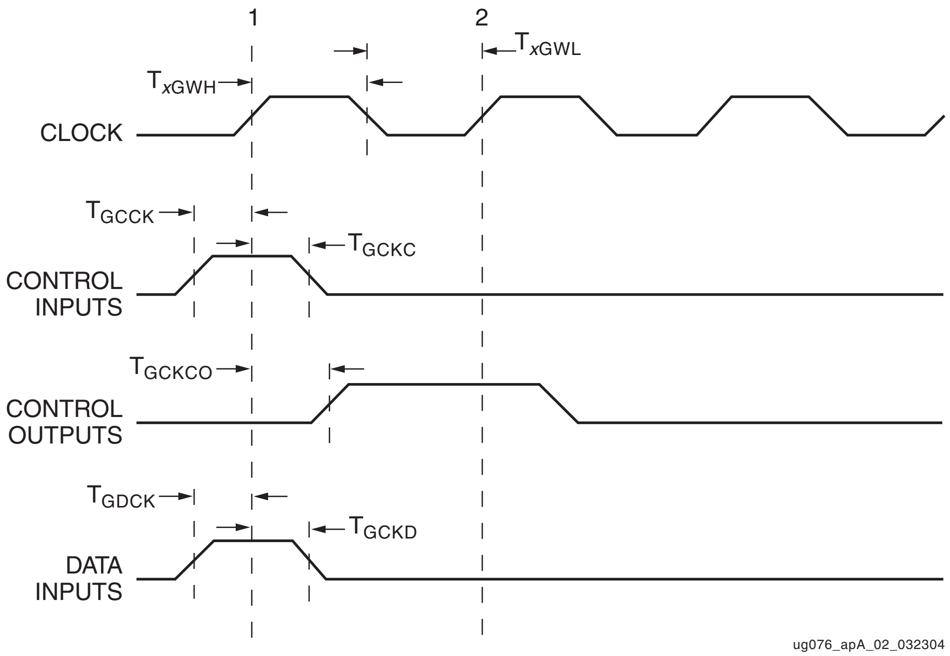


Figure A-2: **MGT Timing Relative to Clock Edge**

The following seven tables list the timing parameters as reported by the implementation tools relative to the clocks given in Table A-1, page 147, along with the MGT signals that are synchronous to each clock. (No signals are synchronous to REFCLK or TXUSRCLK.)

- Table A-2, "RocketIO DCLK Switching Characteristics," page 152
- Table A-3, "RocketIO RXCRCCLK Switching Characteristics," page 152
- Table A-4, "RocketIO TXCRCCLK Switching Characteristics," page 153
- Table A-5, "RocketIO RXUSRCLK2 Switching Characteristics," page 154
- Table A-6, "RocketIO RXUSRCLK2 Switching Characteristics," page 154
- Table A-7, "RocketIO TXUSRCLK Switching Characteristics," page 156

Table A-2: **RocketIO DCLK Switching Characteristics**

Parameter	Function	Signal
Setup and Hold Relative to Clock (DCLK)		
$T_{GT11CCK}$_DEN/ $T_{GT11CKC}$_DEN	Control Input	DEN
$T_{GT11CCK}$_DWE/ $T_{GT11CKC}$_DWE	Control Input	DWE
$T_{GT11DCK}$_DADDR/ $T_{GT11CKD}$_DADDR	Control Input	DADDR
$T_{GT11DCK}$_DI/ $T_{GT11CKD}$_DI	Data Input	DI
Clock to Out		
$T_{GT11CKO}$_DO	Data Output	DO
$T_{GT11CKO}$_DRDY	Control Output	DRDY

Table A-3: **RocketIO RXCRCCLK Switching Characteristics**

Parameter	Function	Signal
Setup and Hold Relative to Clock (RXCRCCLK)		
$T_{GT11CCK}$_RXCRCDATAVALID/ $T_{GT11CKC}$_RXCRCDATAVALID	Control Input	RXCRCDATAVALID
$T_{GT11CCK}$_RXCRCRESET/ $T_{GT11CKC}$_RXCRCRESET	Control Input	RXCRCRESET
$T_{GT11DCK}$_RXCRCDATAWIDTH/ $T_{GT11CKD}$_RXCRCDATAWIDTH	Control Input	RXCRCDATAWIDTH
$T_{GT11DCK}$_RXCRCIN/ $T_{GT11CKD}$_RXCRCIN	Control Input	RXCRCIN
$T_{GT11DCK}$_RXCRCINIT/ $T_{GT11CKD}$_RXCRCINIT	Control Input	RXCRCINIT
$T_{GT11DCK}$_RXCRCPD/ $T_{GT11CKD}$_RXCRCPD	Control Input	RXCRCPD
Clock to Out		
$T_{GT11CKO}$_RXCRCOUT	Control Output	RXCRCOUT

Table A-4: **RocketIO TXCRCCLK Switching Characteristics**

Parameter	Function	Signal
Setup and Hold Relative to Clock (TXCRCCLK)		
$T_{GT11CCK}$_TXCRCDATAVALID/ $T_{GT11CKC}$_TXCRCDATAVALID	Control Input	TXCRCDATAVALID
$T_{GT11CCK}$_TXCRCRESET/ $T_{GT11CKC}$_TXCRCRESET	Control Input	TXCRCRESET
$T_{GT11DCK}$_TXCRCDATAWIDTH/ $T_{GT11CKD}$_TXCRCDATAWIDTH	Control Input	TXCRCDATAWIDTH
$T_{GT11DCK}$_TXCRCIN/ $T_{GT11CKD}$_TXCRCIN	Control Input	TXCRCIN
$T_{GT11DCK}$_TXCRCINIT/ $T_{GT11CKD}$_TXCRCINIT	Control Input	TXCRCINIT
$T_{GT11DCK}$_TXCRCPD/ $T_{GT11CKD}$_TXCRCPD	Control Input	TXCRCPD
Clock to Out		
$T_{GT11CKO}$_TXCRCOUT	Control Output	TXCRCOUT

Table A-5: **RocketIO RXUSRCLK Switching Characteristics**

Parameter	Function	Signal
Setup and Hold Relative to Clock (RXUSRCLK)		
T_{CCCK}_CHBI/T_{CCKC}_CHBI	Control Input	CHBONDI
Clock to Out		
T_{GCKCO}_CHBO	Control Output	CHBONDO
Clock		
TGPWH_RX	Minimum Pulse Width, High	RXUSRCLK
TGPWL_RX	Minimum Pulse Width, Low	RXUSRCLK

Table A-6: **RocketIO RXUSRCLK2 Switching Characteristics**

Parameter	Function	Signal
Setup and Hold Relative to Clock		
$T_{GT11CCK}$_RXRESET/ $T_{GT11CKC}$_RXRESET	Control Input	RXRESET
$T_{GT11DCK}$_CHBONDI/ $T_{GT11CKD}$_CHBONDI	Control Input	CHBONDI
$T_{GT11DCK}$_ENCHANSYNC/ $T_{GT11CKD}$_ENCHANSYNC	Control Input	ENCHANSYNC
$T_{GT11DCK}$_ENMCOMMAALIGN/ $T_{GT11CKD}$_ENMCOMMAALIGN	Control Input	ENMCOMMAALIGN
$T_{GT11DCK}$_ENPCOMMAALIGN/ $T_{GT11CKD}$_ENPCOMMAALIGN	Control Input	ENPCOMMAALIGN
$T_{GT11DCK}$_RXBLOCKSYNC64B66BUSE/ $T_{GT11CKD}$_RXBLOCKSYNC64B66BUSE	Control Input	RXBLOCKSYNC64B66BUSE
$T_{GT11DCK}$_RXCOMMADETUSE/ $T_{GT11CKD}$_RXCOMMADETUSE	Control Input	RXCOMMADETUSE
$T_{GT11DCK}$_RXDATAWIDTH/ $T_{GT11CKD}$_RXDATAWIDTH	Control Input	RXDATAWIDTH
$T_{GT11DCK}$_RXDEC64B66BUSE/ $T_{GT11CKD}$_RXDEC64B66BUSE	Control Input	RXDEC64B66BUSE
$T_{GT11DCK}$_RXDEC8B10BUSE/ $T_{GT11CKD}$_RXDEC8B10BUSE	Control Input	RXDEC8B10BUSE
$T_{GT11DCK}$_RXDESCRAM64B66BUSE/ $T_{GT11CKD}$_RXDESCRAM64B66BUSE	Control Input	RXDESCRAM64B66BUSE
$T_{GT11DCK}$_RXIGNOREBTF/ $T_{GT11CKD}$_RXIGNOREBTF	Control Input	RXIGNOREBTF
$T_{GT11DCK}$_RXINTDATAWIDTH/ $T_{GT11CKD}$_RXINTDATAWIDTH	Control Input	RXINTDATAWIDTH
$T_{GT11DCK}$_RXPMARESET/ $T_{GT11CKD}$_RXPMARESET	Control Input	RXPMARESET
$T_{GT11DCK}$_RXPOLARITY/ $T_{GT11CKD}$_RXPOLARITY	Control Input	RXPOLARITY
$T_{GT11DCK}$_RXSLIDE/ $T_{GT11CKD}$_RXSLIDE	Control Input	RXSLIDE
$T_{GT11DCK}$_RXUSRLOCK/ $T_{GT11CKD}$_RXUSRLOCK	Control Input	RXUSRLOCK
$T_{GT11DCK}$_RXUSRVCOCAL/ $T_{GT11CKD}$_RXUSRVCOCAL	Control Input	RXUSRVCOCAL
$T_{GT11DCK}$_RXUSRVCODAC/ $T_{GT11CKD}$_RXUSRVCODAC	Control Input	RXUSRVCODAC

Table A-6: **RocketIO RXUSRCLK2 Switching Characteristics *(Continued)***

Parameter	Function	Signal
Clock to Out		
$T_{GT11CKO}$_CHBONDDONE	Status Output	CHBONDDONE
$T_{GT11CKO}$_CHBONDO	Status Output	CHBONDO
$T_{GT11CKO}$_RXBUFSTATUS	Status Output	RXBUFSTATUS
$T_{GT11CKO}$_RXCHARISCOMMA	Status Output	RXCHARISCOMMA
$T_{GT11CKO}$_RXCHARISK	Status Output	RXCHARISK
$T_{GT11CKO}$_RXCLKCORCNT	Status Output	RXCLKCORCNT
$T_{GT11CKO}$_RXCOMMADET	Status Output	RXCOMMADET
$T_{GT11CKO}$_RXCYCLELIMIT	Status Output	RXCYCLELIMIT
$T_{GT11CKO}$_RXDATA	Status Output	RXDATA
$T_{GT11CKO}$_RXDISPERR	Status Output	RXDISPERR
$T_{GT11CKO}$_RXLOCK	Status Output	RXLOCK
$T_{GT11CKO}$_RXLOCKUPDATE	Status Output	RXLOCKUPDATE
$T_{GT11CKO}$_RXLOSSOFSYNC	Status Output	RXLOSSOFSYNC
$T_{GT11CKO}$_RXNOTINTABLE	Status Output	RXNOTINTABLE
$T_{GT11CKO}$_RXREALIGN	Status Output	RXREALIGN
$T_{GT11CKO}$_RXRUNDISP	Status Output	RXRUNDISP
$T_{GT11CKO}$_RXSIGDET	Status Output	RXSIGDET
$T_{GT11CKO}$_RXVCOHIGH	Status Output	RXVCOHIGH
Clock		
T_{GPWH}_RX2	Minimum Pulse Width, High	RXUSRCLK2
T_{GPWL}_RX2	Minimum Pulse Width, Low	RXUSRCLK2

