Virtex-5 FPGA RocketIO GTP Transceiver

User Guide

UG196 (v2.1) December 3, 2009

XILINX®

Xilinx is disclosing this user guide, manual, release note, and/or specification (the "Documentation") to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© 2006–2009 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. PCI, PCI Express, PCIe, and PCI-X are trademarks of PCI-SIG. All other trademarks are the property of their respective owners.

Optiver Ex1009
Page 2 of 347

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/06/06	1.0	Initial release to CD.
10/13/06	1.1	Initial release to www.xilinx.com
02/02/07	1.2	Added SXT packages to Package Placement Information in Chapter 4. Inserted RX buffer overflow/underflow footnotes to Table 7-28 and Table 7-30. Added SelectIO to GTP Crosstalk Guidelines in Chapter 10. Added SelectIO to Serial Transceiver Crosstalk Guidelines in Chapter 11. Added Appendix E, Low Latency Design. Removed Virtex-II Pro X FPGA references.
05/25/07	1.3	Chapter 1: Revised line rates in the Overview, page 23. Added to RXBYTEISALIGNED description and removed CRC ports in Table 1-3, page 29. Corrected PCOMMA_DETECT entry and removed CRC_INIT[31:0] attribute in Table 1-5. CRC ports are not part of the GTP_DUAL primitive. See Chapter 8. Chapter 3: Added Providing Clocks in Simulation, page 51. Added multirate clocking design caveats and link to Appendix F. Chapter 4: Added a note 2 to Table 4-1, page 57. Chapter 5: Added to note 5 in Figure 5-1, page 72. Added PCS_COM_CFG and notes to Figure 5-2, page 75. Revised Equation 5-1. Changed PLL clock frequency for FC1, FC2, SFI-5, TFI-5, and the HD-SDI standard in Table 5-3, page 75. Revised the notes for Figure 5-5, page 83. Added PRBSCNTRESET and PLLPOWERDOWN, and revised GTPRESET description in Table 5-6, page 85. Revised GTP Component-Level Resets and Link Idle Reset Support, page 87. Added note to RXPOWERDOWN in Table 5-9, page 94. Added note to Table 5-11, page 96. Chapter 6: Added a BUFG to Figure 6-5. Revised PMA_COM_CFG, OVERSAMPLE_MODE, and added three attributes to Table 6-8, page 118. Revised the "Using the TX Phase-Alignment Circuit to Bypass the TX Buffer," page 119. Revised Figure 6-12, page 120. Added INTDATAWIDTH to Table 6-13, page 123. Revised OVERSAMPLE_MODE in Table 6-15, page 125. Revised TX_DIFF_BOOST in Table 6-17, page 127. Added default value to Table 6-19, page 128. Chapter 7: Revised Figure 7-2, page 141. Updated Table 7-3. Added OOB nominal values to Table 7-6. Added Tuning the CDR, page 152. Revised Table 7-12, page 154. Added note 1 to Table 7-29, page 178. Revised CLK_COR_MAX_LAT. Chapter 8: Added clarification to the CRC block description. Chapter 9: Made changes to Near-End PCS Loopback, Near-End PMA Loopback, Far-End PMA Loopback, and Far-End PCS Loopback, including adding Marginal Conditions and Limitations. Added Table 9-2. Chapter 10: Clarified REFCLK Guidelines, page 236. Added Figure 10-12. Added TERMINATION_IMP to Table 10-2. Added note on analog supplies to Table 10-5, Table 10-16, and Table 10-17. Added SelectIO adjacent to MGTCLK tables at the end of the chapter. Edited AC Coupling, page 240. Added an additional guideline to "Filter Network Design Guidelines." Appendix D: Added PCS_COM_CFG to Table D-2, Table D-7, and Table D-8. Revised bit 4 and 6 in Table D-3. Appendix E: Added note 2 to Table E-2, page 337. Added Appendix F.

Optiver Ex1009
Page 3 of 347

Date	Version	Revision
09/12/07	1.4	Revised maximum line rate from 3.2 Gb/s to 3.75 Gb/s in entire document. Removed SAS standards from Table 1-1 and Table 5-3. Added notes to Table 5-3. Added note 2 to Figure 5-3, page 81. Added clarification to note 4 under Figure 5-5, page 83. Added notes to Table 5-6, page 85. Added bullet under Link Idle Reset Support, page 87. Clarified situations in Table 5-8, page 90 and added note 2 to Table 5-9, page 94. Replaced REFCLKPOWERDNB with REFCLKPWRDNB. Added notes to Figure 5-10, page 99. Revised Figure 6-12, page 120. Increased CDR rate tolerance on page 149. Revised Table 7-11, page 153. Revised Using the CRC Blocks section including Figure 8-3 and adding Figure 8-4, page 210. Changed specific banks for FF1136 and FF1738 in Table 10-17, page 242. Removed Optimal Cable Length section on page 262. In Table D-1, page 305, revised PLL_DIVSEL_FB changed binary value for attribute value 5. Added Figure E-1, page 335. Revised PMA + Interface in Table E-1, page 336 and Table E-2. Also revised Comma Alignment and Not Oversampling in Table E-2, page 337. Added note to Figure F-2, page 341.
12/11/07	1.5	Made minor copy-edits. Added new reference in Preface [Ref 1] and revised others. Fixed directions of MGTRXN/MGTRXP and MGTTXN/MGTTXP ports in Table 1-2, page 28. Corrected domain of RXELECIDLE port in Table 1-3, page 29 and Table 7-5, page 144. Added summary tables of CRC ports (Table 1-4, page 37) and CRC attributes (Table 1-6, page 44). Corrected GTP_DUAL placement for XC5VSX50T in Figure 4-3, page 63. Added four LXT packages to Chapter 4: XC5VLX20T-FF323, XC5VLX30T-FF323, XC5VLX155T-FF1136, and XC5VLX155T-FF1738. Added statement about needing a stable reference clock to After Turning on a Reference Clock, page 91. Corrected arrow directions in Figure 6-9, page 114. Changed the number of TXUSRCLK2 clock cycles to wait in "Using the TX Phase-Alignment Circuit to Bypass the TX Buffer," page 119. Changed the number of TXUSRCLK2 clock cycles in Figure 6-12, page 120. Revised Table 6-9, page 119 and added a new footnote. Inserted clock domain for TXENPRBSTST ports in Table 6-13, page 123. Reversed Boost On and Boost Off designations and adjusted default setting in Table 6-19, page 128. Rewrote last paragraph in Differential Voltage Control, page 127. Revised Example 6 values in Table 7-7, page 148. Updated Table 7-11, page 153. Changed PLL_CLKDIV_FB in Equation 7-6 to PLL_DIVSEL_FB. Revised Figure 7-11, page 163. Changed RX_BUFFER in Enabling Clock Correction, page 187 to RX_BUFFER_USE. Revised Slave Elastic Buffer contents in Figure 7-33, page 187. Added footnote to Table 10-1, page 219. Added "Boundary-Scan Testing Guidelines," page 222. Revised first bullet under "Unused and Partially Used GTP_DUAL Column Guidelines," page 223. Added XC5VLX155T to the footnotes in Table 10-20, page 244 and Table 10-21, page 244. Added new subsection FPGA Logic Interface, page 289. Updated Table A-12, page 290 and added three footnotes. Updated latencies in Table E-1, page 336 and Table E-2, page 337. Added Index, page 343.

Optiver Ex1009
Page 4 of 347

Date	Version	Revision
02/11/08	1.6	Made minor copy-edits. Updated links to Xilinx website. Revised bullets in PCI Express and OC-12/48 protocol descriptions in Table 1-1, page 24. Revised Figure 5-1, page 72. Updated conditions for Equation 5-1. Updated PLLPOWERDOWN and TXDETECTRX descriptions in Table 5-6, page 85. Corrected arrow direction in Figure 6-2, page 106, Figure 6-3, page 106, Figure 7-35, page 202, and Figure 7-36, page 202. Rewrote "Boundary-Scan Testing Guidelines," page 222 and "Unused and Partially Used GTP_DUAL Column Guidelines," page 223.
09/23/08	1.7	Added 3G-SDI to Table 1-1, page 24 and Table 5-3, page 75. Revised RXPRBSERR port description in Table 1-3, page 29 and Table 7-18, page 161 to equals or exceeds. Corrected direction of RXVALID to Out in Table 7-5, page 144. Revised PRBS_ERR_THRESHOLD attribute description in Table 7-19, page 161 to equals or exceeds. Corrected the overflow and underflow encodings of RXBUFSTATUS in Table 7-28, page 177 and Table 7-30, page 184. In Table 10-5, page 223, renamed table title, corrected power supply names, and added MGTRREF. Added note about BGA adjacency guidelines to SelectIO to GTP Crosstalk Guidelines, page 242.
11/10/08	1.8	• Added table note 1 to Table 1-1, page 24 on spread-spectrum clocking. • Added SIM_MODE attribute to Table 1-5, page 37 and Table 3-1, page 49. • Added SIM_MODE, page 50. • Added table note 6 to Table 5-3, page 75. • Added RXRESET as a recommended reset to the "After connecting RXN/RXP" row in Table 5-8, page 90 and to After Connecting RXP/RXN, page 92. • Changed "power control" to "power down" throughout Generic GTP_DUAL Power-Down Capabilities, page 96. • Removed Relative Power Savings and Recovery Time columns from Table 5-11, page 96. In Table 6-4, page 112, corrected encoding of TXKERR and TXRUNDISP and defined them based on the interface width. • Revised TXINHIBIT, page 129. • Added paragraph about PLL_RXDIVSEL_OUT to Normal Operation Mode, page 152. • Added table note 4 to Table 7-11, page 153. • Revised bits [23:16] and table note in Table 7-12, page 154. • Added note above Figure 10-2, page 222. • Updated Figure 10-2. • Added Figure 10-3, page 210 and associated note. • Added three bulleted notes under Figure 10-4, page 223. • Renamed "Unused and Partially Used GTP_DUAL Column Guidelines," page 223. • Rewrote "Partially used GTP_DUAL column," page 223. • In Table 10-6, page 223, updated table note 1 and added table note 2.
12/01/08	1.8.1	Corrected Section 1 header.
03/25/09	1.9	• In Chapter 10, added new sections Resistor Calibration Circuit, page 220 and Power Supply Design and Filtering, page 225. • Revised Providing Power, page 230 in Chapter 10.

Optiver Ex1009
Page 5 of 347

Date	Version	Revision
06/10/09	2.0	• Updated table note 3 in Table 10-12, page 234.
12/03/09	2.1	Added note 1 to Table 1-1. Revised description in Table 1-3 for RXBYTEISALIGNED0 and RXBYTEISALIGNED1, page 31. Changed attributes in Table 1-5 from TXRX_INVERT0 and TXRX_INVERT1 to TXRX_INVERT_0 and TXRX_INVERT_1, page 44. Revised Overview, page 47. Added XC5VSX240T-FF1738 to Package Placement Information, page 60, Figure 4-6, Figure 4-7, and Figure 4-8. Revised description in Table 5-6 for RESETDONE0 and RESETDONE1, page 85 and revised note 1 and note 2. Revised GTP Reset When the GTPRESET Port is Asserted, page 87, and Figure 5-8. Revised reset pins in Table 5-7 for Comma Detect and Align, page 89. Added reset information to Table 5-8 for RXPOWERDOWN and restart comma alignment. Added After Coming out of RXPOWERDOWN (RXPOWERDOWN Not Equal to 00), page 91. Added Restart Comma Alignment, page 91. Added note 3 to Table 5-9. Extensively revised TX Buffering, Phase Alignment, and TX Skew Reduction, page 115 through page 121. Changed PMACLK to XCLK on page 115, page 116, page 120, page 176, and page 177. Removed reference to note 2 from the TX Phase Alignment description for Skew Reduction in Table 6-6. Revised Description in Table 6-8 for PLL_TXDIVSEL_OUT_0 and PLL_TXDIVSEL_OUT_1, page 118 . Changed attributes from TXRX_INVERT0 and TXRX_INVERT1 to TXRX_INVERT_0 and TXRX_INVERT_1, page 119. In Overview, page 115, added new first paragraph and revised the fourth overview paragraph on page 116. Revised description in Table 6-17 for TX_DIFF_BOOST0 and TX_DIFF_BOOST1, page 127. Revised CDR Reset, page 151. Revised description in Table 7-20 for RXBYTEISALIGNED0 and RXBYTEISALIGNED1, page 164. Added note 1 to Table 7-21. Revised Enabling the 8B/10B Decoder, page 173. Revised Disparity Errors and Not-in-Table Errors, page 174. Added note 1 and note 2 to Table 7-26. Revised Using RX Phase Alignment to Bypass the RX Elastic Buffer, page 179. Extensively revised Description, page 193 though page 199. Changed port RXRESET in Table 7-34 to RXRESET0 and RXRESET1, page 200. Revised Marginal Conditions and Limitations, page 216. Added note to Far-End PCS Loopback, page 217. Added GTP Transceiver Power Supply Sharing, page 229. Added attributes ALIGN_COMMA_WORD, page 305 and REFCLK_SEL[2:0], page 306 to Table D-1. Revised DRP addresses for DEC_ MCOMMA_DETECT_1, DEC_PCOMMA_DETECT_0, MCOMMA_DETECT_1, and PCOMMA_DETECT_0 in Table D-2. Revised column title in Table F-1 and Table F-2 from REFCLK_SEL to REFCLK_SEL[2:0].

Optiver Ex1009
Page 6 of 347

Table of Contents

Revision History 3

Preface: About This Guide

Guide Contents 17
Additional Documentation 18
Additional Support Resources 19
Typographical Conventions 19
Online Document 20

Section 1: FPGA Level Design

Chapter 1: Introduction to the RocketIO GTP Transceiver

Overview 23
Ports and Attributes 28

Chapter 2: RocketIO GTP Transceiver Wizard

Chapter 3: Simulation

Overview 47
Ports and Attributes 49
Description 49
Limitations 50
SmartModel Attributes 50
SIM_GTPRESET_SPEEDUP 50
SIM_MODE. 50
SIM_PLL_PERDIV2 50
SIM_RECEIVER_DETECT_PASS 51
Power-Up and Reset 51
Link Idle Reset 51
Toggling GSR 51
Providing Clocks in Simulation 51
Simulating in Verilog 51
Defining GSR/GTS in a Test Bench 51
Simulating in VHDL 52
Examples 53
Simulation Environment Setup Example (ModelSim SE 6.1d on Linux) 53
SIM_PLL_PERDIV2 Calculation Example 54
Example for PCI Express Design 54
Example for Gigabit Ethernet Design 54
Example for XAUI Design 55

Chapter 4: Implementation

Overview . . . 57
Ports and Attributes . . . 57
Description . . . 58
Example of a UCF for GTP_DUAL Placement . . . 59
Package Placement Information . . . 60

Chapter 5: Tile Features

Tile Features Overview . . . 71
Shared PMA PLL . . . 72
Overview . . . 72
Ports and Attributes . . . 72
Description . . . 74
Examples . . . 76
Configuring the Shared PMA PLL for XAUI Operation . . . 76
Configuring the Shared PMA PLL for OC-48 Operation . . . 77
Configuring the Shared PMA PLL for Gigabit Ethernet Operation . . . 78
Configuring the Shared PMA PLL for PCI Express Operation . . . 79
Clocking . . . 80
Overview . . . 80
Ports and Attributes . . . 82
Description . . . 82
Clocking from an External Source . . . 82
Clocking from a Neighboring GTP_DUAL Tile . . . 83
Clocking using GREFCLK . . . 84
Reset . . . 84
Overview . . . 84
Ports and Attributes . . . 85
Description . . . 86
GTP Reset in Response to Completion of Configuration . . . 86
GTP Reset When the GTPRESET Port is Asserted . . . 87
GTP Component-Level Resets . . . 87
Link Idle Reset Support . . . 87
Resetting the GTP_DUAL Tile . . . 89
Examples . . . 91
Power Control . . . 94
Overview . . . 94
Ports and Attributes . . . 94
Description . . . 96
Generic GTP_DUAL Power-Down Capabilities . . . 96
Power-Down Features for PCI Express Operation . . . 97
Power-Down Transition Times . . . 98
Examples . . . 98
Dynamic Reconfiguration Port . . . 101
Overview . . . 101
Ports and Attributes . . . 101
Description . . . 101

Apotex Ex1009
Page 8 of 347

Chapter 6: GTP Transmitter (TX)

Transmitter Overview 103
FPGA TX Interface 104
Overview 104
Ports and Attributes 104
Description 105
Configuring the Width of the Interface 105
Connecting TXUSRCLK and TXUSRCLK2 106
Examples 107
TXOUTCLK Driving a GTP TX in 1-Byte Mode 107
TXOUTCLK Driving GTP TX in 2-Byte Mode 108
TXOUTCLK Driving Multiple Transceivers for a 2-Byte Datapath 109
REFCLKOUT Driving Multiple Transceivers with a 2-Byte Interface 110
Configurable 8B/10B Encoder 111
Overview 111
Ports and Attributes 112
Description 113
Enabling 8B/10B Encoding 113
8B/10B Bit and Byte Ordering 113
K Characters 114
Running Disparity 114
8B/10B Bypass 115
TX Buffering, Phase Alignment, and TX Skew Reduction 115
Overview 115
Description 119
Using the TX Buffer 119
Phase-Alignment Procedure 119
Minimizing TX Skew 120
Bypassing the TX Buffer 121
TX Polarity Control 122
Overview 122
Ports and Attributes 122
Description 122
TX PRBS Generator 123
Overview 123
Ports and Attributes 123
Description 124
Parallel In to Serial Out 124
Overview 124
Ports and Attributes 124
Description 125
Configurable TX Driver 126
Overview 126
Ports and Attributes 127
Description 127
Differential Voltage Control 127
Pre-emphasis 128
Configurable Termination Impedance 129
TXINHIBIT 129
Receive Detect Support for PCI Express Operation 130
Overview 130

Optiver Ex1009
Page 9 of 347

Ports and Attributes . . . 130
Description . . . 131
TX OOB/Beacon Signaling . . . 133
Overview . . . 133
Ports and Attributes . . . 133
Description . . . 134
Beacon Signaling for PCI Express Operations . . . 134
OOB Signaling for SATA Operations . . . 134

Chapter 7: GTP Receiver (RX)

Receiver Overview . . . 137
RX Termination and Equalization . . . 139
Overview . . . 139
Ports and Attributes . . . 139
Description . . . 140
Optional Built-in AC Coupling . . . 141
Configurable Termination Impedance . . . 142
Configurable Termination Voltage . . . 142
Optional Configurable RX Linear Equalization . . . 143
RX OOB/Beacon Signaling . . . 143
Overview . . . 143
Ports and Attributes . . . 144
Description . . . 146
Detecting Electrical Idle for PCI Express Operation . . . 146
Detecting OOB for SATA Operation . . . 147
Example . . . 148
RX Clock Data Recovery . . . 149
Overview . . . 149
Ports and Attributes . . . 149
Description . . . 150
CDR Reset . . . 151
Tuning the CDR . . . 152
Horizontal Sample Point Shift . . . 153
Serial In to Parallel Out . . . 155
Overview . . . 155
Ports and Attributes . . . 155
Description . . . 156
Oversampling . . . 157
Overview . . . 157
Ports and Attributes . . . 157
Description . . . 158
Configuring the 5x Line Rate . . . 158
Configuring the PCS Internal Datapath and Clocks . . . 159
Activating and Operating the Oversampling Block . . . 159
RX Polarity Control . . . 160
Overview . . . 160
Ports and Attributes . . . 160
Description . . . 160
PRBS Detection . . . 161
Overview . . . 161
Ports and Attributes . . . 161

Apotex Ex1009
Page 10 of 347

Description 162
Configurable Comma Alignment and Detection 162
Overview 162
Ports and Attributes 163
Description 166
Enabling Comma Alignment 166
Configuring Comma Patterns 166
Activating Comma Alignment 167
Alignment Status Signals 167
Alignment Boundaries 167
Manual Alignment 168
Configurable Loss-of-Sync State Machine 169
Overview 169
Ports and Attributes 169
Description 170
Configurable 8B/10B Decoder 172
Overview 172
Ports and Attributes 172
Description 173
Enabling the 8B/10B Decoder 173
8B/10B Decoder Bit and Byte Order 173
K Characters and 8B/10B Commas 174
RX Running Disparity 174
Disparity Errors and Not-in-Table Errors 174
Configurable RX Elastic Buffer and Phase Alignment 176
Overview 176
Ports and Attributes 177
Description 178
Using the RX Elastic Buffer 178
Using RX Phase Alignment to Bypass the RX Elastic Buffer 179
Bypassing the RX Elastic Buffer while Using Built-in Oversampling 182
Configurable Clock Correction 183
Overview 183
Ports and Attributes 184
Description 187
Enabling Clock Correction 187
Setting RX Elastic Buffer Limits 187
Setting Clock Correction Sequences 187
Clock Correction Options 188
Monitoring Clock Correction 188
Configurable Channel Bonding (Lane Deskew) 189
Overview 189
Ports and Attributes 190
Description 193
Enabling Channel Bonding 193
Channel Bonding Mode 193
Connecting Channel Bonding Ports 193
Setting the Channel Bonding Sequence 197
Setting the Maximum Skew 198
Precedence between Channel Bonding and Clock Correction 199
FPGA RX Interface 200
Overview 200

Ports and Attributes . . . 200
Description . . . 201
Configuring the Width of the Interface . . . 201
Connecting RXUSRCLK and RXUSRCLK2 . . . 202

Chapter 8: Cyclic Redundancy Check

Overview . . . 205
Ports and Attributes . . . 206
Description . . . 207
Using CRC for Error Checking . . . 207
The CRC Primitive . . . 208
Using the CRC Blocks . . . 209
Integrating the CRC Blocks for TX . . . 211
Integrating the CRC Blocks for RX . . . 211
Implementation of the CRC Block . . . 212
References . . . 212

Chapter 9: Loopback

Overview . . . 213
Ports and Attributes . . . 214
Description . . . 214
Near-End PCS Loopback . . . 214
Near-End PMA Loopback . . . 215
Marginal Conditions and Limitations . . . 215
Far-End PMA Loopback . . . 216
Marginal Conditions and Limitations . . . 216
Far-End PCS Loopback . . . 217

Chapter 10: GTP-to-Board Interface

Analog Design Guidelines . . . 219
Overview . . . 219
Ports and Attributes . . . 219
Resistor Calibration Circuit . . . 220
Power Supply Design and Filtering . . . 225
Linear Regulator Selection Criteria . . . 226
Regulator Design Guidelines . . . 228
Ferrite Selection Guidelines . . . 228
Capacitor Selection Guidelines . . . 229
Filter Network Design Guidelines . . . 229
Boundary-Scan Testing Guidelines . . . 229
GTP Transceiver Power Supply Sharing . . . 229
Providing Power . . . 230
Overview . . . 230
Description . . . 230
Partially Used Column . . . 231
Fully Unused Column . . . 234
Fully Used Column . . . 235
Example . . . 236
REFCLK Guidelines . . . 236

Apotex Ex1009
Page 12 of 347

Overview 236
GTP Reference Clock Checklist 239
Description 239
Oscillator Selection 239
Sourcing More Than One Differential Clock Input Pair from One Oscillator 239
Switching between Two Different Reference Clocks 240
AC Coupling 240
Unused Reference Clock Inputs of GTP_DUAL Tiles for Clock Forwarding 240
Examples of Vendors and Devices 241
SelectIO to GTP Crosstalk Guidelines 242

Section 2: Board Level Design

Chapter 11: Design Constraints Overview

Powering Transceivers 250
Power Distribution Architecture 250
Regulator Selection 251
Filtering 251
Reference Clock 251
Clock Sources 251
Clock Traces 251
Coupling 252
DC Coupling 252
AC Coupling 252
External Capacitor Value Selection 252
SelectIO to Serial Transceiver Crosstalk Guidelines 255

Chapter 12: PCB Materials and Traces

How Fast is Fast? 257
Dielectric Losses 257
Relative Permittivity 257
Loss Tangent 258
Skin Effect and Resistive Losses 258
Choosing the Substrate Material 258
Traces 259
Trace Geometry 259
Trace Characteristic Impedance Design 259
Trace Routing 261
Plane Splits 261
Return Currents 261
Simulating Lossy Transmission Lines 262
Cable 262
Connectors 262
Skew Between Conductors 262

Chapter 13: Design of Transitions

Excess Capacitance and Inductance 263
Time Domain Reflectometry 263

BGA Package 265
SMT Pads 265
Differential Vias 269
P/N Crossover Vias 272
SMA Connectors 272
Backplane Connectors 272
Microstrip/Stripline Bends 272

Chapter 14: Guidelines and Examples

Summary of Guidelines 277
BGA Escape Example 278
HM-Zd Design Example 278

Section 3: Appendices

Appendix A: MGT to GTP Transceiver Design Migration

Overview 283
Primary Differences 283
MGTs per Device 284
Clocking 284
Serial Rate Support 285
Encoding Support and Clock Multipliers 286
Flexibility 287
Board Guidelines 287
Power Supply Filtering 287
Other Minor Differences 289
Termination 289
FPGA Logic Interface 289
CRC 289
Loopback 289
Serialization 290
Defining Clock Correction and Channel Bonding Sequences 290
RXSTATUS Bus 291
Pre-emphasis, Differential Swing, and Equalization 291

Appendix B: OOB/Beacon Signaling

OOB Signaling for SATA Operation 293
Beacon Signaling for PCI Express Operation 294

Appendix C: 8B/10B Valid Characters

Appendix D: DRP Address Map of the GTP_DUAL Tile

DRP Address by Attribute 307
DRP Address by Bit Location 322

Apotex Ex1009
Page 14 of 347

Appendix E: Low Latency Design

GTP TX Latency 336
GTP RX Latency 337

Appendix F: Advanced Clocking

Example 341

Index 343

Optiver Ex1009
Page 15 of 347

Apple Ex1009
Page 16 of 347

About This Guide

This document shows how to use the RocketIO™ GTP transceivers in Virtex®-5 FPGAs. Complete and up-to-date documentation of the Virtex-5 family of FPGAs is available on the Xilinx website at http://www.xilinx.com/virtex5.

Guide Contents

This manual contains the following chapters and appendices:

- "Section 1: FPGA Level Design"
  - Chapter 1, Introduction to the RocketIO GTP Transceiver
  - Chapter 2, RocketIO GTP Transceiver Wizard
  - Chapter 3, Simulation
  - Chapter 4, Implementation
  - Chapter 5, Tile Features
  - Chapter 6, GTP Transmitter (TX)
  - Chapter 7, GTP Receiver (RX)
  - Chapter 8, Cyclic Redundancy Check
  - Chapter 9, Loopback
  - Chapter 10, GTP-to-Board Interface
- "Section 2: Board Level Design"
  - Chapter 11, Design Constraints Overview
  - Chapter 12, PCB Materials and Traces
  - Chapter 13, Design of Transitions
  - Chapter 14, Guidelines and Examples
- "Section 3: Appendices"
  - Appendix A, MGT to GTP Transceiver Design Migration
  - Appendix B, OOB/Beacon Signaling
  - Appendix C, 8B/10B Valid Characters
  - Appendix D, DRP Address Map of the GTP_DUAL Tile
  - Appendix E, Low Latency Design
  - Appendix F, Advanced Clocking

Optiver Ex1009
Page 17 of 347

Additional Documentation

The following documents are also available for download at http://www.xilinx.com/virtex5.

- Virtex-5 Family Overview

  The features and product selection of the Virtex-5 family are outlined in this overview.

- Virtex-5 FPGA Data Sheet: DC and Switching Characteristics

  This data sheet contains the DC and Switching Characteristic specifications for the Virtex-5 family.

- Virtex-5 FPGA User Guide

  This user guide includes chapters on:

  - Clocking Resources
  - Clock Management Technology (CMT)
  - Phase-Locked Loops (PLLs)
  - Block RAM and FIFO memory
  - Configurable Logic Blocks (CLBs)
  - SelectIO™ Resources
  - I/O Logic Resources
  - Advanced I/O Logic Resources

- Virtex-5 FPGA RocketIO GTX Transceiver User Guide

  This guide describes the RocketIO™ GTX transceivers available in the Virtex-5 TXT and FXT platforms.

- Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC User Guide

  This user guide describes the dedicated Tri-Mode Ethernet Media Access Controller available in the Virtex-5 LXT, SXT, TXT, and FXT platform devices.

- Virtex-5 FPGA Integrated Endpoint Block User Guide for PCI Express Designs

  This user guide describes the integrated Endpoint blocks in the Virtex-5 LXT, SXT, TXT, and FXT platform devices for PCI Express® designs.

- Virtex-5 XtremeDSP Design Considerations

  This guide describes the XtremeDSP™ slice and includes reference designs for using the DSP48E.

- Virtex-5 FPGA Configuration Guide

  This all-encompassing configuration guide includes chapters on configuration interfaces (serial and SelectMAP), bitstream encryption, Boundary-Scan and JTAG configuration, reconfiguration techniques, and readback through the SelectMAP and JTAG interfaces.

- Virtex-5 FPGA System Monitor User Guide

  The System Monitor functionality available in all the Virtex-5 devices is outlined in this guide.

- Virtex-5 FPGA Packaging and Pinout Specification

  This specification includes the tables for device/package combinations and maximum I/Os, pin definitions, pinout tables, pinout diagrams, mechanical drawings, and thermal specifications.

Cpuver Ex1009
Page 18 of 347

The following documents provide supplemental material useful to this user guide:

1. Athavale, Abhijit and Carl Christensen. *High-Speed Serial I/O Made Simple*. http://www.xilinx.com/publications/books/serialio/serialio-book.pdf
2. *Synthesis and Simulation Design Guide* http://www.xilinx.com/support/software_manuals.htm
3. Granberg, Tom. *Handbook of Digital Techniques for High-Speed Design*. Prentice-Hall. ISBN-10: 0-13-142291-X. ISBN-13: 978-0131422919.
4. Grover, Frederick W., Ph.D. 1946. *Inductance Calculations: Working Formulas and Tables*. New York: D. Van Nostrand Company, Inc.
5. Johnson, Howard, Martin Graham. *High-Speed Signal Propagation: Advanced Black Magic*. Prentice-Hall. ISBN-10: 0-13-084408-X. ISBN-13: 978-0130844088.
6. Montrose, Mark I. 1999. *EMC and the Printed Circuit Board*. The Institute of Electrical and Electronics Engineers, Inc. ISBN 0-7803-4703-X.
7. Smith, Larry D. November 1984. *Decoupling Capacitor Calculations for CMOS Circuits*. Proceedings EPEP Conference.
8. Williams, Ross N. *The Painless Guide to CRC Error Detection Algorithms*. http://www.ross.net/crc/ (CRC pitstop).
9. DS083, *Virtex-II Pro and Virtex-II Pro X Platform FPGAs Complete Data Sheet*.
10. UG024, *RocketIO Transceiver User Guide*.
11. UG076, *Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide*.
12. XAPP209, *IEEE 802.3 Cyclic Redundancy Check*.
13. XAPP562, *Configurable LocalLink CRC Reference Design*.
14. UG351, *Virtex-5 FPGA RocketIO Transceiver Signal Integrity Simulation Kit User Guide*.

Additional Support Resources

To find additional documentation, see the Xilinx website at:

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

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

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

Typographical Conventions

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

Convention	Meaning or Use	Example
Italic font	References to other documents	See the *Virtex-5 Configuration Guide* for more information.
	Emphasis in text	The address (F) is asserted *after* clock event 2.
Underlined Text	Indicates a link to a web page.	http://www.xilinx.com/virtex5

Online Document

The following conventions are used in this document:

Convention	Meaning or Use	Example
Blue text	Cross-reference link to a location in the current document	See the section Additional Documentation for details. Refer to "Clock Management Technology (CMT)" in Chapter 2 for details.
Blue, underlined text	Hyperlink to a website (URL)	Go to http://www.xilinx.com for the latest documentation.

Oplover Ex1009
Page 20 of 347

Section 1: FPGA Level Design

This section provides the information needed to incorporate RocketIO™ GTP transceivers into an FPGA design, including:

- The features and characteristics of the GTP transceivers
- How to use the RocketIO GTP Wizard to configure the transceivers
- Mapping of transceiver instances to device resources
- Simulation of GTP transceiver designs
- Board-level clocking and power requirements

This section includes the following chapters:

Introduction to the RocketIO GTP Transceiver

RocketIO GTP Transceiver Wizard

Simulation

Implementation

Tile Features

GTP Transmitter (TX)

GTP Receiver (RX)

Cyclic Redundancy Check

Loopback

GTP-to-Board Interface

Cpuver Ex1009
Page 22 of 347

Chapter 1

Introduction to the RocketIO GTP Transceiver

Overview

The RocketIO™ GTP transceiver is a power-efficient transceiver for Virtex®-5 FPGAs. The GTP transceiver is highly configurable and tightly integrated with the programmable logic resources of the FPGA. It provides the following features to support a wide variety of applications:

- Current Mode Logic (CML) serial drivers/buffers with configurable termination, voltage swing, and coupling
- Programmable TX pre-emphasis and RX equalization for optimized signal integrity
- Line rates from 100 Mb/s to 3.75 Gb/s, with optional 5x digital oversampling required for rates between 100 Mb/s and 500 Mb/s
- Optional built-in PCS features, such as 8B/10B encoding, comma alignment, channel bonding, and clock correction
- Fixed latency modes for minimized, deterministic datapath latency
- Beacon signaling for PCI Express® designs and Out-of-Band signaling including COM signal support for SATA designs

The first-time user is recommended to read *High-Speed Serial I/O Made Simple* [Ref 1], which discusses high-speed serial transceiver technology and its applications.

Table 1-1 lists some of the standard protocols designers can implement using the GTP transceiver. The Xilinx CORE Generator™ tool includes a Wizard to automatically configure GTP transceivers to support one of these protocols or perform custom configuration (see Chapter 2, RocketIO GTP Transceiver Wizard).

Optiver Ex1009
Page 23 of 347

Table 1-1: **List of Standards Supported by the GTP_DUAL Tile**

Protocols Supported[1]	Protocol Data Rates Supported	Miscellaneous Features
PCI Express, Rev. 1.0a PCI Express, Rev. 1.1	2.5 Gb/s	• TX receive detect • Loss of Signal (LOS)/Idle state detect • Low power states • Beacon signaling • Ground referenced termination
XAUI 802.3ae D5p0	3.125 Gb/s	LOS
OC-12/48	622.08/2488.32 Mb/s	Allow bypassing FIFOs for synchronous operation
FC-1, Rev. 4.0	1.0625 Gb/s	Rate negotiation (allows operating the TX and RX at different speeds)
FC-2, Rev. 4.0	2.125 Gb/s	
10GFC	3.1875 Gb/s	
SDI HD-SDI DVB-ASI 3G-SDI	143/176/270/360 Mb/s 1.485/1.4835 Gb/s 270 Mb/s 2.970 Gb/s	
10GBASE-CX4 802.3ak/D4.0	3.125 Gb/s	
Gigabit Ethernet (1000BASE-CX 802.3z/D5.0)	1.25 Gb/s	
SATA Generation 1/2, Rev. 1.0a SATA Generation 2, Rev. 1.0a	1.5 Gb/s[2] 3.0 Gb/s	• Rate negotiation for Generation 2 (entire link operates at Generation 1/Generation 2 speeds) • LOS • OOB beacon
Serial RapidIO	1.25/2.5/3.125 Gb/s	
CPRI, Version 2.0	614.4/1228.8/2457.6 Mb/s	
Infiniband (Volume 2, Release 1.1)	2.5 Gb/s	
SFI-5	2.488 – 3.125 Gb/s	Synchronous clocking (bypass FIFOs)
OBSAI RP3 (Spec. Issue 1.0)	768/1536/3072 Mb/s	
Aurora	100 Mb/s – 3.75 Gb/s	

Notes:

1. Support for the standards described in this table does not imply electrical or protocol compliance with specific standards. All protocols or standards require additional IP to be implemented in the FPGA logic.
2. Spread-spectrum clocking (SSC) is only supported for line rates greater than 2 Gb/s.

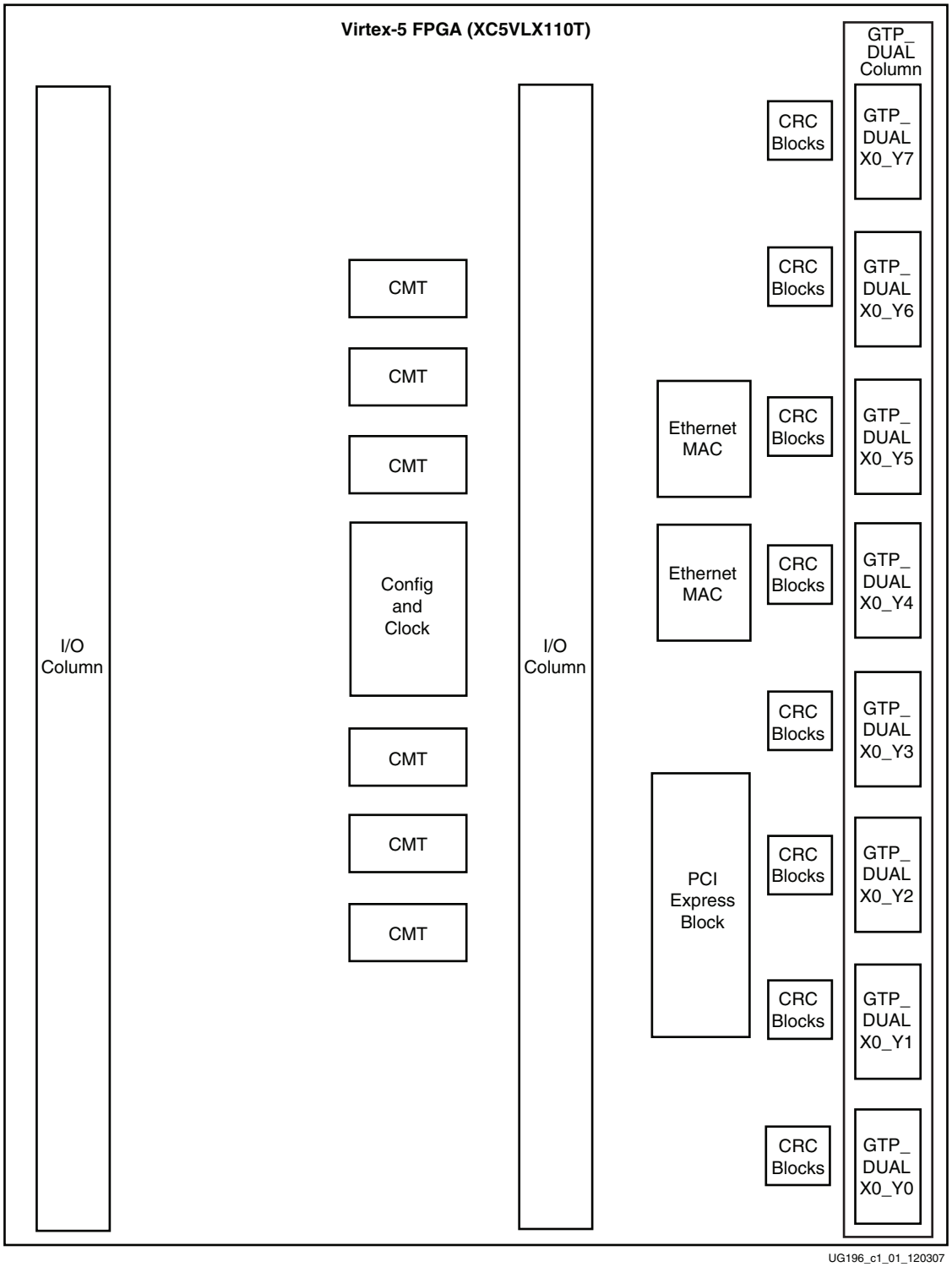
GTP transceivers are placed as dual transceiver GTP_DUAL tiles in Virtex-5 LXT and SXT platform devices. This configuration allows two transceivers to share a single PLL with the TX and RX functions of both, reducing size and power consumption.

Figure 1-1 shows GTP_DUAL tile placement in an example Virtex-5 device (XC5VLX110T). All the GTP_DUAL tiles form a single GTP_DUAL column on the right side of the device

Power Ex1009
Page 24 of 347

as shown in Figure 1-1. Additional information on the functional blocks in Figure 1-1 is available in the following locations:

- Chapter 8, Cyclic Redundancy Check, provides more details on the CRC blocks in Figure 1-1.
- The *Virtex-5 FPGA Configuration Guide* provides more information on the Config and Clock, CMT, and I/O blocks.
- The *Virtex-5 Embedded Tri-Mode Ethernet MAC User Guide* provides detailed information on the Ethernet MAC.
- The *Virtex-5 FPGA Integrated Endpoint Block User Guide for PCI Express Designs* provides detailed information on PCI Express compliance.



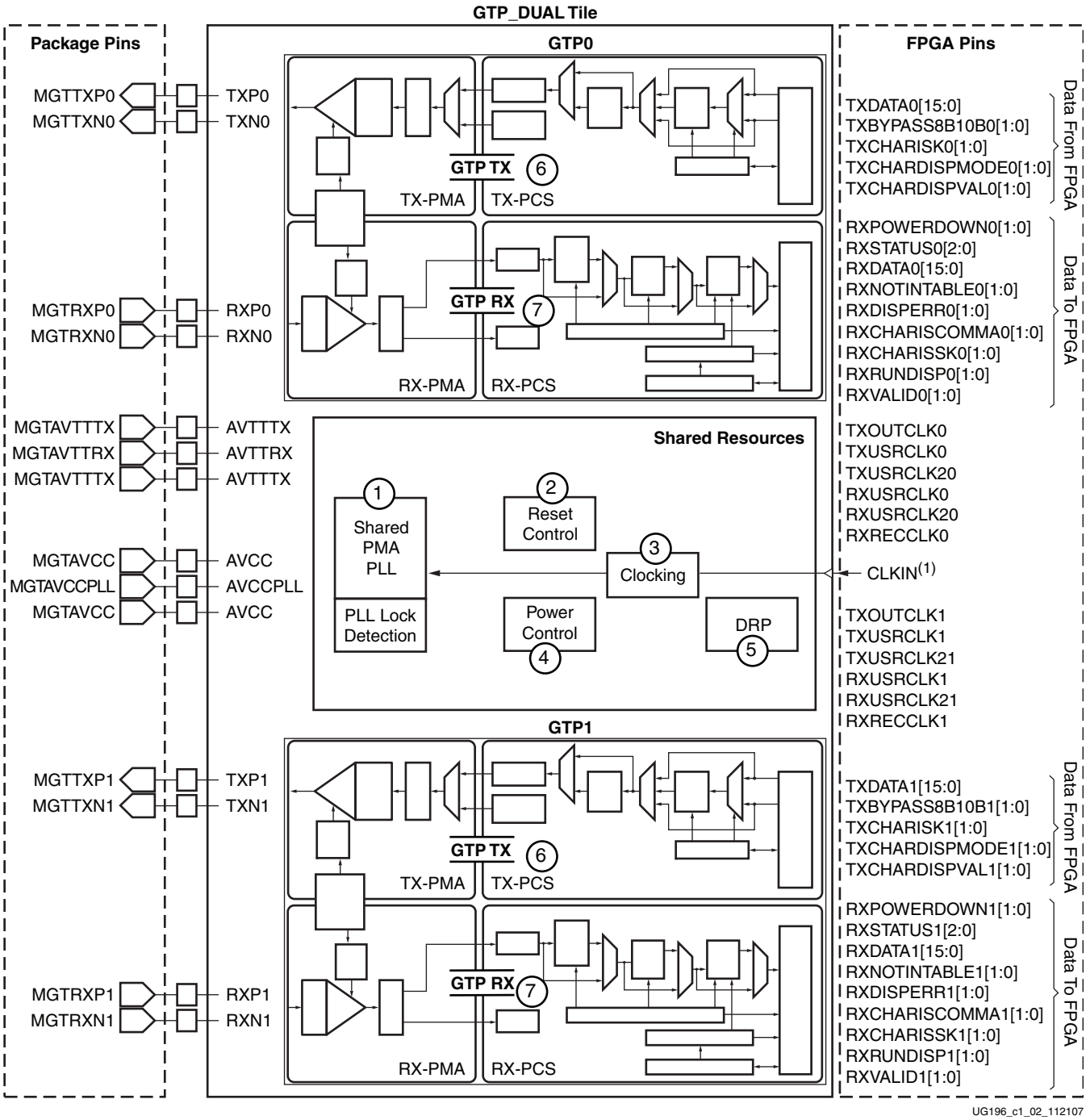
Notes:

1. This figure does *not* illustrate exact size, location, or scale of the functional blocks to each other. It does show the correct number of available resources.
2. To improve clarity, this figure does *not* show the CLB, DSP, and Block RAM columns.

Figure 1-1: **GTP_DUAL Tile Inside the Virtex-5 XC5VLX110T FPGA**

Cypher Ex1009
Page 26 of 347

Figure 1-2 shows a diagram of a GTP_DUAL tile, containing two GTP transceivers and a shared resources block. The GTP_DUAL tile is the HDL primitive used to operate GTP transceivers in the FPGA.



Notes:

1. CLKIN is a simplification for a clock source. See Figure 5-3, page 81 for details on CLKIN.

Figure 1-2: **GTP_DUAL Tile Block Diagram**

The procedures for configuring and using each of the seven major blocks in the GTP_DUAL tile shown in Figure 1-2 are discussed in detail in these sections:

1. Shared PMA PLL (Chapter 5)
2. Reset, page 84 (Chapter 5)
3. Clocking, page 80 (Chapter 5)
4. Power Control, page 94 (Chapter 5)
5. Dynamic Reconfiguration Port, page 101 (Chapter 5)
6. GTP Transmitter (TX), page 103 (Chapter 6)
7. GTP Receiver (RX), page 137 (Chapter 7)

Ports and Attributes

This section contains alphabetical tables of pins (Table 1-2), ports (Table 1-3 and Table 1-4), and attributes (Table 1-5 and Table 1-6). In all Port and Attribute tables in this guide, names that end with 0 are for the GTP0 transceiver on the tile, and names that end with 1 are for the GTP1 transceiver. Names that do not end with 0 or 1 are shared.

Table 1-2 lists alphabetically the signal names, directions, and descriptions of the GTP_DUAL analog pins and provides links to their detailed descriptions.

Table 1-2: **GTP_DUAL Analog Pin Summary**

Pin	Dir	Description	Section (Page)
MGTAVCCPLL	In	Analog supply for the shared PMA PLL, the clock routing, and the muxing network of the GTP_DUAL tile.	Analog Design Guidelines (page 219)
MGTAVTTRX	In	Analog supply for the receiver circuits and the termination of the GTP_DUAL tile.	Analog Design Guidelines (page 219)
MGTAVTTRXC	In	Analog supply for the resistor calibration and the standby circuit of the entire device.	Analog Design Guidelines (page 219)
MGTAVTTTX	In	Analog supply for the transmitter termination and driver circuits of the GTP_DUAL tile.	Analog Design Guidelines (page 220)
MGTAVCC	In	Analog supply for the internal analog circuits of the GTP_DUAL tile.	Analog Design Guidelines (page 219)
MGTREFCLKP MGTREFCLKN	In	Differential clock input pin pair for the reference clock of the GTP_DUAL tile.	Analog Design Guidelines (page 220)
MGTRREF	In	Reference resistor input for the entire device.	Analog Design Guidelines (page 220)

Cypher Ex1009
Page 28 of 347

Table 1-2: **GTP_DUAL Analog Pin Summary *(Continued)***

Pin	Dir	Description	Section (Page)
MGTRXN0 MGTRXP0 MGTRXN1 MGTRXP1	In (Pad)	Differential complements forming a differential receiver input pair for each transceiver.	RX Termination and Equalization (page 139)
MGTTXN0 MGTTXP0 MGTTXN1 MGTTXP1	Out (Pad)	Differential complements forming a differential transmitter output pair for each transceiver.	Configurable TX Driver (page 127)

Table 1-3 lists alphabetically the signal names, clock domains, directions, and descriptions for the GTP_DUAL ports, and provides links to their detailed descriptions.

Table 1-3: **GTP_DUAL Port Summary**

Port	Dir	Domain	Description	Section (Page)
CLKIN	In	Async	Reference clock input to the shared PMA PLL.	Shared PMA PLL (page 73), Clocking (page 82), Power Control (page 94)
DADDR[6:0]	In	DCLK	DRP address bus.	Dynamic Reconfiguration Port (page 101)
DCLK	In	N/A	DRP interface clock.	Dynamic Reconfiguration Port (page 101)
DEN	In	DCLK	Enables DRP read or write operations.	Dynamic Reconfiguration Port (page 101)
DI[15:0]	In	DCLK	Data bus for writing configuration data from the FPGA logic to the GTP_DUAL tile.	Dynamic Reconfiguration Port (page 101)
DO[15:0]	Out	DCLK	Data bus for reading configuration data from the GTP_DUAL tile to the FPGA logic.	Dynamic Reconfiguration Port (page 101)
DRDY	Out	DCLK	Indicates the operation is complete for DRP write operations or data is valid for DRP read operations.	Dynamic Reconfiguration Port (page 101)
DWE	In	DCLK	Indicates whether the DRP operation is a read or a write.	Dynamic Reconfiguration Port (page 101)
GTPRESET	In	Async	Starts the full GTP_DUAL reset sequence.	Reset (page 85)
GTPTEST[3:0]	In	Async	Factory test pins. Must be strapped Low for normal operation.	

Optiver Ex1009
Page 29 of 347

Table 1-3: **GTP_DUAL Port Summary *(Continued)***

Port	Dir	Domain	Description	Section (Page)
INTDATAWIDTH	In	Async	Sets the internal datapath width for the GTP_DUAL tile. 0: 8-bit internal datapath width 1: 10-bit internal datapath width	Shared PMA PLL (page 73), FPGA TX Interface (page 104), TX PRBS Generator (page 123), Parallel In to Serial Out (page 124), Serial In to Parallel Out (page 155), PRBS Detection (page 161), Configurable RX Elastic Buffer and Phase Alignment (page 177), Configurable Clock Correction (page 184), Configurable Channel Bonding (Lane Deskew) (page 190), FPGA RX Interface (page 200)
LOOPBACK0[2:0] LOOPBACK1[2:0]	In	Async	Sets the loopback mode.	Loopback (page 214)
PHYSTATUS0 PHYSTATUS1	Out	Async	Indicates completion of several PHY functions, including power management state transitions and receiver detection.	Receive Detect Support for PCI Express Operation (page 130)
PLLLKDET	Out	Async	Indicates that the VCO rate is within acceptable tolerances of the desired rate.	Shared PMA PLL (page 73)
PLLLKDETEN	In	Async	Enables the PLL lock detector.	Shared PMA PLL (page 73)
PLLPOWERDOWN	In	Async	Powers down the shared PMA PLL.	Reset (page 85), Power Control (page 94)
PRBSCNTRESET0 PRBSCNTRESET1	In	RXUSRCLK2	Resets the PRBS error counter.	Reset (page 85), PRBS Detection (page 161)
REFCLKOUT	Out	N/A	Provides access to the reference clock provided to the shared PMA PLL (CLKIN).	Shared PMA PLL (page 73), Clocking (page 82), FPGA TX Interface (page 104), TX Buffering, Phase Alignment, and TX Skew Reduction (page 117), FPGA RX Interface (page 200)
REFCLKPWRDNB	In	Async	Powers down the GTP reference clock circuit (active Low).	Power Control (page 94)

Power Ex1009
Page 30 of 347

Table 1-3: **GTP_DUAL Port Summary *(Continued)***

Port	Dir	Domain	Description	Section (Page)
RESETDONE0 RESETDONE1	Out	Async	Indicates when the GTP transceiver has finished reset and is ready for use.	Reset (page 84), RX Clock Data Recovery (page 149)
RXBUFRESET0 RXBUFRESET1	In	Async	Resets the RX buffer logic.	Reset (page 85), Configurable RX Elastic Buffer and Phase Alignment (page 177), Configurable Clock Correction (page 184)
RXBUFSTATUS0[2:0] RXBUFSTATUS1[2:0]	Out	RXUSRCLK2	Indicates the overflow/underflow status of the RX buffer.	Configurable RX Elastic Buffer and Phase Alignment (page 178), Configurable Clock Correction (page 184)
RXBYTEISALIGNED0 RXBYTEISALIGNED1	Out	RXUSRCLK2	Indicates if the parallel data stream is properly aligned on byte boundaries according to comma detection. When PCOMMA_DETECT_(0/1) = TRUE, asserted for alignment to PCOMMA value. When MCOMMA_DETECT_(0/1) = TRUE, asserted for alignment to MCOMMA value.	Configurable Comma Alignment and Detection (page 162)
RXBYTEREALIGN0 RXBYTEREALIGN1	Out	RXUSRCLK2	Indicates if the byte alignment within the serial data stream has changed due to a comma detection.	Configurable Comma Alignment and Detection (page 164)
RXCDRRESET0 RXCDRRESET1	In	RXUSRCLK2	Reset for the RX CDR. Also resets the rest of the RX PCS.	Reset (page 85), RX Clock Data Recovery (page 149)
RXCHANBONDSEQ0 RXCHANBONDSEQ1	Out	RXUSRCLK2	Indicates when RXDATA contains the start of a channel bonding sequence.	Configurable Channel Bonding (Lane Deskew) (page 190)
RXCHANISALIGNED0 RXCHANISALIGNED1	Out	RXUSRCLK2	Indicates if the channel is properly aligned with the master transceiver according to observed channel bonding sequences in the data stream.	Configurable Channel Bonding (Lane Deskew) (page 190)
RXCHANREALIGN0 RXCHANREALIGN1	Out	RXUSRCLK2	Held High for at least one cycle when the receiver has changed.	Configurable Channel Bonding (Lane Deskew) (page 190)
RXCHARISCOMMA0[1:0] RXCHARISCOMMA1[1:0]	Out	RXUSRCLK2	Asserted when RXDATA is an 8B/10B comma.	Configurable 8B/10B Decoder (page 172)
RXCHARISK0[1:0] RXCHARISK1[1:0]	Out	RXUSRCLK2	Asserted when RXDATA is an 8B/10B K character.	Configurable 8B/10B Decoder (page 172)

Optiver Ex1009
Page 31 of 347

Table 1-3: **GTP_DUAL Port Summary *(Continued)***

Port	Dir	Domain	Description	Section (Page)
RXCHBONDI0[2:0] RXCHBONDI1[2:0]	In	RXUSRCLK	FPGA channel bonding control. Used only by slaves.	Configurable Channel Bonding (Lane Deskew) (page 190)
RXCHBONDO0[2:0] RXCHBONDO1[2:0]	Out	RXUSRCLK	FPGA channel bonding control.	Configurable Channel Bonding (Lane Deskew) (page 190)
RXCLKCORCNT0[2:0] RXCLKCORCNT1[2:0]	Out	RXUSRCLK2	Reports the status of the elastic buffer clock correction.	Configurable Clock Correction (page 184)
RXCOMMADET0 RXCOMMADET1	Out	RXUSRCLK2	Asserted when the comma alignment block detects a comma.	Configurable Comma Alignment and Detection (page 164)
RXCOMMADETUSE0 RXCOMMADETUSE1	In	RXUSRCLK2	Activates the comma detection and alignment circuit.	Configurable Comma Alignment and Detection (page 164)
RXDATA0 RXDATA1	Out	RXUSRCLK2	Receive data bus of the receive interface to the FPGA.	FPGA RX Interface (page 200)
RXDATAWIDTH0 RXDATAWIDTH1	In	RXUSRCLK2	Selects the width of the RXDATA receive data connection to the FPGA.	FPGA RX Interface (page 200)
RXDEC8B10BUSE0 RXDEC8B10BUSE1	In	RXUSRCLK2	Enables the 8B/10B decoder.	Configurable 8B/10B Decoder (page 172)
RXDISPERR0[1:0] RXDISPERR1[1:0]	Out	RXUSRCLK2	Indicates if RXDATA was received with a disparity error.	Configurable 8B/10B Decoder (page 172)
RXELECIDLE0 RXELECIDLE1	Out	Async	Indicates the differential voltage between RXN and RXP dropped below the minimum threshold.	RX OOB/Beacon Signaling (page 144)
RXELECIDLERESET0 RXELECIDLERESET1	In	Async	Resets the RX Clock Data Recovery circuit, used by the mandatory Link Idle Reset circuit.	Reset (page 85), RX Clock Data Recovery (page 149)
RXENCHANSYNC0 RXENCHANSYNC1	In	RXUSRCLK2	Enables channel bonding.	Configurable Channel Bonding (Lane Deskew) (page 190)
RXENELECIDLERESETB	In	Async	Enables the RXELECIDLERESET inputs, used by the mandatory Link Idle Reset circuit (active Low).	Reset (page 85), RX Clock Data Recovery (page 149)
RXENEQB0 RXENEQB1	In	Async	Enables receiver equalization (active Low).	RX Termination and Equalization (page 139)
RXENMCOMMAALIGN0 RXENMCOMMAALIGN1	In	RXUSRCLK2	Aligns the byte boundary when comma minus is detected.	Configurable Comma Alignment and Detection (page 164)

Apotex Ex1009
Page 32 of 347

Table 1-3: **GTP_DUAL Port Summary *(Continued)***

Port	Dir	Domain	Description	Section (Page)
RXENPCOMMAALIGN0 RXENPCOMMAALIGN1	In	RXUSRCLK2	Aligns the byte boundary when comma plus is detected.	Configurable Comma Alignment and Detection (page 164)
RXENPRBSTST0[1:0] RXENPRBSTST1[1:0]	In	RXUSRCLK2	Receiver test pattern checker control.	PRBS Detection (page 161)
RXENSAMPLEALIGN0 RXENSAMPLEALIGN1	In	RXUSRCLK2	When High, the 5x oversampler in the PCS continually adjusts its sample point. When Low, it samples only at the point that was active before the port went Low.	Oversampling (page 157)
RXEQMIX0[1:0] RXEQMIX1[1:0]	In	Async	Sets the wideband/high-pass mix ratio for the RX equalizer.	RX Termination and Equalization (page 139)
RXEQPOLE0[3:0] RXEQPOLE1[3:0]	In	Async	Sets high-pass filter pole location for the RX equalizer.	RX Termination and Equalization (page 139)
RXLOSSOFSYNC0[1:0] RXLOSSOFSYNC1[1:0]	Out	RXUSRCLK2	FPGA status related to byte stream synchronization, depending on the state of the RX_LOSS_OF_SYNC_FSM attribute.	Configurable Loss-of-Sync State Machine (page 170)
RXNOTINTABLE0[1:0] RXNOTINTABLE1[1:0]	Out	RXUSRCLK2	Indicates if RXDATA is the result of an illegal 8B/10B code and is in error.	Configurable 8B/10B Decoder (page 172)
RXOVERSAMPLEERR0 RXOVERSAMPLEERR1	Out	RXUSRCLK2	Indicates the FIFO in oversampling circuit has either overflowed or underflowed.	Oversampling (page 157)
RXPMASETPHASE0 RXPMASETPHASE1	In	RXUSRCLK2	Aligns the PMA receiver recovered clock with the PCS user clocks, allowing the RX elastic buffer to be bypassed.	Configurable RX Elastic Buffer and Phase Alignment (page 178)
RXPOLARITY0 RXPOLARITY1	In	RXUSRCLK2	Inverts the polarity of incoming data.	RX Polarity Control (page 160)
RXPOWERDOWN0[1:0] RXPOWERDOWN1[1:0]	In	Async	Powers down the RX lanes.	Power Control (page 94), Receive Detect Support for PCI Express Operation (page 131)
RXPRBSERR0 RXPRBSERR1	Out	RXUSRCLK2	Indicates if the number of errors in PRBS testing equals or exceeds the value set by the PRBS_ERR_THRESHOLD attribute.	PRBS Detection (page 161)
RXRECCLK0 RXRECCLK1	Out	N/A	Recovered clocks derived from the RX Clock Data Recovery circuit. Clocks the RX logic between the PMA and the RX elastic buffer.	FPGA RX Interface (page 200)
RXRESET0 RXRESET1	In	Async	Active-High reset for the RX PCS logic.	Reset (page 85), FPGA RX Interface (page 200)

Table 1-3: **GTP_DUAL Port Summary *(Continued)***

Port	Dir	Domain	Description	Section (Page)
RXRUNDISP0[1:0] RXRUNDISP1[1:0]	Out	RXUSRCLK2	Shows the running disparity of the 8B/10B encoder when RXDATA is received.	Configurable 8B/10B Decoder (page 172)
RXSLIDE0 RXSLIDE1	In	RXUSRCLK2	Implements a comma alignment bump control, allowing manual comma alignment.	Configurable Comma Alignment and Detection (page 164)
RXSTATUS0[2:0] RXSTATUS1[2:0]	Out	RXUSRCLK2	Shows the status of PCI Express or SATA operations. The decoding depends on the setting of RX_STATUS_FMT.	TX OOB/Beacon Signaling (page 133), RX OOB/Beacon Signaling (page 144), Receive Detect Support for PCI Express Operation (page 130)
RXUSRCLK20 RXUSRCLK21	In	N/A	Input clock used for the interface between the FPGA and the GTP transceiver.	FPGA RX Interface (page 201)
RXUSRCLK0 RXUSRCLK1	In	N/A	Input clock used for internal RX logic after the RX elastic buffer.	FPGA RX Interface (page 201)
RXVALID0 RXVALID1	Out	RXUSRCLK2	Indicates symbol lock and valid data on RXDATA and RXCHARISK[1:0] for PCI Express operations.	RX OOB/Beacon Signaling (page 144)
TXBUFDIFFCTRL0[2:0] TXBUFDIFFCTRL1[2:0]	In	Async	Controls the strength of the TX pre-drivers. Tie this port to the same value as TXDIFFCTRL.	Configurable TX Driver (page 127)
TXBUFSTATUS0[1:0] TXBUFSTATUS1[1:0]	Out	TXUSRCLK2	TX buffer status. Indicates TX buffer overflow or underflow.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 117)
TXBYPASS8B10B0[1:0] TXBYPASS8B10B1[1:0]	In	TXUSRCLK2	Controls the operation of the TX 8B/10B encoder on a per-byte basis.	Configurable 8B/10B Encoder (page 112)
TXCHARDISPMODE0[1:0] TXCHARDISPMODE1[1:0]	In	TXUSRCLK2	TXCHARDISPMODE and TXCHARDISPVAL allow the 8B/10B disparity of outgoing data to be controlled when 8B/10B encoding is enabled. When 8B/10B encoding is disabled, TXCHARDISPMODE is used to extend the data bus for TX interfaces whose width is a multiple of 10.	Configurable 8B/10B Encoder (page 112)
TXCHARDISPVAL0[1:0] TXCHARDISPVAL1[1:0]	In	TXUSRCLK2	TXCHARDISPVAL and TXCHARDISPMODE allow the disparity of outgoing data to be controlled when 8B/10B encoding is enabled. When 8B/10B encoding is disabled, TXCHARDISPVAL is used to extend the data bus for 10- and 20-bit TX interfaces.	Configurable 8B/10B Encoder (page 112)
TXCHARISK0[1:0] TXCHARISK1[1:0]	In	TXUSRCLK2	Set High to send TXDATA as an 8B/10B K character.	Configurable 8B/10B Encoder (page 112)

Apotex Ex1009
Page 34 of 347

Table 1-3: **GTP_DUAL Port Summary *(Continued)***

Port	Dir	Domain	Description	Section (Page)
TXCOMSTART0 TXCOMSTART1	In	TXUSRCLK2	Initiates the transmission of the COM sequence selected by TXCOMTYPE (SATA only).	TX OOB/Beacon Signaling (page 133)
TXCOMTYPE0 TXCOMTYPE1	In	TXUSRCLK2	Selects the type of COM signal to send (SATA only).	TX OOB/Beacon Signaling (page 133)
TXDATA0 TXDATA1	In	TXUSRCLK2	Transmitting data bus.	FPGA TX Interface (page 104)
TXDATAWIDTH0 TXDATAWIDTH1	In	TXUSRCLK2	Selects the width of the TXDATA port.	FPGA TX Interface (page 104)
TXDETECTRX0 TXDETECTRX1	In	TXUSRCLK2	Depending on the state of RXPOWERDOWN and TXPOWERDOWN, TXDETECTRX activates the receive detection feature for PCI Express operations or selects PCI Express compliant loopback mode.	Power Control (page 94), Receive Detect Support for PCI Express Operation (page 131)
TXDIFFCTRL0[2:0] TXDIFFCTRL1[2:0]	In	Async	Controls the transmitter differential output swing.	Configurable TX Driver (page 127)
TXELECIDLE0 TXELECIDLE1	In	TXUSRCLK2	Drives TXN and TXP to the same voltage to perform electrical idle/beaconing for PCI Express operations.	Power Control (page 94), TX OOB/Beacon Signaling (page 133)
TXENC8B10BUSE0 TXENC8B10BUSE1	In	TXUSRCLK2	Enables the 8B/10B encoder.	Configurable 8B/10B Encoder (page 113), FPGA TX Interface (page 104)
TXENPMAPHASEALIGN	In	Async	Allows both GTP transceivers in a GTP_DUAL tile to align their XCLKs with their TXUSRCLKs, allowing their TX buffers to be bypassed, and allows the XCLKs in multiple GTP transceivers to be synchronized.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 117)
TXENPRBSTST0[1:0] TXENPRBSTST1[1:0]	In	TXUSRCLK2	Transmitter test pattern generation control.	TX PRBS Generator (page 123)
TXINHIBIT0 TXINHIBIT1	In	TXUSRCLK2	Inhibits data transmission.	
TXKERR0[1:0] TXKERR1[1:0]	Out	TXUSRCLK2	Indicates if an invalid code for a K character was specified.	Configurable 8B/10B Encoder (page 113)
TXOUTCLK0 TXOUTCLK1	Out	N/A	Provides a parallel clock generated by the internal dividers of the GTP transceiver. ***Note:*** When INTDATAWIDTH is High, the duty cycle is 60/40 instead of 50/50. TXOUTCLK cannot drive TXUSRCLK when the TX phase-alignment circuit is used.	FPGA TX Interface (page 105), TX Buffering, Phase Alignment, and TX Skew Reduction (page 117)

Table 1-3: **GTP_DUAL Port Summary *(Continued)***

Port	Dir	Domain	Description	Section (Page)
TXPMASETPHASE	In	Async	Aligns XCLK with TXUSRCLK for both GTP transceivers in the GTP_DUAL tile.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 117)
TXPOLARITY0 TXPOLARITY1	In	TXUSRCLK2	Specifies if the final transmitter output is inverted.	TX Polarity Control (page 122)
TXPOWERDOWN0[1:0] TXPOWERDOWN1[1:0]	In	Async	Powers down the TX lanes.	Power Control (page 95), Receive Detect Support for PCI Express Operation (page 131), TX OOB/Beacon Signaling (page 133)
TXPREEMPHASIS0[2:0] TXPREEMPHASIS1[2:0]	In	Async	Controls the relative strength of the main drive and the pre-emphasis.	Configurable TX Driver (page 127)
TXRESET0 TXRESET1	In	Async	Resets the PCS of the GTP transmitter, including the phase adjust FIFO, the 8B/10B encoder, and the FPGA TX interface.	Reset (page 85), FPGA TX Interface (page 105)
TXRUNDISP0[1:0] TXRUNDISP1[1:0]	Out	TXUSRCLK2	Indicates the current running disparity of the 8B/10B encoder.	Configurable 8B/10B Encoder (page 113)
TXUSRCLK0 TXUSRCLK1	In	N/A	Provides a clock for the internal TX PCS datapath.	FPGA TX Interface (page 105), TX Buffering, Phase Alignment, and TX Skew Reduction (page 118)
TXUSRCLK20 TXUSRCLK21	In	N/A	Synchronizes the FPGA logic with the TX interface.	FPGA TX Interface (page 105)

Cypher Ex1009
Page 36 of 347

Table 1-4 lists alphabetically the signal names, clock domains, directions, and descriptions for the CRC ports, and provides links to their detailed descriptions.

Table 1-4: **CRC Port Summary**

Port	Dir	Domain	Description	Section (Page)
CRCCLK	In	N/A	CRC clock.	Cyclic Redundancy Check (page 206)
CRCDATAVALID	In	CRCCLK	Indicates valid data on the CRCIN inputs.	Cyclic Redundancy Check (page 206)
CRCDATAWIDTH[2:0]	In	CRCCLK	Indicates how many input data bytes are valid.	Cyclic Redundancy Check (page 206)
CRCIN[63:0]	In	CRCCLK	CRC input data. The maximum datapath width is eight bytes.	Cyclic Redundancy Check (page 206)
CRCOUT[31:0]	Out	CRCCLK	32-bit CRC output. CRCOUT is the byte-reversed, bit-inverted CRC value corresponding to the CRC calculation on valid bytes from the previous clock cycle and the previous CRC value. CRCDATAVALID must be driven High.	Cyclic Redundancy Check (page 206)
CRCRESET	In	CRCCLK	Synchronous reset of the CRC registers.	Cyclic Redundancy Check (page 206)

Table 1-5 lists alphabetically the attribute names, default values, and directions of the GTP_DUAL attributes, and provides links to their detailed descriptions.

Table 1-5: **GTP_DUAL Attribute Summary**

Attribute	Description	Section (Page)
AC_CAP_DIS_0 AC_CAP_DIS_1	Disables built-in AC coupling capacitors on receiver inputs when set to TRUE.	RX Termination and Equalization (page 140)
ALIGN_COMMA_WORD_0 ALIGN_COMMA_WORD_1	Controls alignment of detected commas within a multi-byte datapath.	Configurable Comma Alignment and Detection (page 165)
CHAN_BOND_1_MAX_SKEW_0 CHAN_BOND_1_MAX_SKEW_1 CHAN_BOND_2_MAX_SKEW_0 CHAN_BOND_2_MAX_SKEW_1	Sets the maximum amount of lane skew allowed when using channel bonding. Must be set less than one-half the minimum distance between channel bonding sequences.	Configurable Channel Bonding (Lane Deskew) (page 190)
CHAN_BOND_LEVEL_0 CHAN_BOND_LEVEL_1	Indicates the amount of internal pipelining used for the elastic buffer control signals.	Configurable Channel Bonding (Lane Deskew) (page 191)
CHAN_BOND_MODE_0 CHAN_BOND_MODE_1	Defines the channel bonding mode of operation for the transceiver.	Configurable Channel Bonding (Lane Deskew) (page 191)

Table 1-5: **GTP_DUAL Attribute Summary *(Continued)***

Attribute	Description	Section (Page)
CHAN_BOND_SEQ_1_1_0 CHAN_BOND_SEQ_1_1_1 CHAN_BOND_SEQ_1_2_0 CHAN_BOND_SEQ_1_2_1 CHAN_BOND_SEQ_1_3_0 CHAN_BOND_SEQ_1_3_1 CHAN_BOND_SEQ_1_4_0 CHAN_BOND_SEQ_1_4_1	Used in conjunction with CHAN_BOND_SEQ_1_ENABLE to define channel bonding sequence 1.	Configurable Channel Bonding (Lane Deskew) (page 191)
CHAN_BOND_SEQ_1_ENABLE_0 CHAN_BOND_SEQ_1_ENABLE_1	Sets which parts of channel bonding sequence 1 are don't cares.	Configurable Channel Bonding (Lane Deskew) (page 191)
CHAN_BOND_SEQ_2_1_0 CHAN_BOND_SEQ_2_1_1 CHAN_BOND_SEQ_2_2_0 CHAN_BOND_SEQ_2_2_1 CHAN_BOND_SEQ_2_3_0 CHAN_BOND_SEQ_2_3_1 CHAN_BOND_SEQ_2_4_0 CHAN_BOND_SEQ_2_4_1	Used in conjunction with CHAN_BOND_SEQ_2_ENABLE to define the second channel bonding sequence.	Configurable Channel Bonding (Lane Deskew) (page 191)
CHAN_BOND_SEQ_2_ENABLE_0 CHAN_BOND_SEQ_2_ENABLE_1	Sets which parts of channel bonding sequence 2 are don't cares.	Configurable Channel Bonding (Lane Deskew) (page 191)
CHAN_BOND_SEQ_2_USE_0 CHAN_BOND_SEQ_2_USE_1	Determines if the second channel bonding sequence is to be used.	Configurable Channel Bonding (Lane Deskew) (page 191)
CHAN_BOND_SEQ_LEN_0 CHAN_BOND_SEQ_LEN_1	Defines the length in bytes of the channel bonding sequence that the transceiver matches to detect opportunities for channel bonding.	Configurable Channel Bonding (Lane Deskew) (page 193)
CLK_COR_ADJ_LEN_0 CLK_COR_ADJ_LEN_1	Defines the size of the adjustment (number of bytes repeated or skipped) in a clock correction.	Configurable Clock Correction (page 185)
CLK_COR_DET_LEN_0 CLK_COR_DET_LEN_1	Defines the length of the sequence that the transceiver matches to detect opportunities for clock correction.	Configurable Clock Correction (page 185)
CLK_COR_INSERT_IDLE_FLAG_0 CLK_COR_INSERT_IDLE_FLAG_1	Controls whether the RXRUNDISP input status indicates running disparity or inserted-idle (clock correction sequence) flag.	Configurable Clock Correction (page 185)
CLK_COR_KEEP_IDLE_0 CLK_COR_KEEP_IDLE_1	Controls whether the elastic buffer must retain at least one clock correction sequence in the byte stream.	Configurable Clock Correction (page 185)
CLK_COR_MAX_LAT_0 CLK_COR_MAX_LAT_1	Specifies the maximum elastic buffer latency.	Configurable Clock Correction (page 185)

Apotex Ex1009
Page 38 of 347

Table 1-5: **GTP_DUAL Attribute Summary *(Continued)***

Attribute	Description	Section (Page)
CLK_COR_MIN_LAT_0 CLK_COR_MIN_LAT_1	Specifies the minimum elastic buffer latency.	Configurable Clock Correction (page 185)
CLK_COR_PRECEDENCE_0 CLK_COR_PRECEDENCE_1	Determines whether clock correction or channel bonding takes precedence when both operations are triggered at the same time. Set to TRUE to give clock correction precedence.	Configurable Clock Correction (page 185)
CLK_COR_REPEAT_WAIT_0 CLK_COR_REPEAT_WAIT_1	Specifies the minimum number of RXUSRCLK cycles without clock correction that must occur between successive clock corrections.	Configurable Clock Correction (page 185)
CLK_COR_SEQ_1_1_0 CLK_COR_SEQ_1_1_1 CLK_COR_SEQ_1_2_0 CLK_COR_SEQ_1_2_1 CLK_COR_SEQ_1_3_0 CLK_COR_SEQ_1_3_1 CLK_COR_SEQ_1_4_1	The CLK_COR_SEQ_1 attributes are used in conjunction with CLK_COR_SEQ_1_ENABLE to define clock correction sequence 1.	Configurable Clock Correction (page 186)
CLK_COR_SEQ_1_ENABLE_0 CLK_COR_SEQ_1_ENABLE_1	Sets which parts of clock correction sequence 1 are don't cares.	Configurable Clock Correction (page 186)
CLK_COR_SEQ_2_1_0 CLK_COR_SEQ_2_1_1 CLK_COR_SEQ_2_2_0 CLK_COR_SEQ_2_2_1 CLK_COR_SEQ_2_3_0 CLK_COR_SEQ_2_3_1 CLK_COR_SEQ_2_4_0 CLK_COR_SEQ_2_4_1	Used in conjunction with CLK_COR_SEQ_2_ENABLE to define the second clock correction sequence.	Configurable Clock Correction (page 186)
CLK_COR_SEQ_2_ENABLE_0 CLK_COR_SEQ_2_ENABLE_1	Sets which parts of clock correction sequence 2 are don't cares.	Configurable Clock Correction (page 186)
CLK_COR_SEQ_2_USE_0 CLK_COR_SEQ_2_USE_1	Determines if the second clock correction sequence is to be used.	Configurable Clock Correction (page 186)
CLK_CORRECT_USE_0 CLK_CORRECT_USE_1	Set to TRUE to enable clock correction.	Configurable Clock Correction (page 186)
CLK25_DIVIDER	Sets the divider used to divide CLKIN down to an internal rate close to 25 MHz.	Clocking (page 82), Power Control (page 95)
CLKINDC_B	Must be set to TRUE. Oscillators driving the dedicated reference clock inputs must be AC coupled.	Clocking (page 82)
COM_BURST_VAL_0[3:0] COM_BURST_VAL_1[3:0]	Number of bursts transmitted for a SATA COM sequence.	TX OOB/Beacon Signaling (page 134)

Optiver Ex1009
Page 39 of 347

Table 1-5: **GTP_DUAL Attribute Summary *(Continued)***

Attribute	Description	Section (Page)
COMMA_10B_ENABLE_0 COMMA_10B_ENABLE_1	Sets which bits of MCOMMA/PCOMMA must be matched to incoming data and which bits are don't cares.	Configurable Comma Alignment and Detection (page 165)
COMMA_DOUBLE_0 COMMA_DOUBLE_1	When TRUE, a PCOMMA match followed immediately by an MCOMMA match is required for comma detection. Used to detect A1/A2 framing characters for SONET.	Configurable Comma Alignment and Detection (page 165)
DEC_MCOMMA_DETECT_0 DEC_MCOMMA_DETECT_1	Enables detection of negative 8B/10B commas.	Configurable 8B/10B Decoder (page 173)
DEC_PCOMMA_DETECT_0 DEC_PCOMMA_DETECT_1	Enables detection of positive 8B/10B commas.	Configurable 8B/10B Decoder (page 173)
DEC_VALID_COMMA_ONLY_0 DEC_VALID_COMMA_ONLY_1	Limits the set of commas to which RXCHARISCOMMA responds.	Configurable 8B/10B Decoder (page 173)
MCOMMA_10B_VALUE_0 MCOMMA_10B_VALUE_1	Defines comma minus to raise RXCOMMADET and align the parallel data.	Configurable Comma Alignment and Detection (page 165)
MCOMMA_DETECT_0 MCOMMA_DETECT_1	Set to TRUE to allow minus comma detection and alignment.	Configurable Comma Alignment and Detection (page 165)
OOB_CLK_DIVIDER	Sets the squelch clock rate based on CLKIN.	RX OOB/Beacon Signaling (page 145)
OOBDETECT_THRESHOLD_0 OOBDETECT_THRESHOLD_1	Sets the minimum differential voltage between RXN and RXP before a signal is recognized as a valid PCI Express electrical idle or a SATA OOB signal.	RX OOB/Beacon Signaling (page 145)
OVERSAMPLE_MODE	Enables 5x oversampling.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 118), Parallel In to Serial Out (page 125), Serial In to Parallel Out (page 155), Oversampling (page 158), Configurable RX Elastic Buffer and Phase Alignment (page 178)
PCI_EXPRESS_MODE_0 PCI_EXPRESS_MODE_1	Enables specific PCI Express operations.	Power Control (page 95), Configurable Channel Bonding (Lane Deskew) (page 193)
PCOMMA_10B_VALUE_0 PCOMMA_10B_VALUE_1	Defines comma plus to raise RXCOMMADET and align the parallel data.	Configurable Comma Alignment and Detection (page 165)
PCOMMA_DETECT_0 PCOMMA_DETECT_1	Set to TRUE to allow plus comma detection and alignment.	Configurable Comma Alignment and Detection (page 165)
PCS_COM_CFG	Shared PLL configuration settings.	Shared PMA PLL (page 73)

Power Ex1009
Page 40 of 347

Table 1-5: **GTP_DUAL Attribute Summary *(Continued)***

Attribute	Description	Section (Page)
PLL_DIVSEL_FB	Controls the feedback divider of the shared PMA PLL.	Shared PMA PLL (page 73)
PLL_DIVSEL_REF	Controls the reference clock divider of the shared PMA PLL.	Shared PMA PLL (page 73)
PLL_RXDIVSEL_OUT_0 PLL_RXDIVSEL_OUT_1	Defines the nominal line rate for the receiver based on the shared PMA PLL rate.	Shared PMA PLL (page 73), Serial In to Parallel Out (page 155)
PLL_SATA_0 PLL_SATA_1	Tie to FALSE. When FALSE, allows TX SATA operations to work at the SATA Generation 1 (1.5 Gb/s) or SATA Generation 2 (3 Gb/s) rate.	TX OOB/Beacon Signaling (page 134)
PLL_TXDIVSEL_COMM_OUT	Sets a common line rate divider for both GTP transceivers in a tile. Can be used instead of PLL_TXDIVSEL_OUT if both transceivers are using the same TX divider value.	Shared PMA PLL (page 73), TX Buffering, Phase Alignment, and TX Skew Reduction (page 118), Parallel In to Serial Out (page 125), TX OOB/Beacon Signaling (page 134)
PLL_TXDIVSEL_OUT_0 PLL_TXDIVSEL_OUT_1	Sets the divider for the TX line rate for each GTP transceiver.	Shared PMA PLL (page 73), TX Buffering, Phase Alignment, and TX Skew Reduction (page 118), Parallel In to Serial Out (page 125), TX OOB/Beacon Signaling (page 134)
PMA_CDR_SCAN_0 PMA_CDR_SCAN_1	Allows direct control of the CDR sampling point.	RX Clock Data Recovery (page 150)
PMA_COM_CFG	Common configuration attribute for the PMA.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 118), Loopback (page 214), Marginal Conditions and Limitations (page 215)
PMA_RX_CFG_0 PMA_RX_CFG_1	Adjusts CDR operation for oversampling and PLL_RXDIVSEL_OUT settings.	RX Clock Data Recovery (page 150)
PRBS_ERR_THRESHOLD_0 PRBS_ERR_THRESHOLD_1	Sets the error threshold for the PRBS checker.	PRBS Detection (page 161)
RCV_TERM_GND_0 RCV_TERM_GND_1	Sets the RX termination voltage to GND. Used with internal and external AC coupling to support TXDETECTRX functionality for PCI Express operation.	RX Termination and Equalization (page 140)
RCV_TERM_MID_0 RCV_TERM_MID_1	Activates the internal RX termination voltage. Set to TRUE when RX built-in AC coupling is used.	RX Termination and Equalization (page 140)
RCV_TERM_VTTRX_0 RCV_TERM_VTTRX_1	Sets the RX termination voltage to MGTAVTTRX.	RX Termination and Equalization (page 140)

Table 1-5: **GTP_DUAL Attribute Summary *(Continued)***

Attribute	Description	Section (Page)
RX_BUFFER_USE_0 RX_BUFFER_USE_1	Set to TRUE to use the RX elastic buffer.	Configurable RX Elastic Buffer and Phase Alignment (page 178)
RX_DECODE_SEQ_MATCH_0 RX_DECODE_SEQ_MATCH_1	Determines whether sequences are matched against 8B/10B decoded data or undecoded data.	Configurable Clock Correction (page 186)
RX_LOS_INVALID_INCR_0 RX_LOS_INVALID_INCR_1	Defines the number of valid characters required to decrement the error count by 1 for the purpose of loss-of-sync determination.	Configurable Loss-of-Sync State Machine (page 170)
RX_LOS_THRESHOLD_0 RX_LOS_THRESHOLD_1	Defines the error count required to move the Loss of Sync state machine from the SYNC_ACQUIRED to the SYNC_LOST state.	Configurable Loss-of-Sync State Machine (page 170)
RX_LOSS_OF_SYNC_FSM_0 RX_LOSS_OF_SYNC_FSM_1	Defines the behavior of the RXLOSSOFSYNC outputs.	Configurable Loss-of-Sync State Machine (page 170)
RX_SLIDE_MODE_0 RX_SLIDE_MODE_1	Selects between sliding in the PMA or in the PCS.	Configurable Comma Alignment and Detection (page 166)
RX_STATUS_FMT_0 RX_STATUS_FMT_1	Sets whether the RX_STATUS port is used to report the status of PCI Express or SATA features.	RX OOB/Beacon Signaling (page 145)
RX_XCLK_SEL_0 RX_XCLK_SEL_1	Selects which clock is used on the PMA side of the RX elastic buffer. The default setting is RXREC (RX recovered clock). RXUSR (RX USRCLK) should be used when bypassing the RX elastic buffer.	Configurable RX Elastic Buffer and Phase Alignment (page 178)
SATA_BURST_VAL_0 SATA_BURST_VAL_1	Number of bursts required for the SATA OOB detector to declare a COM match.	RX OOB/Beacon Signaling (page 145)
SATA_IDLE_VAL_0 SATA_IDLE_VAL_1	Number of idles required for the SATA OOB detector to declare a COM match.	RX OOB/Beacon Signaling (page 145)
SATA_MAX_BURST_0 SATA_MAX_BURST_1	Sets the threshold for the SATA detector to reject a burst in terms of squelch clock cycles.	RX OOB/Beacon Signaling (page 145)
SATA_MAX_INIT_0 SATA_MAX_INIT_1	Sets the maximum time allowed for a COMINIT/COMRESET idle for the SATA detector in terms of squelch clock cycles.	RX OOB/Beacon Signaling (page 145)
SATA_MAX_WAKE_0 SATA_MAX_WAKE_1	Sets the maximum time allowed for a COMWAKE idle for the SATA detector in terms of squelch clock cycles.	RX OOB/Beacon Signaling (page 146)
SATA_MIN_BURST_0 SATA_MIN_BURST_1	Sets the threshold for the SATA detector to reject a burst in terms of squelch clock cycles.	RX OOB/Beacon Signaling (page 146)

Apotex Ex1009
Page 42 of 347

Table 1-5: **GTP_DUAL Attribute Summary *(Continued)***

Attribute	Description	Section (Page)
SATA_MIN_INIT_0 SATA_MIN_INIT_1	Sets the minimum time allowed for a COMINIT/COMRESET idle for the SATA detector in terms of squelch clock cycles.	RX OOB/Beacon Signaling (page 146)
SATA_MIN_WAKE_0 SATA_MIN_WAKE_1	Sets the minimum time allowed for a COMWAKE idle for the SATA detector in terms of squelch clock cycles.	RX OOB/Beacon Signaling (page 146)
SIM_GTPRESET_SPEEDUP	Shortens the time it takes to finish the GTPRESET sequence and PLL lock during simulation.	Simulation (page 49)
SIM_MODE	This simulation-only attribute chooses between FAST and LEGACY simulation models.	Simulation (page 49)
SIM_PLL_PERDIV2	Specifies the length of one symbol in picoseconds for simulation.	Simulation (page 49)
SIM_RECEIVER_DETECT_PASS0 SIM_RECEIVER_DETECT_PASS1	Controls the receiver detect function.	Simulation (page 49)
TERMINATION_CTRL[4:0]	Controls internal termination calibration circuit.	Analog Design Guidelines, (page 220)
TERMINATION_IMP_0 TERMINATION_IMP_1	Selects the termination impedance for the TX driver and RX receiver.	Analog Design Guidelines, (page 220)
TERMINATION_OVRD	Selects whether the external 50Ω precision resistor, connected to the MGTRREF pin, or an override value is used, as defined by TERMINATION_CTRL.	Analog Design Guidelines, (page 220)
TRANS_TIME_FROM_P2_0 TRANS_TIME_FROM_P2_1	Transition time from the P2 power-down state in internal 25 MHz clock cycles. The exact time depends on the CLKIN rate and the setting of CLK25_DIVIDER.	Power Control (page 95)
TRANS_TIME_NON_P2_0 TRANS_TIME_NON_P2_1	Transition time to or from any power-down state except P2 in internal 25 MHz clock cycles. The exact time depends on the CLKIN rate and the setting of CLK25_DIVIDER.	Power Control (page 95)
TRANS_TIME_TO_P2_0 TRANS_TIME_TO_P2_1	Transition time to the P2 power-down state in internal 25 MHz clock cycles. The exact time depends on the CLKIN rate and the setting of CLK25_DIVIDER.	Power Control (page 95)
TX_BUFFER_USE_0 TX_BUFFER_USE_1	Indicates whether or not the TX buffer is used.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 118)

Table 1-5: **GTP_DUAL Attribute Summary *(Continued)***

Attribute	Description	Section (Page)
TX_DIFF_BOOST_0 TX_DIFF_BOOST_1	Changes the strength of the TX driver and pre-emphasis buffers. When set to TRUE, the pre-emphasis percentage is *boosted* or increased. See Table 6-19, page 128 for nominal differential swing and pre-emphasis values. Overall differential swing is reduced when TX_DIFF_BOOST is TRUE.	Configurable TX Driver (page 126)
TX_SYNC_FILTERB	This parameter must be left at its default value of 1.	
TX_XCLK_SEL_0 TX_XCLK_SEL_1	Selects the clock used to drive the clock domain in the PCS following the TX buffer. Set to TXOUT (TXOUTCLK) when using the TX buffer. Set to TXUSR (TXUSRCLK) when bypassing the TX buffer.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 119)
TXRX_INVERT_0 TXRX_INVERT_1	Controls inverters that optimize the clock paths within the GTP transceiver. When bypassing the TX buffer, set to `00100`. Otherwise, set to `00000`.	TX Buffering, Phase Alignment, and TX Skew Reduction (page 119)

Notes:

1. Refer to Appendix D for information about the mapping of these attributes to DRP binary values.

Table 1-6 lists the attribute name, default value, and direction of the CRC attribute, and provides a link to the detailed description.

Table 1-6: **CRC Attribute Summary**

Attribute	Type	Description	Section (Page)
CRC_INIT[31:0]	32-bit Hex	32-bit value for initial state of CRC internal registers in the CRC32/CRC64 block.	Cyclic Redundancy Check (page 207)

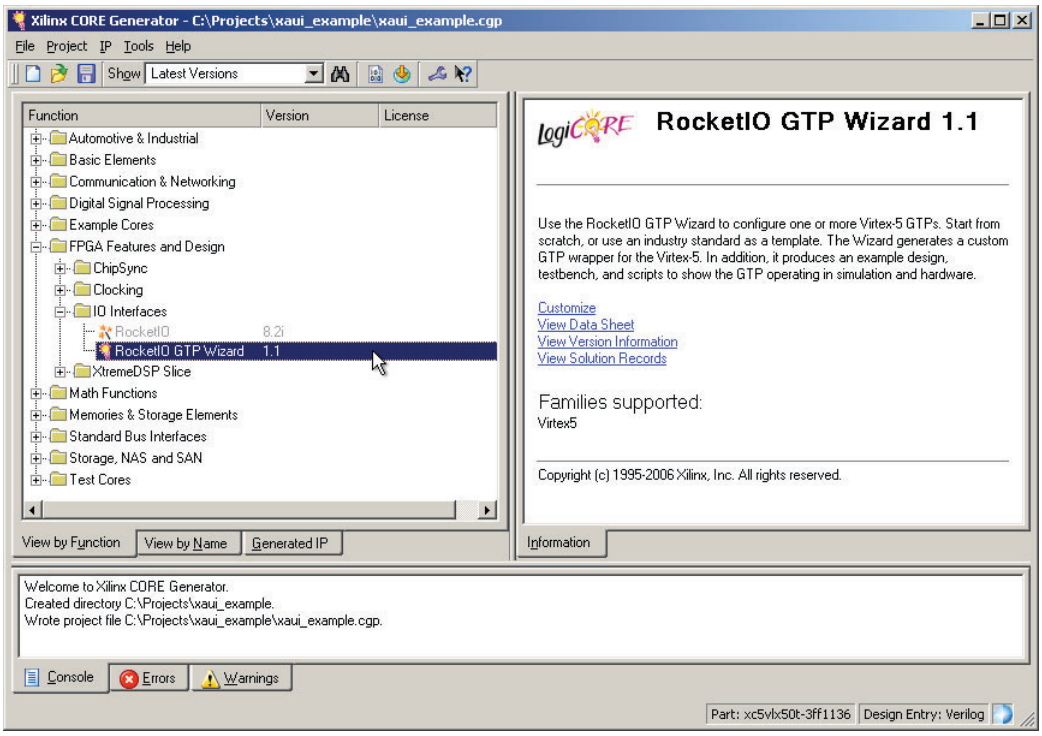
RocketIO GTP Transceiver Wizard

The RocketIO GTP Transceiver Wizard is the preferred tool to generate a wrapper to instantiate a GTP_DUAL primitive. The Wizard can be found in the Xilinx CORE Generator tool. Be sure to download the most up-to-date IP Update before using the Wizard. Details on how to use this wizard can be found in UG188, *RocketIO GTP Transceiver Wizard User Guide*.

1. Start the Xilinx CORE Generator tool.
2. Locate the RocketIO GTP Wizard in the taxonomy tree under:

   `/FPGA Features & Design/IO Interfaces`

   See Figure 2-1.



UG196_c2_01_100406

Figure 2-1: **Locating the RocketIO GTP Wizard**

3. Double click **`RocketIO GTP Wizard`** to launch the Wizard.

Oplover Ex1009
Page 46 of 347

Chapter 3

Simulation

Overview

Simulations using GTP_DUAL tiles have specific prerequisites that the simulation environment and the test bench must fulfill.

The *Synthesis and Simulation Design Guide* [Ref 2] explains how to set up the simulation environment for supported simulators depending on the used Hardware Description Language (HDL). This design guide can be downloaded from the Xilinx website at http://www.xilinx.com/support/software_manuals.htm.

The prerequisites for simulating a design with GTP transceivers using the SmartModels are:

Note: SmartModel support is legacy.

- Simulator with a SWIFT interface to support *SmartModels*, which are encrypted versions of the HDL used for implementation of the modeled block
- Installed GTP_DUAL SmartModel
- Correct setting of the environment variable that points to the SmartModel installation directory
- Correct setup of the simulator for SmartModel use (initialization file, environment variable(s))
- Compilation of the SmartModel wrapper files into the *UNISIM* and *SIMPRIM* libraries
- Compilation of the GTP_DUAL SmartModel into a simulation library
- Correct simulator resolution (Verilog)
- Correct compilation order of simulation libraries

The user guide of the simulator and the *Synthesis and Simulation Design Guide* provide a detailed list of settings for SmartModel support. The COMPXLIB tool with sl_admin facilitates the setup of the supported simulator.

The prerequisites for simulating a design with GTP transceivers using the SecureIP models are:

Note: New designs must use SecureIP models.

- A Verilog LRM-IEEE 1364-2005 encryption compliant simulator with support for SecureIP models. The SecureIP models are encrypted versions of the Verilog HDL used for implementation of the modeled block.
- A mixed-language simulator for VHDL simulation. SecureIP models use a Verilog standard. A mixed-language simulator is required to use SecureIP models in a VHDL design. The simulator must be capable of simulating VHDL and Verilog simultaneously.

Optiver Ex1009
Page 47 of 347

- An installed GTP SecureIP model.
- Correct setup of the simulator for SmartModel use (initialization file, environment variable(s))
- Running of COMPXLIB, which compiles the simulation libraries, such as UNISIM and SIMPRIMS, in the correct order.

The user guide of the simulator and the *Synthesis and Simulation Design Guide* [Ref 2] provide a detailed list of settings for SecureIP support.

Cpuver Ex1009
Page 48 of 347

Ports and Attributes

The GTP_DUAL primitive has attributes intended only for simulation. Table 3-1 lists the *simulation-only* attributes of the GTP_DUAL tile. The names of these attributes start with *SIM_*.

Table 3-1: **GTP_DUAL Simulation-Only Attributes**

Attribute	Description
SIM_GTPRESET_SPEEDUP	This attribute shortens the time it takes to finish the GTPRESET sequence and lock the shared PMA PLL during simulation. `1`: Shorten the GTPRESET cycle time (fast initialization is approximately 300 ns). The value of SIM_PLL_PERDIV2 defines the PLL frequency in this mode. Because SIM_PLL_PERDIV2 cannot be changed on the fly during simulation, this mode cannot be used for multirate designs. `0`: The GTPRESET sequence is simulated with its original duration (standard initialization is approximately 160 µs). This mode must be used for multirate designs.
SIM_MODE	This simulation-only attribute chooses between two available UNISIM/SIMPRIM simulation models. FAST: When this attribute is set to FAST, a faster simulation model of the PMA is used to cut simulation run time. LEGACY: When this attribute is set to LEGACY, the legacy simulation model of the PMA is used, which results in a longer simulation run time.
SIM_PLL_PERDIV2	This attribute specifies a 9-bit hex value equal to half the period of the PLL clock frequency in picoseconds. For example, 400 ps (decimal) is equal to `0x190` (hexadecimal), which is the default value. If SIM_PLL_PERDIV2 is not set correctly, poor locking behavior and incorrect clock frequencies will occur in simulation.
SIM_RECEIVER_DETECT_PASS0 SIM_RECEIVER_DETECT_PASS1	This attribute is used to simulate the TXDETECTRX feature in each GTP transceiver. TRUE (default): Simulates an RX connection to the TX serial ports. TXDETECTRX reports that an RX port is connected. FALSE: Simulates a disconnected TX port. TXDETECTRX reports that the RX port is not detected.

There are no simulation-only ports.

Description

The behavior of the GTP_DUAL tile is modeled using a SmartModel. The SmartModel allows the design containing GTP_DUAL tiles to be simulated in the following design phases:

- Register Transfer Level (RTL)/Pre-Synthesis Simulation
- Post-Synthesis Simulation/Pre-NGDBuild Simulation
- Post-NGDBuild/Pre-Map Simulation
- Post-Map/Partial Timing Simulation
- Post-Place and Route/Timing Simulation

Limitations

The analog nature of some blocks inside the GTP_DUAL tile generates some restrictions when simulated using an HDL simulator. Receiver detection and OOB/beacon signaling are analog features of the GTP_DUAL tile that can only be modeled in a limited way with an HDL simulator. The shared PMA PLL is another analog block in the GTP_DUAL tile that is difficult to model precisely. For this reason, several simulation-only attributes are provided to work around these limitations.

SmartModel Attributes

SIM_GTPRESET_SPEEDUP

The SIM_GTPRESET_SPEEDUP attribute can be used to shorten the simulated lock time of the shared PMA PLL.

If TXOUTCLK or RXRECCLK is used to generate clocks in the design, these clocks occasionally flatline while the GTP_DUAL tile is locking. If a PLL or digital clock manager (DCM) is used to divide TXOUTCLK or RXRECCLK, the final output clock is not ready until both the GTP_DUAL tile and the PLL or DCM have locked. Equation 3-1 provides an estimate of the time required before a stable source from TXOUTCLK or RXRECCLK is available in simulation, including the time required for any PLLs or DCMs used.

$$t_{USRCLKstable} \cong t_{GTPRESETsequence} + t_{locktimePLL} + t_{locktimeDCM} \qquad \textit{Equation 3-1}$$

If either the PLL or the DCM is not used, the respective term can be removed from the lock time equation. When simulating multirate designs where the shared PMA PLL frequency or REFCLK frequency changes, SIM_GTPRESET_SPEEDUP must be set to FALSE. Appendix F, Advanced Clocking illustrates multirate design examples.

SIM_MODE

This simulation-only attribute chooses between two available UNISIM/SIMPRIM simulation models. The LEGACY setting selects the legacy simulation model of the PMA of the GTP transceiver. The FAST setting selects a faster simulation model of the PMA of the GTP transceiver to cut simulation run time.

The legacy model is available for existing users who have been using simulation models with ISE 11.1 and older software for their designs; however, this legacy model will be phased out after ISE® 11.1 software. The FAST setting is highly recommended for new customer designs.

This attribute can be used independently of the SIM_GTPRESET_SPEEDUP attribute.

SIM_PLL_PERDIV2

The GTP_DUAL tile contains an analog PLL to generate the transmit and receive clocks out of a reference clock. Because HDL simulators do not fully model the analog PLL, the GTP_DUAL SmartModel includes an equivalent behavioral model to simulate the PLL output. The SIM_PLL_PERDIV2 attribute is used by the behavioral model to generate the PLL output as accurately as possible. It must be set to one-half the period of the shared PMA PLL. See Examples, page 53 for how to calculate SIM_PLL_PERDIV2 for a given rate.

Apotex Ex1009
Page 50 of 347

SIM_RECEIVER_DETECT_PASS

The GTP_DUAL tile includes a TXDETECTRX feature that allows the transmitter to detect if its serial ports are currently connected to a receiver by measuring rise time on the TXP/TXN differential pin pair (see Receive Detect Support for PCI Express Operation, page 130).

The GTP_DUAL SmartModel includes an attribute for simulating TXDETECTRX called SIM_RECEIVER_DETECT_PASS. This attribute allows TXDETECTRX to be simulated for each GTP transceiver without modelling the measurement of rise time on the TXP/TXN differential pin pair.

SIM_RECEIVER_DETECT_PASS should be set to TRUE by default. When TRUE, the attribute models a connected receiver, and TXDETECTRX operations indicate a receiver is connected. To model a disconnected receiver, SIM_RECEIVER_DETECT_PASS for the transceiver is set to FALSE.

Power-Up and Reset

Link Idle Reset

To simulate correctly, the Link Idle Reset circuit described in Reset, page 84 must be implemented and connected to each GTP_DUAL instance. This circuit is included automatically when the Wizard is used to configure the GTP_DUAL instance.

Toggling GSR

The GSR signal is a global routing of nets in the design that provide a means of setting or resetting applicable components in the device during configuration.

The simulation behavior of this signal is modeled using the glbl module in Verilog and the ROC/ROCBUF components in VHDL.

Providing Clocks in Simulation

In simulation, the clocks inside the PMA are generated using the SIM_PLL_PERDIV2 parameter (in picoseconds). Any other clocks driven into the user clock must have the same level of precision, or TX buffer errors (and RX buffer errors in systems without clock correction) can result. When generating USRCLK, USRCLK2, or reference clock signals in the test bench, the clock periods must be related to SIM_PLL_PERDIV2 and also be a round number (in picoseconds). In some cases, the simulation a clock rate is slightly different from the clock rate used in the actual design.

