Stratix II GX Device Handbook, Volume 1

101 Innovation Drive
San Jose, CA 95134
www.altera.com

SIIGX5V1-4.4

Contents

Section I. Stratix II GX Device Data Sheet

Chapter 1. Introduction

Chapter 2. Stratix II GX Architecture

Chapter 3. Configuration & Testing

Chapter 4. DC and Switching Characteristics

Chapter 5. Reference and Ordering Information

Chapter Revision Dates

The chapters in this book, *Stratix II GX Device Handbook, Volume 1*, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Introduction
Revised: *October 2007*
Part number: *SIIGX51001-1.6*

Chapter 2. Stratix II GX Architecture
Revised: *October 2007*
Part number: *SIIGX51003-2.2*

Chapter 3. Configuration & Testing
Revised: *October 2007*
Part number: *SIIGX51005-1.4*

Chapter 4. DC and Switching Characteristics
Revised: *June 2009*
Part number: *SIIGX51006-4.6*

Chapter 5. Reference and Ordering Information
Revised: *August 2007*
Part number: *SIIGX51007-1.3*

About this Handbook

This handbook provides comprehensive information about the Altera® Stratix II GX family of devices.

How to Contact Altera

For the most up-to-date information about Altera products, refer to the following table.

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Email	www.altera.com/literature
Altera literature services	Website	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to table:
(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*

Visual Cue	Meaning
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>*.**pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

Section I. Stratix II GX Device Data Sheet

This section provides designers with the data sheet specifications for Stratix® II GX devices. They contain feature definitions of the transceivers, internal architecture, configuration, and JTAG boundary-scan testing information, DC operating conditions, AC timing parameters, a reference to power consumption, and ordering information for Stratix II GX devices.

This section includes the following chapters:

- Chapter 1, Introduction
- Chapter 2, Stratix II GX Architecture
- Chapter 3, Configuration & Testing
- Chapter 4, DC and Switching Characteristics
- Chapter 5, Reference and Ordering Information

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

1. Introduction

SIIGX51001-1.6

The Stratix® II GX family of devices is Altera's third generation of FPGAs to combine high-speed serial transceivers with a scalable, high-performance logic array. Stratix II GX devices include 4 to 20 high-speed transceiver channels, each incorporating clock and data recovery unit (CRU) technology and embedded SERDES capability at data rates of up to 6.375 gigabits per second (Gbps). The transceivers are grouped into four-channel transceiver blocks and are designed for low power consumption and small die size. The Stratix II GX FPGA technology is built upon the Stratix II architecture and offers a 1.2-V logic array with unmatched performance, flexibility, and time-to-market capabilities. This scalable, high-performance architecture makes Stratix II GX devices ideal for high-speed backplane interface, chip-to-chip, and communications protocol-bridging applications.

Features

This section lists the Stratix II GX device features.

- Main device features:
  - TriMatrix memory consisting of three RAM block sizes to implement true dual-port memory and first-in first-out (FIFO) buffers with performance up to 550 MHz
  - Up to 16 global clock networks with up to 32 regional clock networks per device region
  - High-speed DSP blocks provide dedicated implementation of multipliers (at up to 450 MHz), multiply-accumulate functions, and finite impulse response (FIR) filters
  - Up to four enhanced PLLs per device provide spread spectrum, programmable bandwidth, clock switch-over, real-time PLL reconfiguration, and advanced multiplication and phase shifting
  - Support for numerous single-ended and differential I/O standards
  - High-speed source-synchronous differential I/O support on up to 71 channels
  - Support for source-synchronous bus standards, including SPI-4 Phase 2 (POS-PHY Level 4), SFI-4.1, XSBI, UTOPIA IV, NPSI, and CSIX-L1
  - Support for high-speed external memory, including quad data rate (QDR and QDRII) SRAM, double data rate (DDR and DDR2) SDRAM, and single data rate (SDR) SDRAM

- Support for multiple intellectual property megafunctions from Altera® MegaCore® functions and Altera Megafunction Partners Program (AMPP^{SM}) megafunctions
- Support for design security using configuration bitstream encryption
- Support for remote configuration updates

- Transceiver block features:
  - High-speed serial transceiver channels with clock data recovery (CDR) provide 600-megabits per second (Mbps) to 6.375-Gbps full-duplex transceiver operation per channel
  - Devices available with 4, 8, 12, 16, or 20 high-speed serial transceiver channels providing up to 255 Gbps of serial bandwidth (full duplex)
  - Dynamically programmable voltage output differential (V_{OD}) and pre-emphasis settings for improved signal integrity
  - Support for CDR-based serial protocols, including PCI Express, Gigabit Ethernet, SDI, Altera's SerialLite II, XAUI, CEI-6G, CPRI, Serial RapidIO, SONET/SDH
  - Dynamic reconfiguration of transceiver channels to switch between multiple protocols and data rates
  - Individual transmitter and receiver channel power-down capability for reduced power consumption during non-operation
  - Adaptive equalization (AEQ) capability at the receiver to compensate for changing link characteristics
  - Selectable on-chip termination resistors (100, 120, or 150 Ω) for improved signal integrity on a variety of transmission media
  - Programmable transceiver-to-FPGA interface with support for 8-, 10-, 16-, 20-, 32-, and 40-bit wide data transfer
  - 1.2- and 1.5-V pseudo current mode logic (PCML) for 600 Mbps to 6.375 Gbps (AC coupling)
  - Receiver indicator for loss of signal (available only in PIPE mode)
  - Built-in self test (BIST)
  - Hot socketing for hot plug-in or hot swap and power sequencing support without the use of external devices
  - Rate matcher, byte-reordering, bit-reordering, pattern detector, and word aligner support programmable patterns
  - Dedicated circuitry that is compliant with PIPE, XAUI, and GIGE
  - Built-in byte ordering so that a frame or packet always starts in a known byte lane
  - Transmitters with two PLL inputs for each transceiver block with independent clock dividers to provide varying clock rates on each of its transmitters

- 8B/10B encoder and decoder perform 8-bit to 10-bit encoding and 10-bit to 8-bit decoding
- Phase compensation FIFO buffer performs clock domain translation between the transceiver block and the logic array
- Receiver FIFO resynchronizes the received data with the local reference clock
- Channel aligner compliant with XAUI

Certain transceiver blocks can be bypassed. Refer to the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook* for more details.

Table 1–1 lists the Stratix II GX device features.