Table A-7: **RocketIO TXUSRCLK Switching Characteristics**

Parameter	Function	Signal
Setup and Hold **Relative to Clock (TXUSRCLK2)**		
$T_{GT11CCK}$_CSUPMARESET/ $T_{GT11CKC}$_CSUPMARESET	Control Input	CSUPMARESET
$T_{GT11CCK}$_TXRESET/ $T_{GT11CKC}$_TXRESET	Control Input	TXRESET
$T_{GT11DCK}$_LOOPBACK/ $T_{GT11CKD}$_LOOPBACK	Control Input	LOOPBACK
$T_{GT11DCK}$_POWERDOWN/ $T_{GT11CKD}$_POWERDOWN	Control Input	POWERDOWN
$T_{GT11DCK}$_TXBYPASS8B10B/ $T_{GT11CKD}$_TXBYPASS8B10B	Control Input	TXBYPASS8B10B
$T_{GT11DCK}$_TXCHARDISPMODE/ $T_{GT11CKD}$_TXCHARDISPMODE	Control Input	TXCHARDISPMODE
$T_{GT11DCK}$_TXCHARDISPVAL/ $T_{GT11CKD}$_TXCHARDISPVAL	Control Input	TXCHARDISPVAL
$T_{GT11DCK}$_TXCHARISK/ $T_{GT11CKD}$_TXCHARISK	Control Input	TXCHARISK
$T_{GT11DCK}$_TXDATA/ $T_{GT11CKD}$_TXDATA	Data Input	TXDATA
$T_{GT11DCK}$_TXDATAWIDTH/ $T_{GT11CKD}$_TXDATAWIDTH	Control Input	TXDATAWIDTH
$T_{GT11DCK}$_TXENC64B66BUSE/ $T_{GT11CKD}$_TXENC64B66BUSE	Control Input	TXENC64B66BUSE
$T_{GT11DCK}$_TXENC8B10BUSE/ $T_{GT11CKD}$_TXENC8B10BUSE	Control Input	TXENC8B10BUSE
$T_{GT11DCK}$_TXENOOB/ $T_{GT11CKD}$_TXENOOB	Control Input	TXENOOB
$T_{GT11DCK}$_TXGEARBOX64B66BUSE/ $T_{GT11CKD}$_TXGEARBOX64B66BUSE	Control Input	TXGEARBOX64B66BUSE
$T_{GT11DCK}$_TXINHIBIT/ $T_{GT11CKD}$_TXINHIBIT	Control Input	TXINHIBIT
$T_{GT11DCK}$_TXINTDATAWIDTH/ $T_{GT11CKD}$_TXINTDATAWIDTH	Control Input	TXINTDATAWIDTH
$T_{GT11DCK}$_TXPMARESET/ $T_{GT11CKD}$_TXPMARESET	Control Input	TXPMARESET
$T_{GT11DCK}$_TXPOLARITY/ $T_{GT11CKD}$_TXPOLARITY	Control Input	TXPOLARITY
$T_{GT11DCK}$_TXSCRAM64B66BUSE/ $T_{GT11CKD}$_TXSCRAM64B66BUSE	Control Input	TXSCRAM64B66BUSE

Table A-7: **RocketIO TXUSRCLK Switching Characteristics *(Continued)***

Parameter	Function	Signal
$T_{GT11DCK}$_TXSYNC/ $T_{GT11CKD}$_TXSYNC	Control Input	TXSYNC
$T_{GT11DCK}$_TXUSRLOCK/ $T_{GT11CKD}$_TXUSRLOCK	Control Input	TXUSRLOCK
$T_{GT11DCK}$_TXUSRVCOCAL/ $T_{GT11CKD}$_TXUSRVCOCAL	Control Input	TXUSRVCOCAL
$T_{GT11DCK}$_TXUSRVCODAC/ $T_{GT11CKD}$_TXUSRVCODAC	Control Input	TXUSRVCODAC
Clock to Out		
$T_{GT11CKO}$_CDRSTATUS	Status Output	CDRSTATUS
$T_{GT11CKO}$_RXCALFAIL	Status Output	RXCALFAIL
$T_{GT11CKO}$_TXBUFERR	Status Output	TXBUFERR
$T_{GT11CKO}$_TXCALFAIL	Status Output	TXCALFAIL
$T_{GT11CKO}$_TXCYCLELIMIT	Status Output	TXCYCLELIMIT
$T_{GT11CKO}$_TXKERR	Status Output	TXKERR
$T_{GT11CKO}$_TXLOCK	Status Output	TXLOCK
$T_{GT11CKO}$_TXLOCKUPDATE	Status Output	TXLOCKUPDATE
$T_{GT11CKO}$_TXRUNDISP	Status Output	TXRUNDISP
$T_{GT11CKO}$_TXVCOHIGH	Status Output	TXVCOHIGH
Clock		
T_{GPWH}_TX	Minimum Pulse Width, High	TXUSRCLK
T_{GPWL}_TX	Minimum Pulse Width, Low	TXUSRCLK
T_{GPWH}_TX2	Minimum Pulse Width, High	TXUSRCLK2
T_{GPWL}_TX2	Minimum Pulse Width, Low	TXUSRCLK2

Appendix B

8B/10B Valid Characters

Valid Data and Control Characters

8B/10B encoding includes a set of Data characters and K-characters. Eight-bit values are coded into 10-bit values, keeping the serial line DC balanced. K-characters are special Data characters designated with a CHARISK. K-characters are used for specific informative designations. Table B-1 and Table B-2 show the Data and K tables of valid characters.

Table B-1: **Valid Data Characters**

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
D0.0	000 00000	100111 0100	011000 1011
D1.0	000 00001	011101 0100	100010 1011
D2.0	000 00010	101101 0100	010010 1011
D3.0	000 00011	110001 1011	110001 0100
D4.0	000 00100	110101 0100	001010 1011
D5.0	000 00101	101001 1011	101001 0100
D6.0	000 00110	011001 1011	011001 0100
D7.0	000 00111	111000 1011	000111 0100
D8.0	000 01000	111001 0100	000110 1011
D9.0	000 01001	100101 1011	100101 0100
D10.0	000 01010	010101 1011	010101 0100
D11.0	000 01011	110100 1011	110100 0100
D12.0	000 01100	001101 1011	001101 0100
D13.0	000 01101	101100 1011	101100 0100
D14.0	000 01110	011100 1011	011100 0100
D15.0	000 01111	010111 0100	101000 1011
D16.0	000 10000	011011 0100	100100 1011
D17.0	000 10001	100011 1011	100011 0100
D18.0	000 10010	010011 1011	010011 0100
D19.0	000 10011	110010 1011	110010 0100
D20.0	000 10100	001011 1011	001011 0100

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits	Current RD –	Current RD +
	HGF EDCBA	**abcdei fghj**	**abcdei fghj**
D21.0	000 10101	101010 1011	101010 0100
D22.0	000 10110	011010 1011	011010 0100
D23.0	000 10111	111010 0100	000101 1011
D24.0	000 11000	110011 0100	001100 1011
D25.0	000 11001	100110 1011	100110 0100
D26.0	000 11010	010110 1011	010110 0100
D27.0	000 11011	110110 0100	001001 1011
D28.0	000 11100	001110 1011	001110 0100
D29.0	000 11101	101110 0100	010001 1011
D30.0	000 11110	011110 0100	100001 1011
D31.0	000 11111	101011 0100	010100 1011
D0.1	001 00000	100111 1001	011000 1001
D1.1	001 00001	011101 1001	100010 1001
D2.1	001 00010	101101 1001	010010 1001
D3.1	001 00011	110001 1001	110001 1001
D4.1	001 00100	110101 1001	001010 1001
D5.1	001 00101	101001 1001	101001 1001
D6.1	001 00110	011001 1001	011001 1001
D7.1	001 00111	111000 1001	000111 1001
D8.1	001 01000	111001 1001	000110 1001
D9.1	001 01001	100101 1001	100101 1001
D10.1	001 01010	010101 1001	010101 1001
D11.1	001 01011	110100 1001	110100 1001
D12.1	001 01100	001101 1001	001101 1001
D13.1	001 01101	101100 1001	101100 1001
D14.1	001 01110	011100 1001	011100 1001
D15.1	001 01111	010111 1001	101000 1001
D16.1	001 10000	011011 1001	100100 1001
D17.1	001 10001	100011 1001	100011 1001
D18.1	001 10010	010011 1001	010011 1001
D19.1	001 10011	110010 1001	110010 1001
D20.1	001 10100	001011 1001	001011 1001
D21.1	001 10101	101010 1001	101010 1001

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits `HGF EDCBA`	Current RD – `abcdei fghj`	Current RD + `abcdei fghj`
D22.1	001 10110	011010 1001	011010 1001
D23.1	001 10111	111010 1001	000101 1001
D24.1	001 11000	110011 1001	001100 1001
D25.1	001 11001	100110 1001	100110 1001
D26.1	001 11010	010110 1001	010110 1001
D27.1	001 11011	110110 1001	001001 1001
D28.1	001 11100	001110 1001	001110 1001
D29.1	001 11101	101110 1001	010001 1001
D30.1	001 11110	011110 1001	100001 1001
D31.1	001 11111	101011 1001	010100 1001
D0.2	010 00000	100111 0101	011000 0101
D1.2	010 00001	011101 0101	100010 0101
D2.2	010 00010	101101 0101	010010 0101
D3.2	010 00011	110001 0101	110001 0101
D4.2	010 00100	110101 0101	001010 0101
D5.2	010 00101	101001 0101	101001 0101
D6.2	010 00110	011001 0101	011001 0101
D7.2	010 00111	111000 0101	000111 0101
D8.2	010 01000	111001 0101	000110 0101
D9.2	010 01001	100101 0101	100101 0101
D10.2	010 01010	010101 0101	010101 0101
D11.2	010 01011	110100 0101	110100 0101
D12.2	010 01100	001101 0101	001101 0101
D13.2	010 01101	101100 0101	101100 0101
D14.2	010 01110	011100 0101	011100 0101
D15.2	010 01111	010111 0101	101000 0101
D16.2	010 10000	011011 0101	100100 0101
D17.2	010 10001	100011 0101	100011 0101
D18.2	010 10010	010011 0101	010011 0101
D19.2	010 10011	110010 0101	110010 0101
D20.2	010 10100	001011 0101	001011 0101
D21.2	010 10101	101010 0101	101010 0101
D22.2	010 10110	011010 0101	011010 0101