Simulating in Verilog

The GSR and global 3-state (GTS) signals are defined in the `$XILINX/verilog/src/glbl.v` module. Because the `glbl.v` module connects the global signals to the design, it is necessary to compile this module with the other design files and load it along with the `design.v` and `testfixture.v` files for simulation.

Defining GSR/GTS in a Test Bench

There are two ways to handle GSR and GTS in a test bench:

Optiver Ex1009
Page 51 of 347

1. In most cases, GSR and GTS do not need to be defined in the test bench. The `glbl.v` file declares the GSR and GTS signals and automatically pulses GSR for 100 ns. This handling is sufficient for back-end simulations and functional simulations as well.
2. If GSR or GTS needs to be emulated in the test bench, the following snippet of code must be added to the `testfixture.v` file:

```
assign glbl.GSR = gsr_r;
assign glbl.GTS = gts_r;
initial
begin
gts_r = 1'b0;
gsr_r = 1'b1;
#(16*CLOCKPERIOD);
gsr_r = 1'b0;
end
```

Simulating in VHDL

The `ROCBUF` cell controls the emulated GSR signal in a test bench. This component creates a buffer for the GSR signal and provides an input port on the buffer to drive GSR. This port must be declared in the entity list and driven through the test bench.

The VHDL code for this cell, located in `EX_ROCBUF.vhd`, is listed below:

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
library UNISIM;
use UNISIM.all;
entity EX_ROCBUF is
port (
CLOCK, ENABLE, SRP,RESET : in std_logic;
C_OUT: out std_logic_vector (3 downto 0)
);
end EX_ROCBUF;

architecture A of EX_ROCBUF is
signal GSR : std_logic;
signal COUNT : std_logic_vector (3 downto 0);
component ROCBUF
port (
I : in std_logic;
O : out std_logic
);
end component;

begin
U1 : ROCBUF port map (I => SRP, O => GSR);

//dummy process
COUNTER : process (CLOCK, ENABLE, RESET)
begin
...
end process COUNTER;
end A
```

The VHDL code for this test bench, located in `EX_ROCBUF_tb.vhd`, is listed below:

```
entity EX_ROCBUF_tb is
end EX_ROCBUF_tb;
```


Apotex Ex1009
Page 52 of 347

```
architecture behavior of EX_ROCBUF_tb is
declare component EX_ROCBUF
declare signals
begin
EX_ROCBUF_inst: EX_ROCBUF PORT MAP(
CLOCK => CLOCK,
ENABLE => ENABLE,
SRP => SRP,
RESET => RESET,
COUT => COUT
);
Clk_generation: process
Begin
...
End process
reset <= '1', '0' after CLK_PERIOD * 30;
SRP <= '1', '0' after CLK_PERIOD * 25;
end
```

Further details can be found in the *Synthesis and Simulation Design Guide* [Ref 2].

Examples

Simulation Environment Setup Example (ModelSim SE 6.1d on Linux)

This section provides an example of how to set up a simulation environment for SmartModel support. This is a prerequisite for simulating designs containing GTP_DUAL tile(s).

This example uses ModelSim SE 6.1d, the HDL simulator from Mentor Graphics, with RedHat Enterprise Linux 3.0 as the operating system and version 8.1i of the Xilinx ISE™ development system. The *Synthesis and Simulation Guide* provides guidelines and examples for a different HDL simulator or Xilinx ISE development system[1].

Use **`setenv`** to set these environment variables:

- `XILINX` Location of the installed Xilinx ISE system (for example, `/opt/Xilinx/ise_8_1_i`)
- `MODEL_TECH` Location of the installed ModelSim simulator (for example, `/edatools/mentor/modelsim/6.1d/`)
- `LMC_HOME` `$XILINX/smartmodel/lin/installed_lin`
- `LMC_CONFIG` `$LMC_HOME/data/linux.lmc`
- `LD_LIBRARY_PATH` `$LMC_HOME/lib/linux.lib:$LD_LIBRARY_PATH`

The initialization file (`Modelsim.ini`) contains these settings:

```
libsm = $MODEL_TECH/libsm.sl
libswift = $LMC_HOME/lib/linux.lib/libswift.so
Resolution = 1ps ;(one picosecond simulator resolution)
```

1. If there is a contradiction between this example and the documentation of your simulator, the simulator documentation has precedence. If a newer version of the Xilinx ISE development system is used, check the Xilinx website for additional information.

The location of SmartModel in the ISE directory tree is:

```
$XILINX/virtex5/smartmodel/lin/image
```

The selected options for the COMPXLIB tool are:

```
compxlib -s mti_se -l all -arch all -smartmodel_setup
```

These options use the COMPXLIB tool to compile all libraries for all languages for the ModelSim SE 6.1d HDL simulator. The default output directory is `$XILINX/language/target_simulator`. The compiled libraries are specified to be written to `$XILINX/vhdl/mti_se` and `$XILINX/verilog/mti_se`.

SIM_PLL_PERDIV2 Calculation Example

This section provides examples of how to calculate the correct value for the simulation-only attribute SIM_PLL_PERDIV2.

The period of the PLL can be calculated using Equation 3-2 and Equation 3-3.

$$\text{PLL SPEED} = \left(\frac{\text{REFCLK}}{\text{PLL\_DIV\_REF}}\right) \times \text{DIV} \times (\text{PLL\_DIVSEL\_FB}) \qquad \textit{Equation 3-2}$$

$$\text{SIM\_PLL\_PERDIV2} = \frac{(1/\ \text{PLL SPEED})}{2} \qquad \textit{Equation 3-3}$$

The terms used in Equation 3-2 and Equation 3-3 are defined as follows:

- REFCLK is the speed of the clock tied to the CLKIN input of the GTP_DUAL tile in MHz.
- PLL_DIVSEL_REF is an attribute that defines the dividing factor of the reference clock divider of the shared PMA PLL.
- PLL_DIVSEL_FB is an attribute that defines the dividing factor of the feedback divider (which acts like a multiplication factor) of the shared PMA PLL.
- DIV = 5 when INTDATAWIDTH is High (10-bit mode) or DIV = 4 when INTDATAWIDTH is Low (8-bit mode).

Example for PCI Express Design

To calculate PLL SPEED and SIM_PLL_PERDIV2 for the PCI Express example, these values are assigned:

- REFCLK = 100 MHz
- PLL_DIVSEL_REF = 2
- DIV = 5
- PLL_DIVSEL_FB = 5

Using Equation 3-2, PLL SPEED is 1.25 GHz, meaning that the period is 800 ps. Using Equation 3-3, SIM_PLL_PERDIV2 is 800 divided by 2 equal to 400 decimal or `190` hexadecimal.

Example for Gigabit Ethernet Design

To calculate PLL SPEED and SIM_PLL_PERDIV2 for the Gigabit Ethernet example, the following values are assigned:

- REFCLK = 125 MHz

Optiver Ex1009
Page 54 of 347

- PLL_DIVSEL_REF = 1
- DIV = 5
- PLL_DIVSEL_FB = 2

Using Equation 3-2, PLL SPEED is 1.25 GHz, meaning that the period is 800 ps. Using Equation 3-3, SIM_PLL_PERDIV2 is 800 divided by 2 or 400 decimal (`190` hexadecimal).

Example for XAUI Design

To calculate PLL SPEED and SIM_PLL_PERDIV2 for the XAUI example, the following values are assigned:

- REFCLK = 156.25 MHz
- PLL_DIVSEL_REF = 1
- DIV = 5
- PLL_DIVSEL_FB = 2

Using Equation 3-2, PLL SPEED is 1.5625 GHz, meaning that the period is 640 ps. Using Equation 3-3, SIM_PLL_PERDIV2 is 640 divided by 2 or 320 decimal (`140` hexadecimal).

Optiver Ex1009
Page 55 of 347

Oplover Ex1009
Page 56 of 347

Chapter 4

Implementation

Overview

This chapter provides the information needed to map GTP_DUAL tiles instantiated in a design to device resources, including:

- The location of the GTP_DUAL tiles on the available device and package combinations.
- The pad numbers of external signals associated with each GTP_DUAL tile.
- How GTP_DUAL tiles and clocking resources instantiated in a design are mapped to available locations with a user constraints file (UCF).

It is a common practice to define the location of GTP transceivers early in the design process to ensure correct usage of clock resources and to facilitate signal integrity analysis during board design. The implementation flow facilitates this practice through the use of location constraints in the UCF.

While this chapter describes how to instantiate GTP_DUAL clocking components, the details of the different GTP_DUAL tile clocking options are discussed in Clocking, page 80.

Ports and Attributes

Table 4-1 shows the external ports associated with each GTP_DUAL tile.

Table 4-1: **GTP_DUAL Tile External Ports**

Port	Dir	Domain	Description
MGTTXP0 MGTTXN0 MGTTXP1 MGTTXN1	Out	Embedded TX Clock	Differential transmit data pairs for GTP transceivers 0 and 1
MGTRXP0 MGTRXN0 MGTRXP1 MGTRXN1	In	Embedded RX Clock	Differential receive data pairs for GTP transceivers 0 and 1
MGTREFCLKP MGTREFCLKN	In	N/A	Differential reference clock input pair
MGTAVCCPLL(2)	Analog	Analog	Pad for 1.2V supply for PLL

Optiver Ex1009
Page 57 of 347

Table 4-1: **GTP_DUAL Tile External Ports *(Continued)***

Port	Dir	Domain	Description
MGTAVCC(2)	Analog	Analog	Two pads for 1.0V supply for transceiver mixed signal circuitry
MGTAVTTRX(2)	Analog	Analog	Pad for 1.2V supply for RX circuitry
MGTAVTTTX(2)	Analog	Analog	Two pads for 1.2V supply for TX circuitry

Notes:

1. These port names have the prefix *MGT* to identify them easily in a pad file that is often used to create symbols for board design schematics. In this document, the MGT prefix was removed from those names; however, names with and without the MGT prefix are synonymous with each other.
2. Nominal values. Refer to DS202: *Virtex-5 FPGA Data Sheet* for exact values and marginal conditions.

There are no attributes for this section.

Description

The position of GTP_DUAL tiles is specified by an XY coordinate system that describes the column number and its relative position within that column. In current members of the Virtex-5 LXT and SXT platforms, all GTP_DUAL tiles are located in a single column along one side of the die. As a result the X coordinate for all of the GTP_DUAL tiles is 0. Package Placement Information, page 60 lists the GTP_DUAL tile position information for all available device and package combinations along with the pad numbers for the external signals associated with each tile.

There are two ways to create a UCF for designs that utilize GTP_DUAL tiles. The preferred method is by using the RocketIO GTP Transceiver Wizard (see Chapter 2, RocketIO GTP Transceiver Wizard). The Wizard automatically generates UCF templates that configure the transceivers and contain placeholders for GTP_DUAL placement information. The UCFs generated by the Wizard can then be edited to customize operating parameters and placement information for the application.

The second approach is to create the UCF by hand. When using this approach, the designer must enter both configuration attributes that control transceiver operation as well as tile location parameters. Care must be taken to ensure that all of the parameters needed to configure the GTP_DUAL tile are correctly entered.

Apotex Ex1009
Page 58 of 347

Example of a UCF for GTP_DUAL Placement

This section shows key elements of a UCF that instantiates seven GTP_DUAL tiles. The file implements the example configuration shown in Figure 5-5, page 83. The device and package combination chosen in this example is an XC5VLX110T-FF1136.

```
;
; Instantiate the GTP_DUAL tiles in locations X0Y7 to X0Y1
;
INST design_root/gtp_dual[1]/gtp_dual   LOC=GTP_DUAL_X0Y1;
INST design_root/gtp_dual[2]/gtp_dual   LOC=GTP_DUAL_X0Y2;
INST design_root/gtp_dual[3]/gtp_dual   LOC=GTP_DUAL_X0Y3;
INST design_root/gtp_dual[4]/gtp_dual   LOC=GTP_DUAL_X0Y4;
INST design_root/gtp_dual[5]/gtp_dual   LOC=GTP_DUAL_X0Y5;
INST design_root/gtp_dual[6]/gtp_dual   LOC=GTP_DUAL_X0Y6;
INST design_root/gtp_dual[7]/gtp_dual   LOC=GTP_DUAL_X0Y7;
;
; Connect the REFCLK_PAD_(N/P) differential pair to the middle
; GTP_DUAL tile (GTP_DUAL_X0Y4)
;
NET refclk_pad_n LOC=P4;
NET refclk_pad_p LOC=P3;
```

The instantiation of the GTP_DUAL tiles and the IBUFDS primitive is typically done in HDL code within the design hierarchy. That code also connects the output of the IBUFDS primitive to the CLKIN inputs of the GTP_DUAL tiles, as illustrated by the following Verilog code fragment:

```
//
// Instantiate the GTP_DUAL tiles
//
genvar tile_num;

generate for (tile_num = 1; tile_num <= 7; ++tile_num)

  begin: gtp_dual

    GTP_DUAL gtp_dual
    (
      .CLKIN(refclk),

      … The remaining GTP_DUAL ports are not shown
    )