Table 1–1. Stratix II GX Device Features (Part 1 of 2)

Feature	EP2SGX30C/D		EP2SGX60C/D/E			EP2SGX90E/F		EP2SGX130/G
	C	D	C	D	E	E	F	G
ALMs	13,552		24,176			36,384		53,016
Equivalent LEs	33,880		60,440			90,960		132,540
Transceiver channels	4	8	4	8	12	12	16	20
Transceiver data rate	600 Mbps to 6.375 Gbps		600 Mbps to 6.375 Gbps			600 Mbps to 6.375 Gbps		600 Mbps to 6.375 Gbps
Source-synchronous receive channels *(1)*	31		31	31	42	47	59	73
Source-synchronous transmit channels	29		29	29	42	45	59	71
M512 RAM blocks (32 × 18 bits)	202		329			488		699
M4K RAM blocks (128 × 36 bits)	144		255			408		609
M-RAM blocks (4K × 144 bits)	1		2			4		6
Total RAM bits	1,369,728		2,544,192			4,520,448		6,747,840
Embedded multipliers (18 × 18)	64		144			192		252
DSP blocks	16		36			48		63
PLLs	4		4	4	8	8		8
Maximum user I/O pins	361		364	364	534	558	650	734

Table 1–1. Stratix II GX Device Features (Part 2 of 2)

Feature	EP2SGX30C/D		EP2SGX60C/D/E			EP2SGX90E/F		EP2SGX130/G
	C	D	C	D	E	E	F	G
Package	780-pin FineLine BGA		780-pin FineLine BGA		1,152-pin FineLine BGA	1,152-pin FineLine BGA	1,508-pin FineLine BGA	1,508-pin FineLine BGA

Note to Table 1–1:
(1) Includes two sets of dual-purpose differential pins that can be used as two additional channels for the differential receiver or differential clock inputs.

Stratix II GX devices are available in space-saving FineLine BGA packages (refer to Table 1–2). All Stratix II GX devices support vertical migration within the same package. Vertical migration means that you can migrate to devices whose dedicated pins, configuration pins, and power pins are the same for a given package across device densities. For I/O pin migration across densities, you must cross-reference the available I/O pins using the device pin-outs for all planned densities of a given package type to identify which I/O pins are migratable. Table 1–3 lists the Stratix II GX device package sizes.

Table 1–2. Stratix II GX Package Options (Pin Counts and Transceiver Channels)

Device	Transceiver Channels	Source-Synchronous Channels		Maximum User I/O Pin Count		
		Receive *(1)*	Transmit	780-Pin FineLine BGA (29 mm)	1,152-Pin FineLine BGA (35 mm)	1,508-Pin FineLine BGA (40 mm)
EP2SGX30C	4	31	29	361	—	—
EP2SGX60C	4	31	29	364	—	—
EP2SGX30D	8	31	29	361	—	—
EP2SGX60D	8	31	29	364	—	—
EP2SGX60E	12	42	42	—	534	—
EP2SGX90E	12	47	45	—	558	—
EP2SGX90F	16	59	59	—	—	650
EP2SGX130G	20	73	71	—	—	734

Note to Table 1–2:
(1) Includes two differential clock inputs that can also be used as two additional channels for the differential receiver.

Table 1–3. Stratix II GX FineLine BGA Package Sizes			
Dimension	**780 Pins**	**1,152 Pins**	**1,508 Pins**
Pitch (mm)	1.00	1.00	1.00
Area (mm^2)	841	1,225	1,600
Length width (mm × mm)	29 × 29	35 × 35	40 × 40

Referenced Document

This chapter references the following document:

- *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*

Document Revision History

Table 1–4 shows the revision history for this chapter.

Table 1–4. Document Revision History		
Date and Document Version	**Changes Made**	**Summary of Changes**
October 2007, v1.6	Updated "Features" section.	
	Minor text edits.	
August 2007, v1.5	Added "Referenced Documents" section.	
	Minor text edits.	
February 2007, v1.4	• Changed 622 Mbps to 600 Mbps on page 1-2 and Table 1–1. • Deleted "DC coupling" from the Transceiver Block Features list. • Changed 4 to 6 in the PLLs row (columns 3 and 4) of Table 1–1.	
	Added the "Document Revision History" section to this chapter.	Added support information for the Stratix II GX device.
June 2006, v1.3	• Updated Table 1–2.	
April 2006, v1.2	• Updated Table 1–1. • Updated Table 1–2.	Updated numbers for receiver channels and user I/O pin counts in Table 1–2.
February 2006, v1.1	• Updated Table 1–1.	
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	

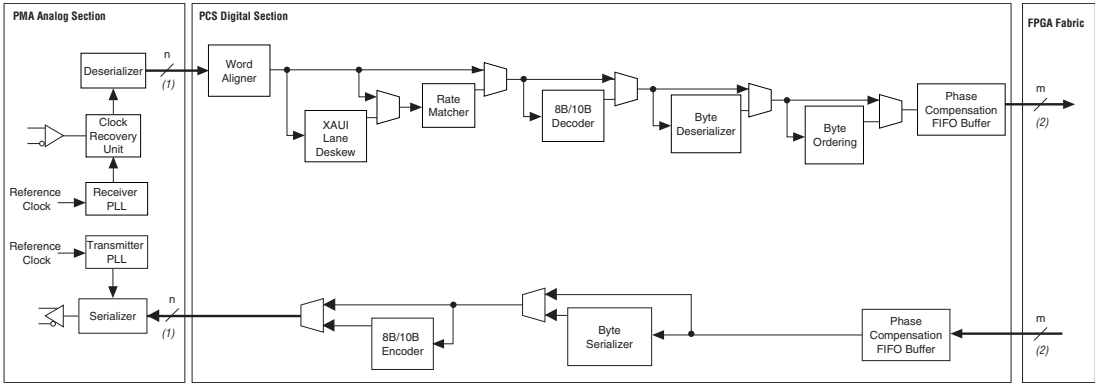
2. Stratix II GX Architecture

SIIGX51003-2.2

Transceivers

Stratix® II GX devices incorporate dedicated embedded circuitry on the right side of the device, which contains up to 20 high-speed 6.375-Gbps serial transceiver channels. Each Stratix II GX transceiver block contains four full-duplex channels and supporting logic to transmit and receive high-speed serial data streams. The transceivers deliver bidirectional point-to-point data transmissions, with up to 51 Gbps (6.375 Gbps per channel) of full-duplex data transmission per transceiver block. Figure 2–1 shows the function blocks that make up a transceiver channel within the Stratix II GX device.