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
D23.2	010 10111	111010 0101	000101 0101
D24.2	010 11000	110011 0101	001100 0101
D25.2	010 11001	100110 0101	100110 0101
D26.2	010 11010	010110 0101	010110 0101
D27.2	010 11011	110110 0101	001001 0101
D28.2	010 11100	001110 0101	001110 0101
D29.2	010 11101	101110 0101	010001 0101
D30.2	010 11110	011110 0101	100001 0101
D31.2	010 11111	101011 0101	010100 0101
D0.3	011 00000	100111 0011	011000 1100
D1.3	011 00001	011101 0011	100010 1100
D2.3	011 00010	101101 0011	010010 1100
D3.3	011 00011	110001 1100	110001 0011
D4.3	011 00100	110101 0011	001010 1100
D5.3	011 00101	101001 1100	101001 0011
D6.3	011 00110	011001 1100	011001 0011
D7.3	011 00111	111000 1100	000111 0011
D8.3	011 01000	111001 0011	000110 1100
D9.3	011 01001	100101 1100	100101 0011
D10.3	011 01010	010101 1100	010101 0011
D11.3	011 01011	110100 1100	110100 0011
D12.3	011 01100	001101 1100	001101 0011
D13.3	011 01101	101100 1100	101100 0011
D14.3	011 01110	011100 1100	011100 0011
D15.3	011 01111	010111 0011	101000 1100
D16.3	011 10000	011011 0011	100100 1100
D17.3	011 10001	100011 1100	100011 0011
D18.3	011 10010	010011 1100	010011 0011
D19.3	011 10011	110010 1100	110010 0011
D20.3	011 10100	001011 1100	001011 0011
D21.3	011 10101	101010 1100	101010 0011
D22.3	011 10110	011010 1100	011010 0011
D23.3	011 10111	111010 0011	000101 1100

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits	Current RD –	Current RD +
	HGF EDCBA	abcdei fghj	abcdei fghj
D24.3	011 11000	110011 0011	001100 1100
D25.3	011 11001	100110 1100	100110 0011
D26.3	011 11010	010110 1100	010110 0011
D27.3	011 11011	110110 0011	001001 1100
D28.3	011 11100	001110 1100	001110 0011
D29.3	011 11101	101110 0011	010001 1100
D30.3	011 11110	011110 0011	100001 1100
D31.3	011 11111	101011 0011	010100 1100
D0.4	100 00000	100111 0010	011000 1101
D1.4	100 00001	011101 0010	100010 1101
D2.4	100 00010	101101 0010	010010 1101
D3.4	100 00011	110001 1101	110001 0010
D4.4	100 00100	110101 0010	001010 1101
D5.4	100 00101	101001 1101	101001 0010
D6.4	100 00110	011001 1101	011001 0010
D7.4	100 00111	111000 1101	000111 0010
D8.4	100 01000	111001 0010	000110 1101
D9.4	100 01001	100101 1101	100101 0010
D10.4	100 01010	010101 1101	010101 0010
D11.4	100 01011	110100 1101	110100 0010
D12.4	100 01100	001101 1101	001101 0010
D13.4	100 01101	101100 1101	101100 0010
D14.4	100 01110	011100 1101	011100 0010
D15.4	100 01111	010111 0010	101000 1101
D16.4	100 10000	011011 0010	100100 1101
D17.4	100 10001	100011 1101	100011 0010
D18.4	100 10010	010011 1101	010011 0010
D19.4	100 10011	110010 1101	110010 0010
D20.4	100 10100	001011 1101	001011 0010
D21.4	100 10101	101010 1101	101010 0010
D22.4	100 10110	011010 1101	011010 0010
D23.4	100 10111	111010 0010	000101 1101
D24.4	100 11000	110011 0010	001100 1101

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
D25.4	100 11001	100110 1101	100110 0010
D26.4	100 11010	010110 1101	010110 0010
D27.4	100 11011	110110 0010	001001 1101
D28.4	100 11100	001110 1101	001110 0010
D29.4	100 11101	101110 0010	010001 1101
D30.4	100 11110	011110 0010	100001 1101
D31.4	100 11111	101011 0010	010100 1101
D0.5	101 00000	100111 1010	011000 1010
D1.5	101 00001	011101 1010	100010 1010
D2.5	101 00010	101101 1010	010010 1010
D3.5	101 00011	110001 1010	110001 1010
D4.5	101 00100	110101 1010	001010 1010
D5.5	101 00101	101001 1010	101001 1010
D6.5	101 00110	011001 1010	011001 1010
D7.5	101 00111	111000 1010	000111 1010
D8.5	101 01000	111001 1010	000110 1010
D9.5	101 01001	100101 1010	100101 1010
D10.5	101 01010	010101 1010	010101 1010
D11.5	101 01011	110100 1010	110100 1010
D12.5	101 01100	001101 1010	001101 1010
D13.5	101 01101	101100 1010	101100 1010
D14.5	101 01110	011100 1010	011100 1010
D15.5	101 01111	010111 1010	101000 1010
D16.5	101 10000	011011 1010	100100 1010
D17.5	101 10001	100011 1010	100011 1010
D18.5	101 10010	010011 1010	010011 1010
D19.5	101 10011	110010 1010	110010 1010
D20.5	101 10100	001011 1010	001011 1010
D21.5	101 10101	101010 1010	101010 1010
D22.5	101 10110	011010 1010	011010 1010
D23.5	101 10111	111010 1010	000101 1010
D24.5	101 11000	110011 1010	001100 1010
D25.5	101 11001	100110 1010	100110 1010

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits `HGF EDCBA`	Current RD – `abcdei fghj`	Current RD + `abcdei fghj`
D26.5	101 11010	010110 1010	010110 1010
D27.5	101 11011	110110 1010	001001 1010
D28.5	101 11100	001110 1010	001110 1010
D29.5	101 11101	101110 1010	010001 1010
D30.5	101 11110	011110 1010	100001 1010
D31.5	101 11111	101011 1010	010100 1010
D0.6	110 00000	100111 0110	011000 0110
D1.6	110 00001	011101 0110	100010 0110
D2.6	110 00010	101101 0110	010010 0110
D3.6	110 00011	110001 0110	110001 0110
D4.6	110 00100	110101 0110	001010 0110
D5.6	110 00101	101001 0110	101001 0110
D6.6	110 00110	011001 0110	011001 0110
D7.6	110 00111	111000 0110	000111 0110
D8.6	110 01000	111001 0110	000110 0110
D9.6	110 01001	100101 0110	100101 0110
D10.6	110 01010	010101 0110	010101 0110
D11.6	110 01011	110100 0110	110100 0110
D12.6	110 01100	001101 0110	001101 0110
D13.6	110 01101	101100 0110	101100 0110
D14.6	110 01110	011100 0110	011100 0110
D15.6	110 01111	010111 0110	101000 0110
D16.6	110 10000	011011 0110	100100 0110
D17.6	110 10001	100011 0110	100011 0110
D18.6	110 10010	010011 0110	010011 0110
D19.6	110 10011	110010 0110	110010 0110
D20.6	110 10100	001011 0110	001011 0110
D21.6	110 10101	101010 0110	101010 0110
D22.6	110 10110	011010 0110	011010 0110
D23.6	110 10111	111010 0110	000101 0110
D24.6	110 11000	110011 0110	001100 0110
D25.6	110 11001	100110 0110	100110 0110
D26.6	110 11010	010110 0110	010110 0110

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits **HGF EDCBA**	Current RD – **abcdei fghj**	Current RD + **abcdei fghj**
D27.6	110 11011	110110 0110	001001 0110
D28.6	110 11100	001110 0110	001110 0110
D29.6	110 11101	101110 0110	010001 0110
D30.6	110 11110	011110 0110	100001 0110
D31.6	110 11111	101011 0110	010100 0110
D0.7	111 00000	100111 0001	011000 1110
D1.7	111 00001	011101 0001	100010 1110
D2.7	111 00010	101101 0001	010010 1110
D3.7	111 00011	110001 1110	110001 0001
D4.7	111 00100	110101 0001	001010 1110
D5.7	111 00101	101001 1110	101001 0001
D6.7	111 00110	011001 1110	011001 0001
D7.7	111 00111	111000 1110	000111 0001
D8.7	111 01000	111001 0001	000110 1110
D9.7	111 01001	100101 1110	100101 0001
D10.7	111 01010	010101 1110	010101 0001
D11.7	111 01011	110100 1110	110100 1000
D12.7	111 01100	001101 1110	001101 0001
D13.7	111 01101	101100 1110	101100 1000
D14.7	111 01110	011100 1110	011100 1000
D15.7	111 01111	010111 0001	101000 1110
D16.7	111 10000	011011 0001	100100 1110
D17.7	111 10001	100011 0111	100011 0001
D18.7	111 10010	010011 0111	010011 0001
D19.7	111 10011	110010 1110	110010 0001
D20.7	111 10100	001011 0111	001011 0001
D21.7	111 10101	101010 1110	101010 0001
D22.7	111 10110	011010 1110	011010 0001
D23.7	111 10111	111010 0001	000101 1110
D24.7	111 11000	110011 0001	001100 1110
D25.7	111 11001	100110 1110	100110 0001
D26.7	111 11010	010110 1110	010110 0001
D27.7	111 11011	110110 0001	001001 1110

Table B-1: **Valid Data Characters *(Continued)***

Data Byte Name	Bits **HGF EDCBA**	Current RD – **abcdei fghj**	Current RD + **abcdei fghj**
D28.7	111 11100	001110 1110	001110 0001
D29.7	111 11101	101110 0001	010001 1110
D30.7	111 11110	011110 0001	100001 1110
D31.7	111 11111	101011 0001	010100 1110

Table B-2: **Valid Control "K" Characters**

Special Code Name	Bits **HGF EDCBA**	Current RD – **abcdei fghj**	Current RD + **abcdei fghj**
K28.0	000 11100	001111 0100	110000 1011
K28.1	001 11100	001111 1001	110000 0110
K28.2	010 11100	001111 0101	110000 1010
K28.3	011 11100	001111 0011	110000 1100
K28.4	100 11100	001111 0010	110000 1101
K28.5	101 11100	001111 1010	110000 0101
K28.6	110 11100	001111 0110	110000 1001
K28.7[1]	111 11100	001111 1000	110000 0111
K23.7	111 10111	111010 1000	000101 0111
K27.7	111 11011	110110 1000	001001 0111
K29.7	111 11101	101110 1000	010001 0111
K30.7	111 11110	011110 1000	100001 0111

Notes:

1. Used for testing and characterization only.

Appendix C

Dynamic Configuration Bus

The Virtex-4 RocketIO transceivers provide a simple, parallel programming bus for dynamically configuring the PMA and PCS attribute settings. Two transceivers share the TX PLL. This gives the end user real-time control of all of the transceiver features without the need to use partial reconfiguration or to bring out discrete control ports to the fabric for each and every attribute, which is unmanageable. This configuration bus is identical to the Dynamic Reconfiguration Bus (DRP) for other Virtex-4 primitives, such as the DCM and system monitor.