  end

endgenerate


//
// Instantiate the IBUFDS for the reference clock
//
IBUFDS ref_clk_buffer
(
  .IP(refclk_pad_n),
  .IN(refclk_pad_p),
  .O(refclk)
)
```

Package Placement Information

The diagrams in this section illustrate GTP_DUAL placement for the following packages:

- LXT packages:
  - XC5VLX20T-FF323
  - XC5VLX30T-FF323
  - XC5VLX30T-FF665
  - XC5VLX50T-FF665
  - XC5VLX50T-FF1136
  - XC5VLX85T-FF1136
  - XC5VLX110T-FF1136
  - XC5VLX110T-FF1738
  - XC5VLX155T-FF1136
  - XC5VLX155T-FF1738
  - XC5VLX220T-FF1738
  - XC5VLX330T-FF1738
- SXT packages:
  - XC5VSX35T-FF665
  - XC5VSX50T-FF665
  - XC5VSX50T-FF1136
  - XC5VSX95T-FF1136
  - XC5VSX240T-FF1738

Figure 4-1 illustrates the nomenclature used in each of these diagrams. The GTP_DUAL placement name is the name used in the UCF to map GTP_DUAL tiles instantiated in the design to specific tiles on the device. The board-level pin names and numbers are the names placed in the PKG file generated by the ISE design flow. This file is typically used by board-level schematic capture and layout tools to create component symbols and layout footprints.

Cypher Ex1009
Page 60 of 347

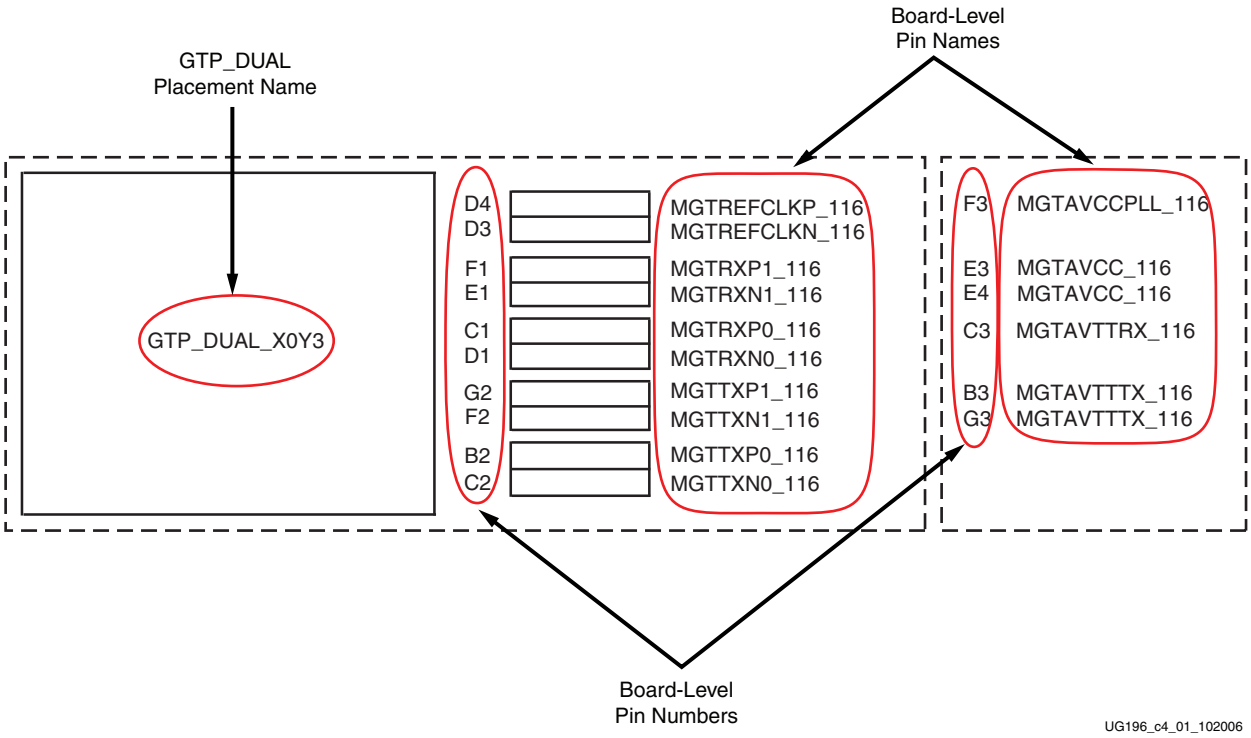


Figure 4-1: **Placement Diagram Nomenclature**

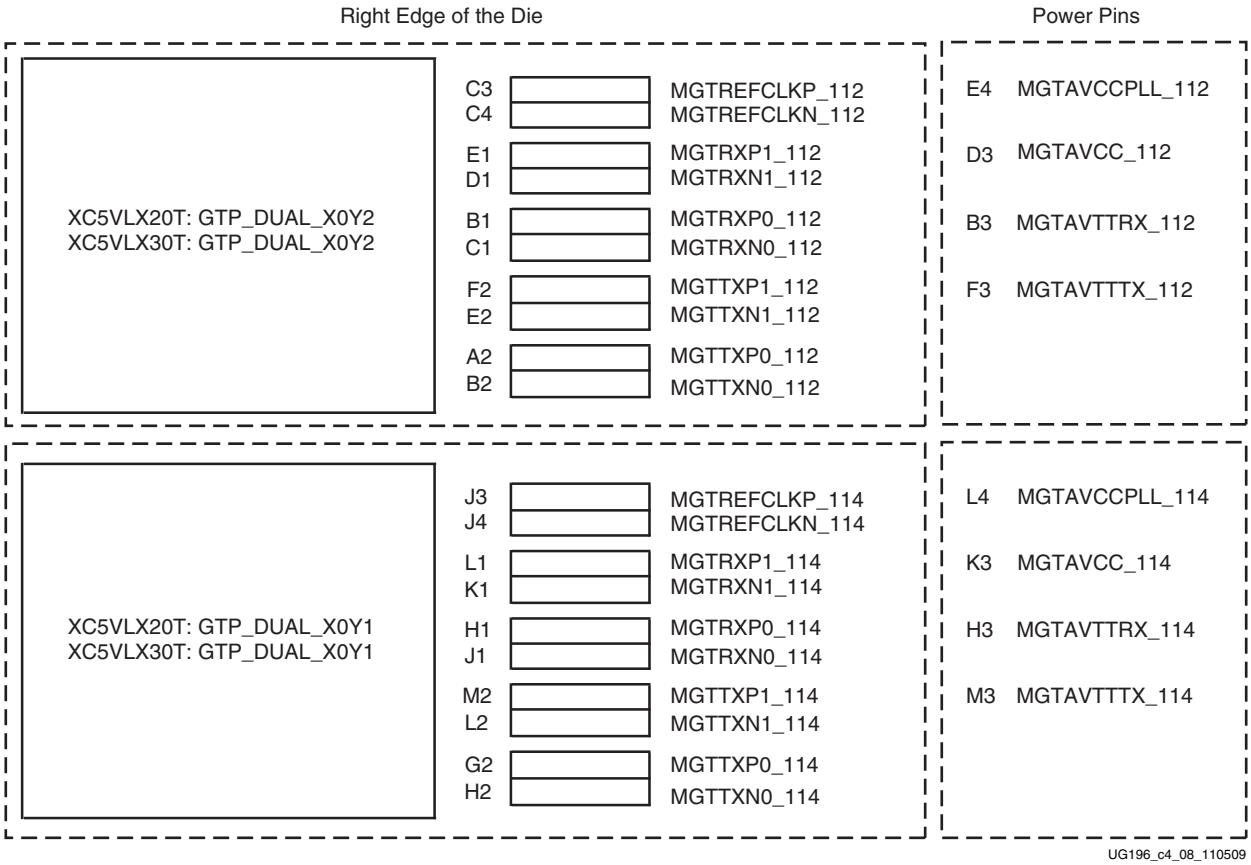


Figure 4-2: **XC5VLX20T-FF323 and XC5VLX30T-FF323 GTP Placement**

Oplover Ex1009
Page 62 of 347

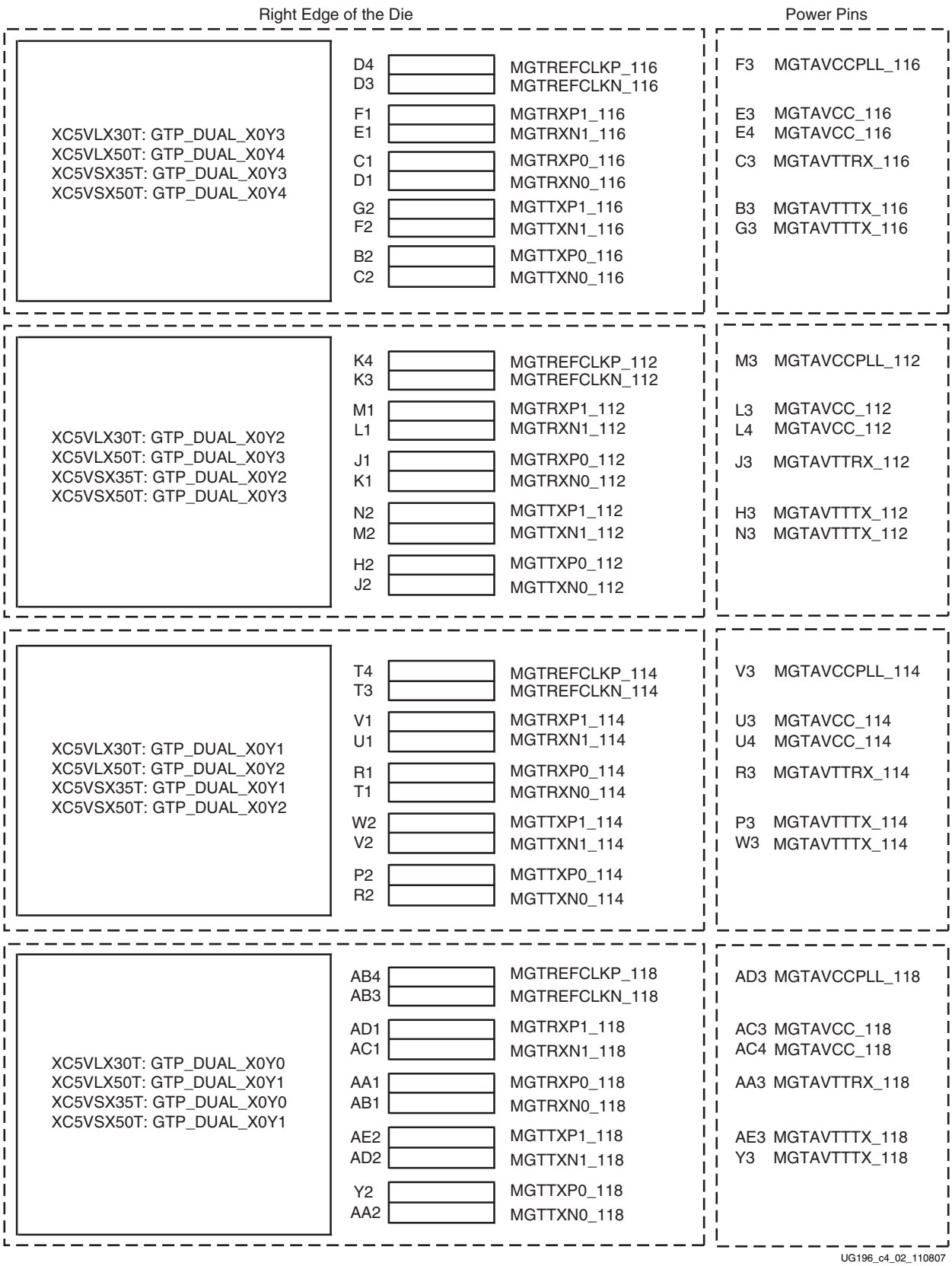


Figure 4-3: **XC5VLX30T-FF665, XC5VLX50T-FF665, XC5VSX35T-FF665, and XC5VSX50T-FF665 GTP Placement**

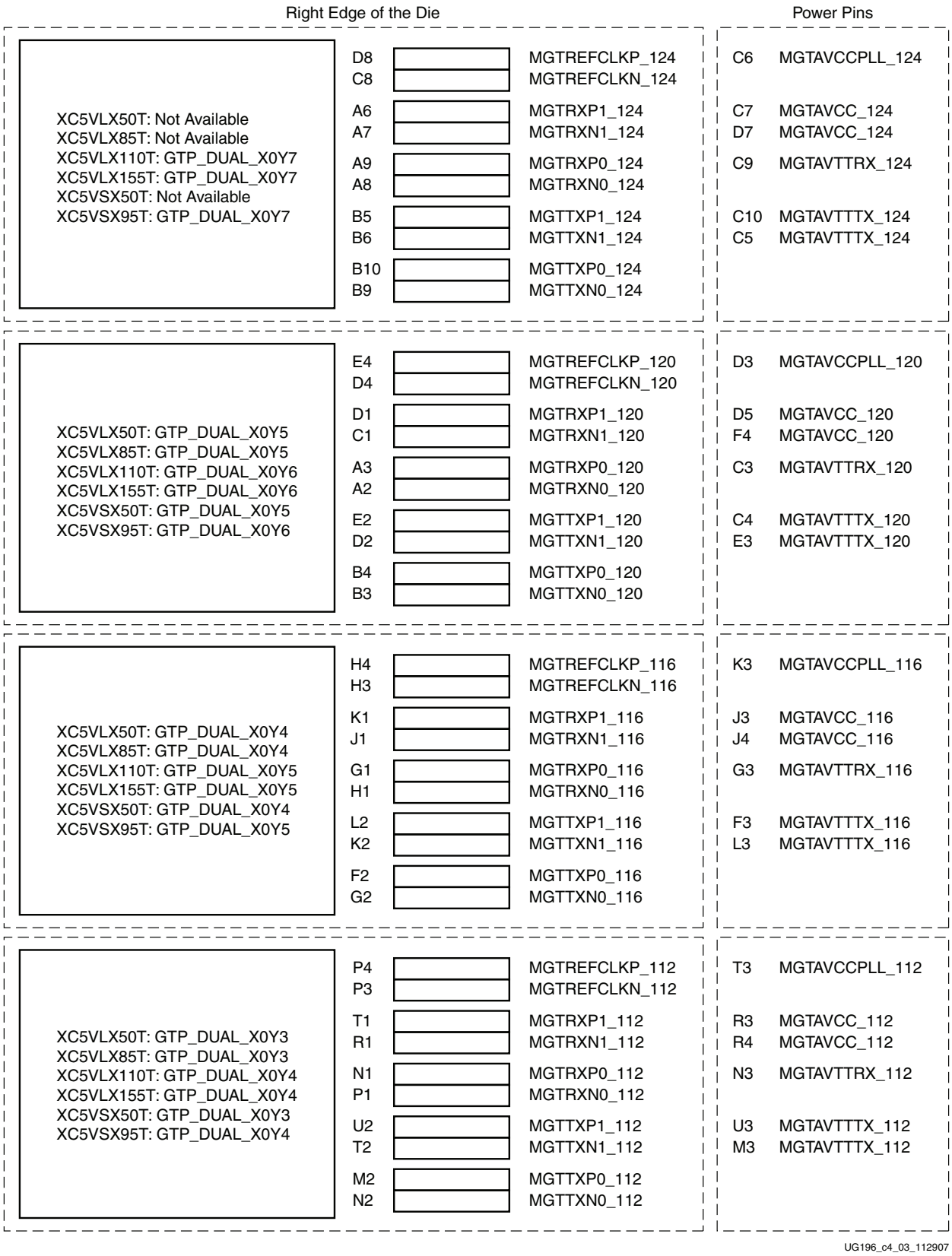


Figure 4-4: **XC5VLX50T-FF1136, XC5VLX85T-FF1136, XC5VLX110T-FF1136, XC5VLX155T-FF1136, XC5VSX50T-FF1136, and XC5VSX95T-FF1136 GTP Placement (1 of 2)**

Power Ex1009
Page 64 of 347

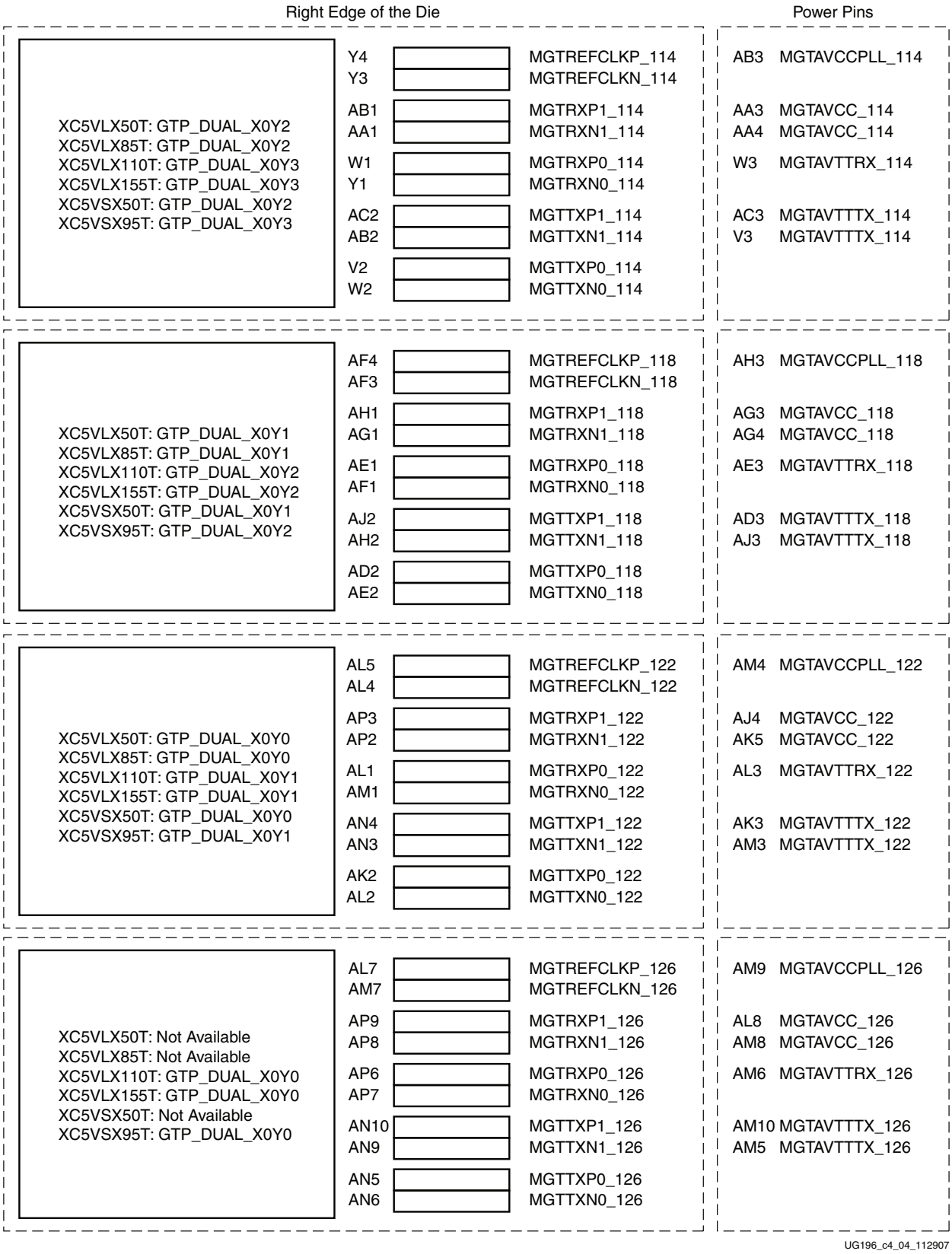


Figure 4-5: **XC5VLX50T-FF1136, XC5VLX85T-FF1136, XC5VLX110T-FF1136, XC5VLX155T-FF1136, XC5VSX50T-FF1136, and XC5VSX95T-FF1136 GTP Placement (2 of 2)**

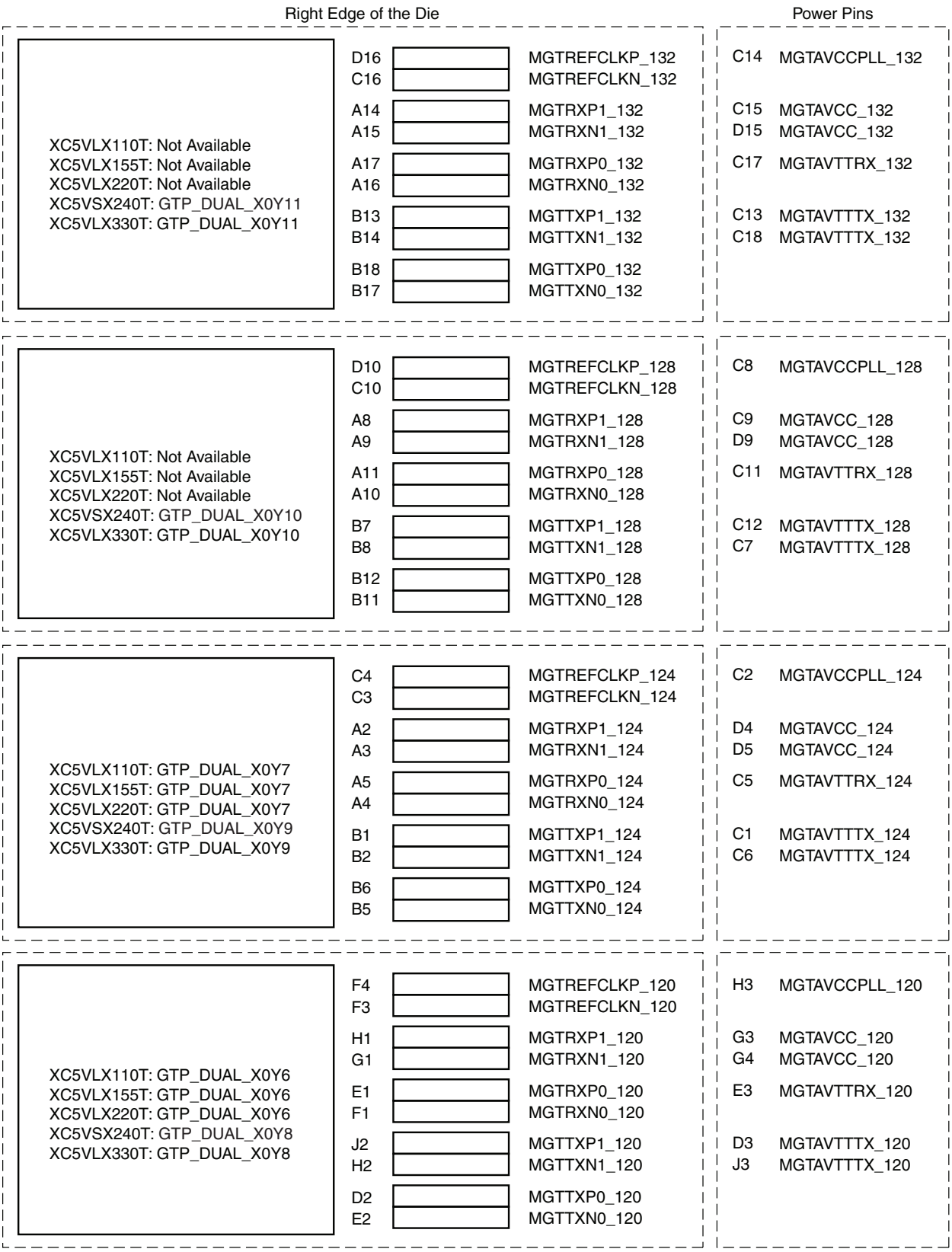


Figure 4-6: **XC5VLX110T-FF1738, XC5VLX155T-FF1738, XC5VLX220T-FF1738, XC5VSX240T-FF1738, and XC5VLX330T-FF1738 GTP Placement (1 of 3)**

Apotex Ex1009
Page 66 of 347

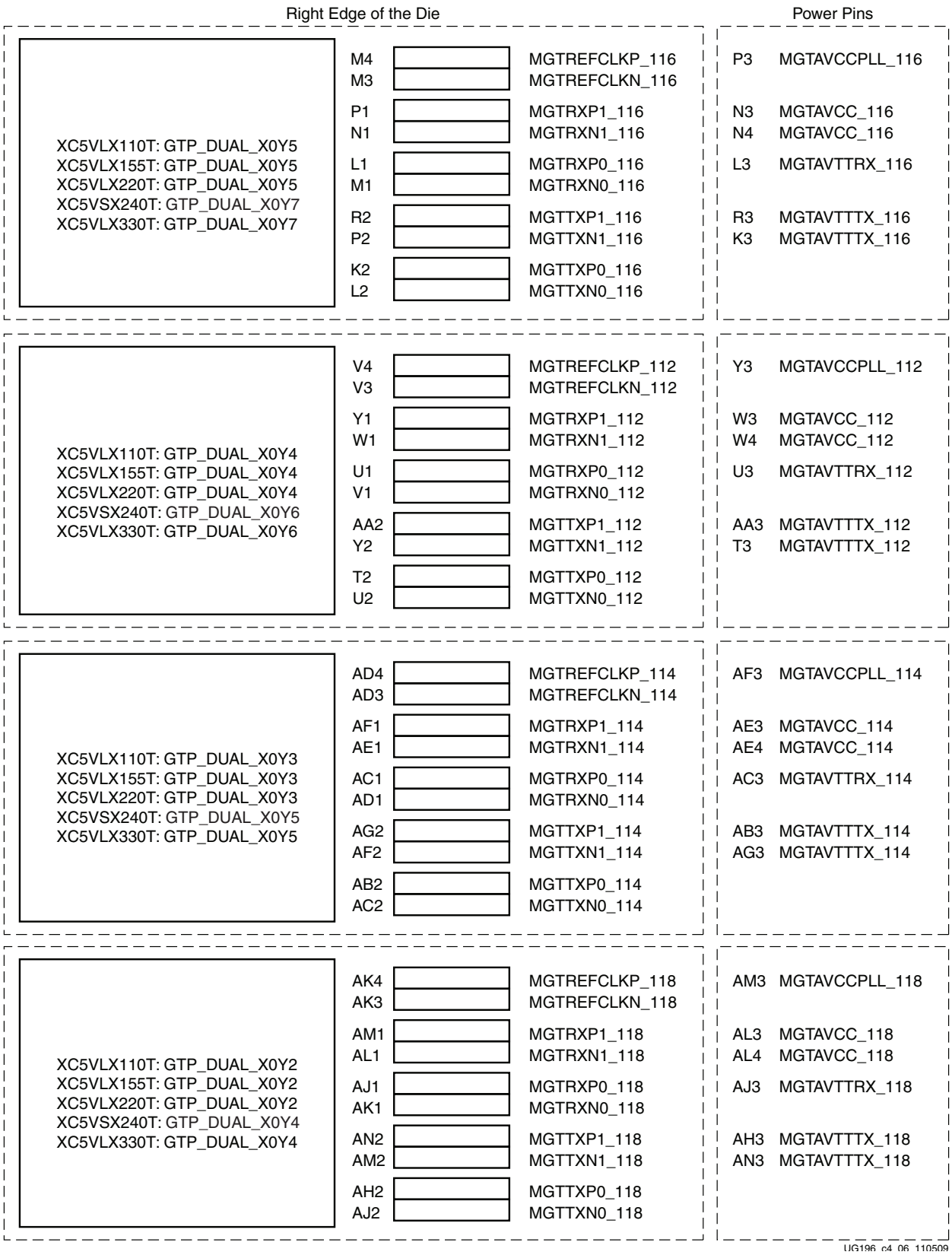


Figure 4-7: **XC5VLX110T-FF1738, XC5VLX155T-FF1738, XC5VLX220T-FF1738, XC5VSX240T-FF1738, and XC5VLX330T-FF1738 GTP Placement (2 of 3)**

Optiver Ex1009
Page 67 of 347

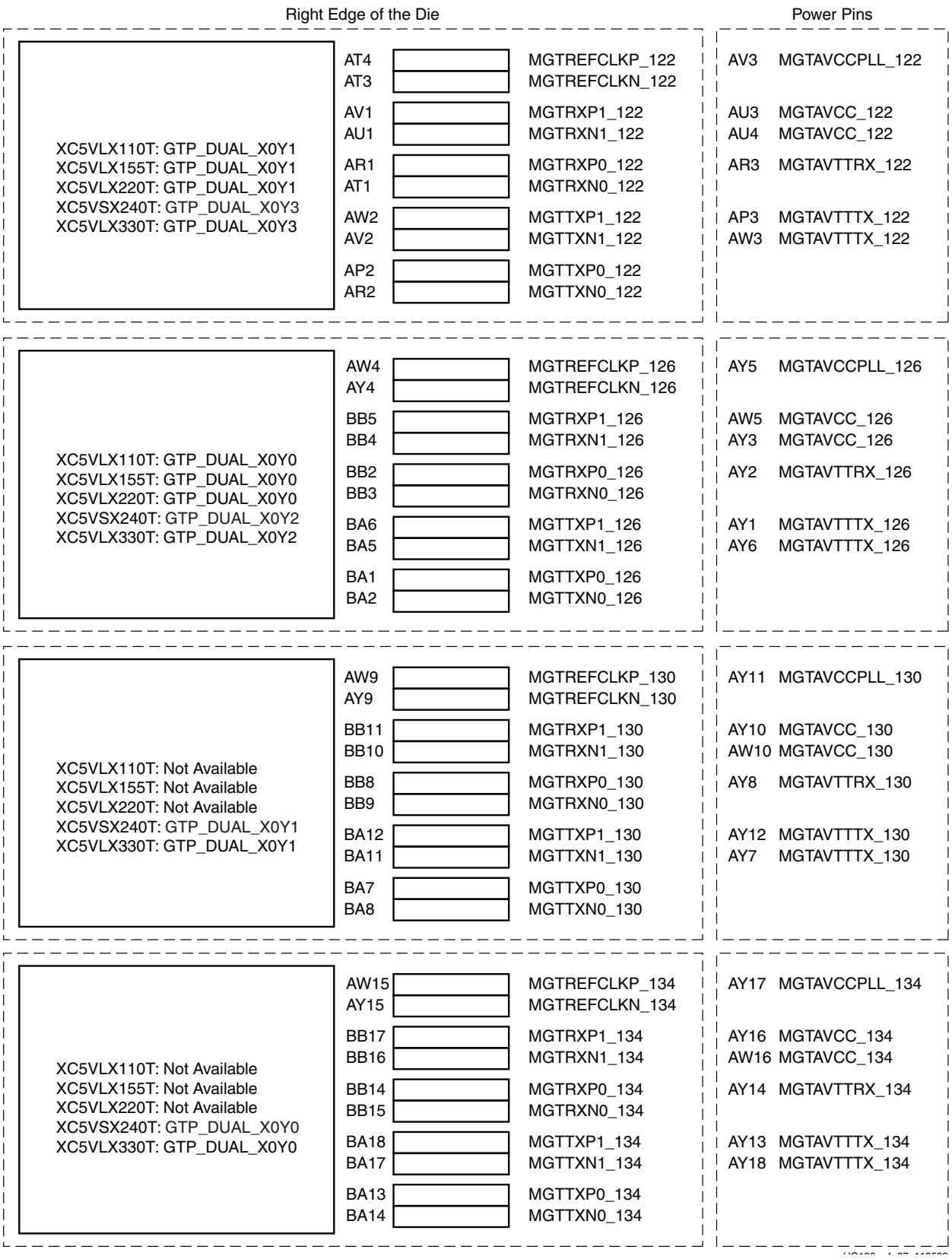


Figure 4-8: **XC5VLX110T-FF1738, XC5VLX155T-FF1738, XC5VLX220T-FF1738, XC5VSX240T-FF1738, and XC5VLX330T-FF1738 GTP Placement (3 of 3)**

Power Ex1009
Page 68 of 347

Optiver Ex1009
Page 69 of 347

Cpuver Ex1009
Page 70 of 347

Chapter 5

Tile Features

Tile Features Overview

To minimize power consumption and area, many important GTP functions are shared between two transceivers. These functions include the generation of a high-speed serial clock, resets, power control, and dynamic reconfiguration.

Correct clocking and reset behavior is critical for any GTP transceiver design. This chapter describes the following steps that must be performed when configuring the GTP_DUAL tile:

- Set the shared PMA PLL rate
- Set the reference clock source
- Implement the Link Idle Reset circuit

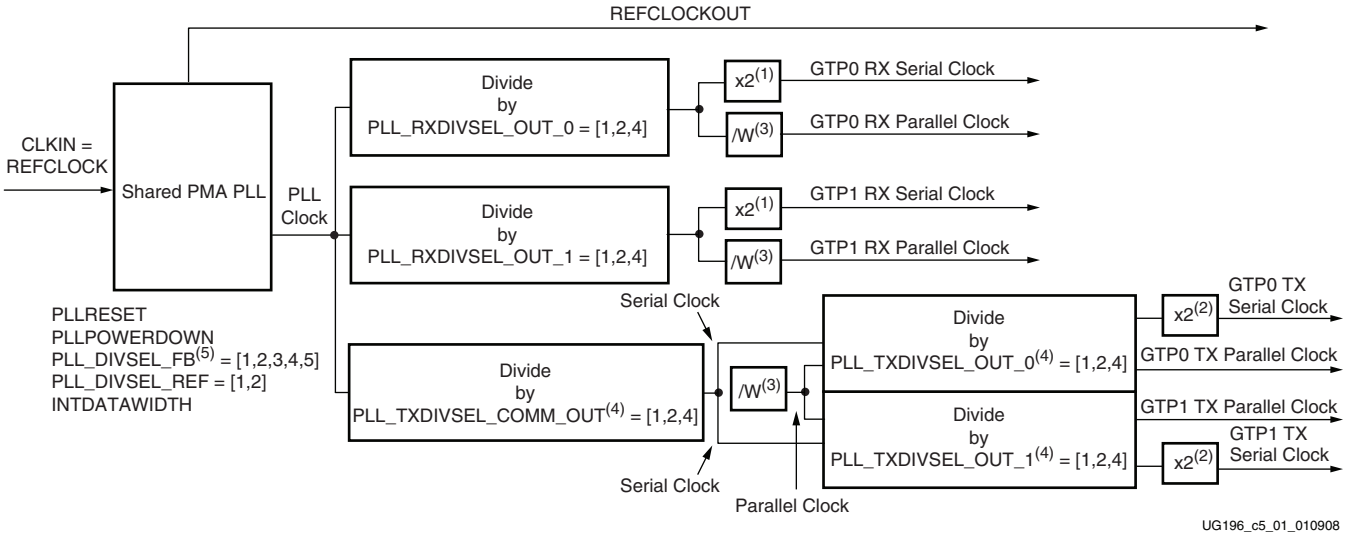
These steps are performed automatically when the Wizard is used to configure the GTP_DUAL tile.

Optiver Ex1009
Page 71 of 347

Shared PMA PLL

Overview

This section describes the shared PMA PLL of the GTP_DUAL tile, which is illustrated in Figure 5-1. Each GTP_DUAL tile includes one shared PMA PLL used to generate a high-speed serial clock from a high-quality reference clock (CLKIN). The high-speed clock from this block drives the TX and RX PMA blocks for both GTP transceivers in the tile.



Notes:

1. The Serial In to Parallel Out (SIPO) block in each receiver uses both edges of the high-speed clock. As a result, the effective RX serial clock rate is 2 x PLL Clock/PLL_RXDIVSEL_OUT_n.
2. The Parallel In to Serial Out (PISO) block in each transmitter uses both edges of the high-speed clock. As a result, the effective TX serial clock rate is 2 x PLL Clock/[PLL_TXDIVSEL_OUT_n, PLL_TXDIVSEL_COMM_OUT].
3. The parallel clock rate is divided to match the internal datapath width. When INTDATAWIDTH is Low (8-bit internal width), W = 4. When INTDATAWIDTH is High (10-bit internal width), W = 5.
4. Refer to Chapter 9, Loopback, about the correct setting of these attributes for specific loopback modes.
5. When INTDATAWIDTH is Low, PLL_DIVSEL_FB can only be set to 1, 2, or 4. For PLL_DIVSEL_FB = 1, set PCS_COM_CFG to `28'h1680A07`, otherwise set PCS_COM_CFG to `28'h1680A0E` (default).

Figure 5-1: **Shared PMA PLL Detail**

The shared PMA PLL generates the high-speed clock (PLL clock) used by both transceivers in the GTP_DUAL tile. After the shared PMA PLL rate is set (PLL clock), the TX and RX output dividers (dividers ending with _OUT) are set to determine the TX and RX line rates for each transceiver.

Ports and Attributes

Table 5-1 defines the shared PMA PLL ports.

Cpuver Ex1009
Page 72 of 347

Table 5-1: **Shared PMA PLL Ports**

Port	Dir	Domain	Description
CLKIN	In	Async	Reference clock input to the shared PMA PLL. See Clocking, page 80 for more information about the different ways this port can be driven.
INTDATAWIDTH	In	Async	Sets the internal datapath width for the GTP_DUAL tile. If Low, the internal datapath width is set to 8 bits. If High, the internal datapath width is set to 10 bits.
PLLLKDET	Out	Async	This port indicates that the VCO rate is within acceptable tolerances of the desired rate when High. Neither GTP transceiver in the tile operates reliably until this condition is met.
PLLLKDETEN	In	Async	This port enables the PLL lock detector and should always be tied High.
REFCLKOUT	Out	Async	The REFCLKOUT port from each GTP_DUAL tile provides direct access to the reference clock provided to the shared PMA PLL (CLKIN). It can be routed for use in the FPGA logic.

Table 5-2 defines the shared PMA PLL attributes.

Table 5-2: **Shared PMA PLL Attributes**

Attribute	Description
PCS_COM_CFG[27:0][(1)]	For PLL_DIVSEL_FB = 1, PCS_COM_CFG is set to `28'h1680A07`, otherwise it is set to `28'h1680A0E` (default).
PLL_DIVSEL_FB	Controls the feedback divider. Valid settings for PLL_DIVSEL_FB are 1, 2, 3, 4, and 5. PLL_DIVSEL_FB is multiplied by 4 or 5, depending on the width of the internal datapath as set by INTDATAWIDTH. If INTDATAWIDTH is Low, the feedback divider N is set to PLL_DIVSEL_FB x 4. If INTDATAWIDTH is High, the feedback divider N is set to PLL_DIVSEL_FB x 5.
PLL_DIVSEL_REF	Controls the reference clock divider. Valid settings for PLL_DIVSEL_REF are 1 and 2.
PLL_RXDIVSEL_OUT_0 PLL_RXDIVSEL_OUT_1	Divides the PLL clock to produce a high-speed RX clock. Because both edges of the clock are used, the divided clock must run at one-half the desired RX line rate. Permitted divider settings are 1, 2, and 4. See Serial In to Parallel Out, page 155 for details.
PLL_TXDIVSEL_COMM_OUT	Divides the PLL clock to produce a high-speed TX clock. Because both edges of the clock are used, the divided clock must run at one-half the desired TX line rate. Permitted divider settings are 1, 2, and 4. This divider provides a clock to both GTP transceivers and should be used when the same divider value is needed for both. When PLL_TXDIVSEL_COMM_OUT is used, both PLL_TXDIVSEL_OUT attributes must be set to 1. See Parallel In to Serial Out, page 124 for details.
PLL_TXDIVSEL_OUT_0 PLL_TXDIVSEL_OUT_1	Divides the PLL clock to produce a high-speed TX clock. Because both edges of the clock are used, the divided clock must run at one-half the desired TX line rate. Permitted divider settings are 1, 2, and 4. Each GTP transceiver has its own PLL_TXDIVSEL_OUT. If the transceivers require different dividers, these attributes must be used instead of PLL_TXDIVSEL_COMM_OUT, and PLL_TXDIVSEL_COMM_OUT must be set to 1. See Parallel In to Serial Out, page 124 for details. When bypassing the TX buffer, PLL_TXDIVSEL_OUT_0 and PLL_TXDIVSEL_OUT_1 must be set to divide by 1. See TX Buffering, Phase Alignment, and TX Skew Reduction, page 115 for details.

Notes:

1. In the ISE tool, version 9.2i and above, this attribute is included in the GTP_DUAL instance. Older ISE tool versions require setting this attribute with the user-constraints file (UCF) when a non-default value is needed.

Description

The first step to using the GTP_DUAL tile is to set the output of the shared PMA PLL (PLL clock) to a rate that can be used by each GTP transceiver to generate an appropriate serial line rate and the corresponding parallel clock rate. Both GTP TX and RX blocks are equipped with an independent divider that can divide the PLL clock by a factor of 1, 2, or 4. This divider allows the TX and RX blocks of each GTP transceiver to run at different rates, related by an integer multiple.

The PLL clock rate must be set to one-half the required line rate before the independent dividers. For example, if an RX line rate of 2.5 Gb/s is desired in GTP0, and the independent divider in the GTP RX block is set to 1, the PLL clock should be set to 1.25 GHz.

The shared PMA PLL has a nominal operation range. The *Virtex-5 FPGA Data Sheet* specifies the operating range of the shared PMA PLL including the marginal conditions. The PLL clock must be set within this operating range. Equation 5-1 shows how to set the PLL clock based on CLKIN (the reference clock), PLL_DIVSEL_FB, PLL_DIVSEL_REF, and INTDATAWIDTH. PLL_DIVSEL_FB and PLL_DIVSEL_REF control dividers inside the PLL. INTDATAWIDTH controls the internal parallel data width of the entire GTP_DUAL tile.

Equation 5-1 shows how to set the rate of the PLL clock.

$$f_{PLL\ Clock} = f_{CLKIN} \times \frac{PLL\_DIVSEL\_FB \times DIV}{PLL\_DIVSEL\_REF} \qquad \text{Equation 5-1}$$

The following conditions apply to Equation 5-1:

- PLL_DIVSEL_REF = [1, 2].
- PLL_DIVSEL_FB = [1, 2, 3, 4, 5], where 3 and 5 cannot be used when INTDATAWIDTH is Low.

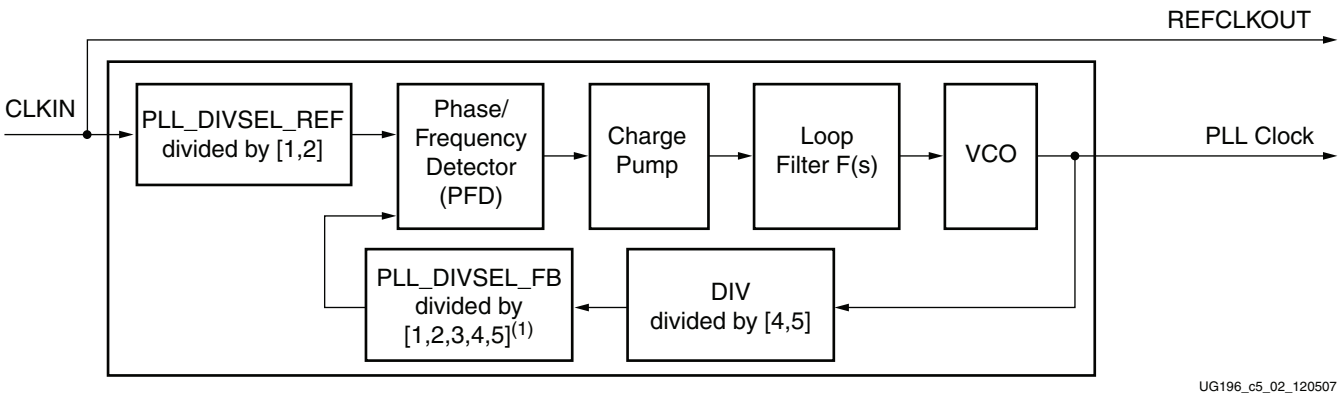
  When PLL_DIVSEL_FB = 1, PCS_COM_CFG is set to `28'h1680A07`, otherwise PCS_COM_CFG is set to `28'h1680A0E` (default).
- When OVERSAMPLE_MODE is TRUE, DIV = 5 regardless of the setting of INTDATAWIDTH.

  When OVERSAMPLE_MODE is FALSE, if INTDATAWIDTH is Low, then DIV = 4 or if INTDATAWIDTH is High, then DIV = 5.
- The PLL clock frequency range is as specified in the *Virtex-5 FPGA Data Sheet*.
- The smallest possible divider values must be selected.

The programmable dividers allow support for various standards.

Figure 5-2 shows a conceptual view of the shared PMA PLL from which the PLL clock is generated.

Opover Ex1009
Page 74 of 347



Notes:

1. When INTDATAWIDTH is Low, PLL_DIVSEL_FB can only be set to 1, 2, or 4.

Figure 5-2: **Shared PMA PLL Conceptual View**

Table 5-3 shows example settings for several standard protocols. The PLL clock frequency values shown in Table 5-3 are in the operation range of the shared PMA PLL specified in the *Virtex-5 FPGA Data Sheet*.

Table 5-3: **Communication Standards**

Standard	Line Rate [Gb/s]	TX/RX USRCLK Frequency [MHz]	TX/RX USRCLK2 Frequency [MHz]		Reference Clock Frequency REFCLK [MHz]	PLL Clock Frequency [GHz]	Reference Clock Divider Setting PLL_DIVSEL_REF	Feedback Loop Divider Setting PLL_DIVSEL_FB	Divider Settings PLL_RXDIVSEL_OUT_(0/1) PLL_TXDIVSEL_OUT_(0/1) PLL_TXDIVSEL_COMM_OUT(3)
			1-byte Logic Interface(1)	2-byte Logic Interface(2)					
INTDATAWIDTH is High (10-bit Internal Datapath) → (DIV = 5)									
FC2	2.125	212.5	212.5	106.25	212.5	1.0625	1	1	1
FC1	1.0625	106.25	106.25	53.125	106.25	1.0625	1	2	2
XAUI	3.125	312.5	312.5	156.25	156.25	1.5625	1	2	1
10GBASE-CX4	3.125	312.5	312.5	156.25	156.25	1.5625	1	2	1
GigE	1.25	125	125	62.5	125	1.25	1	2	2
Aurora	3.75(4)(5)	375	375	187.5	187.5	1.875	1	2	1
	2.5	250	250	125	125	1.25	1	2	1
	1.25	125	125	62.5	125	1.25	1	2	2
Serial RapidIO	3.125	312.5	312.5	156.25	156.25	1.5625	1	2	1
	2.5	250	250	125	125	1.25	1	2	1
	1.25	125	125	62.5	125	1.25	1	2	2
SATA Generation 2	3	300	300	150	150	1.5	1	2	1
SATA Generation 1(6)	1.5	150	150	75	150	1.5	1	2	2
PCI Express	2.5	250	250	125	100	1.25	2	5	1
Infiniband	2.5	250	250	125	125	1.25	1	2	1
HD-SDI(7)	1.485	148.5	148.5	74.25	148.5	1.485	1	2	2
3G-SDI	2.970	148.5	297.0	148.5	148.5	1.485	1	2	1

Table 5-3: **Communication Standards *(Continued)***

Standard	Line Rate [Gb/s]	TX/RX USRCLK Frequency [MHz]	TX/RX USRCLK2 Frequency [MHz]		Reference Clock Frequency REFCLK [MHz]	PLL Clock Frequency [GHz]	Reference Clock Divider Setting PLL_DIVSEL_REF	Feedback Loop Divider Setting PLL_DIVSEL_FB	Divider Settings PLL_RXDIVSEL_OUT_(0/1) PLL_TXDIVSEL_OUT_(0/1) PLL_TXDIVSEL_COMM_OUT(3)
			1-byte Logic Interface(1)	2-byte Logic Interface(2)					
CPRI(8)	2.4576	245.76	245.76	122.88	122.88	1.2288	1	2	1
	1.2288	122.88	122.88	61.44	122.88	1.2288	1	2	2
	0.6144	61.44	61.44	30.72	122.88	1.2288	1	2	4
OBSAI(8)	1.536	153.6	153.6	76.8	153.6	1.536	1	2	2
	0.768	76.8	76.8	38.4	153.6	1.536	1	2	4
SFI-5	3.125	312.5	312.5	156.25	156.25	1.5625	1	2	1
TFI-5	3.1104	311.04	311.04	155.52	155.52	1.5552	1	2	1
INTDATAWIDTH is Low (8-bit Internal Datapath) → (DIV = 4)									
OC12	0.62208	77.76	77.76	38.88	155.52	1.24416	1	2	4
OC48	2.488	311.04	311.04	155.52	155.52	1.24416	1	2	1
SFI-5(4)	2.488	311.04	311.04	155.52	155.52	1.24416	1	2	1
SPI-5(4)	2.488	311.04	311.04	155.52	155.52	1.24416	1	2	1
TFI-5(4)	2.488	311.04	311.04	155.52	155.52	1.24416	1	2	1

Notes:

1. 1-byte (8-/10-bit) user interface. Drive RXDATAWIDTH and TXDATAWIDTH Low and set RX/TXUSRCLK2 rate equal to the RX/TXUSRCLK rate.
2. 2-byte (16-/20-bit) user interface. Drive RXDATAWIDTH and TXDATAWIDTH High and set RX/TXUSRCLK2 rate equal to 0.5x the RX/TXUSRCLK rate.
3. See Parallel In to Serial Out, page 124 and Serial In to Parallel Out, page 155 for more details about the divider setting.
4. Maximum data rate.
5. Data rates of 3.5 Gb/s and higher require a 2-byte internal logic interface.
6. When interfacing the GTP receiver to a spread-spectrum signal, the PLL_RXDIVSEL_OUT attribute must be set to "1" to limit the line rates that can be received while using spread spectrum clocking to those greater than 2 Gb/s.
7. Other frequency is 0.1% lower.
8. Synchronous system.

See Clocking, page 80 for details on supplying CLKIN to the shared PMA PLL.

Examples

Configuring the Shared PMA PLL for XAUI Operation

The three methods to configure the shared PMA PLL for XAUI are described below:

1. Use the RocketIO GTP Transceiver Wizard.

   The wizard includes a protocol file for XAUI that allows it to automatically configure the GTP_DUAL primitive for use in a XAUI design.

2. Use the settings from Table 5-3.

   Table 5-3 includes the settings for common configurations of popular protocols. XAUI settings are included in Table 5-3, along with other protocols that use 8B/10B encoding.

Cover Ex1009
Page 76 of 347

3. Use Equation 5-1 as described in the following steps:
   a. Determine the required line rates.

      For XAUI, both TX and RX use a line rate of 3.125 Gb/s.

   b. Determine the internal datapath width.

      Because XAUI is an 8B/10B-encoded standard, an internal datapath width of 10 bits is required. See Configurable 8B/10B Encoder, page 111 and Configurable 8B/10B Decoder, page 172 for more information about encoding and internal datapath width requirements.

   c. Determine the desired reference clock rate.

      This example uses a reference clock running at 156.25 MHz, a common rate for XAUI.

   d. Calculate the required PLL clock rate.

      Because the Serial In to Parallel Out (SIPO) block uses both edges of the clock to deserialize data, it must be fed a clock running at 3.125/2 = 1.5625 GHz. Because this RX rate of 1.5625 GHz is within the PLL operation range, the external divider (PLL_RXDIVSEL_OUT) must be one. The PLL clock rate is thus 1.5625 x 1 = 1.5625 GHz.

   e. Calculate the required DIV value.

      Because the internal datapath width must be 10 bits (INTDATAWIDTH is High), DIV = 5.

   f. Calculate the required PLL divider ratio.

      Using the values f_{CLKIN}, DIV, and f_{PLL_CLOCK} determined above, rearrange Equation 5-1 to calculate the divider ratio as shown in Equation 5-2. The result is a ratio of 2.

$$\frac{PLL\_DIVSEL\_FB}{PLL\_DIVSEL\_REF} = \frac{f_{PLL\_Clock}}{f_{CLKIN} \times DIV} = \frac{1.565\ GHz}{156.25\ MHz \times 5} = 2 \qquad \textit{Equation 5-2}$$

   g. Select the PLL divider values.

      Select the smallest divider values that result in the required PLL divider ratio. In this case, using PLL_DIVSEL_FB = 2 and PLL_DIVSEL_REF = 1 results in a ratio of 2.

Configuring the Shared PMA PLL for OC-48 Operation

This example shows how to set the shared PMA PLL divider settings for OC-48 using Equation 5-1. The RocketIO GTP Transceiver Wizard and Table 5-3 are simpler alternatives. This example is provided only to illustrate the process with Equation 5-1.

Use Equation 5-1 as described in the following steps:

1. Determine the required line rates.

   For OC-48, both TX and RX use a line rate of 2.488 Gb/s.

2. Determine the internal datapath width.

   Because OC-48 uses no encoding and a datapath that is a multiple of eight bits, an internal datapath width of eight bits is required.

3. Determine the desired reference clock rate.

   This example uses a reference clock running at 155.5 MHz.

4. Calculate the required PLL clock rate.

   Because the SIPO block uses both edges of the clock to deserialize data, it must be fed a clock running at 2.488/2 = 1.244 GHz. Because this RX rate of 1.244 GHz is within the operating range of the PLL, the external divider (PLL_RXDIVSEL_OUT) must be one. The PLL clock rate is thus 1.244 x 1 = 1.244 GHz.

5. Calculate the required DIV value.

   Because the internal datapath width must be eight bits (INTDATAWIDTH is Low), DIV = 4.

6. Calculate the required PLL divider ratio.

   Using the values f_{CLKIN}, DIV, and f_{PLL_CLOCK} determined above, rearrange Equation 5-1 to calculate the divider ratio as shown in Equation 5-3. The result is a ratio of 2.

$$\frac{PLL\_DIVSEL\_FB}{PLL\_DIVSEL\_REF} = \frac{f_{PLL\_Clock}}{f_{CLKIN} \times DIV} = \frac{1.244\ GHz}{155.5\ MHz \times 4} = 2 \qquad \textit{Equation 5-3}$$

7. Select the PLL divider values.

   Select the smallest divider values that result in the required PLL divider ratio. In this case, using PLL_DIVSEL_FB = 2 and PLL_DIVSEL_REF = 1 results in a ratio of 2.

Configuring the Shared PMA PLL for Gigabit Ethernet Operation

This example shows how to set the shared PMA PLL divider settings for Gigabit Ethernet using Equation 5-1. The RocketIO GTP Transceiver Wizard and Table 5-3 are simpler alternatives. This example is provided only to illustrate the process with Equation 5-1.

Use Equation 5-1 as described in the following steps:

1. Determine the required line rates.

   For Gigabit Ethernet, both TX and RX use a line rate of 1.25 Gb/s.

2. Determine the internal datapath width.

   Because Gigabit Ethernet uses 8B/10B encoding, an internal datapath width of 10 bits is required.

3. Determine the desired reference clock rate.

   This example uses a reference clock running at 125 MHz.

4. Calculate the required PLL clock rate.

   Because the SIPO block uses both edges of the clock to deserialize data, it must be fed a clock running at 1.25/2 = 0.625 GHz. Because this RX rate of 0.625 GHz is below the operating range of the PLL, the external divider (PLL_RXDIVSEL_OUT) must be 2 to allow the PLL to run twice as fast (1.25 GHz). The PLL clock rate is thus 0.625 x 2 = 1.25 GHz.

5. Calculate the required DIV value.

   Because the internal datapath width must be 10 bits (INTDATAWIDTH is High), DIV = 5.

Apotex Ex1009
Page 78 of 347

6. Calculate the required PLL divider ratio.

   Using the values f_{CLKIN}, DIV, and f_{PLL_CLOCK} determined above, rearrange Equation 5-1 to calculate the divider ratio as shown in Equation 5-4. The result is a ratio of 2.

$$\frac{PLL\_DIVSEL\_FB}{PLL\_DIVSEL\_REF} = \frac{f_{PLL\_Clock}}{f_{CLKIN} \times DIV} = \frac{1.25\ GHz}{125\ MHz \times 5} = 2 \qquad \textit{Equation 5-4}$$

7. Select the PLL divider values.

   Select the smallest divider values that result in the required PLL divider ratio. In this case, using PLL_DIVSEL_FB = 2 and PLL_DIVSEL_REF = 1 results in a ratio of 2.

Configuring the Shared PMA PLL for PCI Express Operation

This example shows how to set the shared PMA PLL divider settings for PCI Express operation using Equation 5-1. The RocketIO GTP Transceiver Wizard and Table 5-3 are simpler alternatives. This example is provided only to illustrate the process with Equation 5-1.

Use Equation 5-1 as described in the following steps:

1. Determine the required line rates.

   For PCI Express operation, both TX and RX use a line rate of 2.5 Gb/s.

2. Determine the internal datapath width.

   Because PCI Express designs use 8B/10B encoding, an internal datapath width of 10 bits is required.

3. Determine the desired reference clock rate.

   This example uses a reference clock running at 100 MHz.

4. Calculate the required PLL clock rate.

   Because the SIPO block uses both edges of the clock to deserialize data, it must be fed a clock running at 2.5/2 = 1.25 GHz. Because this RX rate of 1.25 GHz is within the operating range of the PLL, the external divider (PLL_RXDIVSEL_OUT) must be one. The PLL clock rate is thus 1.25 x 1 = 1.25 GHz.

5. Calculate the required DIV value.

   Because the internal datapath width must be 10 bits (INTDATAWIDTH is High), DIV = 5.

6. Calculate the required PLL divider ratio.

   Using the values f_{CLKIN}, DIV, and f_{PLL_CLOCK} determined above, rearrange Equation 5-1 to calculate the divider ratio as shown in Equation 5-5. The result is a ratio of 2.5.

$$\frac{PLL\_DIVSEL\_FB}{PLL\_DIVSEL\_REF} = \frac{f_{PLL\_Clock}}{f_{CLKIN} \times DIV} = \frac{1.25\ GHz}{100\ MHz \times 5} = 2.5 \qquad \textit{Equation 5-5}$$

7. Select the PLL divider values.

   Select the smallest divider values that result in the required PLL divider ratio. In this case, using PLL_DIVSEL_FB = 5 and PLL_DIVSEL_REF = 2 results in a ratio of 2.5.

Optiver Ex1009
Page 79 of 347

Clocking

Overview

For proper high-speed operation, the GTP transceiver requires a high-quality, low-jitter, reference clock. Because of the shared PMA PLL architecture inside the GTP_DUAL tile, each reference clock sources both channels. The reference clock is used to produce the PLL clock, which is divided by one, two, or four to make individual TX and RX serial clocks and parallel clocks for each GTP transceiver. See Shared PMA PLL, page 72 for details.

The GTP_DUAL reference clock is provided through the CLKIN port. There are three ways to drive the CLKIN port (see Figure 5-3):

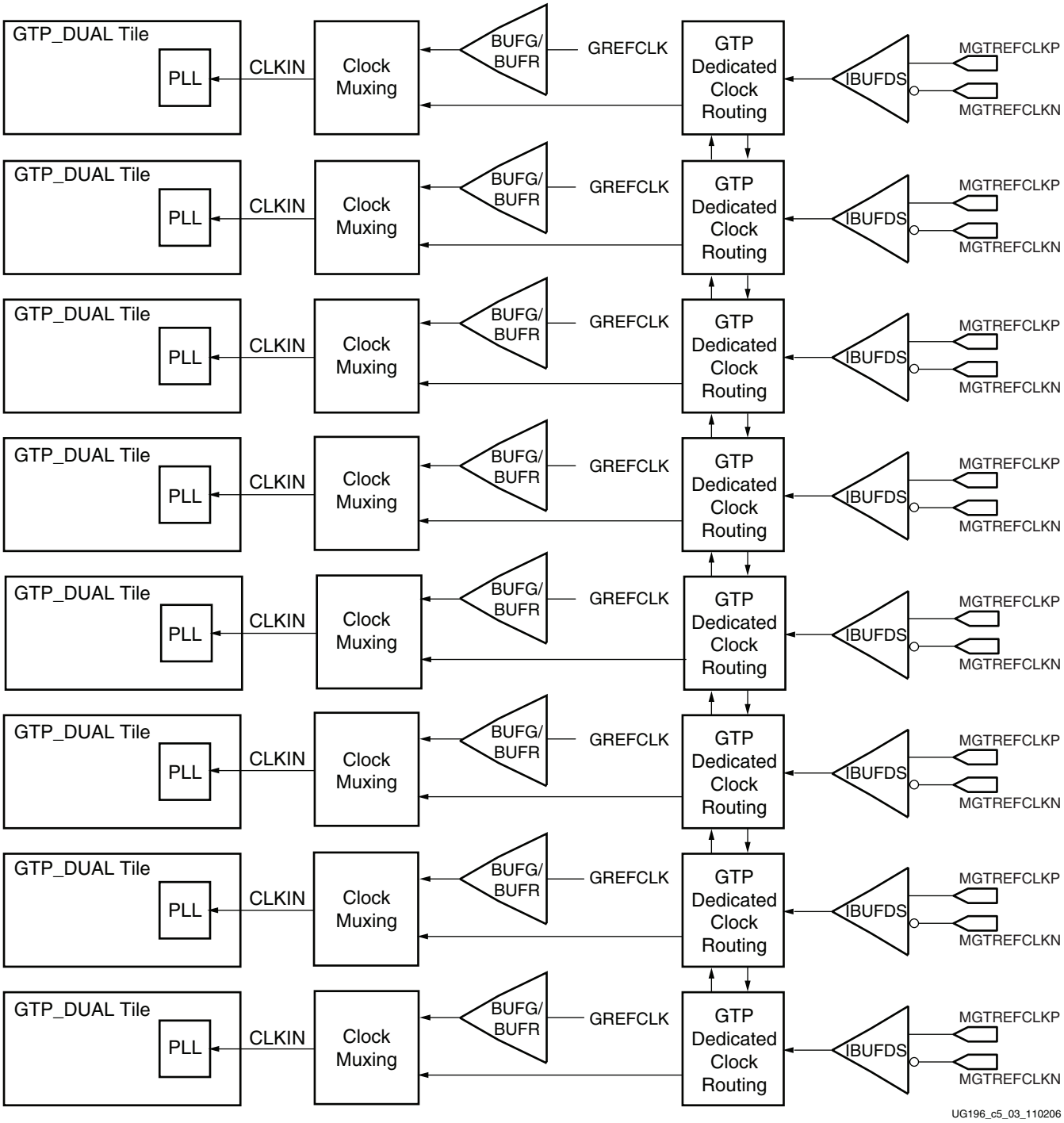
- Using an external oscillator to drive GTP dedicated clock routing
- Using a clock from a neighboring GTP_DUAL tile through GTP dedicated clock routing
- Using a clock from inside the FPGA (GREFCLK)

Using the dedicated clock routing provides the best possible clock to the GTP_DUAL tiles. Each GTP_DUAL tile has a pair of dedicated clock pins, represented by IBUFDS primitives, that can be used to drive the dedicated clock routing. Refer to REFCLK Guidelines in Chapter 10 for IBUFDS details.

This clocking section shows how to select the dedicated clocks for use by one or more GTP_DUAL tiles. Guidelines for driving these pins on the board are discussed in Chapter 10, GTP-to-Board Interface.

When GREFCLK clocking is used for a specific GTP_DUAL tile, the dedicated clock routing is not used. Instead, the global clock resources of the FPGA are connected to the shared PMA PLL. GREFCLK clocking is not recommended for most designs because of the increased jitter introduced by the FPGA clock nets.

Cypher Ex1009
Page 80 of 347



Notes:

1. Refer to REFCLK Guidelines in Chapter 10 for IBUFDS details.
2. Refer to Appendix F, Advanced Clocking, for advanced clocking scenarios. All GTP_DUAL tiles between the source of the reference clock and a tile using the reference clock must be instantiated in the design and must have REFCLKPWRDNB asserted High.

Figure 5-3: **GTP Transceiver Clocking**

Ports and Attributes

Table 5-4 defines the shared clocking ports.

Table 5-4: **Shared Clocking Ports**

Port	Dir	Clock Domain	Description
CLKIN	In	N/A	Reference clock input to the shared PMA PLL.
REFCLKOUT	Out	N/A	The REFCLKOUT port from each GTP_DUAL tile provides access to the reference clock provided to the shared PMA PLL (CLKIN). It can be routed for use in the FPGA logic.

Table 5-5 defines the shared clocking attributes.

Table 5-5: **Shared Clocking Attributes**

Attribute	Description
CLK25_DIVIDER	The internal digital logic for GTP_DUAL tile management runs at about 25 MHz. CLK25_DIVIDER is set to get an internal clock for the tile. 1: CLKIN $\leq$ 25 MHz 2: 25 MHz < CLKIN $\leq$ 50 MHz 3: 50 MHz < CLKIN $\leq$ 75 MHz 4: 75 MHz < CLKIN $\leq$ 100 MHz 5: 100 MHz < CLKIN $\leq$ 125 MHz 6: 125 MHz < CLKIN $\leq$ 150 MHz 10: 150 MHz < CLKIN $\leq$ 250 MHz 12: CLKIN > 250 MHz
CLKINDC_B	Must be set to TRUE. Oscillators driving the dedicated reference clock inputs must be AC coupled.

Description

Clocking from an External Source

Each GTP_DUAL tile has a pair of dedicated pins that can be connected to an external clock source. To use these pins, an IBUFDS primitive is instantiated. In the user constraints file (UCF), the IBUFDS input pins are constrained to the locations of the dedicated clock pins for the GTP_DUAL tile. In the design, the output of the IBUFDS is connected to the CLKIN port. The locations of the dedicated pins for all the GTP_DUAL tiles are documented in Chapter 4, Implementation. Chapter 10, GTP-to-Board Interface, provides a selection of suitable external oscillators and describes the board-level requirements for the dedicated reference clock. Figure 5-4 shows a differential GTP clock pin pair sourced by an external oscillator on the board. Refer to REFCLK Guidelines in Chapter 10 for IBUFDS details.

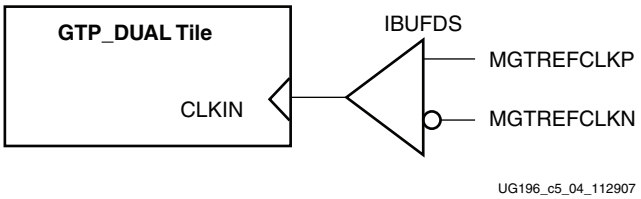


Figure 5-4: **Single GTP_DUAL Tile Clocked Externally**

Cpuver Ex1009
Page 82 of 347

Clocking from a Neighboring GTP_DUAL Tile

The external clock from one GTP_DUAL tile can be used to drive the CLKIN ports of neighboring tiles. The example in Figure 5-5 uses the clock from one GTP_DUAL tile to clock six neighboring tiles. A GTP_DUAL tile shares its clock with its neighbors using the dedicated clock routing resources. Refer to REFCLK Guidelines in Chapter 10 for IBUFDS details.

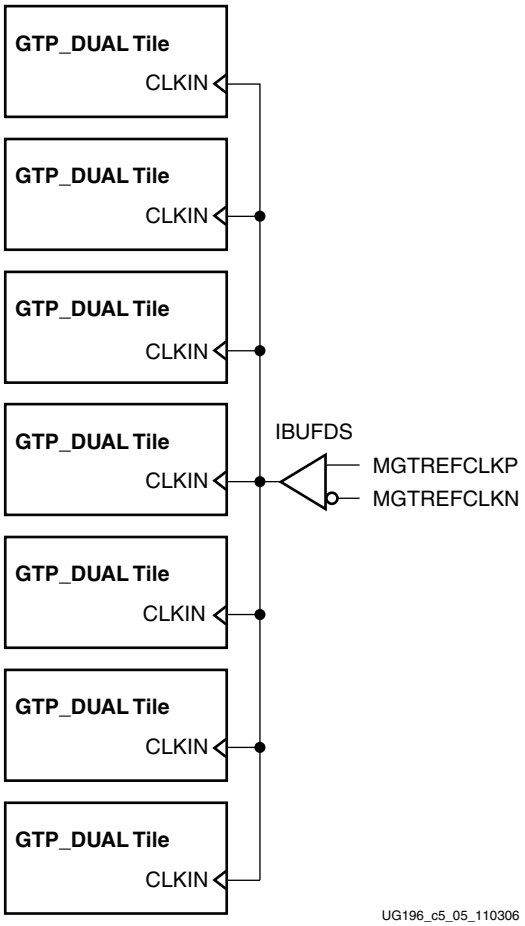


Figure 5-5: **Multiple GTP_DUAL Tiles with Shared Reference Clock**

Note: The following rules must be observed when sharing a reference clock to ensure that jitter margins for high-speed designs are met:

1. The number of GTP_DUAL tiles *above* the sourcing GTP_DUAL tile must *not* exceed three.
2. The number of GTP_DUAL tiles *below* the sourcing GTP_DUAL tile must *not* exceed three.
3. The total number of GTP_DUAL tiles sourced by the external clock pin pair (MGTREFCLKN/MGTREFCLKP) must *not* exceed seven.
4. All the GTP_DUAL tiles between the source of the reference clock and a tile using the reference clock, including the tile with an IBUFDS in use, must be instantiated in the design and must have REFCLKPWRDNB asserted High.

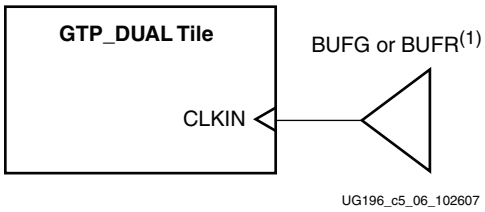
The maximum number of GTP transceivers that can be sourced by a single clock pin pair is 14. Designs with more than 14 transceivers require the use of multiple external clock pins to ensure that the rules for controlling jitter are followed. When multiple clock pins are used, an external buffer can be used to drive them from the same oscillator. The same

oscillator must be used when the GTP transceivers are combined to form a single channel using channel bonding (see Configurable Channel Bonding (Lane Deskew), page 189).

Clocking using GREFCLK

The internal clock nets of the FPGA can provide the reference clock for the GTP_DUAL tile by connecting the output of a global clock buffer (BUFG) or a regional clock buffer (BUFR) to the CLKIN port. This type of clocking, called GREFCLK clocking, has the lowest performance of any of the three clocking methods, because FPGA clocking resources introduce too much jitter for operation at high rates. GREFCLK clocking should be avoided, if possible. See the *Virtex-5 FPGA Data Sheet* for the jitter margins at different speeds.

Figure 5-6 shows how a GTP_DUAL tile connects to a BUFR or a BUFG. If a BUFR is used, it must be located in the same region as the GTP_DUAL tile.



Notes:

1. Refer to the *Virtex-5 FPGA Data Sheet* and the *Virtex-5 FPGA Configuration Guide* for the maximum clock frequency and jitter limitations of BUFR.

Figure 5-6: **Single GTP_DUAL Tile Clocked from the FPGA**

Reset

Overview

The GTP_DUAL tile must be reset before any of the GTP transceivers can be used. There are three ways to reset a GTP_DUAL tile:

1. Power up and configure the FPGA. Power-up reset is covered in this section.
2. Drive the GTPRESET port High to trigger a full asynchronous reset of the GTP_DUAL tile. GTPRESET is covered in this section.
3. Assert one or more of the individual reset signals on the block to reset a specific subcomponent of the tile. These resets are covered in detail in the sections for each subcomponent.

This section also includes the instructions for implementing the Link Idle Reset circuit. This circuit must be implemented with all instances of the GTP_DUAL tile to allow the RX CDR circuit to operate correctly.

Apotex Ex1009
Page 84 of 347

Ports and Attributes

Table 5-6 defines the shared tile reset ports.

Table 5-6: **Shared Tile Reset Ports**

Port	Dir	Domain	Description
GTPRESET(1,2)	In	Async	This port is driven High to start the full GTP_DUAL reset sequence. This sequence takes about 160 µs to complete, and systematically resets all subcomponents of the GTP_DUAL tile.
PLLPOWERDOWN(1,2)	In	Async	Powers down the shared PMA PLL. Driving PLLPOWERDOWN from High to Low triggers a GTPRESET.
PRBSCNTRESET0 PRBSCNTRESET1	In	RXUSRCLK2	Resets the PRBS error counter.
RESETDONE0 RESETDONE1	Out	Async	This port goes High when the GTP transceiver has finished reset and is ready for use. For this signal to work correctly, CLKIN and all clock inputs on the individual GTP transceiver (TXUSRCLK, TXUSRCLK2, RXUSRCLK, RXUSRCLK2) must be driven. If either RXPOWERDOWN or TXPOWERDOWN equals `11` after a GTPRESET, RXRESET, or TXRESET, the RESETDONE does not go High.
RXBUFRESET0(1) RXBUFRESET1(2)	In	Async	This active-High signal resets the RX buffer logic.
RXCDRRESET0(1) RXCDRRESET1(2)	In	RXUSRCLK2	Individual reset signal for the RX CDR and the RX part of the PCS for this channel. This signal is driven High to cause the CDR to give up its current lock and return to the shared PMA PLL frequency.
RXELECIDLERESET0 RXELECIDLERESET1	In	Async	These active-High reset inputs reset the GTP transceiver receive logic when the link is in a power-down state. Figure 5-9 shows how these signals are connected. If the link idle reset is not supported, these signals are strapped Low.
RXENELECIDLERESETB	In	Async	When asserted, this active-Low signal enables the RXELECIDLERESET(0/1) inputs. Figure 5-9 shows how this signal is connected when the link idle reset is supported. If RXELECIDLERESET(0/1) are not used, this signal is strapped High.
RXRESET0(1) RXRESET1(2)	In	Async	Active-High reset for the RX PCS logic.
TXRESET0(1) TXRESET1(2)	In	Async	Resets the PCS of the GTP transmitter, including the phase adjust FIFO, the 8B/10B encoder, and the FPGA TX interface.

Notes:

1. When these resets are active, RESETDONE0 is driven Low. All resets are asynchronous, negative-edge triggered, and synchronized internally to a specific clock domain. When either RXPOWERDOWN0 or TXPOWERDOWN0 equals `11` after a GTPRESET, RXRESET0, or TXRESET0, the RESETDONE0 does not go High.
2. When these resets are active, RESETDONE1 is driven Low. All resets are asynchronous, negative-edge triggered, and synchronized internally to a specific clock domain. When either RXPOWERDOWN1 or TXPOWERDOWN1 equals `11` after a GTPRESET, RXRESET1, or TXRESET1, the RESETDONE1 does not go High.

There are no attributes in this section.

Description

GTP Reset in Response to Completion of Configuration

Figure 5-7 shows the GTP_DUAL reset sequence following completion of configuration of a powered-up GTP_DUAL tile. The same sequence is activated any time PLLPOWERDOWN goes from High to Low during normal operation.

Refer to Power Control, page 94 on power-down for details about PLLPOWERDOWN.

Notes:

1. The timing of the reset sequencer inside the GTP_DUAL tile depends on the frequency of CLK25. The estimates given in this figure assume that the frequency of CLK25 is 25 MHz.

Figure 5-7: **GTP_DUAL Reset Sequence Following Configuration**

The following GTP_DUAL sections are affected by the reset sequence after configuration:

- Shared PMA PLL
- GTP0 transmit section (PMA and PCS)
- GTP0 receive section (PMA and PCS)
- GTP1 transmit section (PMA and PCS)
- GTP1 receive section (PMA and PCS)

Apotex Ex1009
Page 86 of 347

GTP Reset When the GTPRESET Port is Asserted

Figure 5-8 is similar to Figure 5-7, showing the full reset sequence occurring in response to the falling edge of a pulse on GTPRESET. GTPRESET acts as an asynchronous reset signal.

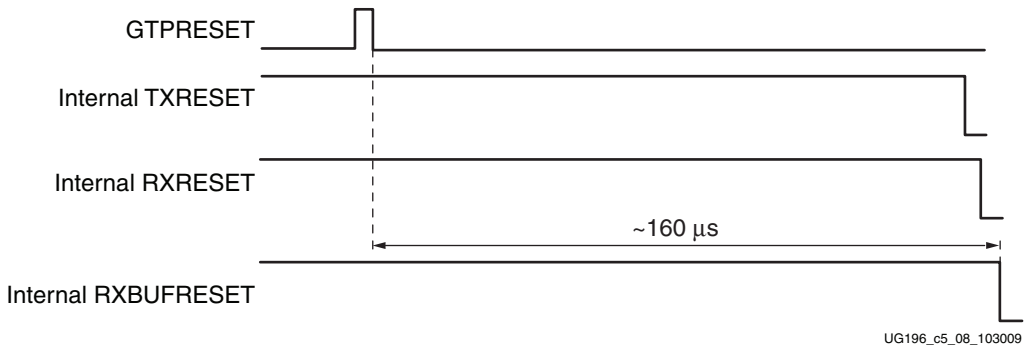


Figure 5-8: **Reset Sequence Triggered by the GTPRESET Pulse**

The following GTP_DUAL sections are affected by the GTPRESET sequence:

- Shared PMA PLL
- GTP0 transmit section (PMA and PCS)
- GTP0 receive section (PMA and PCS)
- GTP1 transmit section (PMA and PCS)
- GTP1 receive section (PMA and PCS)

GTP Component-Level Resets

Component resets are primarily used for special cases. These resets are needed when only the reset of a specific GTP_DUAL subsection is required. Table 5-6, page 85 provides an overview of component level resets. Table 5-7 describes the component level resets.

All component resets are asynchronous with the exception of PRBSCNTRESET, which is synchronous to RXUSRCLK2 and is effective only on its rising edge.

Link Idle Reset Support

During operation, an electrical idle condition can occur on the GTP receiver, causing RXELECIDLE to be driven High. The following events can cause an RX electrical idle condition:

- An open RXP/RXN differential input pair
- A transmitter on the other side of the communication link powers down
- An OOB/beacon signaling sequence
- All designs using the second-order loop of the RXCDR (e.g., PCI Express designs).

During an electrical idle condition the Clock Data Recovery (CDR) circuit in the receiver can lose lock. To restart the CDR after an electrical idle condition, RXELECIDLERESET and RXENELCIDLERESETB must be asserted. RX Clock Data Recovery, page 149 describes the RXELECIDLERESET, RXENELECIDLERESETB, and the CDR circuit in more detail.

Figure 5-9 shows the Link Idle Reset circuit required in all designs using the second-order loop of the RX CDR. Because RESETDONE is used in the circuit, TXUSRCLK, TXUSRCLK2, RXUSRCLK, and RXUSRCLK2 must all be clocked on active GTP transceivers.

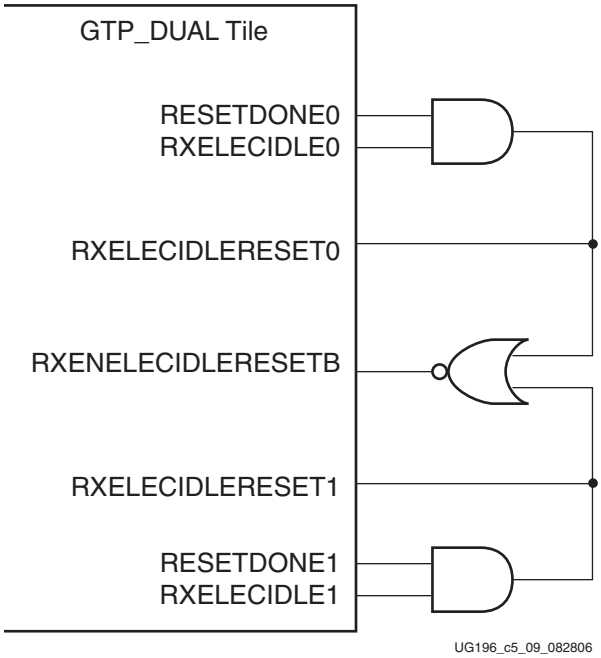


Figure 5-9: **Link Idle Reset Implementation**

Note: If a RXRECCLK is used to generate or derive any of the USRCLKs and an Electrical Idle condition occurs, the derived USRCLKs will flatline, because RXRECCLK flatlines when the generating CDR is in reset. In this case, RXELECIDLE(0/1) can be used as a selection signal of a BUFGMUX to multiplex between the RXRECCLK(0/1) and a different CDR independent clock source.

Cpuver Ex1009
Page 88 of 347

Resetting the GTP_DUAL Tile

Each GTP_DUAL tile offers several ways to reset its subcomponents. Table 5-7 shows all the different ways of resetting a GTP_DUAL tile, and the subcomponents that are affected by each type of reset.

Table 5-7: **Available Resets Pins and the Components Reset by these Reset Pins**

	Component	Configuration	GTPRESET	PLLPOWERDOWN (Falling Edge)	TXRESET	RXCDRRESET	RXRESET	RXBUFRESET	RXELECIDLERESET	PRBSCNTRESET
GTP-to-Board Interface	Termination Resistor Calibration	✓								
Shared Resources	Shared PMA PLL	✓	✓	✓						
	PLL Lock Detection	✓	✓	✓						
	Reset Control	✓	✓	✓						
	Power Control	✓	✓	✓						
	Clocking	✓	✓	✓						
	DRP	✓								
TX PCS	FPGA TX Interface	✓	✓	✓	✓					
	8B/10B Encoder	✓	✓	✓	✓					
	TX Buffer	✓	✓	✓	✓					
	PRBS Generator	✓	✓	✓	✓					
	Polarity Control	✓	✓	✓	✓					
TX PMA	PISO	✓	✓	✓						
	TX Pre-emphasis	✓	✓	✓						
	TX OOB and PCI Express	✓	✓	✓						
	TX Driver	✓	✓	✓						
RX PCS	FPGA RX Interface	✓	✓	✓		✓	✓			
	RX Elastic Buffer	✓	✓	✓		✓	✓	✓		
	RX Status Control	✓	✓	✓		✓	✓			
	8B/10B Decoder	✓	✓	✓		✓	✓			
	Comma Detect and Align	✓	✓	✓			✓			
	RX LOS State Machine	✓	✓	✓		✓	✓			
	RX Polarity	✓	✓	✓		✓	✓			
	PRBS Checker	✓	✓	✓		✓	✓			✓
	5x Oversampler	✓	✓	✓		✓	✓			

Table 5-7: **Available Resets Pins and the Components Reset by these Reset Pins *(Continued)***

	Component	Configuration	GTPRESET	PLLPOWERDOWN (Falling Edge)	TXRESET	RXCDRRESET	RXRESET	RXBUFRESET	RXELECIDLERESET	PRBSCNTRESET
RX PMA	SIPO	✓	✓	✓		✓			✓	
	RX CDR	✓	✓	✓		✓			✓	
	RX Termination and Equalization	✓	✓	✓						
	RX OOB	✓	✓	✓		✓				
Loopback	Loopback paths	✓	✓	✓						

The reset that occurs after configuration and the GTPRESET port are the most common ways to prepare GTP_DUAL tile(s) for operation, but certain situations can require the use of other reset ports. Table 5-8 outlines some of these situations and the recommended resets.

Table 5-8: **Recommended Resets for Common Situations**

Situation	Components to be Reset	Recommended Reset(1)
After power-up and configuration	Entire GTP_DUAL tile	Reset after configuration is automatic
After turning on a reference clock	Shared PMA PLL	GTPRESET
After changing a reference clock	Shared PMA PLL	GTPRESET
After coming out of RXPOWERDOWN (RXPOWERDOWN not equal to `00`)	RX PCS	RXRESET
Restart comma alignment	RX PCS	RXRESET
After changing the reference clock divider settings with the DRP	Shared PMA PLL	GTPRESET
Parallel clock source reset	TX PCS, RX PCS, Phase Alignment	TXRESET, RXRESET
After remote power-up	RX CDR	RXELECIDLERESET
After an electrical idle condition in SATA OOB/PCI Express operations	RX CDR	RXELECIDLERESET
After connecting RXN/RXP	RX CDR	RXELECIDLERESET, RXRESET
After a TX buffer error	TX buffer	TXRESET
After an RX buffer error	RX buffer	RXBUFRESET
Before channel bonding	RX CDR, then RXBUFFER after CDR is locked	RXELECIDLERESET, RXBUFRESET
After a PRBS error	PRBS error counter	PRBSCNTRESET
After an oversampler error	Oversampler	RXRESET

Notes:

1. The recommended reset has the smallest impact on the other components of the GTP_DUAL tile.

Cpuver Ex1009
Page 90 of 347

Examples

Power-up and Configuration

All GTP_DUAL tiles are reset automatically after configuration. The supplies for the calibration resistor and calibration resistor reference must be powered up before configuration to ensure correct calibration of the termination impedance of all transceivers.

After Turning on a Reference Clock

The reference clock source(s) and the power to the GTP_DUAL tile must be available before configuring the FPGA. The reference clock must be stable before configuration, especially when using PLL based clock sources (e.g., voltage controlled crystal oscillators). If the reference clock(s) or GTP_DUAL tile(s) are powered up after configuration, apply GTPRESET to allow the shared PMA PLL(s) to lock.

After Coming out of RXPOWERDOWN (RXPOWERDOWN Not Equal to 00)

When switching from a power-down state (RXPOWERRDOWN equal to `01`, `10`, or `11`) to normal operation (RXPOWERDOWN equal to `00`) the RX PCS must be reset by using RXRESET. Refer to Power Control, page 94 for additional information about RXPOWERDOWN.

After Changing a Reference Clock

Whenever the reference clock input to a GTP_DUAL tile is changed, the shared PMA PLL must be reset afterwards to ensure that it locks to the new frequency. The GTPRESET port must be used for this purpose.

Parallel Clock Source Reset

The clocks driving TXUSRCLK, RXUSRCLK, TXUSRCLK2, and RXUSRCLK2 must be stable for correct operation. These clocks are often driven from a PLL or DCM in the FPGA to meet phase and frequency requirements. If the DCM or PLL loses lock and begins producing incorrect output, TXRESET and RXRESET must be used to hold transceiver PCS in reset until the clock source is locked again.

If the TX or RX buffer is bypassed and phase alignment is in use, phase alignment must be performed again after the clock source relocks.

After Remote Power-up

If the remote source of incoming data is powered up after the GTP transceiver receiving its data is operating, the RX CDR must be reset to ensure a clean lock to the incoming data. RXELECIDLERESET must be used for this purpose.

Restart Comma Alignment

Restart the comma alignment when:

- The RXCDRESET sequence is complete
- RXBUFRESET is used

This is important when ALIGN_COMMA_WORD equals 2. Refer Configurable Comma Alignment and Detection, page 162 for additional details.

Electrical Idle Reset

When the differential voltage of the RX input to a GTP transceiver drops to OOB or electrical idle levels, the RX CDR can be pulled out of lock by the apparent sudden change in frequency. To ensure the CDR can relock, it must be held in reset until the signal returns using RXELECIDLERESET.

Figure 5-9, page 88 shows the Link Idle Reset circuit. This circuit is required for designs using electrical idle or OOB/beacon signals. The circuit asserts RXELECIDLERESET whenever RXELECIDLE is detected. RXELECIDLE is asserted whenever the RXOOB circuit is reset. RESETDONE prevents the RXELECIDLERESET signal from being asserted while another reset is in progress.

After Connecting RXP/RXN

When the RX data to the GTP transceiver comes from a connector that can be plugged in and unplugged, the RX CDR must be reset when the data source is plugged in to ensure that it can lock to incoming data. RXELECIDLERESET is used to perform this reset. RXRESET must be applied after RXELECIDLERESET to flush the buffer, prevent an underflow or overflow from occurring, and re-initialize the comma alignment circuit.

After a TX Buffer Error

When the TX buffer overflows or underflows, it must be reset using TXRESET to ensure correct behavior.

After an RX Buffer Error

After an RX buffer overflow or underflow, the RX elastic buffer must be reset using the RXBUFRESET port to ensure correct behavior.

Before Channel Bonding

For successful channel bonding, the RX elastic buffers of all the bonded transceivers must be written using the same recovered frequency and read using the same RXUSRCLK frequency.

To provide the same RXUSRCLK frequency to all bonded transceivers, a low-skew clock buffer (for example, a BUFG) is used to drive all the RXUSRCLK ports from the same clock source. Bonding should not be attempted until the clock source is stable.

To provide the same recovered clock to all bonded transceivers:

- All TX data sources must be locked to the same reference clock
- All bonded transceivers must have CDR lock to the incoming data

The required reset for channel bonding is as follows:

- Assert RXELECIDLERESET to reset the CDR for all bonded transceivers
- Wait for CDR lock and bit alignment on all bonded transceivers
- Apply RXBUFRESET to all bonded transceivers
- Attempt channel bonding

See RX Clock Data Recovery in Chapter 7 for recommended methods of detecting CDR lock.

Computer Ex1009
Page 92 of 347

After a PRBS Error

To clear the RXPRBSERR signal after the PRBS error threshold is exceeded, PRBSERRRESET should be asserted.

After an Oversampler Error

If RXOVERSAMPLEERR goes High to indicate an overflow or underflow in the oversampling block, asserting RXRESET clears it.

Power Control

Overview

The GTP_DUAL tiles support a range of power-down modes. These modes support both generic power management capabilities as well as those defined in the PCI Express and SATA specifications.

Ports and Attributes

Table 5-9 defines the power ports.

Table 5-9: **Power Ports**

Port	Dir	Domain	Description
CLKIN	In	N/A	Reference clock input to the shared PMA PLL. The CLKIN rate in conjunction with CLK25_DIVIDER determines the timing of power-down state transitions for PCI Express operation.
PLLPOWERDOWN(1)	In	Async	Powers down the shared PMA PLL: `0`: Shared PMA PLL is powered up `1`: Shared PMA PLL is powered down
REFCLKPWRDNB(2)	In	Async	Powers down the GTP reference clock circuit: `0`: Reference clock circuit is powered down `1`: Reference clock circuit is powered up
RXPOWERDOWN0[1:0] RXPOWERDOWN1[1:0](3)	In	Async	Powers down the RX lanes. The encoding complies with the PCI Express encoding. TX and RX can be powered down separately. However, for PCI Express compliance, TXPOWERDOWN and RXPOWERDOWN have to be used together. `00`: P0 (normal operation) `01`: P0s (low recovery time power down) `10`: P1 (longer recovery time, Receiver Detection is still on) `11`: P2 (lowest power state). In the P2 power state, RXRECCLK of this GTP transceiver is indeterminate. It is either a static `1` or a static `0`.
TXDETECTRX0 TXDETECTRX1	In	TXUSRCLK2	Depending on the state of RXPOWERDOWN and TXPOWERDOWN, TXDETECTRX activates the receive detection sequence or selects the PCI Express complaint loopback mode (see Table 5-13, page 97). The receive detection sequence ends when PHYSTATUS is asserted to indicate that the results of the test are ready on RXSTATUS.
TXELECIDLE0 TXELECIDLE1	In	TXUSRCLK2	Drives TXN and TXP to the same voltage to perform electrical idle/beaconing for PCI Express operation.

Apotex Ex1009
Page 94 of 347

Table 5-9: **Power Ports *(Continued)***

Port	Dir	Domain	Description
TXPOWERDOWN0[1:0] TXPOWERDOWN1[1:0]	In	Async	Powers down the TX lanes. The encoding complies with the PCI Express encoding. TX and RX can be powered down separately, however, for PCI Express compliance, TXPOWERDOWN and RXPOWERDOWN have to be used together. `00`: P0 (normal operation) `01`: P0s (low recovery time power down) `10`: P1 (longer recovery time, Receiver Detection is still on) `11`: P2 (lowest power state)

Notes:

1. Because of the shared PMA PLL, a power down via PLLPOWERDOWN or REFCLKPWRDNB affects both channels.
2. REFCLKPWRDNB must be High when a GTP_DUAL tile is sourcing a reference clock.
3. When switching from a power-down state (RXPOWERRDOWN equal to `01`, `10`, or `11`) to normal operation (RXPOWERDOWN equal to `00`), the RX PCS must be reset by using RXRESET. When either RXPOWERDOWN or TXPOWERDOWN equals `11` after a GTPRESET, RXRESET, or TXRESET, the RESETDONE does not go High.

Table 5-10 defines the power attributes.

Table 5-10: **Power Attributes**

Attribute	Description
CLK25_DIVIDER	The internal digital logic for GTP_DUAL tile management runs at about 25 MHz. CLK25_DIVIDER is set to get an internal clock for the tile. The CLK25_DIVIDER in conjunction with CLKIN determines the timing of power-down state transitions for PCI Express operation by adjusting the internal 25 MHz clock rate.
PCI_EXPRESS_MODE_0 PCI_EXPRESS_MODE_1	Setting this attribute to TRUE enables certain operations specific to PCI Express designs, specifically, recognizing TXELECIDLE = `1`, TXCHARDISPMODE = `1`, TXCHARDISPVAL = `0` as a request to power down the channel. TXCHARDISPMODE = `1` and TXCHARDISPVAL = `0` encode the PIPE interface signal TXCompliance = `1` of the PIPE specification. The TXCHARDISPMODE and TXCHARDISPVAL settings encode for PIPE and enable special support for fast training sequence (FTS) lane deskew.
TRANS_TIME_FROM_P2_0 TRANS_TIME_FROM_P2_1	Transition time from the P2 state in internal 25 MHz clock cycles. The exact time depends on the CLKIN rate and the setting of CLK25_DIVIDER. The P2 state is related to the PCI Express power state definition.
TRANS_TIME_NON_P2_0 TRANS_TIME_NON_P2_1	Transition time to or from any state except P2 in internal 25 MHz clock cycles. The exact time depends on the CLKIN rate and the setting of CLK25_DIVIDER. This setting is related to the PCI Express power state definition.
TRANS_TIME_TO_P2_0 TRANS_TIME_TO_P2_1	Transition time to the P2 state in internal 25 MHz clock cycles. The exact time depends on the CLKIN rate and the setting of CLK25_DIVIDER. This setting is related to the PCI Express power state definition.

Description

The GTP_DUAL tile offers different levels of power control. Each channel in each direction can be powered down separately using TXPOWERDOWN and RXPOWERDOWN. Additionally, the shared PMA PLL and the reference clock section can be powered down, which affects both channels and both directions. The ports PLLPOWERDOWN and REFCLKPWRDNB directly affect the shared PMA PLL and therefore both channels of the GTP_DUAL tile.

Generic GTP_DUAL Power-Down Capabilities

The GTP_DUAL tile provides several power-down features that can be used in a wide variety of applications. Table 5-11 summarizes these capabilities.

Table 5-11: **Basic Power-Down Functions Summary**

Function	Controlled By	Affects
REFCLK Power Down	REFCLKPWRDNB	TX and RX for both transceivers in a tile, and all downstream GTP_DUAL tiles sharing that REFCLK
PLL Power Down	PLLPOWERDOWN	TX and RX for both transceivers in a GTP_DUAL tile
TX Power Down	TXPOWERDOWN[1:0]	TX in a single transceiver
RX Power Down(1)	RXPOWERDOWN[1:0]	RX in a single transceiver

Notes:

1. When RXPOWERDOWN[1:0] is set to `11`, which is the lowest power state (P2), then RXRECCLK of this transceiver is indeterminate. RXRECCLK of this GTP transceiver is either a static `1` or a static `0`.

REFCLK Power Down

To activate the REFCLK power-down mode, the active-Low REFCLKPWRDNB signal is asserted. When REFCLKPWRDNB is asserted, toggling of all circuitry clocked by the REFCLK input is suppressed, including the shared PMA PLL and all clocks derived from it. In addition, asserting REFCLKPWRDNB disables the dedicated clock routing circuitry associated with that tile. If the GTP_DUAL tiles share a common reference, REFCLK is suppressed to tiles that are downstream in the clock routing chain. Figure 5-3, page 81 illustrates how the dedicated clock routing blocks forward REFCLKs between GTP_DUAL tiles.

Recovery from this power state is indicated by the assertion of the PLLLKDET signal on the tile whose REFCLKPWRDNB signal is asserted and all downstream tiles whose reference clocks are affected.

PLL Power Down

To activate the PLL power-down mode, the active-High PLLPOWERDOWN signal is asserted. When PLLPOWERDOWN is asserted, the shared PMA PLL and all clocks derived from it are stopped.

Recovery from this power state is indicated by the assertion of the PLLLKDET signal on the tile whose REFCLKPWRDNB signal is asserted.

Apotex Ex1009
Page 96 of 347

TX and RX Power Down

When the TX and RX power-down signals are used in non PCI Express implementations, TXPOWERDOWN and RXPOWERDOWN can be used independently. However, when these interfaces are used in non PCI Express applications, only two power states are supported, as shown in Table 5-12. When using this power-down mechanism, the following must be TRUE:

- TXPOWERDOWN[1] and TXPOWERDOWN[0] are connected together.
- RXPOWERDOWN[1] and RXPOWERDOWN[0] are connected together.
- TXDETECTRX must be strapped Low.
- TXELECIDLE must be strapped to TXPOWERDOWN[1] and TXPOWERDOWN[0].

Table 5-12: **TX and RX Power States for Non PCI Express Operation**

TXPOWERDOWN[1:0] or RXPOWERDOWN[1:0]	Description
00	P0 mode. Transceiver TX or RX is actively sending or receiving data.
11	P2 mode. Transceiver TX or RX is idle.

Power-Down Features for PCI Express Operation

The GTP_DUAL tile implements all functions needed for power-down states compatible with those defined in the PCI Express and PIPE specifications. When implementing PCI Express compatible power control, the following conditions must be met:

- The TXPOWERDOWN and RXPOWERDOWN signals on each GTP transceiver must be connected together to ensure that they are in the same state at all times.
- The REFCLKPWRDNB and PLLPOWERDOWN signals must be held in an inactive state.

Table 5-13: **TX and RX Power States for PCI Express Operation**

TXPOWERDOWN[1:0] and RXPOWERDOWN[1:0]	TXDETECTRX	TXELECIDLE	Description
00 (P0 state)	0	0	The PHY is transmitting data. The MAC provides data bytes to be sent every clock cycle.
	0	1	The PHY is not transmitting and is in the electrical idle state.
	1	0	The PHY goes into loopback mode.
	1	1	Not permitted.
01 (P0s state)	Don't Care	0	The MAC should always put the PHY into the electrical idle state while in the P0s state. The PHY behavior is undefined if TXELECIDLE is deasserted while in the P0s or P1 state.
		1	The PHY is not transmitting and is in the electrical idle state.

Table 5-13: **TX and RX Power States for PCI Express Operation *(Continued)***

TXPOWERDOWN[1:0] and RXPOWERDOWN[1:0]	TXDETECTRX	TXELECIDLE	Description
`10` (P1 state)	Don't Care	0	Not permitted. The MAC must always put the PHY into the electrical idle state while in the P1 state. The PHY behavior is undefined if TXELECIDLE is deasserted while in the P0s or P1 state.
	0	1	The PHY is idle.
	1	1	The PHY does a receiver detection operation.
`11` (P2 state)	Don't Care	0	The PHY transmits beacon signaling.
		1	The PHY is idle.

The GTP transceiver acknowledges changes in the PCI Express power mode by asserting the PHYSTATUS signal for one clock cycle.

Power-Down Transition Times

The delays between changes in the power-down state when TXPOWERDOWN and RXPOWERDOWN are changed are controlled by the TRANS_TIME_FROM_P2, TRANS_TIME_NON_P2, and TRANS_TIME_TO_P2 attributes as described in Table 5-10, page 95.

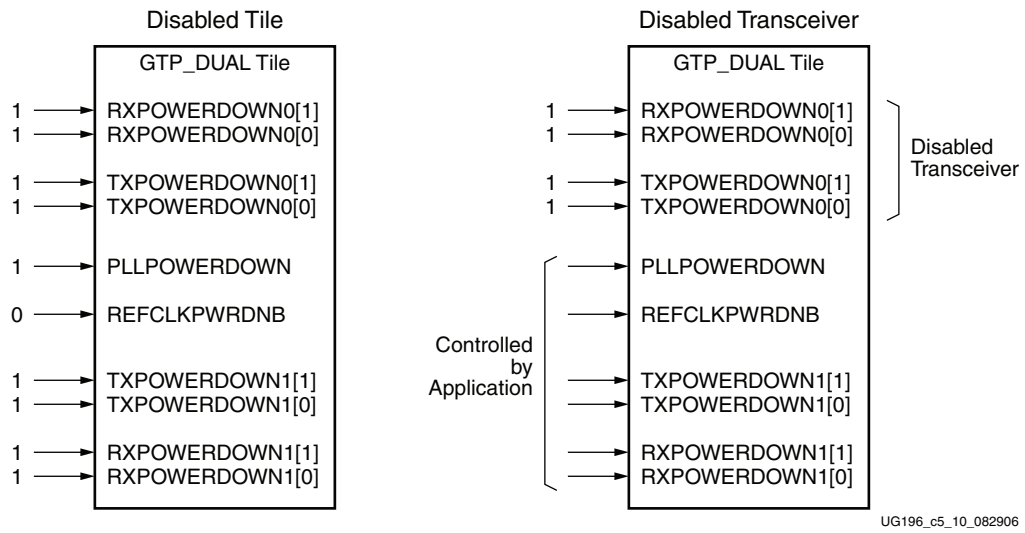
Each TRANS_TIME delay is set in terms of internal 25 MHz clock cycles. The internal 25 MHz clock rate is set using the CLK25_DIVIDER attribute and the reference clock rate. Equation 5-6 is used to determine the actual rate.

$$\text{Transition time in ns} = \left(\frac{\text{CLK25\_DIVIDER}}{\text{CLKIN}}\right) \times \text{TRANS\_TIME attribute} \qquad \textit{Equation 5-6}$$

Examples

The example in Figure 5-10 shows the recommended method to power down an unused tile or an unused transceiver in a tile.

Opulver Ex1009
Page 98 of 347



Notes:

1. REFCLKPWRDNB must be High when a GTP_DUAL tile is sourcing a reference clock.

Figure 5-10: **Powering Down an Unused Tile**

Figure 5-11 shows how to connect power-down signals for a four-lane PIPE compatible configuration.

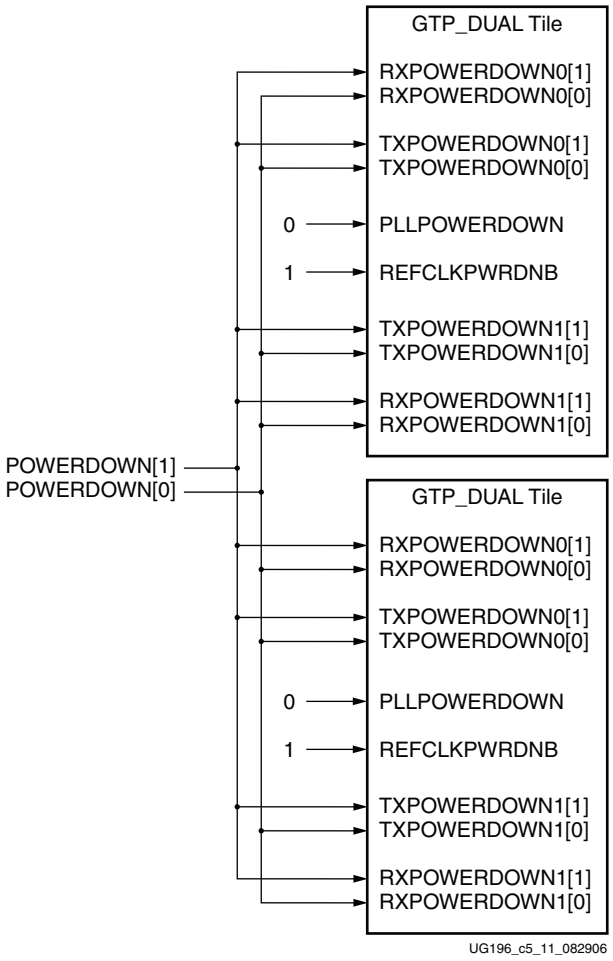


Figure 5-11: **4x PIPE Compatible Configuration**

Apotex Ex1009
Page 100 of 347

Dynamic Reconfiguration Port

Overview

The dynamic reconfiguration port (DRP) allows the dynamic change of parameters of the GTP_DUAL tile. The DRP interface is a processor-friendly synchronous interface with an address bus (DADDR) and separated data buses for reading (DO) and writing (DI) configuration data to the GTP_DUAL tile. An enable signal (DEN), a read/write signal (DWE), and a ready/valid signal (DRDY) are the control signals that implement read and write operations, indicate operation completion, or indicate the availability of data.

Ports and Attributes

Table 5-14 defines the DRP ports.

Table 5-14: **DRP Ports**

Port	Dir	Clock Domain	Description
DADDR[6:0]	In	DCLK	DRP address bus.
DCLK	In	N/A	DRP interface clock.
DEN	In	DCLK	Set to `1` to enable a read or write operation. Set to `0` on DCLK cycles where no operation is required.
DI[15:0]	In	DCLK	Data bus for writing configuration data from the FPGA logic to the GTP_DUAL tile.
DO[15:0]	Out	DCLK	Data bus for reading configuration data from the GTP_DUAL tile to the FPGA logic.
DRDY	Out	DCLK	Indicates operation is complete for write operations or data is valid for read operations.
DWE	In	DCLK	Set to `0` for read operations. Set to `1` for write operations.

There are no attributes in this section.

Description

The *Virtex-5 FPGA Configuration Guide* provides detailed information on the DRP interface. Refer to Appendix D, DRP Address Map of the GTP_DUAL Tile, for a map of GTP_DUAL DRP attributes sorted alphabetically by name and by address.

Stopping the reference clock during a DRP operation can prevent the correct termination of the operation.

Optiver Ex1009
Page 101 of 347

Cpuver Ex1009
Page 102 of 347

Chapter 6

GTP Transmitter (TX)

This chapter shows how to configure and use each of the functional blocks inside the GTP transmitter.

Transmitter Overview

Each GTP transceiver in the GTP_DUAL tile includes an independent transmitter, made up of a PCS and a PMA. Figure 6-1 shows the functional blocks of the transmitter. Parallel data flows from the FPGA into the FPGA TX interface, through the PCS and PMA, and then out the TX driver as high-speed serial data. Refer to Appendix E, Low Latency Design, for latency information on this block diagram.

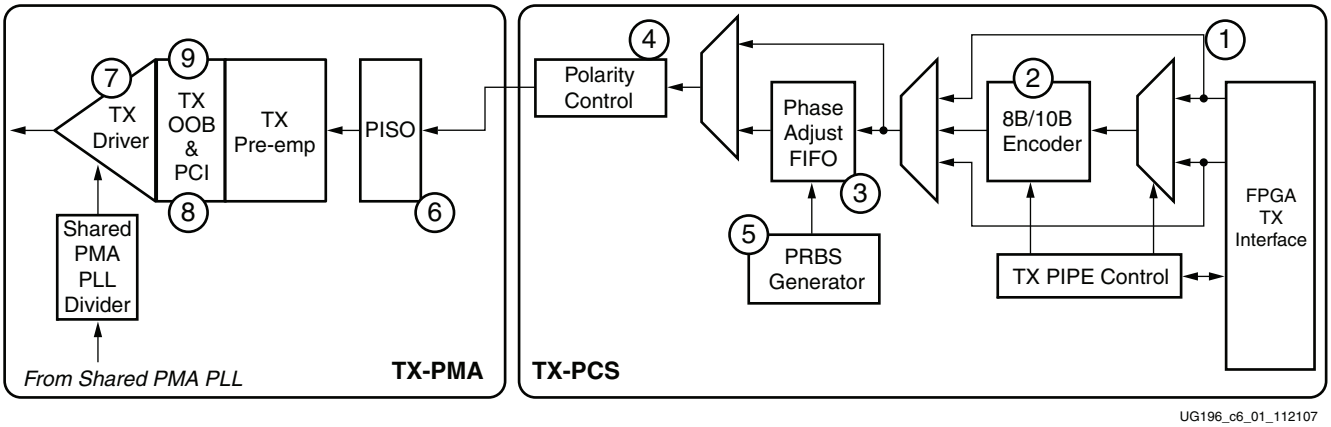


Figure 6-1: **GTP TX Block Diagram**

The key elements within the GTP transmitter are:

1. FPGA TX Interface, page 104
2. Configurable 8B/10B Encoder, page 111
3. TX Buffering, Phase Alignment, and TX Skew Reduction, page 115
4. TX Polarity Control, page 122
5. TX PRBS Generator, page 123
6. Parallel In to Serial Out, page 124
7. Configurable TX Driver, page 126
8. Receive Detect Support for PCI Express Operation, page 130
9. TX OOB/Beacon Signaling, page 133

Optiver Ex1009
Page 103 of 347

FPGA TX Interface

Overview

The FPGA TX interface is the FPGA's gateway to the TX datapath of the GTP transceiver. Applications transmit data through the GTP transceiver by writing data to the TXDATA port on the positive edge of TXUSRCLK2.

The width of the port can be configured to be one or two bytes wide. The actual width of the port depends on the GTP_DUAL tile's INTDATAWIDTH setting (controls the width of the internal datapath), and whether or not the 8B/10B encoder is enabled. Port widths can be 8 bits, 10 bits, 16 bits, and 20 bits.

The rate of the parallel clock (TXUSRCLK2) at the interface is determined by the TX line rate, the width of the TXDATA port, and whether or not 8B/10B encoding is enabled. A second parallel clock (TXUSRCLK) must be provided for the internal PCS logic in the transmitter. This section shows how to drive the parallel clocks and explains the constraints on those clocks for correct operation.

Ports and Attributes

Table 6-1 defines the FPGA TX interface ports.

Table 6-1: **FPGA TX Interface Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the width of the internal datapath for the entire GTP_DUAL tile. This shared port is also described in Shared PMA PLL, page 72. • 0: Internal datapath is 8 bits wide • 1: Internal datapath is 10 bits wide
REFCLKOUT	Out	N/A	The REFCLKOUT port from each GTP_DUAL tile provides direct access to the reference clock provided to the shared PMA PLL (CLKIN). It can be routed for use in the FPGA logic.
TXDATA0[15:0] TXDATA1[15:0]	In	TXUSRCLK2	The bus for transmitting data. The width of this port depends on TXDATAWIDTH: • TXDATAWIDTH = 0: TXDATA[7:0] = 8 bits wide • TXDATAWIDTH = 1: TXDATA[15:0] = 16 bits wide When a 10-bit or a 20-bit bus is required, the TXCHARDISPVAL and TXCHARDISPMODE ports from the 8B/10B encoder are concatenated with the TXDATA port. See Figure 6-3, page 106.
TXDATAWIDTH0 TXDATAWIDTH1	In	TXUSRCLK2	Selects the width of the TXDATA port. • 0: TXDATA is 8 bits or 10 bits wide • 1: TXDATA is 16 bits or 20 bits wide
TXENC8B10BUSE	In	TXUSRCLK2	TXENC8B10BUSE is set High to enable the 8B/10B encoder. INTDATAWIDTH must also be High. 0: 8B/10B encoder bypassed. This option reduces latency. 1: 8B/10B encoder enabled. INTDATAWIDTH must be High.

Apotex Ex1009
Page 104 of 347

Table 6-1: **FPGA TX Interface Ports *(Continued)***

Port	Dir	Clock Domain	Description
TXOUTCLK0 TXOUTCLK1	Out	N/A	This port provides a parallel clock generated by the GTP transceiver. This clock can be used to drive TXUSRCLK for one or more GTP transceivers. The rate of the clock depends on INTDATAWIDTH: • INTDATAWIDTH is Low: $F_{TXOUTCLK}$ = Line Rate/8 • INTDATAWIDTH is High: $F_{TXOUTCLK}$ = Line Rate/10 ***Note:*** • When INTDATAWIDTH is High, the duty cycle is 60/40 instead of 50/50. • When oversampling is enabled, the line rate in the calculation of $F_{TXOUTCLK}$ is equal to the oversampled line rate, not the PMA line rate.
TXRESET0 TXRESET1	In	Async	Resets the PCS of the GTP transmitter, including the phase adjust FIFO, the 8B/10B encoder, and the FPGA TX interface.
TXUSRCLK0 TXUSRCLK1	In	N/A	Use this port to provide a clock for the internal TX PCS datapath. This clock must always be provided. The rate depends on INTDATAWIDTH: • INTDATAWIDTH is Low: $F_{TXUSRCLK}$ = Line Rate/8 • INTDATAWIDTH is High: $F_{TXUSRCLK}$ = Line Rate/10
TXUSRCLK20 TXUSRCLK21	In	N/A	Use this port to synchronize the FPGA logic with the TX interface. This clock must be positive-edge aligned to TXUSRCLK. The rate of this clock depends on $F_{TXUSRCLK}$ and TXDATAWIDTH: • TXDATAWIDTH = `0`: $F_{TXUSRCLK2} = F_{TXUSRCLK}$ • TXDATAWIDTH = `1`: $F_{TXUSRCLK2} = F_{TXUSRCLK}/2$

There are no attributes in this section.

Description

The FPGA TX interface allows parallel data to be written to the GTP transceiver for transmission as serial data. To use the interface:

- The width of the data interface must be configured
- TXUSRCLK2 and TXUSRCLK must be connected to clocks running at the correct rate

Configuring the Width of the Interface

Table 6-2 shows how the interface width for the TX datapath is selected. 8B/10B encoding is discussed in more detail in Configurable 8B/10B Encoder, page 111.

Table 6-2: **TX Datapath Width Configuration**

INTDATAWIDTH	TXDATAWIDTH	TXENC8B10BUSE	FPGA TX Interface Width
0	0	N/A	8 bits
0	1	N/A	16 bits
1	0	0	10 bits
1	0	1	8 bits
1	1	0	20 bits
1	1	1	16 bits

Figure 6-2 shows how TXDATA is transmitted serially when the internal datapath is 8 bits (INTDATAWIDTH is Low) and 8B/10B encoding is disabled.

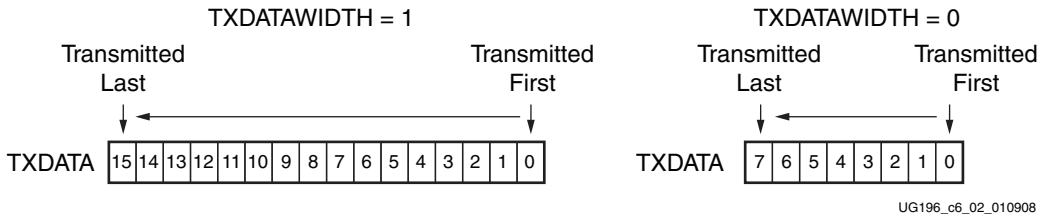


Figure 6-2: **8B/10B Bypassed, 8-Bit Internal Datapath**

Figure 6-3 shows how TXDATA is transmitted serially when the internal datapath is 10 bits (INTDATAWIDTH is High) and 8B/10B encoding is disabled. When TXDATA is 10 bits or 20 bits wide, the TXCHARDISPMODE and TXCHARDISPVAL ports are taken from the 8B/10B encoder interface and used to send the extra bits.

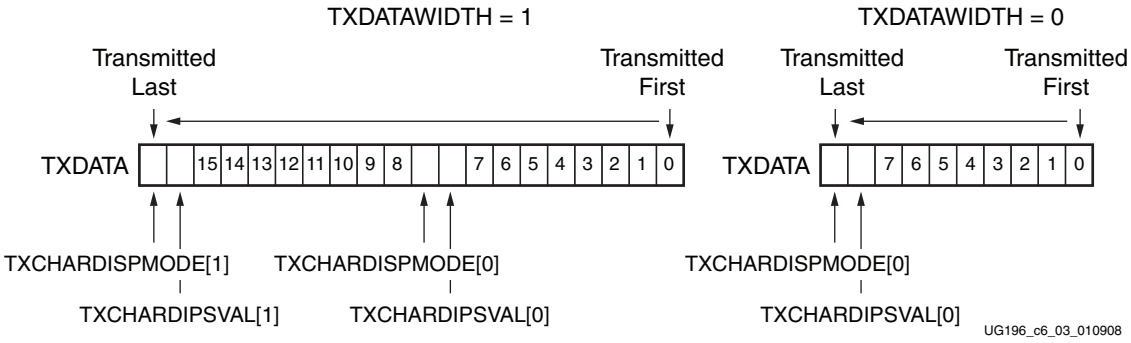


Figure 6-3: **8B/10B Bypassed, 10-Bit Internal Datapath**

When 8B/10B encoding is used, the data interface is a multiple of 8 bits (Figure 6-2), and the data is encoded before it is transmitted serially. Configurable 8B/10B Encoder, page 111 provides more details about bit ordering when using 8B/10B encoding.

Connecting TXUSRCLK and TXUSRCLK2

The FPGA TX interface includes two parallel clocks: TXUSRCLK and TXUSRCLK2. TXUSRCLK is the internal clock for the PCS logic in the GTP transmitter. The required rate for TXUSRCLK depends on the internal datapath width of the GTP_DUAL tile (INTDATAWIDTH) and the TX line rate of the GTP transmitter (Parallel In to Serial Out,

Oplver Ex1009
Page 106 of 347

page 124 describes how the TX line rate is determined). Equation 6-1 shows how to calculate the required rate for TXUSRCLK.

$$\text{TXUSRCLK Rate} = \frac{\text{Line Rate}}{\text{Internal Datapath Width}} \qquad \textit{Equation 6-1}$$

TXUSRCLK2 is the main synchronization clock for all signals into the TX side of the GTP transceiver. Most signals into the TX side of the GTP transceiver are sampled on the positive edge of TXUSRCLK2. TXUSRCLK2 is the same rate at TXUSRCLK when TXDATAWIDTH = `0`, and one half the rate of TXUSRCLK when TXDATAWIDTH = `1`. Equation 6-2 shows how to calculate the required rate for TXUSRCLK2 based on TXDATAWIDTH.

$$\begin{aligned} \text{TXUSRCLK2 Rate} &= \text{TXUSRCLK (TXDATAWIDTH = 0)} \\ \text{TXUSRCLK2 Rate} &= \frac{\text{TXUSRCLK}}{2} \text{ (TXDATAWIDTH = 1)} \end{aligned} \qquad \textit{Equation 6-2}$$

There are some rules about the relationships between clocks that must be observed for TXUSRCLK, TXUSRCLK2, and CLKIN. First, TXUSRCLK and TXUSRCLK2 must be positive edge aligned, with as little skew as possible between them. As a result, low-skew clock resources (BUFGs and BUFRs) should be used to drive TXUSRCLK and TXUSRCLK2. When TXUSRCLK and TXUSRCLK2 have the same frequency, the same clock resource is used to drive both. When the two clocks have different frequencies, TXUSRCLK is divided to get TXUSRCLK2. The designer must ensure that the two are positive edge aligned. The Examples section shows various clock configurations that meet this requirement.

Even though they might run at different frequencies, TXUSRCLK, TXUSRCLK2, and CLKIN must have the same oscillator as their source. Thus TXUSRCLK and TXUSRCLK2 must be multiplied or divided versions of CLKIN. The GTP transceiver provides access to CLKIN in two ways: the REFCLKOUT pin (shared by both GTP transceivers in the GTP_DUAL tile) and the TXOUTCLK pin. The Examples section shows several clock configurations with each pin.

REFCLKOUT is the same as CLKIN. It is free-running, meaning that it operates even before the shared PMA PLL is locked. However, because REFCLKOUT uses the CLKIN rate, it might require multiplication and division to produce the required rates for TXUSRCLK and TXUSRCLK2.

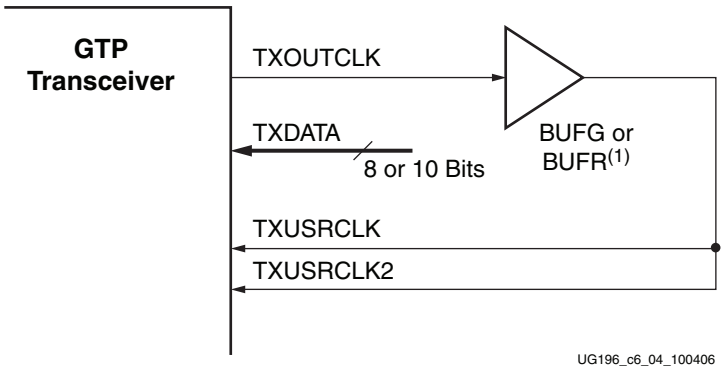
TXOUTCLK provides a copy of CLKIN already divided to the TXUSRCLK rate, potentially requiring fewer dividers. However, TXOUTCLK is not free-running. It is only valid after the shared PMA PLL is locked and cannot be used when TX phase alignment is turned on (see TX Buffering, Phase Alignment, and TX Skew Reduction, page 115).

Examples

Figure 6-4 through Figure 6-8 show different ways FPGA clock resources can be used to drive the parallel clocks for the TX interface.

TXOUTCLK Driving a GTP TX in 1-Byte Mode

In Figure 6-4, TXOUTCLK is used to drive TXUSRCLK and TXUSRCLK2 for 1-byte mode (TXDATAWIDTH = `0`).



Notes:

1. Refer to the *Virtex-5 FPGA Data Sheet* and the *Virtex-5 FPGA Configuration Guide* for the maximum clock frequency and jitter limitations of BUFR.

Figure 6-4: **TXOUTCLK Drives TXUSRCLK and TXUSRCLK2**

TXOUTCLK Driving GTP TX in 2-Byte Mode

The examples in Figure 6-5 and Figure 6-6 use 2-byte datapaths (TXDATAWIDTH = 1). In these cases, TXOUTCLK drives TXUSRCLK, and TXOUTCLK is divided by two using a DCM or PLL to drive TXUSRCLK2.

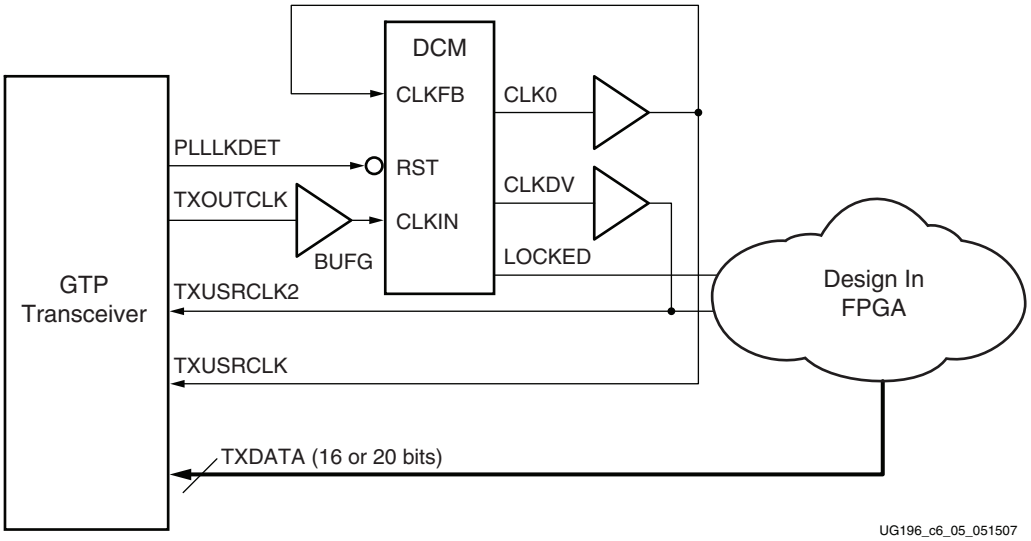


Figure 6-5: **DCM Provides Clocks for 2-Byte Datapath**

Apotex Ex1009
Page 108 of 347

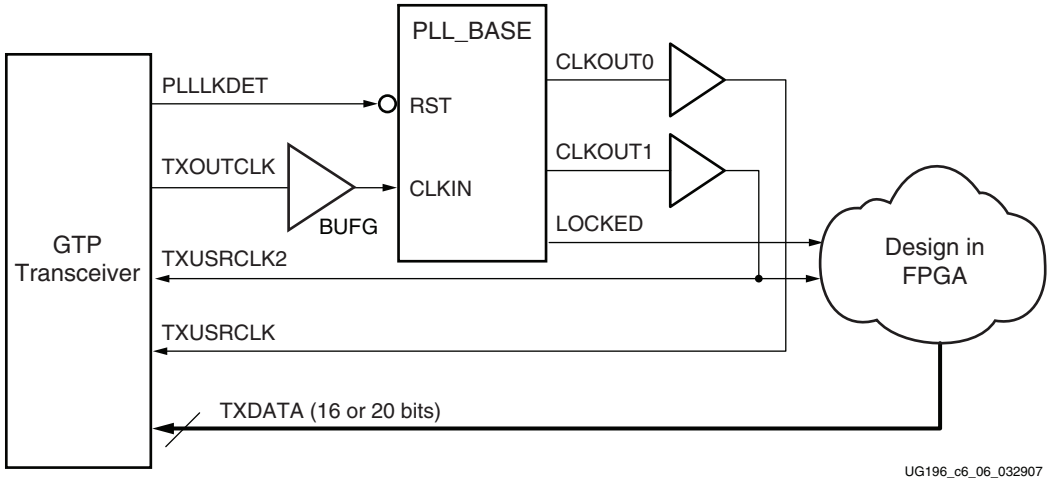


Figure 6-6: **PLL Provides Clocks for a 2-Byte Datapath**

TXOUTCLK Driving Multiple Transceivers for a 2-Byte Datapath

Figure 6-7 shows TXOUTCLK driving multiple GTP user clocks. In this situation, the frequency must be correct for all GTP transceivers, and they must share the same reference clock. In Figure 6-7, because the top GTP transceiver uses a 2-byte interface, it requires a divided clock for TXUSRCLK2.

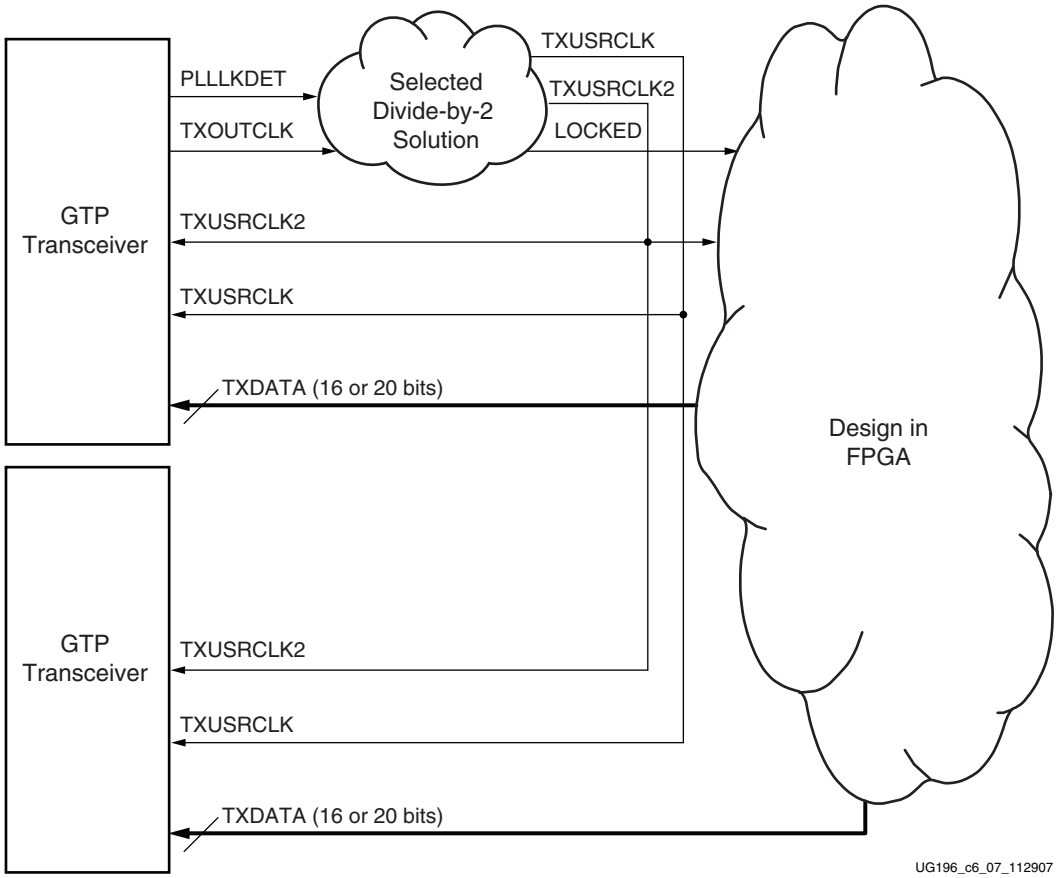


Figure 6-7: **TXOUTCLK Drives Multiple GTP Transceivers with a 2-Byte Interface**

REFCLKOUT Driving Multiple Transceivers with a 2-Byte Interface

Figure 6-8 shows how REFCLKOUT can be used to generate USRCLK signals. REFCLKOUT runs continuously, even when the GTP_DUAL tile is reset. However, extra clocking resources might be needed to generate the correct USRCLK frequency. In Figure 6-8, a PLL is used to generate the TXUSRCLK and TXUSRCLK2 frequencies from REFCLKOUT. A DCM can be used instead of the PLL, but the PLL is more convenient when the REFCLKOUT rate is not an integer multiple of the required TXUSRCLK rates.

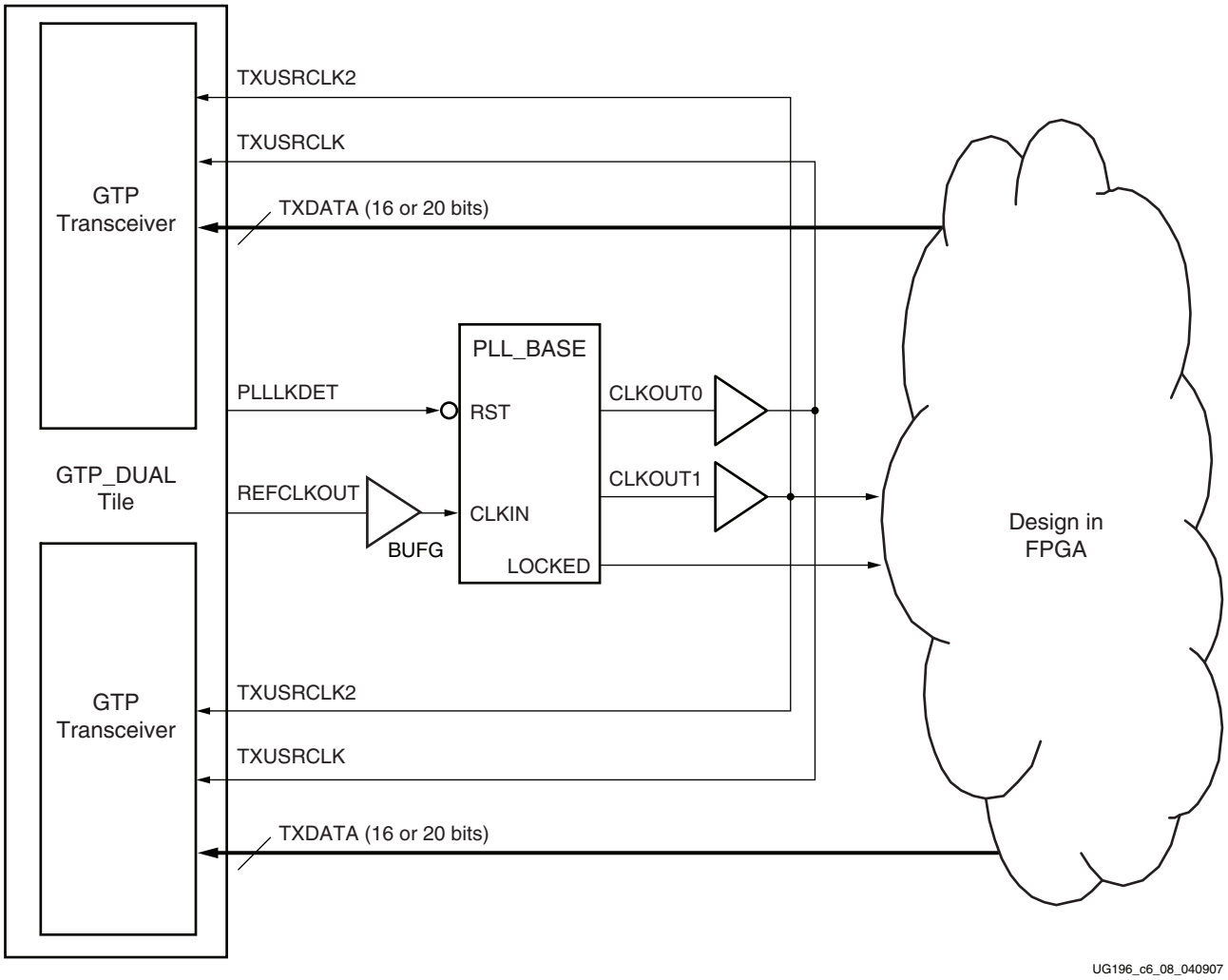


Figure 6-8: **REFCLKOUT Driving Multiple Transceivers with a 2-Byte Interface**

Apotex Ex1009
Page 110 of 347

Configurable 8B/10B Encoder

Overview

Many protocols use 8B/10B encoding on outgoing data. 8B/10B is an industry-standard encoding scheme that trades two bits of overhead per byte for improved performance. Table 6-3 outlines the benefits and costs of 8B/10B. Appendix C shows how 8-bit values are mapped to 10-bit data and control sequences in 8B/10B.

Table 6-3: **8B/10B Trade-Offs**

8B/10B Benefits	8B/10B Costs
• DC Balanced: No increase in bit errors due to line charging on AC-coupled channels.	• Two-bit overhead per byte: Every byte transmitted is mapped to a 10-bit character. As a result, 20% of the channel bandwidth is consumed for overhead.
• Limited Run Lengths: The maximum number of bits without a transition is 5, making it easy for receivers to achieve and maintain lock.	• Both sides of the channel must use 8B/10B. 8B/10B data must be decoded before it can be used.
• Error Detection: All single-bit errors and many multibit errors can be detected using disparity and out-of-table error checking.	
• Control Characters: 8B/10B allows bytes to be marked as control characters. This feature is heavily used in many standard protocols.	

The GTP transceiver includes an 8B/10B encoder to encode TX data without consuming FPGA resources. If encoding is not needed, the block can be disabled to minimize latency.

Ports and Attributes

Table 6-4 defines the TX encoder ports.

Table 6-4: **TX Encoder Ports**

Port	Dir	Clock Domain	Description
TXBYPASS8B10B0[1:0] TXBYPASS8B10B1[1:0]	In	TXUSRCLK2	TXBYPASS8B10B controls the operation of the TX 8B/10B encoder on a per-byte basis. It is only effective when both TXENC8B10B and INTDATAWIDTH are High (8B/10B is enabled). TXBYPASS8B10B[1] corresponds to TXDATA[15:8], and TXBYPASS8B10B[0] corresponds to TXDATA[7:0]. `00`: Both bytes are 8B/10B encoded `01`: Only TXDATA[15:8] is 8B/10B encoded `10`: Only TXDATA[7:0] is 8B/10B encoded `11`: Neither byte is 8B/10B encoded