Figure 2–1. Stratix II GX Transceiver Block Diagram



Notes to Figure 2–1:

(1) *n* represents the number of bits in each word that need to be serialized by the transmitter portion of the PMA or have been deserialized by the receiver portion of the PMA. *n* = 8, 10, 16, or 20.

(2) *m* represents the number of bits in the word that pass between the FPGA logic and the PCS portion of the transceiver. *m* = 8, 10, 16, 20, 32, or 40.

Transceivers within each block are independent and have their own set of dividers. Therefore, each transceiver can operate at different frequencies. Each block can select from two reference clocks to provide two clock domains that each transceiver can select from.

There are up to 20 transceiver channels available on a single Stratix II GX device. Table 2–1 shows the number of transceiver channels and their serial bandwidth for each Stratix II GX device.

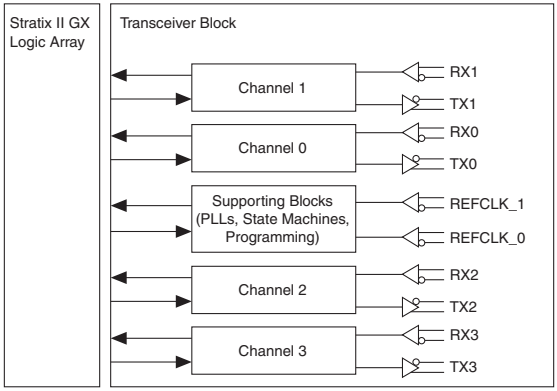
Table 2–1. Stratix II GX Transceiver Channels

Device	Number of Transceiver Channels	Serial Bandwidth (Full Duplex)
EP2SGX30C	4	51 Gbps
EP2SGX60C	4	51 Gbps
EP2SGX30D	8	102 Gbps
EP2SGX60D	8	102 Gbps
EP2SGX60E	12	153 Gbps
EP2SGX90E	12	153 Gbps
EP2SGX90F	16	204 Gbps
EP2SGX130G	20	255 Gbps

Figure 2–2 shows the elements of the transceiver block, including the four transceiver channels, supporting logic, and I/O buffers. Each transceiver channel consists of a receiver and transmitter. The supporting logic contains two transmitter PLLs to generate the high-speed clock(s) used by the four transmitters within that block. Each of the four transmitter channels has its own individual clock divider. The four receiver PLLs within each transceiver block generate four recovered clocks. The transceiver channels can be configured in one of the following functional modes:

- PCI Express (PIPE)
- OIF CEI PHY Interface
- SONET/SDH
- Gigabit Ethernet (GIGE)
- XAUI
- Basic (600 Mbps to 3.125 Gbps single-width mode and 1 Gbps to 6.375 Gbps double-width mode)
- SDI (HD, 3G)
- CPRI (614 Mbps, 1228 Mbps, 2456 Mbps)
- Serial RapidIO (1.25 Gbps, 2.5 Gbps, 3.125 Gbps)

Figure 2–2. Elements of the Transceiver Block



Each Stratix II GX transceiver channel consists of a transmitter and receiver. The transceivers are grouped in four and share PLL resources. Each transmitter has access to one of two PLLs. The transmitter contains the following:

- Transmitter phase compensation first-in first-out (FIFO) buffer
- Byte serializer (optional)
- 8B/10B encoder (optional)
- Serializer (parallel-to-serial converter)
- Transmitter differential output buffer

The receiver contains the following:

- Receiver differential input buffer
- Receiver lock detector and run length checker
- Clock recovery unit (CRU)
- Deserializer
- Pattern detector
- Word aligner
- Lane deskew
- Rate matcher (optional)
- 8B/10B decoder (optional)
- Byte deserializer (optional)
- Byte ordering
- Receiver phase compensation FIFO buffer

Designers can preset Stratix II GX transceiver functions using the Quartus® II software. In addition, pre-emphasis, equalization, and differential output voltage (V_{OD}) are dynamically programmable. Each Stratix II GX transceiver channel supports various loopback modes and is

capable of built-in self test (BIST) generation and verification. The ALT2GXB megafunction in the Quartus II software provides a step-by-step menu selection to configure the transceiver.

Figure 2–1 shows the block diagram for the Stratix II GX transceiver channel. Stratix II GX transceivers provide PCS and PMA implementations for all supported protocols. The PCS portion of the transceiver consists of the word aligner, lane deskew FIFO buffer, rate matcher FIFO buffer, 8B/10B encoder and decoder, byte serializer and deserializer, byte ordering, and phase compensation FIFO buffers.

Each Stratix II GX transceiver channel is also capable of BIST generation and verification in addition to various loopback modes. The PMA portion of the transceiver consists of the serializer and deserializer, the CRU, and the high-speed differential transceiver buffers that contain pre-emphasis, programmable on-chip termination (OCT), programmable voltage output differential (V_{OD}), and equalization.

Transmitter Path

This section describes the data path through the Stratix II GX transmitter. The Stratix II GX transmitter contains the following modules:

- Transmitter PLLs
- Access to one of two PLLs
- Transmitter logic array interface
- Transmitter phase compensation FIFO buffer
- Byte serializer
- 8B/10B encoder
- Serializer (parallel-to-serial converter)
- Transmitter differential output buffer