For more details of this interface, see the Virtex-4 Configuration Guide.

Note:

1. This feature is for ADVANCED USERS ONLY. Direct modification of these attributes should only be done with a thorough understanding of the capabilities, performance, and side-effects of the resulting settings. For most applications, the default attribute settings provided in the software primitives should be adequate.
2. Improper settings can cause excessive power consumption.
3. Since modification of some registers can affect the PLL performance during reconfiguration, the time to reconfigure is approximately 10 µs to 10 ms.

Interface Description

The dynamic configuration bus consists of a simple, 2-byte-wide interface. The user accessible ports are defined in Table C-1. During FPGA configuration or a PMARESET cycle, these ports have no effect on transceiver attributes.

Table C-1: **Dynamic Configuration Bus Ports**

Port	I/O	Size	Definition
DADDR	I	8	Dynamic configuration address bus
DCLK	I	1	Dynamic configuration bus clock
DEN	I	1	Dynamic configuration bus enable when set to a logic 1
DI	I	16	Dynamic configuration input data bus
DO	O	16	Dynamic configuration output data bus
DRDY	O	1	Strobe that indicates read/write cycle is complete
DWE	I	1	Dynamic configuration write enable when set to a logic 1

See the figures entitled *Write Timing with Wait States* and *Read Timing with Wait States* in the *Virtex-4 Configuration Guide*.

Memory Map

The configuration registers are 16- bit byte-wide and memory mapped based on DADDR[7:0]. The memory assignment and resistor layout for MGT A are shown in Table C-2 through Table C-14. The memory assignment and resistor layout for MGT B are shown in Table C-15 through Table C-27. Table C-28 shows the PLL configuration settings.

Table C-2: **Dynamic Configuration Memory Map: MGT A Address 40 - 44**

Bit	Address									
	40	Def	41	Def	42	Def	43	Def	44	Def
15	RXCRCINITVAL [15:0]	N/A	RESERVED	N/A	RESERVED[7:0]	N/A	RESERVED	N/A	RESERVED [14:0]	0
14							CLK_COR__8B10_DE			
13							CLK_CORRECT_USE			
12							CLK_COR_SEQ_LEN[2:0]			
11										
10										
9							CLK_COR_SEQ_DROP			
8							CLK_COR_SEQ_2_USE			
7					COMMA32		RESERVED			
6					PCOMMA_DETECT		RESERVED			
5					MCOMMA_DETECT		CLK_COR_MAX_LAT[5:0]			
4					DEC_VALID_COMMA_ONLY					
3					DEC_PCOMMA_DETECT					
2					DEC_MCOMMA_DETECT					
1					ALIGN_COMMA_WORD [1:0]					
0									TXPD	

Table C-3: **Dynamic Configuration Memory Map: MGT A Address 45 - 49**

Bit	Address									
	45	Def	46	Def	47	Def	48	Def	49	Def
15	UNUSED [15:0]	0	RXDIGRESET	0	RXEQ[15:0]	1	RXCRCINITVAL[31:16]	N/A	RESERVED	N/A
14			RXSELDACFIX [3:0]	0		0				
13				0		0				
12				0		0				
11				0		0				
10			RXSELDAC TRAN[4:0]	1		0				
9				0		0				
8				0		0				
7				0		1				
6				0		0				
5			RXLKAPD	0		0				
4			RXRSDPD	0		0				
3			RESERVED	0		0				
2				0		0				
1				0		0				
0			RXPD	0		0				

Table C-4: **Dynamic Configuration Memory Map: MGT A Address 4A - 4E**

Bit	Address									
	4A	Def	4B	Def	4C	Def(1)	4D	Def	4E	Def
15	RESERVED [24:8]	N/A	SH_INVALID_CNT _MAX[7:0]	N/A	TXDAT_PRDRV_DAC [1:0](2)	1	UNUSED [15:0]	N/A	UNUSED[1:0]	1
14						1				0
13					TXASYNCDIVIDE[0]	X			RXCDRLOS[5:0]	0
12					TXPOST_TAP_PD	1				0
11					TXPOST_TAP_DAC [4:0]	1				0
10						1				0
9						1				0
8						0				0
7			SH_CNT_MAX [7:0]			1			RXTADJ(2)	0
6					RESERVED[2:0]	0			RXDCCOUPLE	0
5						0			UNUSED	0
4					TXDAT_TAP_DAC [4:0]	1			RXLKADJ[4:0]	0
3						0				0
2						1				0
1						1				0
0						0				0

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, “Restricted/Reserved Attributes,” for details.

Table C-5: **Dynamic Configuration Memory Map: MGT A Address 4F - 53**

Bit	Address									
	4F	Def	50	Def	51	Def	52	Def	53	Def
15	RXEQ[31:16]	1	TXCRCINITVAL [15:0]	N/A	RESERVED	N/A	RESERVED [5:0]	N/A	RXFDCAL_CLOCK _DIVIDE[1:0]	N/A
14		0								
13		0							TXFDCAL_CLOCK _DIVIDE[1:0]	
12		0								
11		0							RXBY_32	
10		0							REPEATER	
9		0					COMMA_10B _MASK[9:0]		ENABLE_DCDR	
8		0							SAMPLE_8X	
7		1							DCDR_FILTER[2:0][1]	
6		0								
5		0								
4		0							RXUSRDIVISOR[4:0]	
3		0								
2		0								
1		0								

Notes:

1. Restricted. See Appendix F, “Restricted/Reserved Attributes,” for details.

Table C-6: **Dynamic Configuration Memory Map: MGT A Address 54 - 58**

Bit	Address									
	54	Def[1]	55	Def	56	Def[1]	57	Def	58	Def
15	TXPRE_TAP_DAC [2:0]	0	UNUSED [15:0]	0	UNUSED	0	RXEQ[47:32]	0	TXCRCINITVAL [31:16]	N/A
14		0		0	RXSELDACFIX [4]	1		0		
13		0		0	RXRCPADJ[2:0][2]	X		0		
12	RESERVED	0		0		X		0		
11	TXHIGHSIGNALEN[2]	1		0		X		0		
10	RESERVED	1		0	UNUSED[1:0]	1		0		
9	TXTERMTRIM [3:0]	1		0		1		0		
8		1		0	RXAFEEQ[8:0]	0		0		
7		0		0		0		1		
6		0		0		0		0		
5	TXASYNCDIVIDE[1]	X		0		0		0		
4	TXSLEWRATE[3]	1		0		0		0		
3	TXPOST_PRDRV_DAC [2:0][2]	1		0		0		0		
2		1		0		0		0		
1		1		0		0		0		
0	TXDAT_PRDRV_DAC [2][2]	1		0		0		0		

Notes:

1. The default X depends on the operation. See Table C-28, page 196e for details.
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.
3. TXSLEWRATE is default to 1 for all serial rates below 6.25 Gb/s.

Table C-7: **Dynamic Configuration Memory Map: MGT A Address 59 - 5D**

Bit	Address									
	59	Def	5A	Def	5B	Def	5C	Def[1]	5D	Def
15	RESERVED	N/A	RESERVED [212:6]	N/A	RXRECCLK1_USE_SYNC	N/A	RESERVED [4:0]	0	UNUSED [15:0]	0
14					TXOUTCLK1_USE_SYNC			0		0
13					TXCLK0_FORCE_PMACLK			0		0
12					RXCLK0_FORCE_PMACLK			0		0
11					TX_CLOCK_DIVIDER[1:0]			0		0
10							TXPRE_PRDRV _DAC[2]	1		0
9					RX_CLOCK_DIVIDER[1:0]			1		0
8								1		0
7					TXCRCENABLE		TXPRE_TAP _PD	1		0
6					TXCRCSAMECLOCK		TXPRE_TAP _DAC[4:3]	0		0
5					TXCRCINVERTGEN			0		0
4					TXCRCCLOCKDOUBLE		RESERVED	0		0
3					RXCRCENABLE		TXCLKMODE [3:0]	X		0
2					RXCRCSAMECLOCK			X		0
1					RXCRCINVERTGEN			X		0
0					RXCRCCLOCKDOUBLE			X		0

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-8: **Dynamic Configuration Memory Map: MGT A Address 5E - 62**

Bit	Address									
	5E	Def[1]	5F	Def	60	Def	61	Def	62	Def
15	RESERVED[5:0]	0	RESERVED	0	RXBYPASS_CAL[2]	N/A	MCOMMA_32B_ VALUE/ MCOMMA_10B_ VALUE [15:0]	N/A	RESERVED [15:0]	0
14		0	RXEQ[62:48]	1	RXFDET_HYS_CAL [2:0]					0
13		0		0						0
12		0		0						0
11		0		0	RXFDET_LCK_CAL [2:0]					0
10		0		0						0
9	PMA_BIT_SLIP	0		0						0
8	RXASYNCDIVIDE [1:0]	X		0	RXFDET_HYS_SEL [2:0]					0
7		X		1						0
6	RXCLKMODE [5:0]	X		0						0
5		X		0	RXFDET_LCK_SEL [2:0]					0
4		X		0						0
3		X		0						0
2		X		0	RXVCO_CTRL _ENABLE[2]					0
1		X		0	RXCYCLE_LIMIT _SEL[1:0][2]					0
0	RXLB	0		0						0

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details

Table C-9: **Dynamic Configuration Memory Map: MGT A Address 63 - 67**

Bit	Address									
	63	Def	64	Def	65	Def	66	Def	67	Def
15	CLK_COR_SEQ _2_2 [4:0]	N/A	CLK_COR_SEQ _1_2 [4:0]	N/A	RESERVED [9:0]	0	CHAN_BOND _SEQ_2_2[4:0]	N/A	CHAN_BOND _SEQ_1_2[4:0]	N/A
14						0				
13						0				
12						0				
11						0				
10	CLK_COR_SEQ _2_1 [11:0]		CLK_COR_SEQ _1_1 [11:0]			0	CHAN_BOND _SEQ_2_1 [10:0]		CHAN_BOND _SEQ_1_1 [10:0]	
9						0				
8						0				
7						0				
6						0				
5					UNUSED[4:0]	0				
4						0				
3						0				
2						0				
1						0				
0					RESERVED	0				