TXCHARDISPMODE0[1:0] TXCHARDISPMODE1[1:0]	In	TXUSRCLK2	TXCHARDISPMODE and TXCHARDISPVAL allow the 8B/10B disparity of outgoing data to be controlled when 8B/10B encoding is enabled. When 8B/10B encoding is disabled, TXCHARDISPMODE is used to extend the data bus for TX interfaces whose width is a multiple of 10 (see FPGA TX Interface, page 104 for details). TXCHARDISPMODE[1] corresponds to TXDATA[15:8] TXCHARDISPMODE[0] corresponds to TXDATA[7:0] Table 6-5, page 114 shows how TXCHARDISPMODE is used to control the disparity of outgoing data when 8B/10B encoding is enabled.
TXCHARDISPVAL0[1:0] TXCHARDISPVAL1[1:0]	In	TXUSRCLK2	TXCHARDISPVAL and TXCHARDISPMODE allow the disparity of outgoing data to be controlled when 8B/10B encoding is enabled. When 8B/10B encoding is disabled, TXCHARDISPVAL is used to extend the data bus for 10- and 20-bit TX interfaces (see FPGA TX Interface, page 104 for details). TXCHARDISPVAL[1] corresponds to TXDATA[15:8] TXCHARDISPVAL[0] corresponds to TXDATA[7:0] Table 6-5, page 114 shows how TXCHARDISPVAL is used to control the disparity of outgoing data when 8B/10B encoding is enabled.
TXCHARISK0[1:0] TXCHARISK1{1:0]	In	TXUSRCLK2	TXCHARISK is set High to send TXDATA as an 8B/10B K character. TXCHARISK should only be asserted for TXDATA values in the K-character table of the 8B/10B table in Appendix C, 8B/10B Valid Characters. TXCHARISK[1] corresponds to TXDATA[15:8] TXCHARISK[0] corresponds to TXDATA[7:0] TXCHARISK is undefined for bytes that bypass 8B/10B encoding.

Cpuver Ex1009
Page 112 of 347

Table 6-4: **TX Encoder Ports *(Continued)***

Port	Dir	Clock Domain	Description
TXENC8B10BUSE0 TXENC8B10BUSE1	In	TXUSRCLK2	TXENC8B10BUSE is set High to enable the 8B/10B encoder. INTDATAWIDTH must also be High. `0`: 8B/10B encoder bypassed. This option reduces latency. `1`: 8B/10B encoder enabled. INTDATAWIDTH must be `1`.
TXKERR0[1:0] TXKERR1[1:0]	Out	TXUSRCLK2	TXKERR indicates if an invalid code for a K character was specified. For 1-byte interfaces: TXKERR[0] corresponds to TXDATA[7:0] For 2-byte interfaces: TXKERR[0] corresponds to TXDATA[15:8] TXKERR[1] corresponds to TXDATA[7:0]
TXRUNDISP0[1:0] TXRUNDISP1[1:0]	Out	TXUSRCLK2	TXRUNDISP indicates the current running disparity of the 8B/10B encoder. This disparity corresponds to TXDATA clocked in several cycles earlier. For 1-byte interfaces: TXRUNDISP[0] corresponds to previous TXDATA[7:0] data For 2-byte interfaces: TXRUNDISP[0] corresponds to previous TXDATA[15:8] data TXRUNDISP[1] corresponds to previous TXDATA[7:0] data

There are no attributes in this section.

Description

Enabling 8B/10B Encoding

To disable the 8B/10B encoder on a given GTP transceiver, TXENC8B10BUSE must be driven Low. To enable the 8B/10B encoder, TXENC8B10BUSE must be driven High. When the encoder is turned off, the operation of the TXDATA port is as described in FPGA TX Interface, page 104.

8B/10B Bit and Byte Ordering

The order of the bits after the 8B/10B encoder is the opposite of the order shown in Appendix C, 8B/10B Valid Characters, because 8B/10B encoding requires bit a0 to be transmitted first, and the GTP transceiver always transmits the right-most bit first. To match with 8B/10B, the 8B/10B encoder in the GTP transceiver automatically reverses the bit order (Figure 6-9).

For the same reason, when a 2-byte interface is used, the first byte to be transmitted (byte 0) must be placed on TXDATA[7:0], and the second placed on TXDATA[15:8]. This placement ensures that the byte 0 bits are all sent before the byte 1 bits, as required by 8B/10B encoding.

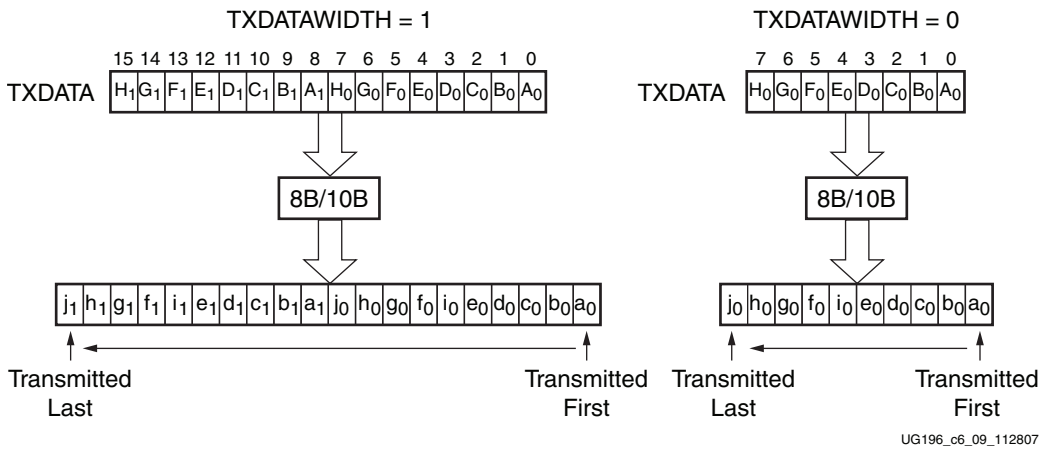


Figure 6-9: **8B/10B Encoding**

K Characters

The 8B/10B table includes special characters (K characters) that are often used for control functions. To transmit TXDATA as a K character instead of regular data, the TXCHARISK port must be driven High. If TXDATA is not a valid K character, the encoder drives TXKERR High.

Running Disparity

8B/10B uses running disparity to balance the number of ones and zeros transmitted. Whenever a character is transmitted, the encoder recalculates the running disparity. The current TX running disparity can be read from the TXCHARDISP port. This running disparity is calculated several cycles after the TXDATA is clocked into the FPGA TX interface, so it cannot be used to decide the next value to send, as required in some protocols.

Normally, running disparity is used to determine whether a positive or negative 10-bit code is transmitted next. The encoder allows the next disparity value to be controlled directly as well, to accommodate protocols that use disparity to send control information. For example, an Idle character sent with reversed disparity might be used to trigger clock correction. Table 6-5 shows how the TXCHARDISPMODE and TXCHARDISPVAL ports are used to control outgoing disparity values.

Table 6-5: **TXCHARDISPMODE and TXCHARDISPVAL vs. Outgoing Disparity**

TXCHARDISPMODE	TXCHARDISPVAL	Outgoing Disparity
0	0	Calculated normally by the 8B/10B encoder
0	1	Inverts normal running disparity when encoding TXDATA
1	0	Forces running disparity negative when encoding TXDATA
1	1	Forces running disparity positive when encoding TXDATA

Apotex Ex1009
Page 114 of 347

8B/10B Bypass

The encoder offers total control of outgoing data using the TXBYPASS8B10B signal. To bypass the 8B/10B encoding and write the outgoing 10-bit code directly, TXBYPASS8B10B must be driven High. When TXBYPASS8B10B is High, the TX interface for the byte is the same as in Figure 6-2, page 106. Bypassing the encoder using TXBYPASS8B10B does not reduce latency, but it does allow each byte of the TX interface to be bypassed individually on a cycle-by-cycle basis.

TX Buffering, Phase Alignment, and TX Skew Reduction

Overview

The GTP TX datapath has two internal parallel clock domains used in the PCS: the PMA parallel clock domain (XCLK), and the TXUSRCLK domain. To transmit data, the XCLK rate must match the TXUSRCLK rate and phase differences between the two domains must be resolved. Figure 6-10 shows the XCLK and USRCLK domains.

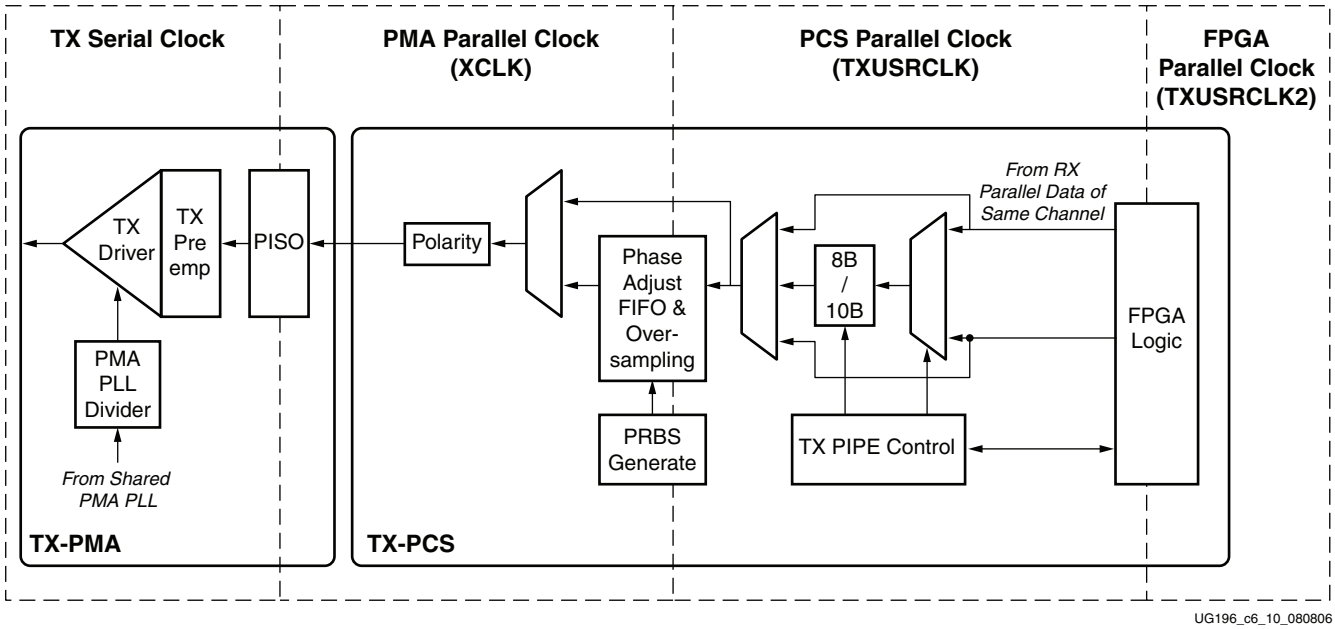


Figure 6-10: **Clock Domains and Alignment Logic**

The GTP transmitter includes a TX buffer and a TX phase-alignment circuit to resolve phase differences between the XCLK and TXUSRCLK domains. All TX datapaths must use one of these circuits. Table 6-6 shows trade-offs between buffering and phase alignment.

Table 6-6: **Buffering and Phase-Alignment Trade-Offs**

	TX Buffer	TX Phase Alignment
Ease of Use	The TX buffer is used when possible. It is robust and easy to operate.	Phase alignment requires extra logic and additional constraints on clock sources. TXOUTCLK cannot be used.
Latency	If low latency is critical, the TX buffer must be bypassed.(1)	Phase alignment uses fewer registers in the datapath.(1, 2)
Skew Reduction	The TX buffer and phase alignment blocks can be used for skew reduction. When the TXUSRCLK frequency is greater than 290 MHz, the output skew can be 10 UI greater than the specification in the DS202 data sheet.	At TXUSRCLK frequencies above 290 MHz, the lowest TX lane-to-lane skew can be achieved by bypassing the TX buffer and performing phase alignment.(1)
Oversampling	The TX buffer is required for oversampling.	

Notes:

1. Bypassing the TX buffer is an advanced feature. TX buffer bypass can operate only under certain system-level conditions and data rates.
2. When the buffer is bypassed, TXUSRCLK must be driven directly from REFCLKOUT using a BUFR regional clock buffer to provide maximum system margin across system conditions.

The TX phase-alignment circuit can also be used to minimize skew between GTP transceivers. Figure 6-11 shows how the phase-alignment circuit can reduce lane skew by aligning the XCLK domains of multiple GTP transceivers to a common clock.

Figure 6-11 shows multiple lanes running before and after phase alignment to a common clock. Before phase alignment, the XCLK of each transmitter has an arbitrary phase. After phase alignment, the XCLK of each transmitter is aligned to the common clock. The only difference in phase is caused by the skew on the common clock global network. After phase alignment to a common clock, all data is transmitted simultaneously as long as the datapath latency is matched.

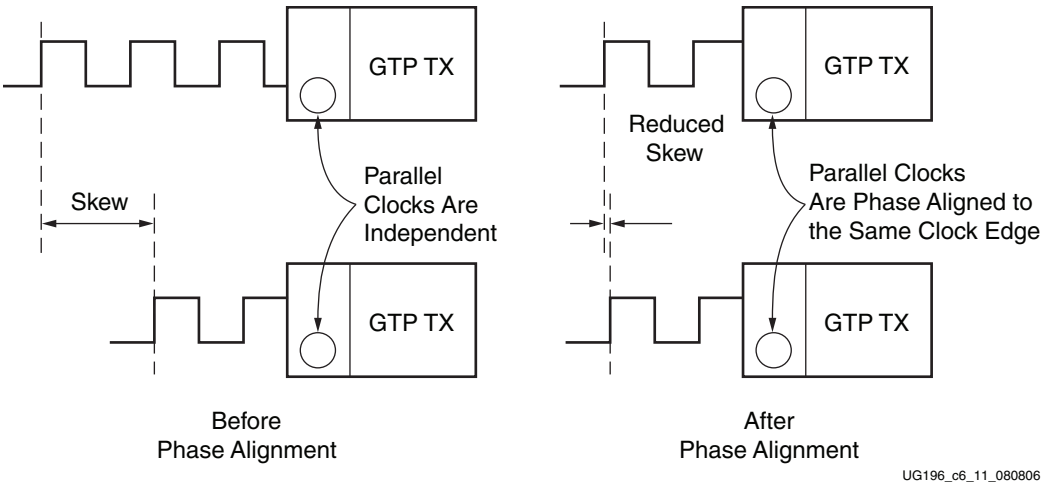


Figure 6-11: **Phase-Alignment Detail**

Apotex Ex1009
Page 116 of 347

When oversampling is enabled (OVERSAMPLE_MODE = TRUE), the TX buffer is used for bit interpolation and must always be active. See Oversampling, page 157 for more information about built-in 5x oversampling.

Table 6-7 defines the signals comprising the TX buffering and phase-alignment ports.

Table 6-7: **TX Buffering and Phase-Alignment Ports**

Port	Dir	Clock Domain	Description
PLLLKDET	Out	Async	This port indicates that the VCO rate is within acceptable tolerances of the desired rate when High. Neither GTP transceiver in the tile operates reliably until this condition is met.
REFCLKOUT	Out	N/A	The REFCLKOUT port from each GTP_DUAL tile provides direct access to the reference clock provided to the shared PMA PLL (CLKIN). It can be routed for use in the FPGA logic.
TXBUFSTATUS0[1:0] TXBUFSTATUS1[1:0]	Out	TXUSRCLK2	TX buffer status. TXBUFSTATUS[1]: TX buffer overflow or underflow `1`: FIFO has overflowed or underflowed `0`: No overflow/underflow error TXBUFSTATUS[0]: TX buffer fullness `1`: FIFO is at least half full `0`: FIFO is less than half full If TXBUFSTATUS[1] goes High, it remains High until TXRESET is asserted.
TXENPMAPHASEALIGN	In	Async	When activated, both GTP transmitters in a GTP_DUAL tile can align their XCLKs with their TXUSRCLKs, allowing their TX buffers to be bypassed. This also allows the XCLKs in multiple GTP transmitters to be synchronized to reduce TX skew between them.
TXOUTCLK0 TXOUTCLK1	Out	N/A	This port provides a parallel clock generated by the GTP transceiver. This clock can be used to drive TXUSRCLK for one or more GTP transceivers. The clock rate depends on INTDATAWIDTH: • INTDATAWIDTH is Low: $F_{TXOUTCLK}$ = Line Rate/8 • INTDATAWIDTH is High: $F_{TXOUTCLK}$ = Line Rate/10 ***Note:*** • When INTDATAWIDTH is High, the duty cycle is 60/40 instead of 50/50. TXOUTCLK cannot drive TXUSRCLK when the TX phase-alignment circuit is used. • When oversampling is enabled, the line rate in the calculation of $F_{TXOUTCLK}$ is equal to the oversampled line rate, not the PMA line rate.
TXPMASETPHASE	In	Async	When activated, TXPMASETPHASE aligns XCLK with TXUSRCLK for both GTP transmitters in the GTP_DUAL tile.

Table 6-7: **TX Buffering and Phase-Alignment Ports *(Continued)***

Port	Dir	Clock Domain	Description
TXUSRCLK0 TXUSRCLK1	In	N/A	Use this port to provide a clock for the internal TX PCS datapath. This clock must always be provided. Its rate depends on INTDATAWIDTH: • INTDATAWIDTH is Low: $F_{TXUSRCLK}$ = Line Rate/8 • INTDATAWIDTH is High: $F_{TXUSRCLK}$ = Line Rate/10

Table 6-8 defines the TX buffering and phase-alignment attributes.

Table 6-8: **TX Buffering and Phase-Alignment Attributes**

Attribute	Description
OVERSAMPLE_MODE	This shared attribute activates the built-in 5x digital oversampling circuits in both GTP_DUAL transceivers. Oversampling must be enabled when running the GTP transceivers at line rates between 100 Mb/s and 500 Mb/s. TRUE: Built-in 5x digital oversampling enabled for both GTP transceivers on the tile FALSE: Digital oversampling disabled See Oversampling, page 157 for more details about 5x digital oversampling.
PLL_TXDIVSEL_COMM_OUT	Divides the PLL clock to produce a high-speed TX clock. Because both edges of the clock are used, the divided clock must run at one-half the desired TX line rate. The available divider settings are 1, 2, and 4. This divider is used when the same divider value is needed for both GTP transceivers and to provide a clock to both GTP transceivers. When PLL_TXDIVSEL_COMM_OUT is used, both PLL_TXDIVSEL_OUT attributes must be set to `1`. See Parallel In to Serial Out, page 124.
PLL_TXDIVSEL_OUT_0 PLL_TXDIVSEL_OUT_1	Divides the PLL clock to produce a high-speed TX clock. Because both edges of the clock are used, the divided clock must run at one-half the desired TX line rate. The available divider settings are 1, 2, and 4. Each GTP transceiver has a separate PLL_TXDIVSEL_OUT. When transceivers require different dividers, these attributes must be used instead of PLL_TXDIVSEL_COMM_OUT, and PLL_TXDIVSEL_COMM_OUT must be set to `1`. See Parallel In to Serial Out, page 124. When performing phase alignment to minimize TX skew or to bypass the TX buffer, PLL_TXDIVSEL_OUT_0 and PLL_TXDIVSEL_OUT_1 must be set to divide by 1. See Bypassing the TX Buffer, page 121 for details.
PMA_COM_CFG	Common PMA configuration attribute. This attribute is left at its default value. The value is automatically set by the RocketIO GTP Transceiver Wizard.
TX_BUFFER_USE_0 TX_BUFFER_USE_1	Determines whether or not the TX buffer is used. TRUE: Use the TX buffer. FALSE: Bypass the TX buffer. The phase-alignment circuit must be used when TX_BUFFER_USE is FALSE.

Apotex Ex1009
Page 118 of 347

Table 6-8: **TX Buffering and Phase-Alignment Attributes *(Continued)***

Attribute	Description
TX_XCLK_SEL0 TX_XCLK_SEL1	Selects the clock used to drive the clock domain in the PCS following the TX buffer. The attribute must be set as follows: TXOUT: Use when TX_BUFFER_USE = TRUE TXUSR: Use when TX_BUFFER_USE = FALSE
TXRX_INVERT_0 TXRX_INVERT_1	Controls inverters that optimize the clock paths within the GTP transceiver for different modes of operation. The attribute must be set as follows: `00000`: Use when TX_BUFFER_USE = TRUE `00100`: Use when TX_BUFFER_USE = FALSE

Description

Using the TX Buffer

To use the TX buffer to resolve phase differences between the domains, TX_BUFFER_USE must be set to TRUE. The buffer should be reset whenever TXBUFSTATUS indicates an overflow or an underflow. The buffer can be reset using GTPRESET (see Reset, page 84) or TXRESET (see FPGA TX Interface, page 104). Assertion of GTPRESET triggers a sequence that resets the entire GTP_DUAL tile.

Phase-Alignment Procedure

The phase-alignment circuit is used to:

- Minimize TX skew
- Bypass the TX buffer for deterministic or low-latency applications

When performing phase alignment to minimize TX skew or to bypass the TX buffer, PLL_TXDIVSEL_OUT_0 and PLL_TXDIVSEL_OUT_1 must be set to 1. Any divide by 2 or divide by 4 factor must be specified using PLL_TXDIVSEL_COMM_OUT, which is common to both channels. When PLL_TXDIVSEL_OUT_0 and PLL_TXDIVSEL_OUT_1 are set to divide by 1, the GTP0 transmitter and GTP1 transmitter of a given GTP_DUAL tile operate at the same line rate.

The phase alignment procedure must be performed when any of the following conditions occur:

- The FPGA is configured
- GTPRESET is asserted
- PLLPOWERDWNB is deasserted
- The clocking source is changed

The phase-alignment procedure shown in Figure 6-12 is outlined in the steps below. TXENPMAPHASEALIGN and TXPMASETPHASE are shared tile pins (see Shared PMA PLL, page 72), so the procedure is always applied to both GTP transceivers on the tile. TXOUTCLK cannot be the source for TXUSRCLK when the TX phase-alignment circuit is used. SeeFPGA TX Interface, page 104 for details.

1. Wait for all clocks to stabilize. The shared PMA clock and TXUSRCLK must be stable before phase-alignment is attempted. If REFCLKOUT drives TXUSRCLK directly, wait for PLLLKDET to be asserted before moving on to step 2. If TXUSRCLK comes from a

Optiver Ex1009
Page 119 of 347

DCM or PLL, wait for PLLLKDET and the LOCKED signal from the DCM or PLL before moving on to step 2.

2. Drive TXENPMAPHASEALIGN High. Keep TXENPMAPHASEALIGN High unless the phase-alignment procedure must be repeated. Driving TXENPMAPHASEALIGN Low causes phase-alignment to be lost.
3. Wait 512 TXUSRCLK2 clock cycles and drive TXPMASETPHASE High.
4. Wait the number of required TXUSRCLK2 clock cycles as specified in Table 6-9, and drive TXPMASETPHASE Low. The phase of the XCLK is now aligned with TXUSRCLK.

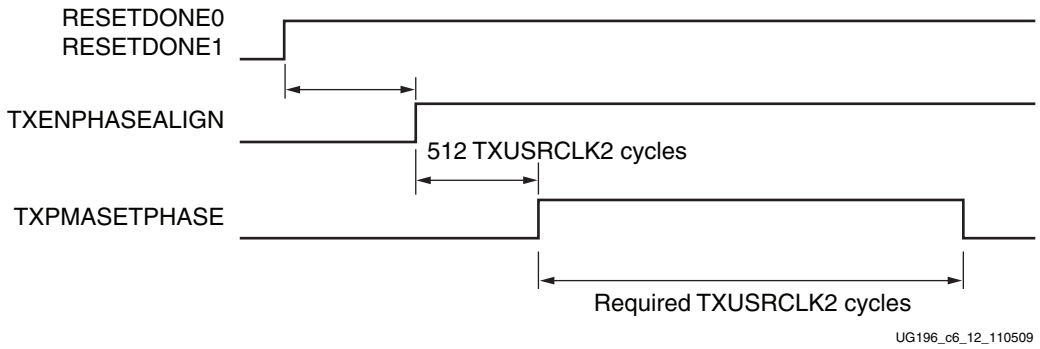


Figure 6-12: **TX Phase-Alignment Procedure**

It is critical that the PMA parallel clock (XCLK) and TXUSRCLK both be stable before phase alignment is attempted:

- If REFCLKOUT drives TXUSRCLK directly, wait for PLLLKDET to be asserted before phase aligning.
- If TXUSRCLK comes from a DCM or PLL, wait for PLLLKDET and the LOCKED signal from the DCM or PLL before phase aligning.

Minimizing TX Skew

In this model, The TX buffer is enabled and the TX phase-alignment circuit is used to minimize TX skew across multiple transceivers. At TXUSRCLK frequencies above 290 MHz, the TX lane-to-lane skew may be 10 UI greater than the data sheet specification (see DS202).Table 6-14 shows the TXUSRCLK generation for this use model.

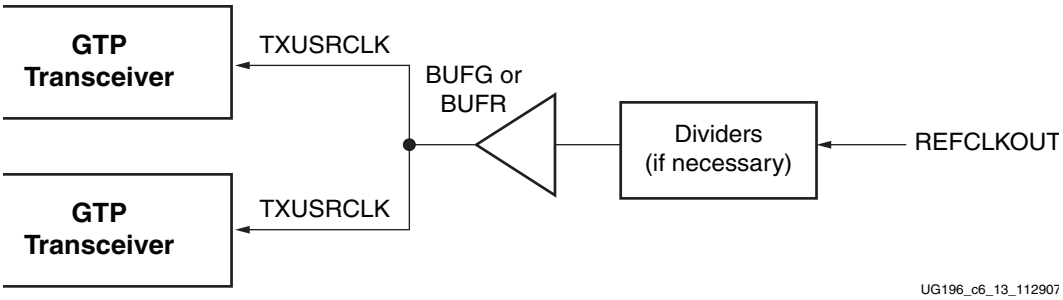


Figure 6-13: **TX Low-Skew Phase-Alignment Configuration**

The required steps for this use model are:

1. Set attributes as recommended when using the TX buffer:

Apotex Ex1009
Page 120 of 347

a. TX_BUFFER_USE = TRUE
b. TX_XCLK_SEL = TXOUT
c. TXRX_INVERT = 00000

2. Perform the Phase-Alignment Procedure, page 119.
3. Toggle TXRESET.

Table 6-9 shows the number of required TXUSRCLK2 clock cycles for this use model.

Table 6-9: **Number of Required TXUSRCLK2 Clock Cycles**

PLL_TXDIVSEL_COMM_OUT	TXUSRCLK2 Wait Cycles
1	16,384
2	32,768
4	65,536

Bypassing the TX Buffer

In this use model, the TX buffer is bypassed and the TX phase-alignment circuit is used to align the PMA and PCS parallel clocks and to provide deterministic and low latency.

When the TX buffer is bypassed, TXUSRCLK is driven directly from REFCLKOUT using a BUFR regional clock buffer to provide maximum system margin across system conditions, the same as shown in Figure 6-13.

The required steps for this use model are shown below.

1. Set attributes to bypass the TX Buffer:
   a. TX_BUFFER_USE = FALSE
   b. TX_XCLK_SEL = TXUSR
   c. TXRX_INVERT = 00100
2. Perform the phase alignment procedure as outlined in Phase-Alignment Procedure, page 119.

Table 6-10 shows the number of required TXUSRCLK2 clock cycles for this use model.

Table 6-10: **Number of Required TXUSRCLK2 Clock Cycles**

PLL_TXDIVSEL_COMM_OUT	TXUSRCLK2 Wait Cycles
1	4096
2	8192
4	16384

Notes:

1. When bypassing the TX buffer, PLL_TXDIVSEL_OUT_0 and PLL_TXDIVSEL_OUT_1 must be set to divide by 1. Any divide by 2 or divide by 4 factor must be specified using PLL_TXDIVSEL_COMM_OUT, which is common to both channels. When PLL_TXDIVSEL_OUT_0 and PLL_TXDIVSEL_OUT_1 are set to divide by 1, the GTP0 transmitter and GTP1 transmitter of a given GTP_DUAL tile operate at the same line rate.

TX Polarity Control

Overview

The GTP transceiver includes a TX polarity control function to invert outgoing data from the PCS before serialization and transmission. The TXPOLARITY port is driven High to invert the polarity of outgoing data.

Ports and Attributes

Table 6-11 defines the TX polarity control ports.

Table 6-11: **TX Polarity Control Ports**

Port	Dir	Clock Domain	Description
TXPOLARITY0 TXPOLARITY1	In	TXUSRCLK2	Specifies whether the final transmitter output is inverted or not. `0`: Not inverted. TXP is positive, and TXN is negative. `1`: Inverted: TXP is negative, and TXN is positive.

There are no attributes in this section.

Description

The GTP transceiver can invert the polarity of its TX data before it is transmitted. This feature can be used to avoid hardware fixes for swapped TXP/TXN differential traces on a board.

Cypher Ex1009
Page 122 of 347

TX PRBS Generator

Overview

Pseudo-random bit sequences (PRBS) are commonly used to test the signal integrity of high-speed links. These sequences appear random but have specific properties that can be used to measure the quality of a link.

The GTP PRBS block can generate several industry-standard PRBS patterns. Table 6-12 lists the available PRBS patterns and their typical uses.

Table 6-12: **Pseudo-Random Bit Sequences**

Name	Polynomial	Length of Sequence (bits)	Consecutive Zeros	Typical Use
PRBS-7	$1 + X^6 + X^7$ (inverted)	$2^7 - 1$	7	Used to test channels with 8B/10B.
PRBS-23	$1 + X^{18} + X^{23}$ (inverted)	$2^{23} - 1$	23	ITU-T Recommendation O.150, Section 5.6. One of the recommended test patterns in the SONET specification.
PRBS-31	$1 + X^{28} + X^{31}$ (inverted)	$2^{31} - 1$	31	ITU-T Recommendation O.150, Section 5.8. A recommended PRBS test pattern for 10 Gigabit Ethernet. See IEEE 802.3ae-2002.

Ports and Attributes

Table 6-13 defines the TX PRBS generator ports.

Table 6-13: **TX PRBS Generator Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the width of the internal datapath for the entire GTP_DUAL tile. The PRBS generator only works when INTDATAWIDTH is High (10-bit internal datapath).
TXENPRBSTST0[1:0] TXENPRBSTST1[1:0]	In	TXUSRCLK2	Transmitter test pattern generation control. A PRBS is generated by enabling the test pattern generation circuit. INTDATAWIDTH must also be High (10-bit internal data width mode) when the PRBS generator is enabled. `00`: Test pattern generation off (standard operation mode) `01`: Enable $2^7 - 1$ PRBS generation `10`: Enable $2^{23} - 1$ PRBS generation `11`: Enable $2^{31} - 1$ PRBS generation Because PRBS patterns are deterministic, the receiver can check the received data against a sequence of its own PRBS generator.

There are no attributes in this section.

Optiver Ex1009
Page 123 of 347

Description

Each GTP transceiver includes a built-in PRBS generator. This feature can be used in conjunction with other test features, such as loopback and the built-in PRBS checker, to run tests on a given channel. Only the 10-bit internal data width mode is supported, which implies that INTDATAWIDTH must also be driven High when the PRBS generator is enabled.

To use the PRBS generator, INTDATAWIDTH should be driven High to set the internal datapath width to 10 bits. The PRBS test mode is selected using the TXENPRBSTST port. Table 6-13 lists the available settings.

Parallel In to Serial Out

Overview

The Parallel In to Serial Out (PISO) block is the core of the GTP TX datapath. It serializes parallel data from the PCS using a high-speed clock from the shared PMA PLL.

The PISO block serializes 8 or 10 bits per parallel clock cycle, depending on the internal data width for the tile (INTDATAWIDTH). The clock rate is determined by the shared PMA PLL rate, divided by a local TX divider.

Ports and Attributes

Table 6-14 defines the TX PISO ports.

Table 6-14: **TX PISO Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the width of the internal datapath for the entire GTP_DUAL tile. This shared port is also described in Shared PMA PLL, page 72. `0`: Internal datapath is 8 bits wide `1`: Internal datapath is 10 bits wide

Cpuver Ex1009
Page 124 of 347

Table 6-15 defines the TX PISO attributes.

Table 6-15: **TX PISO Attributes**

Attribute	Description
OVERSAMPLE_MODE	This shared attribute activates the built-in 5x digital oversampling circuits in both GTP transceivers. Oversampling must be enabled when running the GTP transceivers at line rates between 100 Mb/s and 500 Mb/s. TRUE: Built-in 5x digital oversampling enabled for both GTP transceivers on the tile FALSE: Digital oversampling disabled See Oversampling, page 157 for more details about 5x digital oversampling.
PLL_TXDIVSEL_OUT_0 PLL_TXDIVSEL_OUT_1	Sets the divider for the TX line rate for the individual GTP transceiver. If PLL_TXDIVSEL_COMM_OUT is greater than 1, PLL_TXDIVSEL_OUT must be set to 1, and the TX line rate is set by PLL_TXDIVSEL_COMM_OUT. The divider can be set to 1, 2, or 4.
PLL_TXDIVSEL_COMM_OUT	Sets a common line rate divider for both GTP transceivers in a tile. The common divider should be used whenever both GTP transceivers in a tile use the same line rate because it reduces skew between lanes. PLL_TXDIVSEL_OUT must always be set to 1 when PLL_TXDIVSEL_COMM_OUT is greater than 1. The divider can be set to 1, 2, or 4.

Description

Equation 6-3 shows how to calculate the TX line rate when operating without oversampling (OVERSAMPLE_MODE = FALSE). If PLL_TXDIVSEL_COMM_OUT is greater than 1, PLL_TXDIVSEL_OUT must be set to 1, and vice versa. PLL_TXDIVSEL_COMM_OUT should be used when both GTP transceivers in the GTP_DUAL tile use the same TX line rate.

$$\text{TX Line Rate} = \frac{\text{PLL Clock Rate} \times 2}{\text{PLL\_TXDIVSEL\_OUT} \times \text{PLL\_TXDIVSEL\_COMM\_OUT}} \quad \textit{Equation 6-3}$$

When oversampling is activated, use Equation 6-4 to calculate the line rate.

$$\text{TX Line Rate} = \frac{\text{PLL Clock Rate} \times 2}{\text{PLL\_TXDIVSEL\_OUT} \times \text{PLL\_TXDIVSEL\_COMM\_OUT} \times 5} \quad \textit{Equation 6-4}$$

See Oversampling, page 157 for more information about oversampling.

Configurable TX Driver

Overview

The GTP TX driver is a high-speed current-mode differential output buffer. To maximize signal integrity, it includes these features:

- Differential voltage control
- Pre-emphasis
- Configurable termination impedance

The impact of each of these features on signal integrity depends on the board and the receiver. See Chapter 10, GTP-to-Board Interface, for a detailed discussion of how to use them to maximize signal integrity. Figure 6-14 shows the different segments of the TX driver.

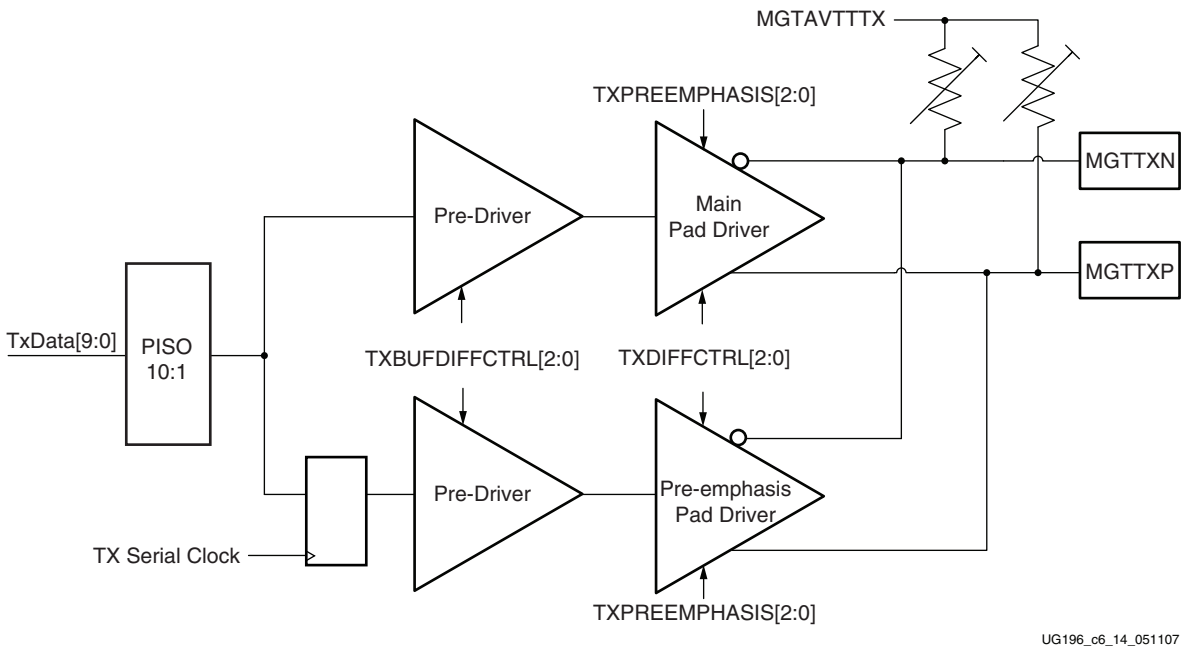


Figure 6-14: **TX Driver Segments**

Oplover Ex1009
Page 126 of 347

Ports and Attributes

Table 6-16 defines the TX driver ports.

Table 6-16: **TX Driver Ports**

Ports	Dir	Clock Domain	Description
MGTTXN0 MGTTXN1 MGTTXP0 MGTTXP1	Out (Pad)	TX Serial Clock	TXN and TXP are the differential complements of each other forming a differential transmitter output pair. These ports represent pads, so their locations must be constrained (see Chapter 4, Implementation), and they must be brought to the top level of the design.
TXBUFDIFFCTRL0[2:0] TXBUFDIFFCTRL1[2:0]	In	Async	Controls the strength of the pre-drivers. This port should be tied to the same value as the TXDIFFCTRL port for the transceiver.
TXDIFFCTRL0[2:0] TXDIFFCTRL1[2:0]	In	Async	Controls the transmitter differential output swing. Table 6-19 shows the approximate differential voltage swing that results from each setting.
TXPREEMPHASIS0[2:0] TXPREEMPHASIS1[2:0]	In	Async	Controls the relative strength of the main drive and pre-emphasis as shown in Table 6-19.

Table 6-17 defines the TX driver attributes.

Table 6-17: **TX Driver Attributes**

Attribute	Description
TX_DIFF_BOOST0 TX_DIFF_BOOST1	Changes the strength of the TX driver and pre-emphasis buffers. When set to TRUE, the pre-emphasis percentage is *boosted* or increased. See Table 6-19 for nominal differential swing and pre-emphasis values. Overall differential swing of the non-transition bits is reduced when TX_DIFF_BOOST is TRUE. The default setting is TX_DIFF_BOOST equals FALSE.

Description

Differential Voltage Control

The amplitude of the TX driver's differential swing can be controlled using the TXDIFFCTRL and TXBUFDIFFCTRL ports. As shown in Figure 6-14, TXDIFFCTRL controls the drive strength of the main pad driver and the pre-emphasis pad driver. Table 6-18 shows the differential output voltage that results from each of the possible settings for the port. Table 6-19 shows the expected transmitter pre-emphasis a percentage of overall TX swing.

Optiver Ex1009
Page 127 of 347

Table 6-18: **Transmitter Output Swing**

Port	Value	Transmitter Differential Swing (mV)
TXDIFFCTRL0[2:0] = TXBUFDIFFCTRL0[2:0] TXDIFFCTRL1[2:0] = TXBUFDIFFCTRL1[2:0]	000	1100
	001	1050
	010	1000
	011	900
	100	800
	101	600
	110	400
	111	0

Table 6-19: **Transmitter Pre-emphasis Settings**

Port	Value	Transmitter Pre-emphasis (%) (1,2)	
		Pre-emphasis Boost Off TX_DIFF_BOOST = FALSE (Default Setting)	Pre-emphasis Boost On TX_DIFF_BOOST = TRUE
TXPREEMPHASIS0[2:0] TXPREEMPHASIS1[2:0]	000	2	3
	001	2	3
	010	2.5	4
	011	4.5	10.5
	100	9.5	18.5
	101	16	28
	110	23	39
	111	31	52

Notes:

1. Nominal values. Refer to the *Virtex-5 FPGA Data Sheet* for the exact values.
2. Refer to *Handbook of Digital Techniques for High-Speed Design* [Ref 3] and *High-Speed Signal Propagation: Advanced Black Magic* [Ref 5] for detailed explanations of pre-emphasis.

Each of the pad drivers is driven by a pre-driver whose output amplitude is controlled by TXBUFDIFFCTRL. Although the pre-driver output swing does not change the output swing at the TX pad, the inputs of TXBUFDIFFCTRL and TXDIFFCTRL must always be tied together.

If TX_DIFF_BOOST is set to TRUE, the percentage of pre-emphasis/de-emphasis is increased. The peak-to-peak amplitude for the transition bits is not changed by the TX_DIFF_BOOST attribute. An increased level of de-emphasis decreases the swing/amplitude of the non-transition bits. A transition bit occurs when there is a change from 0 to 1 or 1 to 0. A non-transition bit occurs when the preceding bit of the current bit has the same value.

Pre-emphasis

Serial traces tend to attenuate high frequencies more than low frequencies. Pre-emphasis is a technique used to pre-equalize transmitted data. It compensates for the excess high-

Apuver Ex1009
Page 128 of 347

frequency loss by transmitting high-frequency signals with more power than low-frequency signals.

The pre-emphasis technique used by the GTP transceiver is de-emphasis. When a serial bit is repeated, the power of the repeated bit is reduced. For example, if two ones are sent in a row, the power of the second one is reduced compared to the first. The power of slowly changing signals (the low-frequency components) is reduced relative to quickly changing signals (the high-frequency components). Refer to *Handbook of Digital Techniques for High-Speed Design*, pages 465-471 [Ref 3] and *High-Speed Signal Propagation: Advanced Black Magic*, pages 211-215 [Ref 5] for more detailed explanations of pre-emphasis and de-emphasis.

Each GTP transceiver has a TXPREEMPHASIS port for controlling pre-emphasis. Table 6-19 shows the percentage decrease of signal amplitude for de-emphasized bits at each TXPREEMPHASIS level. The higher the percentage, the more de-emphasis is applied.

Be careful not to overcompensate for high-frequency losses. Using too much pre-emphasis can cause signal distortion, and combining pre-emphasis with RX equalization (see RX Termination and Equalization, page 139) can overemphasize the high-frequency signal in the channel.

Configurable Termination Impedance

The termination impedance of each transmitter must be matched as closely as possible to the impedance of the serial trace it is driving to minimize distortion due to reflections. See Chapter 10, GTP-to-Board Interface, for more information about how the termination impedance is calibrated.

TXINHIBIT

The TX driver includes the TXINHIBIT port. When TXINHIBIT is driven High, the TX differential pin pair is forced to a constant value (TXP Low, TXN High).

Receive Detect Support for PCI Express Operation

Overview

The GTP transceiver supports receiver detect functionality as defined by the *PHY Interface for PCI Express (PIPE) Specification*. This function drives the TX link to one state followed by a second state and then measures the time required for the link voltage to change. The PIPE specification provides details about the receiver detection mechanism.

Ports and Attributes

Table 6-20 defines the receive detect support ports.

Table 6-20: **Receive Detect Support Ports**

Port	Dir	Domain	Description
PHYSTATUS0 PHYSTATUS1	Out	Async	This signal is asserted High to indicate completion of several PHY functions, including power management state transitions and receiver detection. When this signal transitions during entry and exit from P2 and RXUSRCLK2 is not running, the signaling is asynchronous.
RXSTATUS0[2:0] RXSTATUS1[2:0]	Out	RXUSRCLK2	The decoding of RXSTATUS[2:0] depends on the setting of RX_STATUS_FMT. When RX_STATUS_FMT = PCIE: `000`: Receiver Not Present (when in receiver detection sequence)/Received Data OK (during normal operation). `001`: Reserved. `010`: Reserved. `011`: Receiver Present (when in receiver detection sequence). `100`: 8B/10B Decode Error. `101`: Elastic Buffer Overflow. Different than defined in the PIPE specification. `110`: Elastic Buffer Underflow. Different than defined in the PIPE specification. `111`: Receive Disparity Error. When RX_STATUS_FMT = SATA: RXSTATUS[0]: TXCOMSTART operation complete RXSTATUS[1]: COMWAKE signal received RXSTATUS[2]: COMRESET/COMINIT signal received

Cpuver Ex1009
Page 130 of 347

Table 6-20: **Receive Detect Support Ports *(Continued)***

Port	Dir	Domain	Description
TXDETECTRX0 TXDETECTRX1	In	TXUSRCLK2	Activates the receive detection sequence. The sequence ends when PHYSTATUS is asserted to indicate that the results of the test are ready on RXSTATUS.
TXPOWERDOWN0[1:0] TXPOWERDOWN1[1:0] RXPOWERDOWN0[1:0] RXPOWERDOWN1[1:0]	In	Async	Controls the power state of the TX and RX links. The encoding complies with the PCI Express encoding. TX and RX can be powered down separately. However, for PCI Express compliance, TXPOWERDOWN and RXPOWERDOWN must be used together. `00`: P0 (normal operation) `01`: P0s (low recovery time power down) `10`: P1 (longer recovery time/Receiver Detection still on) `11`: P2 (lowest power state)

There are no attributes in this section.

Description

The PCI Express specification includes a feature that allows the transmitter on a given link to detect if a receiver is present. The decision if a receiver is present is based on the rise time of TXP/TXN. Figure 6-15 shows the circuit model used for receive detection. The GTP transceiver must be in the P1 power-down state to perform receiver detection. Also receiver detection requires a 75 to 200 nF external coupling capacitor between the transmitter and receiver, and the receiver must be terminated to GND.

The detection sequence starts with the assertion of TXDETECTRX. In response, the receive detect logic drives TXN and TXP to $V_{DD} - V_{SWING}/2$ and then releases them. After a programmable interval, the levels of TXN and TXP are compared with a threshold voltage. At the end of the sequence, RXSTATUS and PHYSTATUS reflect the results of the receiver detection.

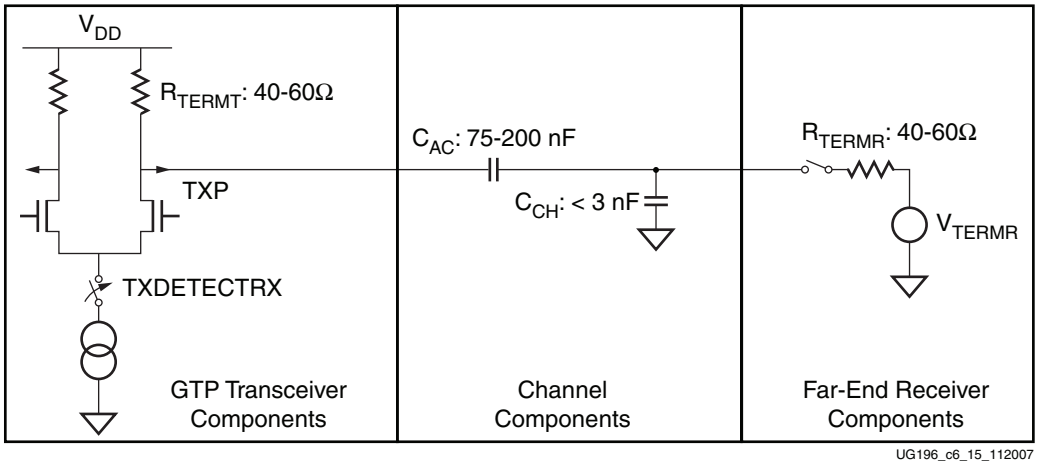


Figure 6-15: **Receive Detection Circuit Model**

Figure 6-16 illustrates the rise times associated with the receiver present and receiver absent conditions, along with the detection threshold.

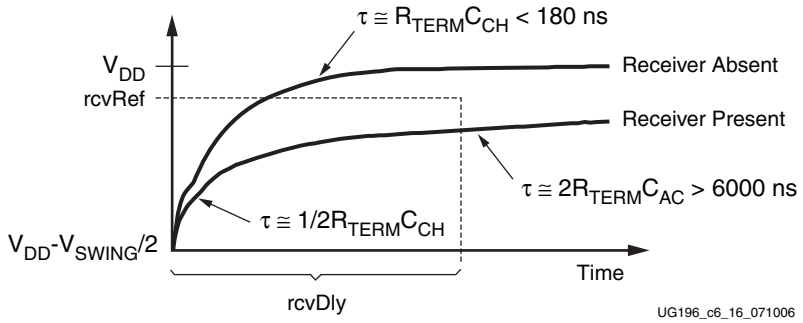


Figure 6-16: **Receive Detection Thresholds**

When the PHY has completed the receiver detect sequence, it asserts PHYSTATUS for one clock and drives the RXSTATUS signals to the appropriate code. After the receiver detection is completed (as signaled by the assertion of PHYSTATUS), TXDETECTRX must be deasserted. Figure 6-17 shows this process.

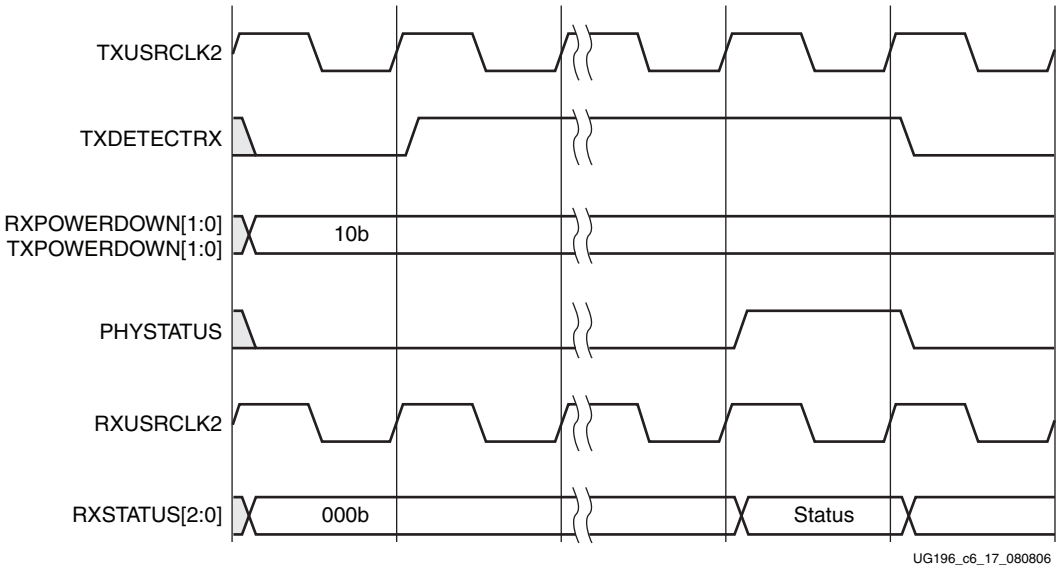


Figure 6-17: **Receiver Detect Waveforms**

Optiver Ex1009
Page 132 of 347

TX OOB/Beacon Signaling

Overview

Each GTP transceiver provides support for generating the Out-of-Band (OOB) sequences described in the Serial ATA (SATA) specification and beaconing for the PCI Express specification. See Appendix B, OOB/Beacon Signaling, for an overview of OOB signaling and how it is used in these protocols.

GTP_DUAL support for SATA OOB signaling consists of the analog circuitry required to encode the OOB signal state and state machines to format bursts of OOB signals for SATA COM sequences (COMRESET, COMWAKE, and COMINIT). Each GTP transceiver also supports SATA auto-negotiation by allowing the timing of the COM sequences to be changed based on the divider settings used for the TX line rate.

GTP_DUAL support for PCI Express OOB signaling-based beacons consists of support for generating OOB states using interface signals defined in the *PHY Interface for the PCI Express (PIPE) Specification*. The format of the beacon sequence is controlled by the FPGA logic.

Ports and Attributes

Table 6-21 describes the ports that control OOB/beacon signaling.

Table 6-21: **TX OOB/Beacon Signaling Ports**

Port	Dir	Domain	Description
RXSTATUS0[2:0] RXSTATUS1[2:0]	Out	RXUSRCLK2	The decoding of RXSTATUS[2:0] depends on the setting of RX_STATUS_FMT: • When RX_STATUS_FMT = PCIE: RXSTATUS is not used for PCIe TXELECIDLE • When RX_STATUS_FMT = SATA: RXSTATUS[0]: TXCOMSTART operation complete RXSTATUS[1]: COMWAKE signal received RXSTATUS[2]: COMRESET/COMINIT signal received
TXCOMSTART0 TXCOMSTART1	In	TXUSRCLK2	Initiates the transmission of the COM* sequence selected by TXCOMTYPE (SATA only).
TXCOMTYPE0 TXCOMTYPE1	In	TXUSRCLK2	Selects the type of COM signal to send (SATA only): `0`: COMRESET/COMINIT `1`: COMWAKE
TXELECIDLE0 TXELECIDLE1	In	TXUSRCLK2	When in the P2 power state, this signal controls whether an electrical idle or a beacon indication is driven out onto the TX pair.
TXPOWERDOWN0[1:0] TXPOWERDOWN1[1:0]	In	Async	Powers down the TX lanes. The GTP_DUAL tile must be in the P2 power state (TXPOWERDOWN = `11`) to generate beacon signaling.

Optiver Ex1009
Page 133 of 347

Table 6-22 defines the OOB/beacon signaling attributes.

Table 6-22: **TX OOB/Beacon Signaling Attributes**

Attribute	Description
COM_BURST_VAL_0 COM_BURST_VAL_1	Number of bursts in a COM sequence.
PLL_SATA_0 PLL_SATA_1	Tie to FALSE. When FALSE, PLL_SATA allows TX SATA operations to work at the SATA Generation 1 (1.5 Gb/s) or SATA Generation 2 (3 Gb/s) rate.
PLL_TXDIVSEL_COMM_OUT	Sets the common divider for the TX line rate. If either PLL_TXDIVSEL_OUT is not equal to 1, PLL_TXDIVSEL_COMM_OUT must be set to 1. Can be set to 1, 2, or 4.
PLL_TXDIVSEL_OUT_0 PLL_TXDIVSEL_OUT_1	Sets the divider for the TX line rate for the individual GTP transceiver. If PLL_TXDIVSEL_COMM_OUT is not equal to 1, PLL_TXDIVSEL_OUT must be set to 1. Can be set to 1, 2, or 4.

Description

The GTP_DUAL tile supports two OOB/beacon signaling modes: one for SATA operations and one for PCI Express operations. The use of these two mechanisms is mutually exclusive.

Beacon Signaling for PCI Express Operations

Beacon signaling for PCI Express operations is performed when the GTP_DUAL tile is in the P2 power state. Transmission of a beacon is initiated by the deassertion of TXELECIDLE, as shown in Figure 6-18. FPGA control logic controls beacon timing by the sequencing of TXELECIDLE.

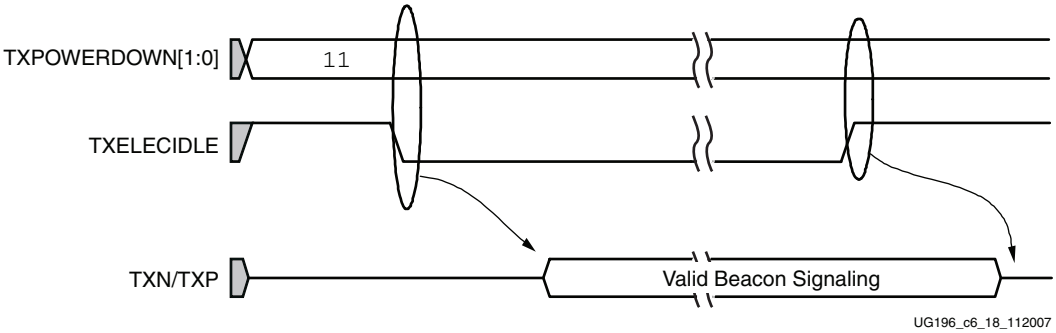


Figure 6-18: **Beacon Generation for PCI Express Operations**

OOB Signaling for SATA Operations

OOB signaling for SATA operation is initiated through the use of the TXELECIDLE, TXCOMSTART, and TXCOMTYPE ports. When TXELECIDLE is held High, assertion of TXCOMSTART for one TXUSRCLK2 cycle initiates the transmission of a COM sequence. The type of COM sequence generated is controlled by the TXCOMTYPE port as shown in Table 6-21.

Apotex Ex1009
Page 134 of 347

The number of bursts in the COM sequence is controlled by the COM_BURST_VAL attribute. The timing of the COM sequences transmitted is correct as long as the PLL clock (see Shared PMA PLL, page 72) is set to 1.5 GHz, and PLL_TXDIVSEL_OUT for each channel is set to either 1 (for a 3.0 Gb/s SATA Generation 2 rate) or 2 (for a 1.5 Gb/s SATA Generation 1 rate).

Note: If the SATA rate needs to be changed at run time (e.g., for SATA auto-negotiation), the GTP_DUAL DRP (see Dynamic Reconfiguration Port, page 101) can be used to change the appropriate PLL_TXDIVSEL_OUT attribute. The COM sequence timing is adjusted automatically.

When a TXCOMSTART operation completes, RXSTATUS[0] is driven High for one RXUSRCLK2 cycle. Be careful not to use the output of RXSTATUS[0] without synchronizing it to the TXUSRCLK2 domain, if TXUSRCLK2 and RXUSRCLK2 are driven by separate clocks.

Apple Ex1009
Page 136 of 347

Chapter 7

GTP Receiver (RX)

This chapter shows how to configure and use each of the functional blocks inside the GTP receiver.

Receiver Overview

Each GTP transceiver includes an independent receiver, made up of a PCS and a PMA. Figure 7-1 shows the functional blocks of the receiver (RX). High-speed serial data flows from traces on the board into the PMA of the RX, into the PCS, and finally into the FPGA logic. Refer to Appendix E, Low Latency Design, for latency information on this block diagram.

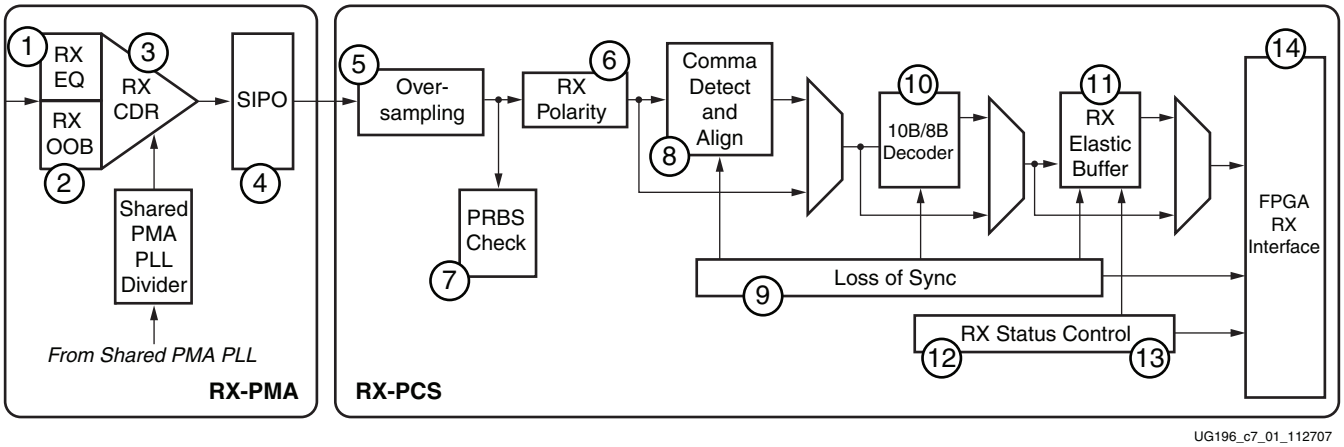


Figure 7-1: **GTP RX Block Diagram**

The key elements within the GTP receiver are:

1. RX Termination and Equalization, page 139
2. RX OOB/Beacon Signaling, page 143
3. RX Clock Data Recovery, page 149
4. Serial In to Parallel Out, page 155
5. Oversampling, page 157
6. RX Polarity Control, page 160
7. PRBS Detection, page 161
8. Configurable Comma Alignment and Detection, page 162
9. Configurable Loss-of-Sync State Machine, page 169

10. Configurable 8B/10B Decoder, page 172
11. Configurable RX Elastic Buffer and Phase Alignment, page 176
12. Configurable Clock Correction, page 183
13. Configurable Channel Bonding (Lane Deskew), page 189
14. FPGA RX Interface, page 200

Apotex Ex1009
Page 138 of 347

RX Termination and Equalization

Overview

The first stage of the RX datapath in each GTP transceiver is the RX current mode logic (CML) receiver. This block determines the value of the incoming high-speed differential signal.

At high speeds, error-free data reception requires good signal integrity. The CML receiver includes circuits to allow the termination of the channel to be optimized for the best possible signal integrity.

The receiver also features an RX equalization circuit that allows it to compensate for high-frequency losses in the channel, improving the quality of the received signal. This circuit can be tuned to meet the specific requirements of the physical channel used in the design.

Ports and Attributes

The combination of the wideband signal and its high-pass filtered derivative compensates for the low-pass characteristic of the transmission line on the board. The parameters of the mixing ratio and the pole of the high-pass filter are described in Table 7-1.

Table 7-1: **RX Termination and Equalization Ports**

Port	Dir	Clock Domain	Description
MGTRXN0 MGTRXN1 MGTRXP0 MGTRXP1	In (Pad)	RX Serial Clock	RXN and RXP are the differential complements of each other forming a differential receiver input pair. These ports represent pads, so their locations must be constrained (see Chapter 4, Implementation), and they must be brought to the top level of the design.
RXENEQB0 RXENEQB1	In	Async	Active-Low port for enabling linear receive equalization: `0`: Receiver equalization is enabled `1`: Receiver equalization is disabled
RXEQMIX0[1:0] RXEQMIX1[1:0]	In	Async	This port controls the wideband/high-pass mix ratio for the RX equalization circuit. The following ratios are available: `00`: 50% wideband, 50% high-pass `01`: 62.5% wideband, 37.5% high-pass `10`: 75% wideband, 25% high-pass `11`: 37.5% wideband, 62.5% high-pass
RXEQPOLE0[3:0] RXEQPOLE1[3:0]	In	Async	This port controls the location of the pole in the RX equalizer high-pass filter. Adjusting this value shifts the threshold for rejecting low-frequency signals. The following settings are available: `0xxx`: Filter pole depends on resistor calibration `1000`: 0% nominal pole `1001`: -12.5% `1010`: -25.0% `1011`: -37.5% `1100`: +12.5% `1101`: +25.0% `1110`: +37.5% `1111`: +50.0%

Optiver Ex1009
Page 139 of 347

Table 7-2 describes the attributes and settings regarding termination for both receivers of the GTP_DUAL tile.

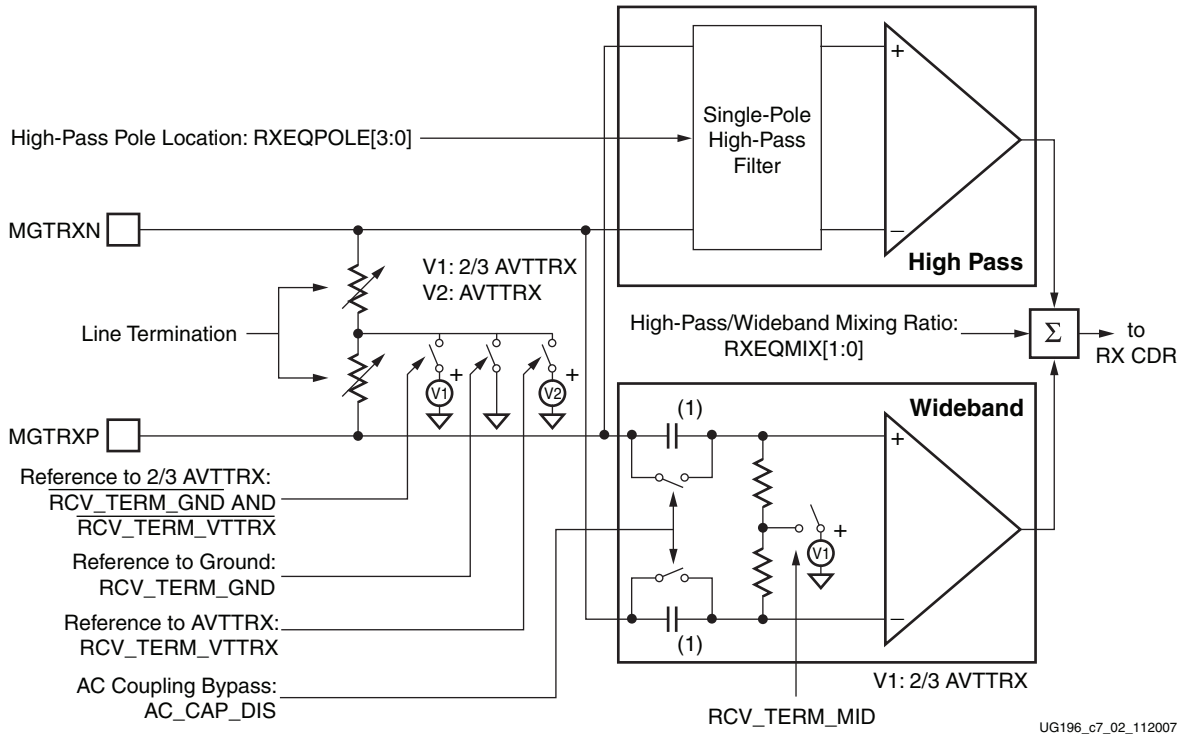
Table 7-2: **RX Termination and Equalization Attributes**

Attribute	Description
AC_CAP_DIS_0 AC_CAP_DIS_1	Bypasses the built-in AC coupling in the receiver. Use this attribute when DC coupling is required. TRUE: Built-in AC coupling capacitors are bypassed. DC coupling to the receiver is possible. FALSE: Built-in AC coupling capacitors are enabled. This setting is the default for this parameter. See Chapter 10, GTP-to-Board Interface, for details about when it is appropriate to add an additional external capacitor for AC coupling.
RCV_TERM_GND_0 RCV_TERM_GND_1	Activates the Ground reference for the receiver termination network. TRUE: Ground reference for receiver termination activated. FALSE: Ground reference for receiver termination disabled. See Table 7-3, page 142 for valid combinations of RX termination attributes.
RCV_TERM_MID_0 RCV_TERM_MID_1	Activates the 2/3 AVTTRX reference for the termination circuit after the built-in AC coupling circuit in the receiver. RCV_TERM_MID should always be set to !AC_CAP_DIS. TRUE: 2/3 AVTTRX reference after built-in AC coupling is activated. FALSE: 2/3 AVTTRX reference after built-in AC coupling is disabled.
RCV_TERM_VTTRX_0 RCV_TERM_VTTRX_1	Activates the AVTTRX reference for receiver termination network. TRUE: AVTTRX reference for receiver termination activated. FALSE: AVTTRX reference for receiver termination disabled. See Table 7-3, page 142 for valid combinations of RX termination attributes.
TERMINATION_IMP_0 TERMINATION_IMP_1	Selects the termination impedance for the TX driver and RX CML receiver. These attributes are always set to 50 to select 50Ω termination impedance. See Chapter 10, GTP-to-Board Interface, for details on calibration of impedance values.

Description

The GTP_DUAL receivers are connected via differential pad pairs RXN and RXP to the transmission line on the board. Figure 7-2 illustrates the internal architecture of one receiver channel inside the GTP_DUAL block.

Oplover Ex1009
Page 140 of 347



Notes:

1. The internal AC coupling capacitor is not a replacement for offchip coupling capacitors. It is an isolation capacitor for nonstandard termination voltages.
2. These port names have the prefix *MGT* to identify them easily in a pad file that is very often used to create symbols for board design schematics. In this document, the MGT prefix was removed from those names. However, names with and without the MGT prefix are synonymous to each other.

Figure 7-2: **Receiver Channel Internal Architecture**

The receiver includes four main features for optimizing signal integrity:

- Optional Built-in AC Coupling
- Configurable Termination Impedance
- Configurable Termination Voltage
- Optional Configurable RX Linear Equalization

Optional Built-in AC Coupling

The GTP receiver includes small AC-coupling capacitors that isolate the receiver from the line. These capacitors are required when the RX termination is set to GND and are enabled by default. When GND termination is not used, these capacitors can be disabled to allow for DC coupling or more optimal off-chip AC coupling. See Chapter 10, GTP-to-Board Interface, for more details about how to pick an appropriately sized external AC coupling capacitor.

The AC_CAP_DIS attribute controls the AC-coupling bypass switch shown in Figure 7-2. As required when built-in AC coupling is activated, RCV_TERM_MID activates the 2/3 MGTAVTTRX reference behind the built-in, AC-coupling circuit.

To turn on built-in AC coupling, AC_CAP_DIS is set to FALSE, and RCV_TERM_MID is set to TRUE. To disable the built-in AC coupling, AC_CAP_DIS is set to TRUE, and RCV_TERM_MID is set to FALSE.

Configurable Termination Impedance

To minimize distortion due to reflections, the termination impedance of each receiver must be matched as closely as possible to the impedance of the serial trace to which it is connected. See Chapter 10, GTP-to-Board Interface, for more information about how the termination impedance is calibrated.

Configurable Termination Voltage

As shown in Figure 7-2, the line termination circuit connects the RXP and RXN pads through the calibrated line termination impedances to one of three termination voltages: GND, MGTAVTTRX, or 2/3 MGTAVTTRX (from the GTP receiver). At any given time, only one of these connections is active.

The appropriate RX termination voltage is selected based on the type of coupling used, the TX termination voltage, and the requirements of the protocol being used. Table 7-3 summarizes the termination options available in the GTP receiver. Table 7-4 shows how to activate the different RX termination options.

Table 7-3: **Recommended RX Termination Settings**

Coupling Type	Valid RX Termination Settings	Notes
DC coupled	2/3 MGTAVTTRX, MGTAVTTRX	The TX termination (and TX termination voltage) must match the selected RX termination (and RX termination voltage) to prevent DC currents. 2/3 MGTAVTTRX is optimal for the receiver.
External AC coupling only	2/3 MGTAVTTRX	Recommended configuration because it permits connections to the widest variety of transceivers including hot-swap applications.
Built-in AC coupling only	MGTAVTTRX, 2/3 MGTAVTTRX, GND	The RX termination (and RX termination voltage) must match the TX termination (and TX termination voltage) to prevent DC currents. The built-in AC-coupling capacitors do not block a DC current path between the RX termination of the receiver and the TX termination of the transmitter. Only an external AC-coupling capacitor prevents a DC current path between the transmitter's TX termination and the receiver's RX termination.
Built-in and external AC coupling	MGTAVTTRX, 2/3 MGTAVTTRX, GND	Use GND to support the PCI Express TXDETECTRX feature. See Receive Detect Support for PCI Express Operation, page 130.

Table 7-4: **RX Termination Attribute Settings**

RX Termination Voltage	RCV_TERM_GND	RCV_TERM_VTTRX
MGTAVTTRX	FALSE	TRUE
2/3 MGTAVTTRX	FALSE	FALSE
GND	TRUE	FALSE

Cpuver Ex1009
Page 142 of 347

Optional Configurable RX Linear Equalization

High-speed serial traces and connections typically attenuate high-frequency signals more than low-frequency signals. As a result, high-frequency data passing through a channel tends to get distorted as the high-frequency components of the signal lose more power than the low-frequency components.

The GTP receiver includes a linear equalizer circuit that can be used to compensate for signal distortion on the line due to high-frequency attenuation. To activate the equalizer, the active-Low RXENEQB is driven Low. The equalizer is disabled by driving RXENEQB High.

While the equalizer is active, it uses a separate receive buffer to filter out low-frequency signals and capture only the high-frequency components of incoming data. This signal is then mixed with the input captured by the regular receive buffer to produce a signal with amplified high-frequency components. The ratio of the signal from the high-frequency buffer and the regular (wideband) buffer is controlled by the RXEQMIX port. Table 7-1, page 139 shows the different ratios.

The cutoff frequency for the high-frequency buffer is controlled by the RXEQPOLE port. By moving the pole of the filter in the high-frequency path, the frequency range of the high-frequency buffer can be controlled. Table 7-1 shows the possible settings for RXEQPOLE.

RX OOB/Beacon Signaling

Overview

The GTP_DUAL tile supports decoding the out-of-band (OOB) sequences described in the Serial ATA (SATA) specification and supports beaconing as described in the PCI Express specification. An overview of OOB signaling and how it is used in these protocols can be found in Appendix B, OOB/Beacon Signaling.

Support for SATA OOB signaling consists of the analog circuitry required to decode the OOB signal state and state machines to decode bursts of OOB signals for SATA COM sequences (COMRESET, COMWAKE, and COMINIT).

The GTP_DUAL tile supports beacons that are PCI Express compliant by using interface signals defined in the *PHY Interface for the PCI Express (PIPE) Specification*. The FPGA logic decodes the beacon sequence.

Optiver Ex1009
Page 143 of 347

Ports and Attributes

Table 7-5 defines the RX OOB/beacon signaling ports.

Table 7-5: **RX OOB/Beacon Signaling Ports**

Port	Dir	Clock Domain	Description
RXELECIDLE0 RXELECIDLE1	Out	Async	Indicates the differential voltage between RXN and RXP dropped below the minimum threshold (OOBDETECT_THRESHOLD). Signals below this threshold are OOB signals. `1`: OOB signal detected. The differential voltage is below the minimum threshold. `0`: OOB signal not detected. The differential voltage is above the minimum threshold. This port is intended for PCI Express and SATA standards.
RXSTATUS0[2:0] RXSTATUS1[2:0]	Out	RXUSRCLK2	The decoding of RXSTATUS[2:0] depends on the setting of RX_STATUS_FMT. When RX_STATUS_FMT = PCIE: `000`: Receiver not present (when in receiver detection sequence)/Received Data OK (during normal operation). `001`: Reserved. `010`: Reserved. `011`: Receiver present (when in receiver detection sequence). `100`: 8B/10B decode error. `101`: RX Elastic Buffer Overflow. Stays asserted until cleared (different from that defined in the PIPE specification). `110`: RX Elastic Buffer Underflow. Stays asserted until cleared (different from that defined in the PIPE specification). `111`: Receive Disparity Error. When RX_STATUS_FMT = SATA: RXSTATUS[0]: TXCOMSTART operation complete RXSTATUS[1]: COMWAKE signal received RXSTATUS[2]: COMRESET/COMINIT signal received
RXVALID0 RXVALID1	Out	RXUSRCLK2	Indicates symbol lock and valid data on RXDATA and RXCHARISK[1:0] when High, as defined in the PIPE specification.

Apotex Ex1009
Page 144 of 347

Table 7-6 defines the RX OOB/beacon signaling attributes.

Table 7-6: **RX OOB/Beacon Signaling Attributes**

Attribute	Description
OOB_CLK_DIVIDER	Sets the squelch clock rate. The squelch clock must be set between 25 MHz and 37.5 MHz, but as close to 25 MHz as possible for the SATA OOB detector to work correctly. Squelch Clock rate = CLKIN/OOB_CLK_DIVIDER Valid divider settings are 1, 2, 4, 6, 8, 10, 12, and 14.
OOBDETECT_THRESHOLD_0 OOBDETECT_THRESHOLD_1	Sets the minimum differential voltage between RXN and RXP. When the differential voltage drops below this level, the incoming signal is an OOB signal. This 3-bit binary encoded attribute has the following nominal values of the OOB threshold voltage(1):

Value	OOB Nominal Threshold Voltage [mV]
000	70
001	85
010	92
011	99
100	105
101	112
110	118
111	127

RX_STATUS_FMT_0 RX_STATUS_FMT_1	Defines which status encoding is used: PCIE: PCI Express encoding SATA: SATA encoding
SATA_BURST_VAL_0 SATA_BURST_VAL_1	Number of bursts required to declare a COM match. The default for SATA_BURST_VAL is 4, which is the burst count specified in SATA for COMINIT, COMRESET, and COMWAIT.
SATA_IDLE_VAL_0 SATA_IDLE_VAL_1	Number of idles required to declare a COM match. Each idle is an OOB signal with a length that matches either COMINIT/COMRESET or COMWAIT. When the SATA detector starts to count one type of idle (for example, COMRESET/COMINIT), it resets the count if it receives the other type. This value defaults to 3 to match the SATA specification.
SATA_MAX_BURST_0 SATA_MAX_BURST_1	Sets the threshold for the SATA detector to reject a burst in terms of squelch clock cycles. SATA_MAX_BURST has valid values between 1 and 61 (the default is 7) and must be greater than SATA_MIN_BURST. See the Description section to learn how to calculate the best value for a given squelch clock rate.
SATA_MAX_INIT_0 SATA_MAX_INIT_1	Sets the maximum time allowed for a COMINIT/COMRESET idle for the SATA detector in terms of squelch clock cycles. SATA_MAX_INIT has valid values between 1 and 61 (the default is 22) and must be greater than SATA_MIN_INIT. See the Description section to learn how to calculate the best value for a given squelch clock rate.

Table 7-6: **RX OOB/Beacon Signaling Attributes *(Continued)***

Attribute	Description
SATA_MAX_WAKE_0 SATA_MAX_WAKE_1	Sets the maximum time allowed for a COMWAKE idle for the SATA detector in terms of squelch clock cycles. SATA_MAX_WAKE has valid values between 1 and 61 (the default is 7) and must be greater than SATA_MIN_WAKE. See the Description section to learn how to calculate the best value for a given squelch clock rate.
SATA_MIN_BURST_0 SATA_MIN_BURST_1	Sets the threshold for the SATA detector to reject a burst in terms of squelch clock cycles. SATA_MIN_BURST has valid values between 1 and 61 (the default is 5) and must be less than SATA_MAX_BURST. See the Description section to learn how to calculate the best value for a given squelch clock rate.
SATA_MIN_INIT_0 SATA_MIN_INIT_1	In SATA, OOB signals are used as idles in COMINIT, COMRESET, and COMWAKE. The minimum length of an idle that must be accepted for COMINIT/COMRESET signals in SATA is 304 ns. If the idle is shorter than 175 ns, it cannot be used for COMINIT/COMRESET. SATA_MIN_INIT is used to set the minimum time allowed for a COMINIT/COMRESET idle for the SATA detector in terms of squelch clock cycles. SATA_MIN_INIT has valid values between 1 and 61 (the default is 12) and must be less than SATA_MAX_INIT. See the Description section to learn how to calculate the best value for a given squelch clock rate. The squelch clock is set based using OOB_CLK_DIVIDER.
SATA_MIN_WAKE_0 SATA_MIN_WAKE_1	In SATA, OOB signals are used as idles in COMINIT, COMRESET, and COMWAKE. The minimum length of an idle that must be accepted for COMWAKE signals in SATA is 101 ns. If the idle is shorter than 55 ns, it cannot be used for COMWAKE. SATA_MIN_WAKE is used to set the minimum time allowed for a COMWAKE idle for the SATA detector in terms of squelch clock cycles. SATA_MIN_WAKE has valid values between 1 and 61 (the default is 4) and must be less than SATA_MAX_WAKE. See the Description section to learn how to calculate the best value for a given squelch clock rate. The squelch clock is set based using OOB_CLK_DIVIDER.

Notes:

1. OOB nominal values, consult DS202: *Virtex-5 FPGA Data Sheet* for specified OOB values.

Description

The GTP_DUAL tile supports two RX status modes: one for SATA operation and one for PCI Express operation. The decoding mode is configured by the RX_STATUS_FMT attribute.

Detecting Electrical Idle for PCI Express Operation

Regardless of the state of the RX_STATUS_FMT attribute, the RXELECIDLE port indicates if the differential voltage between the RXN and RXP serial I/O pins falls within the OOB threshold defined by the OOBDETECT_THRESHOLD attribute. This signal can be used to decode PCI Express beacon sequences. The latency between the arrival of an OOB signal and the assertion of RXELECIDLE is found in the *Virtex-5 FPGA Data Sheet*.

Apotex Ex1009
Page 146 of 347

Detecting OOB for SATA Operation

Each GTP transceiver includes a SATA OOB detector to decode SATA COM sequences. When RX_STATUS_FMT is set to *SATA*, the pins of the RXSTATUS port are used to indicate the arrival of COM sequences. Figure 7-3 shows the SATA OOB detector. It divides CLKIN down to create a squelch clock that runs at approximately 25 MHz. This clock is used to sample the output of the block that detects OOB signals to look for transitions between regular data and OOB signals. Both edges of the squelch clock are used. The squelch detector FSM uses the transitions to calculate the length of each burst and each idle. It uses this information to drive the RXSTATUS port, indicating which COM sequences have been found.

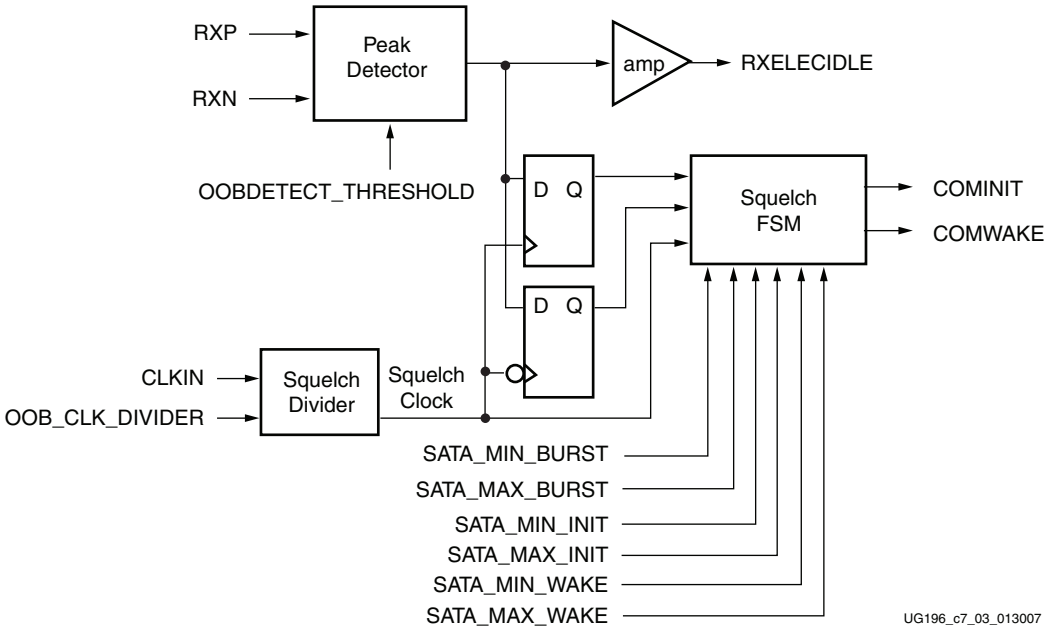


Figure 7-3: **SATA OOB Detector Block Diagram**

Before the SATA OOB detector can work, it must be configured to use the reference clock provided (CLKIN). The OOB_CLK_DIVIDER attribute must be set to produce a squelch clock between 25 MHz and 37.5 MHz, but as close to 25 MHz as possible. In addition, the MIN and MAX times for bursts and idles must be set based on the squelch clock rate. The formula to set the squelch clock is shown below:

$$\text{Parameter in squelch cycles} = \left(\frac{\text{Parameter in ns}}{1000}\right) \times \text{squelch clock frequency in MHz} \times 2 \qquad \textit{Equation 7-1}$$

All the minimum values are defined with a minimum time below which the signal must be rejected, and a minimum time above which the signal always meets the minimum time requirement. After calculating each of these in terms of squelch clock cycles, the appropriate MIN parameter is set to an integer value between these two numbers.

Similarly, for all maximum values, there is a maximum time above which the signal must be rejected, and a maximum time below which the signal always meets the maximum time requirement. After calculating these in terms of squelch clock, the appropriate MAX parameter is set to an integer value between these two numbers.

Example

Table 7-7 shows some example settings for the squelch clock divider.

Table 7-7: **Example Clock Generation Settings for SATA**

Parameter	Example 1	Example 2	Example 3	Example 4	Example 5	Example 6
CLKIN (MHz)	25	75	100	150	250	300
OOB_CLK_DIVIDER	1	2	4	6	10	12
Squelch Clock (MHz)	25	37.5	25	25	25	25
Effective Sample Period (ns)	20	13.33333333	20	20	20	20

Table 7-8 shows all the minimum and maximum values defined in SATA for burst and idle lengths, along with example calculations based on the squelch clock frequencies computed in Table 7-7.

Table 7-8: **Example SATA Attribute Settings**

Parameter	ns	cycles	cycles	cycles	cycles	cycles	cycles
Shortest burst width that must be rejected	55	2.8	4.1	2.8	2.8	2.8	2.357142857
MinBurstWidth		**4**	**6**	**4**	**4**	**4**	**3**
Shortest burst width that must be accepted	101	5.1	7.6	5.1	5.1	5.1	4.341428571
Nominal burst length	107	5.3	8.0	5.3	5.3	5.3	4.572857143
Longest burst width that must be accepted	112	5.6	8.4	5.6	5.6	5.6	4.8
MaxBurstWidth		**7**	**11**	**7**	**7**	**7**	**6**
Longest burst width that must be rejected	175	8.8	13.1	8.8	8.8	8.8	7.5
Shortest idle width that must be rejected for COMINIT	175	8.8	13.1	8.8	8.8	8.8	7.5
MinInitWidth		**12**	**18**	**12**	**12**	**12**	**10**
Shortest idle width that must be accepted for COMINIT	304	15.2	22.8	15.2	15.2	15.2	13.02857143
Nominal idle width for COMINIT	320	16.0	24.0	16.0	16.0	16.0	13.71428571
Longest idle width that must be accepted for COMINIT	336	16.8	25.2	16.8	16.8	16.8	14.4
MaxInitWidth		**22**	**32**	**22**	**22**	**22**	**18**
Longest idle width that must be rejected for COMINIT	525	26.3	39.4	26.3	26.3	26.3	22.5
Shortest idle width that must be rejected for COMWAKE	55	2.8	4.1	2.8	2.8	2.8	2.357142857
MinWakeWidth		**4**	**6**	**4**	**4**	**4**	**3**
Shortest idle width that must be accepted for COMWAKE	101.3	5.1	7.6	5.1	5.1	5.1	4.341428571
Nominal idle width for COMWAKE	106.7	5.3	8.0	5.3	5.3	5.3	4.572857143
Longest idle width that must be accepted for COMWAKE	112	5.6	8.4	5.6	5.6	5.6	4.8
MaxWakeWidth		**7**	**11**	**7**	**7**	**7**	**6**
Longest idle width that must be rejected for COMWAKE	175	8.8	13.1	8.8	8.8	8.8	7.5

Cpuver Ex1009
Page 148 of 347

RX Clock Data Recovery

Overview

The RX Clock Data Recovery (CDR) circuit in each GTP transceiver extracts a recovered clock from incoming data. As long as the line rate of the recovered clock matches the line rate of the receiver within ±1000 ppm and there are sufficient transitions in the data, the CDR can extract a clock. The CDR has advanced features, including a scanning feature that can be used to evaluate the quality of the received signal.

Ports and Attributes

Table 7-9 defines RX CDR signaling ports.

Table 7-9: **RX CDR Signaling Ports**

Port	Dir	Clock Domain	Description
RESETDONE0 RESETDONE1	Out	Async	This port goes High when the GTP transceiver has finished reset and is ready for use. For this signal to work correctly, CLKIN and all clock inputs on the individual GTP transceiver (TXUSRCLK, TXUSRCLK2, RXUSRCLK, RXUSRCLK2) must be driven.
RXCDRRESET0(1) RXCDRRESET1(2)	In	RXUSRCLK2	Individual reset signal for the RX CDR and the RX part of the PCS for this channel. This signal is driven High to cause the CDR to give up its current lock and return to the shared PMA PLL frequency.
RXELECIDLERESET	In	Async	This port is required to hold the CDR in reset while the receiver is in electrical idle. This functionality is required when using OOB signaling and also during start-up when transients might temporarily put the CDR in the electrical idle state. The RX Clock Data Recovery section shows how this port must be connected for all GTP designs. `0`: CDR operates normally. `1`: CDR held in reset. RXELECIDLERESET must be High while RXELECIDLE is High during normal operation.
RXENELECIDLERESETB	In	Async	This port is required to enable the CDR reset function while the receiver is in electrical idle. This functionality is required when using OOB signaling and also during start-up when transients might temporarily put the CDR in the electrical idle state. The RX Clock Data Recovery section shows how this port must be connected for all GTP designs. `0`: CDR reset function enabled `1`: CDR reset function disabled

Notes:

1. When this reset is active, RESETDONE0 is driven Low. All resets are asynchronous, positive-edge triggered, and synchronized internally to a specific clock domain.
2. When this reset is active, RESETDONE1 is driven Low. All resets are asynchronous, positive-edge triggered, and synchronized internally to a specific clock domain.

Table 7-10 defines the RX CDR attributes.

Table 7-10: **RX CDR Attributes**

Attribute	Description
PMA_CDR_SCAN_0 PMA_CDR_SCAN_1	This 27-bit attribute allows direct control of the CDR sampling point. In normal operation, this attribute should be left at the default value set by the RocketIO GTP Transceiver Wizard.
PMA_RX_CFG_0 PMA_RX_CFG_1	This 25-bit attribute allows the operation of the CDR to be adjusted for tests. In normal operation, this attribute must be left at the default value set by the RocketIO GTP Transceiver Wizard.

Description

Before serial data received from the line can be used, the embedded clock in the signal must be recovered. The CDR circuit in each GTP transceiver is responsible for this function. It takes a divided, high-speed serial clock from the shared PMA PLL and adjusts its phase and frequency until its transitions match the incoming data. As shown in Figure 7-4, the result is a clock that matches the clock originally used to generate the serial stream.

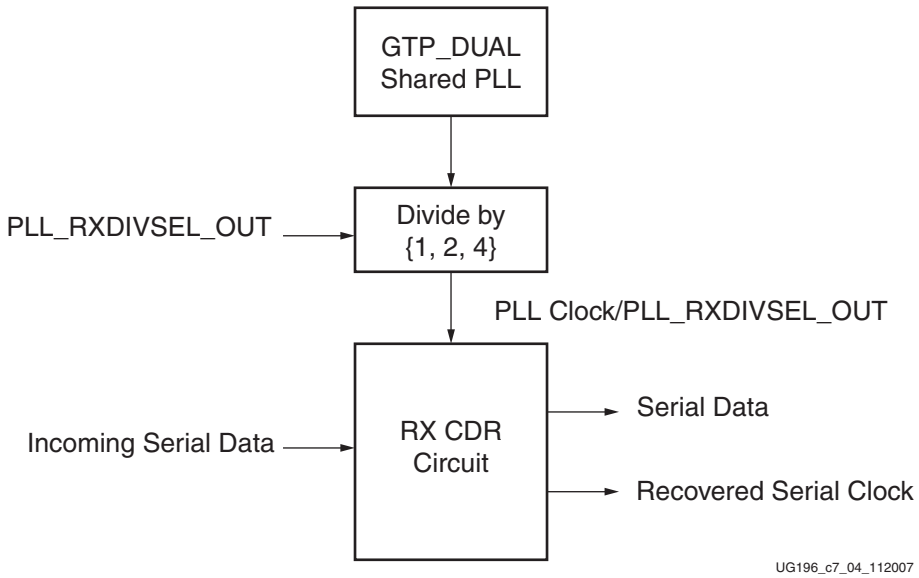


Figure 7-4: **Conceptual View of RX CDR Circuit**

Because transitions in the incoming data are used to recover the serial clock, long runs without transitions can introduce error.

Cypher Ex1009
Page 150 of 347

CDR Reset

The CDR must be reset before it can operate on incoming data. There are several ways to reset the CDR:

- Use the GTPRESET port to reset all components in the GTP_DUAL tile, including the CDR in each transceiver. See Reset, page 84 for more details.
- Use the RXCDRRESET port to reset the CDR block, the OOB circuits for SATA (see RX OOB/Beacon Signaling, page 143), the RX elastic buffer (see Configurable RX Elastic Buffer and Phase Alignment, page 176), and the remaining sections of the RX PCS.
- Use the RXELECIDLERESET and RXENELECIDLERESETB ports, which are part of the link idle reset described in Reset, page 84. The Link Idle Reset circuit must be implemented whenever a GTP receiver is used with the second-order loop turned on. The RXENELECIDLERESET port enables the RXELECIDLE RESET port, which is used to reset the CDR without resetting any other blocks in response to electrical idle conditions.

Figure 7-5 shows the timing of the internal reset signals when RXCDRRESET is asserted. RXCDRRESET can be asserted asynchronously. When it is deasserted, an internal CDR reset pulse, synchronized to an internally generated 1 MHz clock, resets the CDR. Similarly, a reset pulse is generated for the SATA OOB circuit (internal SATA reset), the RX PCS datapath (internal RXRESET), and the RX buffer (internal RXBUFRESET). The entire sequence completes in approximately 5 µs.

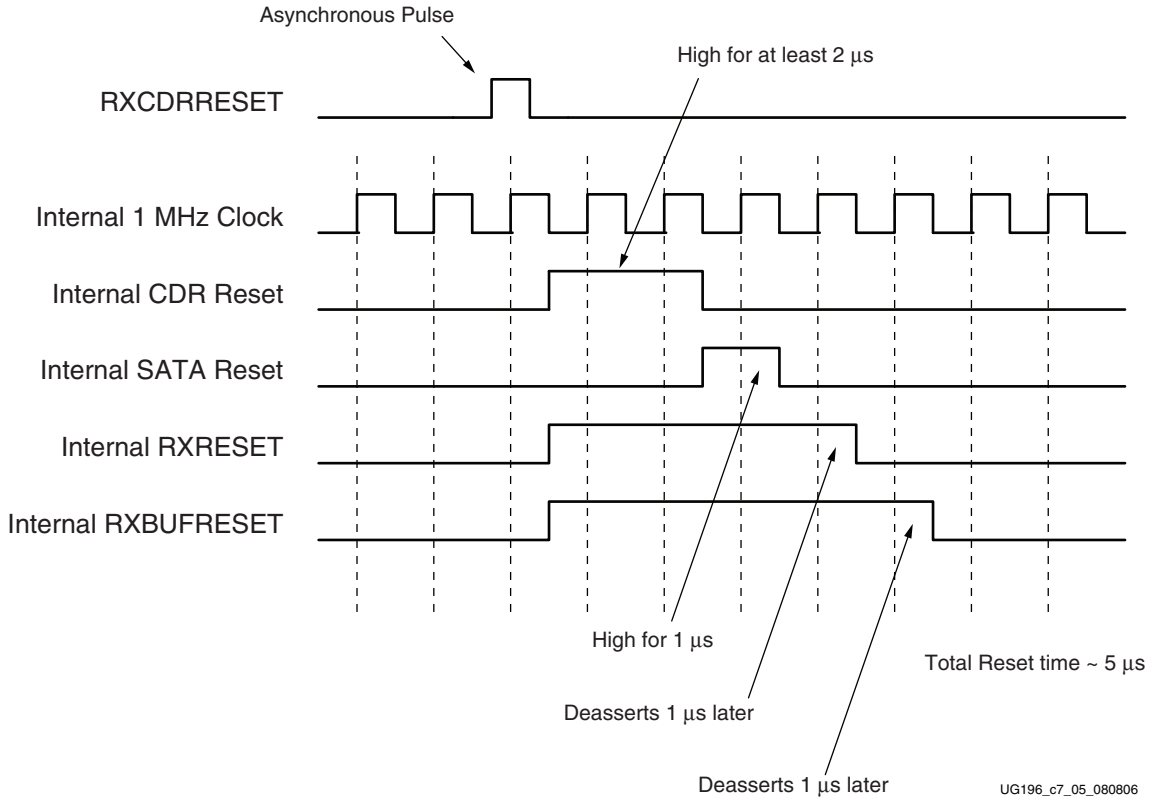


Figure 7-5: **Reset Sequence Triggered by RXCDRRESET**

Tuning the CDR

Depending upon application requirements, the PMA_RX_CFG attribute can be used to optimize the performance of the CDR. It controls specific functions of the CDR and its loop filter. When using this parameter, the transceiver does not operate unless used as recommended in this section. The CDR includes an optional second-order loop to let the CDR track the incoming data with wider frequency differences.

The optimum PMA_RX_CFG settings for the CDR are outlined in this section for three possible application categories. In Table 7-11, the last column (second-order loop ON/OFF) is for information purposes only because it is automatically set by the PMA_RX_CFG value.

Normal Operation Mode

Synchronous-systems applications are where the local reference clock and the reference clock of incoming data come from the same source (SONET, for example). In some other synchronous-system applications, the system interface is synchronized where a clean-up PLL cleans the recovered clock and uses it as the local reference clock for the GTP_DUAL tile.

Spread-spectrum applications, such as Serial ATA, use spread-spectrum clocking where both the local reference clock and the incoming data are frequency-modulated with up to 0.5% downspread. For SATA, the interface can be asynchronous where along with instantaneous frequency differences caused by the spread-spectrum modulation, there can also be a ±1000 ppm static frequency difference between the local reference clock and the reference clock of incoming data.

When interfacing the GTP receiver to a spread-spectrum signal, the PLL_RXDIVSEL_OUT attribute must be set to “1” to limit the line rates that can be received while using spread-spectrum clocking to those greater than 2 Gb/s.

Spread-spectrum clocking applications in synchronous systems is when the spread spectrum clock is applied to both sides of the link simultaneously and the two communicating ports are not allowed to differ by more than 600 ppm at any time (PCI Express systems).

All applications where a Lock-to-reference scheme or oversampling mode is not required must configure the CDR for normal-operation mode.

Applications Using GTP Oversampling Mode

System interfaces running between 100 Mb/s and 500 Mb/s are required to use the GTP transceiver's built-in 5x oversampler. These systems must use the PMA_RX_CFG attribute settings for the Oversampling mode.

Apotex Ex1009
Page 152 of 347

Applications Using Lock-to-Reference Mode