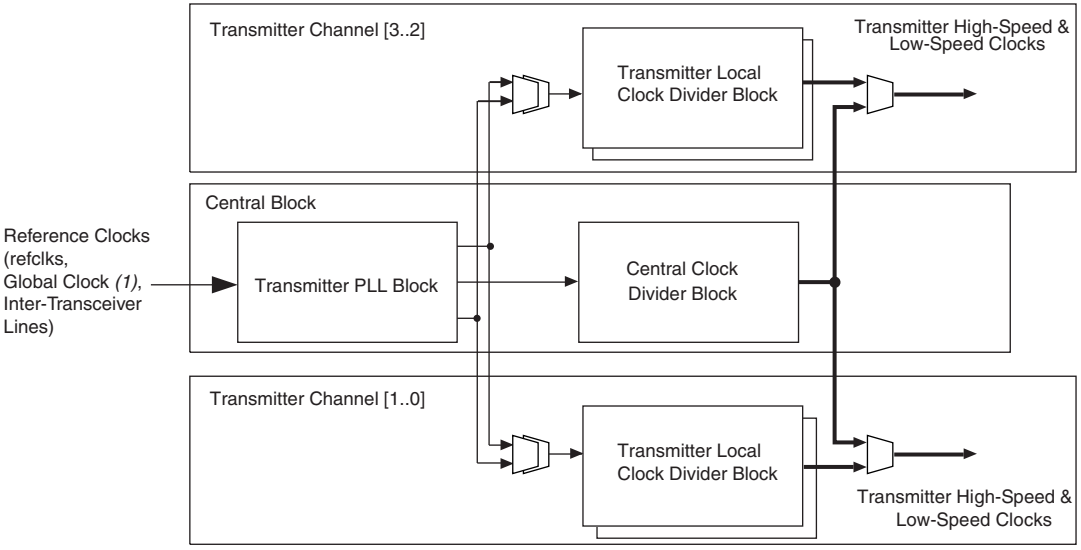
Transmitter PLLs

Each transceiver block has two transmitter PLLs which receive two reference clocks to generate timing and the following clocks:

- High-speed clock used by the serializer to transmit the high-speed differential transmitter data
- Low-speed clock to load the parallel transmitter data of the serializer

The serializer uses high-speed clocks to transmit data. The serializer is also referred to as parallel in serial out (PISO). The high-speed clock is fed to the local clock generation buffer. The local clock generation buffers divide the high-speed clock on the transmitter to a desired frequency on a per-channel basis. Figure 2–3 is a block diagram of the transmitter clocks.

Figure 2–3. Clock Distribution for the Transmitters Note (1)



Note to Figure 2–3:
(1) The global clock line must be driven by an input pin.

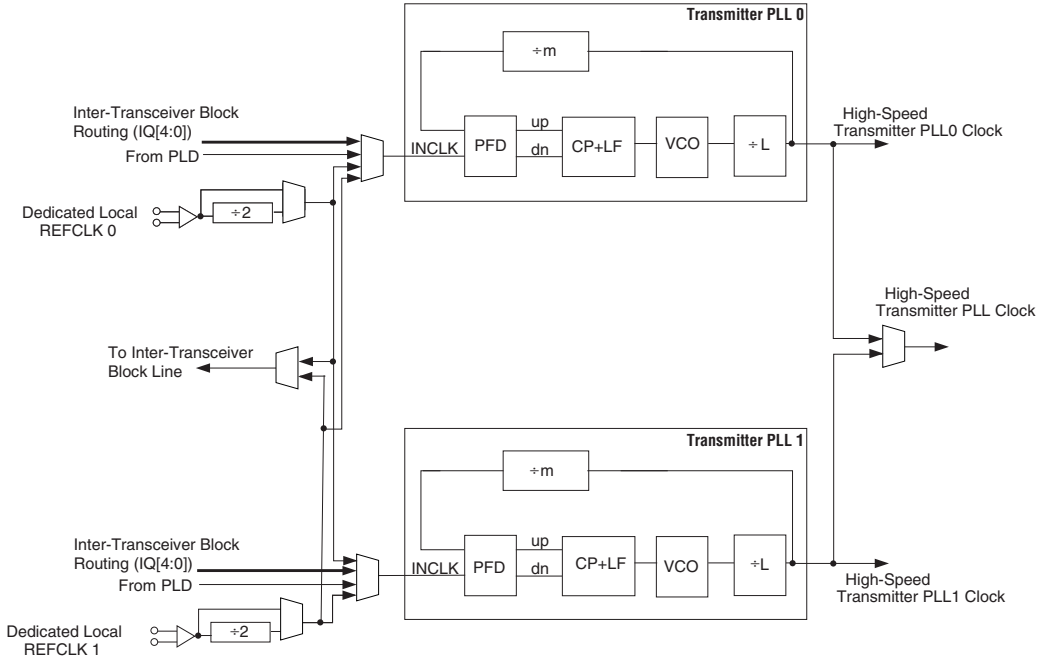
The transmitter PLLs in each transceiver block clock the PMA and PCS circuitry in the transmit path. The Quartus II software automatically powers down the transmitter PLLs that are not used in the design. Figure 2–4 is a block diagram of the transmitter PLL.

The transmitter phase/frequency detector references the clock from one of the following sources:

- Reference clocks
- Reference clock from the adjacent transceiver block
- Inter-transceiver block clock lines
- Global clock line driven by input pin

Two reference clocks, `REFCLK0` and `REFCLK1`, are available per transceiver block. The inter-transceiver block bus allows multiple transceivers to use the same reference clocks. Each transceiver block has one outgoing reference clock which connects to one inter-transceiver block line. The incoming reference clock can be selected from five inter-transceiver block lines `IQ[4..0]` or from the global clock line that is driven by an input pin.

Figure 2–4. Transmitter PLL Block Note (1)



Note to Figure 2–4:
(1) The global clock line must be driven by an input pin.

The transmitter PLLs support data rates up to 6.375 Gbps. The input clock frequency is limited to 622.08 MHz. An optional `pll_locked` port is available to indicate whether the transmitter PLL is locked to the reference clock. Both transmitter PLLs have a programmable loop bandwidth parameter that can be set to low, medium, or high. The loop bandwidth parameter can be statically set in the Quartus II software.

Table 2–2 lists the adjustable parameters in the transmitter PLL.

Table 2–2. Transmitter PLL Specifications

Parameter	Specifications
Input reference frequency range	50 MHz to 622.08 MHz
Data rate support	600 Mbps to 6.375 Gbps
Multiplication factor (W)	1, 4, 5, 8, 10, 16, 20, 25
Bandwidth	Low, medium, or high

Transmitter Phase Compensation FIFO Buffer

The transmitter phase compensation FIFO buffer resides in the transceiver block at the PCS/FPGA boundary and cannot be bypassed. This FIFO buffer compensates for phase differences between the transmitter PLL clock and the clock from the PLD. After the transmitter PLL has locked to the frequency and phase of the reference clock, the transmitter FIFO buffer must be reset to initialize the read and write pointers. After FIFO pointer initialization, the PLL must remain phase locked to the reference clock.

Byte Serializer

The FPGA and transceiver block must maintain the same throughput. If the FPGA interface cannot meet the timing margin to support the throughput of the transceiver, the byte serializer is used on the transmitter and the byte deserializer is used on the receiver.