Table C-10: **Dynamic Configuration Memory Map: MGT A Address 68 - 6C**

Bit	Address									
	68	Def	69	Def	6A	Def[1]	6B	Def	6C	Def
15	RESERVED	N/A	MCOMMA_32B_ VALUE/ MCOMMA_10B_ VALUE[31:16]	N/A	RESERVED	0	CLK_COR_ SEQ_2_3[9:0]	N/A	CLK_COR_ SEQ_1_3[9:0]	N/A
14	RXVCODAC_INIT [9:0][3]				TXOUTDIV2 SEL_B[3:0][2]	X				
13						X				
12						X				
11						X				
10					TXCRTL[9:0][3]	1				
9						0				
8						0				
7						0				
6						0				
5						0	CLK_COR_ SEQ_2_2[10:5]		CLK_COR_ SEQ_1_2[10:5]	
4	RXSLOWDOWN_CAL [1:0][3]					0				
3						X				
2	RXBYPASS_FDET[3]					X				
1	RXLOOPCAL_WAIT [1:0][3]					0				
0					RESERVED	0				

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. TXOUTDIV2SEL_2 must equal TXOUTDIV2SEL_1. The attribute TXOUTDIV2SEL sets both TXOUTDIV2SEL_1/TXOUTDIV2SEL_2 registers upon configuration, but must be written to separately when using the Dynamic Configuration Bus. This attribute only affects the MGT B of a file.
3. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-11: **Dynamic Configuration Memory Map: MGT A Address 6D - 71**

Bit	Address									
	6D	**Def(1)**	**6E**	**Def**	**6F**	**Def**	**70**	**Def**	**71**	**Def**
15	RESERVED	0	CHAN_BOND_ SEQ_2_3[9:0]	N/A	CHAN_BOND_ SEQ_1_3[9:0]	N/A	BYPASS_CAL(3)	N/A	PCOMMA_32B_ VALUE/ PCOMMA_10B_ VALUE [15:0]	N/A
14	RXOUTDIV2 SEL_2[3:0](2)	X					FDET_HYS_CAL [2:0}			
13		X								
12		X								
11		X					FDET_LCK_CAL [2:0}			
10	UNUSED [12:0]	1								
9		0								
8		0					FDET_HYS_SEL [2:0}			
7		0								
6		0								
5		0	CHAN_BOND_ SEQ_2_2[10:5]		CHAN_BOND_ SEQ_1_2[10:5]		FDET_LCK_SEL [2:0}			
4		0								
3		X								
2		X					VCO_CTRL_ ENABLE(3)			
1		0					CYCL_LIMIT_ SEL[1:0](3)			
0	RESERVED	0								

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. RXOUTDIV2SEL_2 must equal RXOUTDEV2SEL_1. The attribute RXOUTDEIV2SEL sets both RXOUTDIV2SEL_1/RXOUTDIV2SEL_2 registers upon configuration, but must be written to separately when using the Dynamic Configuration Bus.
3. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-12: **Dynamic Configuration Memory Map: MGT A Address 72 - 76**

Bit	Address									
	72	Def[1]	73	Def	74	Def	75	Def[1]	76	Def
15	RESERVED [14:0]	0	CLK_COR_SEQ _2_MASK[3:0]	N/A	CLK_COR_SEQ _1_MASK[3:0]	N/A	RESERVED [14:0]	0	CHAN_BOND_SEQ _2_MASK[3:0]	N/A
14		0						0		
13		0						0		
12		0						0		
11		0	CLK_COR_SEQ _2_4[10:0]		CLK_COR_SEQ _1_4[10:0]			0	CHAN_BOND_SEQ _2_4[10:0]	
10		0						0		
9		0						0		
8		0						0		
7		0						0		
6		0						0		
5		0						0		
4		0						0		
3		0						0		
2		1						1		
1		1						1		
0	TXCPSEL[2]	X	CLK_COR_SEQ _2_3[10]		CLK_COR_SEQ _1-3[10]		RSCPSEL	X	CHAN_BOND_SEQ _2_3[10]	

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-13: **Dynamic Configuration Memory Map: MGT A Address 77 - 7B**

Bit	Address									
	77	Def	78	Def	79	Def	7A	Def[1]	7B	Def
15	CHAN_BOND_SEQ_1_MASK[3:0]	N/A	RESERVED	N/A	PCOMMA_32B_VALUE/ PCOMMA_10B_VALUE [31:16]	N/A	TXOUTDIV2SEL_A[3:0][2]	X	RX_LOS_THRESHOLD [7:0]	N/A
14			VCODAC_INIT [9:0][3]					X		
13								X		
12								X		
11	CHAN_BOND_SEQ_1_4[10:0]						TXPLLNDIVSEL[3:0]	X		
10								X		
9								X		
8								X		
7							UNUSED[1:0]	0	RX_LOS_INVALID_INCR [7:0]	
6								0		
5			SLOWDOWN_CAL[1:0][3]				TXLOOPFILT [3:0]	X		
4								X		
3			RESERVED					X		
2			LOOPCAL_WAIT[1:0][3]					X		
1							RESERVED	0		
0	CHAN_BOND_SEQ_1_3[10]		BYPASS_CAL[3]					0		

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. TXOUTDIV2SEL_2 must equal TXOUTDIV2SEL_1. The attribute TXOUTDIV2SEL sets both TXOUTDIV2SEL_1/TXOUTDIV2SEL_2 registers upon configuration, but must be written to separately when using the Dynamic Configuration Bus. This attribute only affects the MGT A of a file.
3. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-14: **Dynamic Configuration Memory Map: MGT A Address 7C - 7F**

Bit	Address							
	7C	Def	7D	Def[1]	7E	Def	7F	Def
15	RESERVED[4:0]	N/A	RXOUTDIV2SEL_1 [3:0][3]	X	CHAN_BOND_TUNE [7:0]	N/A	RESERVED	N/A
14				X			TX_BUFFER_USE	
13				X			RX_BUFFER_USE	
12				X			CHAN_BOND_SEQ_LEN [2:0]	
11			RXPLLNDIVSEL[3:0]	X				
10	CLK_COR_MIN_LAT [5:0]			X				
9				X			CHAN_BOND_SEQ_USE	
8				X			CHAN_BOND_ONE_SHOT	
7			UNUSED[1:0]	0	TXENABLE		CHAN_BOND_MODE [1:0]	
6				0	RXENABLE			
5			RXLOOPFILT[3:0][2]	X	CCCB_ARBITRATOR_DISAB		CHAN_BOND_LIMIT [5:0]	
4	RESERVED			X	OPPOSITE_SELECT			
3	PCS_BIT_SLIP			X	POWER_ENABLE			
2	DIGRX_SYNC_MODE			X	RESERVED[2:0]			
1	DIGRX_FWDCLK[1:0]		RXDIGRX	0				
0			RXPFDTX	0				

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. RXOUTDIV2SEL_1 must equal RXOUTDIV2SEL_2. The attribute RXOUTDIV2SEL sets both RXOUTDIV2SEL_1/RXOUTDIV2SEL_2 registers upon configuration, but must be written to separately when using the Dynamic Configuration Bus.
3. Restricted. See Appendix F, “Restricted/Reserved Attributes,” for details.

Table C-15: **Dynamic Configuration Memory Map: MGT B Address 40 - 44**

Bit	Address									
	40	Def	41	Def	42	Def	43	Def	44	Def
15	RXCRCINITVAL [15:0]	N/A	RESERVED[9:0]	0	RESERVED[7:0]	N/A	RESERVED	N/A	RXEQ[15:0]	1
14				0			CLK_COR__8B10_DE			0
13				0			CLK_CORRECT_USE			0
12				0			CLK_COR_SEQ_LEN[2:0]			0
11				0						0
10				0						0
9				0			CLK_COR_SEQ_DROP			0
8				0			CLK_COR_SEQ_2_USE			0
7				0	COMMA32		RESERVED			10
6				0	PCOMMA_DETECT		RESERVED			0
5			UNUSED[4:0]	0	MCOMMA_DETECT		CLK_COR_MAX LAT[5:0]			0
4				0	DEC_VALID_COMMA_ONLY					0
3				0	DEC_PCOMMA_DETECT					0
2				0	DEC_MCOMMA_DETECT					0
1				0	ALIGN_COMMA_WORD[1:0]					0
0			RESERVED	0						0

Table C-16: **Dynamic Configuration Memory Map: MGT B Address 45 - 49**

Bit	Address									
	45	Def	46	Def	47	Def	48	Def	49	Def[1]
15	UNUSED[10:0]	0	RESERVED[14:0]	0	RESERVED	N/A	RXCRCINITVAL [31:16]	N/A	RESERVED	0
14		0		0					RXOUTDIV2 SEL_2[3:0][3]	X
13		0		0						X
12		0		0						X
11		0		0						X
10		1		0					RXCTRL1[9:0][2]	1
9		0		0						0
8		0		0						0
7		0		0						0
6		0		0						0
5		1		0						0
4	TXPHASESELA	0		0						0
3	TXPHASESELB	0		0						X
2	RESERVED	0		0						X
1	PMACLKENABLE[2]	0		0						0
0	PMACOREPWR ENABLE	1	TXPD	0					RESERVED	0

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. RXOUTDIV2SEL_2 must equal RXOUTDEV2SEL_1. The attribute RXOUTDIV2SEL sets both RXOUTDIV2SEL_1/RXOUTDIV2SEL_2 registers upon configuration, but must be written to separately when using the Dynamic Configuration Bus.
3. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-17: **Dynamic Configuration Memory Map: MGT B Address 4A - 4E**

Bit	Address									
	4A	Def	4B	Def	4C	Def	4D	Def	4E	Def[1]
15	RESERVED[24:8]	N/A	SH_INVALID_CNT_MAX[7:0]	N/A	RXEQ[31:16]	1	BANDGAPSEL	0	TXDAT_PRDRV_DAC[1:0][2]	1
14						0	RESERVED[14:0]	0		1
13						0		1	TXASYNCDIVIDE [0]	X
12						0		1	TXPOST_TAP_PD	1
11						0		1	TXPOST_TAP_DAC [4:0]	1
10						0		1		1
9						0		1		1
8						0		0		0
7			SH_CNT_MAX[7:0]			1		0		1
6						0		1	RESERVED[2:0]	0
5						0		1		0
4						0		1	TXDAT_TAP_DAC [4:0]	1
3						0		0		0
2						0		1		1
1						0		1		1
0						0		0		0

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-18: **Dynamic Configuration Memory Map: MGT B Address 4F - 53**