In some applications, it is better for the CDR to sample data using the local reference clock without attempting to lock to the incoming frequency (lock-to-reference). Lock-to-reference is typically used in applications that perform digital oversampling in the FPGA logic. Table 7-11 shows the required PMA_RX_CFG to put the CDR into lock-to-reference mode.

Table 7-11: **Required PMA_RX_CFG Setting for Different Operation Modes**

Application	Output Divider Settings	PMA_RX_CFG Setting	1st Order Loop(1)	2nd Order Loop(1)
Oversampling mode, asynchronous applications up to ±100 ppm(2), synchronous protocols (e.g., SONET, CPRI)	1	25'h09f0088	ON	OFF
	2	25'h09f0088	ON	OFF
	4	25'h09f0088	ON	OFF
Asynchronous applications(3) up to ±1000 ppm, spread spectrum clocking, PCIe applications	1	25'h09f0089	ON	ON
	2(4)	25'h09f0051	ON	ON
	4(4)	25'h09f0051	ON	ON
Lock-to-reference mode	1, 2, 4	25'h09F0000	OFF	OFF

Notes:

1. The first order loop provides fine adjustment of the sample point, which is sufficient in applications with small ppm offsets. If the application has a large ppm offset, the second order loop is also needed to allow a faster adjustment of the sample point.
2. The maximum ppm offset allowed depends on the selected output divider setting. Refer to the "RocketIO GTP Transceiver Specifications" section in the *Virtex-5 FPGA Data Sheet* for the maximum ppm offsets.
3. In asynchronous applications where the variance is greater than ±100 ppm, the second order loop of the CDR is needed to adjust the sample point. For protocols with a small variance, such as SONET and CPRI, only the first order loop is required to adjust the sample point. The second order loop is necessary for asynchronous applications where the variance is greater than ±100 ppm.
4. When interfacing the GTP receiver to a spread-spectrum signal, the PLL_RXDIVSEL_OUT attribute must be set to "1" to limit the line rates that can be received while using spread-spectrum clocking to those greater than 2 Gb/s.

Horizontal Sample Point Shift

One advanced feature of the CDR of the GTP transceiver is the built-in horizontal sample point shift. During normal operation, the CDR finds the transition points in incoming data and uses them to recover the frequency of the incoming clock.

The transition points are also used to select the optimal time to sample data. To minimize the chance of error, the CDR attempts to sample data as far from transition points as possible (that is, at the time when the bit value is most stable). This position is the center of the data eye (see Figure 7-6).

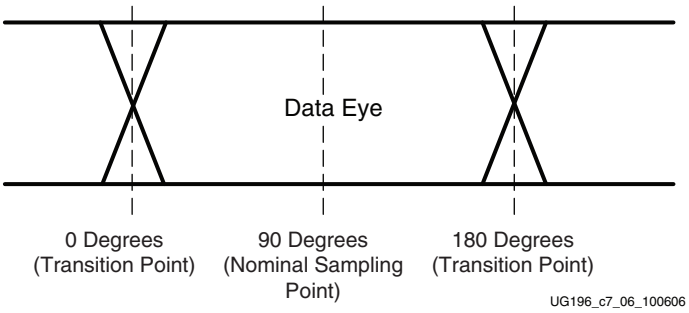


Figure 7-6: **Normal CDR Operation**

A rough measure of the quality of the incoming signal, as seen by the CDR, can be obtained by determining how close to the transition points of the eye the sampling point can be before the number of received bit errors increases significantly. Figure 7-7 shows an example of a possible scan with an eye from clean data (good signal integrity) versus a scan of bad data.

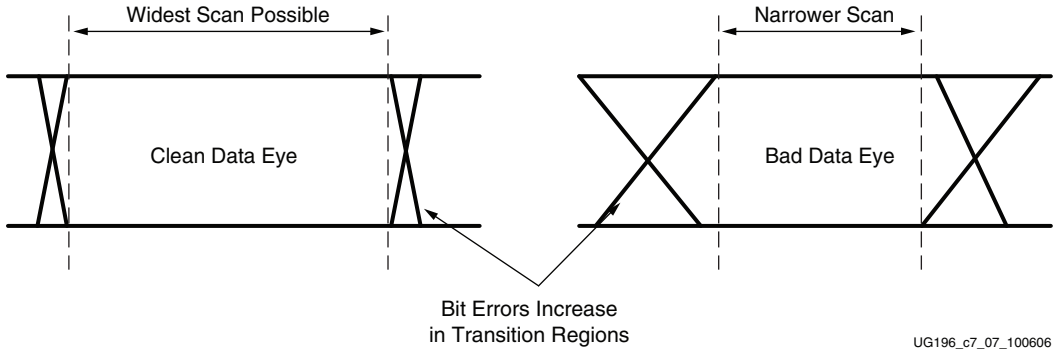


Figure 7-7: **Scanning to Evaluate the Data Eye**

This functionality is supported through the PMA_CDR_SCAN attribute. Dynamic scans are performed by changing PMA_CDR_SCAN using the DRP while receiving known data, such as a PRBS pattern to determine the number of bit errors.

PMA_CDR_SCAN has an 8-bit field that controls the position of the data sampling point relative to the first transition point in the eye. Resetting the field to 0 puts the sampling point at the first transition point. Setting the field to 127 puts the data sampling point at the 180° point, the second transition point of the eye (see Figure 7-6). The default value for each field is 64, the nominal 90° sampling point.

There are three fields to accommodate the different possible PLL_RXDIVSEL_OUT settings. Table 7-12 shows how to set PMA_CDR_SCAN for PLL_RXDIVSEL_OUT = 1.

The horizontal sample point shift is not supported for PLL_RXDIVSEL_OUT = 2 or PLL_RXDIVSEL_OUT = 4.

Table 7-12: **Settings for PMA_CDR_SCAN based on PLL_RXDIVSEL_OUT**

PLL_RXDIVSEL_OUT	PMA_CDR_SCAN					
	[26]	[25]	[24]	[23:16]	[15:8]	[7:0]
1	1	1	0	8'hC0(1)	8'h76(1)	8'h00 - 8'h80

Notes:

1. Do not change from the default value set by the RocketIO GTP Transceiver Wizard. When changing the lower bits of PMA_CDR_SCAN, the upper bits must be masked to prevent an accidental change from the default value.

Apotex Ex1009
Page 154 of 347

Serial In to Parallel Out

Overview

The Serial In to Parallel Out (SIPO) block deserializes serial data from the GTP receiver PMA and presents it as parallel data to the PCS.

Ports and Attributes

Table 7-13 defines the SIPO ports.

Table 7-13: **SIPO Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the width of the internal datapath for the entire GTP_DUAL tile. This shared port is also described in Shared PMA PLL, page 72. `0`: Internal datapath is 8 bits wide `1`: Internal datapath is 10 bits wide

Table 7-14 defines the SIPO attributes.

Table 7-14: **SIPO Attributes**

Attribute	Description
OVERSAMPLE_MODE	This shared attribute activates the built-in 5x digital oversampling circuits in both GTP_DUAL transceivers. Oversampling must be enabled when running the GTP transceivers at line rates between 100 Mb/s and 500 Mb/s. TRUE: Built-in 5x digital oversampling enabled for both GTP transceivers on the tile FALSE: Digital oversampling disabled See Oversampling, page 157 for more details about 5x digital oversampling.
PLL_RXDIVSEL_OUT_0 PLL_RXDIVSEL_OUT_1	This divider defines the nominal line rate for the receiver. It can be set to 1, 2, or 4. RX Line Rate = PLL Clock * 2/PLL_RXDIVSEL_OUT

Description

The SIPO block is the heart of the RX datapath. It uses both edges of a high-speed clock to deserialize incoming data and present it to the PCS.

The serial clock rate to the SIPO, which is the RX line rate of the GTP transceiver, depends on the PLL clock rate and the setting of PLL_RXDIVSEL_OUT in the GTP transceiver. Equation 7-2 shows how to set the RX line rate. See Shared PMA PLL, page 72 for more information on how to set the PLL clock rate.

$$\text{RX Line Rate} = \frac{\text{PLL Clock} \times 2}{\text{PLL\_RXDIVSEL\_OUT}} \qquad \textit{Equation 7-2}$$

The parallel clock rate to the parallel side of the SIPO is the XCLK rate of the GTP transceiver. This rate matches the USERCLK rate of the GTP transceiver, which is used internally in the PCS. See Configurable RX Elastic Buffer and Phase Alignment, page 176 for more details about the clock domains in the RX side of the GTP transceiver. The XCLK rate depends on the internal datapath width used in the tile, because this value is the width of the parallel data the SIPO produces. Equation 7-3 shows how to calculate the parallel (XCLK) rate of the SIPO.

$$\text{RX XCLK Rate} = \frac{\text{RX Line Rate}}{\text{Internal Datapath Width}} \qquad \textit{Equation 7-3}$$

When OVERSAMPLE_MODE is FALSE, the internal datapath width is 8 bits when INTDATAWIDTH is Low and is 10 bits when INTDATAWIDTH is High. When OVERSAMPLE_MODE is TRUE, the internal datapath width is 10 bits. See Oversampling, page 157 for more details about receiver operation when oversampling is activated.

Both the serial and parallel clocks for the SIPO are generated from the recovered clock in the CDR circuit.

Oplover Ex1009
Page 156 of 347

Oversampling

Overview

Each GTP transceiver includes built-in 5x oversampling to enable serial rates from 100 Mb/s to 500 Mb/s. At these low rates, the regular CDR must operate at five times the desired line rate to stay within its operating limits.

The digital oversampling circuit takes parallel data from the SIPO at five times the desired line rate and uses the position of bit value transitions to recover a clock. The transition points are also used to pick an optimal sampling point to recover 2 bits of data from each set of 10 bits presented.

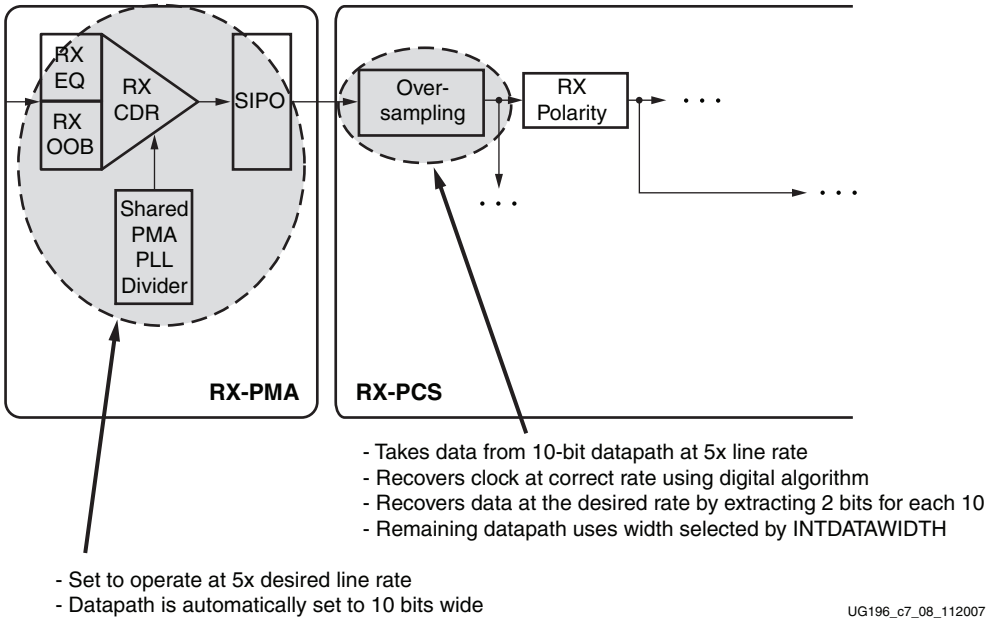


Figure 7-8: **GTP RX Block Diagram**

Ports and Attributes

Table 7-15 defines the ports for built-in digital oversampling.

Table 7-15: **RX DCDR Ports**

Port	Dir	Clock Domain	Description
RXENSAMPLEALIGN0 RXENSAMPLEALIGN1	In	RXUSRCLK2	When High, the 5x oversampler in the PCS continually adjusts its sample point. When Low, it samples only at the point that was active before the port went Low.
RXOVERSAMPLEERR0 RXOVERSAMPLEERR1	Out	RXUSRCLK2	When High, indicates the FIFO in the oversampling circuit has either overflowed or underflowed. The PCS must be reset to resume proper operation.

Table 7-16 defines the attributes for built-in digital oversampling.

Table 7-16: **RX DCDR Attributes**

Attribute	Description
OVERSAMPLE_MODE	This shared attribute is also defined in Shared PMA PLL, page 72 and TX Buffering, Phase Alignment, and TX Skew Reduction, page 115. It applies to both sides of both GTP transceivers on the GTP_DUAL tile. When TRUE, 5x oversampling is ON.

Description

Each GTP transceiver includes a built-in 5x digital oversampling circuit, which must be used when operating the transceiver at line rates between 100 and 500 Mb/s. Oversampling applies to both transceivers in a GTP_DUAL tile. If oversampling is activated for one transceiver, it is activated for both.

Configuring the GTP transceiver to use oversampling requires the following steps:

- Configuring the 5x line rate
- Configuring the PCS internal datapath and clocks
- Activating and operating the oversampling block

The RocketIO GTP Transceiver Wizard automatically configures the GTP_DUAL tile and makes the oversampling ports available when generating a GTP wrapper with oversampling enabled.

Configuring the 5x Line Rate

The SIPO in the RX PMA must provide the oversampling block with 10 bits per parallel cycle, sampled at a rate five times faster than the desired line rate. The required line rate for the PMA is given by Equation 7-4.

$$PMALineRate = 5 \times DesiredLineRate \qquad \textit{Equation 7-4}$$

To achieve the required line rate, the RX divider for the transceiver must be set such that the resulting required PLL clock rate is within the PLL operating range of 1.0 to 2.0 GHz (see the *Virtex-5 FPGA Data Sheet*). Equation 7-5 shows the relationship between the required rate and the PLL clock. Serial In to Parallel Out, page 155 provides more information about the local RX divider.

$$f_{PLLClock} = \frac{PMALineRate \times PLL\_RXDIVSEL\_OUT}{2} \qquad \textit{Equation 7-5}$$

The shared PMA PLL, which is discussed in detail in Shared PMA PLL, page 72, must be set to produce the required PLL clock frequency. Equation 7-6 shows how the PLL clock frequency is related to the frequency of CLKIN (the reference clock to the tile). Oversampling mode automatically uses a 10-bit internal datapath in the PMA, regardless of the INTDATAWIDTH setting. The lowest possible ratio of PLL_DIVSEL_FB to PLL_DIVSEL_REF is recommended.

$$f_{PLLClock} = f_{CLKIN} \times \frac{5 \times PLL\_DIVSEL\_FB}{PLL\_DIVSEL\_REF} \qquad \textit{Equation 7-6}$$

Apotex Ex1009
Page 158 of 347

Configuring the PCS Internal Datapath and Clocks

Figure 7-9 shows the clock domains of the GTP RX datapath when oversampling is used. The RX serial clock runs at the PMA line rate calculated previously. The XCLK runs at the resulting parallel rate for a 10-bit datapath, PMALineRate/10. The oversampled data from the SIPO is fed into the oversampling block at the XCLK rate.

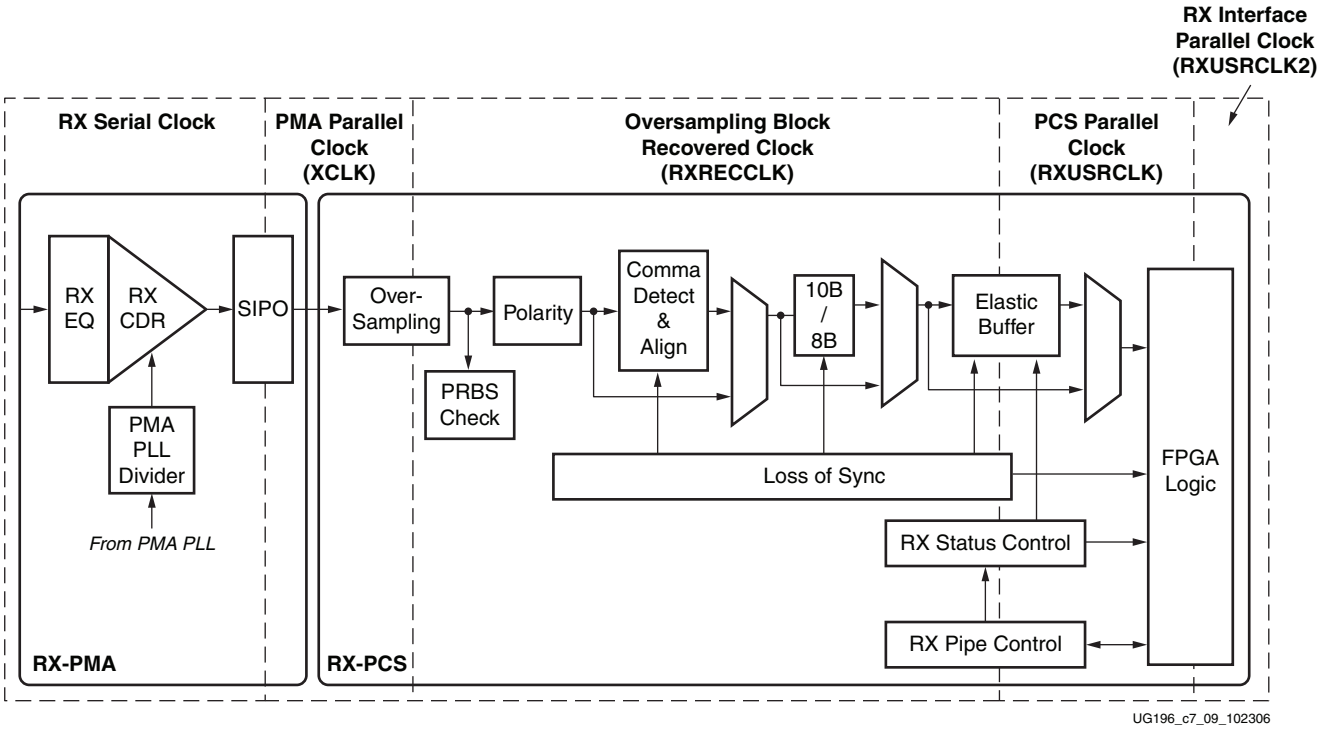


Figure 7-9: **RX Clock Domains When Using Built-In Oversampling**

The oversampling block produces 2 bits of data for every 10 bits received. This data is fed into the remaining PCS datapath at the required clock rate for the desired line rate, which is given by Equation 7-5. This PCS rate is used for RXUSRCLK. A multiple of the rate is used for RXUSRCLK2, depending on the selected RX datapath width.

When oversampling is used, the oversampled line rate, not the PMA line rate, and the PCS internal datapath width, which is set by INTDATAWIDTH, must be used for RXUSRCLK calculations. FPGA RX Interface, page 200 provides more details about the relationship between RXUSRCLK and RXUSRCLK2.

Activating and Operating the Oversampling Block

After the PMA line rate and PCS datapath are set, the oversampling block can be enabled by setting OVERSAMPLING_MODE to TRUE. This attribute affects both transceivers in the GTP_DUAL tile.

The oversampling block includes a small buffer to hold data before passing it to the PCS. This buffer can overflow/underflow if the PMA and PCS frequencies are different, such as if RXUSRCLK stopped temporarily due to other events in the system. If an error in the oversampling block buffer occurs, the transceiver asserts the RXOVERSAMPLEERR signal. This error can be cleared by asserting RXRESET or RXCDRRESET.

RXENSAMPLEALIGN is tied High so the oversampling block always attempts to find the best possible recovered clock and sample point in the incoming data.

RX Polarity Control

Overview

The GTP receiver can invert incoming data using the RX polarity control function. This function is useful in designs where the RXP and RXN signals can be accidentally connected in reverse. The RXPOLARITY port is driven High to invert the polarity of incoming data.

Ports and Attributes

Table 7-17 defines the RX polarity ports.

Table 7-17: **RX Polarity Ports**

Port	Dir	Clock Domain	Description
RXPOLARITY0 RXPOLARITY1	In	RXUSRCLK2	The RX Polarity port is used to invert the polarity of incoming data. `1`: Invert polarity `0`: Regular polarity

There are no attributes in this section.

Description

The RX polarity port is used to invert the polarity of incoming data. Driving this port High causes the RXN pin to be treated as RXP and the RXP pin to be treated as RXN.

Cypher Ex1009
Page 160 of 347

PRBS Detection

Overview

The GTP receiver includes a built-in PRBS checker. This checker can be set to check for one of three industry-standard PRBS patterns. The checker is self-synchronizing and works on the incoming data before comma alignment or decoding. This function can be used to test the signal integrity of the channel.

Ports and Attributes

Table 7-18 defines the PRBS detection ports.

Table 7-18: **PRBS Detection Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the width of the internal datapath for the entire GTP_DUAL tile. The PRBS checker is only valid when INTDATAWIDTH is High (10-bit internal datapath).
PRBSCNTRESET0 PRBSCNTRESET1	In	RXUSRCLK2	Resets the PRBS error counter. PRBSCNTRESET is applied synchronously and is effective only on its rising edge.
RXENPRBSTST0[1:0] RXENPRBSTST1[1:0]	In	RXUSRCLK2	Receiver test pattern checker control: `00`: Disable PRBS checkers `01`: Enable 2^7-1 PRBS checker `10`: Enable 2^{23}-1 PRBS checker `11`: Enable 2^{31}-1 PRBS checker INTDATAWIDTH must also be High (10-bit internal data width mode) when the PRBS checker is enabled.
RXPRBSERR0 RXPRBSERR1	Out	RXUSRCLK2	RXPRBSERR goes High when the number of errors in PRBS testing equals or exceeds the value set by the PRBS_ERR_THRESHOLD attribute.

Table 7-19 defines the PRBS detection attributes.

Table 7-19: **PRBS Detection Attributes**

Attribute	Description
PRBS_ERR_THRESHOLD0 PRBS_ERR_THRESHOLD1	Sets the error threshold for the PRBS checker. If PRBS testing is enabled, a counter counts the number of errors. If the number of errors equals or exceeds the value of PRBS_ERR_THRESHOLD, the output RXPRBSERR goes High. This attribute is set as a 32-bit hex value.

Description

To use the built-in PRBS checker, RXENPRBSTST is set to match the PRBS pattern being sent to the receiver. The RXENPRBSTST entry in Table 7-18 shows the available settings. Only the 10-bit internal data width mode is supported, which implies that INTDATAWIDTH must also be driven High when the PRBS checker is enabled.

When the PRBS checker is running, it attempts to find the selected PRBS pattern in the incoming data. When it finds the pattern, it can detect PRBS errors by comparing the incoming pattern with the expected pattern.

The checker counts the number of errors it sees and compares it with PRBS_ERR_THRESHOLD. When the error count equals or exceeds the threshold, RXPRBSERR is asserted. Asserting PRBSCNTRESET clears RXPRBSERR. GTPRESET, RXCDRRESET, and RXRESET also reset the count.

Configurable Comma Alignment and Detection

Overview

Serial data must be aligned to symbol boundaries before it can be used as parallel data. To make alignment possible, transmitters send a recognizable sequence, usually called a *comma*. The receiver searches for the comma in the incoming data. When it finds a comma, it moves the comma to a byte boundary so the received parallel words match the transmitted parallel words.

Figure 7-10 shows the alignment to a 10-bit comma. The TX parallel data is on the left. The serial data with the comma is highlighted in the middle. The RX receiving unaligned bits are on the right side.

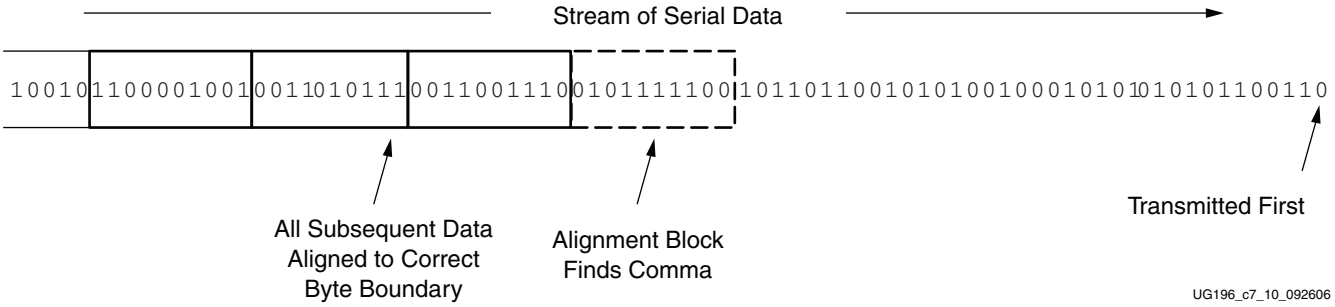


Figure 7-10: **Conceptual View of Comma Alignment (Aligning to a 10-Bit Comma)**

Apple Ex1009
Page 162 of 347

Figure 7-11 shows the TX parallel data is on the left side, and the RX receiving recognizable parallel data is on the right side.

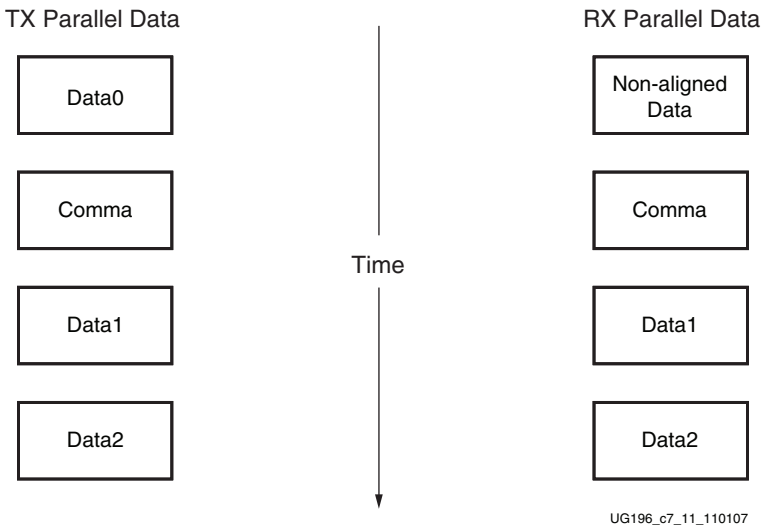


Figure 7-11: **Parallel Data View of Comma Alignment**

The GTP transceiver includes an alignment block that can be programmed to align specific commas to various byte boundaries, or to manually align data using attribute settings (see Table 7-21, page 165). SONET A1/A2 alignment is possible using comma double mode. The block can be bypassed to reduce latency if it is not needed.

Ports and Attributes

Table 7-20 defines the RX comma alignment and detection ports.

Table 7-20: **RX Comma Alignment and Detection Ports**

Port	Dir	Clock Domain	Description
RXBYTEISALIGNED0 RXBYTEISALIGNED1	Out	RXUSRCLK2	This signal from the comma detection and realignment circuit is High to indicate that the parallel data stream is properly aligned on byte boundaries according to comma detection. `0`: Parallel data stream not aligned to byte boundaries `1`: Parallel data stream aligned to byte boundaries There are several cycles after RXBYTEISALIGNED is asserted before aligned data is available at the FPGA RX interface. RXBYTEISALIGNED responds to plus comma alignment when PCOMMA_DETECT_(0/1) is TRUE. RXBYTEISALIGNED responds to minus comma alignment when MCOMMA_DETECT_(0/1) is TRUE.
RXBYTEREALIGN0 RXBYTEREALIGN1	Out	RXUSRCLK2	This signal from the comma detection and realignment circuit indicates that the byte alignment within the serial data stream has changed due to comma detection. `0`: Byte alignment has not changed `1`: Byte alignment has changed Data can be lost when alignment occurs, which can cause data errors (and disparity errors when the 8B/10B decoder is used).
RXCOMMADET0 RXCOMMADET1	Out	RXUSRCLK2	This signal is asserted when the comma alignment block detects a comma. The assertion occurs several cycles before the comma is available at the FPGA RX interface. `0`: Comma not detected `1`: Comma detected
RXCOMMADETUSE0 RXCOMMADETUSE1	In	RXUSRCLK2	RXCOMMADETUSE activates the comma detection and alignment circuit. `0`: Bypass the circuit `1`: Use the comma detection and alignment circuit Bypassing the comma and alignment circuit reduces RX datapath latency.
RXENMCOMMAALIGN0 RXENMCOMMAALIGN1	In	RXUSRCLK2	Aligns the byte boundary when *comma minus* is detected. `0`: Disabled `1`: Enabled
RXENPCOMMAALIGN0 RXENPCOMMAALIGN1	In	RXUSRCLK2	Aligns the byte boundary when *comma plus* is detected. `0`: Disabled `1`: Enabled
RXSLIDE0 RXSLIDE1	In	RXUSRCLK2	RXSLIDE implements a comma alignment *bump* control. When RXSLIDE is asserted, the byte alignment is adjusted by one bit, which permits determination and control of byte alignment by the FPGA logic. Each assertion of RXSLIDE causes just one adjustment. RXSLIDE must be deasserted for two RXUSRCLK2 cycles before it can be reasserted to cause another adjustment. When asserted, RXSLIDE takes precedence over normal comma alignment.

Apotex Ex1009
Page 164 of 347

Table 7-21 defines the RX comma alignment and detection attributes.

Table 7-21: **RX Comma Alignment and Detection Attributes**

Attribute	Description
ALIGN_COMMA_WORD_0 ALIGN_COMMA_WORD_1(1)	Controls alignment of detected commas within a multi-byte datapath. 1: Align comma to either byte within a 2-byte datapath. The comma can be aligned to either the *even* byte [9:0] or the *odd* byte [19:10] of RXDATA at the FPGA when the 2-byte RX interface is selected. 2: Align comma to the *even* byte within a 2-byte datapath. The aligned comma is guaranteed to be aligned to byte RXDATA[9:0]. For ALIGN_COMMA_WORD = 2 to work properly in conjunction with the elastic buffer, both CLK_COR_ADJ_LEN and CLK_COR_MIN_LAT must be even. Protocols that send commas in even and odd positions must set ALIGN_COMMA_WORD to 1. If RXDATAWIDTH is driven Low, ALIGN_COMMA_WORD must be set to 1.
COMMA_10B_ENABLE_0 COMMA_10B_ENABLE_1	Determines which bits of MCOMMA/PCOMMA must be matched to incoming data and which bits can be any value. This attribute is a 10-bit mask with a default value of `1111111111`. Any bit in the mask that is reset to `0` effectively turns the corresponding bit in MCOMMA or PCOMMA to a don't care bit.
COMMA_DOUBLE_0 COMMA_DOUBLE_1	Specifies whether a comma match consists of either a comma plus or a comma minus alone, or whether both are required in the sequence. FALSE: The plus comma (PCOMMA) and minus comma (MCOMMA) are handled separately. An individual match for either can lead to comma detection and alignment. TRUE: A comma match consists of a comma plus followed immediately by a comma minus. The match pattern is 20 or 16 bits (as determined by INTDATAWIDTH). When COMMA_DOUBLE is TRUE, PCOMMA_DETECT should be the same as MCOMMA_DETECT, and RXENPCOMMAALIGN should be the same as RXENMCOMMAALIGN.
MCOMMA_10B_VALUE_0 MCOMMA_10B_VALUE_1	Defines comma minus to raise RXCOMMADET and aligns the parallel data. Reception order is right to left. (MCOMMA_10B_VALUE[0] is received first.) The default value is `1010000011` (K28.5). This definition does not affect 8B/10B encoding or decoding.
MCOMMA_DETECT_0 MCOMMA_DETECT_1	Controls raising of RXCOMMADET on comma minus. FALSE: Do not raise RXCOMMADET when comma minus is detected. TRUE: Raise RXCOMMADET when comma minus is detected. This setting does not affect comma alignment.
PCOMMA_10B_VALUE_0 PCOMMA_10B_VALUE_1	Defines comma plus to raise RXCOMMADET and aligns the parallel data. Reception order is right to left. (PCOMMA_10B_VALUE[0] is received first.) The default value is `0101111100` (K28.5). This definition does not affect 8B/10B encoding or decoding.
PCOMMA_DETECT_0 PCOMMA_DETECT_1	Controls raising of RXCOMMADET on comma plus. FALSE: Do not raise RXCOMMADET when comma plus is detected. TRUE: Raise RXCOMMADET when comma plus is detected. (This setting does not affect comma alignment.)

Table 7-21: **RX Comma Alignment and Detection Attributes *(Continued)***

Attribute	Description
RX_SLIDE_MODE_0 RX_SLIDE_MODE_1	Selects between sliding in the PMA or in the PCS. Legal values are PCS (default) and PMA.

Notes:

1. When ALIGN_COMMA_WORD = 2, RXRESET must be applied after completion of an RXCDRRESET sequence or after resetting the RX buffer using RXBUFRESET.

Description

Enabling Comma Alignment

To enable the comma alignment block, the RXCOMMADETUSE port is driven High. RXCOMMADETUSE is driven Low to bypass the block completely for minimum latency.

Configuring Comma Patterns

To set the comma pattern that the block searches for in the incoming data stream, the MCOMMA_10B_VALUE, PCOMMA_10B_VALUE, and COMMA_10B_ENABLE attributes are used. The comma lengths depend on the INTDATAWIDTH of the tile (see Shared PMA PLL, page 72). Figure 7-12 shows how the COMMA_10B_ENABLE masks each of the comma values to allow partial pattern matching.

Figure 7-12 shows how a COMMA is combined with COMMA_ENABLE to make a wildcarded comma for a 10-bit internal comma.

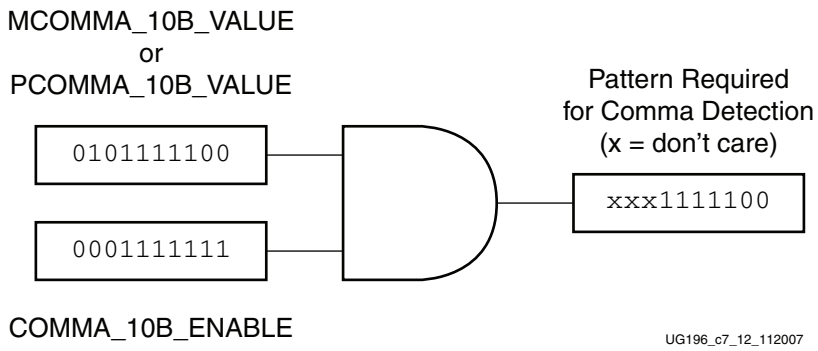


Figure 7-12: **Comma Pattern Masking**

If COMMA_DOUBLE is TRUE, the MCOMMA and PCOMMA patterns are combined so that the block searches for two commas in a row. The number of bits in the comma depends on INTDATAWIDTH (see Shared PMA PLL, page 72). Figure 7-13 shows how the commas are combined when COMMA_DOUBLE is TRUE.

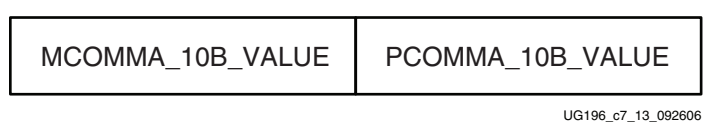


Figure 7-13: **Extended Comma Pattern Definition**

Figure 7-14 shows how COMMA_10B_ENABLE and wildcarding work for a double-width comma.

Oplver Ex1009
Page 166 of 347

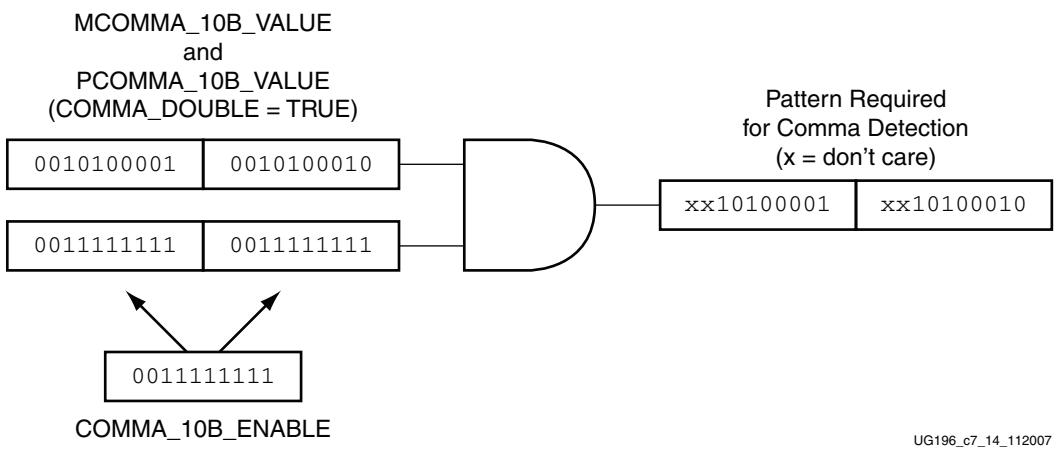


Figure 7-14: **Extended Comma Pattern Masking**

Activating Comma Alignment

Commas are aligned to the closest boundary providing they are found while comma alignment is active. RXENMCOMMAALIGN is driven High to align on the MCOMMA pattern. RXENPCOMMAALIGN is driven High to activate alignment on the PCOMMA pattern. Both enable ports are driven to align to either pattern. When COMMA_DOUBLE is TRUE, both enable ports should always be driven to the same value.

Alignment Status Signals

While MCOMMA or PCOMMA alignment is active, any matching comma pattern causes the block to realign to the closest boundary. After successful alignment, the block holds RXBYTEISALIGNED High. At this time, RXENMCOMMAALIGN and RXENPCOMMAALIGN can be driven Low to turn off alignment and keep the current alignment position. PCOMMA_DETECT must be TRUE for PCOMMAs to cause RXBYTEISALIGNED to go High. Similarly, MCOMMA_DETECT must be TRUE for MCOMMAs to cause RXBYTEISALIGNED to go High.

Commas can arrive while RXBYTEISALIGNED is High. If the commas arrive aligned to boundaries, there is no change. If the commas arrive out of position, the block deasserts RXBYTEISALIGNED until the commas are aligned again. If alignment is still activated for the comma that arrives, the block automatically aligns the new comma to the closest boundary and drives RXBYTEREALIGN High for one RXUSRCLK2 cycle.

Alignment Boundaries

The legal boundaries for alignment are defined by ALIGN_COMMA_WORD. The spacing of the legal boundaries is determined by INTDATAWIDTH, and the number of legal boundary positions is determined by the number of bytes in the RXDATA interface. Figure 7-15 shows the boundaries that can be selected.

RXDATAWIDTH	ALIGN_COMMA_WORD	Possible RX Alignments (Grey = Comma Can Appear on Byte)
0 (1-byte)	1 (Any Boundary)	RXDATA Byte 0
0 (1-byte)	2 (Even Boundary Only)	Invalid Configuration
1 (2-byte)	1 (Any Boundary)	RXDATA Byte 1 | RXDATA Byte 0
1 (2-byte)	2 (Even Boundaries Only)	RXDATA Byte 1 | RXDATA Byte 0

When ALIGN_COMMA_WORD = 2, RXRESET must be applied after completion of an RXCDRRESET sequence or after resetting the RX buffer using RXBUFRESET.

UG196_c7_15_100409

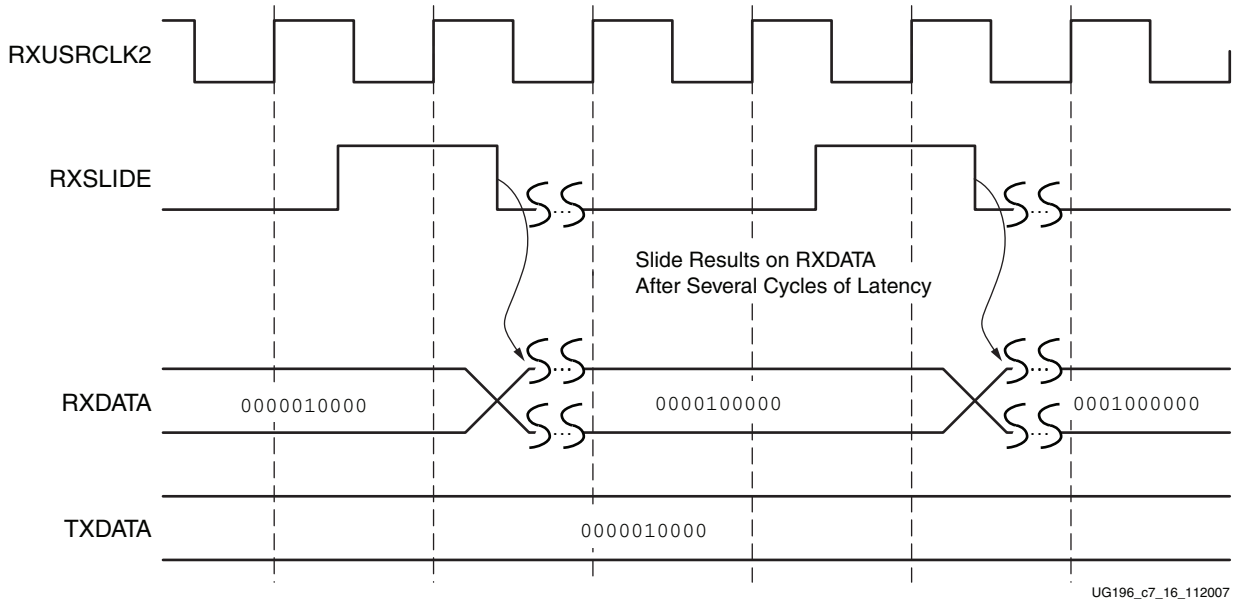
Figure 7-15: **Comma Alignment Boundaries**

Manual Alignment

RXSLIDE can be used to override the automatic comma alignment and to shift the parallel data. RXSLIDE is driven High for one RXUSRCLK2 cycle to shift the parallel data to the left by one bit. RXSLIDE must be Low for at least two RXUSRCLK2 cycles before it can be used again.

Figure 7-16 shows the waveforms for manual alignment using RXSLIDE, before and after the data shift.

Apotex Ex1009
Page 168 of 347



Notes:

1. Latency between the slide and the slide result at RXDATA depends on the number of active RX PCS blocks in the datapath.

Figure 7-16: **Manual Data Alignment Using RXSLIDE**

Configurable Loss-of-Sync State Machine

Overview

Several 8B/10B protocols make use of a standard Loss-of-Sync (LOS) state machine to detect when the channel is malfunctioning. Each GTP receiver includes a LOS state machine that can be activated for protocols requiring it. When the state machine is not used, the LOS state machine's ports can be re-used to monitor the condition of incoming data.

Ports and Attributes

Table 7-22 defines the RX loss-of-sync state machine ports.

Table 7-22: **RX Loss-of-Sync State Machine Ports**

Port	Dir	Clock Domain	Description
RXLOSSOFSYNC0[1:0] RXLOSSOFSYNC1[1:0]	Out	RXUSRCLK2	FPGA Status related to byte stream synchronization. The meaning depends on the state of the RX_LOSS_OF_SYNC_FSM attribute. If RX_LOSS_OF_SYNC_FSM = TRUE, this output reflects the state of an internal Loss-of-Sync FSM as follows: [1] = 1: Sync lost due to either sequence of invalid characters or reset [0] = 1: In the resync state due to a channel bonding sequence or realignment If RX_LOSS_OF_SYNC_FSM = FALSE, this output presents the following information about incoming data: [1] = 1: Received data is not an 8B/10B character or has a disparity error [0] = 1: Channel bonding sequence detected in data

Table 7-23 defines the RX loss-of-sync state machine attributes.

Table 7-23: **RX Loss-of-Sync State Machine Attributes**

Attribute	Description
RX_LOS_INVALID_INCR_0 RX_LOS_INVALID_INCR_1	Defines the number of valid characters required to *cancel out* the appearance of one invalid character for the purpose of loss-of-sync determination. Valid settings are 1, 2, 4, 8, 16, 32, 64, and 128.
RX_LOS_THRESHOLD_0 RX_LOS_THRESHOLD_1	When divided by RX_LOS_INVALID_INCR_(0/1), defines the number of invalid characters required to cause an FSM transition to the *sync lost* state. Valid settings are 4, 8, 16, 32, 64, 128, 256, and 512.
RX_LOSS_OF_SYNC_FSM_0 RX_LOSS_OF_SYNC_FSM_1	RX_LOSS_OF_SYNC_FSM defines the behavior of the RXLOSSOFSYNC[1:0] outputs. FALSE: RXLOSSOFSYNC[1] goes High when invalid data (not in table or disparity error) is found in 8B/10B decoding. RXLOSSOFSYNC[0] goes High when a channel bonding sequence has been written into the elastic buffer. TRUE (default): Loss of sync FSM is in operation and its state is reflected on RXLOSSOFSYNC[1].

Description

Figure 7-17 shows the standard LOS state machine, used in several 8B/10B protocols (for example, XAUI) to detect problems in the incoming data stream.

Cypher Ex1009
Page 170 of 347

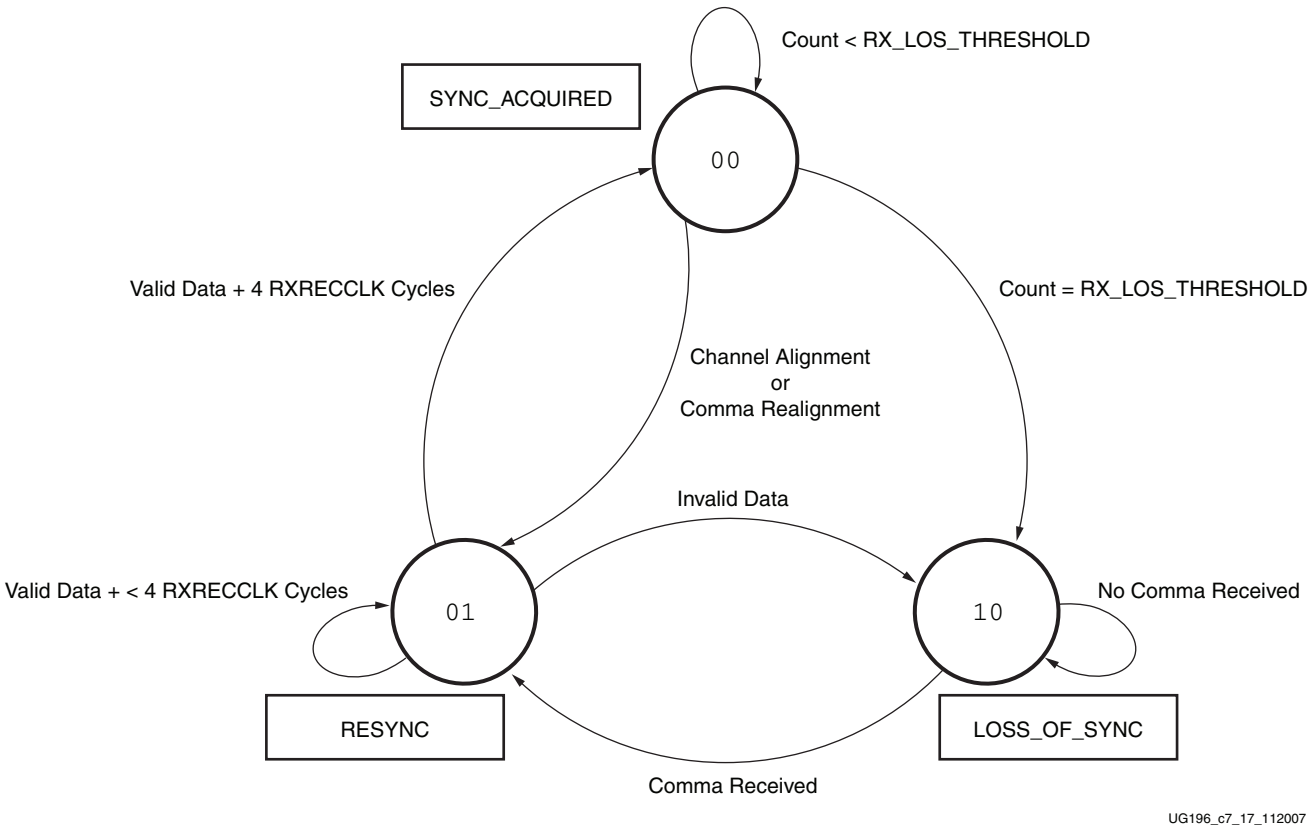


Figure 7-17: **LOS State Machine**

To activate the LOS state machine in the GTP transceiver, RX_LOSS_OF_SYNC_FSM is set to TRUE. While the state machine is active, the RXLOSSOFSYNC port presents its current state.

If the LOS state machine is inactive (RX_LOSS_OF_SYNC_FSM = FALSE), the RXLOSSOFSYNC port presents information about the received data. The RXLOSSOFSYNC entry in Table 7-22 shows the meaning of the RXLOSSOFSYNC port in this case.

The operation of the LOS state machine can be tuned using the RX_LOS_INVALID_INCR and RX_LOS_THRESHOLD attributes. RX_LOS_THRESHOLD adjusts how sensitive the LOS state machine is to bad characters by adjusting the number of characters required to force the machine from the SYNC_ACQUIRED state to the LOSS_OF_SYNC state. The RX_LOS_THRESHOLD entry in Table 7-23 shows the valid settings for this attribute.

The LOS state machine allows the error count in the SYNC_ACQUIRED state to decrease over time, so that sparse errors are eventually discarded. The rate that the error count is decreased is controlled by the RX_LOS_INVALID_INCR attributes, as defined in Table 7-23.

Configurable 8B/10B Decoder

Overview

Many protocols require receivers to decode 8B/10B data. 8B/10B is an industry standard encoding scheme that trades two bits of overhead per byte for improved performance. Table 6-3, page 111 outlines the benefits and costs of 8B/10B. Appendix C shows how 8B/10B maps 10-bit sequences to 8-bit data and control values.

The GTP transceiver includes an 8B/10B decoder to decode RX data without consuming FPGA resources. The decoder includes status signals to indicate errors and incoming control sequences. If decoding is not needed, the block can be disabled to minimize latency.

Ports and Attributes

Table 7-24 defines the RX decoder ports.

Table 7-24: **RX Decoder Ports**

Port	Dir	Clock Domain	Description
RXCHARISCOMMA0[1:0] RXCHARISCOMMA1[1:0]	Out	RXUSRCLK2	RXCHARISCOMMA is asserted when RXDATA is an 8B/10B comma. This signal, which depends on DEC_MCOMMA_DETECT and DEC_PCOMMA_DETECT, is always Low when RXDEC8B10BUSE is Low. RXCHARISCOMMA is a 2-bit signal. Bit 0 corresponds to the lower byte of RXDATA, and bit 1 corresponds to the upper byte. When RXDATAWIDTH is Low (1-byte interface), only bit 0 is used.
RXCHARISK0[1:0] RXCHARISK1[1:0]	Out	RXUSRCLK2	RXCHARISK is asserted when RXDATA is an 8B/10B K character. This signal is always Low when RXDEC8B10BUSE is Low. RXCHARISK is a 2-bit signal. Bit 0 corresponds to the lower byte of RXDATA, and bit 1 corresponds to the upper byte. When RXDATAWIDTH is Low (1-byte interface), only bit 0 is used.
RXDEC8B10BUSE0 RXDEC8B10BUSE1	In	RXUSRCLK2	RXDEC8B10BUSE enables the 8B/10B decoder. `1`: 8B/10B decoder enabled `0`: 8B/10B decoder bypassed (reduces latency)
RXDISPERR0[1:0] RXDISPERR1[1:0]	Out	RXUSRCLK2	When High, RXDISPERR indicates that RXDATA was received with a disparity error. RXDISPERR is a 2-bit signal. Bit 0 corresponds to the lower byte of RXDATA, and bit 1 corresponds to the upper byte. When RXDATAWIDTH is Low (1-byte interface), only bit 0 is used.
RXNOTINTABLE0[1:0] RXNOTINTABLE1[1:0]	Out	RXUSRCLK2	RXNOTINTABLE indicates that RXDATA is the result of an illegal 8B/10B code and is in error. RXNOTINTABLE is a 2-bit signal. Bit 0 corresponds to the lower byte of RXDATA, and bit 1 corresponds to the upper byte. When RXDATAWIDTH is Low (1-byte interface), only bit 0 is used.
RXRUNDISP0[1:0] RXRUNDISP1[1:0]	Out	RXUSRCLK2	RXRUNDISP shows the running disparity of the 8B/10B encoder when RXDATA is received. RXRUNDISP is a 2-bit signal. Bit 0 corresponds to the lower byte of RXDATA, and bit 1 corresponds to the upper byte. When RXDATAWIDTH is Low (1-byte interface), only bit 0 is used.

Cpuver Ex1009
Page 172 of 347

Table 7-25 defines the RX decoder attributes.

Table 7-25: **RX Decoder Attributes**

Attributes	Description
DEC_MCOMMA_DETECT_0 DEC_MCOMMA_DETECT_1	Enables detection of negative 8B/10B commas: TRUE: RXCHARISCOMMA is asserted when RXDATA is a negative 8B/10B comma FALSE: RXCHARISCOMMA does not respond to negative 8B/10B commas
DEC_PCOMMA_DETECT_0 DEC_PCOMMA_DETECT_1	Enables detection of positive 8B/10B commas: TRUE: RXCHARISCOMMA is asserted when RXDATA is a positive 8B/10B comma FALSE: RXCHARISCOMMA does not respond to positive 8B/10B commas
DEC_VALID_COMMA_ONLY_0 DEC_VALID_COMMA_ONLY_1	Limits the set of commas to which RXCHARISCOMMA responds. TRUE: RXCHARISCOMMA is asserted only for K28.1, K28.5, and K28.7 (see 8B/10B K character table in Appendix C, 8B/10B Valid Characters) FALSE: RXCHARISCOMMA responds to any positive or negative 8B/10B comma, depending on the settings for DEC_MCOMMA_DETECT and DEC_PCOMMA_DETECT

Description

Enabling the 8B/10B Decoder

To disable the 8B/10B decoder, RXDEC8B10BUSE is driven Low. With the decoder off, the number of bits per byte on the RX interface is determined by the internal data width of the tile. See FPGA RX Interface, page 200 for details about the deserialization order and parallel bitmapping when 8B/10B decoding is bypassed. When the 8B/10B decoder is bypassed (disabled), RXDEC8B10BUSE equals Low, and INTDATAWIDTH equals 1. The aligned and not decoded data is present on RXDATA, RXCHARISK, and RXDISPERR as illustrated in Table 7-33, page 198.

To enable the 8B/10B decoder, RXDEC8B10BUSE is driven High. INTDATAPATH must also be High (10-bit internal datapath) to enable the 8B/10B decoder because it requires 10-bit wide data.

8B/10B Decoder Bit and Byte Order

The order of the bits into the 8B/10B decoder is the opposite of the order shown in the 8B/10B table in Appendix C, 8B/10B Valid Characters. 8B/10B requires bit a0 to be received first, but the GTP transceiver always receives the right-most bit first. Consequently, the 8B/10B decoder is designed to automatically reverse the bit order of received data before decoding it.

Similarly, because the GTP transceiver receives the right-most bit first, when a 2-byte interface is used, the first byte received (byte 0) is presented on RXDATA[7:0], and the second byte is presented on RXDATA[15:8]. Figure 7-18 shows how the decoder maps 10-bit data to 8-bit values.

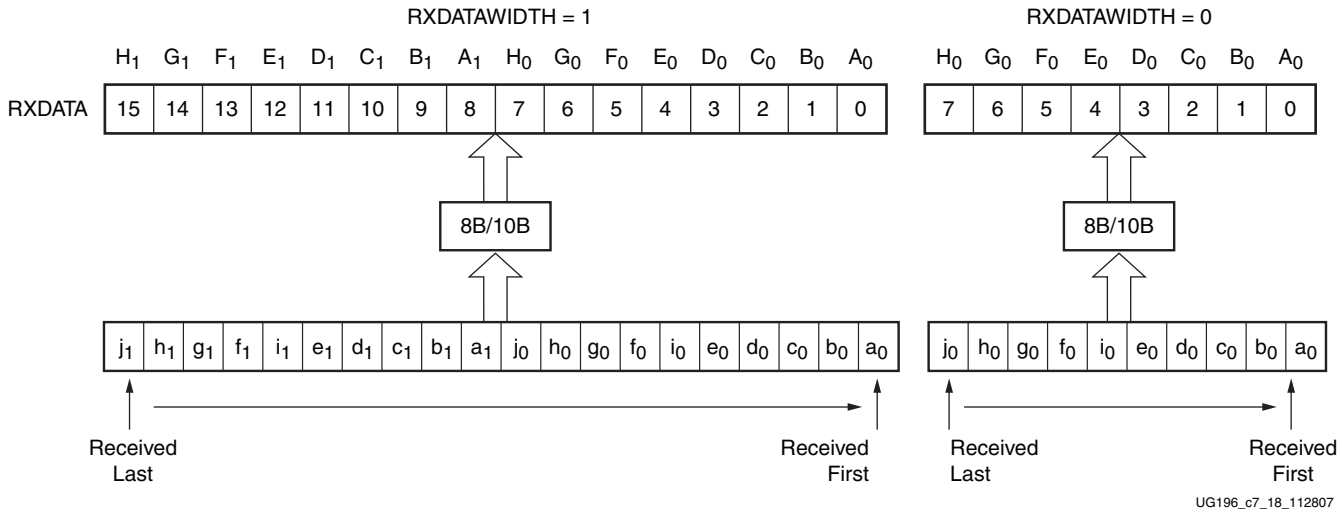


Figure 7-18: **RX Interface with 8B/10B Decoding**

K Characters and 8B/10B Commas

The 8B/10B table (shown in Appendix C, 8B/10B Valid Characters) includes special characters (K characters) that are often used for control functions. When RXDATA is a K character, the decoder drives RXCHARISK High.

If DEC_PCOMMA_DETECT is TRUE, the decoder drives RXCHARISCOMMA High whenever RXDATA is a positive 8B/10B comma. Likewise, if DEC_MCOMMA_DETECT is TRUE, the decoder drives RXCHARISCOMMA High whenever RXDATA is a negative 8B/10B comma.

To limit the set of commas that trigger RXCHARISCOMMA to K28.1, K28.5, and K28.7, DEC_VALID_COMMA_ONLY is set to TRUE. This setting is typically used for Ethernet-based applications. RXCHARISCOMMA does not depend on MCOMMA_10B_VALUE or PCOMMA_10B_VALUE.

RX Running Disparity

The 8B/10B decoder uses a running disparity system to balance the number of `1`s and `0`s transmitted. The 8B/10B decoder tracks the running disparity of incoming data to detect errors. Monitor the RXRUNDISP port to see the current running disparity.

Disparity Errors and Not-in-Table Errors

The decoder drives RXDISPERR High when RXDATA arrives with the wrong disparity. In addition to disparity errors, the 8B/10B decoder detects illegal 10-bit codes (out-of-table errors). The decoder drives the RXNOTINTABLE port High when RXDATA is not a valid 8B/10B character. In this case, the aligned and not decoded data is present on RXDATA, RXCHARISK, and RXDISPERR as illustrated in Figure 7-36, page 202. This behavior is identical to disabling the decoder for bypassing.

Apotex Ex1009
Page 174 of 347

Figure 7-19 shows a waveform with a few error bytes arriving on RXDATA and the RXNOTINTABLE and RXDISPERR ports indicating the error.

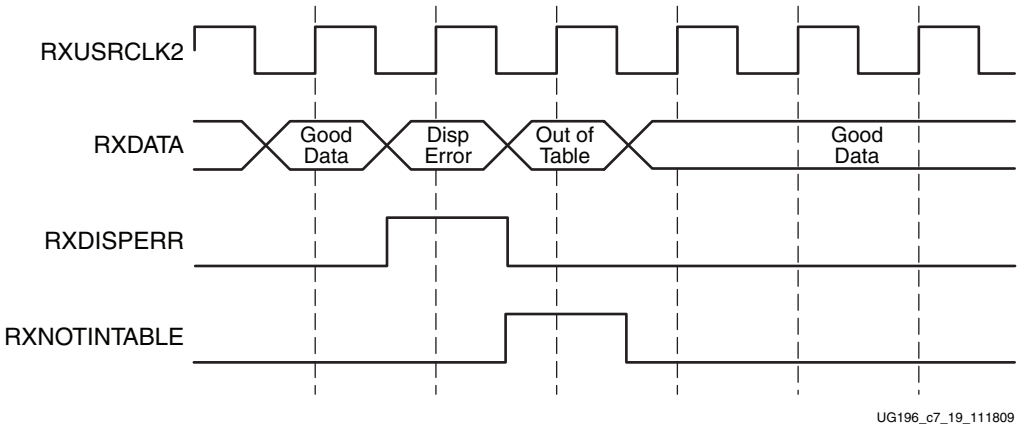


Figure 7-19: **RX Data with 8B/10B Errors**

Configurable RX Elastic Buffer and Phase Alignment

Overview

The GTP RX datapath has two internal parallel clock domains used in the PCS: the PMA parallel clock domain (XCLK) and the RXUSRCLK domain. To receive data, the PMA parallel rate must be sufficiently close to the RXUSRCLK rate, and all phase differences between the two domains must be resolved. Figure 7-20 shows the two parallel clock domains, XCLK and RXUSRCLK.

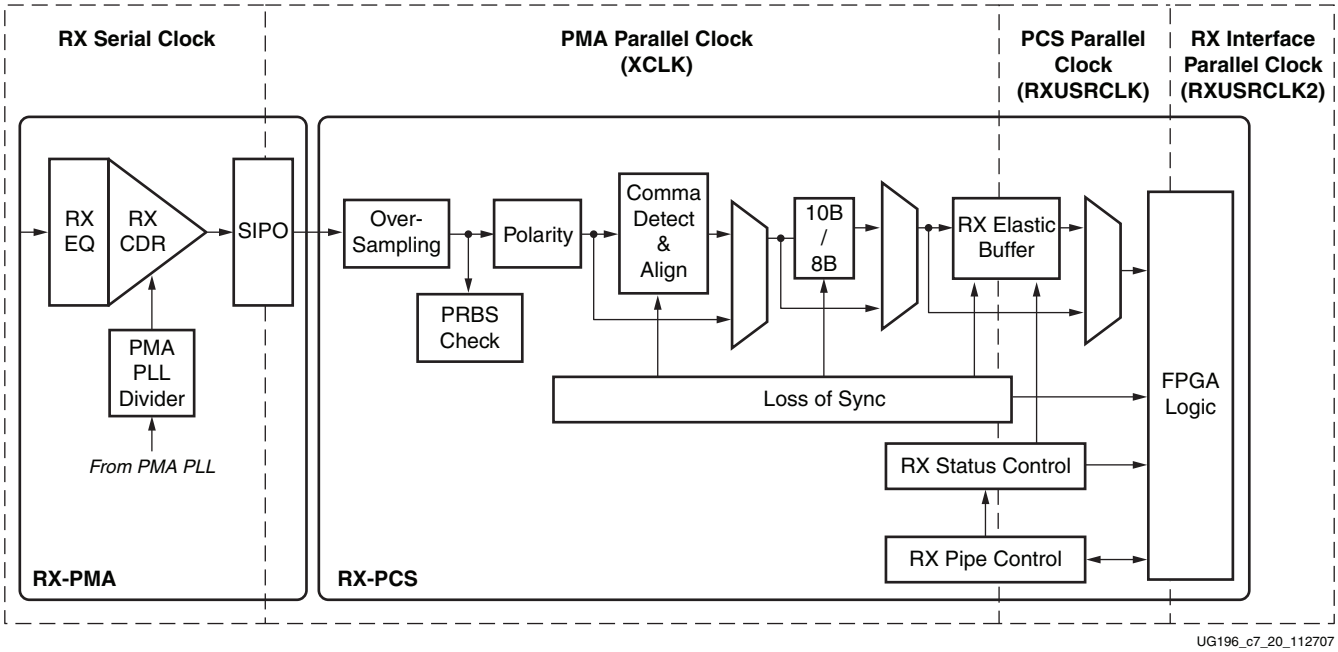


Figure 7-20: **Receiver Parallel Clock Domains**

The GTP transceiver includes an RX elastic buffer to resolve differences between the XCLK and RXUSRCLK domains. The phase of the two domains can also be matched by using the recovered clock from the transceiver to drive RXUSRCLK and adjusting its phase to match XCLK. All RX datapaths must use one of these approaches. The costs and benefits of each approach are shown in Table 7-26.

Cypher Ex1009
Page 176 of 347

Table 7-26: **Buffering vs. Phase Alignment**

	RX Elastic Buffer	RX Phase Alignment
Clocking Options	Can use recovered clock or local clock (with clock correction)	Must use recovered clock
Initialization	Works immediately	Must wait for all clocks to stabilize then perform alignment procedure
Latency	Buffer latency depends on features used (clock correction and channel bonding)	Lower latency than using the RX buffer(1, 2)
Clock Correction/ Channel Bonding	Required for clock correction/channel bonding	
Internal Data Width	Can be 8 or 10 bits wide	Must be 10 bits wide

Notes:

1. Bypassing the RX buffer is an advanced feature. RX buffer bypass can operate only under certain system-level conditions and data rates.
2. When the buffer is bypassed, RXUSRCLK must be driven directly from RXRECCLK using a BUFR regional clock buffer to provide maximum system margin across system conditions.

The RX elastic buffer is also used for clock correction (see Configurable Clock Correction, page 183) and channel bonding (see Configurable Channel Bonding (Lane Deskew), page 189). Clock correction is used in cases where XCLK and RXUSRCLK are not frequency matched. Table 7-27 lists common clock configurations and shows whether they require clock correction.

Table 7-27: **Common Clock Configurations**

	Needs Clock Correction?
Synchronous System (both sides use same physical oscillator for REFCLK)	No
Separate Reference Clocks, RX uses RXRECCLK	No
Separate Reference Clocks, RX uses Local Clock	Yes

Ports and Attributes

Table 7-28 defines the RX elastic buffer and phase-alignment ports.

Table 7-28: **RX Elastic Buffer and Phase-Alignment Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the width of the internal datapath for the entire GTP_DUAL tile. `0`: Internal datapath is 8 bits wide `1`: Internal datapath is 10 bits wide
RXBUFRESET0 RXBUFRESET1	In	Async	Resets the RX elastic buffer logic and re-initializes the RX elastic buffer.

Table 7-28: **RX Elastic Buffer and Phase-Alignment Ports *(Continued)***

Port	Dir	Clock Domain	Description
RXBUFSTATUS0[2:0] RXBUFSTATUS1[2:0]	Out	RXUSRCLK2	Indicates the status of the RX elastic buffer as follows: `000`: Nominal condition `001`: Number of bytes in buffer are less than CLK_COR_MIN_LAT `010`: Number of bytes in buffer are greater than CLK_COR_MAX_LAT `101`: RX elastic buffer underflow(1) `110`: RX elastic buffer overflow(1)
RXPMASETPHASE0 RXPMASETPHASE1	In	RXUSRCLK2	Used to align the XCLK and RXUSRCLK domains when RXUSRCLK is driven by RXRECCLK. Allows the RX elastic buffer to be bypassed.

Notes:

1. If an RX elastic buffer overflow or an RX elastic buffer underflow condition occurs, the content of the RX elastic buffer becomes invalid, and the RX elastic buffer needs re-initialization by asserting RXBUFRESET.

Table 7-29 defines the RX elastic buffer and phase-alignment attributes.

Table 7-29: **RX Elastic Buffer and Phase-Alignment Attributes**

Attribute	Description
OVERSAMPLE_MODE	Enables built-in 5x digital oversampling. Applies to both transceivers in the GTP_DUAL tile. TRUE: Built-in 5x oversampling enabled FALSE: Built-in 5x oversampling disabled TX_BUFFER_USE must be TRUE when OVERSAMPLE_MODE is TRUE. See Oversampling, page 157 for the remaining configuration steps required to use oversampling.
RX_BUFFER_USE_0(1) RX_BUFFER_USE_1(1)	Use or bypass the RX elastic buffer. TRUE: Use the RX elastic buffer (normal mode). FALSE: Permanently bypass the RX elastic buffer. If OVERSAMPLE_MODE is FALSE, RX phase alignment must be used whenever the RX elastic buffer is bypassed.
RX_XCLK_SEL_0 RX_XCLK_SEL_1	Selects which clock is used to drive the RX parallel clock domain (XCLK). "RXREC": (default) XCLK domain driven by the recovered clock from CDR. When OVERSAMPLE_MODE is TRUE, the recovered clock is sourced from the oversampling block. "RXUSR": The RXUSRCLK port drives the RX parallel clock domain. Use this mode when bypassing the RX elastic buffer.

Notes:

1. When the RX elastic buffer is bypassed, 10-bit internal data width is necessary. Therefore, INTDATAWIDTH is High.

Description

Using the RX Elastic Buffer

Figure 7-21 shows how the RX elastic buffer bridges the PMA parallel clock domain (XCLK) and the PCS parallel clock domain (RXUSRCLK) in the RX datapath. This bridging is necessary because there is no guaranteed phase relationship between the parallel clock from the SIPO (XCLK) and the parallel clocks from the FPGA logic (RXUSRCLK and RXUSRCLK2).

Cpuver Ex1009
Page 178 of 347

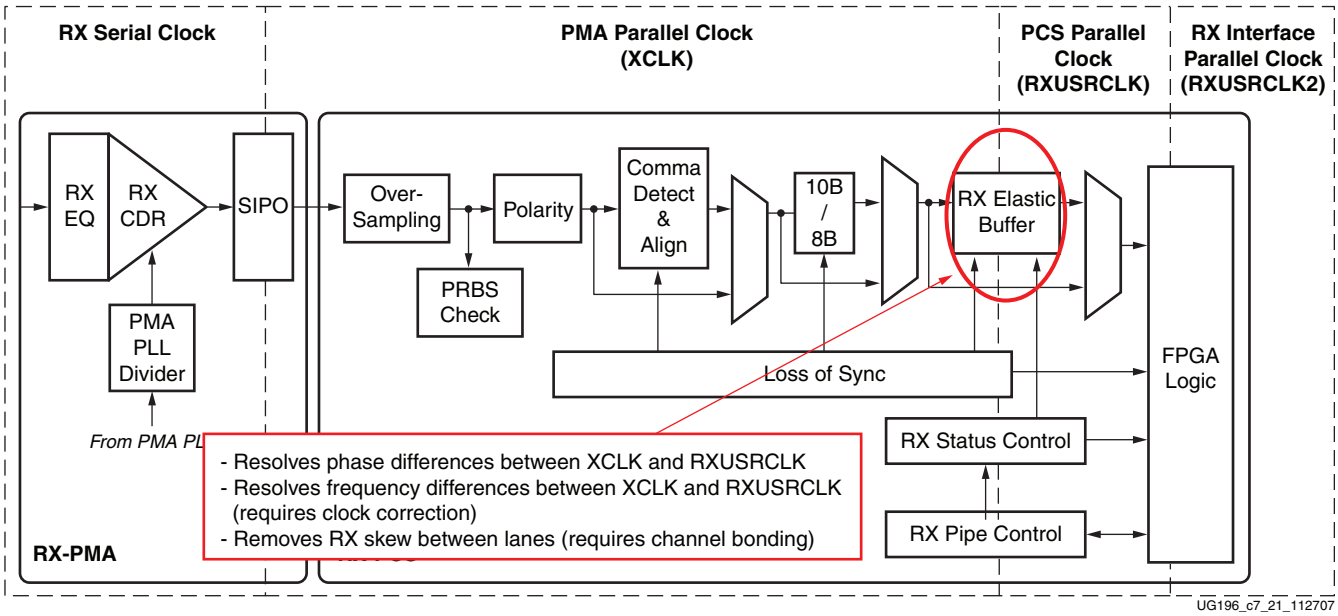


Figure 7-21: **Using the RX Elastic Buffer**

The RX elastic buffer can also be used when OVERSAMPLE_MODE is TRUE.

To use the RX elastic buffer to resolve phase differences between the domains:

- Set RX_BUFFER_USE to TRUE.
- Reset the buffer whenever RXBUFSTATUS indicates an overflow or an underflow.
- The buffer can be reset using GTPRESET (see Reset, page 84), RXRESET, or RXBUFRESET.

Using RX Phase Alignment to Bypass the RX Elastic Buffer

Bypassing the RX buffer is an advanced feature. RX buffer bypass can operate only under certain system-level conditions and data rates. When the buffer is bypassed, RXUSRCLK must be driven directly from RXRECCLK using a BUFR regional clock buffer to achieve maximum system margin across system conditions.

The RX elastic buffer can be bypassed to reduce latency when INTDATAWIDTH is High (10-bit internal datapath width) and RXRECCLK is used to source RXUSRCLK and RXUSRCLK2. When the RX elastic buffer is bypassed, latency through the RX datapath is low and deterministic, but clock correction and channel bonding are not available.

Figure 7-22 shows how phase alignment allows the RX elastic buffer to be bypassed. Before phase alignment, there is no guaranteed phase relationship between the parallel clock generated from the recovered clock in the CDR circuit (XCLK) and the parallel clocks from the FPGA logic (RXUSRCLK and RXUSRCLK2). Phase alignment causes RXRECCLK from the CDR to be adjusted so that there is no significant phase difference between XCLK and RXUSRCLK.

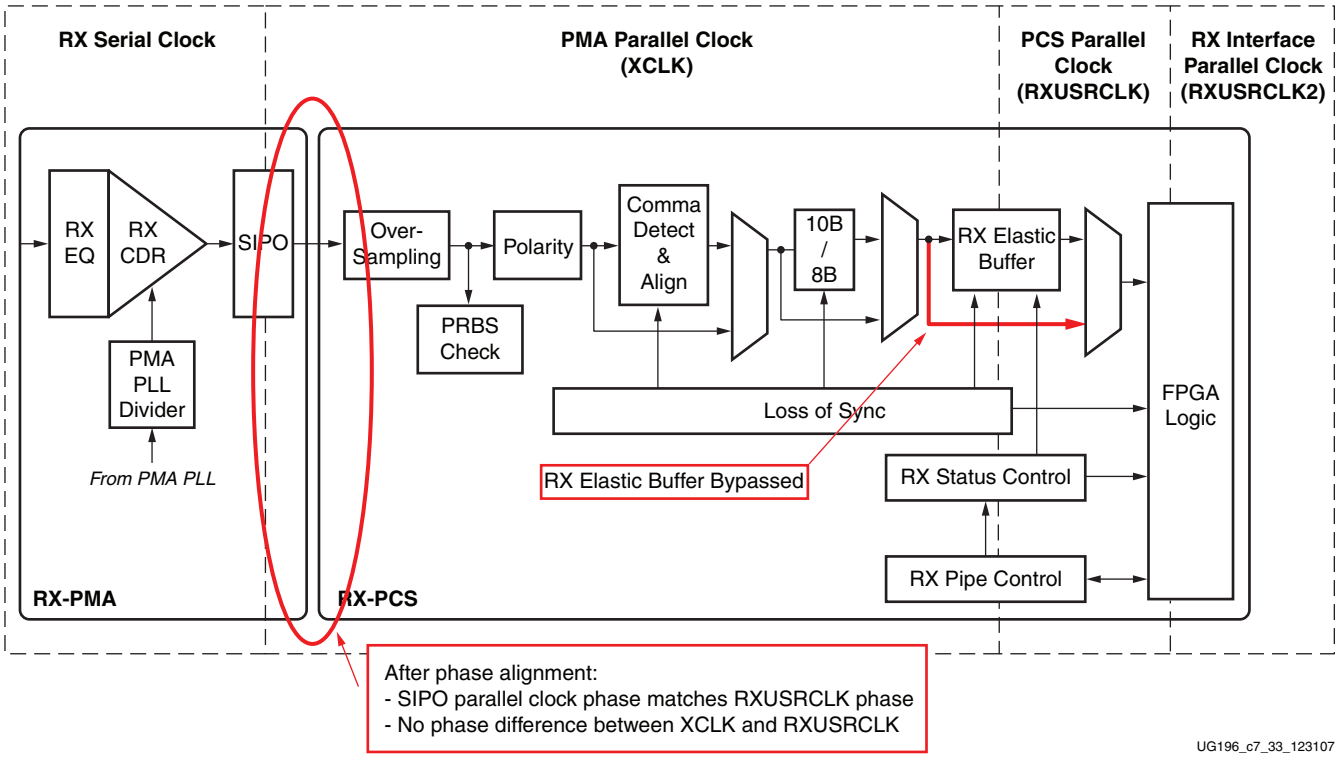


Figure 7-22: **Using Phase Alignment**

Phase alignment can only be used for 10-bit internal datapaths (INTDATAWIDTH is High).

To use RX phase alignment:

1. Set RX_BUFFER_USE to FALSE to bypass the RX elastic buffer.
2. Set RX_XCLK_SEL to *RXUSR*.
3. Source RXUSRCLK and RXUSRCLK2 with the RXRECCLK output. Divide RXRECCLK by 2 if necessary to provide RXUSRCLK2 (see FPGA RX Interface, page 200 for details).
4. Reset the RX datapath using GTPRESET or one of the CDR resets.
5. Wait for the shared PMA PLL and any DCM or PLL used for RXUSRCLK2 to lock.
6. Wait for the CDR to lock and provide a stable RXRECCLK.
7. Drive RXPMASETPHASE High for 32 RXUSRCLK2 cycles and then deassert it.

Step 6 requires careful guidance. Normally, CDR lock is detected by measuring the quality of incoming data. Methods for detecting CDR lock include:

- Finding known data in the incoming data stream (for example, commas or A1/A2 framing characters). In general, several consecutive known data patterns should be received without error to indicate a CDR lock.
- Using the LOS state machine (see Configurable Loss-of-Sync State Machine, page 169). If incoming data is 8B/10B encoded and the CDR is locked, the LOS state machine should move to the SYNC_ACQUIRED state and stay there.

When the RX elastic buffer is bypassed, data received from the PMA might be distorted due to phase differences as it passes to the PCS. This makes it difficult to determine

Power Ex1009
Page 180 of 347

whether or not bad data is received because the CDR is not locked, or the CDR *is* locked and the phase alignment has not yet been attempted. To work around this problem, RX phase alignment must be attempted several times and the output data evaluated after each attempt. Good data is received if the phase is aligned while the RX CDR is locked.

The flow diagram in Figure 7-23 shows the series of steps required for successful RX phase alignment. Any number of clock cycles can be used for the CDR lock time, but using a larger number decreases the number of cycles through the states.

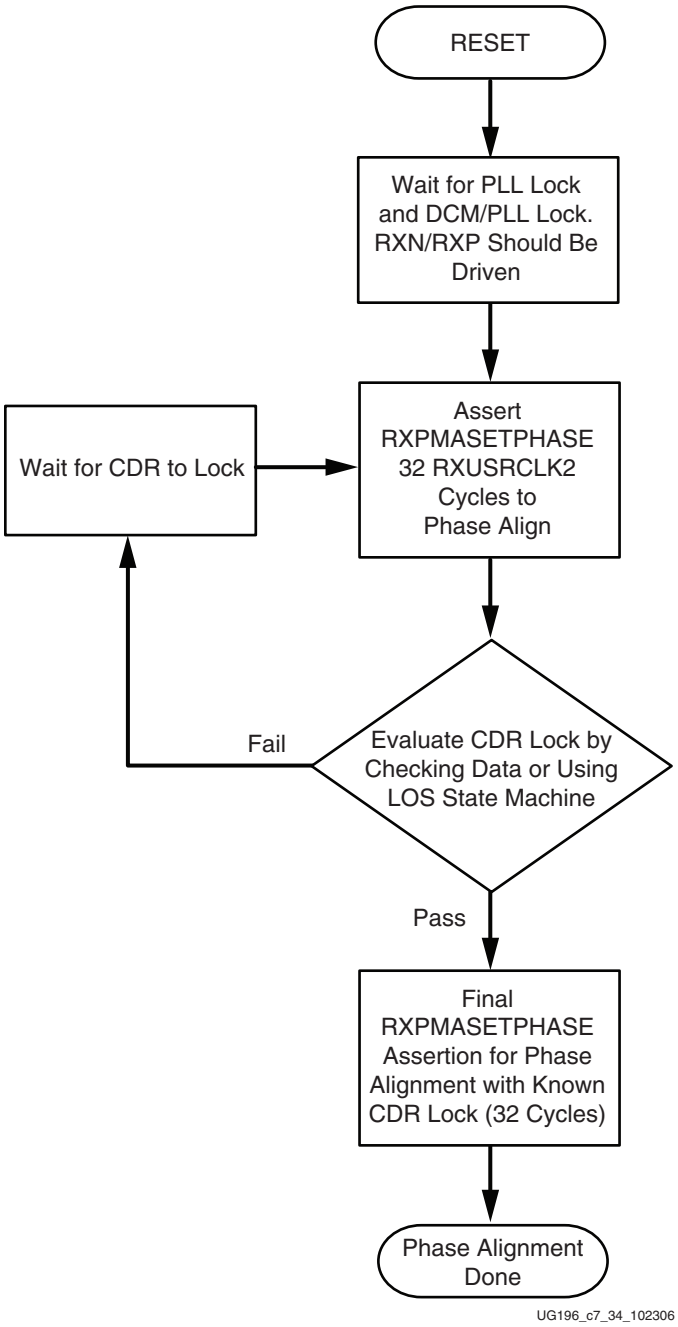


Figure 7-23: **Steps Required for Successful RX Phase Alignment**

Bypassing the RX Elastic Buffer while Using Built-in Oversampling

If OVERSAMPLE_MODE is TRUE to activate built-in oversampling, the RX elastic buffer is bypassed without the use of phase alignment. Instead, a shallow buffer inside the oversampling block is used to resolve phase differences between RXUSRCLK and RXRECCLK. See Oversampling, page 157 for information about monitoring the status of the oversampling block.

Figure 7-24 shows how the RX elastic buffer is bypassed with oversampling enabled. The shallow buffer in the oversampling block resolves any phase differences between RXUSRCLK and the recovered clock it generates.

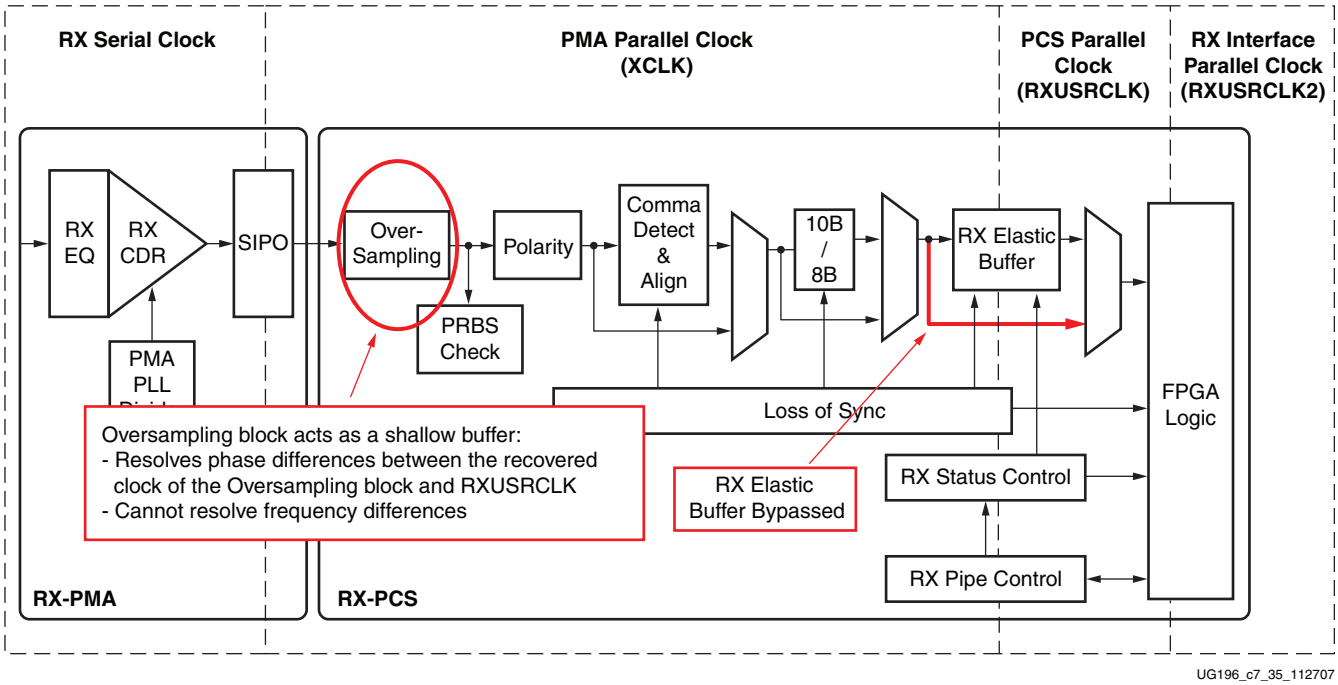


Figure 7-24: **Buffer Bypass with Oversampling Enabled**

To bypass the RX elastic buffer when OVERSAMPLING_MODE is TRUE:

1. Set RX_BUFFER_USE to FALSE to bypass the RX elastic buffer (optional).
2. Set RX_XCLK_SEL to *RXUSR*.
3. Source RXUSRCLK and RXUSRCLK2 with the RXRECCLK output. Divide RXRECCLK by 2 if necessary to provide RXUSRCLK2 (see FPGA RX Interface, page 200 for details).

Apotex Ex1009
Page 182 of 347

Configurable Clock Correction

Overview

The RX elastic buffer has an additional benefit: it can tolerate frequency differences between the XCLK and RXUSRCLK domains by performing clock correction. Clock correction actively prevents the RX elastic buffer from getting too full or too empty by deleting or replicating special idle characters in the data stream.

Figure 7-25 shows a conceptual view of clock correction.

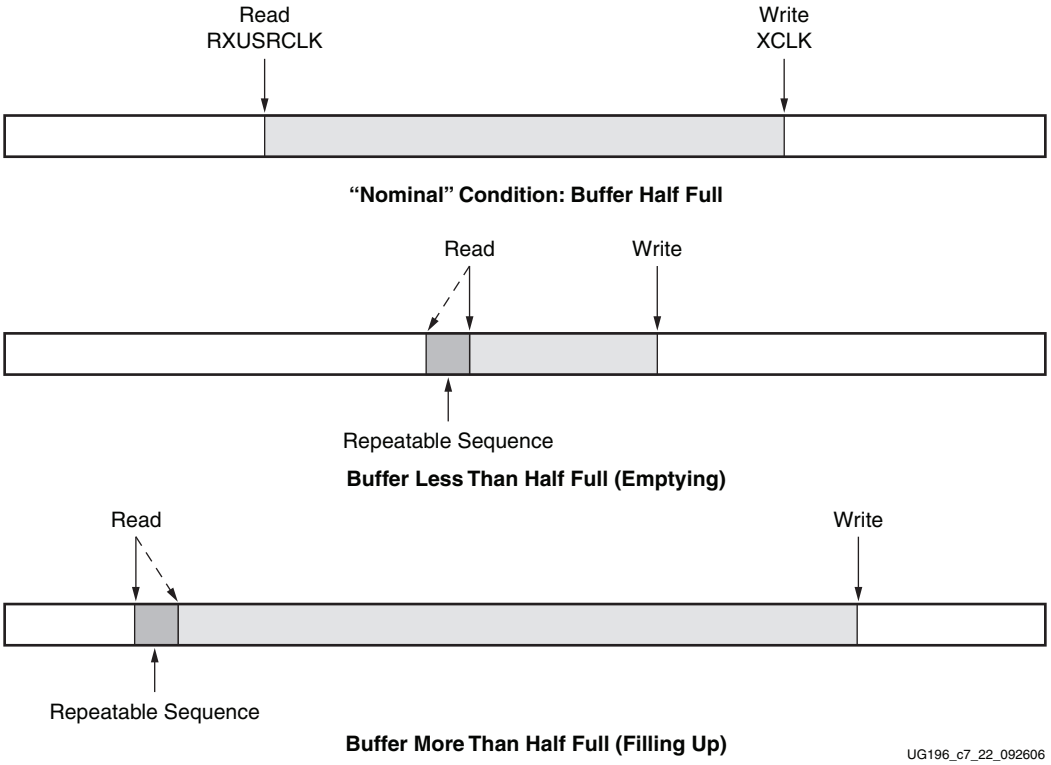


Figure 7-25: **Clock Correction**

Clock correction should be used whenever there is a frequency difference between XCLK and RXUSRCLK. It can be avoided by using the same frequency source for TX and RX, or by using the recovered clock to drive RXUSRCLK. The Configurable RX Elastic Buffer and Phase Alignment section has more details about the steps required if clock correction is not used.

Ports and Attributes

Table 7-30 defines the clock correction ports.

Table 7-30: **Clock Correction Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the bit width for the TX and RX internal datapaths. This port controls both transceivers on the tile: `0`: 8-bit width `1`: 10-bit width(1)
RXBUFRESET0 RXBUFRESET1	In	Async	Resets the RX elastic buffer logic and re-initializes the RX elastic buffer.
RXBUFSTATUS0[2:0] RXBUFSTATUS1[2:0]	Out	RXUSRCLK2	Indicates the status of the RX elastic buffer as follows: `000`: Nominal condition `001`: Number of bytes in buffer is less than CLK_COR_MIN_LAT `010`: Number of bytes in buffer is greater than CLK_COR_MAX_LAT `101`: RX elastic buffer underflow(2) `110`: RX elastic buffer overflow(2)
RXCLKCORCNT0[2:0] RXCLKCORCNT1[2:0]	Out	RXUSRCLK2	Reports the clock correction status of the RX elastic buffer: `000`: No clock correction `001`: 1 sequence skipped `010`: 2 sequences skipped `011`: 3 sequences skipped `100`: 4 sequences skipped `101`: Reserved `110`: 2 sequences added `111`: 1 sequence added

Notes:

1. 10-bit internal data width is necessary when the RX elastic buffer is bypassed.
2. If an RX elastic buffer overflow or an RX elastic buffer underflow condition occurs, the content of the RX elastic buffer becomes invalid, and the RX elastic buffer needs re-initialization by asserting RXBUFRESET.

Table 7-31 defines the clock correction attributes.

Table 7-31: **Clock Correction Attributes**

Attribute	Description
CLK_CORRECT_USE_0 CLK_CORRECT_USE_1	Enables clock correction. FALSE: Clock correction disabled TRUE: Clock correction enabled
CLK_COR_ADJ_LEN_0 CLK_COR_ADJ_LEN_1	This attribute defines the size of the adjustment (number of bytes repeated or skipped) in a clock correction. The bytes skipped or repeated always start from the beginning of the clock correction sequence to allow more bytes to be replaced than in the specified clock correction sequence. Valid lengths are from one to four bytes.
CLK_COR_DET_LEN_0 CLK_COR_DET_LEN_1	This attribute defines the length of the sequence that the transceiver matches to detect opportunities for clock correction. Valid lengths are from one to four bytes.
CLK_COR_INSERT_IDLE_FLAG_0 CLK_COR_INSERT_IDLE_FLAG_1	Controls whether or not the RXRUNDISP input status indicates running disparity or inserted-idle (clock correction sequence) flag. FALSE: RXRUNDISP indicates running disparity when RXDATA is decoded data. TRUE: RXRUNDISP is raised for the first byte of each inserted (repeated) clock correction (Idle) sequence (when RXDATA is decoded data).
CLK_COR_KEEP_IDLE_0 CLK_COR_KEEP_IDLE_1	Controls whether or not the RX elastic buffer must retain at least one clock correction sequence in the byte stream. FALSE: Transceiver can remove all clock correction sequences to further re-center the RX elastic buffer during clock correction. TRUE: In the final RXDATA stream, the transceiver must leave at least one clock correction sequence per continuous stream of clock correction sequences.
CLK_COR_MAX_LAT_0 CLK_COR_MAX_LAT_1	Specifies the maximum RX elastic buffer latency. If the RX elastic buffer exceeds CLK_COR_MAX_LAT, the clock correction circuit removes incoming clock correction sequences to prevent overflow. Valid values for this attribute range from 3 to 48.
CLK_COR_MIN_LAT_0 CLK_COR_MIN_LAT_1	Specifies the minimum RX elastic buffer latency. If the RX elastic buffer drops below CLK_COR_MIN_LAT, the clock correction circuit replicates incoming clock correction sequences to prevent underflow. When the RX elastic buffer is reset, its pointers are set so there are CLK_COR_MIN_LAT unread (and uninitialized) data bytes in the buffer. Valid values for this attribute range from 3 to 48.
CLK_COR_PRECEDENCE_0 CLK_COR_PRECEDENCE_1	Determines whether clock correction or channel bonding takes precedence when both operations are triggered at the same time. TRUE: Clock correction takes precedence over channel bonding if there is opportunity for both FALSE: Channel bonding takes precedence over clock correction if there is opportunity for both
CLK_COR_REPEAT_WAIT_0 CLK_COR_REPEAT_WAIT_1	This attribute specifies the minimum number of RXUSRCLK cycles without clock correction that must occur between successive clock corrections. If this attribute is zero, no limit is placed on how frequently clock correction can occur. Valid values for this attribute range from 0 to 31.

Optiver Ex1009
Page 185 of 347

Table 7-31: **Clock Correction Attributes *(Continued)***

Attribute	Description
CLK_COR_SEQ_1_1_0 CLK_COR_SEQ_1_1_1	The CLK_COR_SEQ_1 attributes are used in conjunction with CLK_COR_SEQ_1_ENABLE to define clock correction sequence 1. The sequence is made up of four subsequences. Each subsequence is 10 bits long. The rules for setting the subsequences depend on INTDATAWIDTH and RX_DECODE_SEQ_MATCH. See the Description section to learn how to set clock correction subsequences. Not all subsequences need to be used. CLK_COR_DET_LEN determines how many of the sequences are used for a match. If CLK_COR_DET_LEN = 1, only CLK_COR_SEQ_1_1 is used. CLK_COR_SEQ_1_ENABLE can be used to make parts of the sequence don't cares. If CLK_COR_SEQ_1_ENABLE[k] is 0, CLK_COR_SEQ_1_k is a don't care subsequence and is always a match.
CLK_COR_SEQ_1_2_0 CLK_COR_SEQ_1_2_1	
CLK_COR_SEQ_1_3_0 CLK_COR_SEQ_1_3_1	
CLK_COR_SEQ_1_4_0 CLK_COR_SEQ_1_4_1	
CLK_COR_SEQ_1_ENABLE_0 CLK_COR_SEQ_1_ENABLE_1	
CLK_COR_SEQ_2_1_0 CLK_COR_SEQ_2_1_1	The CLK_COR_SEQ_2 attributes are used in conjunction with CLK_COR_SEQ_2_ENABLE to define the second clock correction sequence. This second sequence is used as an alternate sequence for clock correction when CLK_COR_SEQ_2_USE is TRUE: if either sequence 1 or sequence 2 arrives, clock correction is performed. The sequence is made up of four subsequences. Each subsequence is 10 bits long. The rules for setting the subsequences depend on INTDATAWIDTH and RX_DECODE_SEQ_MATCH. See the Description section to learn how to set clock correction subsequences. Not all subsequences need to be used. CLK_COR_DET_LEN determines how much of the sequence is used for a match. If CLK_COR_DET_LEN = 1, only CLK_COR_SEQ_2_1 is used. CLK_COR_SEQ_2_ENABLE can be used to make parts of the sequence don't care. If CLK_COR_SEQ_2_ENABLE[k] is 0, CLK_COR_SEQ_2_k is a don't care byte subsequence and is always a match.
CLK_COR_SEQ_2_2_0 CLK_COR_SEQ_2_2_1	
CLK_COR_SEQ_2_3_0 CLK_COR_SEQ_2_3_1	
CLK_COR_SEQ_2_4_0 CLK_COR_SEQ_2_4_1	
CLK_COR_SEQ_2_ENABLE_0 CLK_COR_SEQ_2_ENABLE_1	
CLK_COR_SEQ_2_USE_0 CLK_COR_SEQ_2_USE_1	Determines if the second clock correction sequence is to be used. When set to TRUE, the second clock correction sequence also triggers clock correction.
RX_DECODE_SEQ_MATCH_0 RX_DECODE_SEQ_MATCH_1	Determines whether sequences are matched against the input to the 8B/10B decoder or the output. Used for the clock correction circuit and the channel bonding circuit. TRUE: Sequences are matched against the output of the 8B/10B decoder. K characters and disparity information are used. Bit ordering of the 8B/10B output is used. FALSE: Sequences are matched against unencoded data. Bit ordering is as for an undecoded parallel interface.

Apotex Ex1009
Page 186 of 347

Description

Enabling Clock Correction

Each GTP transceiver includes a clock correction circuit that performs clock correction by controlling the pointers of the RX elastic buffer. To use clock correction, RX_BUFFER_USE is set to TRUE to turn on the RX elastic buffer, and CLK_CORRECT_USE is set to TRUE to turn on the clock correction circuit.

Clock correction is triggered when the RX elastic buffer latency is too high or too low, and the clock correction circuit detects a match sequence. To use clock correction, the clock correction circuit must be configured to set the following items:

- RX elastic buffer limits
- Clock correction sequence

Setting RX Elastic Buffer Limits

The RX elastic buffer limits are set using CLK_COR_MIN_LAT (minimum latency) and CLK_COR_MAX_LAT (maximum latency). When the number of bytes in the RX elastic buffer drops below CLK_COR_MIN_LAT, the clock correction circuit writes an additional CLK_COR_ADJ_LEN bytes from the first clock correction sequence it matches to prevent the buffer from underflowing. Similarly, when the number of bytes in the RX elastic buffer exceeds CLK_COR_MAX_LAT, the clock correction circuit deletes CLK_COR_ADJ_LEN bytes from the first clock correction sequence it matches, starting with the first byte of the sequence.

Setting Clock Correction Sequences

The clock correction sequences are programmed using the CLK_COR_SEQ_1_* attributes and CLK_COR_ADJ_LEN. Each CLK_COR_SEQ_1_* attribute corresponds to one subsequence in clock correction sequence 1. CLK_COR_ADJ_LEN is used to set the number of subsequences to be matched. If INTDATAWIDTH is High (10-bit internal datapath), the clock correction circuit matches all 10 bits of each subsequence. If INTDATAWIDTH is Low (8-bit internal datapath), only the right-most eight bits of each subsequence are used.

A second, alternate clock correction sequence can be activated by setting CLK_COR_SEQ_2_USE to TRUE. The first and second sequences share length settings, but use different subsequence values for matching. Set the CLK_COR_SEQ_2_* attributes to define the subsequence values for the second sequence.

When using 8B/10B decoding (RXDEC8B10BUSE and INTDATAWIDTH are High), RX_DECODE_SEQ_MATCH is set to TRUE to search the output of the 8B/10B decoder for sequence matches instead of non-decoded data. This allows the circuit to look for 8-bit values with either positive or negative disparity, and to distinguish K characters from regular characters (see Configurable 8B/10B Encoder, page 111 and Configurable 8B/10B Decoder, page 172 for details). Figure 7-26 shows how to set a clock correction sequence byte when RX_DECODE_SEQ_MATCH is TRUE.

When RX_DECODE_SEQ_MATCH is FALSE, the sequence must exactly match non-decoded incoming data. The bit order of the data matches the bit order shown in Figure 7-35 and Figure 7-36 for RXDATA with no 8B/10B decoding.

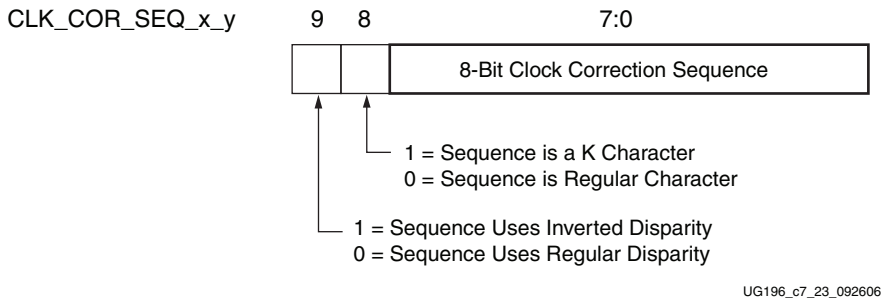


Figure 7-26: **Clock Correction Subsequence Settings with RX_DECODE_SEQ_MATCH = TRUE**

Some protocols use clock correction sequences with *don't care* subsequences. The clock correction circuit can be programmed to recognize these sequences using CLK_COR_SEQ_1_ENABLE and CLK_COR_SEQ_2_ENABLE. When the enable bit for a sequence is Low, that byte is independent from the value, always a match. Figure 7-27 shows the mapping between the clock correction sequences and the clock correction sequence enable bits.

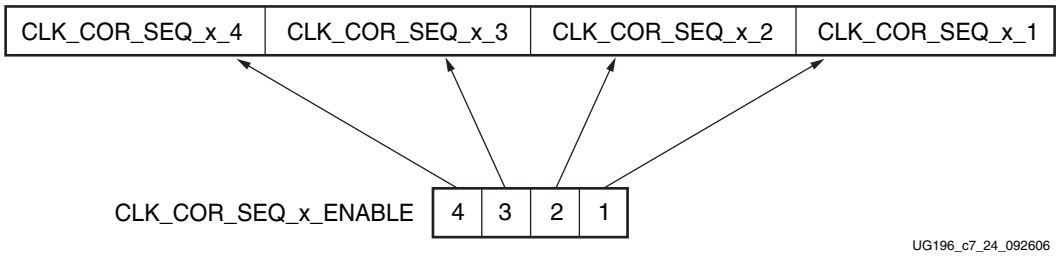


Figure 7-27: **Clock Correction Sequence Mapping**

Clock Correction Options

CLK_COR_REPEAT_WAIT is used to control the clock correction frequency. This value is set to the minimum number of RXUSRCLK cycles required between clock correction events. This attribute is set to 0 to allow clock correction to occur any time.

Some protocols allow clock correction to occur at any time, but require that if the clock correction circuit removes sequences, it should leave at least one sequence in the stream. For protocols with this requirement, CLK_COR_KEEP_IDLE is set to TRUE.

Monitoring Clock Correction

The clock correction circuit can be monitored using the RXCLKCORCNT and RXBUFSTATUS ports. The RXCLKCORCNT entry in Table 7-30, page 184 shows how to decode the values of RXCLKCORCNT to determine the status of the clock correction circuit. The RXBUFSTATUS entry in Table 7-30 shows how to decode the values of RXBUFSTATUS to determine how full the RX elastic buffer is.

In addition to RXCLKCORCNT and RXBUFSTATUS, RXRUNDISP can be taken from the 8B/10B decoder interface (see Configurable 8B/10B Decoder, page 172) and used to indicate when RXDATA has the first byte of a clock correction sequence that was replicated and added to the RX elastic buffer. To use the RXRUNDISP port to indicate inserted idles instead of the current RX running disparity, CLK_COR_INSERT_IDLE_FLAG is set to TRUE.

Petitioner Ex1009
Page 188 of 347

Configurable Channel Bonding (Lane Deskew)

Overview

The RX elastic buffer can also be used for channel bonding. Channel bonding cancels out the skew between GTP lanes by using the RX elastic buffer as a variable latency block. The transmitter sends a pattern simultaneously on all lanes, which the channel bonding circuit uses to set the latency for each lane so that data is presented without skew at the FPGA RX interface.

Figure 7-28 shows a conceptual view of channel bonding.

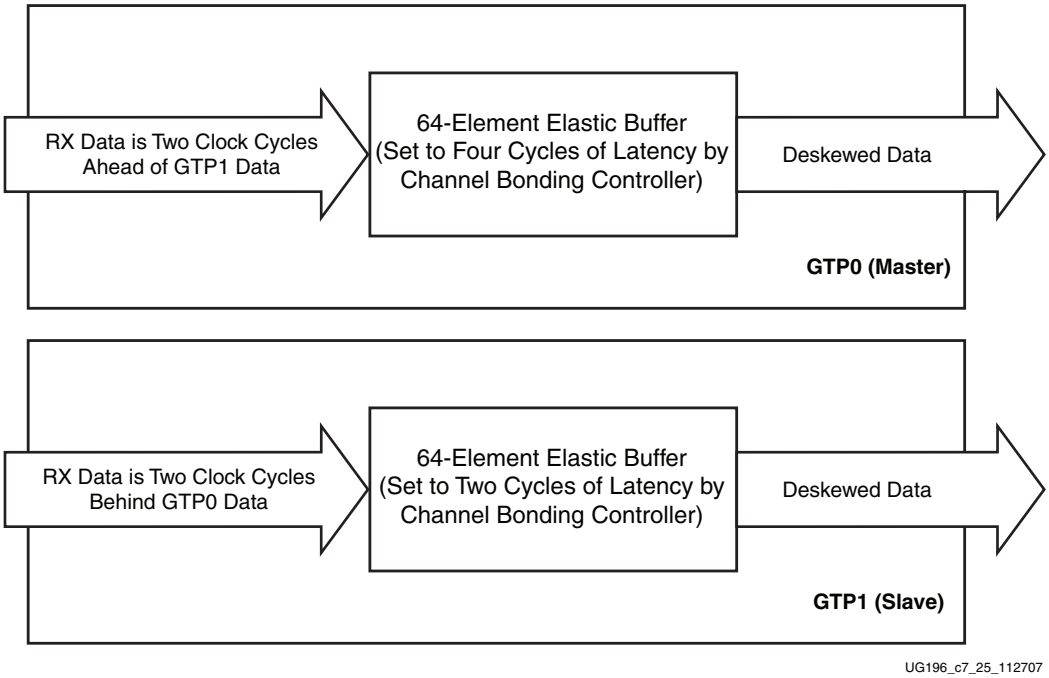


Figure 7-28: **Channel Bonding Conceptual View**

Ports and Attributes

Table 7-32 defines the channel bonding ports.

Table 7-32: **Channel Bonding Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the bit width for the TX and RX internal datapaths. This port controls both transceivers on the tile. 0: 8-bit width 1: 10-bit width(1)
RXCHANBONDSEQ0 RXCHANBONDSEQ1	Out	RXUSRCLK2	Goes High when RXDATA contains the start of a channel bonding sequence.
RXCHANISALIGNED0 RXCHANISALIGNED1	Out	RXUSRCLK2	This signal from the RX elastic buffer is High to indicate that the channel is properly aligned with the master transceiver according to observed channel bonding sequences in the data stream. This signal goes Low if an unaligned channel bonding sequence is detected, indicating that channel alignment was lost.
RXCHANREALIGN0 RXCHANREALIGN1	Out	RXUSRCLK2	This signal from the RX elastic buffer is held High for at least one cycle when the receiver has changed the alignment between this transceiver and the master.
RXCHBONDI0[2:0] RXCHBONDI1[2:0]	In	RXUSRCLK	FPGA channel bonding control. This signal is used only by *slaves*. It is driven from another transceiver's RXCHBONDO port that is the *master* in this configuration.
RXCHBONDO0[2:0] RXCHBONDO1[2:0]	Out	RXUSRCLK	FPGA channel bonding control. This signal is used by the *master* and *slaves* to pass channel bonding and clock correction control to other transceivers' RXCHBONDI ports.
RXENCHANSYNC0 RXENCHANSYNC1	In	RXUSRCLK2	Enable channel bonding (from the FPGA to the *master*). Tie this port High for *slaves*.

Notes:

1. 10-bit internal data width is necessary when the RX elastic buffer is bypassed, the PRBS generation/detection is used, or both.

Table 7-33 defines the channel bonding attributes.

Table 7-33: **Channel Bonding Attributes**

Attribute	Description
CHAN_BOND_1_MAX_SKEW_0 CHAN_BOND_1_MAX_SKEW_1	These attributes control the number of USRCLK cycles that the master waits before ordering the slaves to execute channel bonding. This attribute determines the maximum skew that can be handled by channel bonding. It must always be less than the minimum distance (in bytes or 10-bit codes) between channel bonding sequences. Valid values range from 1 to 14.
CHAN_BOND_2_MAX_SKEW_0 CHAN_BOND_2_MAX_SKEW_1	

Apotex Ex1009
Page 190 of 347

Table 7-33: **Channel Bonding Attributes *(Continued)***

Attribute	Description
CHAN_BOND_LEVEL_0 CHAN_BOND_LEVEL_1	CHAN_BOND_LEVEL indicates the amount of internal pipelining used for the RX elastic buffer control signals. A higher value permits more daisy-chaining of RXCHBONDO and RXCHBONDI to ease placement and routing constraints. To minimize required latency through the RX elastic buffer, CHAN_BOND_LEVEL in the master should be set to the smallest value possible for the required amount of daisy-chaining. See the Description section for channel bonding to learn how to set the channel bonding level.
CHAN_BOND_MODE_0 CHAN_BOND_MODE_1	Defines the channel bonding mode of operation for this transceiver. OFF: No channel bonding. MASTER: This transceiver is master for channel bonding. Its RXCHBONDO port directly drives RXCHBONDI ports on one or more SLAVE transceivers. SLAVE: This transceiver is a slave for channel bonding. Its RXCHBONDI port is directly driven by the RXCHBONDO port of another SLAVE or MASTER transceiver. If its CHAN_BOND_LEVEL setting is greater than 0, its RXCHBONDO port may directly drive RXCHBONDI ports on one or more other SLAVE transceivers.
CHAN_BOND_SEQ_1_1_0 CHAN_BOND_SEQ_1_1_1	The CHAN_BOND_SEQ_1 attributes are used in conjunction with CHAN_BOND_SEQ_1_ENABLE to define channel bonding sequence 1. Each subsequence is 10 bits long. The rules for setting the subsequences depend on INTDATAWIDTH and RX_DECODE_SEQ_MATCH. See the Description section to learn how to set channel bonding subsequences. Not all subsequences need to be used. CHAN_BOND_SEQ_LEN determines how much of the sequence is used for a match. If CHAN_BOND_SEQ_LEN = `1`, only CHAN_BOND_SEQ_1_1 is used. CHAN_BOND_SEQ_1_ENABLE can be used to make parts of the sequence don't cares. If CHAN_BOND_SEQ_1_ENABLE[k] is `0`, CHAN_BOND_SEQ_1_k is a don't-care subsequence and is always a match.
CHAN_BOND_SEQ_1_2_0 CHAN_BOND_SEQ_1_2_1	
CHAN_BOND_SEQ_1_3_0 CHAN_BOND_SEQ_1_3_1	
CHAN_BOND_SEQ_1_4_0 CHAN_BOND_SEQ_1_4_1	
CHAN_BOND_SEQ_1_ENABLE_0 CHAN_BOND_SEQ_1_ENABLE_1	
CHAN_BOND_SEQ_2_1_0 CHAN_BOND_SEQ_2_1_1	The CHAN_BOND_SEQ_2 attributes are used in conjunction with CHAN_BOND_SEQ_2_ENABLE to define the second channel bonding sequence. When CHAN_BOND_SEQ_2_USE is TRUE, the second sequence is used as an alternate sequence to trigger channel bonding. Each subsequence is 10 bits long. The rules for setting the subsequence depend on INTDATAWIDTH and RX_DECODE_SEQ_MATCH. See the Description section to learn how to set channel bonding sequences. Not all subsequences need to be used. CHAN_BOND_SEQ_LEN determines how many of the subsequences are used for a match. If CHAN_BOND_SEQ_LEN = `1`, only CHAN_BOND_SEQ_2_1 is used. CHAN_BOND_SEQ_2_ENABLE can be used to make parts of the sequence don't cares. If CHAN_BOND_SEQ_2_ENABLE[k] is `0`, CHAN_BOND_SEQ_2_k is a don't care subsequence and is always a match.
CHAN_BOND_SEQ_2_2_0 CHAN_BOND_SEQ_2_2_1	
CHAN_BOND_SEQ_2_3_0 CHAN_BOND_SEQ_2_3_1	
CHAN_BOND_SEQ_2_4_0 CHAN_BOND_SEQ_2_4_1	
CHAN_BOND_SEQ_2_ENABLE_0 CHAN_BOND_SEQ_2_ENABLE_1	
CHAN_BOND_SEQ_2_USE_0 CHAN_BOND_SEQ_2_USE_1	Determines if the second channel bonding sequence is to be used. TRUE: Channel bonding can be triggered by channel bonding sequence 1 or 2. FALSE: Channel bonding is only triggered by sequence 1.

Table 7-33: **Channel Bonding Attributes *(Continued)***

Attribute	Description
CHAN_BOND_LEVEL_0 CHAN_BOND_LEVEL_1	CHAN_BOND_LEVEL indicates the amount of internal pipelining used for the RX elastic buffer control signals. A higher value permits more daisy-chaining of RXCHBONDO and RXCHBONDI to ease placement and routing constraints. To minimize required latency through the RX elastic buffer, CHAN_BOND_LEVEL in the master should be set to the smallest value possible for the required amount of daisy-chaining. See the Description section for channel bonding to learn how to set the channel bonding level.
CHAN_BOND_MODE_0 CHAN_BOND_MODE_1	Defines the channel bonding mode of operation for this transceiver. OFF: No channel bonding. MASTER: This transceiver is master for channel bonding. Its RXCHBONDO port directly drives RXCHBONDI ports on one or more SLAVE transceivers. SLAVE: This transceiver is a slave for channel bonding. Its RXCHBONDI port is directly driven by the RXCHBONDO port of another SLAVE or MASTER transceiver. If its CHAN_BOND_LEVEL setting is greater than 0, its RXCHBONDO port may directly drive RXCHBONDI ports on one or more other SLAVE transceivers.
CHAN_BOND_SEQ_1_1_0 CHAN_BOND_SEQ_1_1_1	The CHAN_BOND_SEQ_1 attributes are used in conjunction with CHAN_BOND_SEQ_1_ENABLE to define channel bonding sequence 1. Each subsequence is 10 bits long. The rules for setting the subsequences depend on INTDATAWIDTH and RX_DECODE_SEQ_MATCH. See the Description section to learn how to set channel bonding subsequences. Not all subsequences need to be used. CHAN_BOND_SEQ_LEN determines how much of the sequence is used for a match. If CHAN_BOND_SEQ_LEN = `1`, only CHAN_BOND_SEQ_1_1 is used. CHAN_BOND_SEQ_1_ENABLE can be used to make parts of the sequence don't cares. If CHAN_BOND_SEQ_1_ENABLE[k] is `0`, CHAN_BOND_SEQ_1_k is a don't-care subsequence and is always a match.
CHAN_BOND_SEQ_1_2_0 CHAN_BOND_SEQ_1_2_1	
CHAN_BOND_SEQ_1_3_0 CHAN_BOND_SEQ_1_3_1	
CHAN_BOND_SEQ_1_4_0 CHAN_BOND_SEQ_1_4_1	
CHAN_BOND_SEQ_1_ENABLE_0 CHAN_BOND_SEQ_1_ENABLE_1	
CHAN_BOND_SEQ_2_1_0 CHAN_BOND_SEQ_2_1_1	The CHAN_BOND_SEQ_2 attributes are used in conjunction with CHAN_BOND_SEQ_2_ENABLE to define the second channel bonding sequence. When CHAN_BOND_SEQ_2_USE is TRUE, the second sequence is used as an alternate sequence to trigger channel bonding. Each subsequence is 10 bits long. The rules for setting the subsequence depend on INTDATAWIDTH and RX_DECODE_SEQ_MATCH. See the Description section to learn how to set channel bonding sequences. Not all subsequences need to be used. CHAN_BOND_SEQ_LEN determines how many of the subsequences are used for a match. If CHAN_BOND_SEQ_LEN = `1`, only CHAN_BOND_SEQ_2_1 is used. CHAN_BOND_SEQ_2_ENABLE can be used to make parts of the sequence don't cares. If CHAN_BOND_SEQ_2_ENABLE[k] is `0`, CHAN_BOND_SEQ_2_k is a don't care subsequence and is always a match.