The byte serializer takes words from the FPGA interface and converts them into smaller words for use in the transceiver. The transmit data path after the byte serializer is 8, 10, 16, or 20 bits. Refer to Table 2–3 for the transmitter data with the byte serializer enabled. The byte serializer can be bypassed when the data width is 8, 10, 16, or 20 bits at the FPGA interface.

Table 2–3. Transmitter Data with the Byte Serializer Enabled

Input Data Width	Output Data Width
16 bits	8 bits
20 bits	10 bits
32 bits	16 bits
40 bits	20 bits

If the byte serializer is disabled, the FPGA transmit data is passed without data width conversion.

Table 2–4 shows the data path configurations for the Stratix II GX device in single-width and double-width modes.

Refer to the section "8B/10B Encoder" on page 2–8 for a description of the single- and double-width modes.

Table 2–4. Data Path Configurations *Note (1)*

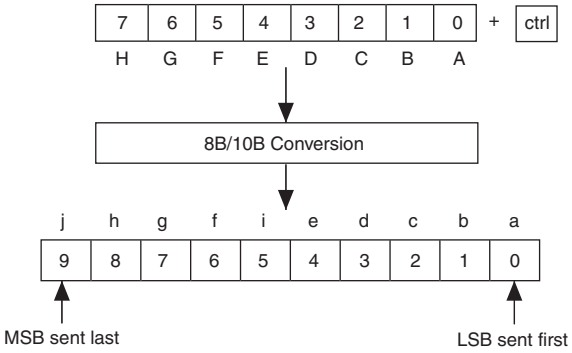
Parameter	Single-Width Mode		Double-Width Mode	
	Without Byte Serialization/ Deserialization	With Byte Serialization/ Deserialization	Without Byte Serialization/ Deserialization	With Byte Serialization/ Deserialization
Fabric to PCS data path width (bits)	8 or 10	16 or 20	16 or 20	32 or 40
Data rate range (Gbps)	0.6 to 2.5	0.6 to 3.125	1 to 5.0	1 to 6.375
PCS to PMA data path width (bits)	8 or 10	8 or 10	16 or 20	16 or 20
Byte ordering *(1)*		✓		✓
Data symbol A (MSB)				✓
Data symbol B		✓		✓
Data symbol C			✓	✓
Data symbol D (LSB)	✓	✓	✓	✓

Note to Table 2–4:
(1) Designs can use byte ordering when byte serialization and deserialization are used.

8B/10B Encoder

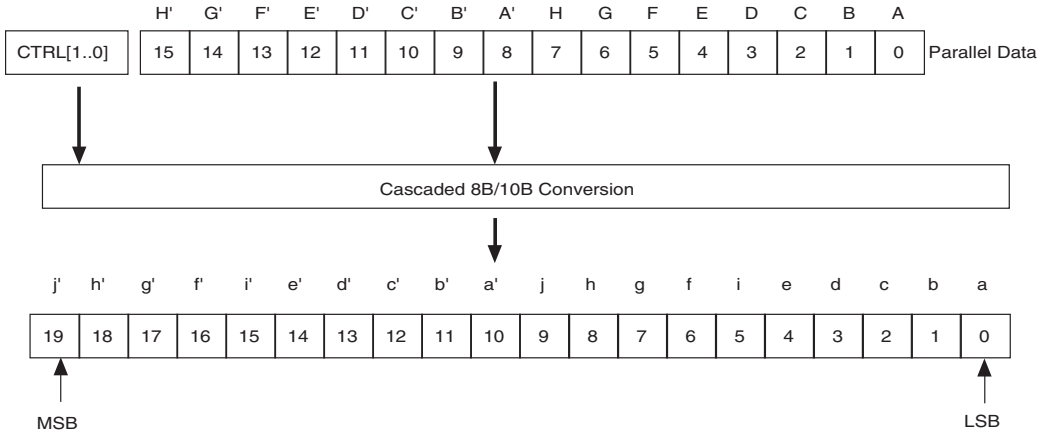
There are two different modes of operation for 8B/10B encoding. Single-width (8-bit) mode supports natural data rates from 622 Mbps to 3.125 Gbps. Double-width (16-bit cascaded) mode supports data rates above 3.125 Gbps. The encoded data has a maximum run length of five. The 8B/10B encoder can be bypassed. Figure 2–5 diagrams the 10-bit encoding process.

Figure 2–5. 8B/10B Encoding Process



In single-width mode, the 8B/10B encoder generates a 10-bit code group from the 8-bit data and 1-bit control identifier. In double-width mode, there are two 8B/10B encoders that are cascaded together and generate a 20-bit (2 × 10-bit) code group from the 16-bit (2 × 8-bit) data + 2-bit (2 × 1-bit) control identifier. Figure 2–6 shows the 20-bit encoding process. The 8B/10B encoder conforms to the IEEE 802.3 1998 edition standards.

Figure 2–6. 16-Bit to 20-Bit Encoding Process



Upon power on or reset, the 8B/10B encoder has a negative disparity which chooses the 10-bit code from the RD-column. However, the running disparity can be changed via the `tx_forcedisp` and `tx_dispval` ports.

Transmit State Machine

The transmit state machine operates in either PCI Express mode, XAUI mode, or GIGE mode, depending on the protocol used. The state machine is not utilized for certain protocols, such as SONET.

GIGE Mode

In GIGE mode, the transmit state machine converts all idle ordered sets (/K28.5/, /Dx.y/) to either /I1/ or /I2/ ordered sets. /I1/ consists of a negative-ending disparity /K28.5/ (denoted by /K28.5/-) followed by a neutral /D5.6/. /I2/ consists of a positive-ending disparity /K28.5/ (denoted by /K28.5/+) and a negative-ending disparity /D16.2/ (denoted by /D16.2/-). The transmit state machines do not convert any of the ordered sets to match /C1/ or /C2/, which are the configuration ordered sets. (/C1/ and /C2/ are defined by [/K28.5/, /D21.5/] and [/K28.5/, /D2.2/], respectively). Both the /I1/ and /I2/ ordered sets guarantee a negative-ending disparity after each ordered set.

XAUI Mode

The transmit state machine translates the XAUI XGMII code group to the XAUI PCS code group. Table 2–5 shows the code conversion.