Bit	Address									
	4F	**Def**	**50**	**Def**	**51**	**Def**	**52**	**Def**	**53**	**Def**
15	RESERVED	N/A	TXCRCINITVAL [15:0]	N/A	UNUSED[11:0]	N/A	RESERVED[5:0]	N/A	RXFDCAL_CLOCK_ DIVIDE[1:0]	N/A
14										
13									TXFDCAL_CLOCK_ DIVIDE[1:0]	
12										
11									RXBY_32	
10									REPEATER	
9							COMMA_10B_ MASK[9:0]		ENABLE_DCDR	
8									SAMPLE_8X	
7									DCDR_FILTER[2:0][1]	
6										
5										
4									RXUSRDIVISOR [4:0]	
3										
2					RXSLOSEL					
1					RXDACSEL					
0					RXCPSEL[1]					

Notes:

1. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-19: **Dynamic Configuration Memory Map: MGT B Address 54 - 58**

Bit	Address									
	54	Def	55	Def	56	Def[1]	57	Def	58	Def
15	RXEQ[47:32]	1	RESERVED [15:0]	0	TXPRE_TAP_DAC[2:0]	0	RESERVED [15:0]	N/A	TXCRCINIT VAL[31:16]	N/A
14		0		0		0				
13		0		0		0				
12		0		0	RESERVED	0				
11		0		0	TXHIGHSIGNALEN[2]	1				
10		0		0	RESERVED	1				
9		0		0	TXTERMTRIM[3:0]	1				
8		0		0		1				
7		1		0		0				
6		0		0		0				
5		0		0	TXASYNCDIVIDE[1]	X				
4		0		0	TXSLEWRATE[3]	1				
3		0		0	TXPOST_PRDRV_ DAC[2:0][2]	1				
2		0		0		1				
1		0		0		1				
0		0		0	TXDAT_PRDRV_ DAC[2][2]	1				

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, “Restricted/Reserved Attributes,” for details.
3. TXSLEWRATE is defaulted to 1 for serial rates below 6.25 Gb/s.

Table C-20: **Dynamic Configuration Memory Map: MGT B Address 59 - 5D**

Bit	Address									
	59	Def[1]	5A	Def	5B	Def	5C	Def	5D	Def
15	RXOUTDIV2 SEL_1[3:0][3]	X	RESERVED [15:0]	N/A	RXRECCLK1_USE_ SYNC	N/A	RESERVED	0	RESERVED[1:0]	0
14		X			TXOUTCLK1_USE_ SYNC		RXEQ[62:48]	1		0
13		X			TXCLK0_FORCE_ PMA			0	RXAPMACLK SEL[1:0]	0
12		X			RXCLK0_FORCE_ PMA			0		0
11	RXPLLNDIVSEL [3:0]	X			TX_CLOCK_DIVIDER [1:0]			0	RXBPMACLK SEL[1:0]	0
10		X						0		0
9		X			RX_CLOCK_DIVIDER [1:0]			0	TXABPMACLK SEL[1:0]	0
8		X						0		0
7	RESERVED[1:0]	0			TXCRCENABLE			1	RESERVED[7:0]	0
6		0			TXCRCSAMECLOCK			0		0
5	RXLOOPFILT [3:0][2]	X			TXCRCINVERTGEN			0		0
4		X			TXCRCCLOCKDOUBLE			0		1
3		X			RXCRCENABLE			0		1
2		X			RXCRCSAMECLOCK			0		1
1	RXDIGRX	0			RXCRCINVERTGEN			0		0
0	RXPFDTX	0			RXCRCCLOCKDOUBLE			0		0

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.
3. RXOUTDIV2SEL_1 must equal RXOUTDIV2SEL_2. The attribute RXOUTDIV2SEL sets both RXOUTDIV2SEL_1/RXOUTDIV2SEL_2 registers upon configuration, but must be written to separately when using the Dynamic Configuration Bus.

Table C-21: **Dynamic Configuration Memory Map: MGT B Address 5E - 62**

Bit	Address									
	5E	Def[1]	5F	Def	60	Def	61	Def	62	Def
15	RESERVED[4:0]	0	RESERVED	N/A	RXBYPASS_CAL[2]	N/A	MCOMMA_32B_ VALUE/ MCOMMA_10B_ VALUE [15:0]	N/A	RXDIGRESET	0
14		0			RXFDET_HYS_ CAL[2:0]				RXSELDAC FIX[3:0]	0
13		0								0
12		0								0
11		0			RXFDET_LCK_ CAL[2:0]					0
10	TXPRE_PRDRV_DAC[2]	1							RXSELDAC TRAN[4:0]	1
9		1								0
8		1			RXFDET_HYS_ SEL[2:0]					0
7	TXPRE_TAP_PD	1								0
6	TXPRE_TAP_DAC[4:3]	0								0
5		0			RXFDET_LCK_ SEL[2:0]				RXLKAPD	0
4	RESERVED	0							RXRSDPD	0
3	TXCLKMODE[3:0]	X							RESERVED	0
2		X			RXVCO_CTRL_ ENABLE[2]				RESERVED	0
1		X			RXCYCLE_LIMIT_S EL**[1:0]				RESERVED	0
0		X							RXPD	0

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-22: **Dynamic Configuration Memory Map: MGT B Address 63 - 67**

Bit	Address									
	63	Def	64	Def	65	Def	66	Def	67	Def
15	CLK_COR_SEQ_2_2 [4:0]	N/A	CLK_COR_SEQ_2_1 [4:0]	N/A	RESERVED [15:0]	0	CHAN_BOND_SEQ_2_2[4:0]	N/A	CHAN_BOND_SEQ_1_2[4:0]	N/A
14						0				
13						0				
12						0				
11						0				
10	CLK_COR_SEQ_2_1 [11:0]		CLK_COR_SEQ_1_1 [11:0]			0	CHAN_BOND_SEQ_2_1 [10:0]		CHAN_BOND_SEQ_1_1 [10:0]	
9						0				
8						0				
7						0				
6						0				
5						0				
4						0				
3						0				
2						0				
1						0				
0						0				

Table C-23: **Dynamic Configuration Memory Map: MGT B Address 68 - 6C**

Bit	Address									
	68	**Def**	**69**	**Def**	**6A**	**Def**	**6B**	**Def**	**6C**	**Def**
15	RESERVED	N/A	MCOMMA_32B_VALUE/ MCOMMA_10B_VALUE [31:16]	N/A	RESERVED[1:0]	1	CLK_COR_SEQ_2_3[9:0]	N/A	CLK_COR_SEQ_1_3[9:0]	N/A
14	RXVCODAC_INIT [9:0][1]					0				
13					RXDCRLOS[5:0]	0				
12						0				
11						0				
10						0				
9						0				
8						0				
7					RXTADJ[1]	0				
6					RXDCCOUPLE	0				
5					UNUSED	0	CLK_COR_SEQ_2_2[10:5]		CLK_COR_SEQ_1_2[10:5]	
4	RXSLOWDOWN_CAL[1:0]				RXLKADJ[4:0]	0				
3						0				
2	RESERVED					0				
1	RXLOOPCAL_WAIT[1:0][1]					0				
0						0				

Notes:

1. Restricted. See Appendix F, “Restricted/Reserved Attributes,” for details.

Table C-24: **Dynamic Configuration Memory Map: MGT B Address 6D - 71**

Bit	Address									
	6D	**Def**	**6E**	**Def**	**6F**	**Def**	**70**	**Def**	**71**	**Def**
15	RESERVED [15:0]	0	CHAN_BOND_SEQ_2_3[9:0]	N/A	CHAN_BOND_SEQ_1_3[9:0]	N/A	RESERVED	N/A	PCOMMA_32B_VALUE/ PCOMMA_10B_VALUE [15:0]	N/A
14		0					FDET_HYS_CAL [2:0}			
13		0								
12		0								
11		0					FDET_LCK_CAL [2:0}			
10		0								
9		0								
8		0					FDET_HYS_SEL [2:0}			
7		0								
6		0								
5		0	CHAN_BOND_SEQ_2_2[10:5]		CHAN_BOND_SEQ_1_2[10:5]		FDET_LCK_SEL [2:0}			
4		0								
3		0								
2		0					VCO_CTRL_ENABLE[1]			
1		0					CYCL_LIMIT_SEL[1:0][1]			
0		0								

Notes:

1. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-25: **Dynamic Configuration Memory Map: MGT B Address 72 - 76**

Bit	Address									
	72	Def[1]	73	Def	74	Def	75	Def	76	Def
15	UNUSED	0	CLK_COR_SEQ_2_MASK[3:0]	N/A	CLK_COR_SEQ_1_MASK[3:0]	N/A	UNUSED[15:0]	0	CHAN_BOND_SEQ_2_MASK[3:0]	N/A
14	RXSELDACFIX [4]	1						0		
13	RXRCPADJ [2:0][2]	X						0		
12		X						0		
11		X	CLK_COR_SEQ_2_4[10:0]		CLK_COR_SEQ_1_4[10:0]			0	CHAN_BOND_SEQ_2_4[10:0]	
10	RESERVED [1:0]	1						0		
9		1						0		
8	RXAFEEQ[8:0]	0						0		
7		0						0		
6		0						0		
5		0						0		
4		0						0		
3		0						0		
2		0						0		
1		0						0		
0		0	CLK_COR_SEQ_2_3[10]		CLK_COR_SEQ_1_3[10]			0	CHAN_BOND_SEQ_2_3[10]	

Notes:

1. The default X depends on the operation. See Table C-28, page 196 for details.
2. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-26: **Dynamic Configuration Memory Map: MGT B Address 77 - 7B**

Bit	Address									
	77	Def	78	Def	79	Def	7A	Def[1]	7B	Def
15	CHAN_BOND_SEQ_1_MASK[3:0]	N/A	RESERVED	N/A	PCOMMA_32B_VALUE/ PCOMMA_10B_VALUE [31:16]	N/A	RESERVED[5:0]	0	RX_LOS_THRESHOLD[7:0]	N/A
14			VCODAC_INIT [9:0][1]					0		
13								0		
12								0		
11	CHAN_BOND_SEQ_1_4[10:0]							0		
10								0		
9							PMA_BIT_SLIP	0		
8							RXASYNCDIVIDE [1:0]	X		
7								X	RX_LOS_INVALID_INCR[7:0]	
6							RXCLKMODE[5:0]	X		
5			SLOWDOWN_CAL[1:0][1]					X		
4								X		
3			RESERVED					X		
2			LOOPCAL_WAIT[1:0][1]					X		
1								X		
0	CHAN_BOND_SEQ_1_3[10]		RESERVED				RXLB	0		

Notes:

1. Restricted. See Appendix F, "Restricted/Reserved Attributes," for details.

Table C-27: **Dynamic Configuration Memory Map: MGT B Address 7C - 7F**

Bit	Address							
	7C	Def	7D	Def	7E	Def	7F	Def
15	RESERVED[4:0]	N/A	UNUSED[15:0]	0	RESERVED[7:0]	N/A	RESERVED	N/A
14				0			TX_BUFFER_USE	
13				0			RX_BUFFER_USE	
12				0			CHAN_BOND_SEQ_LEN[2:0]	
11				0				
10	CLK_COR_MIN_LAT [5:0]			0				
9				0			CHAN_BOND_SEQ_USE	
8				0			CHAN_BOND_ONE_SHOT	
7				0	TXENABLE		CHAN_BOND_MODE [1:0]	
6				0	RXENABLE			
5				0	CCCB_ARBITRATOR_DISAB		CHAN_BOND_LIMIT [5:0]	
4	RESERVED[4:0]			0	OPPOSITE_SELECT			
3				0	POWER_ENABLE			
2				0	RESERVED[2:0]			
1				0				
0				0				

Table C-28: **PLL Configuration Settings**

MODE	VCO Freq (GHz)	REFCLK Freq (MHz)	TXPLLNDIVSEL RXPLLNDIVSEL	TXOUTDIV2SEL RXOUTDIV2SEL	RXLOOPFILT[3:2]T XLOOPFILT[3:2]	RXLOOPFILT[1:0] TXLOOPFILT[1:0]	RXCPSEL TXCPSEL	RXRCPADJ
OC12	2.488	155.52	0100	0010	01	01	0	XXX
OC48	2.488	155.52	0100	0011	01	01	0	100
OC192	4.976	622	0000	0001	01	01	0	111
1XFC	4.25	106.25	1010	0100/0001	01	11	1	XXX
2XFC	4.25	106.25	1010	0011	01	11	1	001
4XFC	4.2	106.25	1010	0010	01	11	1	001
1GE	5	125	0110	0100/0001	11	01	1	XXX
10GE	5.15625	644.5312	0000	0001	11	01	1	011
SRIO1	5	125	0110	0001	11	01	1	XXX
SRIO2	2.5	125	0110	0011	01	01	0	000
SRIO3	3.125	156.25	0110	0010	01	01	1	001
SATA1	3	150	0110	0010	01	01	1	001
SATA2	3	150	0110	0010	01	01	1	001
SXI5	2.488	155.52	0100	0011	01	01	0	100
SXI5	3.125	195.3125	0100	0010	01	01	0	000
SFI-4.2	2.488	155.52	0100	0011	01	01	0	100
IB	2.5	125	0110	0011	01	01	0	000
Aurora	2.5	125	0110	0011	01	01	0	000
Aurora	3.125	156.25	0110	0010	01	01	1	001
PCIe	2.5	125	0110	0011	01	01	0	000
PCIe Gen2	3.125	125	0110	0001	01	01	1	001
XAUI GE	3.125	156.25	0110	0010	01	01	1	001
XAUI FC	3.1875	159.375	0110	0010	01	01	1	010

Appendix D

Virtex-II Pro/Virtex-II Pro X to Virtex-4 RocketIO Transceiver Design Migration

Introduction

This appendix describes important differences regarding migration from the Virtex-II Pro/Virtex-II Pro X to the Virtex-4 RocketIO transceivers. This appendix does *not* describe all of the features and capabilities of these devices, but only highlights relevant PCB, power supply, and reference clock differences. For more information on Virtex-II Pro and Virtex-II Pro X FPGAs, refer to the *Virtex-II Pro Data Sheet* (DS083), the RocketIO *Transceiver User Guide* (UG024), and the RocketIO X *Transceiver User Guide* (UG035).

Primary Differences

Virtex-4 FPGAs are a different family from the Virtex-II Pro/Virtex-II Pro X family. The Virtex-4 FPGAs are not pin compatible. However, many aspects of the MGTs between the families are the same. The primary differences between Virtex-II Pro/Virtex-II Pro X FPGAs are:

- MGTs per device
- Clocking – Naming conventions and actual topography
- Serial rates – supporting 0.622 Gb/s to 10.3125 Gb/s I/O
- Encoding standards – 8B10B, 64B66B, SONET, and others
- Clock multipliers – X10, 16, 20, 32, 40
- Flexibility – partial reconfiguration, PMA programming bus, Dynamic reconfiguration bus
- Board guidelines

MGTs per Device

Virtex-4 FPGAs allow for a large range of MGTs per device. Table D-1 shows the number of MGTs available for each family.

Table D-1: **MGTs per Device**

Virtex-II Pro	4, 8, 12, 16, 20
Virtex-II Pro X	8, 20
Virtex-4	8, 12, 16, 20, 24

Clocking

As with Virtex-II Pro/Virtex-II Pro X MGTs, there are several available clock inputs. Table D-2 shows the clocks for each family and the serial speeds they are available for.

Table D-2: **Available Clock Inputs**

Family	Clock Names	Differential (Internal)	Dedicated Routes	Max Serial Speeds (Gb/s)	Dynamic Switching	Package Input Voltage	Inputs per Device (No. of Package Pins)	Clocks per Device
Virtex-II Pro	BREFCLK		Yes	3.125	Yes[1]	2.5	8[3]	2[3]
	BREFCLK2		Yes	3.125	Yes[1]	2.5	8[3]	2[3]
	REFCLK			2.5	Yes[1]	2.5	8[3]	2[3]
	REFCLK2			2.5	Yes[1]	2.5	8[3]	2[3]
Virtex-II Pro X	BREFCLK	Yes	Yes	10.3125	Yes	2.5/3.3[5]	4[4]	2[4]
	REFCLK			Not recommended	Yes	2.5	8[3]	2[3]
	REFCLK2			Not recommended	Yes	2.5	8[3]	2[3]
Virtex-4	GREFCLK	Yes	Yes	1.0	Yes[2]	TBD	(6)	(6)
	REFCLK1	Yes	Yes	10.3125	Yes[2]	TBD	8	4
	REFCLK2	Yes	Yes	10.3125	Yes[2]	TBD	8	4

Notes:

1. Dynamic selection between the REFCLKs or the BREFCLKs; to switch from REFCLK to BREFCLK or vice versa requires reconfiguration.
2. Reference clock switching is done via an attribute and the Dynamic Configuration Bus.
3. BREFCLK should use dedicated GCLK I/O which decreases GCLK I/O resources for other logic (also two pins per clock).
4. There is only one BREFCLK on each side and cannot drive MGTs on the other side of the chip.
5. Depends on clock speed and implementation.
6. GREFCLK comes from the global clock tree which can come from any FPGA clock input, but should only be used for serial rates under 1.0 Gb/s.

Clock selection has changed slightly over the past three generations of MGTs. Figure D-1 shows how the reference clocks are selected for each device.

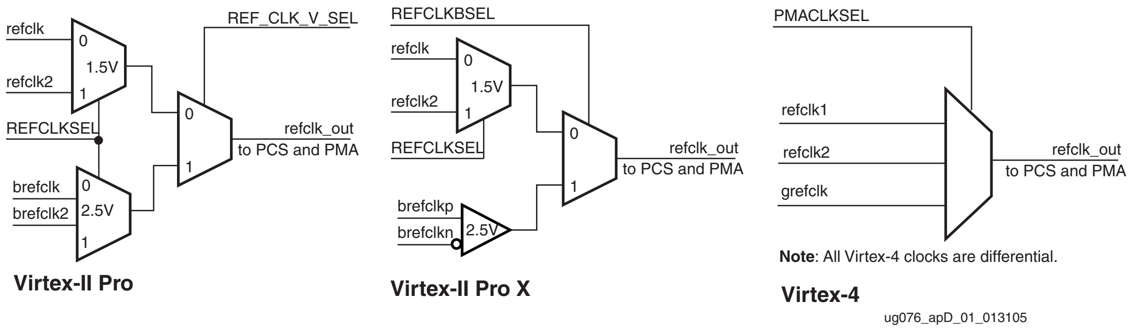


Figure D-1: **Reference Clock Selection for Each Device**

Serial Rate Support

As the MGTs continue to migrate, so do the supported serial rates. Virtex-4 MGTs span the serial range of both Virtex-II Pro and Virtex-II Pro X MGTs. Table D-3 shows the rates supported by each MGT.

Table D-3: **Serial Rate Support**

Serial Rate	Virtex-II Pro	Virtex-II Pro X	Virtex-4
.622 - 2.488	Yes	–	Yes
2.488 - 3.125	Yes	Yes	Yes
3.125 - 10.3125	–	Yes	Yes

Encoding Support and Clock Multipliers

The Virtex-4 MGT continues to support all of the standards supported in the Virtex-II Pro X MGTs. Table D-4 shows the encoding available in the MGT and which clock multipliers are available.

Table D-4: **Encoding Support and Clock Multipliers**

	Virtex-II Pro	Virtex-II Pro X	Virtex-4
Encoding Schemes			
8B/10B	Yes	Yes	Yes
64B/66B	Yes[2]	Yes	Yes
SONET	Yes[2]	Yes[1]	Yes
Others	Yes[3]	Yes[3]	Yes[3]
Clock Multipliers			
X16	–	Yes	Yes
X20	Yes	Yes	Yes
X32	–	Yes	Yes
X40	–	Yes	Yes

Notes:

1. Byte alignment must be done in fabric.
2. Encoding and clocks must be done in fabric.
3. Depending on encode, some functionality must be done in fabric.

Flexibility

In Virtex-II Pro devices, changing of attributes required partial reconfiguration. Virtex-II Pro X devices allow dynamic changing of PMA attributes via the PMA attribute bus. Virtex-4 devices allow all attribute changes from the Dynamic Configuration Bus, plus any default values can be set in the HDL itself.