CHAN_BOND_SEQ_2_2_0 CHAN_BOND_SEQ_2_2_1	
CHAN_BOND_SEQ_2_3_0 CHAN_BOND_SEQ_2_3_1	
CHAN_BOND_SEQ_2_4_0 CHAN_BOND_SEQ_2_4_1	
CHAN_BOND_SEQ_2_ENABLE_0 CHAN_BOND_SEQ_2_ENABLE_1	
CHAN_BOND_SEQ_2_USE_0 CHAN_BOND_SEQ_2_USE_1	Determines if the second channel bonding sequence is to be used. TRUE: Channel bonding can be triggered by channel bonding sequence 1 or 2. FALSE: Channel bonding is only triggered by sequence 1.

Oplver Ex1009
Page 192 of 347

Table 7-33: **Channel Bonding Attributes *(Continued)***

Attribute	Description
CHAN_BOND_SEQ_LEN_0 CHAN_BOND_SEQ_LEN_1	Defines the length in bytes of the channel bonding sequence (1 to 4 bytes) that the transceiver matches to detect opportunities for channel bonding.
PCI_EXPRESS_MODE_0 PCI_EXPRESS_MODE_1	Setting this attribute to TRUE enables certain operations specific to PCI Express operation, specifically, recognizing TXELECIDLE = `1`, TXCHARDISPMODE = `1`, and TXCHARDISPVAL = `0` as a request to power down the channel. TXCHARDISPMODE = `1` and TXCHARDISPVAL = `0` encode the PIPE interface signal TXCompliance = `1` of the PIPE specification. The TXCHARDISPMODE and TXCHARDISPVAL settings encode for PIPE and enable special support for FTS lane deskew.

Description

Enabling Channel Bonding

Each GTP transceiver includes a circuit that performs channel bonding by controlling the pointers of the RX elastic buffer. To use channel bonding, the RX_BUFFER_USE attribute must be TRUE to turn on the elastic buffer.

Each GTP transceiver has a channel bonding circuit. Configuring a GTP transceiver for channel bonding requires the following steps:

1. Set the channel bonding mode for each GTP transceiver.
2. Set master to CHAN_BOND_MODE = MASTER.
3. Set slave to CHAN_BOND_MODE = SLAVE.
4. Connect the channel bonding port from the master to each slave, either directly or by daisy chaining.
5. Set the channel bonding sequence and detection parameters.

Channel Bonding Mode

The channel bonding mode for each GTP transceiver determines whether channel bonding is active and whether the GTP transceiver is the master or a slave. Each set of channel-bonded GTP transceivers must have one master and can have any number of slaves. To turn on channel bonding for a group of GTP transceivers, one transceiver is set to *Master*. The remaining GTP transceivers in the group are set to *Slaves*.

In the general use case, the master and slaves of a channel bonding group are located in the same column (for example, the X0 column of a TXT device). An advanced case is when the master and the slaves of a channel bonding group are located in different columns. Contact your system I/O specialist for details on the advanced case.

Connecting Channel Bonding Ports

The channel bonding operation requires connecting the master GTP RXCHBONDO port to the RXCHBONDI port of all slaves in the group. A direct connection is required for adjacent GTP transceivers. To directly connect a master to a slave:

1. Connect the RXCHBONDO port of the master to the RXCHBONDI port of the slave.
2. Set the CHAN_BOND_LEVEL of the master to 1.

3. Set the CHAN_BOND_LEVEL of each slave to 0.

When GTP transceivers are directly connected, meeting the timing constraints becomes difficult as the transceivers get further apart. The solution to this problem is to connect the transceivers in a daisy chain. Daisy chaining is performed using the CHAN_BOND_LEVEL signal to allow additional pipeline stages between the master and the slave. The RXCHBONDO port of each slave is used as a pipeline stage in the RXCHBONDO path from the master. Figure 7-29 and Figure 7-30 show two daisy-chain examples.

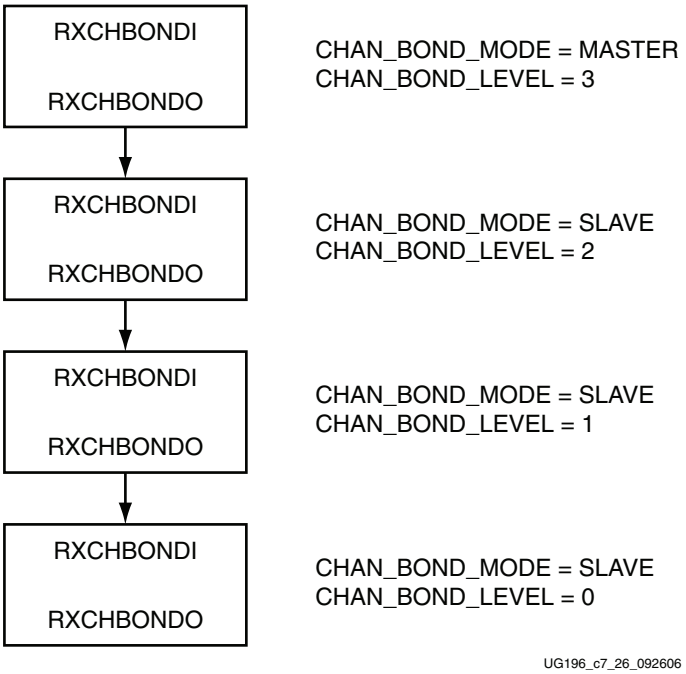
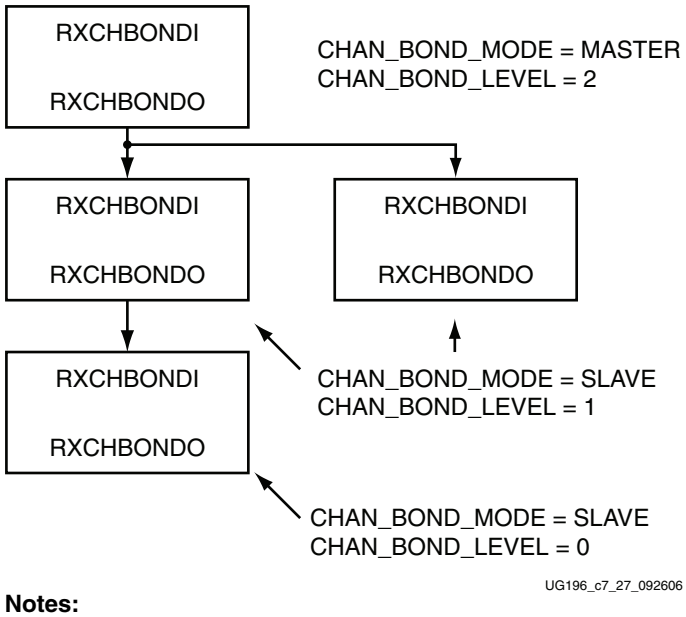


Figure 7-29: **Channel Bonding Daisy Chain Example 1**

Capriver Ex1009
Page 194 of 347



Notes:

1. Each box can represent either transceiver in a GTP_DUAL tile.
2. This channel bonding daisy-chain example can be implemented using from two to four GTP_DUAL tiles.

Figure 7-30: **Channel Bonding Daisy Chain Example 2**

To set up a daisy chain, first connect the GTP transceivers using RXCHBONDO and RXCHBONDI to create a path from the RXCHBONDI port of each slave to the RXCHBONDO port of the master. To set the CHAN_BOND_LEVEL for the GTP transceivers in the chain:

1. Set the CHAN_BOND_LEVEL of the master to 7.
2. Set the CHAN_BOND_LEVEL of each slave to the CHAN_BOND_LEVEL of the GTP transceiver driving the slave's RXCHBONDI port minus 1.
3. Find the slave with the lowest level. Subtract this level from the CHAN_BOND_LEVEL of all GTP transceivers so that the lowest slave has level 0 and the master has the minimum level required to service all the slaves.

Any number of GTP transceivers can be channel bonded together as long as the timing constraints of the design are met and all clocking guidelines are followed (see Clocking from a Neighboring GTP_DUAL Tile, page 83).

When the connections between channel bonding ports among GTP transceivers are being decided, the designer must remember that RXCHBONDI and RXCHBONDO belong to the RXUSRCLK clock domain. Meeting the timing constraint of RXUSRCLK becomes increasingly difficult as RXUSRCLK increases in frequency and as directly connected transceivers get further apart.

Selecting a GTP transceiver in the middle of the GTP_DUAL column to be the master for channel bonding allows for the most flexibility when connecting channel bonding ports. When the channel bonding master is in the middle of the GTP_DUAL column, connections can be made to GTP transceivers north and south of the master. This configuration allows for daisy chaining of up to seven levels north and seven levels south of the channel bonding master. Because of the GTP dedicated clock routing structure, an additional benefit of having the channel bonding master at the center of the GTP_DUAL column is that up to 14 GTP transceivers can be channel bonded together using a single clock pin

pair. If more than 14 GTP transceivers are channel bonded together, the use of additional clock pins is required as well as using the same oscillator (see Clocking from a Neighboring GTP_DUAL Tile, page 83).

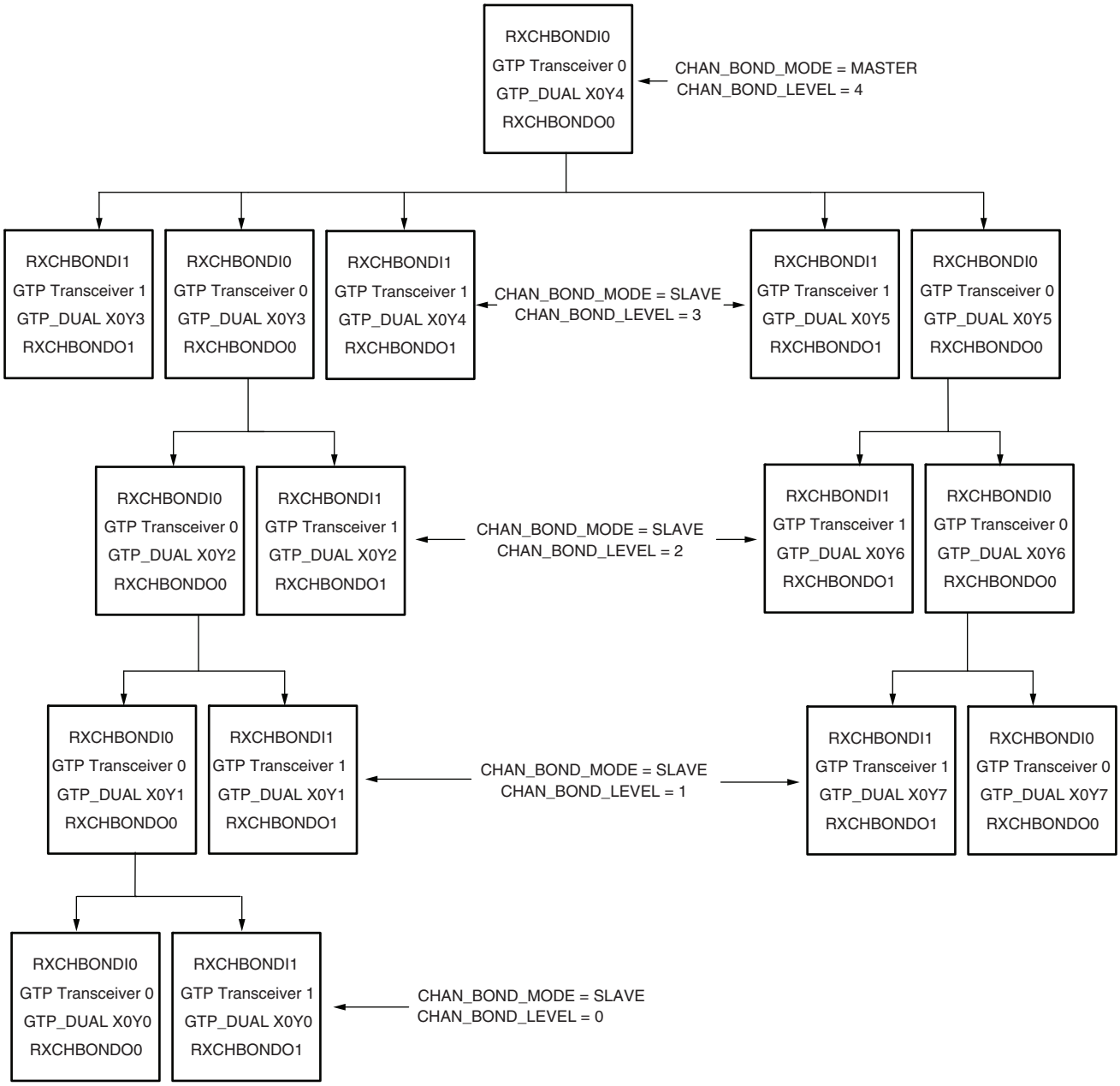
Note: GTP transceivers that are channel bonded together *must* belong to the same column of the device.

As long as timing constraints are met:

- There is no limit to the number of GTP transceivers that can be on a particular CHAN_BOND_LEVEL.
- GTP transceivers from the same GTP_DUAL tile can be on the same CHAN_BOND_LEVEL.
- GTP transceivers from different GTP_DUAL tiles can be on the same CHAN_BOND_LEVEL.

Figure 7-31 shows an example for channel bonding 16 GTP transceivers. This example is only one of many ways to channel bond 16 GTP transceivers. In this example, transceivers of the X0 column are channel-bonded together.

Capriver Ex1009
Page 196 of 347



Note: In this example, the GTP transceivers that are channel bonded together belong to the same column of the device. Contact your system I/O specialist for details on an advanced configuration, where the master and the slaves of a channel bonding group are located in different columns.

UG196_c7_36_100409

Figure 7-31: **Channel Bonding Example using 16 GTP Transceivers**

Setting the Channel Bonding Sequence

The channel bonding sequence is programmed in the same way as the clock correction sequence. CHAN_BOND_SEQ_LEN sets the length of the sequence from one to four subsequences. CHAN_BOND_SEQ_1_* sets the values of the sequence. If CHAN_BOND_SEQ_2_USE is TRUE, CHAN_BOND_SEQ_2_* sets the values for the alternate second sequence.

The number of active bits in each subsequence depends on INTDATAWIDTH (see Chapter 5, Tile Features) and RX_DECODE_SEQ_MATCH (see Configurable Clock Correction, page 183). Figure 7-32 shows how the subsequence bits are mapped.

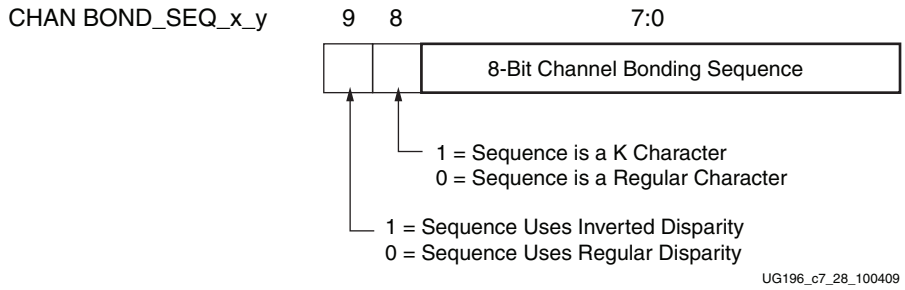


Figure 7-32: **Channel Bonding Sequence Settings**

As with clock correction sequences, channel bonding sequences can have *don't care* subsequences. CHAN_BOND_SEQ_1_ENABLE and CHAN_BOND_SEQ_2_ENABLE set these bytes. Figure 7-33 shows the mapping of the enable attributes for the channel bonding subsequences.

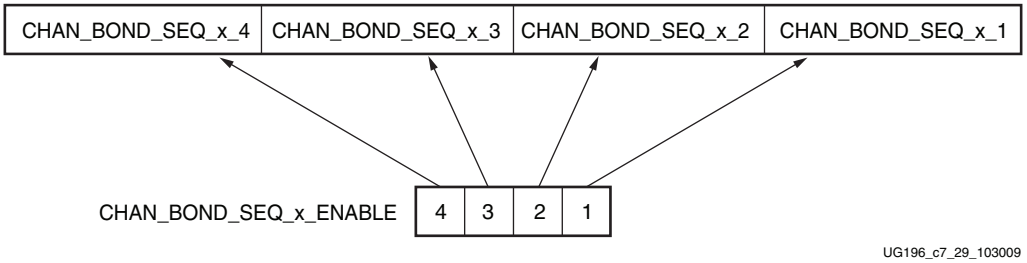


Figure 7-33: **Channel Bonding Sequence Mapping**

Setting the Maximum Skew

When the master receives a channel bonding sequence, it does not trigger channel bonding immediately. Several more bytes must arrive in case the slaves have more latency. This wait time effectively becomes the maximum skew that the RX elastic buffer can handle. If the skew is greater than the wait time, the slaves might not receive the sequence by the time the master triggers channel bonding (see Figure 7-34).

Figure 7-34 shows two FIFOs, one for the master and one for the slave. If the slave is behind the master, the master must wait several cycles before triggering channel bonding, otherwise the slow slave will not have the channel bonding sequence in its buffer.

Cpuver Ex1009
Page 198 of 347

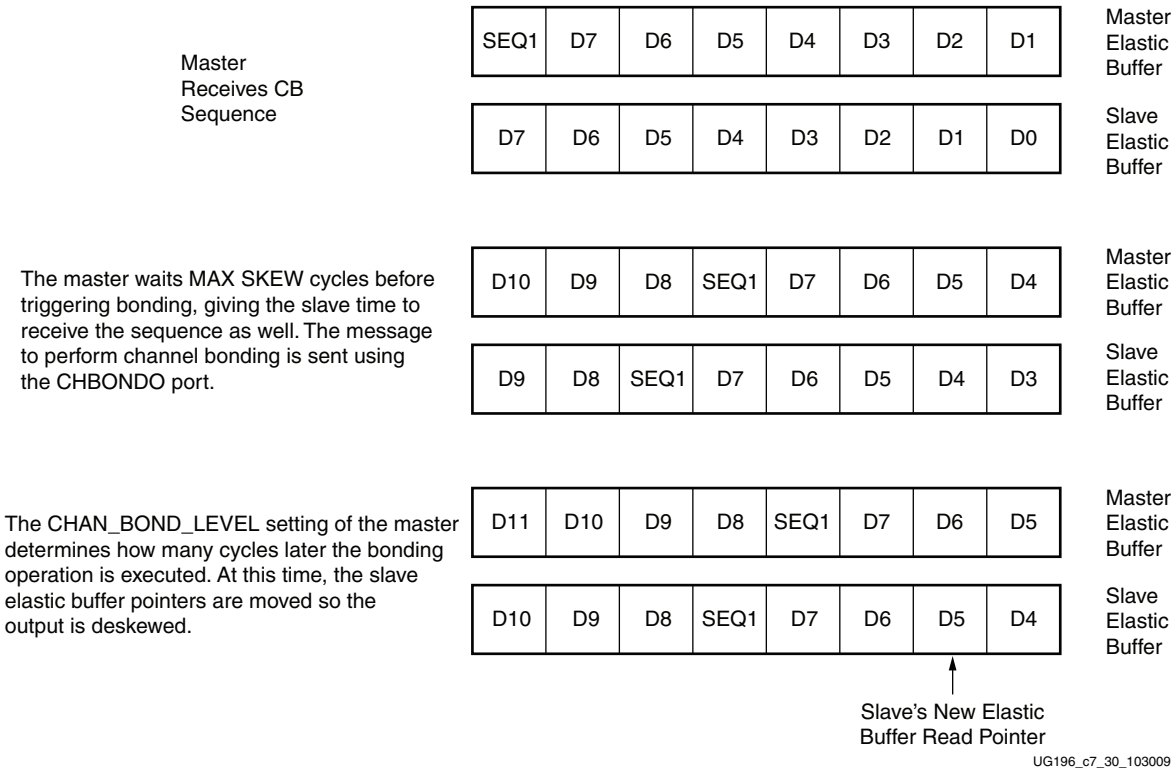


Figure 7-34: **Channel Bonding Example (MAX_SKEW = 2 and Master CHAN_BOND_LEVEL = 1)**

CHAN_BOND_1_MAX_SKEW and CHAN_BOND_2_MAX_SKEW are used to set the maximum skew allowed for channel bonding sequences 1 and 2, respectively. The maximum skew range is 1 to 14 and must be set to less than one-half the minimum distance between channel bonding sequences. This minimum distance is determined by the protocol being used.

Precedence between Channel Bonding and Clock Correction

The clock correction (see Configurable Clock Correction, page 183) and channel bonding circuits both perform operations on the pointers of the RX elastic buffer. Normally, the two circuits work together without conflict, except when clock correction events and channel bonding events occur simultaneously. In this case, one of the two circuits must take precedence. To make clock correction a higher priority than channel bonding, CLK_COR_PRECEDENCE must be set to TRUE. To make channel bonding a higher priority, CLK_COR_PRECEDENCE must be set to FALSE.

Note: For clock correction operation, the master controls clock correction through the RXCHBONDO/RXCHBONDI bus. Control through this bus ensures that the channels remain properly aligned following the channel bonding operation. If the master gets disconnected, the clock correction ceases operation on the slaves.

Optiver Ex1009
Page 199 of 347

FPGA RX Interface

Overview

The FPGA receives RX data from the GTP receiver through the FPGA RX interface. Data is read from the RXDATA port on the positive edge of RXUSRCLK2.

The width of RXDATA can be configured to be one or two bytes wide. The actual width of the port depends on the internal data width of the GTP_DUAL tile, and whether or not the 8B/10B decoder is enabled. Ports widths of 8 bits, 10 bits, 16 bits, and 20 bits are possible.

The rate of the parallel clock (RXUSRCLK2) at the interface is determined by the RX line rate, the width of the RXDATA port, and whether or not 8B/10B decoding is enabled. RXUSRCLK must be provided for the internal PCS logic in the receiver. This section shows how to drive the parallel clocks and explains the constraints on those clocks for correct operation.

Ports and Attributes

Table 7-34 defines the FPGA RX interface ports.

Table 7-34: **FPGA RX Interface Ports**

Port	Dir	Clock Domain	Description
INTDATAWIDTH	In	Async	Specifies the bit width for the TX and RX paths. The bit width of TX and RX must be identical for both channels. `0`: 8-bit width `1`: 10-bit width(1)
REFCLKOUT	Out	N/A	The REFCLKOUT port from each GTP_DUAL tile provides access to the reference clock provided to the shared PMA PLL (CLKIN). It can be routed for use in the FPGA logic.
RXDATA0 RXDATA1	Out	RXUSRCLK2	Receive data bus of the receive interface to the FPGA. The width of RXDATA(0/1) depends on the setting of RXDATAWIDTH(0/1).
RXDATAWIDTH0 RXDATAWIDTH1	In	RXUSRCLK2	Selects the width of the RXDATA(0/1) receive data connection to the FPGA. `0`: 1-byte interface => RXDATA(0/1)[7:0] `1`: 2-byte interface => RXDATA(0/1)[15:0] The clock domain depends on the selected clock (RXRECCLK(0/1), RXUSRCLK(0/1), RXUSRCLK2(0/1)) for this interface.
RXRECCLK0 RXRECCLK1	Out	N/A	Recovered clock from the CDR. Clocks the RX logic between the PMA and the RX elastic buffer. Can be used to drive RXUSRCLK synchronously with incoming data. When RXPOWERDOWN[1:0] is set to `11`, which is P2 the lowest power state, then RXRECCLK of this transceiver is indeterminate. RXRECCLK of this GTP transceiver is either a static `1` or a static `0`.
RXRESET0 RXRESET1	In	Async	PCS RX system reset. Resets the RX elastic buffer, 8B/10B decoder, comma detect, and other receiver registers. This is a per channel subset of GTPRESET.

Apotex Ex1009
Page 200 of 347

Table 7-34: **FPGA RX Interface Ports *(Continued)***

Port	Dir	Clock Domain	Description
RXUSRCLK0 RXUSRCLK1	In	N/A	Input clock used for internal RX logic after the RX elastic buffer. Generally, RXUSRCLK(0/1) runs at either 1/8th or 1/10th the RX baud rate, which for some standards is identical to TXUSRCLK(0/1).
RXUSRCLK20 RXUSRCLK21	In	N/A	Input clock used for the interface between the FPGA and the GTP transceiver. For a 1-byte interface, RXUSRCLK2(0/1) is equal to RXUSRCLK(0/1). For a 2-byte interface, RXUSRCLK2(0/1) runs at one-half RXUSRCLK(0/1). For some standards, TXUSRCLK2(0/1) is identical to TXUSRCLK2(0/1).

Notes:

1. 10-bit internal data width is necessary when the RX elastic buffer is bypassed.

There are no attributes in this section.

Description

The FPGA RX interface allows parallel received data to be read from the GTP transceiver. For this interface to be used, the following must be done:

- The width of the RXDATA port must be configured.
- RXUSRCLK2 and RXUSRCLK must be connected to clocks running at the correct rate.

Configuring the Width of the Interface

Table 7-35 shows how to select the interface width for the RX datapath. 8B/10B decoding is discussed in more detail in Configurable 8B/10B Decoder, page 172.

Table 7-35: **RX Datapath Width Configuration**

INTDATAWIDTH(1,2)	RXDATAWIDTH(3)	RXDEC8B10BUSE	FPGA RX Interface Width
0	0	N/A	8 bits
0	1	N/A	16 bits
1	0	0	10 bits
1	0	1	8 bits
1	1	0	20 bits
1	1	1	16 bits

Notes:

1. 10-bit internal data width is necessary when the RX elastic buffer is bypassed.
2. The internal datapath is 8 bits when INTDATAWIDTH is Low and 10 bits when INTDATAWIDTH is High.
3. The RXDATA interface is one byte wide when RXDATAWIDTH is Low and two bytes wide when RXDATAWIDTH is High.

Figure 7-35 shows how RXDATA is received serially when the internal datapath is eight bits (INTDATAWIDTH is Low) and 8B/10B decoding is disabled.

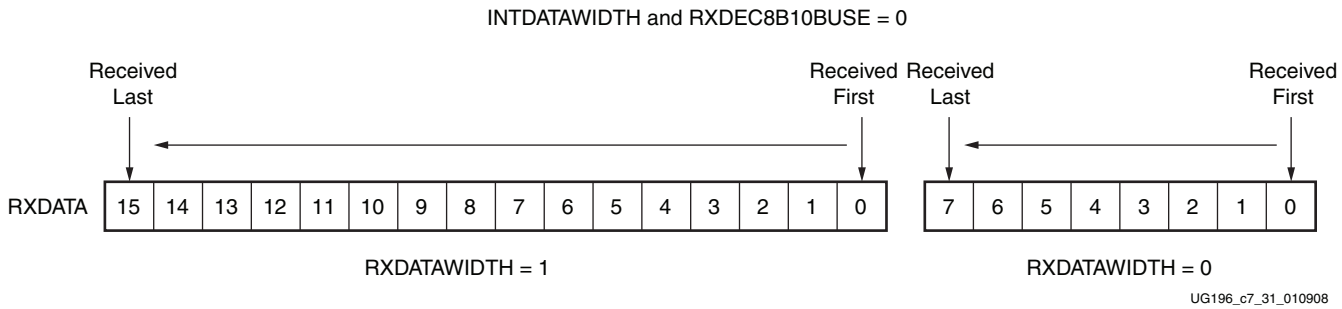


Figure 7-35: **RX Interface with 8B/10B Bypassed (8-Bit Internal Datapath)**

Figure 7-36 shows how RXDATA is received serially when the internal datapath is 10 bits (INTDATAWIDTH is High) and 8B/10B decoding is disabled. When RXDATA is 10 bits or 20 bits wide, the RXDISPERR and RXCHARISK ports are taken from the 8B/10B decoder interface and are used to present the extra bits.

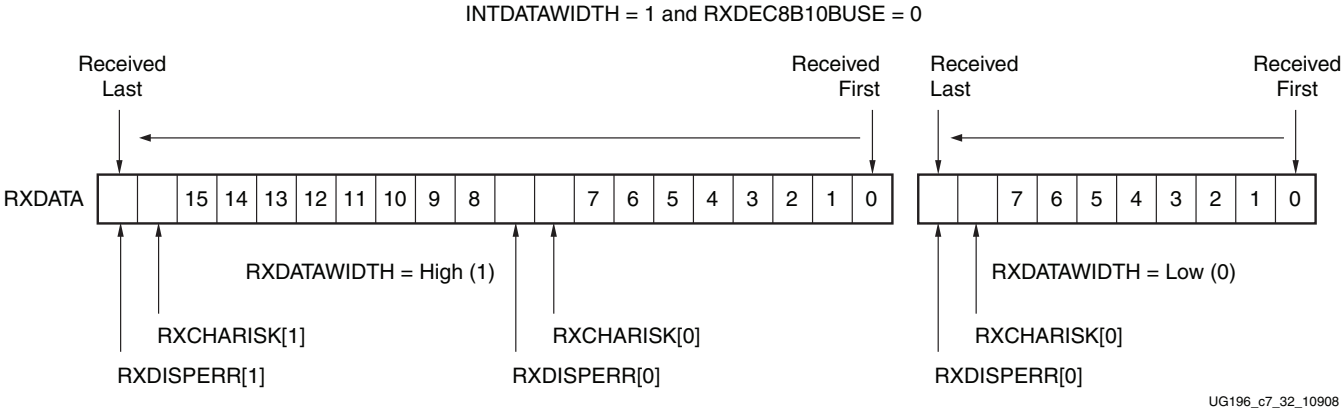


Figure 7-36: **RX Interface with 8B/10B Bypassed (10-Bit Internal Datapath)**

When 8B/10B decoding is used, the data interface is a multiple of 8 bits like in Figure 7-35, but the data is decoded before it is presented at the RXDATA port. Refer to Configurable 8B/10B Decoder, page 172 for more details about bit ordering when using 8B/10B decoding.

Connecting RXUSRCLK and RXUSRCLK2

The FPGA RX interface includes two parallel clocks: RXUSRCLK and RXUSRCLK2. RXUSRCLK is the internal clock for the PCS logic in the GTP receiver. The required rate for RXUSRCLK depends on the internal datapath width of the GTP_DUAL tile (INTDATAWIDTH) and the RX line rate of the GTP receiver (see Serial In to Parallel Out, page 155 to see how RX line rate is determined). Equation 7-7 shows how to calculate the required rate for RXUSRCLK.

$$\text{RXUSRCLK Rate} = \frac{\text{Line Rate}}{\text{Internal Datapath Width}} \qquad \textit{Equation 7-7}$$

RXUSRCLK2 is the main synchronization clock for all signals into the RX side of the GTP transceiver. Most signals into the RX side of the GTP receiver are sampled on the positive edge of RXUSRCLK2. RXUSRCLK2 is the same rate at RXUSRCLK when RXDATAWIDTH

Apotex Ex1009
Page 202 of 347

is Low, and one half the rate of RXUSRCLK when RXDATAWIDTH is High. Equation 7-8 shows how to calculate the required rate for RXUSRCLK2.

$$\text{RXUSRCLK2 Rate} = \frac{\text{Line Rate}}{\text{Internal Datapath Width} \times \text{Bytes in Interface}} \qquad \textit{Equation 7-8}$$

There are some rules about the relationships between clocks that must be observed for RXUSRCLK, RXUSRCLK2, and CLKIN.

First, RXUSRCLK and RXUSRCLK2 must be positive edge aligned, with as little skew as possible between them. As a result, low-skew clock resources (BUFGs and BUFRs) should be used to drive RXUSRCLK and RXUSRCLK2. When the two are the same frequency, the same clock resource is used to drive both. When the two are different frequencies, RXUSRCLK is divided to get RXUSRCLK2. The designer must ensure that the two are positive edge aligned.

If the channel is configured so the same oscillator drives the reference clock for the transmitter and the receiver, REFCLKOUT or TXOUTCLK can be used to drive RXUSRCLK and RXUSRCLK2 in the same way as they are used to drive TXUSRCLK and TXUSRCLK2. When clock correction is turned off, RX phase alignment must be used to align the serial clock and the parallel clocks. See Configurable RX Elastic Buffer and Phase Alignment, page 176 for details about enabling phase alignment.

If separate oscillators are driving the reference clocks for the transmitter and receiver on the channel, and clock correction is not used, RXUSRCLK and RXUSRCLK2 must be driven by RXRECCLK, and the phase-alignment circuit must be used.

If clock correction is used, RXUSRCLK and RXUSRCLK2 can be sourced by RXRECCLK, REFCLKOUT, or TXOUTCLK.

Optiver Ex1009
Page 203 of 347

Cpuver Ex1009
Page 204 of 347

Chapter 8

Cyclic Redundancy Check

Overview

In Virtex-5 devices, each high-speed transceiver tile is paired with two cyclic redundancy check (CRC) integrated blocks. Each CRC block can operate independently as two 32-bit input CRC modules (CRC32) or can be combined into a single 64-bit input CRC module (CRC64). The CRC modules use the standard 32-bit Ethernet polynomial for CRC calculation. The CRC integrated blocks are independent of the transceiver blocks.

Figure 8-1 shows the basic port interface of the CRC block.

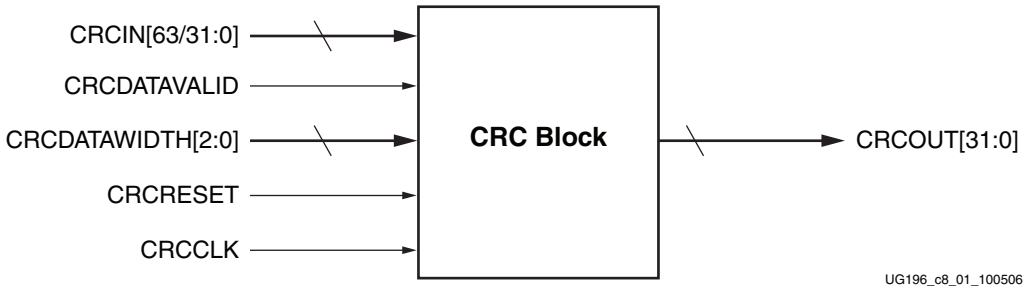


Figure 8-1: **CRC Integrated Block**

For clarification:

- Each GTP_DUAL tile is paired with two integrated CRC blocks.
- Each integrated CRC block can either operate as one 64-bit wide CRC module or as two independent 32-bit wide CRC modules.
- For a given GTP_DUAL tile, four independent 32-bit wide CRC modules are only possible when a 64-bit wide CRC module is not used.

Figure 8-2 shows how CRC modules are typically used in an application.

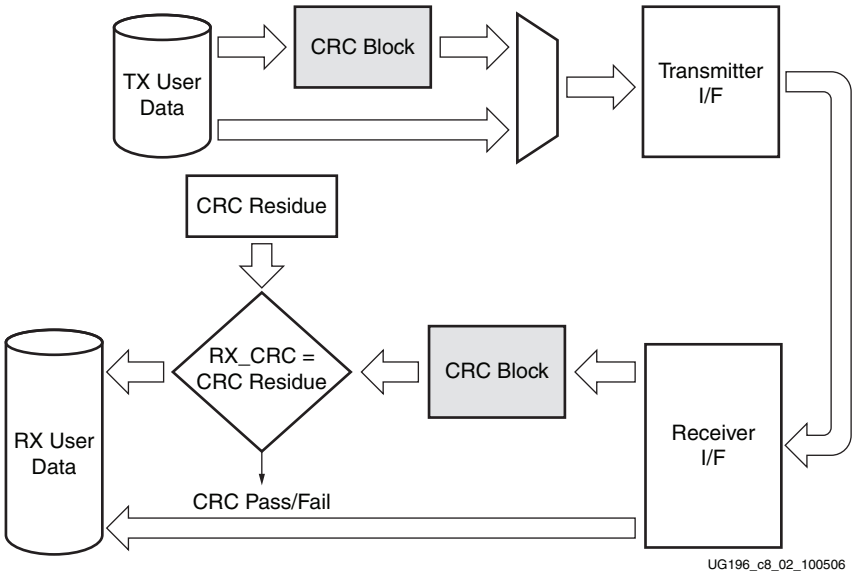


Figure 8-2: **CRC Application**

The CRC block only calculates the CRC for the input data stream using the standard polynomial. The CRC blocks do not perform any data framing. The application is responsible for appending CRC values to outgoing frames and validating the CRC on the RX side.

Ports and Attributes

Table 8-1 defines the CRC 64-bit I/O ports, and Table 8-2 defines the CRC 32-bit I/O ports.

Table 8-1: **CRC 64-Bit I/O Ports**

Port	Dir	Port Size	Clock Domain	Description
CRCCLK	In	1	N/A	CRC clock.
CRCDATAVALID	In	1	CRCCLK	Indicates valid data on the CRCIN inputs. 1: Data valid 0: Data invalid Deasserting this signal causes the CRC value to be held for the same number of cycles that this signal is deasserted.
CRCDATAWIDTH[2:0]	In	3	CRCCLK	Indicates how many input data bytes are valid. Refer to Table 8-4, page 208 and Table 8-5, page 208 for input byte ordering for CRC32 and CRC64, respectively.
CRCIN[63:0]	In	64	CRCCLK	CRC input data. The maximum datapath width is eight bytes.
CRCOUT[31:0]	Out	32	CRCCLK	32-bit CRC output. CRCOUT is the byte-reversed, bit-inverted CRC value corresponding to the CRC calculation on valid bytes from the previous clock cycle and the previous CRC value. **Note:** CRCDATAVALID must be driven High.
CRCRESET	In	1	CRCCLK	Synchronous reset of CRC registers. When CRCRESET is asserted, the CRC block is initialized to the CRC_INIT value.

Apotex Ex1009
Page 206 of 347

Table 8-2: **CRC 32-Bit I/O Ports**

Port	Dir	Port Size	Clock Domain	Description
CRCCLK	In	1	N/A	CRC clock.
CRCDATAVALID	In	1	CRCCLK	Indicates valid data on the CRCIN inputs. `1`: Data valid `0`: Data invalid Deasserting this signal causes the CRC value to be held for the number of cycles that this signal is deasserted.
CRCDATAWIDTH[2:0]	In	3	CRCCLK	Indicates how many input data bytes are valid. Refer to Table 8-4, page 208 and Table 8-5, page 208 for input byte ordering for CRC32 and CRC64, respectively.
CRCIN[31:0]	In	32	CRCCLK	CRC input data. The maximum datapath width is four bytes.
CRCOUT[31:0]	Out	32	CRCCLK	32-bit CRC output. CRCOUT is the byte-reversed, bit-inverted CRC value corresponding to the CRC calculation on valid bytes from the previous clock cycle and the previous CRC value. **Note:** CRCDATAVALID must be driven High.
CRCRESET	In	1	CRCCLK	Synchronous reset of CRC registers. When CRCRESET is asserted, the CRC block is initialized to the CRC_INIT value.

Table 8-3 defines the CRC64 and CRC32 attributes.

Table 8-3: **CRC64/CRC32 Attributes**

Attribute	Description
CRC_INIT[31:0]	32-bit value for initial state of CRC internal registers in the CRC32/CRC64 block. When CRCRESET is applied, the CRC registers are synchronously initialized to this value. The default value is `0xFFFFFFFF`.

Description

Using CRC for Error Checking

A CRC is an error-checking mechanism for a block of data, such as a frame of network traffic. The calculated CRC is used to detect errors after transmission or storage and is calculated on a per clock cycle basis.

A CRC is computed for the contents of a frame and appended to the end of the frame before transmission or storage. When the frame is received or retrieved, it is verified by recalculating the CRC for the contents to confirm that no changes occurred.

CRCs are simple to implement in digital hardware, easy to analyze mathematically, and good at detecting common errors caused by noise in transmission channels.

The CRC Primitive

Each CRC block computes a 32-bit CRC using the CRC32 polynomial specified for PCI Express, Gigabit Ethernet, and other common protocols. The CRC32 polynomial is:

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

Equation 8-1

There are two primitives for instantiating CRC integrated blocks. The 32-bit CRC primitive (CRC32) can process 8, 16, 24, or 32-bit input data and generates a 32-bit CRC. The 64-bit primitive (CRC64) can process 8, 16, 24, 32, 40, 48, 56, or 64-bit input data and also generates a 32-bit CRC. Using the CRC64 primitive consumes both CRC integrated blocks paired with a given transceiver tile.

Table 8-4: **CRC32 – Valid Data Widths**

CRCDATAWIDTH[2:0]	Data Width	CRC Data Bus Bits
000	8-bit	CRCIN[31:24]
001	16-bit	CRCIN[31:16]
010	24-bit	CRCIN[31:8]
011	32-bit	CRCIN[31:0]

Notes:

1. CRCDATAWIDTH[2] must *always* be held Low for the CRC32 primitive.

For CRC64, CRCDATAWIDTH is interpreted as indicated in Table 8-5.

Table 8-5: **CRC64 – Valid Data Widths**

CRCDATAWIDTH[2:0]	Data Width	CRC Data Bus Bits
000	8-bit	CRCIN[63:56]
001	16-bit	CRCIN[63:48]
010	24-bit	CRCIN[63:40]
011	32-bit	CRCIN[63:32]
100	40-bit	CRCIN[63:24]
101	48-bit	CRCIN[63:16]
110	56-bit	CRCIN[63:8]
111	64-bit	CRCIN[63:0]

Cpuver Ex1009
Page 208 of 347

Using the CRC Blocks

Figure 8-3 shows a CRC block calculating the CRC for input data. Also shown in Figure 8-3 is the CRC64 primitive. This operation is performed when the CRC is being generated or checked. CRC_POLY is the fixed CRC32 polynomial used for all calculations.

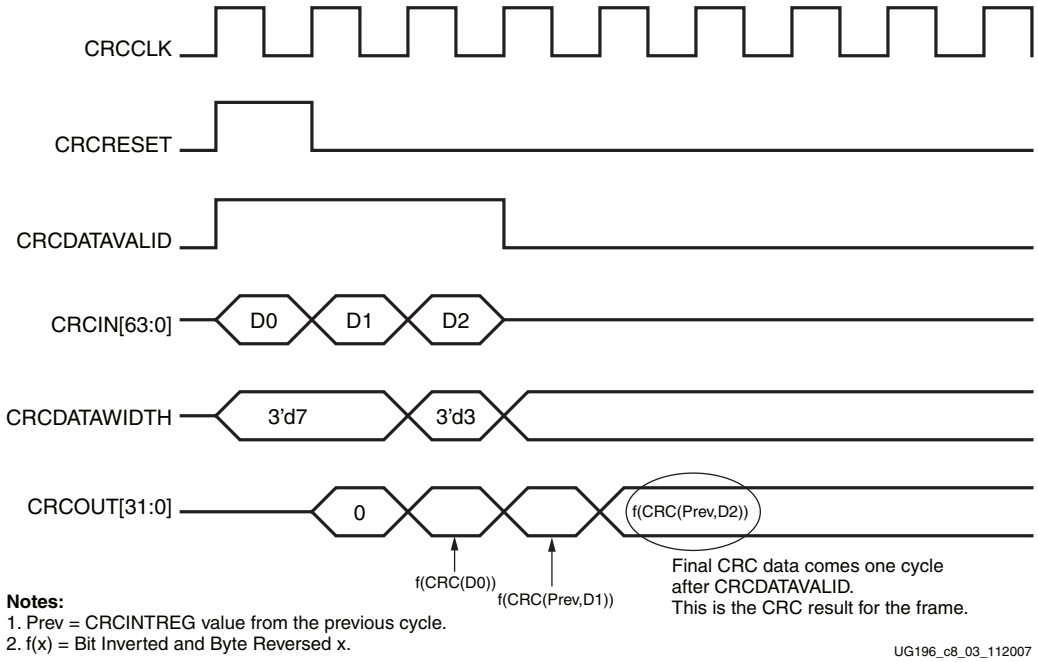


Figure 8-3: **CRC Block Timing**

At the start of each frame, CRCRESET must be applied to set the initial CRC value to CRC_INIT. CRC calculations are cumulative, so this step is required to start the CRC calculation at a known value. CRC_INIT is a 32-bit value for the initial state of the CRC internal register. Its default value is `0xFFFFFFFF`. When CRCRESET is driven Low, on the first cycle the CRC block outputs `0x00000000` on the CRCOUT port. The following cycles will have the calculated CRC value for the data on the CRCIN port. The CRC_INIT value required for a given protocol is specified as part of that protocol's CRC algorithm. Table 8-6 shows the CRC_INIT values for some common protocols that use the CRC32 polynomial.

Table 8-6: **CRC_INIT Values for Some Common Protocols**

Protocol	CRC_INIT
Ethernet	`32'hFFFF_FFFF`
PCI Express	`32'hFFFF_FFFF`
Infiniband	`32'hFFFF_FFFF`
Fibre Channel	`32'hFFFF_FFFF`
SATA	`32'h5232_5032`

The CRCDATAVALID port acts as a clock enable for the CRC block. If CRCDATAVALID is High and CRCRESET has been deasserted, a new CRC value is calculated every clock cycle, and the result appears on CRCOUT after one clock cycle. If CRCDATAVALID is Low, all CRC registers hold their values from the previous cycle.

CRCDATAWIDTH determines how many input data bytes are valid. Table 8-4, page 208 shows how to use CRCDATAWIDTH to set the number of valid input bytes on the CRC32 primitive. Table 8-5, page 208 shows the same for the CRC64 primitive. CRCDATAWIDTH is usually used at the end of a frame, when not all the bytes in a data word contain valid data. CRCDATAWIDTH can also be tied off to accommodate datapaths smaller than 32 bits wide.

To achieve faster throughput using the CRC block, apply the first data byte(s) and a reset on the same clock cycle. As shown in Figure 8-3, the CRC result appears on the output CRCOUT[31:0] one clock cycle after the last valid data byte is applied. If the fastest throughput is not required, the CRC module can be reset on any prior cycle as long as the CRCDATAVALID input is 0.

In Figure 8-4, the CRCRESET is asserted one CRCLK cycle after CRCDATAVALID. This allows back-to-back CRC frame calculations.

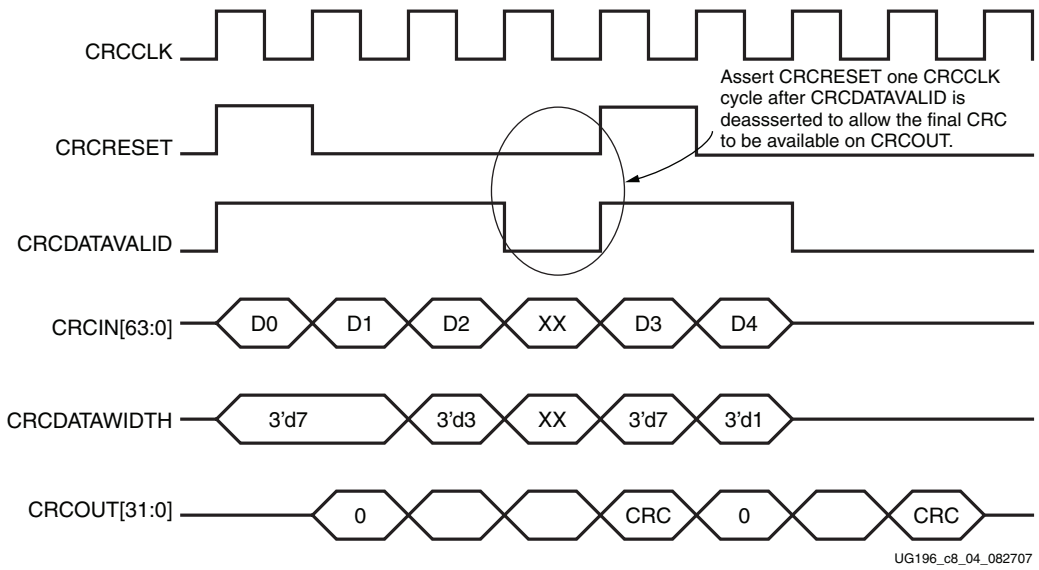


Figure 8-4: **CRCRESET Assertion Timing for Subsequent Frames**

Apple Ex1009
Page 210 of 347

In Figure 8-3, the internal CRCINTREG register shows the raw result of the CRC calculation. CRCOUT provides the bit-inverted, byte-reversed version of the CRC output at the same time. Figure 8-5 shows an example of the byte rotation and bit inversion operations.

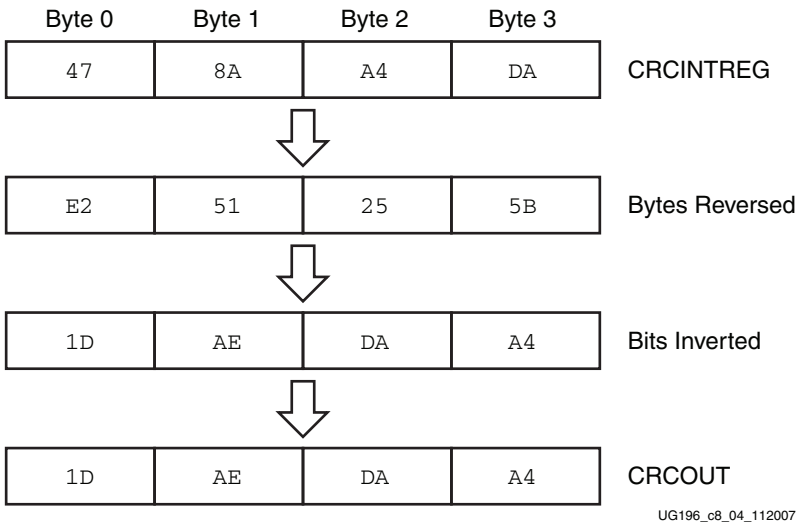


Figure 8-5: **Byte Rotation and Bit Inversion**

Byte rotation and bit inversion of the CRC output value are requirements for many common protocols, so CRCOUT is provided with these operations already performed to save FPGA resources. In designs that use a protocol that does not specify these operations in the CRC algorithm, the user must undo these operations before using the CRC value.

The CRC result for a given frame is the CRCOUT value corresponding to the final byte(s) of the frame. This value appears one cycle after the final byte(s) of the frame are presented to the CRCIN port with CRCDATAVALID High.

Integrating the CRC Blocks for TX

To use a CRC32 or CRC64 primitive to add CRC to frames for transmission or storage, the following logic needs to be built into the FPGA and connected to the CRC block:

- The Start of Frame (SOF) must trigger CRCRESET.
- If the datapath is not as wide as the maximum data width of the CRC primitive, CRCDATAWIDTH must be tied off so that only the bytes in the datapath are valid.
- The CRC output from the CRC block must be added to the frame, usually after the last data byte and before an End of Frame (EOF) character.

Integrating the CRC Blocks for RX

To use a CRC32 or CRC64 primitive to check the CRC on incoming data, the following logic needs to be built in the FPGA and connected to the CRC block:

- The SOF must trigger CRCRESET. This usually requires a decoder of some sort to find the SOF character in the incoming data stream.
- The EOF must trigger a CRC check.
  - If fixed length frames are used, this can be done with a counter.

- If variable length frames are used, this typically requires setting a count based on a length value in the frame, or decoding an EOF character in the incoming data stream.

- The CRC check must be performed. There are two common methods of CRC checking:
  - The compare method

    Calculate CRC over all the data in the frame, and then extract the CRC from the frame and compare it to the calculated CRC. The EOF trigger is used to indicate which CRCOUT value should be used and which bytes of the frame contain the transmitted CRC.
  - The residue method

    Calculate CRC over all the data in the frame *and* the transmitted CRC, and then compare the result to the residue for the CRC32 polynomial. If the two values match, the CRC check passes. The residue value for CRC32, after bit inversion and byte reversal, is `32'h1CDF4421`.
- Remove the transmitted CRC from the frame, if necessary.

Implementation of the CRC Block

Figure 8-6 shows an implementation of the CRC.

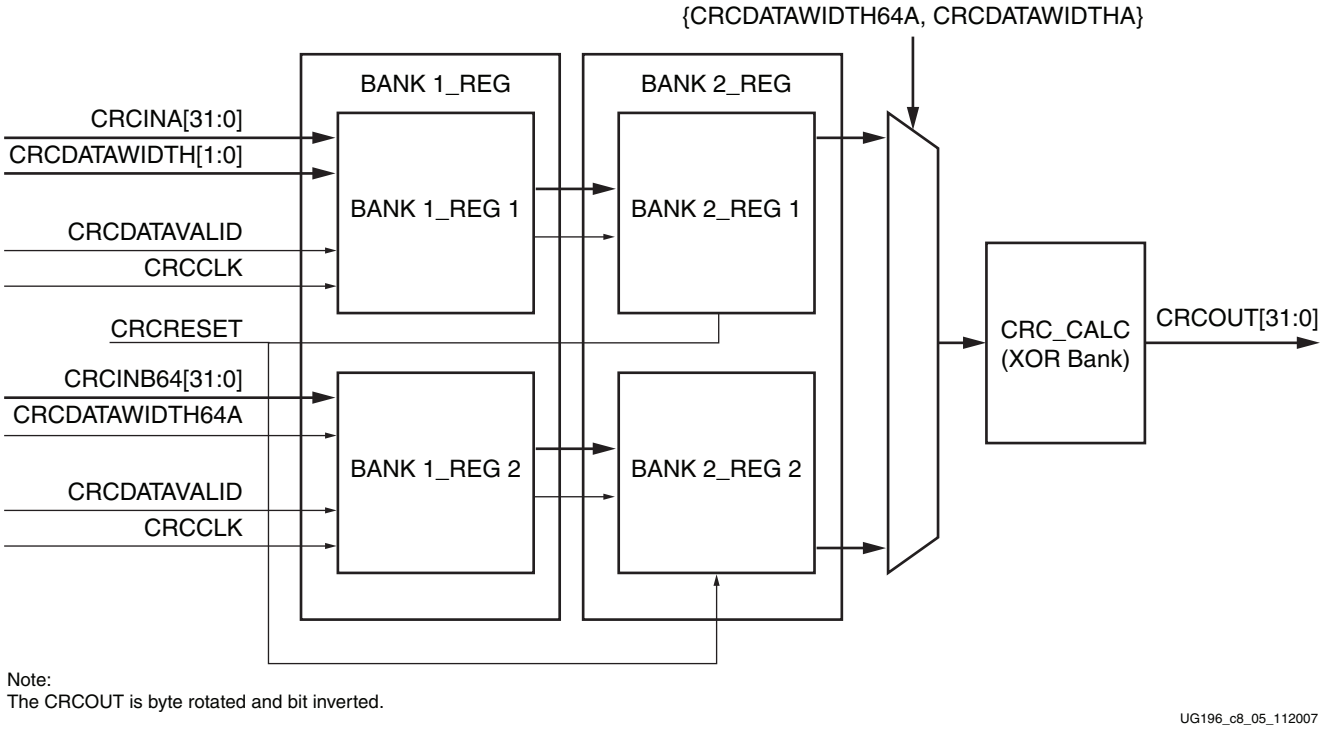


Figure 8-6: **CRC Implementation**

References

Refer to *IEEE 802.3 Cyclic Redundancy Check* [Ref 12] and *Configurable LocalLink CRC Reference Design* [Ref 13] for more information on CRC.

Cover Ex1009
Page 212 of 347

Chapter 9

Loopback

Overview

Loopback modes are specialized configurations of the transceiver datapath where the traffic stream is folded back to the source. Typically, a specific traffic pattern is transmitted and then compared to check for errors. Figure 9-1 illustrates a loopback test configuration with four different loopback modes.

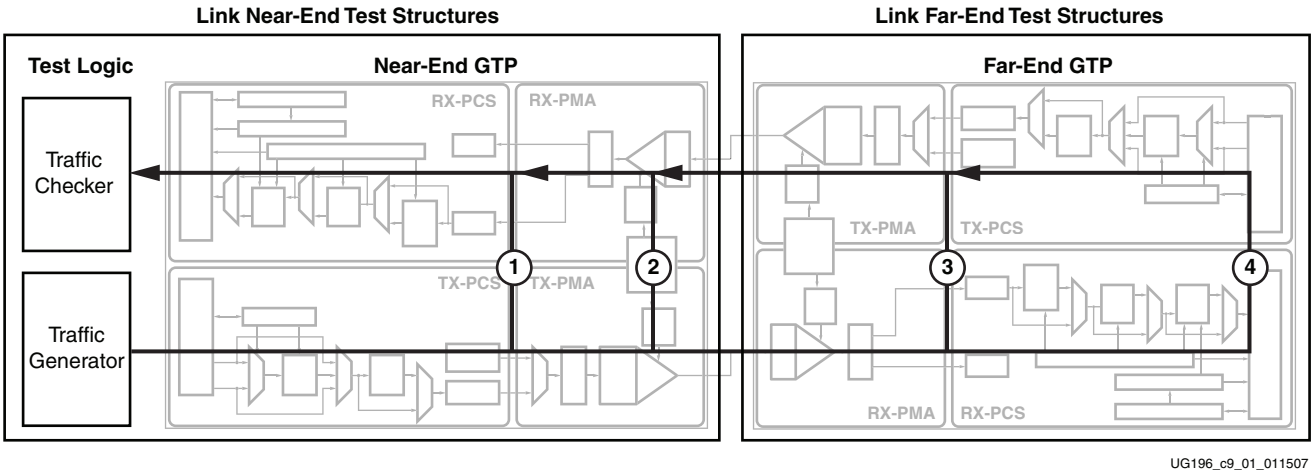


Figure 9-1: **Loopback Testing Overview**

Loopback test modes fall into two broad categories:

- Near-end loopback modes loop transmit data back in the transceiver closest to the traffic generator.
- Far-end loopback modes loop received data back in the transceiver at the far end of the link.

Loopback testing can be used either during development or in deployed equipment for fault isolation. The traffic patterns used can be either application traffic patterns, or specialized pseudo-random bit sequences. Each GTP transceiver has a built-in PRBS generator and checker.

Each channel of the GTP_DUAL tile features several loopback modes to facilitate testing:

- Near-End PCS Loopback (path 1 in Figure 9-1)
- Near-End PMA Loopback (path 2 in Figure 9-1)
- Far-End PMA Loopback (path 3 in Figure 9-1)
- Far-End PCS Loopback (path 4 in Figure 9-1)

The LOOPBACK[2:0] port selects between the normal operation mode and the different loopback modes.

Ports and Attributes

Table 9-1 defines the loopback ports. Table 9-2 defines the loopback attributes.

Table 9-1: **Loopback Ports**

Port	Dir	Clock Domain	Description
LOOPBACK0[2:0] LOOPBACK1[2:0]	In	Async	`000`: Normal operation `001`: Near-End PCS Loopback `010`: Near-End PMA Loopback `011`: Reserved `100`: Far-End PMA Loopback `101`: Reserved `110`: Far-End PCS Loopback(1) `111`: Reserved

Notes:

1. PCI Express compliant.

Table 9-2: **Loopback Attributes**

Attribute	Description
PMA_COM_CFG[89:0]	Common PMA configuration attribute. Leave at the default value automatically set by the RocketIO GTP Transceiver Wizard.

Description

Near-End PCS Loopback

The test data is generated and checked by user logic and then looped back in the PCS. The difference compared to the Near-End PMA Loopback mode is that the PMA section is not involved. The test data is looped back before passing the parallel-to-serial and the serial-to-parallel converter. All analog high-speed circuits in the PMA section can be completely powered down. Figure 9-2 illustrates this configuration.

Apotex Ex1009
Page 214 of 347

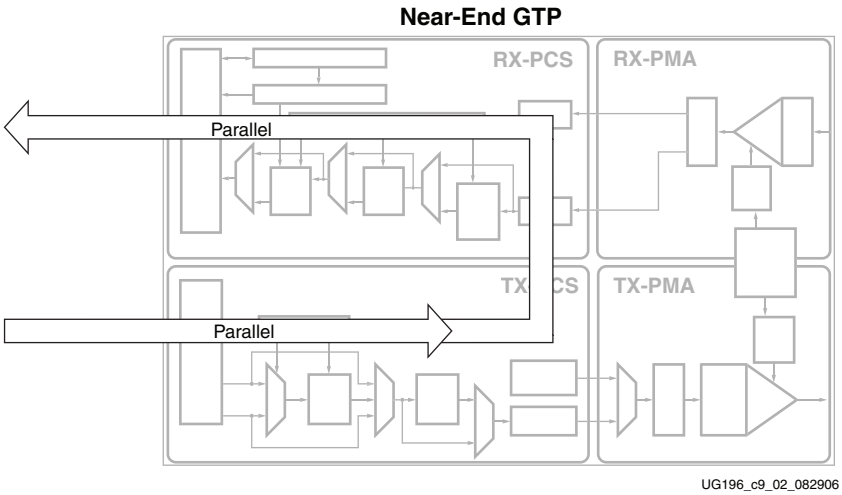


Figure 9-2: **Near-End PCS Loopback**

Near-End PMA Loopback

This mode uses the Near-End source for generating and checking the test data. The loopback occurs in the serial section of the PMA before the line drivers. The test data can be generated and checked by the built-in PRBS block inside the PCS or by user logic. The test data also appears on the package pins. Figure 9-3 shows the configuration without using the built-in PRBS.

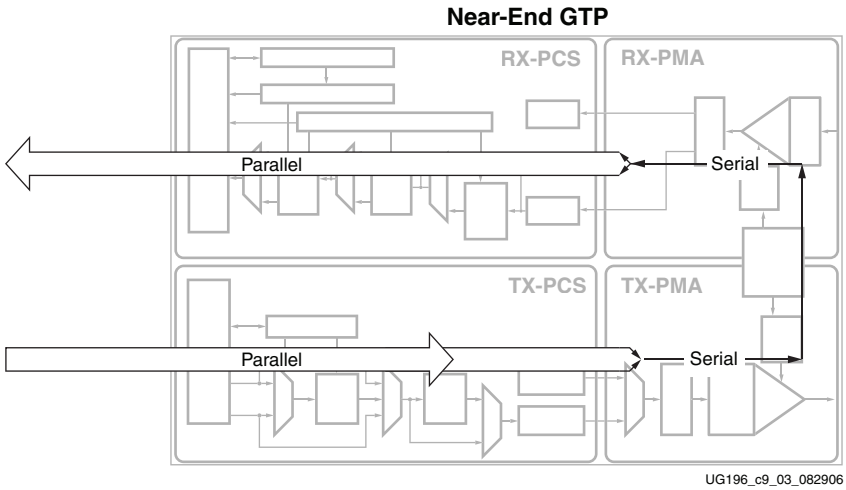


Figure 9-3: **Near-End PMA Loopback**

Marginal Conditions and Limitations

In the Near-End PMA loopback mode, the TXDATA input is routed through the PMA, serialized, and then looped back in through the RX side of the PMA and back out to RXDATA. When this mode is in use, there must be nothing driving the RXP/RXN serial inputs of the channel. The inputs can be left unconnected.

When the device is on a board, the remote transmitter should be 3-stated. If this is not possible, then the following alternatives are available:

- Turn the RX linear equalizer on by driving the RXENEQB(0/1) signal Low.

- Set RXEQMIX(0/1)[1:0] to `11`.
- Change PMA_COM_CFG[44:37] from the default value `00000000` to `10101010`. This setting for PMA_COM_CFG[44:37] must only be used for loopback. With normal data, performance is affected.

The PMA_COM_CFG[44:37] attribute can be changed either statically by specifying the value in the UCF file or dynamically by changing the content of DRP address 26[6:3] from `0000` to `1111`. When switching from loopback mode to normal operation, the default value `00000000` must be restored.

Far-End PMA Loopback

This mode uses the Near-End data source for generating and checking test data. The loopback occurs after passing the serial-to-parallel converter of the PMA. This mode tests the complete PMA section including the serial-to-parallel and parallel-to-serial conversion. Almost the entire PCS section is bypassed because the RX serial inputs are looped back to the TX serial outputs. Figure 9-4 illustrates this mode.

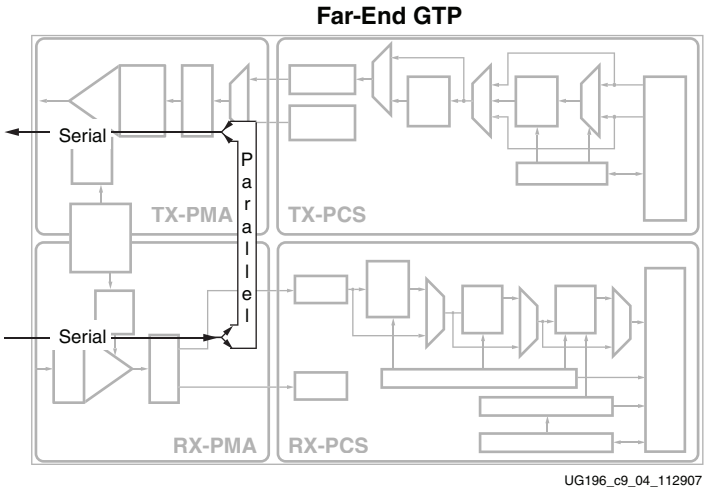


Figure 9-4: **Far-End PMA Loopback**

Marginal Conditions and Limitations

When the Far-End PMA loopback mode is used, ignoring PLL_RXDIVSEL_OUT, both PLL_TXDIVSEL_COMM_OUT and PLL_TXDIVSEL_OUT must be set to `1`. Refer to Shared PMA PLL, page 72 for details about the TX divider settings.

In Far-End PMA loopback mode, the TX buffer is borrowed for the parallel loopback path to compensate for possible phase differences between the TX and RX parallel clocks. For this reason, TX_BUFFER_USE must be TRUE when using this mode. Far-end loopback mode does not support oversampling because the TX buffer is not available. This mode can only be used for one transceiver at a time on any given tile due to clocking restrictions. A transceiver in Far-end PMA loopback mode must use the same reference clock used by the transceiver that is the source of the loopback data.

The transmitter always uses the RX recovered clock to transmit the loopback data. Compared to non-loopback operation, the TX jitter is greater because of greater jitter on the RX recovered clock. The RX CDR can lock to an incoming data stream with a rate offset of up to a nominal ±1000 ppm. The transmitter transmits data with this same rate offset as the RX recovered clock.

Cpuver Ex1009
Page 216 of 347

When one channel of a GTP_DUAL tile is placed in the Far-end PMA loopback mode, the TX side of the other channel no longer operates reliably. Do not attempt nor expect reliable use of the second channel while the first channel is in Far-end PMA loopback.

Far-End PCS Loopback

This mode uses an external source to generate and check the test data. The loopback occurs in the PCS section and the received data is presented to the user logic.

The transmit data of the user logic is not transmitted when this mode is activated. This mode is a PCI Express compliant loopback mode that echoes the received data back to the sender. Figure 9-5 illustrates this configuration.

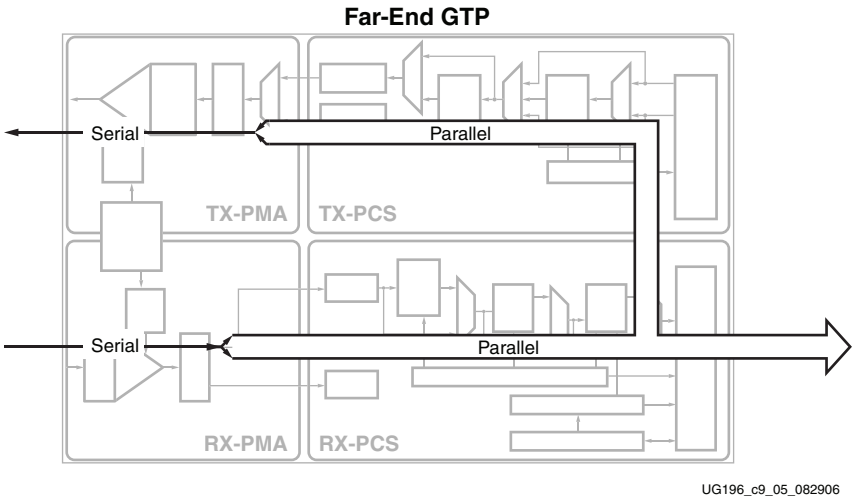


Figure 9-5: **Far-End PCS Loopback**

Note: When the Far-end PCS loopback (also known as PCIe loopback mode) is used, the circuitry retransmits in-table 8B/10B data with the disparity received as required by the PCIe protocol. Clock correction (CC) works properly only if disparity-neutral CC characters are used. If the CC characters are not disparity neutral and CC occurs, the receiver gets disparity errors.

Optiver Ex1009
Page 218 of 347

Chapter 10

GTP-to-Board Interface

Analog Design Guidelines

Overview

In designs with FPGAs that contain GTP transceivers, the overall system performance of a communication link is highly dependent on the characteristics of the power supply and clocking design on both endpoints. This section discusses guidelines and recommendations for these topics.

As a prerequisite, the design guidelines outlined in the *Virtex-5 PCB Designer's Guide* must be observed to keep the power supply and switching noise on the board to a minimum. Additionally, it is highly recommended to use *Point-of-Load* (POL) power distribution techniques as outlined in:
http://www.xilinx.com/publications/xcellonline/xcell_57/xc_pdf/p105-107_57-bellinix.pdf

Refer to the book *EMC and the Printed Circuit Board* [Ref 6] by Mark I. Montrose, sponsored by the IEEE Electromagnetic Compatibility Society for additional guidelines.

Implementation of these guidelines not only improves system margins but is a prerequisite for compliance to regulations as defined by Federal Communications Commission (FCC) and the Verband Deutscher Elektrotechniker (VDE) regarding Electromagnetic Compatibility (EMC), Electromagnetic Interference (EMI), and Radio Frequency Interference (RFI).

Ports and Attributes

Table 10-1 defines the analog pins.

Table 10-1: **Analog Pins**

Pins	Dir	Description
MGTAVCC	In	MGTAVCC is the analog supply for the internal analog circuits of the GTP_DUAL tile.
MGTAVCCPLL	In	MGTAVCCPLL is the analog supply for the shared PMA PLL and the clock routing and muxing network of the GTP_DUAL tile.
MGTAVTTRX	In	MGTAVTTRX is the analog supply for the receiver circuits and termination of the GTP_DUAL tile.
MGTAVTTRXC	In	MGTAVTTRXC is the analog supply for resistor calibration and standby circuit of the entire device.

Optiver Ex1009
Page 219 of 347

Table 10-1: **Analog Pins *(Continued)***

Pins	Dir	Description
MGTAVTTTX	In	MGTAVTTTX is the analog supply for the transmitter termination and driver circuits of the GTP_DUAL tile.
MGTREFCLKP MGTREFCLKN	In	These signals are the differential clock input pin pair(1) for the reference clock of the GTP_DUAL tile.
MGTRREF	In	MGTRREF is the reference resistor input for the entire device.

Notes:

1. This clock can only be accessed by the FPGA logic via the REFCLKOUT port.

Table 10-2 defines the analog attributes.

Table 10-2: **Analog Attributes**

Attribute	Description
TERMINATION_CTRL[4:0]	Controls the internal termination calibration circuit. Refer to Table 10-5, page 224 for encoding.
TERMINATION_IMP_0 TERMINATION_IMP_1	Selects the termination impedance for the TX driver and receiver. See Figure 10-4, page 223 calibrating the impedance values. Always set to 50, which selects the 50Ω termination impedance.
TERMINATION_OVRD	Selects whether the external 50Ω precision resistor connected to the MGTRREF pin or an override value is used, as defined by TERMINATION_CTRL[4:0].

Resistor Calibration Circuit

The resistor calibration circuit in the GTP transceiver is used to precisely calibrate the TX and RX termination resistors via an external precision reference resistor.

Table 10-3 defines the analog ports used for TX and RX termination resistor calibration.

Table 10-3: **Resistor Calibration Circuit Ports**

Pins	Dir	Description
MGTAVTTRXC	In	MGTAVTTRXC is the analog supply for resistor calibration and standby circuit of the entire device.
MGTRREF	In	MGTRREF is the reference resistor input for the entire device.

Apotex Ex1009
Page 220 of 347

Table 10-4 defines the attributes used when calibrating the TX and RX termination resistors.

Table 10-4: **Resistor Calibration Circuit Attributes**

Attribute	Description
TERMINATION_CTRL[4:0]	Controls the internal termination calibration circuit. Refer to Table 10-5, page 224 for encoding.
TERMINATION_IMP_0 TERMINATION_IMP_1	Selects the termination impedance for the TX driver and receiver. See Figure 10-4, page 223 calibrating the impedance values. Always set to 50, which selects the 50Ω termination impedance.
TERMINATION_OVRD	Selects whether the external 50Ω precision resistor connected to the MGTRREF pin or an override value is used, as defined by TERMINATION_CTRL[4:0].

Each Virtex-5 LXT or SXT device requires one 50Ω external precision (1%) resistor on the PCB (connected directly to the MGTRREF pin and to the closest MGTAVTTTX pin).

Figure 10-1 illustrates the connection of the external precision resistor to the MGTRREF and MGTAVTTTX pins of the device. The nominal 50Ω value can be closely matched using a 49.9Ω resistor with 1% tolerance. Each LXT or SXT device requires a filter circuit on the MGTAVTTRXC pin, which powers the resistor calibration circuit of the device.

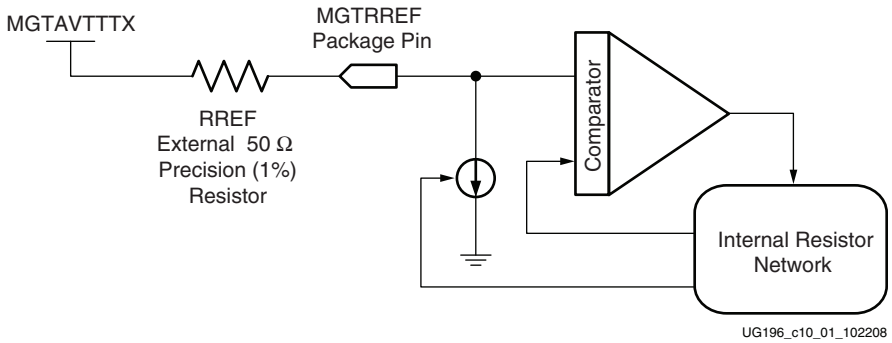


Figure 10-1: **Resistor Calibration Circuit**

Figure 10-2 illustrates the view from the schematic design perspective.

Note: If any GTP transceivers are used in a column, the GTP_DUAL tile, bank 112 that contains the calibration block *must* be connected as illustrated in Figure 10-2.

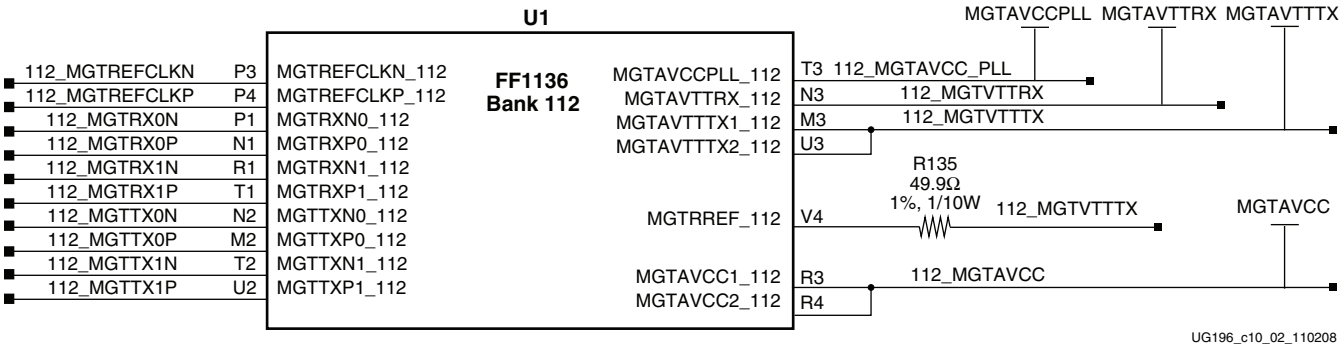
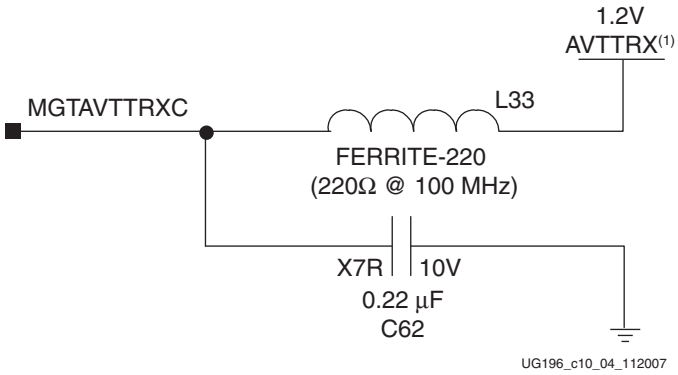


Figure 10-2: **RREF Resistor Schematic Design Perspective**

Figure 10-3 illustrates the power filter network for the MGTAVTTRXC pin.



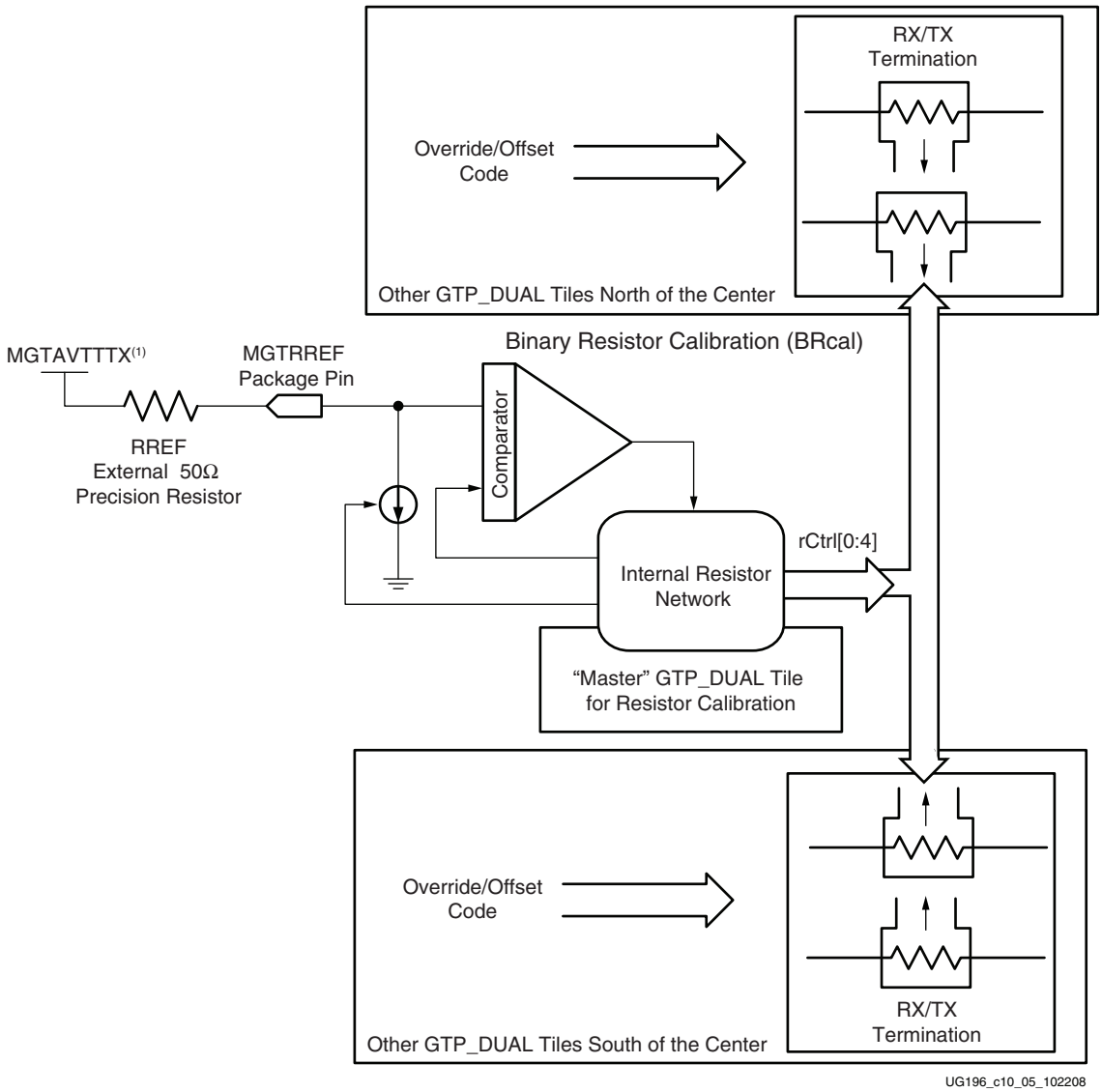
Notes:

1. This analog supply MUST be sourced directly from a dedicated regulator. Derived voltages from other supplies or resistor voltage dividers are NOT permitted.

Figure 10-3: **Power Filter Network for MGTAVTTRXC**

Apple Ex1009
Page 222 of 347

The calculated calibration value of the resistor calibration circuit is shared between all GTP_DUAL primitives of a device (see Figure 10-4).



Notes:

1. This analog supply must be sourced directly from the closest MGTAVTTTX device pin.

Figure 10-4: **Calibration Result Sharing Between All GTP_DUAL Tiles**

The resistor calibration is performed automatically one time during the configuration process. All analog supply voltages must be present and within the proper tolerance as specified in the *Virtex-5 FPGA Data Sheet*.

Note:

- The GTP_DUAL 112 tile contains the calibration circuit for the entire GTP_DUAL column.
- The MGTAVCC supply *must* be powered on all GTP_DUAL tiles between the GTP_DUAL 112 tile and any instantiated tile.
- Filtering of MGTAVCC on unused GTP_DUAL tiles is *not* required.

The calculated calibration value of the resistor calibration circuit can be overwritten independently for each transceiver by using the TERMINATION_CTRL and TERMINATION_OVRD attributes. *This feature is intended for system evaluation purposes only.*

When overriding the control value based on the calibration to an external precision reference resistor, the process independence is lost. As a consequence, the termination value has a tolerance of ± 25% given by the resistor tolerance of the semiconductor process.

Table 10-5 shows the encoding of the TERMINATION_CTRL attribute.

Table 10-5: **TERMINATION_CTRL Attribute Encoding**

TERMINATION_CTRL [4:0]					Nominal Resistance [Ω]	+25% Resistance [Ω]	–25% Resistance [Ω]
0	0	0	0	0	103.0	128.8	77.3
0	0	0	0	1	97.8	122.3	73.4
0	0	0	1	0	93.2	116.4	69.9
0	0	0	1	1	88.9	111.1	66.7
0	0	1	0	0	85.0	106.3	63.8
0	0	1	0	1	81.5	101.9	61.1
0	0	1	1	0	78.2	97.8	58.7
0	0	1	1	1	75.2	94.0	56.4
0	1	0	0	0	72.4	90.5	54.3
0	1	0	0	1	69.8	87.3	52.4
0	1	0	1	0	67.4	84.2	50.5
0	1	0	1	1	65.1	81.4	48.9
0	1	1	0	0	63.0	78.8	47.3
0	1	1	0	1	61.1	76.3	45.8
0	1	1	1	0	59.2	74.0	44.4
0	1	1	1	1	57.5	71.8	43.1
1	0	0	0	0	55.8	69.8	41.9
1	0	0	0	1	54.3	67.8	40.7
1	0	0	1	0	52.8	66.0	39.6
1	0	0	1	1	51.4	64.3	38.6
1	0	1	0	0	50.1	62.6	37.6
1	0	1	0	1	48.8	61.0	36.6
1	0	1	1	0	47.6	59.5	35.7
1	0	1	1	1	46.5	58.1	34.9
1	1	0	0	0	45.4	56.8	34.1
1	1	0	0	1	44.4	55.5	33.3
1	1	0	1	0	43.4	54.2	32.5

Power Ex1009
Page 224 of 347

Table 10-5: **TERMINATION_CTRL Attribute Encoding *(Continued)***

TERMINATION_CTRL [4:0]					Nominal Resistance [Ω]	+25% Resistance [Ω]	–25% Resistance [Ω]
1	1	0	1	1	42.5	53.1	31.8
1	1	1	0	0	41.5	51.9	31.2
1	1	1	0	1	40.7	50.9	30.5
1	1	1	1	0	39.9	49.8	29.9
1	1	1	1	1	39.1	48.8	29.3

Power Supply Design and Filtering

This section focuses on the voltage regulators that directly source each dedicated filter network connected to one of the analog power supply pins of the GTP_DUAL primitive.

A voltage regulator is characterized by:

- Input voltage range
- Output voltage range
- Output voltage current
- Output voltage tolerance
- Output noise voltage
- Power supply rejection ratio (PSRR)

These characteristics are the selection criteria when choosing a voltage regulator for a design with GTP transceivers. The output voltage noise and the PSRR over frequency are often neglected but are very important selection criteria.

As a rule-of-thumb guideline, any substantial noise in the frequency range of 1 MHz and above on the power supply lines contributes to jitter. Depending on the frequency range and amplitude, this noise can degrade the overall system performance. The AVCC supply pins, which source the internal analog circuits of the transceivers, and AVCC_PLL, which sources the shared PMA PLL of the GTP_DUAL primitive, are especially sensitive to power supply noise.

When designing a complete Power Distribution System (PDS), the PSRR of the whole system and of each regulator is load-current and frequency dependent.

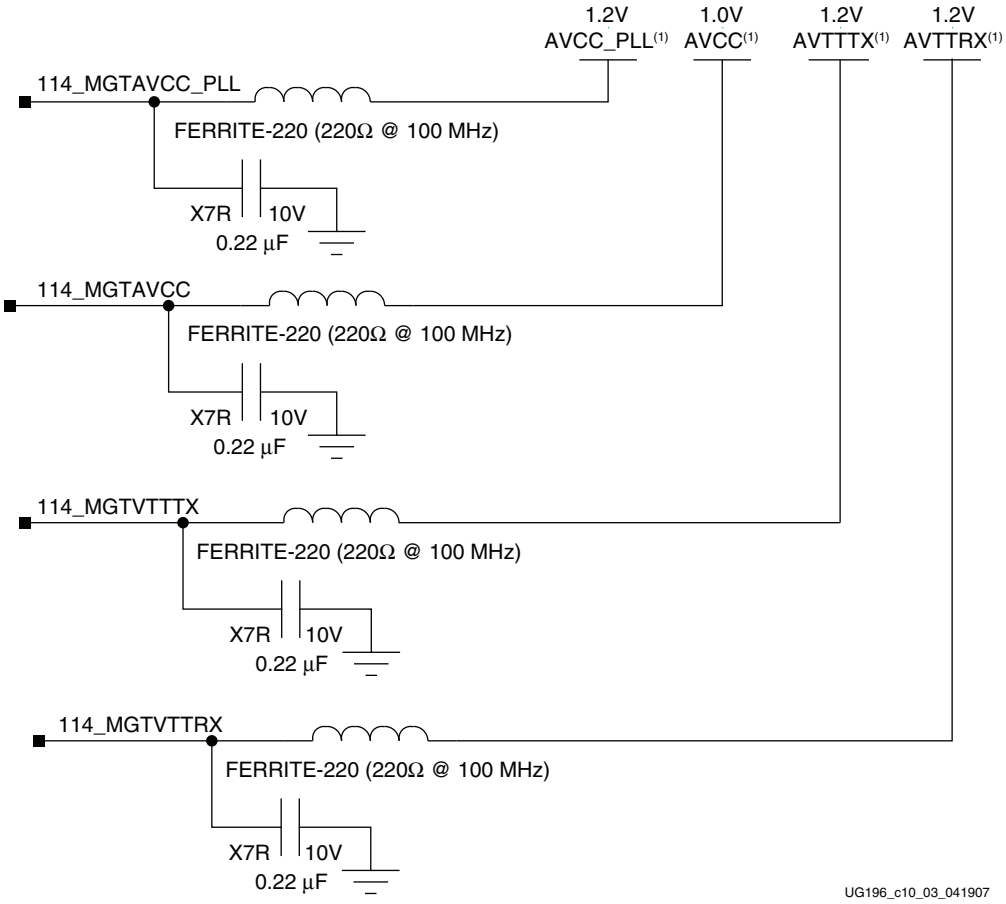
Each used GTP_DUAL tile requires a filter circuit on the following analog power supply pins:

- MGTAVCCPLL
- MGTAVTTTX
- MGTAVTTRX
- MGTAVCC

The correct implementation and placement of these filter circuits is important for suppressing high-frequency noise.

The *Virtex-5 FPGA Data Sheet* provides the required exact voltage level and tolerance ranges of these analog supplies. Adequate filtering must be provided as illustrated in Figure 10-5. Xilinx recommends the use of separate (adjustable) voltage regulators for each supply circuit to facilitate the migration between LXT, SXT, and FXT platforms.

Note: The voltage levels in Figure 10-5 are *nominal* values. Refer to the GTP transceiver portion of the *Virtex-5 FPGA Data Sheet* for the exact values based on operating conditions, especially voltage and temperature.



Notes:

1. These analog supplies MUST be sourced directly from a dedicated regulator. Derived voltages from other supplies or resistor voltage dividers are NOT permitted.

Figure 10-5: **Power Filtering Schematic**

Linear Regulator Selection Criteria

The selection criteria for the linear regulator are:

- Meet or exceed the characteristics specified in the *Virtex-5 FPGA Data Sheet*.
- The PSRR of the linear regulator must provide attenuation in a frequency range where the sourcing power supply or regulator emits noise. Use an adjustable regulator so that the voltage can be changed, if necessary.

Cypher Ex1009
Page 226 of 347

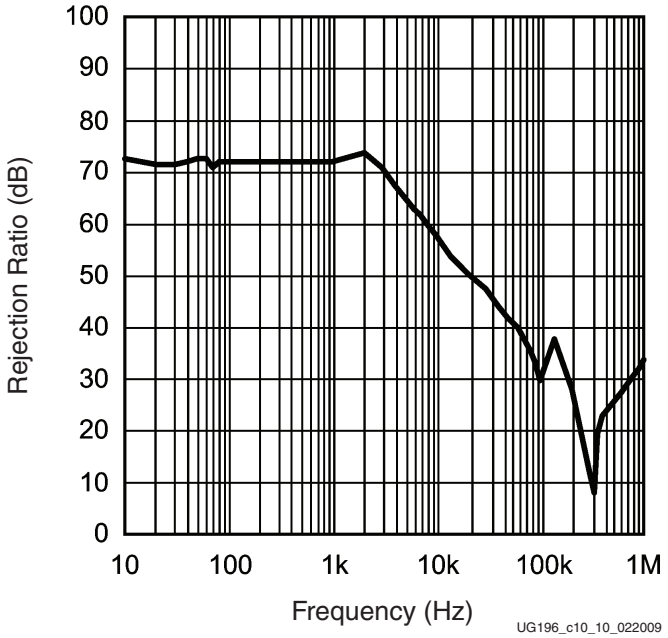


Figure 10-6: **PSRR over the Frequency of a Linear Regulator**

The PSRR over frequency of a linear regulator (see Figure 10-6) is temperature and load-current dependent. Because the PSRR of the regulator in this example has a local minimum of rejection around 300 KHz, extra care must be taken if the sourcing power supply has spurs or high-amplitude noise in this frequency range.

If the sourcing power supply cannot be changed or a different power supply cannot be selected, an additional filter network between the output of the sourcing power supply and the input of the linear regulator is required to prevent substantial noise from passing through the linear regulator at the minimum of attenuation. Because the capacitor on the output of the regulator is part of the regulator control loop, this capacitor not only impacts the regulator stability but the PSRR as well.

Regulator Design Guidelines

The selection criteria for the regulator design are:

- Do not operate the regulator with a V_{IN} that is just slightly over $V_{OUT} + V_{DROPOUT}$.
- Keep in mind that the dropout voltage is highly load dependent.
- Remember that the regulator stability and performance are only guaranteed when using the correct decoupling capacitors (value, ESR, dielectric material) on input, output, and bypass pins of the regulator.
- Strictly follow the layout rules of the regulator data sheet.
- Do not max out the output current of the regulator (allow enough margin under all operating conditions, especially when it comes to high temperature).
- If possible, place the regulator close to the filter network.
- Place the filter network as close as possible to the analog supply pin that it sources.
- Remember that the PSRR of the regulator is output load current and frequency dependent.

Ferrite Selection Guidelines

The selection criteria for the ferrite are:

- Choose a ferrite with a low DC resistance.
- Do not max out the ferrite current rating[(1)].
- Choose a ferrite with a high impedance in a frequency range where you expect or measure the highest spurs or noise levels.

Figure 10-7 illustrates the impedance over frequency characteristics of different ferrites.

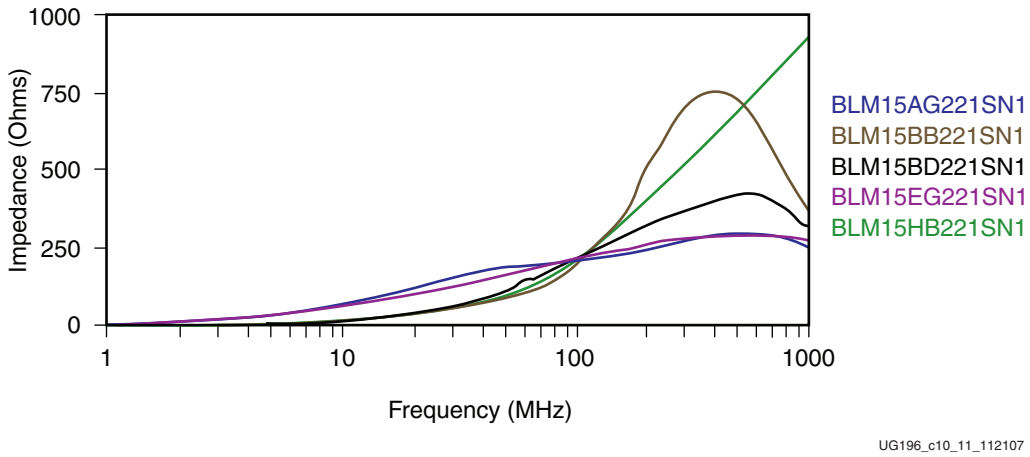


Figure 10-7: **Impedance over Frequency Characteristics of Different Ferrites**

The ferrites are from the same manufacturer, have the same footprint, and have a nominal impedance of 220Ω at 100 MHz. A low DC impedance is required to keep the influence of the load current change to the supply voltage change on the GTP_DUAL power pin to a minimum. Some manufacturers offer S parameters, proprietary-free simulation tools, or

1. If the current rating of a ferrite is maxed out, the magnetic material of the ferrite body is in saturation, which impacts its ability to suppress high frequencies.

Apotex Ex1009
Page 228 of 347

third-party simulation tool support via model libraries for their ferrites and capacitors. These allow the optimization of the filter circuit.

Depending on the spectrum of the noise still on the output pin of the linear regulator, the ferrite with the optimal characteristics must be selected. High-amplitude spurs in the frequency range of 1 MHz and above require special attention.

Capacitor Selection Guidelines

The selection criteria for the capacitor are:

- Low-inductance capacitor
- Dielectric material with a low temperature coefficient
- Dielectric material with a low frequency coefficient

Filter Network Design Guidelines

The selection criteria for the filter network are:

- Place the filter network as close as possible to the device power pin.
- Ensure a low-inductance connection between the capacitor and the power pin.
- Simulate the filter circuit and optimize it, if possible.
- Isolate the analog supply plane between the filter and the FPGA pin. Make sure that no signals can capacitively or inductively couple into this supply.

Boundary-Scan Testing Guidelines

If Boundary-Scan is to be used as part of the product verification, make sure that the analog supply voltage pin MGTAVCC of all GTP_DUAL tiles of the device are powered. The analog supply voltage pin MGTAVCC of all unused GTP_DUAL tiles must be connected to the same supply that supplies V_{CCINT}. V_{CCINT} is the power supply pin for the internal core logic.

GTP Transceiver Power Supply Sharing

To reduce the cost of implementing a design that uses the Virtex-5 FPGA GTP transceivers, some power supply regulators can be shared. A design that intends to share power supplies for the GTP transceivers must adhere to the following rules:

- A power supply regulator used by the GTP transceivers must not be shared with any digital circuit. Events such as reset can cause significant load transients on a power supply that will impact the performance of the GTP transceivers.
- A GTP transceiver power supply (e.g., MGTAVCCPLL) can be shared between different GTP_DUAL tiles and between FPGAs. Because the load current on the regulator and the power distribution network is greater with shared supplies, care must be taken to compensate for any IR drop so that the voltage at the FPGA pin(s) complies with the acceptable operating voltage range defined in the *Virtex-5 FPGA Data Sheet*.
- MGTAVTTRX and MGTAVTTTX can share the same power supply voltage regulator across multiple GTP_DUAL tiles and across multiple FPGAs.
- MGTAVCC and MGTAVCCPLL can be shared across multiple GTP_DUAL tiles and across multiple FPGAs.

Providing Power

Overview

This section focuses on the various configurations for powering the GTP transceivers for optimal power consumption and minimizing board real estate for filtering schemes.

Description

Various use cases for powering the GTP transceivers are described here. The specific rules used to derive these cases are described in the Power Supply Design and Filtering and Resistor Calibration Circuit sections.

Note: MGTAVTTRXC and MGTRREF are column specific, not dual specific. They are repeated in all the use cases for completeness.

Apple Ex1009
Page 230 of 347

Partially Used Column

Table 10-6 shows the configuration for Use Case 1, where:

- a GTP_DUAL tile is used
- one or both transceivers are used
- Boundary-Scan is always functional

Table 10-6: **Use Case 1**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND(1) or transmitter	-
MGTTXP/MGTTXN	Floating, no connection(2) or receiver	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection(3) or reference clock source	-
MGTAVTTTX	1.2V dedicated supply(4)	Y
MGTAVTTRX	1.2V dedicated supply(4)	Y
MGTAVTTRXC	MGTAVTTRX(5)	Y
MGTAVCCPLL	1.2V dedicated supply(4)	Y
MGTAVCC	1.0V dedicated supply(4)	Y
MGTRREF	MGTAVTTTX with resistor(6)	-

Notes:

1. If only a single transceiver is used, connect the pins of the unused transceiver to GND.
2. If only single transceiver is used, leave the pins of the unused transceiver floating.
3. If the reference clock input is unused, leave the pins floating.
4. Refer to Figure 10-5, page 226.
5. Refer to Figure 10-3, page 222.
6. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Table 10-7 shows the configuration for Use Case 2, where:

- a GTP_DUAL tile is unused
- both transceivers are unused
- Boundary-Scan is not functioning

Table 10-7: **Use Case 2**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	GND	-
MGTAVTTRX	GND	-
MGTAVTTRXC	MGTAVTTRX(1)	Y
MGTAVCCPLL	GND	-
MGTAVCC	GND	-
MGTRREF	MGTAVTTTX with resistor(2)	-

Notes:

1. Connect to MGTAVTTRX of a used GTP_DUAL tile. Refer to Figure 10-3, page 222.
2. Connect to MGTAVTTTX of a used GTP_DUAL tile. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Optiver Ex1009
Page 231 of 347

Table 10-8 shows the configuration for Use Case 3, where:

- a GTP_DUAL tile is unused
- both transceivers are unused
- Boundary-Scan is functioning

Table 10-8: **Use Case 3**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	GND	-
MGTAVTTRX	GND	-
MGTAVTTRXC	MGTAVTTRX(1)	Y
MGTAVCCPLL	GND	-
MGTAVCC	V_{CCINT}	-
MGTRREF	MGTAVTTTX with resistor(2)	-

Notes:

1. Connect to MGTAVTTRX of a used GTP_DUAL tile. Refer to Figure 10-3, page 222.
2. Connect to MGTAVTTTX of a used GTP_DUAL tile. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Table 10-9 shows the configuration for Use Case 4, where:

- a GTP_DUAL tile is used only for forwarding resistor calibration information generated in the center tile of the column
- both transceivers are unused
- Boundary-Scan is always functional

Table 10-9: **Use Case 4**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	GND	-
MGTAVTTRX	GND	-
MGTAVTTRXC	MGTAVTTRX(1)	Y
MGTAVCCPLL	GND	-
MGTAVCC	V_{CCINT}	-
MGTRREF	MGTAVTTTX with resistor(2)	-

Notes:

1. Connect to MGTAVTTRX of a used GTP_DUAL tile. Refer to Figure 10-3, page 222.
2. Connect to MGTAVTTTX of a used GTP_DUAL tile. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Apotex Ex1009
Page 232 of 347

Table 10-10 shows the configuration for Use Case 5, where:

- a GTP_DUAL tile is used only for forwarding resistor calibration information generated in the center tile of the column and for forwarding a reference clock
- both transceivers are unused
- Boundary-Scan is always functional

Table 10-10: **Use Case 5**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	GND	-
MGTAVTTRX	GND	-
MGTAVTTRXC	MGTAVTTRX(1)	Y
MGTAVCCPLL	1.2V dedicated supply(2)	Y
MGTAVCC	V_{CCINT}	-
MGTRREF	MGTAVTTTX with resistor(3)	-

Notes:

1. Connect to MGTAVTTRX of a used GTP_DUAL tile. Refer to Figure 10-3, page 222.
2. Refer to Figure 10-5, page 226.
3. Connect to MGTAVTTTX of a used GTP_DUAL tile. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Table 10-11 shows the configuration for Use Case 6, where:

- a GTP_DUAL tile is used only for forwarding a reference clock
- both transceivers are unused
- Boundary-Scan always functional

Note: This Use Case is not recommended because it unnecessarily causes an extra tile to be used.

Table 10-11: **Use Case 6**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	GND	-
MGTAVTTRX	GND	-
MGTAVTTRXC	MGTAVTTRX(1)	Y
MGTAVCCPLL	1.2V dedicated supply(2)	Y
MGTAVCC	V_{CCINT}	-
MGTRREF	MGTAVTTTX with resistor(3)	-

Notes:

1. Connect to MGTAVTTRX of a used GTP_DUAL tile. Refer to Figure 10-3, page 222.
2. Refer to Figure 10-5, page 226.
3. Connect to MGTAVTTTX of a used GTP_DUAL tile. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Table 10-12 shows the configuration for Use Case 7, where:

- a GTP_DUAL tile is used only as the master resistor calibration circuit and/or for clock forwarding
- both transceivers are unused
- Boundary-Scan is always functional

Table 10-12: **Use Case 7**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	1.2V dedicated supply(1)	Y
MGTAVTTRX	GND	-
MGTAVTTRXC	MGTAVTTRX(2)	Y
MGTAVCCPLL	1.2V dedicated supply(1)	Y
MGTAVCC	V_{CCINT}	-
MGTRREF	MGTAVTTTX with resistor(3)	-

Notes:

1. Refer to Figure 10-5, page 226.
2. Connect to MGTAVTTRX of a used GTP_DUAL tile. Refer to Figure 10-3, page 222.
3. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Fully Unused Column

Table 10-13 shows the configuration for Use Case 8, where:

- a GTP_DUAL tile is unused
- both transceivers are unused
- Boundary-Scan is not functioning

Table 10-13: **Use Case 8**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	GND	-
MGTAVTTRX	GND	-
MGTAVTTRXC	Floating, no connection	-
MGTAVCCPLL	GND	-
MGTAVCC	GND	-
MGTRREF	GND	-

Apotex Ex1009
Page 234 of 347

Table 10-14 shows the configuration for Use Case 9, where:

- a GTP_DUAL tile is unused
- both transceivers are unused
- Boundary-Scan is functioning

Table 10-14: **Use Case 9**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND	-
MGTTXP/MGTTXN	Floating, no connection	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection	-
MGTAVTTTX	GND	-
MGTAVTTRX	GND	-
MGTAVTTRXC	Floating, no connection	-
MGTAVCCPLL	GND	-
MGTAVCC	V_{CCINT}	-
MGTRREF	GND	-

Fully Used Column

Table 10-15 shows the configuration for a fully used column, where:

- a GTP_DUAL tile is used
- one or both transceivers are used
- Boundary-Scan is always functional

Use Case 10 is the same as Use Case 1 described in the Partially Used Column section.

Table 10-15: **Use Case 10**

Pin or Pin Pair	Connect To	Filter
MGTRXP/MGTRXN	GND(1) or transmitter	-
MGTTXP/MGTTXN	Floating, no connection(2) or receiver	-
MGTREFCLKP/MGTREFCLKN	Floating, no connection(3) or reference clock source	-
MGTAVTTTX	1.2V dedicated supply(4)	Y
MGTAVTTRX	1.2V dedicated supply(4)	Y
MGTAVTTRXC	MGTAVTTRX(5)	Y
MGTAVCCPLL	1.2V dedicated supply(4)	Y
MGTAVCC	1.0V dedicated supply(4)	Y
MGTRREF	MGTAVTTTX with resistor(6)	-

Notes:

1. If only a single transceiver is used, connect the pins of the unused transceiver to GND.
2. If only a single transceiver is used, leave the pins of the unused transceiver floating.
3. If the reference clock input is unused, leave the pins floating.
4. Refer to Figure 10-5, page 226.
5. Refer to Figure 10-3, page 222.
6. Refer to Figure 10-1, page 221 and Figure 10-4, page 223.

Example

The example shown in Figure 10-8 is a partially used column that applies the Use Cases discussed in this section.

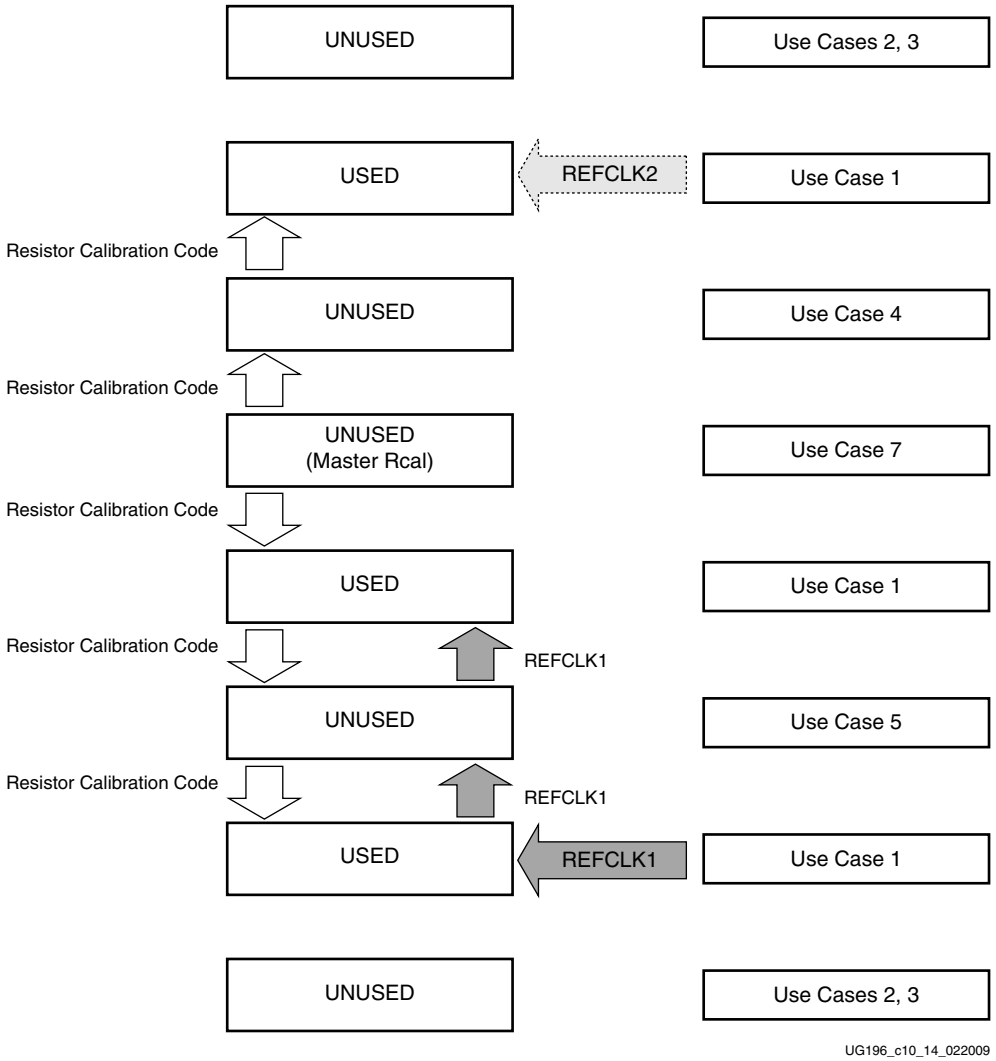


Figure 10-8: **Use Case Example**

REFCLK Guidelines

Overview

This section focuses on the selection of the reference clock source or oscillator. An oscillator is characterized by:

- Frequency range
- Output voltage swing
- Jitter (deterministic, random, peak-to-peak)
- Rise and fall times

Apotex Ex1009
Page 236 of 347

- Supply voltage and current
- Noise specification
- Duty cycle and duty-cycle tolerance
- Frequency stability

These characteristics are selection criteria when choosing an oscillator for a GTP transceiver design.

Figure 10-9 illustrates the convention for the single-ended clock input voltage swing, peak-to-peak as used in the GTP transceiver portion of the *Virtex-5 FPGA Data Sheet*.

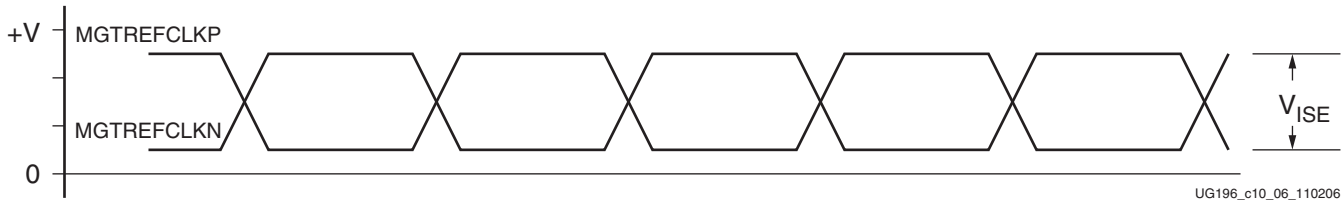


Figure 10-9: **Single-Ended Clock Input Voltage Swing, Peak-to-Peak**

Figure 10-10 illustrates the differential clock input voltage swing, peak-to-peak, which is defined as MGTREFCLKP – MGTREFCLKN.

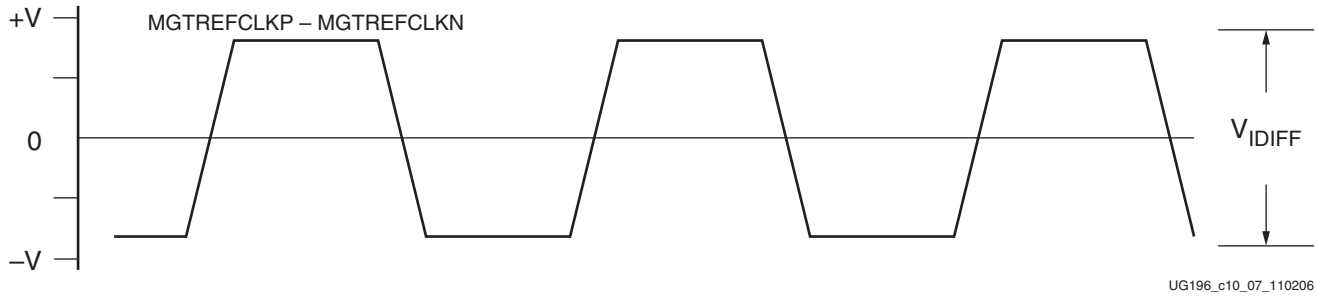


Figure 10-10: **Differential Clock Input Voltage Swing, Peak-to-Peak**

Figure 10-11 shows the rise and fall time convention of the reference clock.

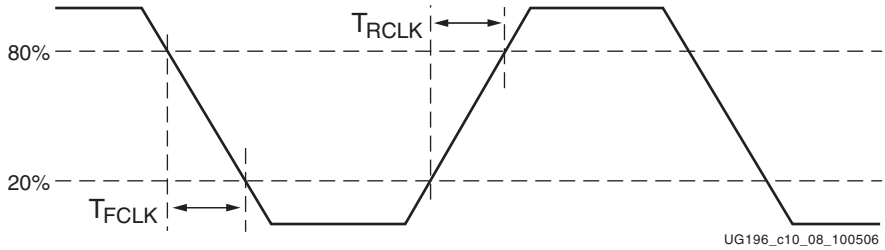
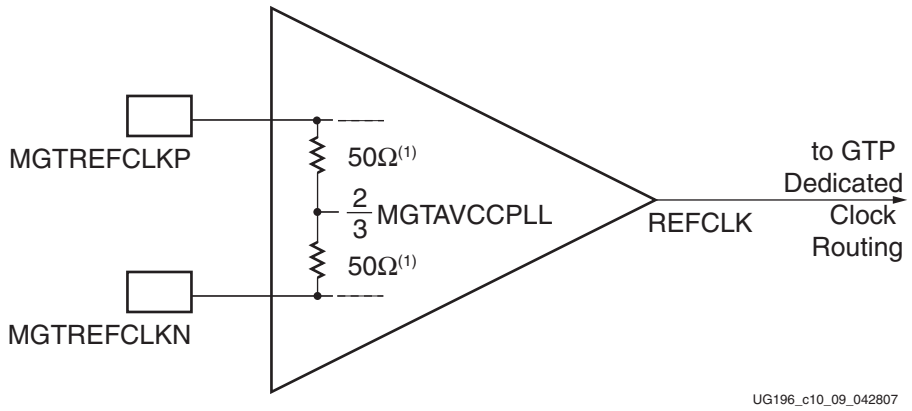


Figure 10-11: **Rise and Fall Times**

Figure 10-12 illustrates the internal details of the IBUFDS. The dedicated differential reference clock input pair MGTREFCLKP/MGTREFCLKN is internally terminated with 100Ωdifferential impedance. The common mode voltage of this differential reference clock input pair is 2/3 of MGTAVCCPLL.



Notes:

1. Nominal values. Refer to the *Virtex-5 FPGA Data Sheet* for exact specifications.

Figure 10-12: **IBUFDS Details**

If the common mode voltage of the driving clock source is different from the common mode voltage of the differential reference clock input pair, then AC coupling capacitors are mandatory to prevent device degradation and/or other damage.

Opiver Ex1009
Page 238 of 347

GTP Reference Clock Checklist

The following criteria must be met when choosing an oscillator for a design with GTP transceivers:

- Provide AC coupling between the oscillator output pins and the dedicated GTP_DUAL clock input pins.
- Ensure that the differential voltage swing of the reference clock is the range as specified in the *Virtex-5 FPGA Data Sheet* (the nominal range is 200 mV – 2000 mV, and the nominal typical value is 1200 mV).
- Meet or exceed the reference clock characteristics as specified in the *Virtex-5 FPGA Data Sheet*.
- Meet or exceed the reference clock characteristics as specified in the standard for which the GTP transceiver provides physical layer support.
- Fulfill the oscillator vendor's requirement regarding power supply, board layout, and noise specification.
- Provide a dedicated point-to-point connection between the oscillator and GTP_DUAL clock input pins.
- Keep impedance discontinuities on the differential transmission lines to a minimum (impedance discontinuities generate jitter).
- Any GTP_DUAL tile that sources a reference clock must be instantiated and REFCLKPWRDNB must be asserted High.

Description

Oscillator Selection

Selecting an oscillator and designing a clock distribution system requires a careful selection of components as well as a proper board layout to ensure that overall system requirements are met.

When designing a clocking system for a design with a GTP transceiver, the requirements of the implemented standard (Ethernet, OC-48, SDI, etc.) as well as the requirements given in the GTP Transceiver section of the *Virtex-5 FPGA Data Sheet* must be fulfilled. However, the requirements for the GTP transceiver reference clock as specified in the *Virtex-5 FPGA Data Sheet* must be met or exceeded. Under these conditions, the GTP transceiver was characterized as specified in the *Virtex-5 FPGA Data Sheet*.

The differential clock input of the GTP_DUAL primitive requires AC coupling capacitors between the oscillator output pins and the dedicated clock input pins of the Virtex-5 device.

Sourcing More Than One Differential Clock Input Pair from One Oscillator

If a clock needs to be shared between more than seven GTP_DUAL primitives of a Virtex-5 device, more than one differential clock input pair is required. Either an oscillator with multiple outputs or a single output oscillator and a multi-output clock buffer is required.

The connection between the dedicated clock input pin pair of a GTP_DUAL primitive and the outputs of the oscillator or buffer MUST be a point-to-point connection. Bifurcated transmission lines, "T-stubs", branches, and daisy chaining are *not* permitted.

Optiver Ex1009
Page 239 of 347

Switching between Two Different Reference Clocks

There are two ways to implement a design that needs to operate at two different clock rates (for example, in an HD-SDI video application):

1. Use the DRP to switch between two different clocks that are sourced from two different GTP_DUAL reference clock pins to the dedicated clock routing of the GTP_DUAL column.
2. Use an external clock multiplexer with one or multiple outputs.

The first solution is limited to up to four GTP_DUAL primitives and therefore to a maximum of eight GTP transceivers. In this configuration, one clock is sourced from the top and one clock is sourced from the bottom of four GTP_DUAL primitives, which are direct neighbors to each other.

The second solution can be used for up to 7 GTP_DUAL primitives and therefore to a maximum of 14 GTP transceivers, if an external multiplexer with one output is used. In this configuration, the GTP_DUAL primitive that sources the clock is located in the middle of seven GTP_DUAL primitives that are direct neighbors to each other. If a clock multiplexer with *n* outputs is used, this solution can be expanded up to *n* times 7 GTP_DUAL primitives and therefore up to *n* times 14 GTP transceivers.

AC Coupling

AC coupling of the oscillator reference clock output to the GTP_DUAL reference clock inputs serves multiple purposes:

- Blocking a DC current between the oscillator and the GTP_DUAL dedicated clock input pins (which reduces the power consumption of both parts as well)
- Common mode voltage independence
- The AC coupling capacitor forms a high-pass filter with the on-chip termination that attenuates a wander[(1)] of the reference clock

To minimize noise and power consumption, external AC coupling capacitors between the sourcing oscillator and the GTP_DUAL dedicated clock reference clock input pins are required.

Unused Reference Clock Inputs of GTP_DUAL Tiles for Clock Forwarding

It is recommended to connect the unused differential input pin clock pair to ground or leave it floating (both MGTREFCLKP and MGTREFCLKN).

1. A wander is low-frequency jitter.

Cypher Ex1009
Page 240 of 347

Examples of Vendors and Devices

The alphabetical list of vendors and devices in Table 10-16 is intended to facilitate the search for parts.

Note: This table does *not* recommend, endorse, or verify the parts list!

Table 10-16: **Vendor and Device Examples**

Vendor	Website	Products	Examples	Remarks
Analog Devices	http://www.analog.com	Voltage regulators		
Epson	http://www.eea.epson.com	Oscillators	EG-2121CA (53.125 - 500 MHz oscillator)	
Integrated Systems	http://www.icst.com	Oscillators, buffers, PLLs	HiPer clock family	
Linear Technology	http://www.linear.com	Voltage regulators	LTC3026 (1.5A, 0.4 - 2.6V adjustable)	Provides SPICE models for some regulators, free simulation tools
Maxim	http://www.maxim-ic.com	Voltage regulators		
Micrel	http://www.micrel.com	Oscillators, clock buffers	SY100EP14U (1:5 driver with 2:1 MUX)	
Murata	http://www.murata.com	EMI suppression, ferrite beads, chip capacitors		Provides S parameters, design libraries for Signal integrity tools, and free simulation tools for their components (http://www.murata.com/designlib/index.html)
National Semiconductor	http://www.national.com	Voltage regulators	LP-3878-ADJ	
ON Semiconductor	http://www.onsemi.com	Voltage regulators	NCP5663 (3.0 A, 0.9Vmin, adjustable)	
Silicon Laboratories	http://www.silabs.com	Oscillators, clock multipliers, jitter attenuators	Si530 family (10 - 1400 MHz)	
TDK	http://www.component.tdk.com	Ferrite beads, capacitors		Provides component S parameters for simulation support, design libraries for Signal Integrity tools (http://www.component.tdk.com/tvcl_sparam.php)
Texas Instruments	http://www.ti.com	Voltage regulators, PLLs, buffer		

Table 10-16: **Vendor and Device Examples *(Continued)***

Vendor	Website	Products	Examples	Remarks
Vectron	http://www.vectron.com	Oscillators		
X2Y Attenuators, LLC	http://www.x2y.com	Low inductance capacitors		

SelectIO to GTP Crosstalk Guidelines

Because a GTP transceiver's performance can degrade in an environment flooded with SelectIO™ activity, it is important to have guidelines for SelectIO usage that minimize the impact on GTP transceiver performance.