Table 2–5. Code Conversion

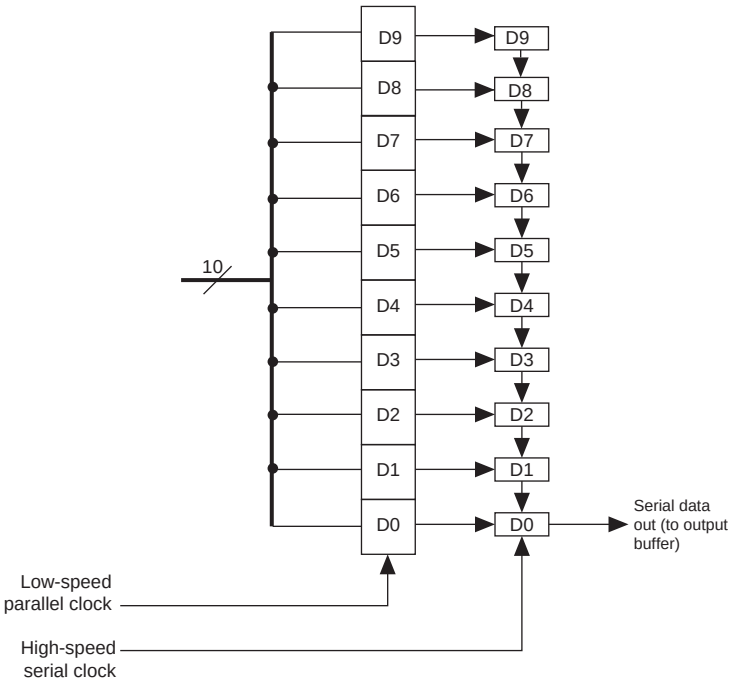
XGMII TXC	XGMII TXD	PCS Code-Group	Description
0	00 through FF	Dxx.y	Normal data
1	07	K28.0 or K28.3 or K28.5	Idle in ||**I**||
1	07	K28.5	Idle in ||**T**||
1	9C	K28.4	Sequence
1	FB	K27.7	Start
1	FD	K29.7	Terminate
1	FE	K30.7	Error
1	See IEEE 802.3 reserved code groups	See IEEE 802.3 reserved code groups	Reserved code groups
1	Other value	K30.7	Invalid XGMII character

The XAUI PCS idle code groups, /K28.0/ (/R/) and /K28.5/ (/K/), are automatically randomized based on a PRBS7 pattern with an $x^7 + x^6 + 1$ polynomial. The /K28.3/ (/A/) code group is automatically generated between 16 and 31 idle code groups. The idle randomization on the /A/, /K/, and /R/ code groups is done automatically by the transmit state machine.

Serializer (Parallel-to-Serial Converter)

The serializer converts the parallel 8, 10, 16, or 20-bit data into a serial data bit stream, transmitting the least significant bit (LSB) first. The serialized data stream is then fed to the high-speed differential transmit buffer. Figure 2–7 is a diagram of the serializer.

Figure 2–7. Serializer *Note (1)*



Note to Figure 2–7:
(1) This is a 10-bit serializer. The serializer can also convert 8, 16, and 20 bits of data.

Transmit Buffer

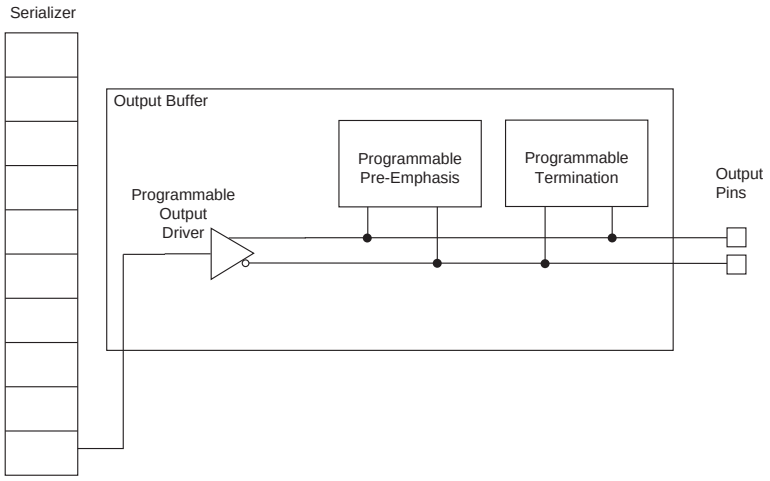
The Stratix II GX transceiver buffers support the 1.2- and 1.5-V PCML I/O standard at rates up to 6.375 Gbps. The common mode voltage (V_{CM}) of the output driver is programmable. The following V_{CM} values are available when the buffer is in 1.2- and 1.5-V PCML.

- V_{CM} = 0.6 V
- V_{CM} = 0.7 V

Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Handbook*.

The output buffer, as shown in Figure 2–8, is directly driven by the high-speed data serializer and consists of a programmable output driver, a programmable pre-emphasis circuit, a programmable termination, and a programmable V_{CM}.

Figure 2–8. Output Buffer



Programmable Output Driver

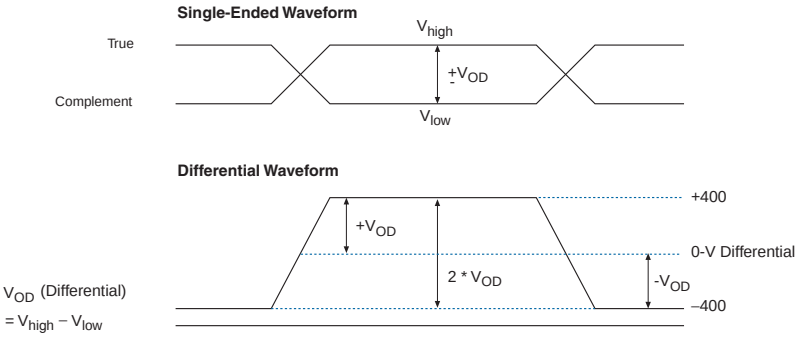
The programmable output driver can be set to drive out differentially 200 to 1,400 mV. The differential output voltage (V_{OD}) can be changed dynamically, or statically set by using the ALT2GXB megafunction or through I/O pins.

The output driver may be programmed with four different differential termination values:

- 100 Ω
- 120 Ω
- 150 Ω
- External termination

Differential signaling conventions are shown in Figure 2–9. The differential amplitude represents the value of the voltage between the true and complement signals. Peak-to-peak differential voltage is defined as $2 \times (V_{HIGH} - V_{LOW}) = 2 \times$ single-ended voltage swing. The common mode voltage is the average of V_{high} and V_{low}.