Board Guidelines

Power Supply Filtering

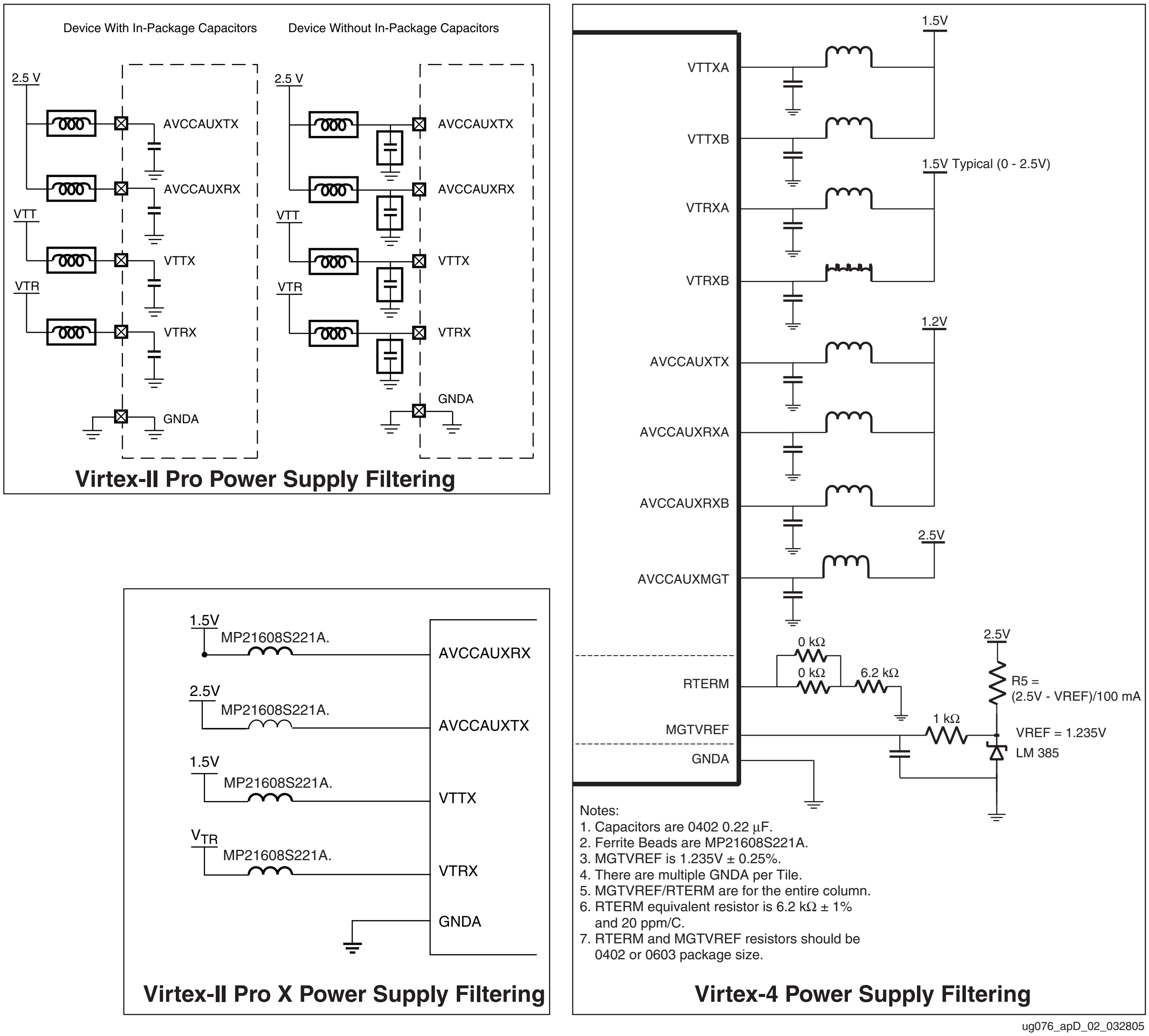
For the Virtex-4 RocketIO transceiver, the voltage level of the power pins has been reduced to 1.2V in the case of the AVCCAUXRX and AVCCAUXTX. See Table D-5 and Figure D-2.

Table D-5: **Power Pin Voltages**

Pin	Virtex-II Pro	Virtex-II Pro X	Virtex-4
AVCCAUXRX	2.5V	1.5V	1.2V
AVCCAUXTX	2.5V	2.5V	1.2V
AVCCAUXMGT	N/A	N/A	2.5V
VTTX	1.8 - 2.5V[1]	1.5V	1.5V
VTRX	1.5 - 2.5V[1]	0.25 - 2.5V[1]	0.25 - 2.5V[1]

Notes:

1. Depends on AC/DC coupling or termination options. See the user guides for more details.



Notes:
1. Capacitors are 0402 0.22 μF.
2. Ferrite Beads are MP21608S221A.
3. MGTVREF is 1.235V ± 0.25%.
4. There are multiple GNDA per Tile.
5. MGTVREF/RTERM are for the entire column.
6. RTERM equivalent resistor is 6.2 kΩ ± 1% and 20 ppm/C.
7. RTERM and MGTVREF resistors should be 0402 or 0603 package size.

Figure D-2: **Virtex-II, Virtex-II Pro, and Virtex-4 Power Supply Filtering**

Other Minor Differences

Termination

In Virtex-II Pro and Virtex-II Pro X devices, the transceivers contained on-chip termination and reference voltages for VTTX and VTRX. In Virtex-4 devices, the MGTs have a second option to use a reference resistor and voltage to create the termination circuitry for each MGT column. These new package pins are RTERM and MGTVREF (see Chapter 6, "Analog and Board Design Considerations" for more details). Table D-6 shows the termination options for each generation.

Table D-6: **Termination Options**

Termination	Virtex-II Pro	Virtex-II Pro X	Virtex-4
Value	50/75Ω	50Ω	50Ω
Voltage Pins	VTTX/VTRX	VTTX/VTRX	VTTX/VTRX

CRC

CRC support has changed over the three generations of transceivers. Table D-7 shows the CRC support for all three transceiver families.

Table D-7: **CRC Transceiver Support**

Virtex-II Pro	Virtex-II Pro X	Virtex-4
32-bit CRC	CRC must be done in fabric.	Flexible and independent CRC from 8 to 64 bits wide with the CRC-32 polynomial.

Loopback

The loopback options of the three generations of transceivers have evolved to improve flexibility. In Virtex-II Pro transceivers, there were two loopback modes. Virtex-II Pro X had three modes and Virtex-4 has four loopback modes. Table D-8 shows this evolution.

Table D-8: **Loopback Options**

Mode	Virtex-II Pro	Virtex-II Pro X	Virtex-4
Parallel Loopback (Tx –> RX)	Yes	Yes	Yes
Serial Pre-Driver	–	Yes	Yes
Serial Post-Driver	Yes	Yes	–
Repeater (RX –> TX)	–	–	Yes
Serial Digital Receiver	–	–	Yes

TKERR[0] vs. TKERR[3]

In Virtex-II Pro X devices, when 64B/66B function was in the clock-to-clock bypass TXKERR[3] indicated an even boundary for bypassing on a block basis. In Virtex-4 devices, this functionality is defined to be TXKERR[0].

Serialization

As in Virtex-II Pro X devices, Virtex-4 also serializes and sends the least significant byte first. This is opposite of the format Virtex-II Pro devices used in sending the most significant byte first.

Defining Clock Correction and Channel Bonding Sequences

The bit definitions of the CLK_COR_SEQ and CHAN_BOND_SEQ have changed to support more encoding functionality. Table D-9 illustrates the differences.

Table D-9: **CLK_COR_SEQ and CHAN_BOND_SEQ Sequences**

Bit Definition	Virtex-II Pro	Virtex-II Pro X	Virtex-4
8B/10B encoded definition	`00110111100`[1]	`00110111100`[2]	`00110111100`[2]
10-bit literal value	`10011111010`[1]	`10011111010`[2]	`10011111010`[2]
64B/66B encoding (sync character)	N/A	`1XX` (sync header)	`1XX` (sync header)
8-bit literal value (for 64B/66B and other encodings)	N/A	`1XX` (8-bit data)	`1XX` (8-bit data)

Notes:

1. Defines K28.5.
2. Defines K28.5 (also depends on CLK_COR_8B10B_DE) (plus all 10 bits are defined).

RXSTATUS Bus

Several buses have changed over the generations to improve the information that is indicated. Table D-10 shows the migration from Virtex-II Pro to Virtex-II Pro X and finally Virtex-4 devices.

Table D-10: **Status Bus Changes**

Description	Virtex-II Pro	Virtex-II Pro X	Virtex-4
Indicates channel bonding complete.	CHBONDONE[1]	CHANBONDDONE	RXSTATUS[5]
Indicates status bus is status, data, event.	N/A	RXBUFSTATUS[1:0]	RXSTATUS[4:3]
Indicates channel bonding or clock correction pointers change.	RXCLKCORCNT	RXCLKCORCNT[2:0]	RXSTATUS[2:0]
Indicates that an RX buffer has under/overflown.	RXBUFSTATUS[1]	N/A[2]	RXBUFERR

Notes:

1. RXCLKCORCNT must go to 3'b101 before channel bonding is complete.
2. Logic can be implemented in the fabric to indicate an under/over flow.

Pre-emphasis, Differential Swing, and Equalization

The differential signaling techniques have continued to get more robust throughout the MGT devices. Table D-11 shows the migration of attributes from Virtex-II Pro to Virtex-II Pro X and finally Virtex-4 devices.

Table D-11: **Signal Optimization Attributes**

Description	Virtex-II Pro	Virtex-II Pro X	Virtex-4
Controls TX pre-emphasis and edge rate	TX_PREMPHASIS	TXEMPHLEVEL	TXPRE_PRDRV_DAC TXPRE_TAP_PD TXSLEWRATE TXPOST_PRDRV_DAC TXDAT_PRDRV_DAC TXPOST_TAP_PD
Controls differential amplitude of the transmitted signal	TX_DIFF_CTRL	TXDOWNLEVEL	TXPRE_TAP_DAC TXPOST_TAP_DAC TXDAT_TAP_DAC
Active equalization	N/A	RXFER	RXAFEEQ
Discrete equalization	N/A	N/A	RXEQ

Appendix E

Related Online Documents

The documents described in this Appendix are accessible on the Xilinx website at www.xilinx.com. Document links shown in blue are clickable in this PDF file, providing easy access to the most current revision of each document.

Application Notes

TBD

Characterization Reports

TBD

White Papers

TBD

Appendix F

Restricted/Reserved Attributes

A group of restricted/reserved attributes allow flexibility into the MGT. These attributes are viewable in the software tools, but they should never be changed from their default settings. Otherwise, the MGT can operate below optimum levels, compromising the overall performance. Table F-1 shows these attributes and their defaults.

Table F-1: **Restricted/Reserved Attributes**

Attribute	Default Value[1]
BANDGAPSEL	FALSE
RXBYPASS_CAL	TBD
BYPASS_CAL	TBD
RXBYPASS_FET	TBD
BYPASS_FET	TBD
RXCPSEL	X
TXCPSEL	X
RXCTRL1	1000000XX00
TXCTRL1	1000000XX00
RXCYCLE_LIMIT_SEL	00
CYCLE_LIMIT_SEL	00
TXDAT_PRDRV_DAC	111
DCDR_FILTER	000
TXHIGHSIGNALEN	1
RXLOOPFILT	XXXX
LOOPFILT	XXXX
RXLOOPWAIT_CAL	TBD
TXPOST_PRDRV_DAC	111
TXPRE_PRDRV_DAC	111
RXRCPADJ	X
RXSLOWDOWN_CAL	00

Table F-1: **Restricted/Reserved Attributes *(Continued)***

Attribute	Default Value[1]
SLOWDOWN_CAL	00
RXTADJ	0
RXVCO_CTRL_ENABLE	TBD
VCO_CTRL_ENABLE	TBD
PMACLKENABLE	0

Notes:

1. The default X depends on the operation. See Table C-27, page 195 for details.

Index