Although the Virtex-5 FPGA package exhibits little package-related crosstalk issues, the pinout of the device might lead to customer designs becoming susceptible to PCB-via related crosstalk issues. The near proximity of SelectIO signals (aggressor) to GTP transceiver analog supplies (victim) results in their PCB via structures being placed in close proximity as well. This ball adjacency and resulting via adjacency creates a via-coupling region between the SelectIO signals and the GTP transceiver analog supplies that is not filtered by on-board power supply filtering. The amount of crosstalk voltage induced on the victim circuit by the aggressor circuit is equal to the rate of change of current in the aggressor times the mutual inductance shared between the two circuits. For an in-depth discussion on via crosstalk and calculations of mutual inductance for various via configurations, refer to *High-Speed Signal Propagation: Advanced Black Magic* by Howard Johnson and Martin Graham [Ref 5]. The sensitivity of the GTP transceiver analog supplies to coupled noise from the PCB results in a degradation of GTP transceiver performance. The MGTAVCC and MGTAVCCPLL supplies are most sensitive to coupled noise.

To minimize the impact on GTP transceiver performance, these BGA adjacency guidelines must be followed:

Note: The BGA adjacency guidelines must be followed as well for every package device combination not listed in Table 10-17 through Table 10-23.

- Avoid utilizing SelectIO nets 1.0 mm (horizontal or vertical) or 1.4 mm (diagonal) away from GTP transceiver analog power supply pins. Ground these SelectIO locations in the PCB, and set the SelectIO output to the highest drive and a forced-Low setting. If these SelectIO outputs must be used, use them either in differential signaling applications or for static control/status signals with low speed and low drive.
- Avoid using a large number of SelectIO signals in I/O banks near GTP transceivers. See Table 10-17 for specific aggressive I/O bank to GTP transceiver pairing.

Table 10-17: **Aggressive I/O Banks**

GTP_DUAL	FF665	FF1136	FF1738
MGT112	12	12	12
MGT114	12	18	18
MGT116	12/16	12	12
MGT118	12/18	18	18/26
MGT120		12/20	20

Apotex Ex1009
Page 242 of 347

Table 10-17: **Aggressive I/O Banks *(Continued)***

GTP_DUAL	FF665	FF1136	FF1738
MGT122		22	18/26
MGT124		20	
MGT126		22	26
MGT128			20/24
MGT130			34
MGT132			24
MGT134			34

- If these SelectIO pins must be used for higher drive/higher speed applications, apply power to the GTP transceiver analog supplies with a plane or wide buses a few layers below the top of the board. Using a blind via to the GTP transceiver analog supplies is better than using a through via. Shield these supply planes or buses with ground planes above and below.
- If a through via to supply the GTP transceiver analog supply pins must be used, use a layer closest to the FPGA for routing signals to these vias. Route SelectIO nets in the uppermost layer available after GTP transceiver high-speed signal and GTP transceiver analog supply routing is implemented.
- If supplying GTP transceiver power from the bottom of the board, route these SelectIO nets in the highest available routing layer.

The absolute worst-case scenario is to supply GTP transceiver analog supplies from the bottom of the board and have all adjacent SelectIO outputs running at high drive and high speed and routed to lower routing layers. For more information, refer to BGA Escape Example, page 278 for information on escaping of SelectIO nets adjacent to GTP transceiver analog supply pins.

The SelectIO signals that have the largest impact on GTP transceiver performance are those whose solder balls are adjacent to GTP transceiver analog supply solder balls (BGA adjacency). Table 10-18 through Table 10-22 provide the GTP transceiver user with pin-specific guidance recommendations to optimize GTP transceiver performance in the presence of SelectIO switching. Specifically, the tables identify those pins which are either 1.0 mm or 1.4 mm away from an GTP transceiver analog supply pin and REFCLK pins. If a pin is both 1.0 mm and 1.4 mm away from two different GTP transceiver analog supply pins, then it is only listed in the 1.0 mm column.

Additionally, it is important to properly shield the traces or planes connecting the analog supply vias to the supply filter network of the analog supplies from SelectIO signals escaping from the BGA field. This is best done by placing GND planes above and below the layer containing the analog traces or planes.

Table 10-18: **SelectIO Net Adjacent to Analog Supplies (FF665 Packages)**

GTP_DUAL Tile	1 mm	1.4 mm
MGT116	E5	D5, F5, G4
MGT112	L5	K5
MGT114	U5	T5, W4
MGT118	AD4	AB5, AD5

Table 10-19: **SelectIO Net Adjacent to MGTCLK (FF665 Packages)**

GTP_DUAL Tile	1 mm	1.4 mm
116_REFCLK	D5	E5
112_REFCLK	K5	J5, L5
114_REFCLK	T5	R5, U5
118_REFCLK	AB5	AA5

Table 10-20: **SelectIO Net Adjacent to Analog Supplies (FF1136 Packages)**

GTP_DUAL Tile	1 mm	1.4 mm
MGT124(1)	E7	E8, E6
MGT120	F5	E6, G5
MGT116	J5	H5, L4
MGT112	–	P5
MGT114	AA5	AB5, AC4
MGT118	AG5	AF5, AH5
MGT122	AK6	AH5, AJ6
MGT126(1)	AK8	AK7, AK9, AL10

Notes:

1. The GTP_DUAL tile is only available on XC5VSX95T, XC5VLX110T, and XC5VLX155T devices.

Table 10-21: **SelectIO Net Adjacent to MGTCLK (FF1136 Packages)**

GTP_DUAL Tile	1 mm	1.4 mm
124_REFCLK(1)	E8	E7, E9
120_REFCLK	F5	–
116_REFCLK	H5	G5, J5
112_REFCLK	P5	N5
114_REFCLK		AA5
118_REFCLK	AF5	AG5
122_REFCLK	–	AK6
126_REFCLK(1)	AK7	AK6, AK8

Notes:

1. The GTP_DUAL tile is only available on XC5VSX95T, XC5VLX110T, and XC5VLX155T devices.

Apple v. Ex1009
Page 244 of 347

Table 10-22: **SelectIO Net Adjacent to Analog Supplies (FF1738 Packages)**

GTP_DUAL	1 mm	1.4 mm
MGT132(1)	E15	E14, D13
MGT128(1)	E9	E10, E8, D7
MGT124	E5	
MGT120		F5, H5
MGT116	N5	P5, R4
MGT112	W5	V5
MGT114	AE5	AD5, AF5, AG4
MGT118	AL5	AK5, AN4
MGT122		AT5, AV5
MGT126	AV5	AV6
MGT130(1)	AV10	AV9, AV11, AW12
MGT134(1)	AV16	AV15, AW18

Notes:

1. The GTP_DUAL tile is only available on XC5VLX330T devices.

Table 10-23: **SelectIO Net Adjacent to MGTCLK (FF1738 Packages)**

GTP_DUAL	1 mm	1.4 mm
132_REFCLK(1)		E17, E15
128_REFCLK(1)	E10	E9
124_REFCLK		
120_REFCLK	F5	E5
116_REFCLK		L5, N5
112_REFCLK	V5	W5
114_REFCLK	AD5	AC5, AE5
118_REFCLK	AK5	AJ5, AL5
122_REFCLK	AT5	AR5
126_REFCLK		AV5
130_REFCLK(1)	AV9	AV8, AV10
134_REFCLK(1)	AV15	AV14, AV16

Notes:

1. The GTP_DUAL tile is only available on XC5VLX330T devices.

Apple Ex1009
Page 246 of 347

Section 2: Board Level Design

This section describes general design guidelines when designing systems and boards where the interconnect exhibits transmission line behavior. This situation occurs when signals have rise and/or fall times smaller than 2.5 times the flight time from one end of the interconnection to the other end. These guidelines also apply to all designs with high-speed transceivers.

These guidelines have been used to create boards that have successfully operated at serial transmission rates in excess of 10 Gb/s. Although designing for 10 Gb/s operation can seem excessive when the application calls for slower speeds, it is better to design knowing that constraints can be relaxed for slower speeds if needed. Other interfaces with challenging signal integrity demands, such as high-speed memory interfaces, also benefit from the approach used for 10 Gb/s design.

This section includes the following chapters:

Design Constraints Overview

PCB Materials and Traces

Design of Transitions

Guidelines and Examples

Optiver Ex1009
Page 247 of 347

Oplover Ex1009
Page 248 of 347

Chapter 11

Design Constraints Overview

Figure 11-1 shows a typical physical interconnect topology between two transceivers. Any physical link connecting two point-to-point high-speed serial transceivers is defined as a channel. A channel begins at the die solder bumps of the transmitter and ends at the die solder bumps of the receiver.

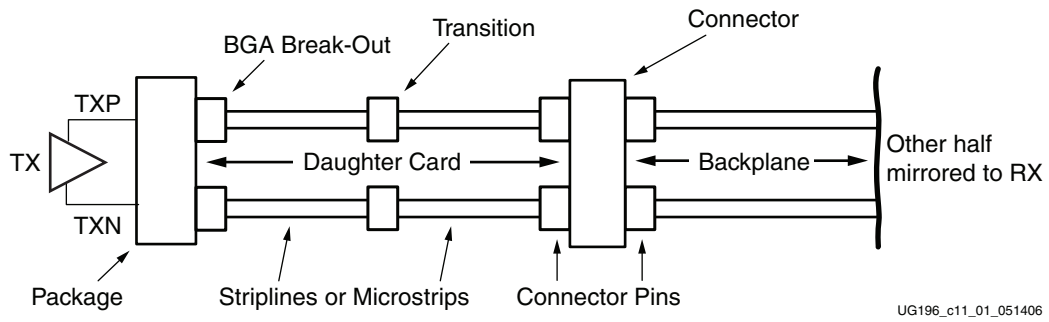


Figure 11-1: **Two Connected Transceivers Forming a Link**

In Figure 11-1, the channel consists of the FPGA package, transmission lines, connectors, and transitions.

Transitions are defined as any section along a multi-gigabit channel where the signal must go from a transmission line to a three-dimensional structure or vice-versa. Vias, connectors, and coupling capacitors are examples of these structures.

While a more comprehensive list of transitions is discussed in Chapter 13, Design of Transitions, some common transitions are:

- Ball grid array (BGA) to PCB microstrip
- Microstrip to stripline vias
- DC blocking capacitors
- Connectors
- Bends and turns in a trace

For optimal performance, the following must be minimized in a given channel:

- Signal attenuation due to losses in the transmission medium
- Impedance transitions at each transition can lead to reflections, ringing, and other artifacts in the signal

At gigahertz signaling speeds, losses due to the transmission medium become significant due to greater signal attenuation with increasing frequency. The attenuation of the high-frequency components slows down the edge and reduces voltage swing, resulting in eye

Optiver Ex1009
Page 249 of 347

closure. See Chapter 12, PCB Materials and Traces, for guidelines on stackup and characteristic trace impedance design.

While there are several transitions in each channel, most or all transitions can be designed for the least negative impact on performance. Because standard non-optimized PCB structures tend to be capacitive at gigahertz frequencies, a convenient figure of merit for transitions is excess capacitance. An ideal transition does not have excess capacitance or inductance.

For each typical transition, techniques to limit excess capacitance and excess inductance are provided to build a robust channel on the first pass. These design rules, techniques, and examples are presented in Chapter 13, Design of Transitions. The goal is to ensure tight control of any impedance variation along the entire channel.

In general, minimizing the number of components and layer changes in the high-speed serial PCB traces brings about the most benefit. Careful design of the traces, vias, and even connector pads is required for gigahertz speeds.

Powering Transceivers

Supplying noise-free power to the transceivers is a critical factor in achieving low link error rates and reliable system-level operation. This section discusses general principles that should be applied to transceiver power supply designs.

Linear regulators are required for directly sourcing the power supplies. Although switching regulators are an appealing option in applications requiring high-power efficiency, they are unsuitable for directly sourcing transceivers because they can introduce switching noise, even if care is taken to eliminate the noise.

Power Distribution Architecture

In most system-level designs, multiple voltage levels are required for powering devices on the board. The supplies for devices that use leading-edge process technologies are typically low voltages in the region of 1V. At these low voltages, it is important that noise levels on these supplies be kept at a minimum.

For this reason, Xilinx recommends the use of point-of-load (POL) power distribution techniques. The POL approach places the power supplies right at the device being powered, hence the name. Background information on this approach can found in http://www.xilinx.com/publications/xcellonline/xcell_57/xc_pdf/p105-107_57-bellinix.pdf. This approach can be extended to the use of separate linear regulators for groups of transceivers. This has several advantages including:

- Increased system reliability by eliminating a single point of failure.
- The ability to independently adjust supply voltages for transceiver groups to support the requirements of different link interfaces.
- Reduction of power requirements from each regulator, which reduces the physical size, simplifies board layout, and eliminates board hot spots.

Figure 11-2 shows how POL power distribution can be applied to powering transceivers.

Apotex Ex1009
Page 250 of 347

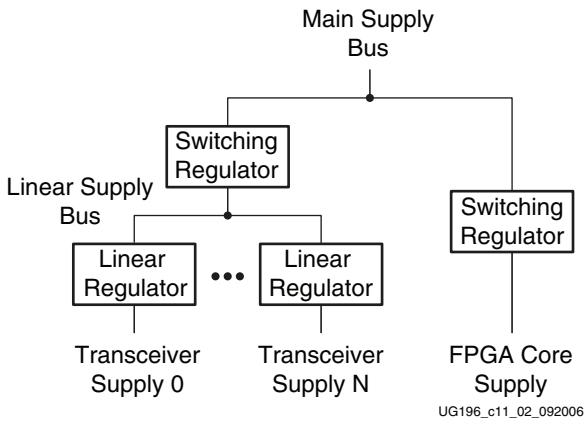


Figure 11-2: **POL Power Distribution Architecture**

In most cases the linear regulator is driven by a switching power supply. Care must be taken to ensure that ripple and other switching noise artifacts are filtered out by the regulator and/or filtering circuitry. Extra care must be taken when using low drop out (LDO) linear regulators because limited voltage margin at the input reduces the ability of the regulator to filter out noise components.

Regulator Selection

The designer must follow the voltage regulator selection guidelines for the specific receiver. A majority of transceiver performance issues are traced to power-supply noise issues.

Filtering

The power distribution system guidelines for the specific receiver must be strictly followed to achieve the specified performance.

Reference Clock

Clock Sources

A high-quality crystal oscillator is essential for good performance. The oscillator manufacturer's power supply design guide must be followed.

When choosing alternate clock sources, the alternate oscillators must meet or exceed the specifications as required by the transceiver data sheet.

Depending on the application and its performance goals, it is possible to stray from the clock source specifications. In that case the specified performance of the transceiver is not guaranteed.

Clock Traces

Because performance of the transceiver is directly related to the quality of its reference clock, every effort must be made to ensure the signal integrity of the clock traces from oscillator to FPGA. Apply the same techniques for 10 Gb/s trace design in Chapter 13, Design of Transitions, to these clock traces.

If more than one clock source is driving a single reference clock differential pair, a high-speed switch should be used. When multiple clock sources are bused together on a single connection, the signal integrity of the clock is not optimal due to the presence of stubs. Using a high-speed switch makes every clock path point-to-point with one driver and one receiver.

One example is when an unpopulated oscillator shares the same trace to the clock input pin as another clock source. The trace segment from the pad to the junction is a stub. Any signal travelling down the segment is reflected due to the different impedance presented by the open end at the pad.

For applications where a single reference clock source must drive multiple inputs, high-speed clock buffers should be used for clock distribution to eliminate stubs and reduce reflections on the clock lines.

Coupling

DC Coupling

DC coupling can be used when the common mode ranges of the interconnecting devices are the same. Any discrepancy in the common mode not only takes margin away from the differential voltage swing but can also damage the device.

AC Coupling

AC coupling isolates the common modes of the two devices and is the preferred configuration in hot-plug applications. The capacitor prevents any DC current from flowing between connected devices.

Some transceivers have built-in DC blocking capacitors with programmable bypass. In most cases, an external DC blocking capacitor is needed to provide adequate system-level performance.

External Capacitor Value Selection

If an external DC blocking capacitor is needed, it is important to select an appropriate value. The selection of a capacitor value is a trade-off between the following contradictory criteria:

- Encoding schemes with longer run lengths require larger capacitance values to reduce pattern dependent jitter (PDJ).
- Higher data rates require smaller capacitor values to reduce edge rate degradation.

PDJ is not an issue in protocols using line codings that preserve DC balance. DC balance is the property where the average number of `1`s and `0`s transmitted are equal. 8B/10B is an example of a line-coding scheme that provides DC balance. When 8B/10B encoding is used, 0.01 µF capacitors in a 0402 (EIA) package are suitable for external AC coupling at 3.125 Gb/s.

Line coding schemes that do not guarantee DC balance require more careful analysis. An example includes SONET, which uses scrambling to ensure adequate symbol transitions but does not provide DC balance. The remainder of this section provides the theory needed to select a blocking capacitor value appropriate for the application.

Apotex Ex1009
Page 252 of 347

Several protocols, including the PCI Express and SATA protocols, specify ranges for blocking capacitors in applications. This is done not only to simplify compliance but also to ensure that the link presence detection features included in these specifications work correctly.

Table 11-1: **PCI Express and SATA Blocking Capacitor Values**

Specification	Required Range
PCI Express Base Specification, Revision 1.1	75 to 200 nF
Serial ATA Specification, Revision 2.5	0 to 12 nF

The blocking capacitor when combined with the termination resistance acts as a high-pass filter. Figure 11-3 shows a simplified circuit model for the link. The internal blocking capacitors are not shown in this model because they do not play a significant role in blocking DC currents from the external link as described in RX Termination and Equalization, page 139.

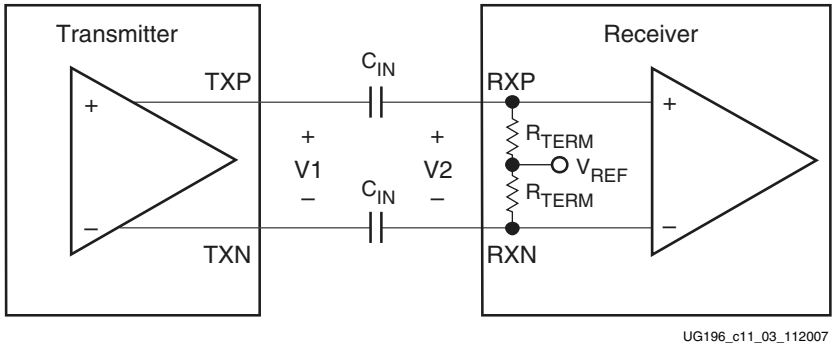


Figure 11-3: **Simplified Link Circuit Model**

Problems occur when the line is held in the on state for an extended period of time. When this happens, charge accumulates on the blocking capacitors and a DC offset is added or subtracted from V2. This offset results in what is known as baseline wander (see Figure 11-4). In Figure 11-4, VTH is the threshold voltage.

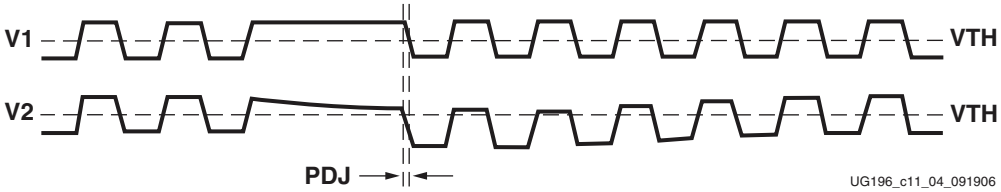


Figure 11-4: **Baseline Wander and PDJ**

The effect of baseline wander is to shift the signal with respect to the threshold points in the receiver. This in turn skews the time at which transitions within the signal are recognized. PDJ is the result of this skew. Figure 11-5 shows an overlay of V1 and V2 in the region of Figure 11-4 where the jitter is greatest and shows several key parameters.

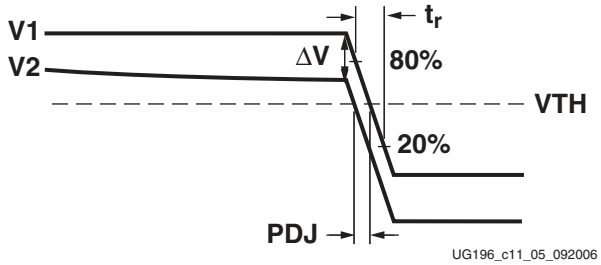


Figure 11-5: **PDJ Detail**

To calculate the blocking capacitor value, several factors must be known:

- t_r: The rise time of the signal
- T: The bit period
- N_{CID}: The maximum number of consecutive identical digits (CIDs)
- PDJ: The amount of pattern dependent jitter that can be tolerated by the system

From Figure 11-5 it can be seen that PDJ can be estimated by:

$$PDJ = \frac{\Delta V}{slope} \qquad \textit{Equation 11-1}$$

The voltage drop can be calculated using Equation 11-2:

$$\Delta V = 0.5 V_{PP}(1 - e^{-t/\tau}) \qquad \textit{Equation 11-2}$$

where:

- τ is the RC time constant (C is the AC coupling capacitor, R = 2 x R_{TERM}).
- t is the total discharge time, which is equal to N_{CID}T.

The slope is defined by Equation 11-3:

$$slope = V_{PP} \times \frac{0.6}{t_r} \qquad \textit{Equation 11-3}$$

Substituting Equation 11-2 and Equation 11-3 into Equation 11-1 and solving for C gives:

$$C = \frac{-T \times N_{CID}}{2 \times R_{TERM} \times \ln\left(1 - \frac{1.2PDJ}{t_r}\right)} \qquad \textit{Equation 11-4}$$

To demonstrate the use of Equation 11-4, calculate the blocking capacitor value needed for a serial link running at 3.125 Gb/s using 8B/10B line coding. This example uses the following assumptions:

• Bit Period (T)	3.200 x 10^{-10} (3.125 Gb/s)
• Signal Rise Time (t_r)	6.400 x 10^{-11} (0.2 UI)
• Pattern Dependent Jitter (PDJ)	3.200 x 10^{-12} (0.01 UI)
• Consecutive Identical Digits (N_{CID})	5 (guaranteed by 8B/10B)
• Termination Resistance (R_{TERM})	50Ω

Oplver Ex1009
Page 254 of 347

Plugging these values into Equation 11-4 gives:

$$C = \frac{-(3.20 \times 10^{-10}) \times 5}{2 \times 50 \times \ln\left(1 - \frac{1.2(3.20 \times 10^{-12})}{6.40 \times 10^{-11}}\right)} = 0.26\ nF \qquad \textit{Equation 11-5}$$

Equation 11-4 is only valid for cases where the problem data pattern consists of a single sequence of consecutive identical digits. More complex pathological cases can occur in protocols using block coding schemes and scrambling.

One example of such a case occurs in the serial digital interface (SDI) used to transmit digital video. Figure 11-6 illustrates this pattern, which is called an equalization test pattern.

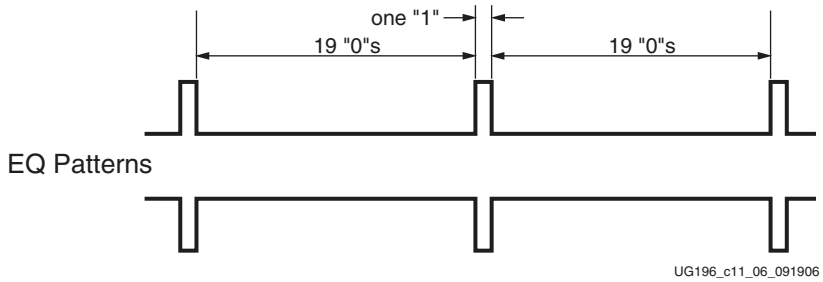


Figure 11-6: **SDI EQ Pathological Waveform**

The waveform is 20 bits long and consists of a single “1” bit followed by 19 “0” bits. The complementary waveform, a single “0” bit followed by 19 “1” bits is also equally likely. The EQ waveform can repeat across the entire length of the active portion of a video line. For standard definition video (SD-SDI), this waveform can consist of up to 720 consecutive repetitions of the 20-bit pattern. For high definition video (HD-SDI), this pattern can repeat up to 1920 consecutive times.

For this case, the blocking capacitor value cannot be calculated using Equation 11-4 because an N_{CID} value of 19 does not reflect the total time that a DC imbalance is being applied on the line. To properly analyze charge accumulation on blocking capacitors for this case, more extensive analysis beyond the scope of this document is required.

SelectIO to Serial Transceiver Crosstalk Guidelines

The breakout of SelectIO signals adjacent to transceiver analog supply pins and REFCLK pins is also important. As noted in SelectIO to GTP Crosstalk Guidelines, page 242, these SelectIO requirements, if not taken into account, can have an effect on transceiver performance. This impact occurs when SelectIO solder balls are adjacent to transceiver analog supply or REFCLK solder balls and their corresponding PCB vias are adjacent as well, creating both package and board coupling mechanisms. The solder balls, which are part of the package, offer some coupling, and the adjacent PCB vias offer two to four times more coupling. Simulation suggests that the amount of coupling due to adjacent PCB vias is affected by which layer the SelectIO escape is located and on how the analog supplies are delivered to the transceivers. Simulation predicts that coupling can be reduced by using the upper PCB routing layers to route SelectIO signals and/or by using a higher layer to distribute transceiver analog power supplies. When a design dictates that SelectIO signals with BGA adjacency to transceiver analog supply pins are to be used for high-drive/high-speed applications, the following guidelines apply:

- Apply power to the transceiver analog supplies with a plane or wide buses a few layers below the top of the board. Using a blind via to the transceiver analog supplies is better than using a through via. Shield the supply plane with GND planes above and below.
- If a through via to supply the transceiver analog supply pins must be used, use an upper layer to supply analog power to these vias. Route SelectIO nets in the uppermost layer available after transceiver signal and transceiver analog supply routing is implemented.
- If supplying transceiver power from the bottom of the board, route these SelectIO nets in the highest available routing layer.
- Do not use SelectIO blocks adjacent to REFCLK pins because the REFCLK pins are a reference clock source to the transceivers either in the same tile or in other tiles.

Figure 11-7 depicts the coupling regions for BGA adjacent SelectIO signals. The primary coupling mechanism is mutual inductive coupling, which occurs in the area between the active signal path and the power via. The secondary coupling mechanism, also shown in Figure 11-7, is capacitive. The primary coupling mechanism is much larger, and the recommendations are designed to minimize this effect.

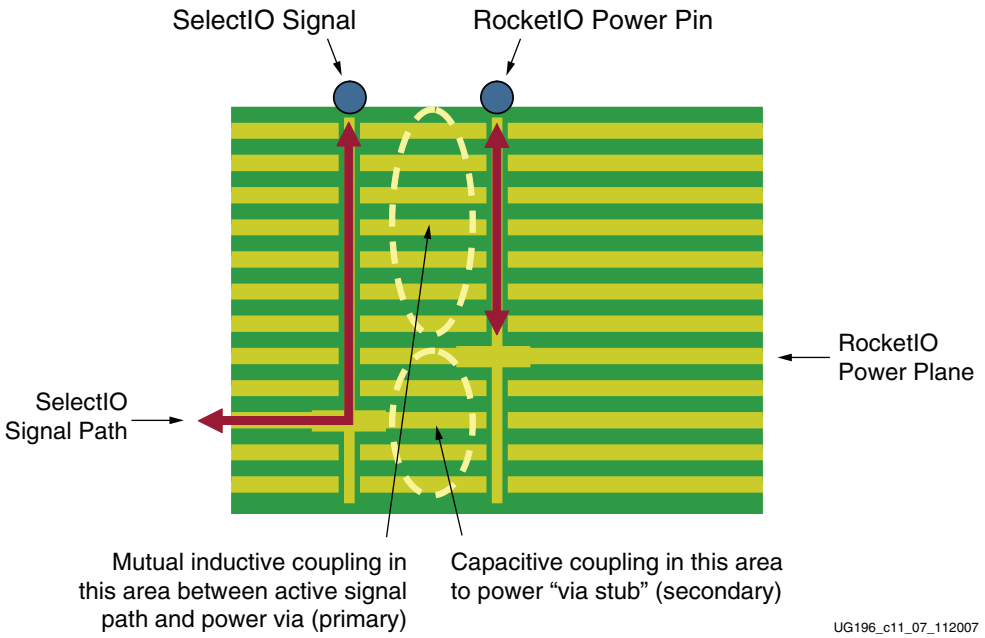


Figure 11-7: **Via Structures for BGA Adjacent SelectIO Signals**

Oplover Ex1009
Page 256 of 347

Chapter 12

PCB Materials and Traces

The choice of PCB materials and cable type can have a large impact on system performance. Although any transmission medium is lossy at gigahertz frequencies, this chapter provides some guidelines on managing signal attenuation so as to obtain optimal performance for a given application.

How Fast is Fast?

Signal edges contain frequency components called harmonics. Each harmonic is a multiple of the signal frequency and has significant amplitude up to a frequency determined by Equation 12-1:

$$f \approx 0.35 / T \qquad \text{Equation 12-1}$$

Where:

f = Frequency in GHz
T = The smaller of signal rise (T_r) or fall (T_f) time in ns

Because dielectric losses in a PCB are frequency dependent, a bandwidth of concern must be determined to find the total loss the PCB. Frequencies must start at the operation frequency and extend to the frequency in Equation 12-1. For example, a 10 Gb/s signal with a 10 ps rise time has a bandwidth from 10 GHz to 35 GHz.

Dielectric Losses

The amount of signal energy lost into the dielectric is a function of the materials characteristics. Some parameters used to describe the material include relative permittivity ε_r (also known as the dielectric constant) and loss tangent. Skin effect is also a contributor to energy loss at line speeds in the gigahertz range.

Relative Permittivity

Relative permittivity is a measure of the effect of the dielectric on the capacitance of a conductor. The higher the relative permittivity, the slower a signal travels on a trace and the lower the impedance of a given trace geometry. A lower ε_r is almost always preferred.

Although the relative permittivity varies with frequency in all materials, FR4 exhibits wide variations in ε_r with frequency. Because ε_r affects impedance directly, FR4 traces can have a spread of impedance values with increasing frequency. While this spread can be less significant at 3.125 Gb/s, it can be a concern at 10 Gb/s operation.

Optiver Ex1009
Page 257 of 347

Loss Tangent

Loss tangent is a measure of how much electromagnetic energy is lost to the dielectric as it propagates down a transmission line. A lower loss tangent allows more energy to reach its destination with less signal attenuation.

As frequency increases, the magnitude of energy loss increases as well, causing the highest frequency harmonics in the signal edge to suffer the most attenuation. This appears as a degradation in the rise and fall times.

Skin Effect and Resistive Losses

The skin effect is the tendency for current to flow preferentially near the outer surface of a conductor. This is mainly due to the larger magnetic fields in higher frequency signals pushing current flow in the perpendicular direction towards the perimeter of the conductor.

As current density near the surface increases, the effective cross-sectional area through which current flows decreases. Resistance increases because the effective cross-sectional area of the conductor is now smaller. Because this skin effect is more pronounced as frequency increases, resistive losses increase with signaling rates.

Resistive losses have a similar effect on the signal as loss tangent. Rise and fall times increase due to the decreased amplitude of the higher harmonics, with the highest frequency harmonics being most affected. In the case of 10 Gb/s signals, even the fundamental frequency can be attenuated to some degree when using FR4.

For example, an 8 mil wide trace at 1 MHz has a resistance on the order of 0.06Ω/inch, while the same trace at 10 Gb/s has a resistance of just over 1Ω/inch. Given a 10 inch trace and 1.6V voltage swing, a voltage drop of 160 mV occurs from resistive losses of the fundamental frequency, not including the losses in the harmonics and dielectric loss.

Choosing the Substrate Material

The goal in material selection is to optimize both performance and cost for a particular application.

FR4, the most common substrate material, provides good performance with careful system design. For long trace lengths or high signaling rates, a more expensive substrate material with lower dielectric loss must be used.

Substrates, such as Nelco, have lower dielectric loss and exhibit significantly less attenuation in the gigahertz range, thus increasing the maximum bandwidth of PCBs. At 3.125 Gb/s, the advantages of Nelco over FR4 are added voltage swing margin and longer trace lengths. At 10 Gb/s, Nelco is necessary unless high-speed traces are kept very short.

The choice of substrate material depends on the total length of the high-speed trace and also the signaling rate.

What-if analysis can be done in HSPICE simulation to evaluate various substrate materials. By varying the dielectric constant, loss tangent, and other parameters of the PCB substrate material. The impact on eye quality can be simulated to justify the use of higher cost materials. The impact of other parameters such as copper thickness can also be explored.

Oplover Ex1009
Page 258 of 347

Traces

Trace Geometry

For any trace, its characteristic impedance is dependent on its stackup geometry as well as the trace geometry. In the case of differential traces, the inductive and capacitive coupling between the tightly coupled pair also determines the characteristic impedance of the traces.

The impedance of a trace is determined by its inductive and capacitive coupling to nearby conductors. For example, these conductors can be planes, vias, pads, connectors, and other traces, including the other closely coupled trace in a differential pair. The substrate properties, conductor properties, flux linkage area, and distance to a nearby conductor determine the amount of coupling and hence, the contribution to the final impedance.

2D field solvers are necessary in resolving these complex interactions and contribute to the calculation of the final impedance of the trace. They are also a useful tool to verify existing trace geometries.

A common misconception is that two 50Ω single-ended traces can be routed side-by-side to give a pair with 100Ω differential impedance. While this approximation might be true if the traces are loosely coupled, routing differential traces in a loosely coupled fashion does not maximize the noise immunity of differential mode signaling.

Tightly coupled differential pairs are required for all high-speed GTP traces because they are more sensitive to noise than slower signals. As a general rule of thumb, tight coupling within a differential pair is achieved by spacing them no more than four trace widths apart.

Wider traces create a larger cross-sectional area for current to flow and reduce resistive losses. Use the widest traces that space constraints allow. Because trace width tolerances are expressed in absolute terms, a wider trace also minimizes the percentage variation of the manufactured trace, resulting in tighter impedance control along the length of the transmission line.

Striplines are preferred over microstrips because the reference planes on both sides of the trace provide radiation shielding. Microstrips are shielded on only one side (by the reference plane) because they run on the top-most or bottom-most layers, leaving the other side exposed to the environment.

For best results, the use of a 2D field solver is recommended for verification.

Trace Characteristic Impedance Design

Because the transceivers use differential signaling, the most useful trace configurations are differential edge-coupled center stripline and differential microstrip. While some backplanes use the differential broadside-coupled stripline configuration, it is not recommended for 10 Gb/s operation, because the P and N vias are asymmetrical and introduce common-mode non-idealities.

With few exceptions, 50Ωcharacteristic impedance (Z_0) is used for transmission lines in the channel. In general, when the width/spacing (W/S) ratio is greater than 0.4 (8 mil wide traces with 20 mil separation), coupling between the P and N signals affects the trace impedance. In this case, the differential traces must be designed to have an odd mode impedance (Z_{0O}) of 50Ω, resulting in a differential impedance (Z_{DIFF}) of 100Ω, because $Z_{DIFF} = 2 \times Z_{0O}$.

The same W/S ratio also must be less than 0.8, otherwise strong coupling between the traces requires narrower, lossier traces for a Z_{0O} of 50Ω To clarify, with Z_{0O} at 50Ω, an even mode impedance (Z_{0E}) of 60Ω or below is desired.

Figure 12-1 through Figure 12-4 show example cross sections of differential structures.

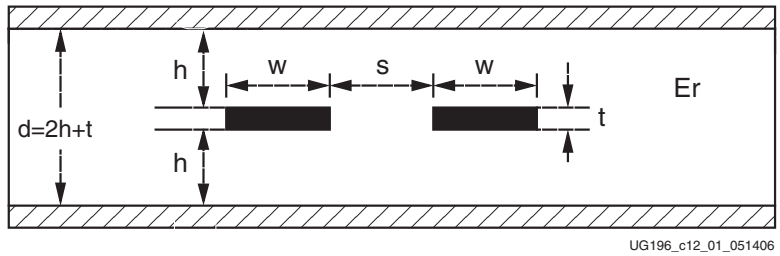


Figure 12-1: **Differential Edge-Coupled Centered Stripline**

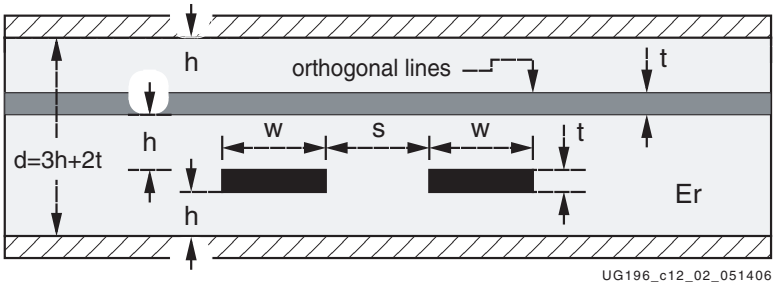


Figure 12-2: **Differential Edge-Coupled Offset Stripline**

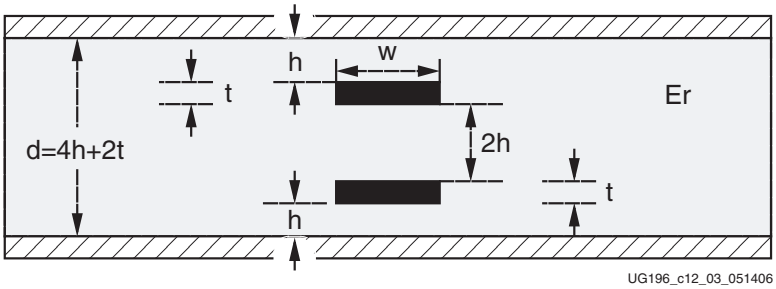


Figure 12-3: **Centered Broadside-Coupled Stripline**

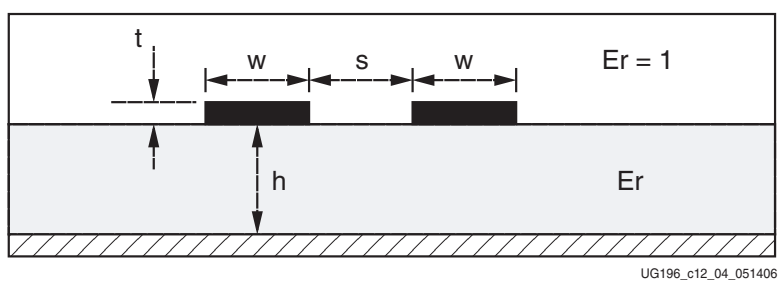


Figure 12-4: **Differential Microstrip**

A good PCB manufacturer understands controlled impedance and allows fine adjustments for line widths to produce a Z_{0O} of 50Ω. The PCB manufacturer also provides the parameters necessary for the specific PCB layout. Some parameters can be calculated or

Oplver Ex1009
Page 260 of 347

simulated from the guideline outlined in the example. Although ±10% tolerance on Z_{0O} is typical and can provide adequate performance, the additional cost of a tighter tolerance results in better channel performance.

Trace Routing

High-speed serial differential traces are routed with the highest priority to ensure that the optimal path is available to these critical traces. This reduces the need for bends and vias and minimizes the potential for impedance transitions. Traces must be kept straight, short, and with as few layer changes as possible. The impact of vias is discussed in Differential Vias, page 269.

Routing of high-speed traces must be avoided near other traces or other potential sources of noise. Traces on neighboring signal planes should run perpendicular to minimize crosstalk.

Striplines are to be used whenever possible, as are the uppermost and lowermost stripline layers to minimize via stubs. When the stackup is being planned, these layers must be placed as close to the top and bottom layers whenever possible.

Design constraints might require microstrips for the BGA exit path or from via to connector launch or SMT pads. In such cases, the microstrip trace must be kept as short as possible.

Right-angled bends must not be used. Mitered 45-degree bends are to be used instead. At a 90-degree bend, the effective width of the trace changes, causing an impedance discontinuity due to the capacitive coupling of the additional conductor area to the reference plane.

The two traces of a differential pair must be length-matched to eliminate skew. Skew creates mismatches in the common mode and reduces the differential voltage swing as a result.

Plane Splits

Ground planes should be used as reference planes for signals, as opposed to noisier power planes. Each reference plane should be contiguous for the length of the trace, because routing over plane splits creates an impedance discontinuity. In this case, the impedance of the trace changes because its coupling to the reference plane is changed abruptly at the plane split.

Return Currents

Routing over plane splits also creates issues with the return current. High-speed signals travel near the surface of the trace due to the skin effect mentioned in Dielectric Losses. Meanwhile, the return current also travels near the surface of the tightly coupled reference plane.

Because of the tight coupling, the return current has the tendency to travel close to the original signal-carrying trace. At the plane split, the return current can no longer follow the same path parallel to the trace, but must instead find an alternative route.

A plane split causes a suboptimal current return path and increases the current loop area, thereby increasing the inductance of the trace at the plane split, changing the impedance of the trace.

Optiver Ex1009
Page 261 of 347

Simulating Lossy Transmission Lines

Due to the different modeling implementations used by various circuit simulators (frequency-domain versus time-domain techniques), it is important to check that the models accurately reflect actual losses. One method is to compare the models against known published configurations.

Cable

Cables are controlled-impedance transmission lines due to the constant physical dimensions of conductor and dielectric along the length of the cable. The highest quality cable shows little variation in these dimensions and also has a wide bandwidth with low loss at high frequencies.

Connectors

The connectors attached to cables should exhibit low parasitic inductance and capacitance for high bandwidth operation.

Skew Between Conductors

When selecting a cable, look for a specification of the skew between the conductors in a cable. If the conductors are not length matched, the skew appears in the common mode and directly reduces the eye height.

Optiver Ex1009
Page 262 of 347

Chapter 13

Design of Transitions

Each transition in the channel must be designed to minimize any negative impact on the link performance. This section addresses the interface at either ends of a transmission line.

Transmission lines have defined and controlled characteristic impedance along their length by definition. However, the three-dimensional structures that they interface do not have easily defined or constant impedance along the signal path. Software tools such as 3D field solvers are necessary for computing the impedance that a 10 Gb/s signal sees as it passes through these structures, while 2D field solvers are sufficient for computing transmission line characteristic impedance.

PCB designers can use the analyses and examples in this section to greatly accelerate the design of such a channel. Cases not covered in this section might need further simulation and board iterations.

Excess Capacitance and Inductance

Most differential transitions are overly capacitive. The P and N paths couple to each other, increasing capacitance. Many transitions have a frequency response identical to that of a lumped capacitor over a wide frequency band.

By design, adding inductance cancels this excess capacitance in many cases except when impacted by density concerns and physical limitations. While techniques such as blind vias, solder balls on a larger pitch, and very small via pads reduce capacitance, they are not always feasible in a design.

Time domain reflectometry (TDR) techniques, either through simulation or measurement, allow the designer to identify excess capacitance or excess inductance in a transition.

Time Domain Reflectometry

To make TDR measurements, a step input is applied to the interconnect. The location and magnitude of the excess capacitance or inductance that the voltage step experiences as it traverses the interconnect can be determined through observing the reflected signal.

A shunt capacitance (see Figure 13-1) causes a momentary dip in the impedance, while a series inductance (see Figure 13-2) causes an impedance discontinuity in the opposite direction. Td is assumed to be the propagation delay through the first transmission line segment on the left. The reflected wave due to the impedance discontinuity takes 2 * Td to return to the TDR port. If the signal propagation speed through the transmission line is known, the location of the excess capacitance or inductance along the channel can be calculated.

Optiver Ex1009
Page 263 of 347

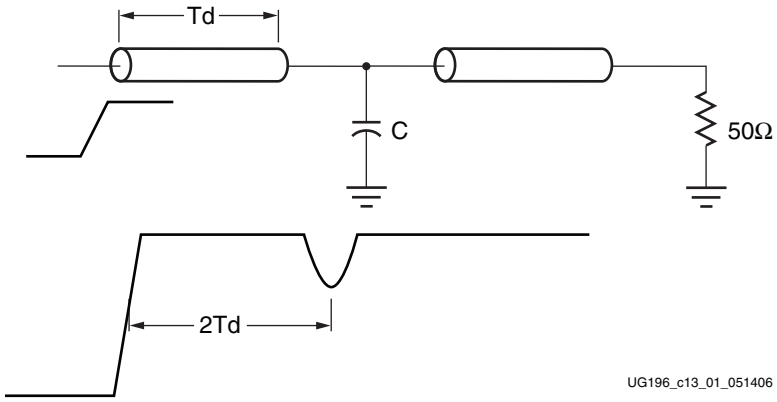


Figure 13-1: **TDR Signature of Shunt Capacitance**

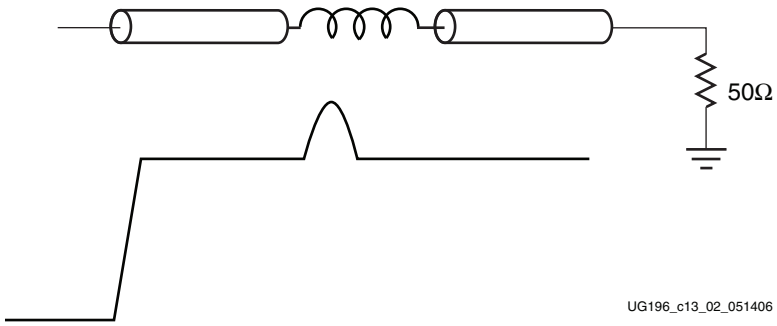


Figure 13-2: **TDR Signature of Series Inductance**

The magnitude of this excess capacitance (C) or inductance (L) can also be extracted from the TDR waveform by integrating the normalized area of the transition's TDR response. The respective equations for capacitance and inductance are:

$$C = -\frac{2}{Z_0}\int_{t1}^{t2}\frac{V_{tdr}(t) - V_{step}}{V_{step}}\,dt \qquad \textit{Equation 13-1}$$

$$L = 2Z_0\int_{t1}^{t2}\frac{V_{tdr}(t) - V_{step}}{V_{step}}\,dt \qquad \textit{Equation 13-2}$$

Figure 13-3 shows the integration of the normalized TDR area.

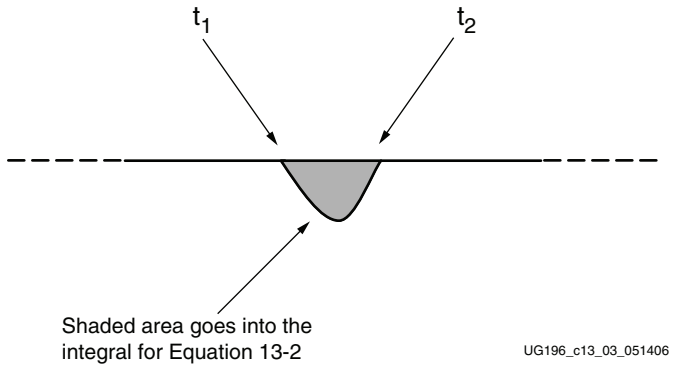


Figure 13-3: **Integration of Normalized TDR Area**

Oppver Ex1009
Page 264 of 347

The results using these equations are not sensitive to rise time variation and are valid for simulated TDR measurements provided that the leading and trailing transmission lines are very close to 50Ω. However, for actual measurements, accuracy is very dependent on Z_0.

BGA Package

The transceiver signal paths within the BGA package are optimized using a 3D full-wave solver. Package traces are designed to be 50Ω high-speed transmission lines, while solder ball and bump regions are tuned to 50Ω.

Flip-chip package transitions are effectively invisible to 10 Gb/s signals. The longest package paths have some insertion loss, less than 1 dB worst-case. To allow full simulation of package effects, the Virtex-5 FPGA RocketIO Transceiver Signal Integrity Simulation kit provides extracted S-parameter models of the package. Refer to the *Virtex-5 FPGA RocketIO Transceiver Signal Integrity Simulation Kit User Guide* [Ref 14] for more information.

SMT Pads

For applications that require AC coupling between transmitter and receiver, SMT pads are introduced in the channel to allow coupling capacitors to be mounted. Standard SMT pads have excess capacitance due to plate capacitance to a nearby reference plane. In the following example, a 5 mil trace with a Z_0 of 50Ω transitions to an 0402 SMT pad that is 28 mils wide, all over 3 mils of FR4.

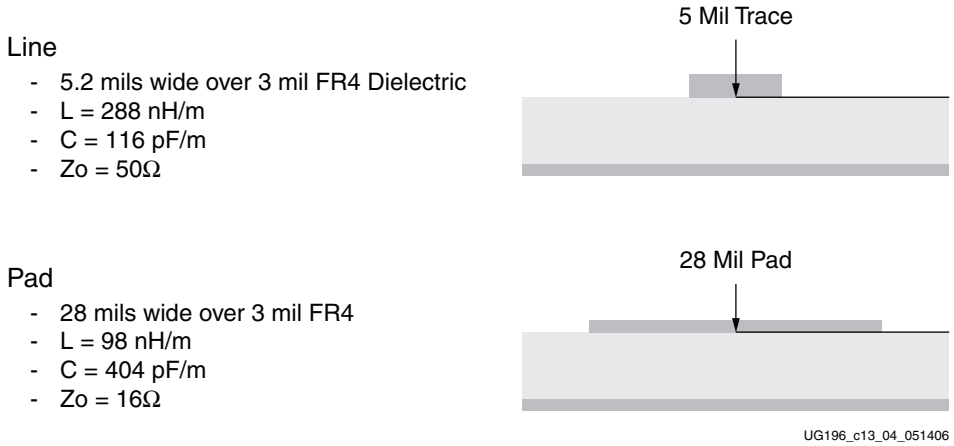


Figure 13-4: **2D Field-Solver Analysis of 5 Mil Trace and 28 Mil Pad**

Using a 2D field solver on these dimensions yields a Z_0 of 50Ω for the 5 mil trace. The Z_0 for the 0402 pad is 16Ω because the pad has too much capacitance and too little inductance, resulting in an impedance of less than 50Ω. Performance of this transition can be optimized in one of two ways.

The first method makes the trace the same width as the pad and moves the ground plane deeper into the stackup to maintain the Z_0 of the transition at 50Ω. This method does not require any special analysis, but there might be some error due to the fringing capacitance of the SMT capacitor body. Trace density is limited because traces are now 28 mils wide.

The second method clears the ground plane underneath the pad, which removes much of the excess capacitance caused by the plate capacitance between the pad and the ground plane. This technique allows for greater trace density than the first method, but requires 3D

Optiver Ex1009
Page 265 of 347

field-solver analysis or measurement along with several board iterations to get the desired performance.

Figure 13-5: **Transition Optimization**

The 2D field-solver example shows that close to 50Ω can be achieved if the ground plane under the pad footprint is cleared out. A 3D field solver is then used to verify this result to a greater degree of accuracy.

Figure 13-6 shows the ground plane cleared away exactly as it was for the 2D simulation. Using frequency domain analysis within HFSS, there is a 20 dB (10x) improvement in return loss using this technique.

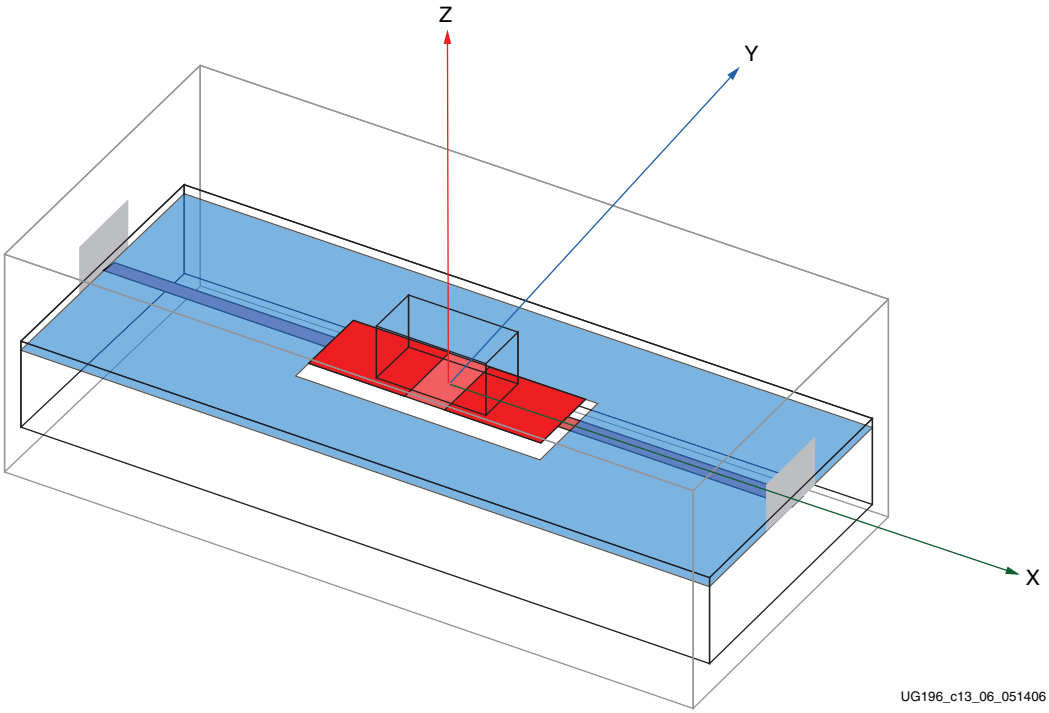


Figure 13-6: **Ansoft HFSS Model of Pad Clear-Out**

The approximately -40 dB/decade slope in Figure 13-8 shows good fit to the frequency response of a lumped capacitor.

Capriver Ex1009
Page 266 of 347

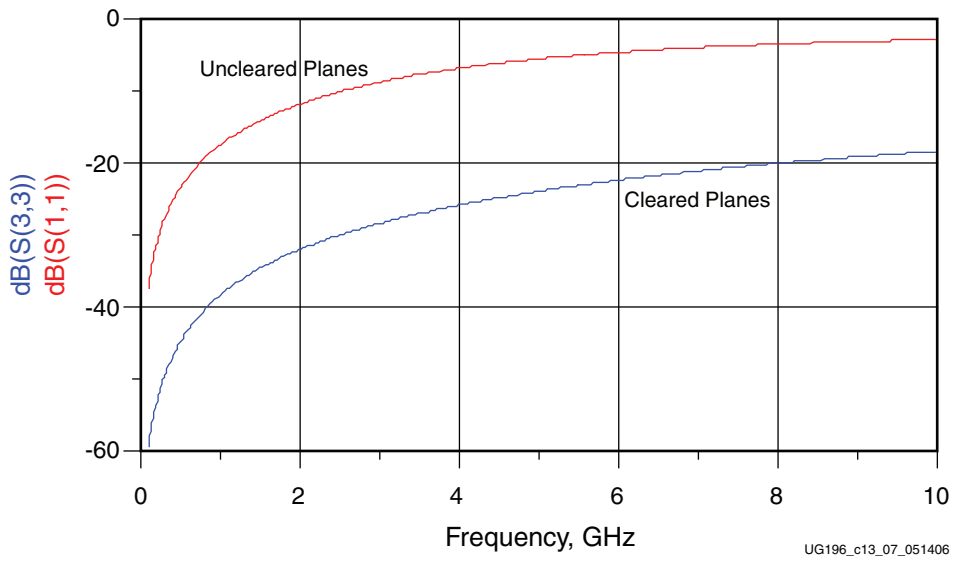


Figure 13-7: **Return Loss Comparison Between 0402 Pad Structures**

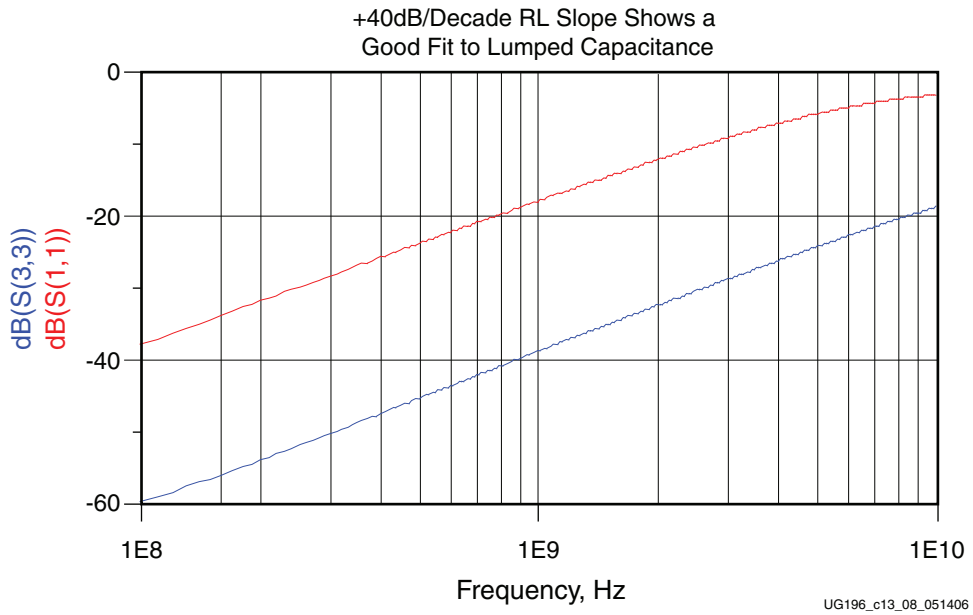


Figure 13-8: **Return Loss Comparison Between 0402 Pad Structures on Log (Frequency) Scale**

Next, using simulated measurements on the same transition modeled in HFSS, the time-domain performance of this transition can be measured by doing a TDR on the S-parameter results from the earlier frequency domain analysis.

In Figure 13-9 and Figure 13-10, the red curve with the large capacitive dip corresponds to the SMT pad without the ground plane cleared from underneath. The blue curve shows that clearing out the ground plane removes much of the excess capacitance. This improvement can be quantified using Equation 13-1 and Equation 13-2.

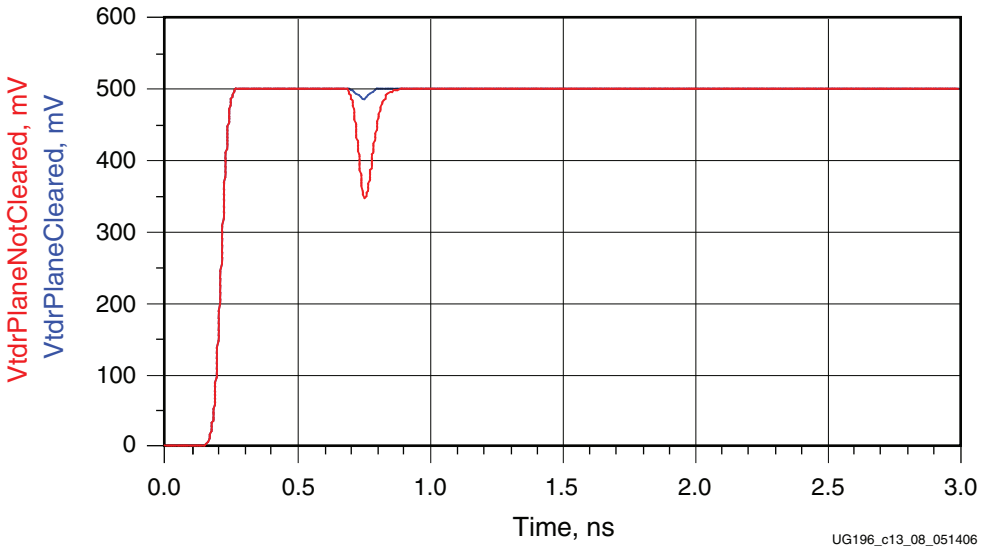


Figure 13-9: **TDR Results Comparing 0402 Pad Structures**

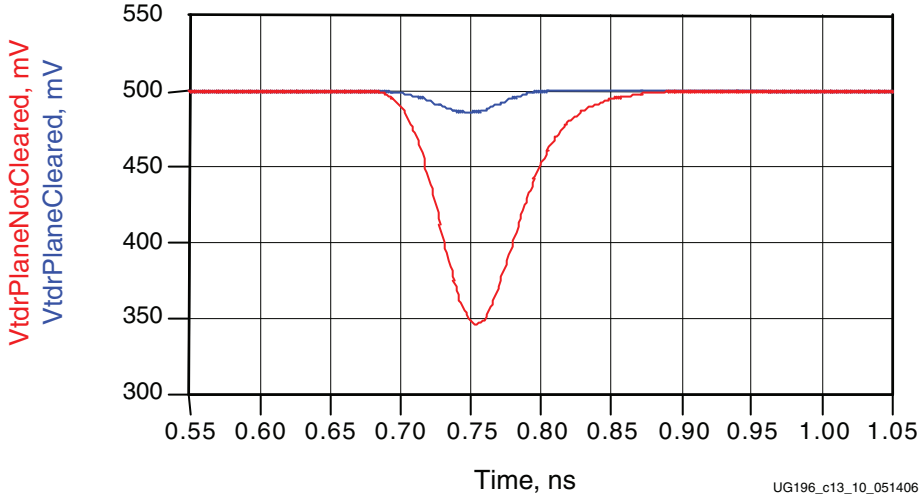


Figure 13-10: **TDR Results Comparing 0402 Pad Structures**

As shown from Figure 13-11 and Figure 13-12, clearing the ground plane under SMT pads yields a significant improvement in the performance of an SMT pad transition. Excess capacitance is reduced by 15x, and return loss is improved by 20 dB.

Apotex Ex1009
Page 268 of 347

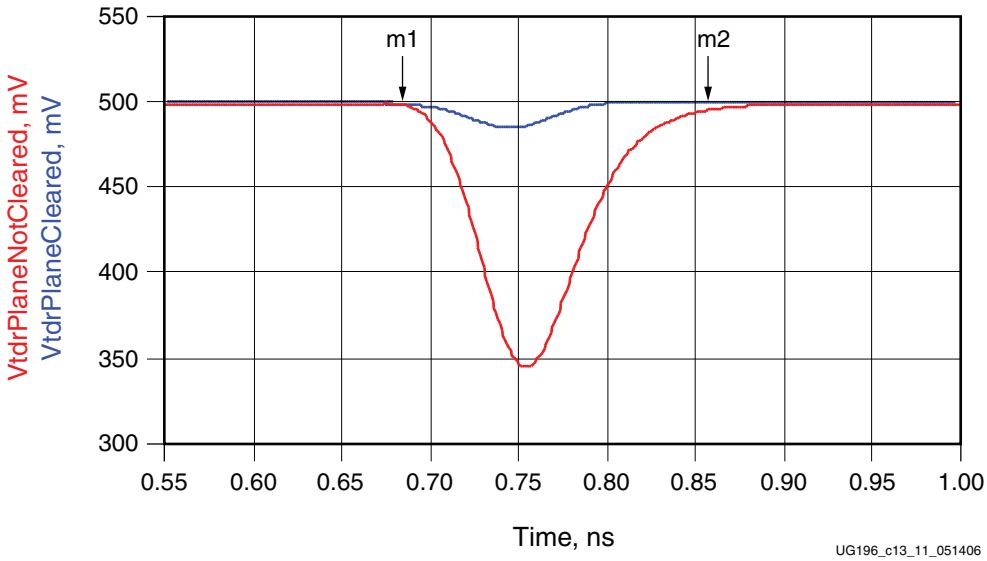


Figure 13-11: **840 fF Excess Capacitance with Ground Plane Intact**

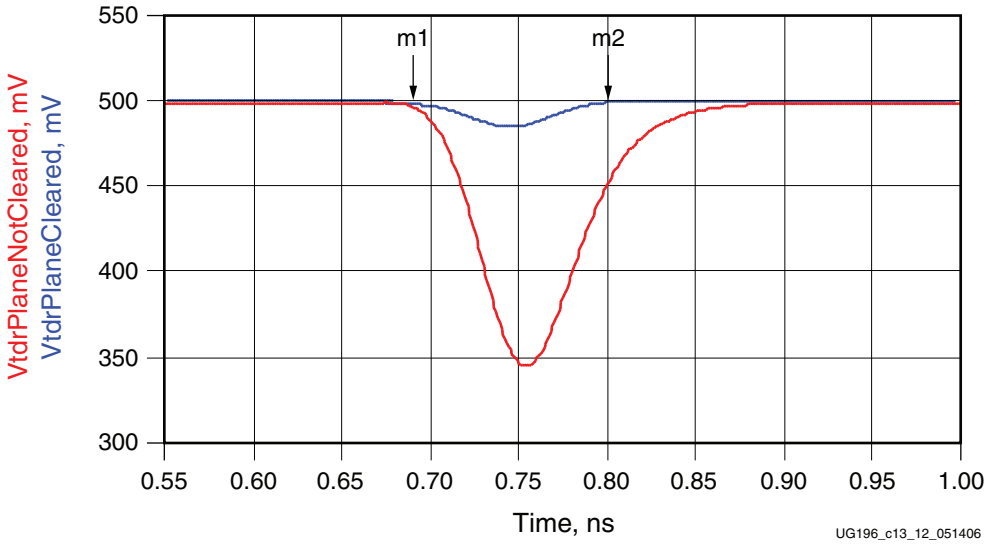


Figure 13-12: **57 fF Excess Capacitance with Ground Plane Intact**

Differential Vias

The most common transition is the differential via where the signal pair must transition from an upper stripline layer or top microstrip to a lower stripline layer or bottom microstrip.

Figure 13-13 shows a Ground-Signal-Signal-Ground (GSSG) type differential via. Ground vias are connected to each ground plane in the stackup, while signal layers only contain pads for the entry and exit layers.

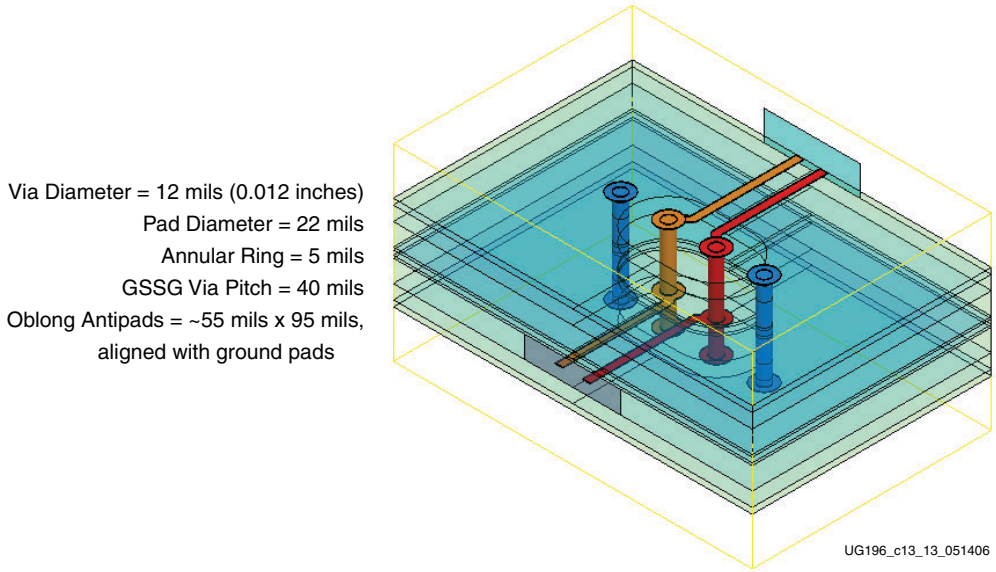


Figure 13-13: **Differential Via Design Example**

A key advantage of a GSSG via is that it allows for the signal's return current to flow in the ground via near the corresponding signal via, reducing excess inductance. The signal path is also symmetrical between the P and N halves of the differential signal, which is critical in controlling common-mode artifacts due to P/N imbalance.

The larger oblong antipads reduce excess fringing capacitance between the via body and the surrounding planes edges. Unused pads are also removed.

A good starting point is to use the dimensions shown in Figure 13-13 as an example differential via design for an 80 mil board. To accommodate density constraints or the lack thereof, the dimensions can be scaled accordingly to preserve the ratios of each dimension relative to the others. Such scaling preserves the impedance performance of the differential via while allowing variation in overall size to better suit specific applications. These final dimensions are limited by manufacturability and density constraints.

While the via length can be varied by a small amount to suit boards that are thicker or thinner than the 80 mil example, changing the ratio of the via length relative to other dimensions affects the via's impedance. For this and other configurations of differential vias, it is best to simulate a model using 3D field-solver tools to ensure that performance targets are met.

Apotex Ex1009
Page 270 of 347

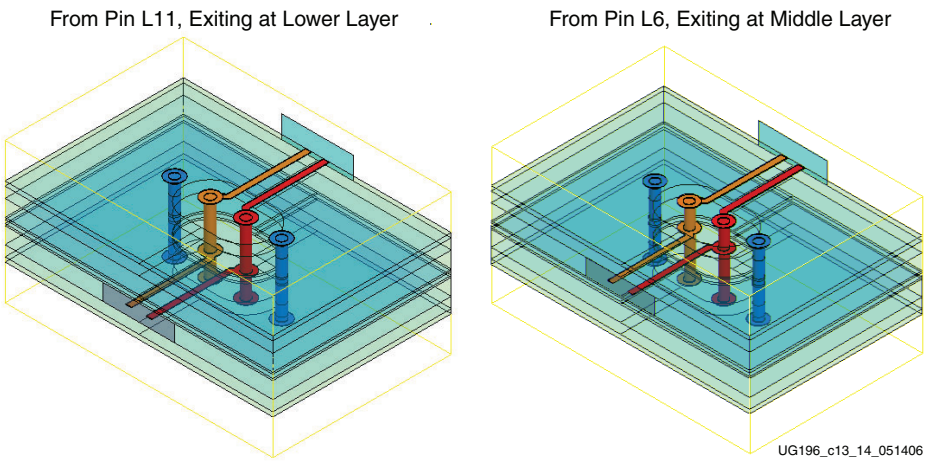


Figure 13-14: **Differential GSSG Via in 16-Layer PCB from Pins L11 and L6**

As a general rule, the P and N paths need to be kept at equal lengths through a transition. Where possible, via stub length should be kept to a minimum by traversing the signal through the entire length of the vias. The analysis shown in Figure 13-15 compares the S-parameter return loss for common-mode (SCC11) and differential (SDD11) responses.

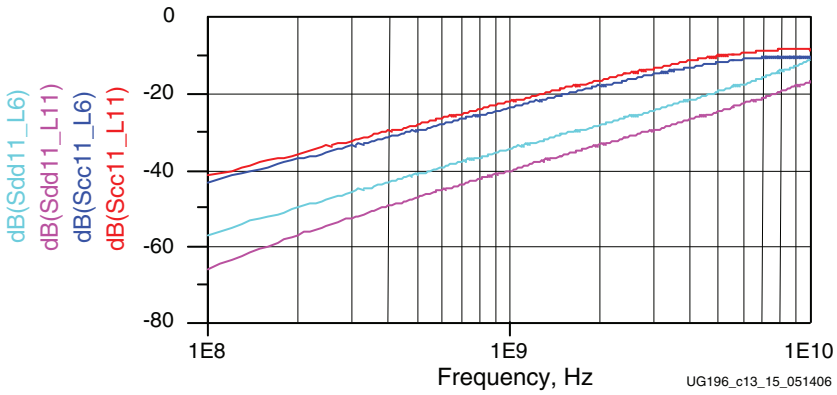


Figure 13-15: **Simulated Return Loss Comparing Differential and Common-Mode Losses for L11 and L6 GSSG Vias**

From the graph in Figure 13-15, the common-mode response is 20 dB worse in terms of return loss. The much worse common-mode response relative to the differential response is the reason why it is a good idea to reduce P/N skew as much as possible before entering a transition. The 60/40 rule of thumb is 40 dB of return loss at 1 GHz, which implies 60 fF of excess capacitance. Because excess capacitance is a single pole response, simple extrapolation rules can be used. For example, a shift to 34 dB return loss doubles the excess capacitance. Due to the excellent performance characteristics of GSSG vias, even long via stubs only double the differential via's capacitance at the most.

Chapter 14, Guidelines and Examples, provides additional examples of differential vias.

P/N Crossover Vias

Some transceivers offer the ability to independently switch the polarity of the transmit and receive signal pairs. This functionality eliminates the need to cross over the P/N signals at the board level, which in turn significantly enhances signal integrity. If possible, P/N crossover vias are to be avoided and the polarity switch of the transceiver should be used.

SMA Connectors

Well-designed SMA connectors can reduce debugging time and allow a high-performance channel to be designed correctly on the first pass. SMA connectors that perform well at 10 Gb/s need to be simulated, designed, and manufactured to meet this performance target. Vendors can also offer design services that ensure that the connector works well on a specific board. Assembly guidelines are crucial in ensuring that the process of mating the connector to the board is well-controlled to give the specified performance.

Xilinx uses Rosenberger SMA connectors almost exclusively because of their excellent performance and because of the points listed in the previous paragraph.

Backplane Connectors

There are numerous signal integrity issues associated with backplane connectors including:

- P/N signal skew
- Crosstalk
- Stubs due to connector pins

Chapter 14, Guidelines and Examples, provides a design example based on the popular HM-Zd connector.

Some connector manufacturers offer not only S parameters, models, and layout guidelines for their connectors but also design support, seminars, and tutorials.

Microstrip/Stripline Bends

A bend in a PCB trace is a transition. When routing differential traces through a 90° corner, the outer trace is longer than the inner trace, which introduces P/N imbalance. Even within a single trace, signal current has the tendency to hug the inside track of a corner, further reducing the actual delay through a bend.

To minimize skew between the P and N paths, 90° turns in microstrips or striplines are routed as two 45° bends to give mitered corners. The addition of a jog-out also allows the trace lengths to be matched. Figure 13-16 shows example bends in traces.

Cypher Ex1009
Page 272 of 347

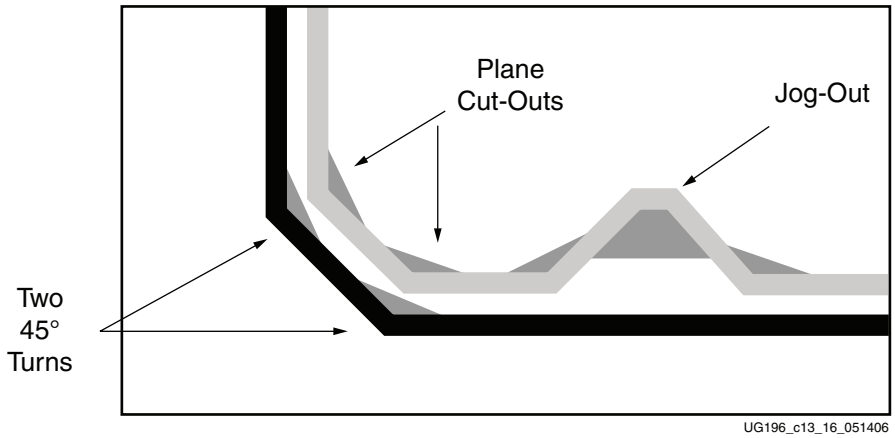


Figure 13-16: **Example Design for 90 Degree Bends in Traces**

Turns add capacitance because the trace at a 90° corner is 41% wider. That difference is reduced to 8% with a 45° turn. The addition of plane cutouts to a depth of 30 mils act to reduce this amount of excess capacitance. The trace was not widened to maintain 50Ω with the plane cutouts in place.

When this mitered bend is simulated with the jog-out and plane cutouts, excess capacitance is reduced and P/N length and phase matching is improved. Without jog-outs, the P/N length mismatch is 16 mils. Given FR4 material, the 16 mil difference translates to a phase mismatch of 4.8° at 5 GHz, or 2.68 ps (0.0268 UI) at 10 Gb/s.

Figure 13-17 through Figure 13-19 show that phase mismatch is reduced to 0.75° with jog-outs and 0.3° with jog-outs and plane cutouts. The combination of jog-outs and plane cutouts yields simulation results that show the excess capacitance of the structure is reduced to 65 fF.

Designers are tempted to widen lines to compensate for the characteristic impedance increase as the lines are separated and couple less strongly. However, even without widening the lines, the combined capacitance of the corners and jog-outs is still overly capacitive, and therefore the uncoupled section of the jog-out must not be widened.

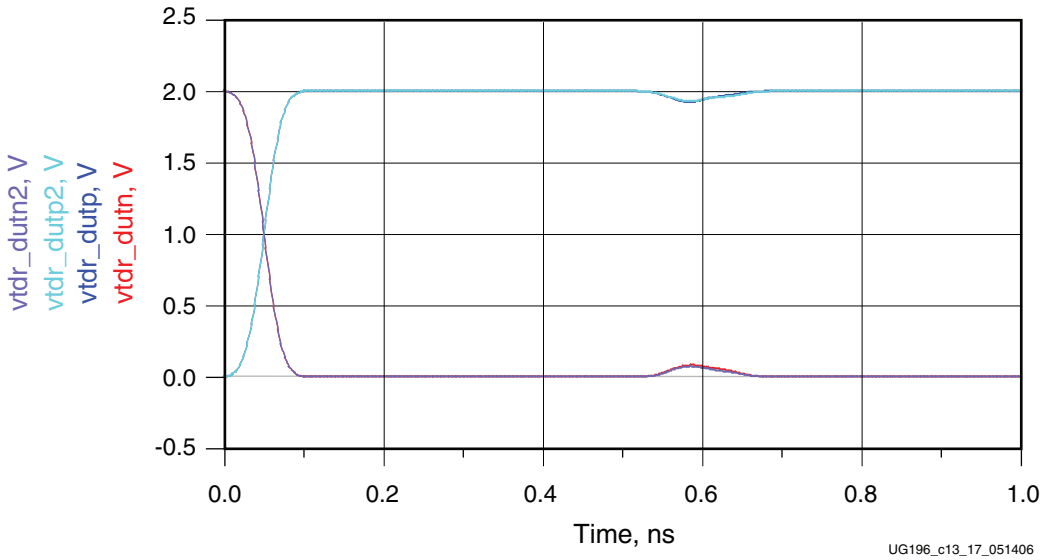


Figure 13-17: **Simulated TDR of 45 Degree Bends with Jog-Outs**

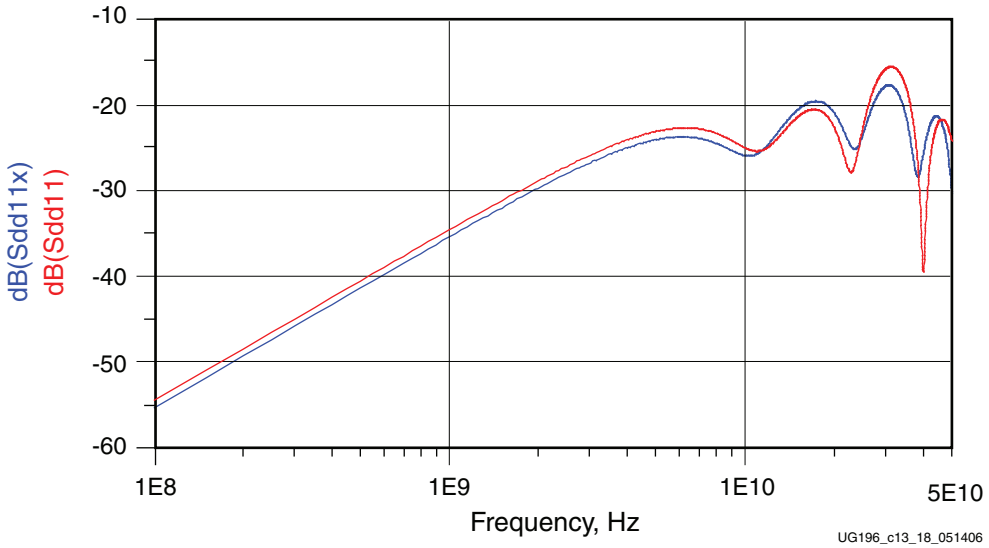


Figure 13-18: **Simulated Return Loss of 45 Degree Bends with Jog-Outs**

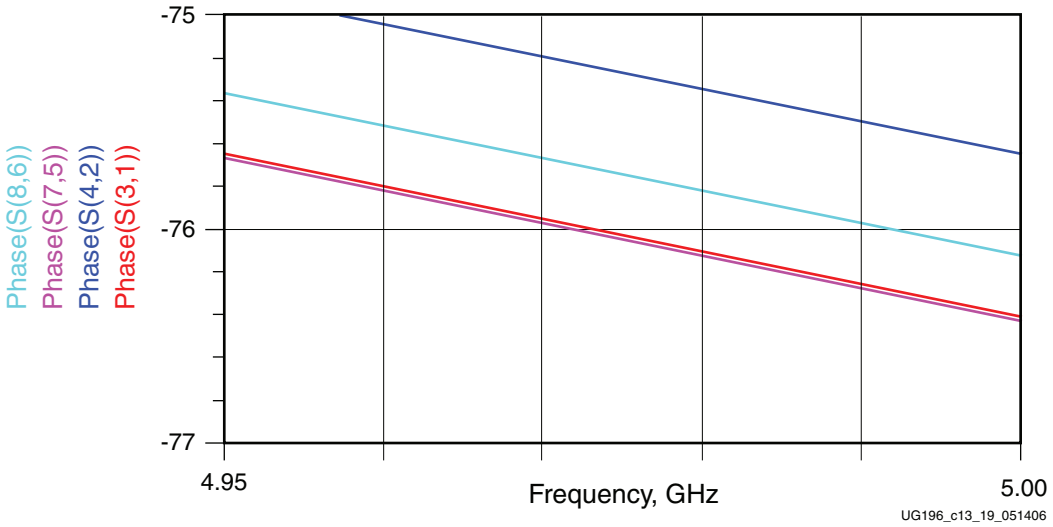
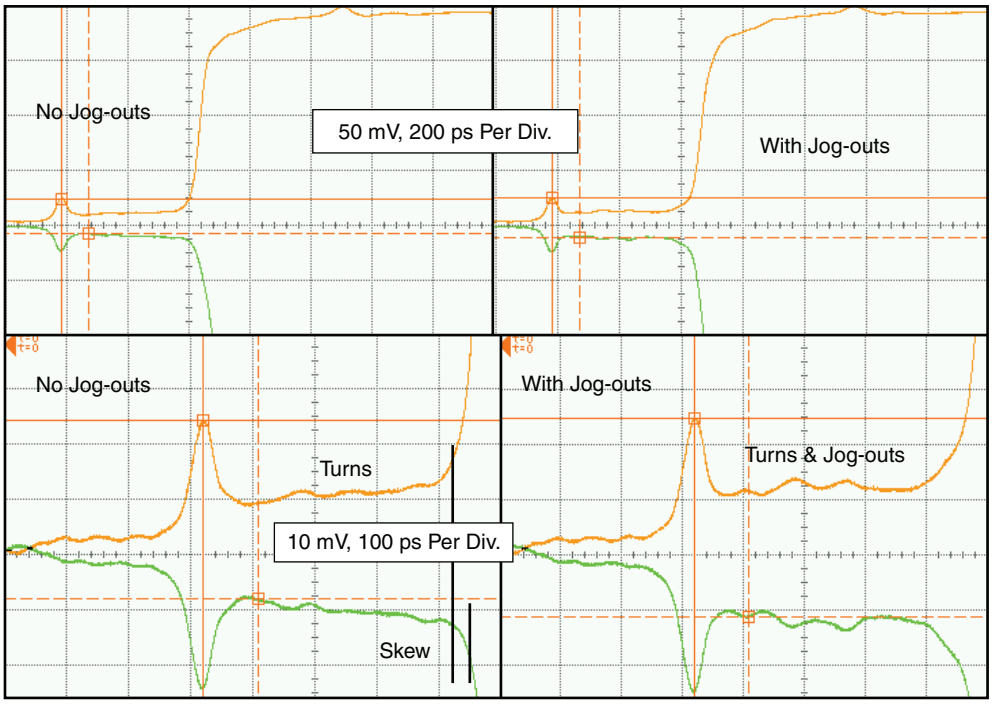


Figure 13-19: **Simulated Phase Response of 45 Degree Bends with Jog-Outs**

For wide traces, curved routing can also be helpful as shown in Figure 13-20.

Apotex Ex1009
Page 274 of 347



UG196_c13_20_051406

Figure 13-20: **Measured TDR of 45 Degree Bends with and without Jog-Outs**

Cpuver Ex1009
Page 276 of 347

Chapter 14

Guidelines and Examples

This chapter discusses high-level PCB guidelines and strategies. Design examples are provided that show how these guidelines are applied how transitions can be modified to accommodate specific applications.

Summary of Guidelines

This high-level summary provides a quick reference to some of the guidelines already covered in previous sections, and also introduces some general strategies when designing high-speed serial channels.

When defining the stack-up, high-speed stripline layers are kept near the bottom of the board. If all high-speed traces can be routed on the top and/or the bottom microstrip layers, there is no need for a stripline layer. Wider traces are preferred and widths of 6 mils to 12 mils are typical.

Unless there are tight space constraints, the differential trace pairs do not need to be coupled closely. For example, instead of using a 5 mil width with 5 mil spacing, the same characteristic trace impedance can be obtained using a 7 mil trace width with 12 mil spacing.

High-speed differential pairs and transitions must be spread apart on adjacent channels generously to limit crosstalk, even if the paths become longer. In most cases, they eventually have to be spread out to match connector pin spacings.

For transitions, large clearances of planes must be provided around and below transitions to limit excess capacitance. Transitions are spaced apart within the same channel. For example, differential vias typically are not placed next to DC blocking capacitors or connectors. However, in some specific cases, performance was acceptable with this placement.

To further limit excess capacitance in vias, the unused pads on vias should be removed and the via stub length is kept to a minimum. By routing from the top microstrip to the bottom microstrip, the via stub can be eliminated. Routing from the top microstrip to the bottom-most stripline layer results in a negligible via stub. If the lowest layers are not available for high-speed striplines, other striplines can be used. However, the via stub should be removed by back-drilling the vias.

Use of minimum spacing and clearance design rules is to be avoided, such as 5 mil pad clearances. These clearances can be detrimental to performance even at lower multi-gigabit rates due to the excess capacitance from the tight spacing.

Most transitions shown in this document have 40 fF to 200 fF of excess capacitance. One exception is a press-fit connector with the PCB pin array having about 500 fF to 800 fF of excess capacitance using these guidelines, with a via stub less than 10 mils. With smaller antipads or longer via stubs, the excess capacitance is much greater. Because the

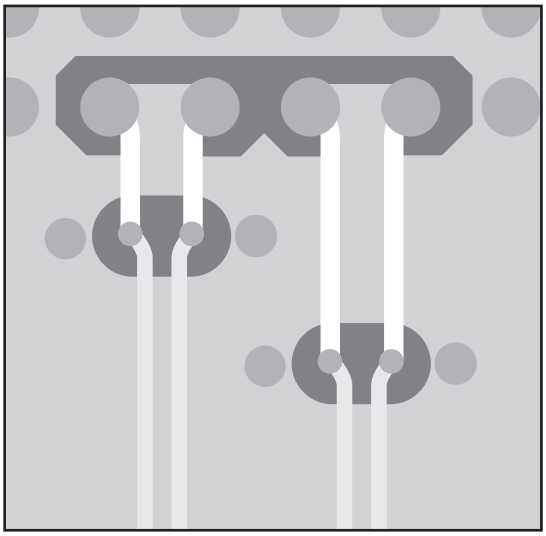
Optiver Ex1009
Page 277 of 347

transceivers have a die capacitance of 500 fF to 600 fF, most transitions can be designed to have a very small impact on performance up to speeds of 10 Gb/s.

These guidelines are recommended to be followed even for designs slower than 10 Gb/s, allowing for more margin at lower speeds such that a smaller output signal swing can be used. Having a 10 Gb/s capable channel also provides the option to upgrade the bandwidth of the system for the next generation product.

BGA Escape Example

The transceiver signal pairs are routed along the edges of the flip-chip BGA. A microstrip is used to escape. When there is adequate spacing from the BGA, the optimized GSSG differential vias are used to change layers, if needed. It is recommended that these vias be staggered, as shown in Figure 14-1, to minimize the formation of slots in the power and ground planes.



UG196_c14_01_051406

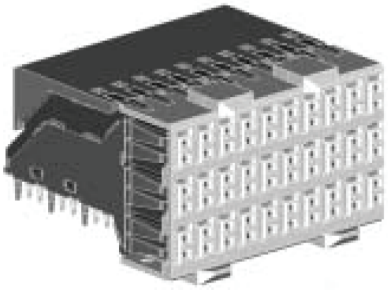
Figure 14-1: **BGA Escape Design Example**

The round BGA pads for the transceiver signals present a small amount of capacitance to a solid PCB ground below. Therefore one option is to open a void in the ground plane below the signal pads with the same diameter as the signal pads. However, simulations show that the void only removes 30 fF of capacitance.

HM-Zd Design Example

For backplane applications, in-line connectors such as the one shown in Figure 14-2, are the most common. Of these connectors, the most common mounting method is press-fit, although SMT connectors offer much better performance.

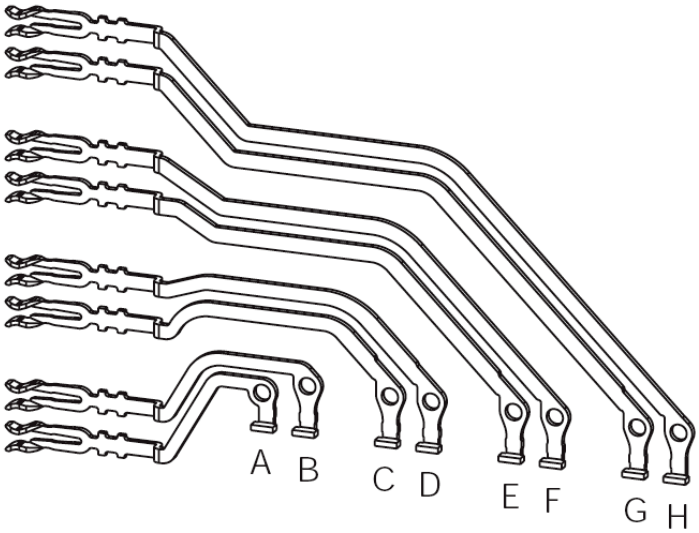
Cpuver Ex1009
Page 278 of 347



UG196_c14_02_051406

Figure 14-2: **Tyco Z-PACK HM-Zd Press-Fit Connector**

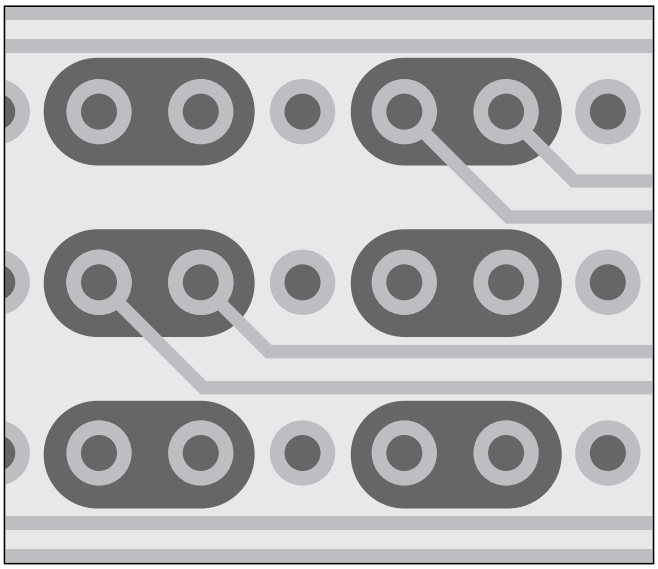
The right-angle connectors have P/N length differences in the signal paths, as shown in Figure 14-3, that require PCB trace lengths to be adjusted to compensate for the skew.



UG196_c14_03_051406

Figure 14-3: **Tyco Z-PACK HM-Zd Press-Fit Connector Internals**

Figure 14-4 shows an example design where the traces are preskewed to compensate for P/N length mismatches within the connector body.



UG196_c14_04_051406

Figure 14-4: **Tyco Z-PACK HM-Zd Press-Fit Connector Design Example**

Press-fit connectors require large vias that allow the connector pins to be inserted. These vias are on a fixed pitch to match the pitch of the connector pins. Having large vias on a tight pitch results in excess capacitance.

To mitigate this excess capacitance, via stubs must be kept short. Because the connector pin is around 95 mils, backdrilling can only be done to that depth. Routing on the lower layers helps to reduce the via stub length.

Making the antipads around the differential vias as large as possible minimizes capacitance. As shown in Figure 14-4, the antipad size is maximized such that the ground reference for the traces extends beyond the edge of the striplines by about 3 mils.

All power and ground planes that do not provide an impedance reference to the striplines or microstrips should be removed. Vias need to be distributed around the periphery of the connector to stitch the ground planes together.

The designer is recommended to taper the wide microstrips or striplines as they enter the connector footprint. However, this technique has not yet been fully validated. The reduced trace width causes additional line loss and inter-symbol interference (ISI) effects from the greater impedance variation. These effects can be offset by the additional performance gained from larger antipads with less excess capacitance.

Cypher Ex1009
Page 280 of 347

Section 3: Appendices

MGT to GTP Transceiver Design Migration

OOB/Beacon Signaling

8B/10B Valid Characters

DRP Address Map of the GTP_DUAL Tile

Low Latency Design

Advanced Clocking

Optiver Ex1009
Page 281 of 347

Cpuver Ex1009
Page 282 of 347

Appendix A

MGT to GTP Transceiver Design Migration

Overview

This appendix describes important differences regarding migration from the Virtex-II Pro and Virtex-4 multi-gigabit transceivers (MGTs) to the Virtex-5 FPGA RocketIO GTP 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-4 FPGAs, refer to *Virtex-II Pro and Virtex-II Pro X Complete Data Sheet* [Ref 9], *RocketIO Transceiver User Guide* [Ref 10], and *Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide* [Ref 11].

Primary Differences

Virtex-5 LXT and SXT FPGAs are a different family from the Virtex-II Pro and Virtex-4 families. The Virtex-5 LXT and SXT devices are not pin compatible with these previous generation devices. However, many aspects of the MGTs and GTP transceivers between the families are the same. The primary differences between these families are:

- Number of MGTs and GTP transceivers per device
- Clocking
- Serial rates and ranges
- Encoding standards – 8B/10B, 64B/66B, SONET, and others
- Clock multiplier settings and PLL ranges
- Flexibility due to partial reconfiguration, PMA programming bus, dynamic reconfiguration port (DRP)
- Board design guidelines

Optiver Ex1009
Page 283 of 347

MGTs per Device

Virtex-5 FPGAs allow for a large range of GTP transceivers per device. Table A-1 shows the number of transceivers available for each family.

Table A-1: **Transceivers per Device**

Virtex Device	Number of Transceivers
Virtex-II Pro FPGA	4, 8, 12, 16, 20
Virtex-4 FPGA	8, 12, 16, 20, 24
Virtex-5 FPGA(1)	8, 12, 16, 24

Notes:

1. Because two GTP transceivers use shared PLL resources in a GTP_DUAL tile, applications where transceivers do not have common clock settings might not be able to use both transceivers in a tile. Thus the total number of available transceivers is reduced in these applications.

Clocking

Virtex devices provide several available clock inputs. Table A-2 shows the clocks for each family and their respective serial speeds.

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

Family	Clock Names	Differential (Internal)	Dedicated Routes	Maximum Serial Speeds (Gb/s)	Dynamic Switching	Package Input Voltage (V)(1)	Inputs per Device	Clocks per Device
Virtex-II Pro FPGA	BREFCLK		Yes	3.125	Yes(2)	2.5	8(3)	2(3)
	BREFCLK2		Yes	3.125	Yes(2)	2.5	8(3)	2(3)
	REFCLK			2.5	Yes(2)	2.5	8(3)	2(3)
	REFCLK2			2.5	Yes(2)	2.5	8(3)	2(3)
Virtex-4 FPGA	GREFCLK	Yes	Yes	1.0	Yes(4)		Note 5	Note 5
	REFCLK1	Yes	Yes	6.5	Yes(4)		8	4
	REFCLK2	Yes	Yes	6.5	Yes(4)		8	4
Virtex-5 FPGA	GREFCLK	Yes	Yes		Yes		1 per GTP_DUAL tile	1 per GTP_DUAL tile
	REFCLK	Yes	Yes	3.75	Yes		1 per GTP_DUAL tile	1 per GTP_DUAL tile

Notes:

1. Nominal values. Refer to the specific data sheet for the exact values.
2. Dynamic selection between the REFCLKs or the BREFCLKs. To switch from REFCLK to BREFCLK or vice versa requires reconfiguration.
3. BREFCLK should use dedicated GCLK I/O, which decreases GCLK I/O resources for other logic (also two pins per clock).
4. Reference clock switching is done via an attribute and the DRP using the RXAPMACLKSEL, RXBPMACLKSEL, and TXABPMACLKSEL attributes. These attributes are located at DRP address `0x5D` on bits [13:12], [11:10], and [9:8], respectively.
5. GREFCLK comes from the global clock tree and can come from any FPGA clock input. It should only be used for serial rates under 1.0 Gb/s.

Apotex Ex1009
Page 284 of 347

Clock selection changed slightly across the first three generations of MGTs. In contrast, the GTP_DUAL tile significantly enhances clocking capabilities by adding dedicated clocks routing and MUXing resources. Figure A-1 shows how the reference clocks are selected for each device.

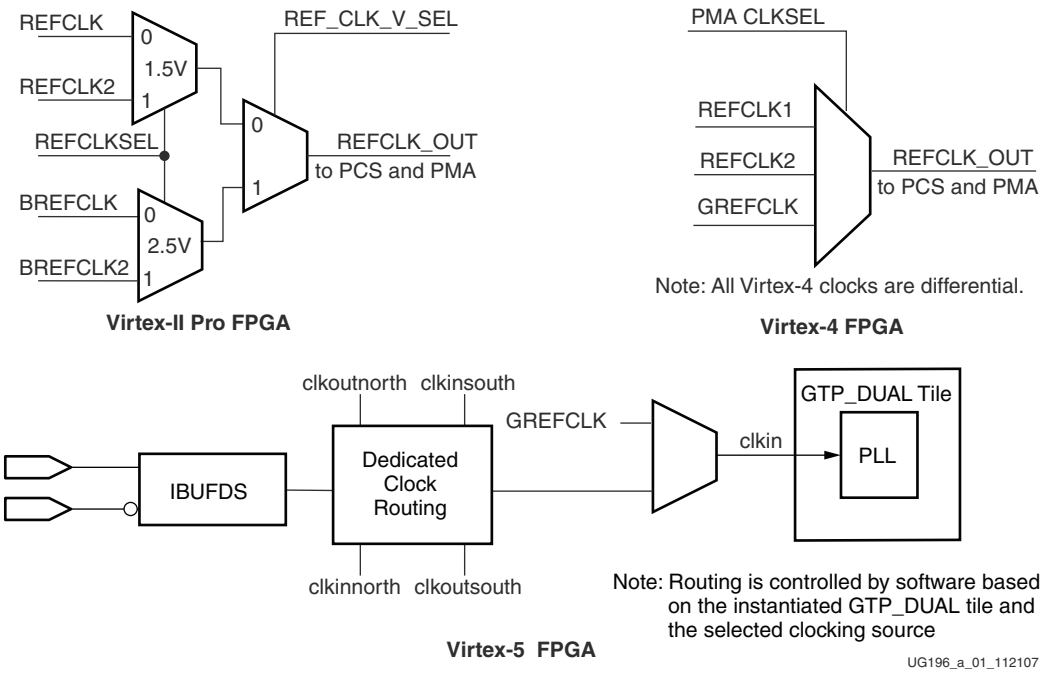


Figure A-1: **Reference Clock Selection for Each Device**

Serial Rate Support

As the Xilinx transceivers continue to migrate, so do the supported serial rates. Table A-3 shows the rates supported by each MGT and GTP transceiver.

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

Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
0.622 – 3.125 Gb/s	0.622 – 6.5 Gb/s	0.100[1] – 3.75 Gb/s

Notes:

1. 100 Mb/s - 500 Mb/s with oversampling.

Encoding Support and Clock Multipliers

Protocol encoding support and clock multiplier support vary depending on the transceiver generation. Table A-4 shows the encoding support available in each MGT and GTP transceiver.

Table A-4: **Encoding Support**

Encoding Schemes	Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
8B/10B	Yes	Yes	Yes
64B/66B	Yes(1)	Yes	Yes(1)
SONET	Yes(1)	Yes	Yes
Others	Yes(2)	Yes(2)	Yes(2)

Notes:

1. Encoding and clocks must be done in the FPGA logic.
2. Depending on the encoding, some functionality must be done in the FPGA logic.

Reference clock multiplication has also evolved over the transceiver generations from a limited selection of multiplier values to a fully programmable solution starting with Virtex-4 devices. Table A-5 shows the clock multiplier values supported in the Virtex-II Pro devices.

Table A-5: **Virtex-II Pro Clock Multipliers**

	Supported Clock Multiplier Values
Virtex-II Pro FPGA	20

The Virtex-4 and Virtex-5 devices use the circuitry shown in Figure A-2 to multiply the reference clock.

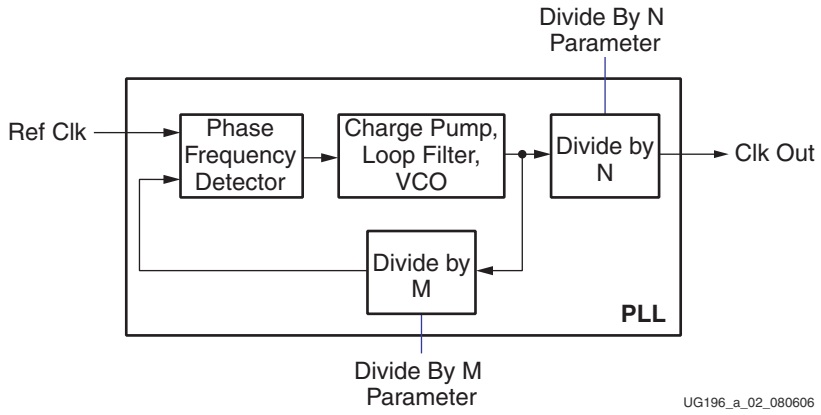


Figure A-2: **Virtex-4 and Virtex-5 Clock Multiplication Circuitry**

Table A-6 shows the parameters used to configure the operation of the clock multiplication circuitry as well as the supported divide values. While Virtex-4 devices support separate multiply ratios for transmit and receive operations, one multiply ratio is used for both in Virtex-5 devices.

Apotex Ex1009
Page 286 of 347

Table A-6: **Virtex-4 and Virtex-5 FPGA Clock Multiplication Parameters**

	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
Divide by M Parameter	TXPLLNDIVSEL, RXPLLNDIVSEL	PLL_DIVSEL_FB
Divide by M Values	8, 10, 16, 20, 32, 40	1, 2, 3, 4, 5
Divide by N Parameter	TXOUTDIV2SEL, RXOUTDIV2SEL	PLL_DIVSEL_REF
Divide by N Values	1, 2, 4, 8, 16, 32	1, 2

Flexibility

In Virtex-II Pro devices, changing of attributes requires partial reconfiguration. Virtex-4 and Virtex-5 devices allow all attribute changes from the DRP, and any default values can be set in the HDL itself.

Board Guidelines

Power Supply Filtering

The Virtex-5 FPGA GTP transceivers simplify power supply design over previous generations in two ways:

1. Only two supply voltages are needed to power the transceivers.
2. Because the two transceivers that share a tile have common power pins, the number of power supply filtering components is reduced.

Table A-7 shows the power pin voltages for all Virtex families, and Figure A-3 shows the power supply filtering for all Virtex families.

Table A-7: **Power Pin Voltages**

Pin	Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver(1)
AVCCAUXRX	2.5V	1.2V	-
AVCCAUXTX	2.5V	1.2V	-
AVCCAUXMGT	N/A	2.5V	-
VTTX	1.8 - 2.5V(2)	1.5V	-
VTRX	1.5 - 2.5V(2)	0.25 - 2.5V(2)	-
MGTAVCCPLL	-	-	1.2V
MGTAVCC	-	-	1.0V
MGTAVTTTX	-	-	1.2V
MGTAVTTRX	-	-	1.2V
MGTAVTTRXC	-	-	1.2V

Notes:

1. Nominal values. Refer to the device data sheets for values and operating conditions.
2. Depends on AC/DC coupling or termination options. See the device user guides for more details.

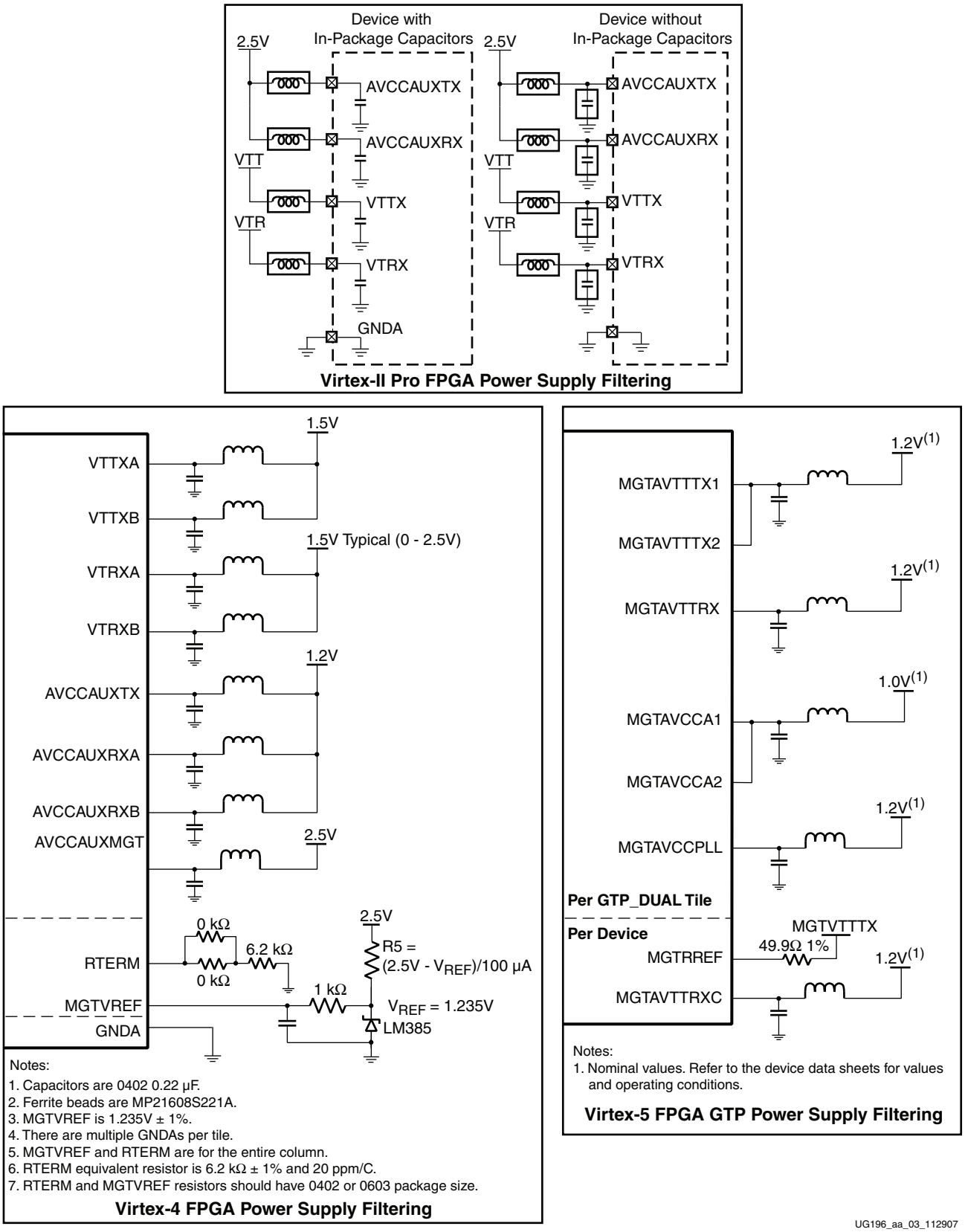


Figure A-3: **Virtex-II, Virtex-II Pro, Virtex-4, Virtex-5 FPGA GTP Power Supply Filtering**

Broadcom Ex1009
Page 288 of 347

Other Minor Differences

Termination

In Virtex-II Pro devices, the transceivers contain on-chip termination and reference voltages for VTTX and VTRX. In Virtex-4 devices, the MGTs use a reference resistor to create the termination circuitry for each MGT column. The Virtex-5 FPGA GTP transceiver simplifies the termination circuitry by providing a calibrated 50Ω termination. Table A-8 shows the termination options for each FPGA generation.

Table A-8: **Termination Options**

Termination	Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
Value	50/75Ω	50Ω	50Ω
Voltage Pins	VTTX/VTRX	VTTX/VTRX	MGTAVTTTX/MGTAVTTRX

FPGA Logic Interface

FPGA logic interface width support varies depending on the transceiver. Table A-9 shows the widths supported in all MGT and GTP transceivers.

Table A-9: **FPGA Logic Interface Support**

Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
1-byte, 2-byte, 4-byte, 8-byte	1-byte, 2-byte, 4-byte, 8-byte	1-byte, 2-byte

CRC

CRC support has changed over the generations of transceivers. Table A-10 shows the CRC support provided by all four transceiver families.

Table A-10: **CRC Transceiver Support**

Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
CRC-32	Independent CRC-32 block supports datapath from 8 to 64 bits wide.	Independent CRC block supports two CRC-32 calculations over 32-bit datapaths or a single CRC-32 calculation over a 64-bit datapath.

Loopback

The loopback options of the Virtex transceivers have evolved to improve flexibility. Virtex-II Pro MGTs have two loopback modes, and Virtex-4 MGTs have four loopback modes. Table A-11 shows this evolution.

Table A-11: **Loopback Options**

Mode	Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
Parallel Loopback (Tx → RX)	Yes	Yes	Yes
Serial Pre-Driver	–	Yes	Yes
Serial Post-Driver	Yes	–	–
Serial Digital Receiver	–	Yes	–
External Data PMA-Only Parallel Loopback	–	–	Yes
PCI Express Repeater	–	–	Yes

Serialization

As in Virtex-4 devices, Virtex-5 FPGA GTP transceivers serialize and send the least significant byte first. Virtex-II Pro devices send the most-significant byte first.

Defining Clock Correction and Channel Bonding Sequences

The bit definitions of CLK_COR_SEQ and CHAN_BOND_SEQ have changed to support more encoding functionality. Table A-12 illustrates the differences.

Table A-12: **CLK_COR_SEQ and CHAN_BOND_SEQ Sequences**

Bit Definition	Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver
8B/10B encoded definition	`00110111100`(1,4)	`00110111100`(2,4)	`0110111100`(3,5)
10-bit literal value	`10011111010`(1,4)	`10011111010`(2,4)	`0011111010`(1,5)
64B/66B encoding (sync character)	N/A	`1XX` (sync header)	N/A
8-bit literal value (for 64B/66B and other encodings)	N/A	`1XX` (8-bit data)	N/A

Notes:

1. Defines K28.5.
2. Defines K28.5 and depends on CLK_COR_8B10B_DE (plus all 10 bits are defined).
3. Defines K28.5 and regular disparity.
4. For Virtex-II Pro and Virtex-4 MGTs, the 11th bit (left-most bit) determines either an 8-bit or 10-bit compare.
5. For Virtex-5 FPGA GTP transceivers, the RX_DECODE_SEQ_MATCH attribute determines if matches occur against the 8-bit output of the 8B/10B decoder (RX_DECODE_SEQ_MATCH = TRUE) or against the undecoded incoming data (RX_DECODE_SEQ_MATCH = FALSE). The undecoded data can be either 8 bits (INTDATAWIDTH is Low) or 10 bits (INTDATAWIDTH is High).

Apotex Ex1009
Page 290 of 347

RXSTATUS Bus

Several buses have changed over the FPGA generations to improve the information that is indicated. Table A-13 shows the migration from Virtex-II Pro to Virtex-5 devices.