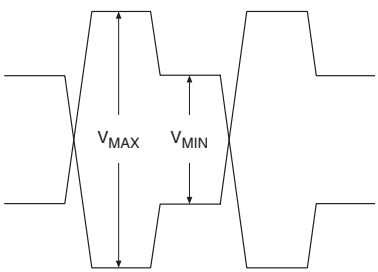
Figure 2–9. Differential Signaling



Programmable Pre-Emphasis

The programmable pre-emphasis module controls the output driver to boost the high frequency components, and compensate for losses in the transmission medium, as shown in Figure 2–10. The pre-emphasis is set statically using the ALT2GXB megafunction or dynamically through the dynamic reconfiguration controller.

Figure 2–10. Pre-Emphasis Signaling



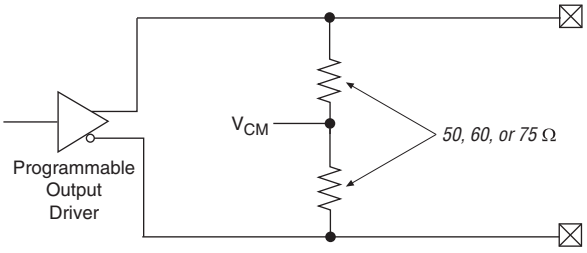
$$\text{Pre-Emphasis \%} = \left(\frac{V_{MAX}}{V_{MIN}} - 1 \right) \times 100$$

Pre-emphasis percentage is defined as ($V_{MAX}/V_{MIN} - 1$) × 100, where V_{MAX} is the differential emphasized voltage (peak-to-peak) and V_{MIN} is the differential steady-state voltage (peak-to-peak).

Programmable Termination

The programmable termination can be statically set in the Quartus II software. The values are 100 Ω, 120 Ω, 150 Ω, and external termination. Figure 2–11 shows the setup for programmable termination.

Figure 2–11. Programmable Transmitter Terminations



PCI Express Receiver Detect

The Stratix II GX transmitter buffer has a built-in receiver detection circuit for use in PIPE mode. This circuit provides the ability to detect if there is a receiver downstream by sending out a pulse on the channel and monitoring the reflection. This mode requires the transmitter buffer to be tri-stated (in electrical idle mode).

PCI Express Electric Idles (or Individual Transmitter Tri-State)

The Stratix II GX transmitter buffer supports PCI Express electrical idles. This feature is only active in PIPE mode. The `tx_forceelecidle` port puts the transmitter buffer in electrical idle mode. This port is available in all PCI Express power-down modes and has specific usage in each mode.

Receiver Path

This section describes the data path through the Stratix II GX receiver. The Stratix II GX receiver consists of the following blocks:

- Receiver differential input buffer
- Receiver PLL lock detector, signal detector, and run length checker
- Clock/data recovery (CRU) unit
- Deserializer
- Pattern detector
- Word aligner

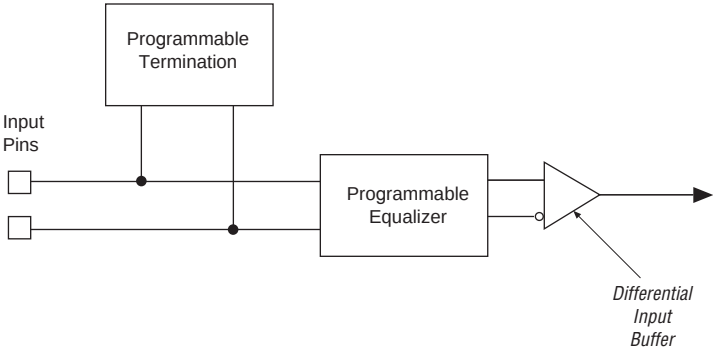
- Lane deskew
- Rate matcher
- 8B/10B decoder
- Byte deserializer
- Byte ordering
- Receiver phase compensation FIFO buffer

Receiver Input Buffer

The Stratix II GX receiver input buffer supports the 1.2-V and 1.5-V PCML I/O standard at rates up to 6.375 Gbps. The common mode voltage of the receiver input buffer is programmable between 0.85 V and 1.2 V. You must select the 0.85 V common mode voltage for AC- and DC-coupled PCML links and the 1.2 V common mode voltage for DC-coupled LVDS links.

The receiver has programmable on-chip 100-, 120-, or 150-Ω differential termination for different protocols, as shown in Figure 2–12. The receiver's internal termination can be disabled if external terminations and biasing are provided. The receiver and transmitter differential termination resistances can be set independently of each other.

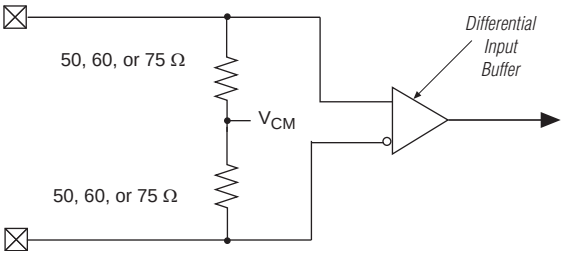
Figure 2–12. Receiver Input Buffer



Programmable Termination

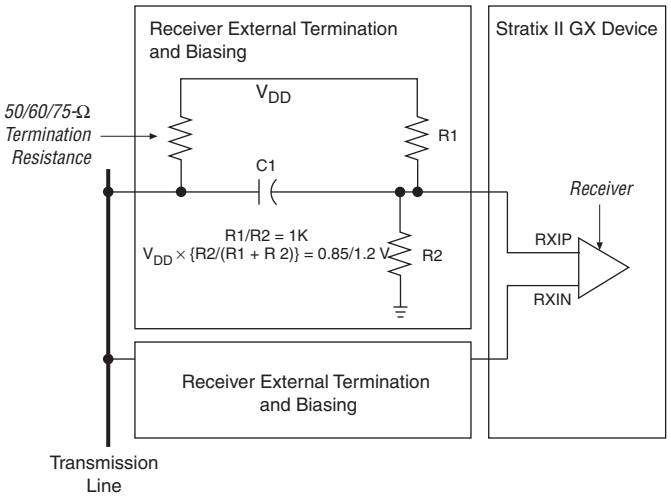
The programmable termination can be statically set in the Quartus II software. Figure 2–13 shows the setup for programmable receiver termination. The termination can be disabled if external termination is provided.