Table A-13: **Status Bus Changes**

Description	Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver(1)
Indicates channel bonding complete	CHBONDONE(2)	RXSTATUS[5]	RXCHANISALIGNED
Indicates status bus is status, data, event	N/A	RXSTATUS[4:3]	RXSTATUS[2:0]
Indicates channel bonding or clock correction pointers change	RXCLKCORCNT	RXSTATUS[2:0]	RXCHANREALIGN
Indicates that an RX buffer has underflow/overflow	RXBUFSTATUS[1]	RXBUFERR	RXBUFSTATUS[2:0]

Notes:

1. Signal optimization settings are independent between both GTP transceivers of a GTP_DUAL tile. GTP0 is indicated by the suffix "0" after the signal name, and GTP1 is indicated by the suffix "1" after the signal name (for example, RXENEQB0 or RXENEQB1).
2. RXCLKCORCNT must go to `3'b101` before channel bonding is complete.

Pre-emphasis, Differential Swing, and Equalization

The differential signaling techniques are more robust in recent Xilinx transceivers. The Virtex-5 FPGA GTP transceiver adds ports to control TX characteristics to simplify reconfiguration. Table A-14 shows the migration of attributes from Virtex-II Pro and Virtex-4 MGTs to Virtex-5 FPGA GTP transceivers.

Table A-14: **Signal Optimization Attributes and Ports**

Description	Virtex-II Pro MGT	Virtex-4 MGT	Virtex-5 FPGA GTP Transceiver(1)	
			Attributes	Ports
Controls TX pre-emphasis and edge rate	TX_PREMPHASIS	TXPRE_PRDRV_DAC TXPRE_TAP_PD TXSLEWRATE TXPOST_PRDRV_DAC TXDAT_PRDRV_DAC TXPOST_TAP_PD		TXPREEMPHASIS[2:0]
Controls differential amplitude of the transmitted signal	TX_DIFF_CTRL	TXPRE_TAP_DAC TXPOST_TAP_DAC TXDAT_TAP_DAC	TX_DIFF_BOOST	TXBUFDIFFCTRL TXDIFFCTRL
Active equalization	N/A	RXAFEEQ	–	RXENEQB RXEQPOLE[3:0] RXEQMIX[1:0]
Discrete equalization	N/A	RXEQ	N/A	N/A

Notes:

1. Signal optimization settings are independent between both GTP transceivers of a GTP_DUAL tile. GTP0 is indicated by the suffix "0" after the signal name, and GTP1 is indicated by the suffix "1" after the signal name (for example RXENEQB0, RXENEQB1).

Cpuver Ex1009
Page 292 of 347

Appendix B

OOB/Beacon Signaling

The GTP transceiver supports Out-of-Band (OOB) signaling for conformance with standards such as SATA and supports beaconing for conformance to the PCI Express specification. OOB signaling mechanisms are used to send low-speed signals between the transmitter and receiver when high-speed serial data transmission is not active, typically when the link is in a power-down state or has not been initialized.

OOB signaling uses non-differential transitions on the differential inputs of the receiver to send information. To send an OOB signal, transmitters drive their serial differential output pins to the same voltage, resulting in a reduced differential voltage between the pins. When the absolute differential voltage drops below a preset threshold level, the receiver detects the signal as an OOB signal. Figure B-1 illustrates this concept.

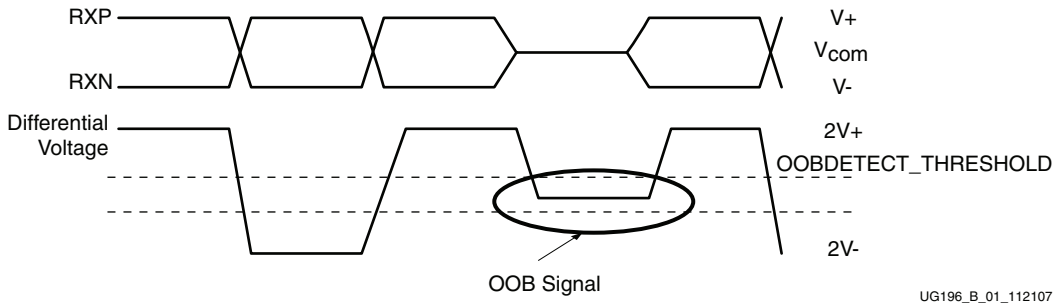


Figure B-1: **OOB Signaling**

TX OOB/Beacon Signaling, page 133 and RX OOB/Beacon Signaling, page 143 provide details of the support for OOB signaling in the GTP transceiver TX and RX logic, respectively. The remainder of this section summarizes the use of OOB signaling in SATA operation and beaconing for PCI Express operation.

OOB Signaling for SATA Operation

SATA operation uses OOB signals as part of its COMWAKE, COMINIT, and COMRESET sequences as shown in Figure B-2. These sequences consist of a fixed length burst of non-OOB data followed by an OOB signal (called an idle signal in SATA). The length of the idle defines the type of COM sequence being received: COMWAKE sequences use 106.7 ns idles, and COMINIT/COMRESET sequences use 320 ns idles. A COM sequence is valid when it is received four times consecutively.

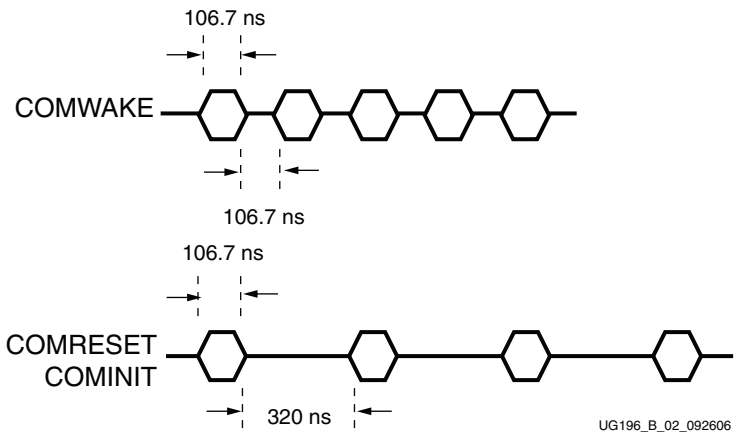


Figure B-2: **SATA COM Sequences**

In addition to the analog circuitry required to encode and detect the OOB signal state, the GTP transceiver includes state machines to format and decode bursts of OOB/beacon signals for SATA (COMRESET, COMWAKE, and COMINIT).

Beacon Signaling for PCI Express Operation

The PCI Express specification uses sequences called *beacons* to wake endpoints from power-down states. A beacon is the transmission of K28.5 (COM) characters. A beacon sequence can have a frequency anywhere from 30 KHz to 500 MHz.

The PCI Express specification describes the beacon mechanism as an inband wake-up indication, and defines the use of a discrete wake signal (WAKE#) as an out-of-band mechanism.

GTP transceiver support for PCI Express beacons uses interface signals defined in the *PHY Interface for the PCI Express (PIPE) Specification*. Control logic in the FPGA manages the format of the beacon sequence.

Apotex Ex1009
Page 294 of 347

Appendix C

8B/10B Valid 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 C-1 shows the valid Data characters. Table C-2, page 303 shows the valid K characters.

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + 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

Oplver Ex1009
Page 296 of 347

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
D21.1	001 10101	101010 1001	101010 1001
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

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
D21.2	010 10101	101010 0101	101010 0101
D22.2	010 10110	011010 0101	011010 0101
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

Apotex Ex1009
Page 298 of 347

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
D21.3	011 10101	101010 1100	101010 0011
D22.3	011 10110	011010 1100	011010 0011
D23.3	011 10111	111010 0011	000101 1100
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

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
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
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

Apotex Ex1009
Page 300 of 347

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
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
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

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
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
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

Apotex Ex1009
Page 302 of 347

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

Data Byte Name	Bits HGF EDCBA	Current RD – abcdei fghj	Current RD + abcdei fghj
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
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 C-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.

Optiver Ex1009
Page 303 of 347

Oplover Ex1009
Page 304 of 347

Appendix D

DRP Address Map of the GTP_DUAL Tile

All attributes are stored as binary values in the DRP table. Some attributes use special mappings from their UCF/HDL values to their binary values. The mappings for these values are shown in Table D-1.

For attributes not listed in Table D-1, use the following rules to determine the binary mapping:

- Attributes with TRUE/FALSE values use “1” to represent TRUE and “0” to represent FALSE.
- Convert integer values to binary.

Table D-1: **Special Attribute Mappings**

Attribute	UCF/HDL Attribute Value	DRP Binary Value
ALIGN_COMMA_WORD	1	0
	2	1
CHAN_BOND_MODE	OFF	00
	MASTER	01
	SLAVE	10
CLK25_DIVIDER	1	000
	2	001
	3	010
	4	011
	5	100
	6	101
	10	110
	12	111

Optiver Ex1009
Page 305 of 347

Table D-1: **Special Attribute Mappings *(Continued)***

Attribute	UCF/HDL Attribute Value	DRP Binary Value
OOB_CLK_DIVIDER	1	000
	2	001
	4	010
	6	011
	8	100
	10	101
	12	110
	14	111
PLL_DIVSEL_FB	1	10000
	2	00000
	3	00001
	4	00010
	5	00011
PLL_DIVSEL_REF	1	010000
	2	000000
PLL_RXDIVSEL_OUT	1	00
	2	01
	4	10
PLL_TXDIVSEL_COMM_OUT	1	00
	2	01
	4	10
PLL_TXDIVSEL_OUT	1	00
	2	01
	4	10
REFCLK_SEL[2:0]	The encoding of the attribute REFCLK_SEL[2:0] is discussed in Appendix F, Advanced Clocking.	

Optiver Ex1009
Page 306 of 347

Table D-1: **Special Attribute Mappings *(Continued)***

Attribute	UCF/HDL Attribute Value	DRP Binary Value
RX_LOS_INVALID_INCR	1	000
	2	001
	4	010
	8	011
	16	100
	32	101
	64	110
	128	111
RX_LOS_THRESHOLD	4	000
	8	001
	16	010
	32	011
	64	100
	128	101
	256	110
	512	111
RX_SLIDE_MODE	PCS	0
	PMA	1
RX_STATUS_FMT	PCIE	0
	SATA	1
RX_XCLK_SEL	RXREC	0
	RXUSR	1
TX_XCLK_SEL	TXOUT	0
	TXUSR	1

DRP Address by Attribute

Table D-2 lists the DRP addresses according to attribute name.

Table D-2: **DRP Address by Attribute**

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AC_CAP_DIS_0																																49<14>
AC_CAP_DIS_1																																6<1>
ALIGN_COMMA_WORD_0																																2b<12>
ALIGN_COMMA_WORD_1																																24<3>
CHAN_BOND_1_MAX_SKEW_0																													2b<13>	2b<14>	2b<15>	2c<0>
CHAN_BOND_1_MAX_SKEW_1																													24<2>	24<1>	24<0>	23<15>
CHAN_BOND_2_MAX_SKEW_0																													2c<1>	2c<2>	2c<3>	2c<4>
CHAN_BOND_2_MAX_SKEW_1																													23<14>	23<13>	23<12>	23<11>
CHAN_BOND_LEVEL_0																														2c<5>	2c<6>	2c<7>
CHAN_BOND_LEVEL_1																														23<10>	23<9>	23<8>
CHAN_BOND_MODE_0																															2c<8>	2c<9>
CHAN_BOND_MODE_1																															23<7>	23<6>
CHAN_BOND_SEQ_1_1_0																							2c<10>	2c<11>	2c<12>	2c<13>	2c<14>	2c<15>	2d<0>	2d<1>	2d<2>	2d<3>

Optiver Ex1009
Page 309 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHAN_BOND_SEQ_1_1_1																							23<5>	23<4>	23<3>	23<2>	23<1>	23<0>	22<15>	22<14>	22<13>	22<12>
CHAN_BOND_SEQ_1_2_0																							2d<4>	2d<5>	2d<6>	2d<7>	2d<8>	2d<9>	2d<10>	2d<11>	2d<12>	2d<13>
CHAN_BOND_SEQ_1_2_1																							22<11>	22<10>	22<9>	22<8>	22<7>	22<6>	22<5>	22<4>	22<3>	22<2>
CHAN_BOND_SEQ_1_3_0																							2d<14>	2d<15>	2e<0>	2e<1>	2e<2>	2e<3>	2e<4>	2e<5>	2e<6>	2e<7>
CHAN_BOND_SEQ_1_3_1																							22<1>	22<0>	21<15>	21<14>	21<13>	21<12>	21<11>	21<10>	21<9>	21<8>
CHAN_BOND_SEQ_1_4_0																							2e<8>	2e<9>	2e<10>	2e<11>	2e<12>	2e<13>	2e<14>	2e<15>	2f<0>	2f<1>
CHAN_BOND_SEQ_1_4_1																							21<7>	21<6>	21<5>	21<4>	21<3>	21<2>	21<1>	21<0>	20<15>	20<14>
CHAN_BOND_SEQ_1_ENABLE_0																												2f<2>	2f<3>	2f<4>	2f<5>	
CHAN_BOND_SEQ_1_ENABLE_1																												20<13>	20<12>	20<11>	20<10>	
CHAN_BOND_SEQ_2_1_0																							2f<6>	2f<7>	2f<8>	2f<9>	2f<10>	2f<11>	2f<12>	2f<13>	2f<14>	2f<15>
CHAN_BOND_SEQ_2_1_1																							20<9>	20<8>	20<7>	20<6>	20<5>	20<4>	20<3>	20<2>	20<1>	20<0>
CHAN_BOND_SEQ_2_2_0																							30<0>	30<1>	30<3>	30<4>	30<5>	30<6>	30<7>	30<8>	30<9>	30<10>

Optiver Ex1009
Page 310 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHAN_BOND_SEQ_2_2_1																							1f<15>	1f<14>	1f<12>	1f<11>	1f<10>	1f<9>	1f<8>	1f<7>	1f<6>	1f<5>
CHAN_BOND_SEQ_2_3_0																							30<11>	47<1>	47<2>	47<3>	47<4>	47<5>	47<6>	47<7>	47<8>	47<9>
CHAN_BOND_SEQ_2_3_1																							1f<4>	8<14>	8<13>	8<12>	8<11>	8<10>	8<9>	8<8>	8<7>	8<6>
CHAN_BOND_SEQ_2_4_0																							47<10>	47<11>	47<12>	47<13>	47<14>	47<15>	48<0>	48<1>	48<2>	48<3>
CHAN_BOND_SEQ_2_4_1																							8<5>	8<4>	8<3>	8<2>	8<1>	8<0>	7<15>	7<14>	7<13>	7<12>
CHAN_BOND_SEQ_2_ENABLE_0																												3a<1>	3a<0>	39<15>	39<14>	
CHAN_BOND_SEQ_2_ENABLE_1																												15<14>	15<15>	16<0>	16<1>	
CHAN_BOND_SEQ_2_USE_0																																39<13>
CHAN_BOND_SEQ_2_USE_1																																16<2>
CHAN_BOND_SEQ_LEN_0																															39<12>	39<11>
CHAN_BOND_SEQ_LEN_1																															16<3>	16<4>
CLK_COR_ADJ_LEN_0																															39<10>	39<9>
CLK_COR_ADJ_LEN_1																															16<5>	16<6>

Optiver Ex1009
Page 311 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK_COR_DET_LEN_0																															39<8>	39<7>
CLK_COR_DET_LEN_1																															16<7>	16<8>
CLK_COR_INSERT_IDLE_FLAG_0																																39<6>
CLK_COR_INSERT_IDLE_FLAG_1																																16<9>
CLK_COR_KEEP_IDLE_0																																39<5>
CLK_COR_KEEP_IDLE_1																																16<10>
CLK_COR_MAX_LAT_0																											39<4>	39<3>	39<2>	39<1>	39<0>	38<15>
CLK_COR_MAX_LAT_1																											16<11>	16<12>	16<13>	16<14>	16<15>	17<0>
CLK_COR_MIN_LAT_0																											38<14>	38<13>	38<12>	38<11>	38<10>	38<9>
CLK_COR_MIN_LAT_1																											17<1>	17<2>	17<3>	17<4>	17<5>	17<6>
CLK_COR_PRECEDENCE_0																																38<8>
CLK_COR_PRECEDENCE_1																																17<7>
CLK_COR_REPEAT_WAIT_0																												38<6>	38<5>	38<4>	38<3>	38<2>

Optiver Ex1009
Page 312 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK_COR_REPEAT_WAIT_1																												17<9>	17<10>	17<11>	17<12>	17<13>
CLK_COR_SEQ_1_1_0																							38<1>	38<0>	37<15>	37<14>	37<13>	37<12>	37<11>	37<10>	37<9>	37<8>
CLK_COR_SEQ_1_1_1																							17<14>	17<15>	18<0>	18<1>	18<2>	18<3>	18<4>	18<5>	18<6>	18<7>
CLK_COR_SEQ_1_2_0																							37<7>	37<6>	37<5>	37<4>	37<3>	37<2>	37<1>	37<0>	36<15>	36<14>
CLK_COR_SEQ_1_2_1																							18<8>	18<9>	18<10>	18<11>	18<12>	18<13>	18<14>	18<15>	19<0>	19<1>
CLK_COR_SEQ_1_3_0																							36<13>	36<12>	36<11>	36<10>	36<9>	36<8>	36<7>	36<6>	36<5>	36<4>
CLK_COR_SEQ_1_3_1																							19<2>	19<3>	19<4>	19<5>	19<6>	19<7>	19<8>	19<9>	19<10>	19<11>
CLK_COR_SEQ_1_4_0																							36<3>	36<2>	36<1>	36<0>	35<15>	35<14>	35<13>	35<12>	35<11>	35<10>
CLK_COR_SEQ_1_4_1																							19<12>	19<13>	19<14>	19<15>	1a<0>	1a<1>	1a<2>	1a<3>	1a<4>	1a<5>
CLK_COR_SEQ_1_ENABLE_0																												35<9>	35<8>	35<7>	35<6>	
CLK_COR_SEQ_1_ENABLE_1																												1a<6>	1a<7>	1a<8>	1a<9>	
CLK_COR_SEQ_2_1_0																							35<5>	35<4>	35<3>	35<2>	35<1>	35<0>	34<15>	34<14>	34<13>	34<12>

Optiver Ex1009
Page 313 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK_COR_SEQ_2_1_1																							1a<10>	1a<11>	1a<12>	1a<13>	1a<14>	1a<15>	1b<0>	1b<1>	1b<2>	1b<3>
CLK_COR_SEQ_2_2_0																							34<11>	34<10>	34<9>	34<8>	34<7>	34<6>	34<5>	34<4>	34<3>	34<2>
CLK_COR_SEQ_2_2_1																							1b<4>	1b<5>	1b<6>	1b<7>	1b<8>	1b<9>	1b<10>	1b<11>	1b<12>	1b<13>
CLK_COR_SEQ_2_3_0																							34<1>	34<0>	33<15>	33<14>	33<13>	33<12>	33<11>	33<10>	33<9>	33<8>
CLK_COR_SEQ_2_3_1																							1b<14>	1b<15>	1c<0>	1c<1>	1c<2>	1c<3>	1c<4>	1c<5>	1c<6>	1c<7>
CLK_COR_SEQ_2_4_0																							33<7>	33<6>	33<5>	33<4>	33<3>	33<2>	33<1>	33<0>	32<15>	32<14>
CLK_COR_SEQ_2_4_1																							1c<8>	1c<9>	1c<10>	1c<11>	1c<12>	1c<13>	1c<14>	1c<15>	1d<0>	1d<1>
CLK_COR_SEQ_2_ENABLE_0																												32<13>	32<12>	32<11>	32<10>	
CLK_COR_SEQ_2_ENABLE_1																												1d<2>	1d<3>	1d<4>	1d<5>	
CLK_COR_SEQ_2_USE_0																																32<9>
CLK_COR_SEQ_2_USE_1																																1d<6>
CLK_CORRECT_USE_0																																38<7>

Optiver Ex1009
Page 314 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLK_CORRECT_USE_1																																17<8>
CLK25_DIVIDER																														26<9>	26<10>	26<11>
CLKINDC_B																																4<3>
CLKNORTH_SEL																																4<8>
CLKSOUTH_SEL																																4<7>
COM_BURST_VAL_0																													32<8>	32<7>	32<6>	32<5>
COM_BURST_VAL_1																													1d<7>	1d<8>	1d<9>	1d<10>
COMMA_10B_ENABLE_0																							32<4>	32<3>	32<2>	32<1>	32<0>	31<15>	31<14>	31<13>	31<12>	31<11>
COMMA_10B_ENABLE_1																							1d<11>	1d<12>	1d<13>	1d<14>	1d<15>	1e<0>	1e<1>	1e<2>	1e<3>	1e<4>
COMMA_DOUBLE_0																																31<10>
COMMA_DOUBLE_1																																1e<5>
DEC_MCOMMA_DETECT_0																																31<9>
DEC_MCOMMA_DETECT_1																																31<8>

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit																															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
DEC_PCOMMA_DETECT_0	1e<6>																															
DEC_PCOMMA_DETECT_1	1e<7>																															
DEC_VALID_COMMA_ONLY_0	31<7>																															
DEC_VALID_COMMA_ONLY_1	1e<8>																															
MCOMMA_10B_VALUE_0	30<13>	30<14>	30<15>	31<0>	31<1>	31<2>	31<3>	31<4>	31<5>	31<6>																						
MCOMMA_10B_VALUE_1	1f<2>	1f<1>	1f<0>	1e<15>	1e<14>	1e<13>	1e<12>	1e<11>	1e<10>	1e<9>																						
MCOMMA_DETECT_0	30<12>																															
MCOMMA_DETECT_1	1f<3>																															
OOB_CLK_DIVIDER	26<14>	26<13>	26<12>																													
OOBDETECT_THRESHOLD_0	3a<3>	3a<4>	3a<5>																													
OOBDETECT_THRESHOLD_1	15<12>	15<11>	15<10>																													
OVERSAMPLE_MODE	26<15>																															
PCI_EXPRESS_MODE_0	46<15>																															

Optiver Ex1009
Page 315 of 347

Optiver Ex1009
Page 316 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCI_EXPRESS_MODE_1																																9<0>
PCOMMA_10B_VALUE_0																							46<14>	46<13>	46<12>	46<11>	46<10>	46<9>	46<8>	46<7>	46<6>	46<5>
PCOMMA_10B_VALUE_1																							9<1>	9<2>	9<3>	9<4>	9<5>	9<6>	9<7>	9<8>	9<9>	9<10>
PCS_COM_CFG					27<0>	27<1>	27<2>	27<3>	27<4>	27<5>	27<6>	27<7>	27<8>	27<9>	27<10>	27<11>	27<12>	27<13>	27<14>	27<15>	28<0>	28<1>	28<2>	28<3>	28<4>	28<5>	28<6>	28<7>	28<8>	28<9>	28<10>	28<11>
PCOMMA_DETECT_0																																46<4>
PCOMMA_DETECT_1																																9<11>
PLL_DIVSEL_FB																												28<12>	28<13>	28<14>	28<15>	29<0>
PLL_DIVSEL_REF																											4<14>	4<13>	4<12>	4<11>	4<10>	4<9>
PLL_RXDIVSEL_OUT_0																															46<3>	46<2>
PLL_RXDIVSEL_OUT_1																															9<15>	0a<0>
PLL_SATA_0																																46<1>
PLL_SATA_1																																9<14>
PLL_TXDIVSEL_COMM_OUT																															4a<8>	4a<9>

Optiver Ex1009
Page 317 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL_TXDIVSEL_OUT_0																															46<0>	45<15>
PLL_TXDIVSEL_OUT_1																															5<3>	5<4>
PMA_CDR_SCAN_0						45<14>	45<13>	45<12>	45<11>	45<10>	45<9>	45<8>	45<7>	45<6>	45<5>	45<4>	45<3>	45<2>	45<1>	45<0>	44<15>	44<14>	44<13>	44<12>	44<11>	44<10>	44<9>	44<8>	44<7>	44<6>	44<5>	44<4>
PMA_CDR_SCAN_1						0a<1>	0a<2>	0a<3>	0a<4>	0a<5>	0a<6>	0a<7>	0a<8>	0a<9>	0a<10>	0a<11>	0a<12>	0a<13>	0a<14>	0a<15>	0b<0>	0b<1>	0b<2>	0b<3>	0b<4>	0b<5>	0b<6>	0b<7>	0b<8>	0b<9>	0b<10>	0b<11>
PMA_RX_CFG_0								49<12>	48<9>	49<8>	49<7>	49<6>	49<5>	49<4>	49<3>	49<2>	49<1>	49<0>	48<15>	49<13>	49<9>	48<8>	48<7>	48<6>	48<5>	48<4>	48<13>	48<12>	48<11>	48<10>	49<11>	49<10>
PMA_RX_CFG_1								6<3>	7<6>	6<7>	6<8>	6<9>	6<10>	6<11>	6<12>	6<13>	6<14>	6<15>	7<0>	6<2>	6<6>	7<7>	7<8>	7<9>	7<10>	7<11>	7<2>	7<3>	7<4>	7<5>	6<4>	6<5>
PRBS_ERR_THRESHOLD_0	44<3>	44<2>	44<1>	44<0>	43<15>	43<14>	43<13>	43<12>	43<11>	43<10>	43<9>	43<8>	43<7>	43<6>	43<5>	43<4>	43<3>	43<2>	43<1>	43<0>	42<15>	42<14>	42<13>	42<12>	42<11>	42<10>	42<9>	42<8>	42<7>	42<6>	42<5>	42<4>
PRBS_ERR_THRESHOLD_1	0b<12>	0b<13>	0b<14>	0b<15>	0c<0>	0c<1>	0c<2>	0c<3>	0c<4>	0c<5>	0c<6>	0c<7>	0c<8>	0c<9>	0c<10>	0c<11>	0c<12>	0c<13>	0c<14>	0c<15>	0d<0>	0d<1>	0d<2>	0d<3>	0d<4>	0d<5>	0d<6>	0d<7>	0d<8>	0d<9>	0d<10>	0d<11>
RCV_TERM_GND_0																																4a<0>
RCV_TERM_GND_1																																5<15>
RCV_TERM_MID_0																																49<15>
RCV_TERM_MID_1																																6<0>
RCV_TERM_VTTRX_0																																4a<1>

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit																															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RCV_TERM_VTTRX_1	5<14>																															
REFCLK_SEL[2:0]	4<6>	4<5>	4<4>																													
RX_BUFFER_USE_0	42<3>																															
RX_BUFFER_USE_1	0d<12>																															
RX_DECODE_SEQ_MATCH_0	42<2>																															
RX_DECODE_SEQ_MATCH_1	0d<13>																															
RX_LOS_INVALID_INCR_0	41<15>	42<0>	42<1>																													
RX_LOS_INVALID_INCR_1	0e<0>	0d<15>	0d<14>																													
RX_LOS_THRESHOLD_0	41<11>	41<12>	41<13>																													
RX_LOS_THRESHOLD_1	0e<4>	0e<3>	0e<2>																													
RX_LOSS_OF_SYNC_FSM_0	41<14>																															
RX_LOSS_OF_SYNC_FSM_1	0e<1>																															
RX_SLIDE_MODE_0	41<10>																															

Optiver Ex1009
Page 318 of 347

Optiver Ex1009
Page 319 of 347

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_SLIDE_MODE_1																																0e<5>
RX_STATUS_FMT_0																																41<9>
RX_STATUS_FMT_1																																0e<6>
RX_XCLK_SEL_0																																41<8>
RX_XCLK_SEL_1																																0e<7>
SATA_BURST_VAL_0																														41<7>	41<6>	41<5>
SATA_BURST_VAL_1																														0e<8>	0e<9>	0e<10>
SATA_IDLE_VAL_0																														41<4>	41<3>	41<2>
SATA_IDLE_VAL_1																														0e<11>	0e<12>	0e<13>
SATA_MAX_BURST_0																											41<1>	41<0>	40<15>	40<14>	40<13>	40<12>
SATA_MAX_BURST_1																											0e<14>	0e<15>	0f<0>	0f<1>	0f<2>	0f<3>
SATA_MAX_INIT_0																											40<11>	40<10>	40<9>	40<8>	40<7>	40<6>
SATA_MAX_INIT_1																											0f<4>	0f<5>	0f<6>	0f<7>	0f<8>	0f<9>

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SATA_MAX_WAKE_0																											40<5>	40<4>	40<3>	40<2>	40<1>	40<0>
SATA_MAX_WAKE_1																											0f<10>	0f<11>	0f<12>	0f<13>	0f<14>	0f<15>
SATA_MIN_BURST_0																											3f<15>	3f<14>	3f<13>	3f<12>	3f<11>	3f<10>
SATA_MIN_BURST_1																											10<0>	10<1>	10<2>	10<3>	10<4>	10<5>
SATA_MIN_INIT_0																											3f<9>	3f<8>	3f<7>	3f<6>	3f<5>	3f<4>
SATA_MIN_INIT_1																											10<6>	10<7>	10<8>	10<9>	10<10>	10<11>
SATA_MIN_WAKE_0																											3f<3>	3f<2>	3f<1>	3f<0>	3e<15>	3e<14>
SATA_MIN_WAKE_1																											10<12>	10<13>	10<14>	10<15>	11<0>	11<1>
TERMINATION_CTRL																												29<1>	29<2>	29<3>	29<4>	29<5>
TERMINATION_OVRD																																29<6>
TRANS_TIME_FROM_P2_0																	3e<12>	3e<11>	3e<10>	3e<9>	3e<8>	3e<7>	3e<6>	3e<5>	3e<4>	3e<3>	3e<2>	3e<1>	3e<0>	3d<15>	3d<14>	3d<13>
TRANS_TIME_FROM_P2_1																	11<3>	11<4>	11<5>	11<6>	11<7>	11<8>	11<9>	11<10>	11<11>	11<12>	11<13>	11<14>	11<15>	12<0>	12<1>	12<2>
TRANS_TIME_NON_P2_0																	3d<12>	3d<11>	3d<10>	3d<9>	3d<8>	3d<7>	3d<6>	3d<5>	3d<4>	3d<3>	3d<2>	3d<1>	3d<0>	3c<15>	3c<14>	3c<13>

Table D-2: **DRP Address by Attribute *(Continued)***

Attribute	Bit																														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
TRANS_TIME_NON_P2_1	13<2>	13<1>	13<0>	12<15>	12<14>	12<13>	12<12>	12<11>	12<10>	12<9>	12<8>	12<7>	12<6>	12<5>	12<4>	12<3>																
TRANS_TIME_TO_P2_0	3b<13>	3b<14>	3b<15>	3c<0>	3c<1>	3c<2>	3c<3>	3c<4>	3c<5>	3c<6>	3c<7>	3c<8>	3c<9>	3c<10>	3c<11>	3c<12>																
TRANS_TIME_TO_P2_1	14<2>	14<1>	14<0>	13<15>	13<14>	13<13>	13<12>	13<11>	13<10>	13<9>	13<8>	13<7>	13<6>	13<5>	13<4>	13<3>																
TX_BUFFER_USE_0	3b<12>																															
TX_BUFFER_USE_1	14<3>																															
TX_DIFF_BOOST_0	4a<13>																															
TX_DIFF_BOOST_1	5<2>																															
TX_SYNC_FILTERB	24<13>																															
TX_XCLK_SEL_0	3a<8>																															
TX_XCLK_SEL_1	15<7>																															

Optiver Ex1009
Page 321 of 347

DRP Address by Bit Location

This section lists the attributes according to DRP address and bit location:

- Table D-3, DRP Addresses 0x00 through 0x07
- Table D-4, DRP Addresses 0x08 through 0x0F
- Table D-5, DRP Addresses 0x10 through 0x17
- Table D-6, DRP Addresses 0x18 through 0x1F
- Table D-7, DRP Addresses 0x20 through 0x27
- Table D-8, DRP Addresses 0x28 through 0x2F
- Table D-9, DRP Addresses 0x30 through 0x37
- Table D-10, DRP Addresses 0x38 through 0x3F
- Table D-11, DRP Addresses 0x40 through 0x47
- Table D-12, DRP Addresses 0x48 through 0x4F

Cpuver Ex1009
Page 322 of 347

Table D-3: **DRP Addresses 0x00 through 0x07**

Bit	Address							
	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	RCV_TERM_MID_1	PMA_RX_CFG_1[13]
1	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	AC_CAP_DIS_1	Do Not Modify
2	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	TX_DIFF_BOOST_1	PMA_RX_CFG_1[12]	PMA_RX_CFG_1[5]
3	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	CLKINDC_B	PLL_TXDIVSEL_OUT_1[1]	PMA_RX_CFG_1[24]	PMA_RX_CFG_1[4]
4	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	REFCLK_SEL[2]	PLL_TXDIVSEL_OUT_1[0]	PMA_RX_CFG_1[1]	PMA_RX_CFG_1[3]
5	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	REFCLK_SEL[1]	Do Not Modify	PMA_RX_CFG_1[0]	PMA_RX_CFG_1[2]
6	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	REFCLK_SEL[0]	Do Not Modify	PMA_RX_CFG_1[11]	PMA_RX_CFG_1[23]
7	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	CLKSOUTH_SEL	Do Not Modify	PMA_RX_CFG_1[22]	PMA_RX_CFG_1[10]
8	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	CLKNORTH_SEL	Do Not Modify	PMA_RX_CFG_1[21]	PMA_RX_CFG_1[9]
9	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	PLL_DIVSEL_REF[0]	Do Not Modify	PMA_RX_CFG_1[20]	PMA_RX_CFG_1[8]
10	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	PLL_DIVSEL_REF[1]	Do Not Modify	PMA_RX_CFG_1[19]	PMA_RX_CFG_1[7]
11	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	PLL_DIVSEL_REF[2]	Do Not Modify	PMA_RX_CFG_1[18]	PMA_RX_CFG_1[6]
12	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	PLL_DIVSEL_REF[3]	Do Not Modify	PMA_RX_CFG_1[17]	CHAN_BOND_SEQ_2_4_1[0]
13	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	PLL_DIVSEL_REF[4]	Do Not Modify	PMA_RX_CFG_1[16]	CHAN_BOND_SEQ_2_4_1[1]
14	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	PLL_DIVSEL_REF[5]	RCV_TERM_VTTRX_1	PMA_RX_CFG_1[15]	CHAN_BOND_SEQ_2_4_1[2]
15	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	RCV_TERM_GND_1	PMA_RX_CFG_1[14]	CHAN_BOND_SEQ_2_4_1[3]

Table D-4: **DRP Addresses 0x08 through 0x0F**

Bit	Address							
	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0	CHAN_BOND_SEQ_2_4_1[4]	PCI_EXPRESS_MODE_1	PLL_RXDIVSEL_OUT_1[0]	PMA_CDR_SCAN_1[11]	PRBS_ERR_THRESHOLD_1[27]	PRBS_ERR_THRESHOLD_1[11]	RX_LOS_INVALID_INCR_1[0]	SATA_MAX_BURST_1[3]
1	CHAN_BOND_SEQ_2_4_1[5]	PCOMMA_10B_VALUE_1[9]	PMA_CDR_SCAN_1[26]	PMA_CDR_SCAN_1[10]	PRBS_ERR_THRESHOLD_1[26]	PRBS_ERR_THRESHOLD_1[10]	RX_LOSS_OF_SYNC_FSM_1	SATA_MAX_BURST_1[2]
2	CHAN_BOND_SEQ_2_4_1[6]	PCOMMA_10B_VALUE_1[8]	PMA_CDR_SCAN_1[25]	PMA_CDR_SCAN_1[9]	PRBS_ERR_THRESHOLD_1[25]	PRBS_ERR_THRESHOLD_1[9]	RX_LOS_THRESHOLD_1[2]	SATA_MAX_BURST_1[1]
3	CHAN_BOND_SEQ_2_4_1[7]	PCOMMA_10B_VALUE_1[7]	PMA_CDR_SCAN_1[24]	PMA_CDR_SCAN_1[8]	PRBS_ERR_THRESHOLD_1[24]	PRBS_ERR_THRESHOLD_1[8]	RX_LOS_THRESHOLD_1[1]	SATA_MAX_BURST_1[0]
4	CHAN_BOND_SEQ_2_4_1[8]	PCOMMA_10B_VALUE_1[6]	PMA_CDR_SCAN_1[23]	PMA_CDR_SCAN_1[7]	PRBS_ERR_THRESHOLD_1[23]	PRBS_ERR_THRESHOLD_1[7]	RX_LOS_THRESHOLD_1[0]	SATA_MAX_INIT_1[5]
5	CHAN_BOND_SEQ_2_4_1[9]	PCOMMA_10B_VALUE_1[5]	PMA_CDR_SCAN_1[22]	PMA_CDR_SCAN_1[6]	PRBS_ERR_THRESHOLD_1[22]	PRBS_ERR_THRESHOLD_1[6]	RX_SLIDE_MODE_1	SATA_MAX_INIT_1[4]
6	CHAN_BOND_SEQ_2_3_1[0]	PCOMMA_10B_VALUE_1[4]	PMA_CDR_SCAN_1[21]	PMA_CDR_SCAN_1[5]	PRBS_ERR_THRESHOLD_1[21]	PRBS_ERR_THRESHOLD_1[5]	RX_STATUS_FMT_1	SATA_MAX_INIT_1[3]
7	CHAN_BOND_SEQ_2_3_1[1]	PCOMMA_10B_VALUE_1[3]	PMA_CDR_SCAN_1[20]	PMA_CDR_SCAN_1[4]	PRBS_ERR_THRESHOLD_1[20]	PRBS_ERR_THRESHOLD_1[4]	RX_XCLK_SEL_1	SATA_MAX_INIT_1[2]
8	CHAN_BOND_SEQ_2_3_1[2]	PCOMMA_10B_VALUE_1[2]	PMA_CDR_SCAN_1[19]	PMA_CDR_SCAN_1[3]	PRBS_ERR_THRESHOLD_1[19]	PRBS_ERR_THRESHOLD_1[3]	SATA_BURST_VAL_1[2]	SATA_MAX_INIT_1[1]
9	CHAN_BOND_SEQ_2_3_1[3]	PCOMMA_10B_VALUE_1[1]	PMA_CDR_SCAN_1[18]	PMA_CDR_SCAN_1[2]	PRBS_ERR_THRESHOLD_1[18]	PRBS_ERR_THRESHOLD_1[2]	SATA_BURST_VAL_1[1]	SATA_MAX_INIT_1[0]
10	CHAN_BOND_SEQ_2_3_1[4]	PCOMMA_10B_VALUE_1[0]	PMA_CDR_SCAN_1[17]	PMA_CDR_SCAN_1[1]	PRBS_ERR_THRESHOLD_1[17]	PRBS_ERR_THRESHOLD_1[1]	SATA_BURST_VAL_1[0]	SATA_MAX_WAKE_1[5]
11	CHAN_BOND_SEQ_2_3_1[5]	PCOMMA_DETECT_1	PMA_CDR_SCAN_1[16]	PMA_CDR_SCAN_1[0]	PRBS_ERR_THRESHOLD_1[16]	PRBS_ERR_THRESHOLD_1[0]	SATA_IDLE_VAL_1[2]	SATA_MAX_WAKE_1[4]
12	CHAN_BOND_SEQ_2_3_1[6]	Do Not Modify	PMA_CDR_SCAN_1[15]	PRBS_ERR_THRESHOLD_1[31]	PRBS_ERR_THRESHOLD_1[15]	RX_BUFFER_USE_1	SATA_IDLE_VAL_1[1]	SATA_MAX_WAKE_1[3]
13	CHAN_BOND_SEQ_2_3_1[7]	Do Not Modify	PMA_CDR_SCAN_1[14]	PRBS_ERR_THRESHOLD_1[30]	PRBS_ERR_THRESHOLD_1[14]	RX_DECODE_SEQ_MATCH_1	SATA_IDLE_VAL_1[0]	SATA_MAX_WAKE_1[2]
14	CHAN_BOND_SEQ_2_3_1[8]	PLL_SATA_1	PMA_CDR_SCAN_1[13]	PRBS_ERR_THRESHOLD_1[29]	PRBS_ERR_THRESHOLD_1[13]	RX_LOS_INVALID_INCR_1[2]	SATA_MAX_BURST_1[5]	SATA_MAX_WAKE_1[1]
15	Do Not Modify	PLL_RXDIVSEL_OUT_1[1]	PMA_CDR_SCAN_1[12]	PRBS_ERR_THRESHOLD_1[28]	PRBS_ERR_THRESHOLD_1[12]	RX_LOS_INVALID_INCR_1[1]	SATA_MAX_BURST_1[4]	SATA_MAX_WAKE_1[0]

Apple v. Ericsson Ex1009
Page 324 of 347

Table D-5: **DRP Addresses 0x10 through 0x17**

Bit	Address							
	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
0	SATA_MIN_BURST_1[5]	SATA_MIN_WAKE_1[1]	TRANS_TIME_FROM_P2_1[2]	TRANS_TIME_NON_P2_1[2]	TRANS_TIME_TO_P2_1[2]	Do Not Modify	CHAN_BOND_SEQ_2_ENABLE_1[2]	CLK_COR_MAX_LAT_1[0]
1	SATA_MIN_BURST_1[4]	SATA_MIN_WAKE_1[0]	TRANS_TIME_FROM_P2_1[1]	TRANS_TIME_NON_P2_1[1]	TRANS_TIME_TO_P2_1[1]	Do Not Modify	CHAN_BOND_SEQ_2_ENABLE_1[1]	CLK_COR_MIN_LAT_1[5]
2	SATA_MIN_BURST_1[3]	Do Not Modify	TRANS_TIME_FROM_P2_1[0]	TRANS_TIME_NON_P2_1[0]	TRANS_TIME_TO_P2_1[0]	Do Not Modify	CHAN_BOND_SEQ_2_USE_1	CLK_COR_MIN_LAT_1[4]
3	SATA_MIN_BURST_1[2]	TRANS_TIME_FROM_P2_1[15]	TRANS_TIME_NON_P2_1[15]	TRANS_TIME_TO_P2_1[15]	TX_BUFFER_USE_1	Do Not Modify	CHAN_BOND_SEQ_LEN_1[1]	CLK_COR_MIN_LAT_1[3]
4	SATA_MIN_BURST_1[1]	TRANS_TIME_FROM_P2_1[14]	TRANS_TIME_NON_P2_1[14]	TRANS_TIME_TO_P2_1[14]	Do Not Modify	Do Not Modify	CHAN_BOND_SEQ_LEN_1[0]	CLK_COR_MIN_LAT_1[2]
5	SATA_MIN_BURST_1[0]	TRANS_TIME_FROM_P2_1[13]	TRANS_TIME_NON_P2_1[13]	TRANS_TIME_TO_P2_1[13]	Do Not Modify	Do Not Modify	CLK_COR_ADJ_LEN_1[1]	CLK_COR_MIN_LAT_1[1]
6	SATA_MIN_INIT_1[5]	TRANS_TIME_FROM_P2_1[12]	TRANS_TIME_NON_P2_1[12]	TRANS_TIME_TO_P2_1[12]	Do Not Modify	Do Not Modify	CLK_COR_ADJ_LEN_1[0]	CLK_COR_MIN_LAT_1[0]
7	SATA_MIN_INIT_1[4]	TRANS_TIME_FROM_P2_1[11]	TRANS_TIME_NON_P2_1[11]	TRANS_TIME_TO_P2_1[11]	Do Not Modify	TX_XCLK_SEL_1	CLK_COR_DET_LEN_1[1]	CLK_COR_PRECEDENCE_1
8	SATA_MIN_INIT_1[3]	TRANS_TIME_FROM_P2_1[10]	TRANS_TIME_NON_P2_1[10]	TRANS_TIME_TO_P2_1[10]	Do Not Modify	Do Not Modify	CLK_COR_DET_LEN_1[0]	CLK_CORRECT_USE_1
9	SATA_MIN_INIT_1[2]	TRANS_TIME_FROM_P2_1[9]	TRANS_TIME_NON_P2_1[9]	TRANS_TIME_TO_P2_1[9]	Do Not Modify	Do Not Modify	CLK_COR_INSERT_IDLE_FLAG_1	CLK_COR_REPEAT_WAIT_1[4]
10	SATA_MIN_INIT_1[1]	TRANS_TIME_FROM_P2_1[8]	TRANS_TIME_NON_P2_1[8]	TRANS_TIME_TO_P2_1[8]	Do Not Modify	OOBDETECT_THRESHOLD_1[2]	CLK_COR_KEEP_IDLE_1	CLK_COR_REPEAT_WAIT_1[3]
11	SATA_MIN_INIT_1[0]	TRANS_TIME_FROM_P2_1[7]	TRANS_TIME_NON_P2_1[7]	TRANS_TIME_TO_P2_1[7]	Do Not Modify	OOBDETECT_THRESHOLD_1[1]	CLK_COR_MAX_LAT_1[5]	CLK_COR_REPEAT_WAIT_1[2]
12	SATA_MIN_WAKE_1[5]	TRANS_TIME_FROM_P2_1[6]	TRANS_TIME_NON_P2_1[6]	TRANS_TIME_TO_P2_1[6]	Do Not Modify	OOBDETECT_THRESHOLD_1[0]	CLK_COR_MAX_LAT_1[4]	CLK_COR_REPEAT_WAIT_1[1]
13	SATA_MIN_WAKE_1[4]	TRANS_TIME_FROM_P2_1[5]	TRANS_TIME_NON_P2_1[5]	TRANS_TIME_TO_P2_1[5]	Do Not Modify	Do Not Modify	CLK_COR_MAX_LAT_1[3]	CLK_COR_REPEAT_WAIT_1[0]
14	SATA_MIN_WAKE_1[3]	TRANS_TIME_FROM_P2_1[4]	TRANS_TIME_NON_P2_1[4]	TRANS_TIME_TO_P2_1[4]	Do Not Modify	CHAN_BOND_SEQ_2_ENABLE_1[4]	CLK_COR_MAX_LAT_1[2]	CLK_COR_SEQ_1_1_1[9]
15	SATA_MIN_WAKE_1[2]	TRANS_TIME_FROM_P2_1[3]	TRANS_TIME_NON_P2_1[3]	TRANS_TIME_TO_P2_1[3]	Do Not Modify	CHAN_BOND_SEQ_2_ENABLE_1[3]	CLK_COR_MAX_LAT_1[1]	CLK_COR_SEQ_1_1_1[8]

Table D-6: **DRP Addresses 0x18 through 0x1F**

Bit	Address							
	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
0	CLK_COR_SEQ_1_1_1[7]	CLK_COR_SEQ_1_2_1[1]	CLK_COR_SEQ_1_4_1[5]	CLK_COR_SEQ_2_1_1[3]	CLK_COR_SEQ_2_3_1[7]	CLK_COR_SEQ_2_4_1[1]	COMMA_10B_ENABLE_1[4]	MCOMMA_10B_VALUE_1[2]
1	CLK_COR_SEQ_1_1_1[6]	CLK_COR_SEQ_1_2_1[0]	CLK_COR_SEQ_1_4_1[4]	CLK_COR_SEQ_2_1_1[2]	CLK_COR_SEQ_2_3_1[6]	CLK_COR_SEQ_2_4_1[0]	COMMA_10B_ENABLE_1[3]	MCOMMA_10B_VALUE_1[1]
2	CLK_COR_SEQ_1_1_1[5]	CLK_COR_SEQ_1_3_1[9]	CLK_COR_SEQ_1_4_1[3]	CLK_COR_SEQ_2_1_1[1]	CLK_COR_SEQ_2_3_1[5]	CLK_COR_SEQ_2_ENABLE_1[4]	COMMA_10B_ENABLE_1[2]	MCOMMA_10B_VALUE_1[0]
3	CLK_COR_SEQ_1_1_1[4]	CLK_COR_SEQ_1_3_1[8]	CLK_COR_SEQ_1_4_1[2]	CLK_COR_SEQ_2_1_1[0]	CLK_COR_SEQ_2_3_1[4]	CLK_COR_SEQ_2_ENABLE_1[3]	COMMA_10B_ENABLE_1[1]	MCOMMA_DETECT_1
4	CLK_COR_SEQ_1_1_1[3]	CLK_COR_SEQ_1_3_1[7]	CLK_COR_SEQ_1_4_1[1]	CLK_COR_SEQ_2_2_1[9]	CLK_COR_SEQ_2_3_1[3]	CLK_COR_SEQ_2_ENABLE_1[2]	COMMA_10B_ENABLE_1[0]	CHAN_BOND_SEQ_2_3_1[9]
5	CLK_COR_SEQ_1_1_1[2]	CLK_COR_SEQ_1_3_1[6]	CLK_COR_SEQ_1_4_1[0]	CLK_COR_SEQ_2_2_1[8]	CLK_COR_SEQ_2_3_1[2]	CLK_COR_SEQ_2_ENABLE_1[1]	COMMA_DOUBLE_1	CHAN_BOND_SEQ_2_2_1[0]
6	CLK_COR_SEQ_1_1_1[1]	CLK_COR_SEQ_1_3_1[5]	CLK_COR_SEQ_1_ENABLE_1[4]	CLK_COR_SEQ_2_2_1[7]	CLK_COR_SEQ_2_3_1[1]	CLK_COR_SEQ_2_USE_1	DEC_MCOMMA_DETECT_1	CHAN_BOND_SEQ_2_2_1[1]
7	CLK_COR_SEQ_1_1_1[0]	CLK_COR_SEQ_1_3_1[4]	CLK_COR_SEQ_1_ENABLE_1[3]	CLK_COR_SEQ_2_2_1[6]	CLK_COR_SEQ_2_3_1[0]	COM_BURST_VAL_1[3]	DEC_PCOMMA_DETECT_1	CHAN_BOND_SEQ_2_2_1[2]
8	CLK_COR_SEQ_1_2_1[9]	CLK_COR_SEQ_1_3_1[3]	CLK_COR_SEQ_1_ENABLE_1[2]	CLK_COR_SEQ_2_2_1[5]	CLK_COR_SEQ_2_4_1[9]	COM_BURST_VAL_1[2]	DEC_VALID_COMMA_ONLY_1	CHAN_BOND_SEQ_2_2_1[3]
9	CLK_COR_SEQ_1_2_1[8]	CLK_COR_SEQ_1_3_1[2]	CLK_COR_SEQ_1_ENABLE_1[1]	CLK_COR_SEQ_2_2_1[4]	CLK_COR_SEQ_2_4_1[8]	COM_BURST_VAL_1[1]	MCOMMA_10B_VALUE_1[9]	CHAN_BOND_SEQ_2_2_1[4]
10	CLK_COR_SEQ_1_2_1[7]	CLK_COR_SEQ_1_3_1[1]	CLK_COR_SEQ_2_1_1[9]	CLK_COR_SEQ_2_2_1[3]	CLK_COR_SEQ_2_4_1[7]	COM_BURST_VAL_1[0]	MCOMMA_10B_VALUE_1[8]	CHAN_BOND_SEQ_2_2_1[5]
11	CLK_COR_SEQ_1_2_1[6]	CLK_COR_SEQ_1_3_1[0]	CLK_COR_SEQ_2_1_1[8]	CLK_COR_SEQ_2_2_1[2]	CLK_COR_SEQ_2_4_1[6]	COMMA_10B_ENABLE_1[9]	MCOMMA_10B_VALUE_1[7]	CHAN_BOND_SEQ_2_2_1[6]
12	CLK_COR_SEQ_1_2_1[5]	CLK_COR_SEQ_1_4_1[9]	CLK_COR_SEQ_2_1_1[7]	CLK_COR_SEQ_2_2_1[1]	CLK_COR_SEQ_2_4_1[5]	COMMA_10B_ENABLE_1[8]	MCOMMA_10B_VALUE_1[6]	CHAN_BOND_SEQ_2_2_1[7]
13	CLK_COR_SEQ_1_2_1[4]	CLK_COR_SEQ_1_4_1[8]	CLK_COR_SEQ_2_1_1[6]	CLK_COR_SEQ_2_2_1[0]	CLK_COR_SEQ_2_4_1[4]	COMMA_10B_ENABLE_1[7]	MCOMMA_10B_VALUE_1[5]	Do Not Modify
14	CLK_COR_SEQ_1_2_1[3]	CLK_COR_SEQ_1_4_1[7]	CLK_COR_SEQ_2_1_1[5]	CLK_COR_SEQ_2_3_1[9]	CLK_COR_SEQ_2_4_1[3]	COMMA_10B_ENABLE_1[6]	MCOMMA_10B_VALUE_1[4]	CHAN_BOND_SEQ_2_2_1[8]
15	CLK_COR_SEQ_1_2_1[2]	CLK_COR_SEQ_1_4_1[6]	CLK_COR_SEQ_2_1_1[4]	CLK_COR_SEQ_2_3_1[8]	CLK_COR_SEQ_2_4_1[2]	COMMA_10B_ENABLE_1[5]	MCOMMA_10B_VALUE_1[3]	CHAN_BOND_SEQ_2_2_1[9]

Apotex Ex1009
Page 326 of 347

Table D-7: **DRP Addresses 0x20 through 0x27**

Bit	Address							
	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
0	CHAN_ BOND_SEQ_ 2_1_1[0]	CHAN_ BOND_SEQ_ 1_4_1[2]	CHAN_ BOND_SEQ_ 1_3_1[8]	CHAN_ BOND_SEQ_ 1_1_1[4]	CHAN_ BOND_1_ MAX_SKEW _1[1]	Do Not Modify	Do Not Modify	PCS_COM_ CFG[27]
1	CHAN_ BOND_SEQ_ 2_1_1[1]	CHAN_ BOND_SEQ_ 1_4_1[3]	CHAN_ BOND_SEQ_ 1_3_1[9]	CHAN_ BOND_SEQ_ 1_1_1[5]	CHAN_ BOND_1_ MAX_SKEW _1[2]	Do Not Modify	Do Not Modify	PCS_COM_ CFG[26]
2	CHAN_ BOND_SEQ_ 2_1_1[2]	CHAN_ BOND_SEQ_ 1_4_1[4]	CHAN_ BOND_SEQ_ 1_2_1[0]	CHAN_ BOND_SEQ_ 1_1_1[6]	CHAN_ BOND_1_ MAX_SKEW _1[3]	Do Not Modify	Do Not Modify	PCS_COM_ CFG[25]
3	CHAN_ BOND_SEQ_ 2_1_1[3]	CHAN_ BOND_SEQ_ 1_4_1[5]	CHAN_ BOND_SEQ_ 1_2_1[1]	CHAN_ BOND_SEQ_ 1_1_1[7]	ALIGN_ COMMA_ WORD_1	Do Not Modify	Do Not Modify	PCS_COM_ CFG[24]
4	CHAN_ BOND_SEQ_ 2_1_1[4]	CHAN_ BOND_SEQ_ 1_4_1[6]	CHAN_ BOND_SEQ_ 1_2_1[2]	CHAN_ BOND_SEQ_ 1_1_1[8]	Do Not Modify	Do Not Modify	Do Not Modify	PCS_COM_ CFG[23]
5	CHAN_ BOND_SEQ_ 2_1_1[5]	CHAN_ BOND_SEQ_ 1_4_1[7]	CHAN_ BOND_SEQ_ 1_2_1[3]	CHAN_ BOND_SEQ_ 1_1_1[9]	Do Not Modify	Do Not Modify	Do Not Modify	PCS_COM_ CFG[22]
6	CHAN_ BOND_SEQ_ 2_1_1[6]	CHAN_ BOND_SEQ_ 1_4_1[8]	CHAN_ BOND_SEQ_ 1_2_1[4]	CHAN_ BOND_ MODE_1[0]	Do Not Modify	Do Not Modify	Do Not Modify	PCS_COM_ CFG[21]
7	CHAN_ BOND_SEQ_ 2_1_1[7]	CHAN_ BOND_SEQ_ 1_4_1[9]	CHAN_ BOND_SEQ_ 1_2_1[5]	CHAN_ BOND_ MODE_1[1]	Do Not Modify	Do Not Modify	Do Not Modify	PCS_COM_ CFG[20]
8	CHAN_ BOND_SEQ_ 2_1_1[8]	CHAN_ BOND_SEQ_ 1_3_1[0]	CHAN_ BOND_SEQ_ 1_2_1[6]	CHAN_ BOND_ LEVEL_1[0]	Do Not Modify	Do Not Modify	Do Not Modify	PCS_COM_ CFG[19]
9	CHAN_ BOND_SEQ_ 2_1_1[9]	CHAN_ BOND_SEQ_ 1_3_1[1]	CHAN_ BOND_SEQ_ 1_2_1[7]	CHAN_ BOND_ LEVEL_1[1]	Do Not Modify	Do Not Modify	CLK25_ DIVIDER[2]	PCS_COM_ CFG[18]
10	CHAN_ BOND_SEQ_ 1_ENABLE_ 1[1]	CHAN_ BOND_SEQ_ 1_3_1[2]	CHAN_ BOND_SEQ_ 1_2_1[8]	CHAN_ BOND_ LEVEL_1[2]	Do Not Modify	Do Not Modify	CLK25_ DIVIDER[1]	PCS_COM_ CFG[17]
11	CHAN_ BOND_SEQ_ 1_ENABLE_ 1[2]	CHAN_ BOND_SEQ_ 1_3_1[3]	CHAN_ BOND_SEQ_ 1_2_1[9]	CHAN_ BOND_2_ MAX_SKEW _1[0]	Do Not Modify	Do Not Modify	CLK25_ DIVIDER[0]	PCS_COM_ CFG[16]
12	CHAN_ BOND_SEQ_ 1_ENABLE_ 1[3]	CHAN_ BOND_SEQ_ 1_3_1[4]	CHAN_ BOND_SEQ_ 1_1_1[0]	CHAN_ BOND_2_ MAX_SKEW _1[1]	Do Not Modify	Do Not Modify	OOB_CLK_ DIVIDER[2]	PCS_COM_ CFG[15]
13	CHAN_ BOND_SEQ_ 1_ENABLE_ 1[4]	CHAN_ BOND_SEQ_ 1_3_1[5]	CHAN_ BOND_SEQ_ 1_1_1[1]	CHAN_ BOND_2_ MAX_SKEW _1[2]	TX_SYNC_ FILTERB	Do Not Modify	OOB_CLK_ DIVIDER[1]	PCS_COM_ CFG[14]

Table D-7: **DRP Addresses 0x20 through 0x27 *(Continued)***

Bit	Address							
	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
14	CHAN_ BOND_SEQ_ 1_4_1[0]	CHAN_ BOND_SEQ_ 1_3_1[6]	CHAN_ BOND_SEQ_ 1_1_1[2]	CHAN_ BOND_2_ MAX_SKEW _1[3]	Do Not Modify	Do Not Modify	OOB_CLK_ DIVIDER[0]	PCS_COM_ CFG[13]
15	CHAN_ BOND_SEQ_ 1_4_1[1]	CHAN_ BOND_SEQ_ 1_3_1[7]	CHAN_ BOND_SEQ_ 1_1_1[3]	CHAN_ BOND_1_ MAX_SKEW _1[0]	Do Not Modify	Do Not Modify	OVERSAMP LE_MODE	PCS_COM_ CFG[12]

Table D-8: **DRP Addresses 0x28 through 0x2F**

Bit	Address							
	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
0	PCS_COM_ CFG[11]	PLL_DIVSEL_ FB[0]	Do Not Modify	Do Not Modify	CHAN_ BOND_1_ MAX_SKEW_ 0[0]	CHAN_ BOND_SEQ_ 1_1_0[3]	CHAN_ BOND_SEQ_ 1_3_0[7]	CHAN_ BOND_SEQ_ 1_4_0[1]
1	PCS_COM_ CFG[10]	TERMINATIO N_CTRL[4]	Do Not Modify	Do Not Modify	CHAN_ BOND_2_ MAX_SKEW_ 0[3]	CHAN_ BOND_SEQ_ 1_1_0[2]	CHAN_ BOND_SEQ_ 1_3_0[6]	CHAN_ BOND_SEQ_ 1_4_0[0]
2	PCS_COM_ CFG[9]	TERMINATIO N_CTRL[3]	Do Not Modify	Do Not Modify	CHAN_ BOND_2_ MAX_SKEW_ 0[2]	CHAN_ BOND_SEQ_ 1_1_0[1]	CHAN_ BOND_SEQ_ 1_3_0[5]	CHAN_ BOND_SEQ_ 1_ENABLE_0 [4]
3	PCS_COM_ CFG[8]	TERMINATIO N_CTRL[2]	Do Not Modify	Do Not Modify	CHAN_ BOND_2_ MAX_SKEW_ 0[1]	CHAN_ BOND_SEQ_ 1_1_0[0]	CHAN_ BOND_SEQ_ 1_3_0[4]	CHAN_ BOND_SEQ_ 1_ENABLE_0 [3]
4	PCS_COM_ CFG[7]	TERMINATIO N_CTRL[1]	Do Not Modify	Do Not Modify	CHAN_ BOND_2_ MAX_SKEW_ 0[0]	CHAN_ BOND_SEQ_ 1_2_0[9]	CHAN_ BOND_SEQ_ 1_3_0[3]	CHAN_ BOND_SEQ_ 1_ENABLE_0 [2]
5	PCS_COM_ CFG[6]	TERMINATIO N_CTRL[0]	Do Not Modify	Do Not Modify	CHAN_ BOND_ LEVEL_0[2]	CHAN_ BOND_SEQ_ 1_2_0[8]	CHAN_ BOND_SEQ_ 1_3_0[2]	CHAN_ BOND_SEQ_ 1_ENABLE_0 [1]
6	PCS_COM_ CFG[5]	TERMINATIO N_OVRD	Do Not Modify	Do Not Modify	CHAN_ BOND_ LEVEL_0[1]	CHAN_ BOND_SEQ_ 1_2_0[7]	CHAN_ BOND_SEQ_ 1_3_0[1]	CHAN_ BOND_SEQ_ 2_1_0[9]
7	PCS_COM_ CFG[4]	Do Not Modify	Do Not Modify	Do Not Modify	CHAN_ BOND_ LEVEL_0[0]	CHAN_ BOND_SEQ_ 1_2_0[6]	CHAN_ BOND_SEQ_ 1_3_0[0]	CHAN_ BOND_SEQ_ 2_1_0[8]
8	PCS_COM_ CFG[3]	Do Not Modify	Do Not Modify	Do Not Modify	CHAN_ BOND_ MODE_0[1]	CHAN_ BOND_SEQ_ 1_2_0[5]	CHAN_ BOND_SEQ_ 1_4_0[9]	CHAN_ BOND_SEQ_ 2_1_0[7]
9	PCS_COM_ CFG[2]	Do Not Modify	Do Not Modify	Do Not Modify	CHAN_ BOND_ MODE_0[0]	CHAN_ BOND_SEQ_ 1_2_0[4]	CHAN_ BOND_SEQ_ 1_4_0[8]	CHAN_ BOND_SEQ_ 2_1_0[6]

Apotex Ex1009
Page 328 of 347

Table D-8: **DRP Addresses 0x28 through 0x2F *(Continued)***

Bit	Address							
	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
10	PCS_COM_CFG[1]	Do Not Modify	Do Not Modify	Do Not Modify	CHAN_BOND_SEQ_1_1_0[9]	CHAN_BOND_SEQ_1_2_0[3]	CHAN_BOND_SEQ_1_4_0[7]	CHAN_BOND_SEQ_2_1_0[5]
11	PCS_COM_CFG[0]	Do Not Modify	Do Not Modify	Do Not Modify	CHAN_BOND_SEQ_1_1_0[8]	CHAN_BOND_SEQ_1_2_0[2]	CHAN_BOND_SEQ_1_4_0[6]	CHAN_BOND_SEQ_2_1_0[4]
12	PLL_DIVSEL_FB[4]	Do Not Modify	Do Not Modify	ALIGN_COMMA_WORD_0	CHAN_BOND_SEQ_1_1_0[7]	CHAN_BOND_SEQ_1_2_0[1]	CHAN_BOND_SEQ_1_4_0[5]	CHAN_BOND_SEQ_2_1_0[3]
13	PLL_DIVSEL_FB[3]	Do Not Modify	Do Not Modify	CHAN_BOND_1_MAX_SKEW_0[3]	CHAN_BOND_SEQ_1_1_0[6]	CHAN_BOND_SEQ_1_2_0[0]	CHAN_BOND_SEQ_1_4_0[4]	CHAN_BOND_SEQ_2_1_0[2]
14	PLL_DIVSEL_FB[2]	Do Not Modify	Do Not Modify	CHAN_BOND_1_MAX_SKEW_0[2]	CHAN_BOND_SEQ_1_1_0[5]	CHAN_BOND_SEQ_1_3_0[9]	CHAN_BOND_SEQ_1_4_0[3]	CHAN_BOND_SEQ_2_1_0[1]
15	PLL_DIVSEL_FB[1]	Do Not Modify	Do Not Modify	CHAN_BOND_1_MAX_SKEW_0[1]	CHAN_BOND_SEQ_1_1_0[4]	CHAN_BOND_SEQ_1_3_0[8]	CHAN_BOND_SEQ_1_4_0[2]	CHAN_BOND_SEQ_2_1_0[0]

Table D-9: **DRP Addresses 0x30 through 0x37**

Bit	Address							
	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
0	CHAN_BOND_SEQ_2_2_0[9]	MCOMMA_10B_VALUE_0[3]	COMMA_10B_ENABLE_0[5]	CLK_COR_SEQ_2_4_0[2]	CLK_COR_SEQ_2_3_0[8]	CLK_COR_SEQ_2_1_0[4]	CLK_COR_SEQ_1_4_0[6]	CLK_COR_SEQ_1_2_0[2]
1	CHAN_BOND_SEQ_2_2_0[8]	MCOMMA_10B_VALUE_0[4]	COMMA_10B_ENABLE_0[6]	CLK_COR_SEQ_2_4_0[3]	CLK_COR_SEQ_2_3_0[9]	CLK_COR_SEQ_2_1_0[5]	CLK_COR_SEQ_1_4_0[7]	CLK_COR_SEQ_1_2_0[3]
2	Do Not Modify	MCOMMA_10B_VALUE_0[5]	COMMA_10B_ENABLE_0[7]	CLK_COR_SEQ_2_4_0[4]	CLK_COR_SEQ_2_2_0[0]	CLK_COR_SEQ_2_1_0[6]	CLK_COR_SEQ_1_4_0[8]	CLK_COR_SEQ_1_2_0[4]
3	CHAN_BOND_SEQ_2_2_0[7]	MCOMMA_10B_VALUE_0[6]	COMMA_10B_ENABLE_0[8]	CLK_COR_SEQ_2_4_0[5]	CLK_COR_SEQ_2_2_0[1]	CLK_COR_SEQ_2_1_0[7]	CLK_COR_SEQ_1_4_0[9]	CLK_COR_SEQ_1_2_0[5]
4	CHAN_BOND_SEQ_2_2_0[6]	MCOMMA_10B_VALUE_0[7]	COMMA_10B_ENABLE_0[9]	CLK_COR_SEQ_2_4_0[6]	CLK_COR_SEQ_2_2_0[2]	CLK_COR_SEQ_2_1_0[8]	CLK_COR_SEQ_1_3_0[0]	CLK_COR_SEQ_1_2_0[6]
5	CHAN_BOND_SEQ_2_2_0[5]	MCOMMA_10B_VALUE_0[8]	COM_BURST_VAL_0[0]	CLK_COR_SEQ_2_4_0[7]	CLK_COR_SEQ_2_2_0[3]	CLK_COR_SEQ_2_1_0[9]	CLK_COR_SEQ_1_3_0[1]	CLK_COR_SEQ_1_2_0[7]
6	CHAN_BOND_SEQ_2_2_0[4]	MCOMMA_10B_VALUE_0[9]	COM_BURST_VAL_0[1]	CLK_COR_SEQ_2_4_0[8]	CLK_COR_SEQ_2_2_0[4]	CLK_COR_SEQ_1_ENABLE_0[1]	CLK_COR_SEQ_1_3_0[2]	CLK_COR_SEQ_1_2_0[8]

Optiver Ex1009
Page 329 of 347

Table D-9: **DRP Addresses 0x30 through 0x37 *(Continued)***

Bit	Address							
	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
7	CHAN_BOND_SEQ_2_2_0[3]	DEC_VALID_COMMA_ONLY_0	COM_BURST_VAL_0[2]	CLK_COR_SEQ_2_4_0[9]	CLK_COR_SEQ_2_2_0[5]	CLK_COR_SEQ_1_ENABLE_0[2]	CLK_COR_SEQ_1_3_0[3]	CLK_COR_SEQ_1_2_0[9]
8	CHAN_BOND_SEQ_2_2_0[2]	DEC_PCOMMA_DETECT_0	COM_BURST_VAL_0[3]	CLK_COR_SEQ_2_3_0[0]	CLK_COR_SEQ_2_2_0[6]	CLK_COR_SEQ_1_ENABLE_0[3]	CLK_COR_SEQ_1_3_0[4]	CLK_COR_SEQ_1_1_0[0]
9	CHAN_BOND_SEQ_2_2_0[1]	DEC_MCOMMA_DETECT_0	CLK_COR_SEQ_2_USE_0	CLK_COR_SEQ_2_3_0[1]	CLK_COR_SEQ_2_2_0[7]	CLK_COR_SEQ_1_ENABLE_0[4]	CLK_COR_SEQ_1_3_0[5]	CLK_COR_SEQ_1_1_0[1]
10	CHAN_BOND_SEQ_2_2_0[0]	COMMA_DOUBLE_0	CLK_COR_SEQ_2_ENABLE_0[1]	CLK_COR_SEQ_2_3_0[2]	CLK_COR_SEQ_2_2_0[8]	CLK_COR_SEQ_1_4_0[0]	CLK_COR_SEQ_1_3_0[6]	CLK_COR_SEQ_1_1_0[2]
11	CHAN_BOND_SEQ_2_3_0[9]	COMMA_10B_ENABLE_0[0]	CLK_COR_SEQ_2_ENABLE_0[2]	CLK_COR_SEQ_2_3_0[3]	CLK_COR_SEQ_2_2_0[9]	CLK_COR_SEQ_1_4_0[1]	CLK_COR_SEQ_1_3_0[7]	CLK_COR_SEQ_1_1_0[3]
12	MCOMMA_DETECT_0	COMMA_10B_ENABLE_0[1]	CLK_COR_SEQ_2_ENABLE_0[3]	CLK_COR_SEQ_2_3_0[4]	CLK_COR_SEQ_2_1_0[0]	CLK_COR_SEQ_1_4_0[2]	CLK_COR_SEQ_1_3_0[8]	CLK_COR_SEQ_1_1_0[4]
13	MCOMMA_10B_VALUE_0[0]	COMMA_10B_ENABLE_0[2]	CLK_COR_SEQ_2_ENABLE_0[4]	CLK_COR_SEQ_2_3_0[5]	CLK_COR_SEQ_2_1_0[1]	CLK_COR_SEQ_1_4_0[3]	CLK_COR_SEQ_1_3_0[9]	CLK_COR_SEQ_1_1_0[5]
14	MCOMMA_10B_VALUE_0[1]	COMMA_10B_ENABLE_0[3]	CLK_COR_SEQ_2_4_0[0]	CLK_COR_SEQ_2_3_0[6]	CLK_COR_SEQ_2_1_0[2]	CLK_COR_SEQ_1_4_0[4]	CLK_COR_SEQ_1_2_0[0]	CLK_COR_SEQ_1_1_0[6]
15	MCOMMA_10B_VALUE_0[2]	COMMA_10B_ENABLE_0[4]	CLK_COR_SEQ_2_4_0[1]	CLK_COR_SEQ_2_3_0[7]	CLK_COR_SEQ_2_1_0[3]	CLK_COR_SEQ_1_4_0[5]	CLK_COR_SEQ_1_2_0[1]	CLK_COR_SEQ_1_1_0[7]

Table D-10: **DRP Addresses 0x38 through 0x3F**

Bit	Address							
	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
0	CLK_COR_SEQ_1_1_0[8]	CLK_COR_MAX_LAT_0[1]	CHAN_BOND_SEQ_2_ENABLE_0[3]	Do Not Modify	TRANS_TIME_TO_P2_0[3]	TRANS_TIME_NON_P2_0[3]	TRANS_TIME_FROM_P2_0[3]	SATA_MIN_WAKE_0[2]
1	CLK_COR_SEQ_1_1_0[9]	CLK_COR_MAX_LAT_0[2]	CHAN_BOND_SEQ_2_ENABLE_0[4]	Do Not Modify	TRANS_TIME_TO_P2_0[4]	TRANS_TIME_NON_P2_0[4]	TRANS_TIME_FROM_P2_0[4]	SATA_MIN_WAKE_0[3]
2	CLK_COR_REPEAT_WAIT_0[0]	CLK_COR_MAX_LAT_0[3]	Do Not Modify	Do Not Modify	TRANS_TIME_TO_P2_0[5]	TRANS_TIME_NON_P2_0[5]	TRANS_TIME_FROM_P2_0[5]	SATA_MIN_WAKE_0[4]
3	CLK_COR_REPEAT_WAIT_0[1]	CLK_COR_MAX_LAT_0[4]	OOBDETECT_THRESHOLD_0[0]	Do Not Modify	TRANS_TIME_TO_P2_0[6]	TRANS_TIME_NON_P2_0[6]	TRANS_TIME_FROM_P2_0[6]	SATA_MIN_WAKE_0[5]

Apotex Ex1009
Page 330 of 347

Table D-10: **DRP Addresses 0x38 through 0x3F *(Continued)***

Bit	Address							
	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
4	CLK_COR_REPEAT_WAIT_0[2]	CLK_COR_MAX_LAT_0[5]	OOBDETECT_THRESHOLD_0[1]	Do Not Modify	TRANS_TIME_TO_P2_0[7]	TRANS_TIME_NON_P2_0[7]	TRANS_TIME_FROM_P2_0[7]	SATA_MIN_INIT_0[0]
5	CLK_COR_REPEAT_WAIT_0[3]	CLK_COR_KEEP_IDLE_0	OOBDETECT_THRESHOLD_0[2]	Do Not Modify	TRANS_TIME_TO_P2_0[8]	TRANS_TIME_NON_P2_0[8]	TRANS_TIME_FROM_P2_0[8]	SATA_MIN_INIT_0[1]
6	CLK_COR_REPEAT_WAIT_0[4]	CLK_COR_INSERT_IDLE_FLAG_0	Do Not Modify	Do Not Modify	TRANS_TIME_TO_P2_0[9]	TRANS_TIME_NON_P2_0[9]	TRANS_TIME_FROM_P2_0[9]	SATA_MIN_INIT_0[2]
7	CLK_CORRECT_USE_0	CLK_COR_DET_LEN_0[0]	Do Not Modify	Do Not Modify	TRANS_TIME_TO_P2_0[10]	TRANS_TIME_NON_P2_0[10]	TRANS_TIME_FROM_P2_0[10]	SATA_MIN_INIT_0[3]
8	CLK_COR_PRECEDENCE_0	CLK_COR_DET_LEN_0[1]	TX_XCLK_SEL_0	Do Not Modify	TRANS_TIME_TO_P2_0[11]	TRANS_TIME_NON_P2_0[11]	TRANS_TIME_FROM_P2_0[11]	SATA_MIN_INIT_0[4]
9	CLK_COR_MIN_LAT_0[0]	CLK_COR_ADJ_LEN_0[0]	Do Not Modify	Do Not Modify	TRANS_TIME_TO_P2_0[12]	TRANS_TIME_NON_P2_0[12]	TRANS_TIME_FROM_P2_0[12]	SATA_MIN_INIT_0[5]
10	CLK_COR_MIN_LAT_0[1]	CLK_COR_ADJ_LEN_0[1]	Do Not Modify	Do Not Modify	TRANS_TIME_TO_P2_0[13]	TRANS_TIME_NON_P2_0[13]	TRANS_TIME_FROM_P2_0[13]	SATA_MIN_BURST_0[0]
11	CLK_COR_MIN_LAT_0[2]	CHAN_BOND_SEQ_LEN_0[0]	Do Not Modify	Do Not Modify	TRANS_TIME_TO_P2_0[14]	TRANS_TIME_NON_P2_0[14]	TRANS_TIME_FROM_P2_0[14]	SATA_MIN_BURST_0[1]
12	CLK_COR_MIN_LAT_0[3]	CHAN_BOND_SEQ_LEN_0[1]	Do Not Modify	TX_BUFFER_USE_0	TRANS_TIME_TO_P2_0[15]	TRANS_TIME_NON_P2_0[15]	TRANS_TIME_FROM_P2_0[15]	SATA_MIN_BURST_0[2]
13	CLK_COR_MIN_LAT_0[4]	CHAN_BOND_SEQ_2_USE_0	Do Not Modify	TRANS_TIME_TO_P2_0[0]	TRANS_TIME_NON_P2_0[0]	TRANS_TIME_FROM_P2_0[0]	Do Not Modify	SATA_MIN_BURST_0[3]
14	CLK_COR_MIN_LAT_0[5]	CHAN_BOND_SEQ_2_ENABLE_0[1]	Do Not Modify	TRANS_TIME_TO_P2_0[1]	TRANS_TIME_NON_P2_0[1]	TRANS_TIME_FROM_P2_0[1]	SATA_MIN_WAKE_0[0]	SATA_MIN_BURST_0[4]
15	CLK_COR_MAX_LAT_0[0]	CHAN_BOND_SEQ_2_ENABLE_0[2]	Do Not Modify	TRANS_TIME_TO_P2_0[2]	TRANS_TIME_NON_P2_0[2]	TRANS_TIME_FROM_P2_0[2]	SATA_MIN_WAKE_0[1]	SATA_MIN_BURST_0[5]

Optiver Ex1009
Page 331 of 347

Table D-11: **DRP Addresses 0x40 through 0x47**

Bit	Address							
	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
0	SATA_MAX_WAKE_0[0]	SATA_MAX_BURST_0[4]	RX_LOS_INVALID_INCR_0[1]	PRBS_ERR_THRESHOLD_0[12]	PRBS_ERR_THRESHOLD_0[28]	PMA_CDR_SCAN_0[12]	PLL_TXDIVSEL_OUT_0[1]	Do Not Modify
1	SATA_MAX_WAKE_0[1]	SATA_MAX_BURST_0[5]	RX_LOS_INVALID_INCR_0[2]	PRBS_ERR_THRESHOLD_0[13]	PRBS_ERR_THRESHOLD_0[29]	PMA_CDR_SCAN_0[13]	PLL_SATA_0	CHAN_BOND_SEQ_2_3_0[8]
2	SATA_MAX_WAKE_0[2]	SATA_IDLE_VAL_0[0]	RX_DECODE_SEQ_MATCH_0	PRBS_ERR_THRESHOLD_0[14]	PRBS_ERR_THRESHOLD_0[30]	PMA_CDR_SCAN_0[14]	PLL_RXDIVSEL_OUT_0[0]	CHAN_BOND_SEQ_2_3_0[7]
3	SATA_MAX_WAKE_0[3]	SATA_IDLE_VAL_0[1]	RX_BUFFER_USE_0	PRBS_ERR_THRESHOLD_0[15]	PRBS_ERR_THRESHOLD_0[31]	PMA_CDR_SCAN_0[15]	PLL_RXDIVSEL_OUT_0[1]	CHAN_BOND_SEQ_2_3_0[6]
4	SATA_MAX_WAKE_0[4]	SATA_IDLE_VAL_0[2]	PRBS_ERR_THRESHOLD_0[0]	PRBS_ERR_THRESHOLD_0[16]	PMA_CDR_SCAN_0[0]	PMA_CDR_SCAN_0[16]	PCOMMA_DETECT_0	CHAN_BOND_SEQ_2_3_0[5]
5	SATA_MAX_WAKE_0[5]	SATA_BURST_VAL_0[0]	PRBS_ERR_THRESHOLD_0[1]	PRBS_ERR_THRESHOLD_0[17]	PMA_CDR_SCAN_0[1]	PMA_CDR_SCAN_0[17]	PCOMMA_10B_VALUE_0[0]	CHAN_BOND_SEQ_2_3_0[4]
6	SATA_MAX_INIT_0[0]	SATA_BURST_VAL_0[1]	PRBS_ERR_THRESHOLD_0[2]	PRBS_ERR_THRESHOLD_0[18]	PMA_CDR_SCAN_0[2]	PMA_CDR_SCAN_0[18]	PCOMMA_10B_VALUE_0[1]	CHAN_BOND_SEQ_2_3_0[3]
7	SATA_MAX_INIT_0[1]	SATA_BURST_VAL_0[2]	PRBS_ERR_THRESHOLD_0[3]	PRBS_ERR_THRESHOLD_0[19]	PMA_CDR_SCAN_0[3]	PMA_CDR_SCAN_0[19]	PCOMMA_10B_VALUE_0[2]	CHAN_BOND_SEQ_2_3_0[2]
8	SATA_MAX_INIT_0[2]	RX_XCLK_SEL_0	PRBS_ERR_THRESHOLD_0[4]	PRBS_ERR_THRESHOLD_0[20]	PMA_CDR_SCAN_0[4]	PMA_CDR_SCAN_0[20]	PCOMMA_10B_VALUE_0[3]	CHAN_BOND_SEQ_2_3_0[1]
9	SATA_MAX_INIT_0[3]	RX_STATUS_FMT_0	PRBS_ERR_THRESHOLD_0[5]	PRBS_ERR_THRESHOLD_0[21]	PMA_CDR_SCAN_0[5]	PMA_CDR_SCAN_0[21]	PCOMMA_10B_VALUE_0[4]	CHAN_BOND_SEQ_2_3_0[0]
10	SATA_MAX_INIT_0[4]	RX_SLIDE_MODE_0	PRBS_ERR_THRESHOLD_0[6]	PRBS_ERR_THRESHOLD_0[22]	PMA_CDR_SCAN_0[6]	PMA_CDR_SCAN_0[22]	PCOMMA_10B_VALUE_0[5]	CHAN_BOND_SEQ_2_4_0[9]
11	SATA_MAX_INIT_0[5]	RX_LOS_THRESHOLD_0[0]	PRBS_ERR_THRESHOLD_0[7]	PRBS_ERR_THRESHOLD_0[23]	PMA_CDR_SCAN_0[7]	PMA_CDR_SCAN_0[23]	PCOMMA_10B_VALUE_0[6]	CHAN_BOND_SEQ_2_4_0[8]
12	SATA_MAX_BURST_0[0]	RX_LOS_THRESHOLD_0[1]	PRBS_ERR_THRESHOLD_0[8]	PRBS_ERR_THRESHOLD_0[24]	PMA_CDR_SCAN_0[8]	PMA_CDR_SCAN_0[24]	PCOMMA_10B_VALUE_0[7]	CHAN_BOND_SEQ_2_4_0[7]
13	SATA_MAX_BURST_0[1]	RX_LOS_THRESHOLD_0[2]	PRBS_ERR_THRESHOLD_0[9]	PRBS_ERR_THRESHOLD_0[25]	PMA_CDR_SCAN_0[9]	PMA_CDR_SCAN_0[25]	PCOMMA_10B_VALUE_0[8]	CHAN_BOND_SEQ_2_4_0[6]
14	SATA_MAX_BURST_0[2]	RX_LOSS_OF_SYNC_FSM_0	PRBS_ERR_THRESHOLD_0[10]	PRBS_ERR_THRESHOLD_0[26]	PMA_CDR_SCAN_0[10]	PMA_CDR_SCAN_0[26]	PCOMMA_10B_VALUE_0[9]	CHAN_BOND_SEQ_2_4_0[5]
15	SATA_MAX_BURST_0[3]	RX_LOS_INVALID_INCR_0[0]	PRBS_ERR_THRESHOLD_0[11]	PRBS_ERR_THRESHOLD_0[27]	PMA_CDR_SCAN_0[11]	PLL_TXDIVSEL_OUT_0[0]	PCI_EXPRESS_MODE_0	CHAN_BOND_SEQ_2_4_0[4]

Power Ex1009
Page 332 of 347

Table D-12: **DRP Addresses 0x48 through 0x4F**

Bit	Address							
	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
0	CHAN_ BOND_SEQ_ 2_4_0[3]	PMA_RX_ CFG_0[14]	RCV_TERM_ GND_0	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
1	CHAN_ BOND_SEQ_ 2_4_0[2]	PMA_RX_ CFG_0[15]	RCV_TERM_ VTTRX_0	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
2	CHAN_ BOND_SEQ_ 2_4_0[1]	PMA_RX_ CFG_0[16]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
3	CHAN_ BOND_SEQ_ 2_4_0[0]	PMA_RX_ CFG_0[17]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
4	PMA_RX_ CFG_0[6]	PMA_RX_ CFG_0[18]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
5	PMA_RX_ CFG_0[7]	PMA_RX_ CFG_0[19]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
6	PMA_RX_ CFG_0[8]	PMA_RX_ CFG_0[20]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
7	PMA_RX_ CFG_0[9]	PMA_RX_ CFG_0[21]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
8	PMA_RX_ CFG_0[10]	PMA_RX_ CFG_0[22]	PLL_ TXDIVSEL_ COMM_OUT [1]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
9	PMA_RX_ CFG_0[23]	PMA_RX_ CFG_0[11]	PLL_ TXDIVSEL_ COMM_OUT [0]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
10	PMA_RX_ CFG_0[2]	PMA_RX_ CFG_0[0]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
11	PMA_RX_ CFG_0[3]	PMA_RX_ CFG_0[1]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
12	PMA_RX_ CFG_0[4]	PMA_RX_ CFG_0[24]	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
13	PMA_RX_ CFG_0[5]	PMA_RX_ CFG_0[12]	TX_DIFF_ BOOST_0	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
14	Do Not Modify	AC_CAP_ DIS_0	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify
15	PMA_RX_ CFG_0[13]	RCV_TERM_ MID_0	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify	Do Not Modify

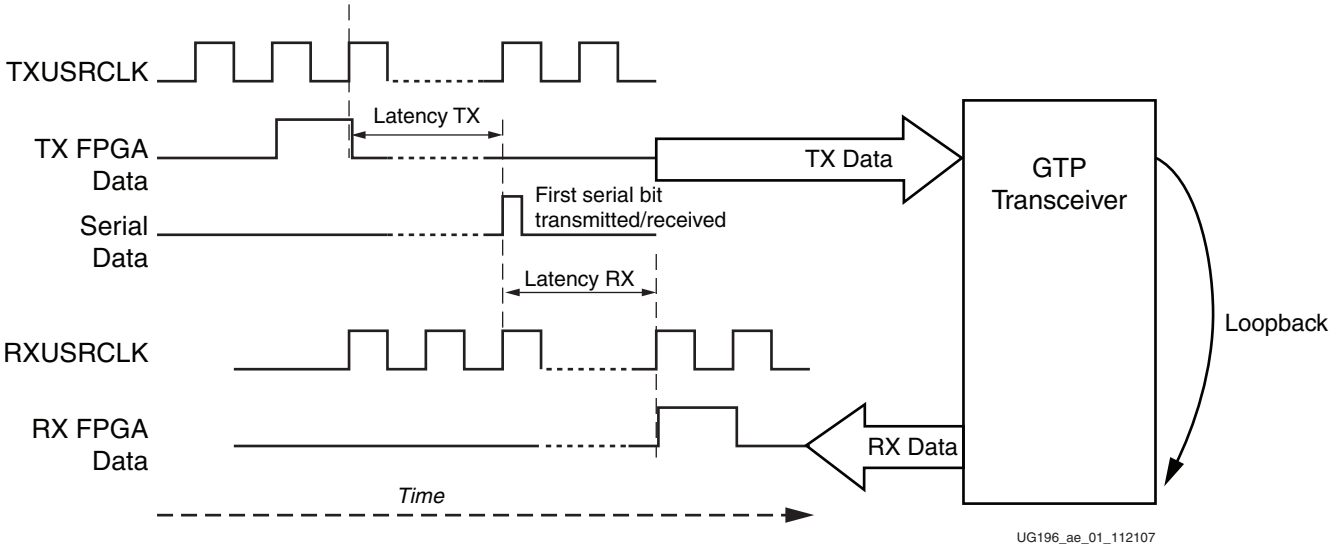
Optiver Ex1009
Page 333 of 347

Cover Ex1009
Page 334 of 347

Appendix E

Low Latency Design

This appendix illustrates the latency of the different functional blocks inside the TX and the RX sections of the GTP transceiver. Figure E-1 shows a pictorial definition of the TX and RX latencies.



Notes:

1. Latency is measured between rising edges of USRCLKs and serial data.

Figure E-1: **Latency Definition**

Each functional block has a latency defined as the time difference between the inputs and the outputs of the specific block. Some blocks in the GTP transceiver can be bypassed, reducing the latency of the datapath through the transmitter or the receiver. The latency of the blocks is deterministic with the exception of the RX elastic buffer (64-element FIFO) and the TX buffer (4-element FIFO). Bypassing buffers requires marginal conditions to be met, for example, phase alignment procedures or USRCLK requirements.

Refer to TX Buffering, Phase Alignment, and TX Skew Reduction, page 115, Configurable RX Elastic Buffer and Phase Alignment, page 176, Connecting TXUSRCLK and TXUSRCLK2, page 106, and Connecting RXUSRCLK and RXUSRCLK2, page 202 for the implications and marginal conditions on bypassing buffers.

GTP TX Latency

Figure E-2 shows a detailed block diagram of the GTP TX. Refer to Chapter 6, GTP Transmitter (TX), and Figure 6-1, page 103 for more details on this figure and the GTP transmitter blocks.

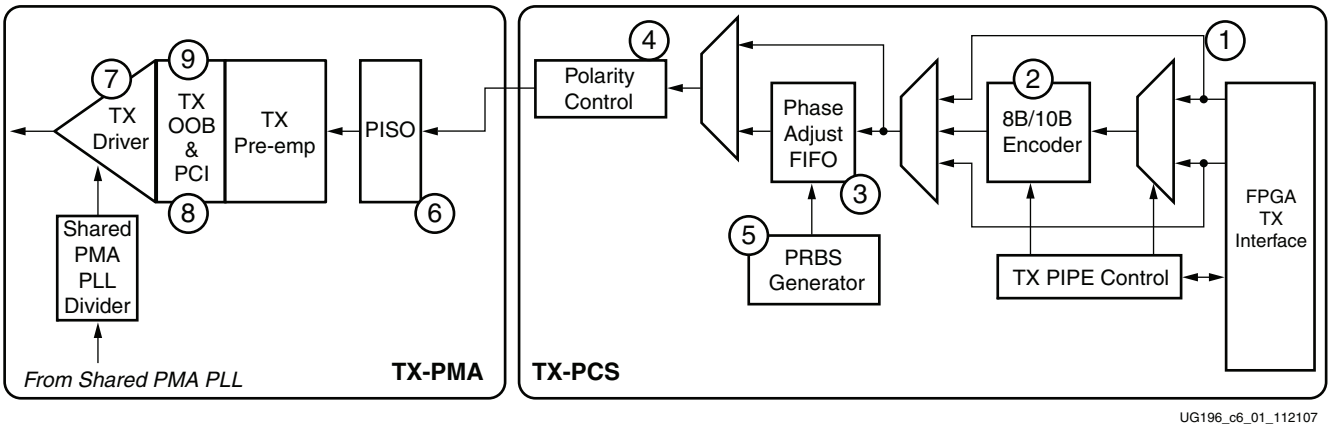


Figure E-2: **GTP TX Block Diagram**

Table E-1 defines the latency for the specific functional blocks or group of functional blocks of the TX section of the GTP transceiver. The values in the Block Number column correspond to the circled numbers in Figure E-2.

Table E-1: **GTP TX Latency**

Block Number	Block Name	Latency	
1	FPGA TX Interface	**TXDATAWIDTH = 0**	**TXDATAWIDTH = 1**
		1 cycle	1.5 cycles
2	8B/10B Encoder	**TXENC8B10BUSE = 0**	**TXENC8B10BUSE = 1**
		0 cycles	1 cycle
3	TX Buffer	**TX_BUFFER_USE = 0**	**TX_BUFFER_USE = 1**
		1 cycle	1-5 cycles
4+6+7+8+9	PMA + Interface	2 cycles	
Total Latency		**Maximum**	**Minimum**
		9.5 cycles	4 cycles

Notes:

1. 1 cycle = 1 clock cycle at the TXUSRCLK rate.

Apotex Ex1009
Page 336 of 347

GTP RX Latency

Figure E-3 shows a detailed block diagram of the GTP RX. Refer to Chapter 7, GTP Receiver (RX), and Figure 7-1, page 137 for more details on this figure and the GTP RX blocks.

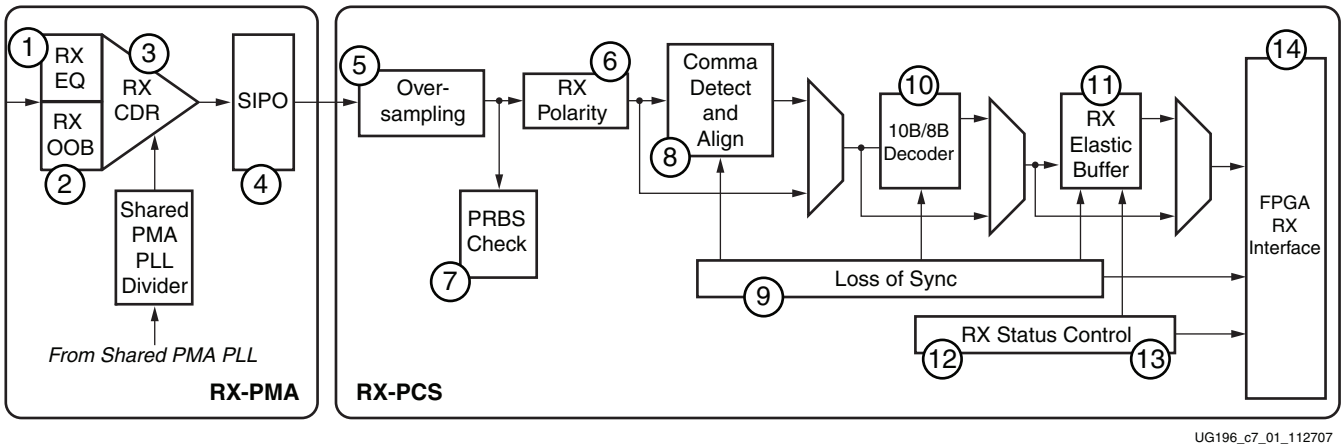


Figure E-3: **GTP RX Block Diagram**

Table E-2 defines the latency for the specific functional blocks or group of functional blocks of the receiver section of the GTP transceiver. The values in the Block Number column correspond to the circled numbers in Figure E-3.

Table E-2: **GTP RX Latency**

Block Number	Block Name	Latency	
1+2+3+4	PMA + Interface	1.5 cycles ± 1 UI	
5+6	Oversampling	**OVERSAMPLE_MODE = FALSE**	**OVERSAMPLE_MODE = TRUE**
		0 cycles	1 cycle
8	Comma Alignment	**RXCOMMADETUSE = 0**	**RXCOMMADETUSE = 1**
		3 cycles	3-5 cycles
10	8B/10B Encoder	**RXDEC8B10BUSE = 0(2)**	**RXDEC8B10BUSE = 1**
		0 cycles	1 cycle
11	RX Elastic Buffer	**RX_BUFFER_USE = 0**	**RX_BUFFER_USE = 1**
		2 cycles	2 cycles + CLK_COR_MIN_LAT
14	FPGA RX Interface	**RXDATAWIDTH = 0**	**RXDATAWIDTH = 1**
		2 cycles	3 cycles
Total Latency		**Maximum**	**Minimum**
Not Oversampling		13.5 + CLK_COR_MIN_LAT	8.5 cycles

Notes:

1. 1 cycle = 1 clock cycle at the RXUSRCLK rate.
2. When the RX elastic buffer is bypassed, a 10-bit internal data width is necessary. Therefore, INTDATAWIDTH is High.

Cpuver Ex1009
Page 338 of 347

Appendix F

Advanced Clocking

Each GTP_DUAL primitive contains a reference clock multiplexing structure that is addressable using the dynamic reconfiguration port (DRP). This structure can connect one out of four different reference clock sources to the CLKIN port of the GTP_DUAL tile's shared PMA PLL.

Direct manipulation of the reference clock multiplexers using the DRP produces flexible reference clocking arrangements instead of clocking with assignments in HDL. The reference clock applied to a particular tile can be changed at run-time. GTPRESET is applied after the clocking arrangement is changed.

There are several rules to keep in mind when designing a multi-clock scheme:

- Only one clock can be forwarded northbound through a tile at a time.
- Only one clock can be forwarded southbound through a tile at a time.
- A clock cannot be forwarded more than three tiles from its tile of origin in either direction.
- A clock can be forwarded northbound or southbound through a tile, even if the GTP transceivers in that tile are not using the forwarded clock.

Overlapping clock regions can be constructed when using advanced clocking, a practice that is not allowed when using HDL to connect the reference clock.

The reference clock multiplexing structure is shown in Figure F-1. The X's in the REFCLK MUX are don't cares. Refer to REFCLK Guidelines in Chapter 10 for IBUFDS details.

Note: Refer to Chapter 3, Simulation for the correct simulation-only attribute settings, SIM_PLL_PERDIV2 and SIM_GTPRESET_SPEEDUP, for simulating multirate designs. Refer to the Clocking, Reset, and Power Control sections of Chapter 5 for additional information.

Optiver Ex1009
Page 339 of 347

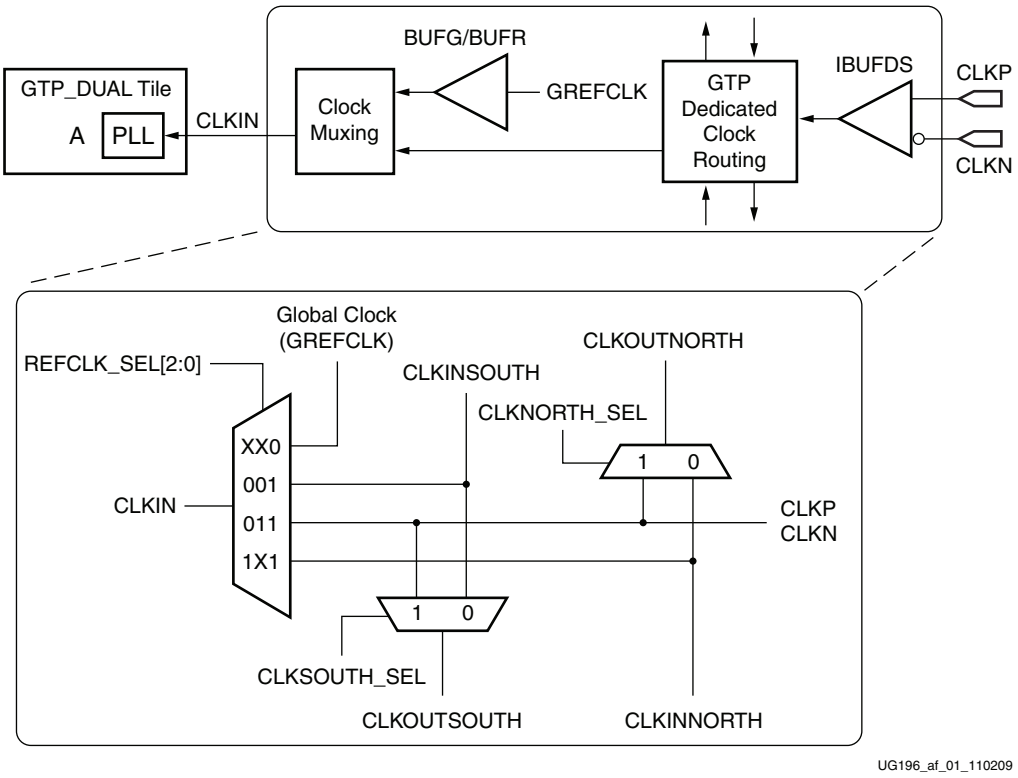


Figure F-1: **Reference Clock Multiplexing Structure**

All the MUX selectors reside in address `0x04` of the DRP and are mapped as shown in Table F-1.

Table F-1: **MUX Selector**

REFCLK_SEL[2:0]	Bit	Address
REFCLK_SEL[0]	6	`0x04`
REFCLK_SEL[1]	5	`0x04`
REFCLK_SEL[2]	4	`0x04`
CLKSOUTH_SEL	7	`0x04`
CLKNORTH_SEL	8	`0x04`

To ensure that other attributes in DRP address `0x04` are not accidentally changed, a read/modify/write procedure must be used to change the MUX selectors.

Apotex Ex1009
Page 340 of 347

Example

The example system shown in Figure F-2 has six GTP_DUAL tiles with four reference clocks. It demonstrates several different clocking schemes.

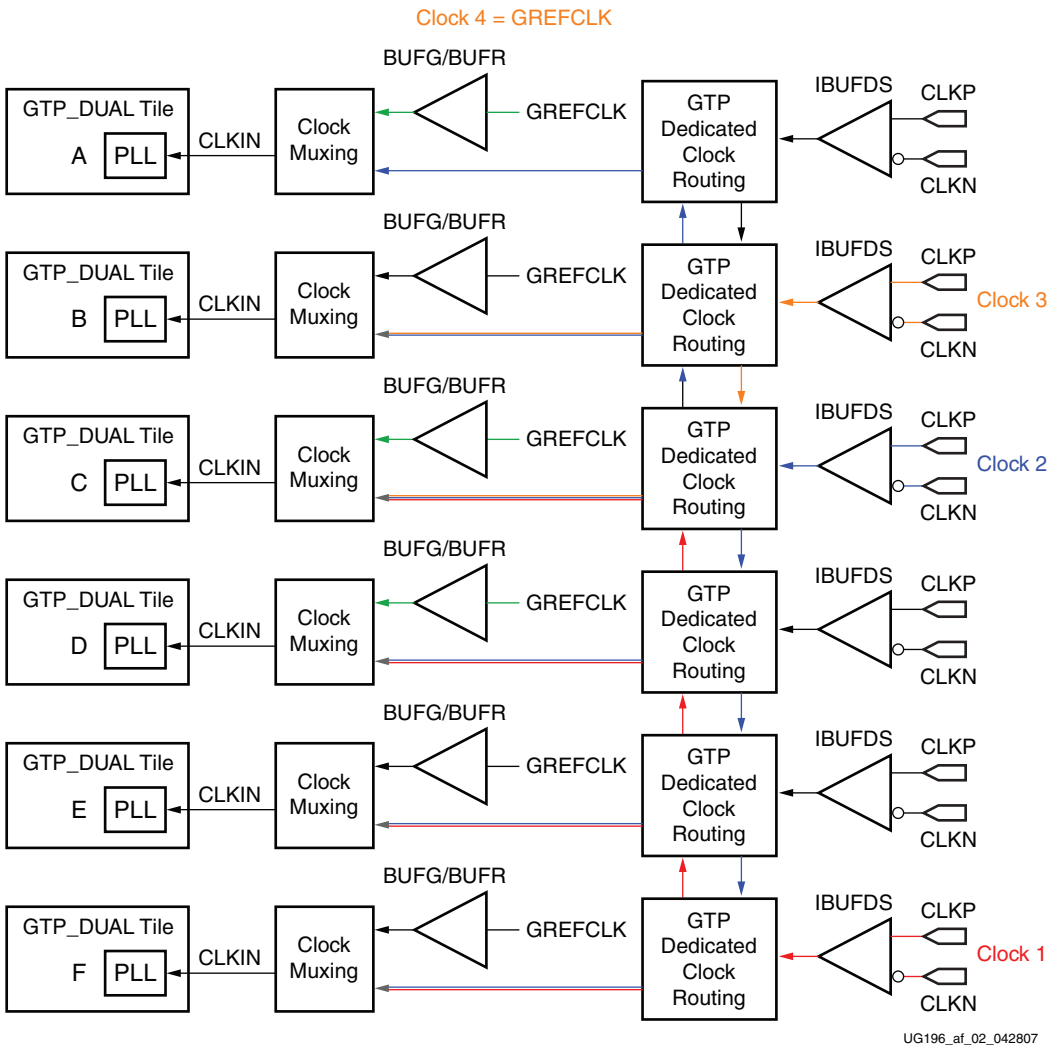


Figure F-2: **Example System**

Note: Any GTP_DUAL tile that sources a reference clock must be instantiated, and REFCLKPWRDNB must be asserted High.

Table F-2 describes the example in Figure F-2. When GTP_DUAL tile D is used to select clock 1 as the CLKIN source, REFCLK_SEL is set to `1x1`.

Table F-2: **GTP Tile Utilization**

GTP_DUAL Tile	CLKIN Options	REFCLK_SEL[2:0]	CLKNORTH_SEL	CLKSOUTH_SEL
A	Clock 2	`1x1`(1)	Don't Care: Nowhere to send the northbound clock.	Don't Care: No southbound clock to forward.
	Clock 4	`xx0`		
B	Clock 2	`1x1`	0: Forward clock 2 northbound to tile A.	1: Drive clock 3 southbound to tile C.
	Clock 3	`011`		
C	Clock 1	`1x1`	1: Drive clock 2 northbound to tile B.	1: Drive clock 2 southbound to tile D.
	Clock 2	`011`		
	Clock 3	`001`		
	Clock 4	`xx0`		
D	Clock 1	`1x1`	0: Forward clock 1 northbound to tile C.	0: Forward clock 2 southbound to tile E.
	Clock 2	`001`		
	Clock 4	`xx0`		
E	Clock 1	`1x1`	0: Forward clock 1 northbound to tile D.	0: Forward clock 2 southbound to tile F.
	Clock 2	`001`		
F	Clock 1	`011`	1: Drive clock 1 northbound to tile E.	Don't Care: Clock 2 cannot be sent more than three tiles from where it originates.
	Clock 2	`001`		

Notes:

1. x = don't care.

Oplover Ex1009
Page 342 of 347

Index

Numerics

1-byte mode 107
2-byte mode 108
2D field solvers 259, 265
3D field solvers 263, 266
5x oversampling 117, 152, 155, 157
8B/10B
  Benefits 111
  Commas 174
  Decoder 172, 173
    Ordering 173
  Decoding 187
  Encoder 104, 111
    Bypassing 115
    Disabling/Enabling 113
    Ordering 113
  Encoding 111

A

AC coupling 140, 141, 240, 252, 265
Alignment
  Comma 162, 167
  Manual 168
  SONET A1/A2 163
Analog design guidelines 219
Analog pin definitions 219
Analog pin summary 28
Attribute mapping 305
Attributes
  Analog 220, 221
  Channel bonding 190
  Clock correction 185
  CRC 207
  Loopback 214
  OOB/beacon signaling 134
  Power 95
  PRBS detection 161
  RX CDR 150
  RX comma alignment and detection 165
  RX decoder 173
  RX digital oversampling (DCDR) 158
  RX elastic buffer 178
  RX loss-of-sync state machine 170
  RX OOB/beacon signaling 145
  RX phase alignment 178
  RX termination and equalization 140
  Shared clocking 82
  Shared PMA PLL 73
  SIPO 155
  TX buffering 118
  TX driver 127
  TX phase alignment 118
  TX PISO 125

B

Backplane connectors 272
Baseline wander 253
Beaconing 133, 134, 143, 293, 294
BGA adjacency guidelines 242
BGA packages 265
  Escape example 278
Bit inversion 211
Block diagram, GTP_DUAL tile 27
Blocking capacitor 252, 253, 277
  Calculation of 254
Boundary-Scan guidelines 229
Byte rotation 211

C

Cables 262
Capacitance and inductance
  Equations 264
  Excess 263
Capacitor guidelines 229
CDR lock, detection of 180
Channel bonding 189
  Configuration 193
  Connecting ports 193
  Reset conditions 92
  Sequence 197
Channel, definition 249
CLKIN 72, 80, 107, 203, 339
  Options 80
Clock buffers, high-speed 252
Clock configurations and correction 177
Clock correction 183
  Attributes 185
  Enabling 187
  Frequency control 188
  Monitoring 188
  Ports 184
  Precedence 199
  Sequences 187
Clock data recovery 149
Clock relationships 203
Clock relock 91
Clock routing, dedicated 80
Clock settings 75
Clock stability 91
Clock traces 251
Clocking rules 107
COM sequence timing 135
Comma
  8B/10B 174
  Alignment 162, 167
  Pattern, configuration 166
Comma alignment
  Enabling 166
Component reset 87
Connectors 262
  HM-Zd 278
  Press fit 280
CORE Generator tool 23, 45
Coupling mechanisms, SelectIO signals 256
CRC 205
  Attributes 44, 207
  Ports 37, 206
CRC checking, methods 212
CRC_INIT value
  Ethernet 209
  Fibre Channel 209
  Infiniband 209
  PCI Express 209
  SATA 209
CRC32 primitive 208
  RX 211
  TX 211
CRC64 primitive 208
  RX 211
  TX 211
CRCOUT 211
CRCRESET 209
Crosstalk 242, 255
Current mode logic 139

Optiver Ex1009
Page 343 of 347

D

Data characters 295
DC balance 252
DC coupling 140, 141, 252
DCM 50, 91, 108, 110, 120, 180
De-emphasis 129
Design migration 283
Detection threshold 132
Dielectric loss 257
Differential clock input pairs, multiple 239
Differential output voltage 127
Differential swing
  Amplitude control 127
Differential via 269, 277
  GSSG 278
Disparity 114, 174
  Errors 174
DIV 74
Divider, RX 74
Divider, TX 74
DRP 101, 135, 339
  Address (by attribute) 307
  Address (by bit location) 322
DRP ports 101
DRP table
  Attribute mapping 305
Dynamic reconfiguration port 101, 339

E

Electrical idle 87
Error checking 207
Ethernet, CRC_INIT value 209
External ports 57

F

Ferrite guidelines 228
Fibre Channel, CRC_INIT value 209
Field solvers
  2D 259, 265
  3D 263, 266
Filter network guidelines 229
FPGA RX interface 200
  Enabling 201
FPGA TX interface 104, 105
FTS lane deskew 193

G

Gigabit Ethernet, shared PMA PLL settings 78
GREFCLK 80
GREFCLK clocking 80, 84
Ground planes 261
GSR 51, 52
GTP transceiver
  Definition 23
  Features 23
  Placement 24
GTP_DUAL column
  Illustration 24
GTP_DUAL tile
  Attribute summary 37
  Block diagram 27
  Configuration 71
  Definition 24
  Placement 58, 60
    Example 24
    LXT 60
    SXT 60
  Port summary 29
  Reset 84
GTP0 28
GTP1 28
GTPRESET 84, 87, 119, 151, 179, 180, 339
GTS 52
Guidelines
  Analog design 219
  BGA adjacency 242
  Boundary-Scan 229
  Capacitors 229
  Ferrites 228
  Filter network 229
  PCB 277
  Reference clock 236
  SelectIO 242, 255
  Signal attenuation 257
  Summary 277
  Voltage regulators 228

H

Harmonics 257
HFSS 266
Horizontal sampling point shift 153

I

IBUFDS primitive 80, 82
Infiniband, CRC_INIT value 209
Inserted idles 188
INTDATAWIDTH 74, 104, 106, 124, 159, 162, 167, 179, 180, 187, 198, 202
Internal datapath width 78
  Gigabit Ethernet 78
  OC-48 77
  PCI Express 79
  XAUI 77
ISE development system 53
ISE tool 73

J

Jitter 252, 253
Jitter margins 83
Jog-outs 272

K

K characters 114, 174, 295
K28.5 290

L

Latency 335
  RX 337
  TX 336
Limitations
  Loopback 215, 216
  Simulation 50
Line rate
  5x 158
  Oversampled 159
  PMA 158
  RX 156
  TX 125
Linear equalizer circuit 143
Linear regulator, selection of 226
Linear regulators 250
Link idle reset 51, 71, 87, 92, 151
Linux 53
Lock to reference 153
Loopback
  Attributes 214
  Ports 214
Loopback mode 213
  Far-End PCS 217
  Far-End PMA 216
  Near-End PCS 214
  Near-End PMA 215
Loss of Sync state machine 169, 180

Optiver Ex1009
Page 344 of 347

Loss tangent 258
LXT package 60

M

Mapping
  Channel bonding sequence 198
  Clock correction sequence 188
MGT differences 283
Microstrips 272, 280
Migration to GTP transceivers 283
ModelSim SE 6.1d 53
Multi-clock design 339

N

Noise, minimizing 240

O

OC-48, shared PMA PLL settings 77
OOB signaling 133, 134, 143, 293
Ordering 113, 173
Oscillator
  Characteristics 236
  Crystal 251
  PLL based 91
  Selection 239
Out-of-Band signaling 133, 134, 143, 293
Overflow
  Buffer 92, 119, 159, 179
  Oversampling block 93
Overlapping clock regions 339
OVERSAMPLE_MODE 74
Oversampling 117, 152, 155, 157
  Configuring 158
Oversampling block 159

P

P/N crossover vias 272
P/N length mismatches 279
P2 power state 134
Package traces and transitions 265
Parallel clock domain, RX
  RXUSRCLK 176, 178
  XCLK 176, 178
Parallel clock domain, TX
  TXUSRCLK 115
  XCLK 115
Parallel clock domains 115, 176
Parallel clock examples 107
Parallel clock rate 200
Pattern dependent jitter 252, 253
PCB guidelines 277
PCI Express
  Beaconing 133, 134, 143, 293, 294
  Blocking capacitor values 253
  CRC 208
  CRC_INIT value 209
  Electrical idle 146, 193
  Far-End PCS Loopback 217
  Power control 94, 97
  Shared PMA PLL settings 79
PDJ 252, 253
Phase alignment
  RX 180
PIPE specification 95, 97, 130, 133, 143, 144, 193, 294
PISO block 124
Placement
  Example 24
  GTP_DUAL tile 60
PLL clock 72, 74
  Setting 74
PLL power down 96
PMA_COM_CFG 216
Point of load 250
POL power distribution 250
Polarity control
  RX 160
  TX 122
Port width 104
Ports
  Channel bonding 190
  Clock correction 184
  CRC 206
  DRP 101
  External 57
  FPGA RX interface 200
  FPGA TX interface 104
  Loopback 214
  PCI Express receive detect 130
  Power 94
  PRBS detection 161
  Reset 85
  RX CDR 149
  RX comma alignment and detection 163
  RX decoder 172
  RX digital oversampling (DCDR) 157
  RX elastic buffer 177
  RX loss-of-sync state machine 169
  RX OOB/beacon signaling 144
  RX phase alignment 177
  RX polarity 160
  RX termination and equalization 139
  Shared clocking 82
  Shared PMA PLL 72
  SIPO 155
  TX buffering 117
  TX driver 127
  TX encoder 112
  TX OOB/beacon signaling 133
  TX phase alignment 117
  TX PISO 124
  TX polarity control 122
  TX PRBS generator 123
Power attributes 95
Power consumption, minimizing 240
Power control 96
Power distribution system 225
Power down 96
  PCI Express designs 95, 131, 193
  PLL 96
  REFCLK 96
  RX 94
  RX and TX 97
  TX 95
  Unused tile or transceiver 98
Power pin voltages 287
Power ports 94
Power state 131
  Non PCI Express 97
  P2 134
  RX 94
  TX 95
Power supply
  Filtering 288
  Principles 250
  Ripple rejection 225, 226
Power-down state
  PCI Express 97
PRBS
  Checker 161
  Generator 123
  Test patterns 123
PRBS error 93
Pre-emphasis 128
Protocols
  Example settings 75
  Supported 23
PSRR 225, 226

Optiver Ex1009
Page 345 of 347

R

Receive detection 131
Receiver detection 130
Recovered clock 149, 176, 179, 183
REFCLK power down 96
REFCLKOUT 107, 120
  USRCLK generation 110
Reference clock 80
  Changing 91
  CLKIN 72, 80
  Guidelines 236
  Multiple 240, 341
  Multiplexing 339
  Sharing rules 83
  Stability 91
  Structure 339
  Unused 240
Regulator selection 251
Regulators
  Linear 250
Related documentation 18
Relative permittivity 257
Relocking 91, 92
Reset
  Channel bonding 92
  Component 87
  GTP_DUAL tile 84
  Link idle 51, 87
  Methods 89
  RX CDR 151
  Situations 90
Reset ports 85
Reset sequence 86
  Affected sections 86
  GTPRESET 87
Resources, additional 19
Return current 261
Rise time
  Receiver 132
RocketIO GTP Transceiver Wizard 45, 58, 76
Running disparity 114, 174
RX CDR 149
  Relock 92
  Reset 91, 151
RX datapath width 201
RX elastic buffer 176
  Bypassing 179
  Error 92
  Limits 187
  Resolving phase differences 179
RX PCS 137
RX phase alignment
  Flow diagram 181
  Procedure 180
RX PMA 137
RX power down 97
RX termination, configuration 142
RXCDRRESET 151
RXELECIDLERESET 91
RXEQMIX 143
RXEQPOLE 143
RXRECCLK 203
RXSTATUS
  Synchronization 135
RXUSRCLK/RXUSRCLK2 178, 202, 203
RXUSRCLK2 200

S

SATA
  Blocking capacitor values 253
  CRC_INIT value 209
  OOB signaling 293
SATA auto-negotiation 133
SATA specification 133, 143
SelectIO guidelines 242, 255
SelectIO signals, performance impact 243
Shared clocking
  Attributes 82
  Ports 82
Shared PMA PLL 72, 74, 158
Shielding 243
Signal attenuation guidelines 257
Signal distortion compensation 143
SIM_PLL_PERDIV2 50, 51
  Calculation of 54
SIPO block 155
Skew
  Cable 262
  Maximum 198
  Minimizing 116
Skin effect 258
SMA connectors 272
SmartModel 47, 49, 53
SmartModel attributes 50
SMT pads 265
SONET A1/A2 alignment 163
Spread spectrum clocking 152
Squelch clock 147
Standards, supported 23
Striplines 272, 280
Substrate material, selection 258
Switch, high-speed 252
Switching regulators, unsuitability 250
SXT package 60

T

TDR 263
Termination impedance 129, 140, 220, 221
Termination voltage 142
Time domain reflectometry 263
Trace geometry 259
Traces, clock 251
Transitions
  Common 249
  Definition 249, 272
  Design of 263
Transmission lines 263
  Impedance 259
  Lossy 262
TX buffer 119
  Bypassing 73, 118
  Error 92
  Status 117
  Trade-offs 116
TX datapath width 105
TX driver 126
TX PCS 103
TX phase alignment
  Clock stability 120
TX PISO attributes 125
TX PISO ports 124
TX PMA 103
TX polarity control 122
TX polarity control ports 122
TX power down 97
TX PRBS generator ports 123
TX_BUFFER_USE 216
TXDATAWIDTH 108
TXOUTCLK 107, 108
  Multiple clocks 109
TXUSRCLK 120
  Calculation 107
TXUSRCLK/TXUSRCLK2 106, 107
TXUSRCLK2 calculation 107

U

UCF 73
  Creation 58
  Example 59
Underflow

Power Ex1009
Page 346 of 347

Buffer 92, 119, 159, 179
Oversampling block 93

V

Valid Data characters 295
Valid K characters 303
Via
Differential 269, 277, 278
Large 280
P/N crossover 272
Stub length 271, 280
Voltage regulator 225
Characteristics 225
Guidelines 228

W

Width
Port 104
RX datapath 201
RXDATA 200
TX datapath 105
Wizard 45, 58, 76

X

XAUI, shared PMA PLL settings 76