Figure 2–13. Programmable Receiver Termination



If a design uses external termination, the receiver must be externally terminated and biased to 0.85 V or 1.2 V. Figure 2–14 shows an example of an external termination and biasing circuit.

Figure 2–14. External Termination and Biasing Circuit



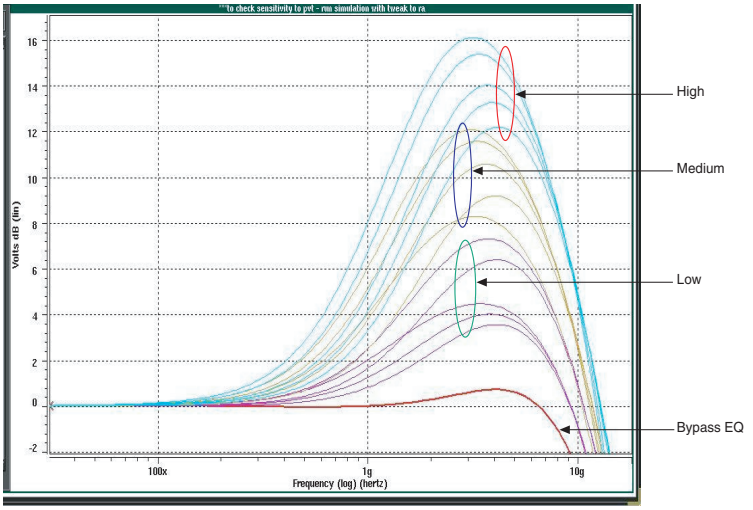
Programmable Equalizer

The Stratix II GX receivers provide a programmable receive equalization feature to compensate the effects of channel attenuation for high-speed signaling. PCB traces carrying these high-speed signals have low-pass filter characteristics. The impedance mismatch boundaries can also cause signal degradation. The equalization in the receiver diminishes the lossy attenuation effects of the PCB at high frequencies.

The Stratix II GX receivers also have adaptive equalization capability that adjusts the equalization levels to compensate for changing link characteristics. The adaptive equalization can be powered down dynamically after it selects the appropriate equalization levels.

The receiver equalization circuit is comprised of a programmable amplifier. Each stage is a peaking equalizer with a different center frequency and programmable gain. This allows varying amounts of gain to be applied, depending on the overall frequency response of the channel loss. Channel loss is defined as the summation of all losses through the PCB traces, vias, connectors, and cables present in the physical link. Figure 2–15 shows the frequency response for the 16 programmable settings allowed by the Quartus II software for Stratix II GX devices.

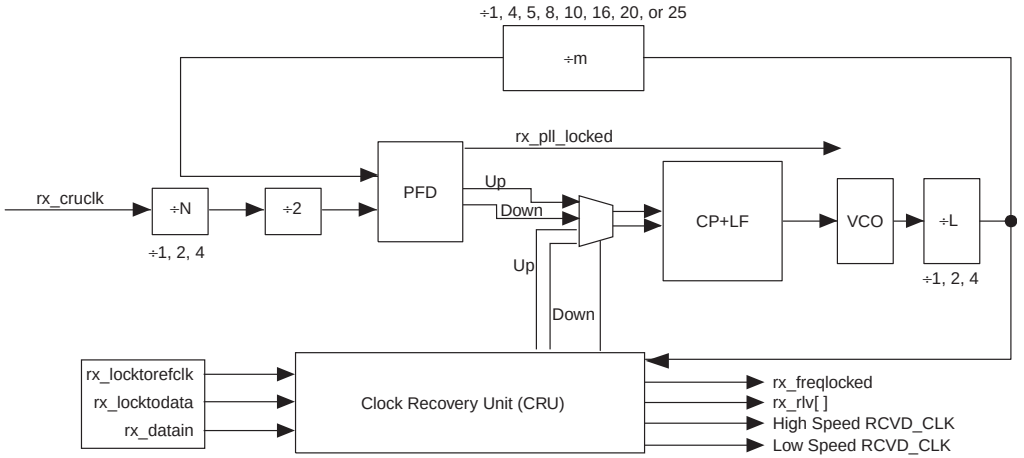
Figure 2–15. Frequency Response



Receiver PLL and CRU

Each transceiver block has four receiver PLLs, lock detectors, signal detectors, run length checkers, and CRU units, each of which is dedicated to a receive channel. If the receive channel associated with a particular receiver PLL or CRU is not used, the receiver PLL and CRU are powered down for the channel. Figure 2–16 shows the receiver PLL and CRU circuits.

Figure 2–16. Receiver PLL and CRU



The receiver PLLs and CRUs can support frequencies up to 6.375 Gbps. The input clock frequency is limited to the full clock range of 50 to 622 MHz but only when using `REFCLK0` or `REFCLK1`. An optional `RX_PLL_LOCKED` port is available to indicate whether the PLL is locked to the reference clock. The receiver PLL has a programmable loop bandwidth which can be set to low, medium, or high. The Quartus II software can statically set the loop bandwidth parameter.

All the parameters listed are programmable in the Quartus II software. The receiver PLL has the following features:

- Operates from 600 Mbps to 6.375 Gbps.
- Uses a reference clock between 50 MHz and 622.08 MHz.
- Programmable bandwidth settings: low, medium, and high.
- Programmable `rx_locktorefclk` (forces the receiver PLL to lock to the reference clock) and `rx_locktodata` (forces the receiver PLL to lock to the data).
- The voltage-controlled oscillator (VCO) operates at half rate and has two modes. These modes are for low or high frequency operation and provide optimized phase-noise performance.
- Programmable frequency multiplication *W* of 1, 4, 5, 8, 10, 16, 20, and 25. Not all settings are supported for any particular frequency.
- Two lock indication signals are provided. They are found in PFD mode (lock-to-reference clock), and PD (lock-to-data).

The CRU has a built-in switchover circuit to select whether the PLL VCO is aligned by the reference clock or the data. The optional port rx_freqlocked monitors when the CRU is in locked-to-data mode.

In the automatic mode, the CRU PLL must be within the prescribed PPM frequency threshold setting of the CRU reference clock for the CRU to switch from locked-to-reference to locked-to-data mode.

The automatic switchover circuit can be overridden by using the optional ports rx_locktorefclk and rx_locktodata. Table 2–6 shows the possible combinations of these two signals.

Table 2–6. Receiver Lock Combinations

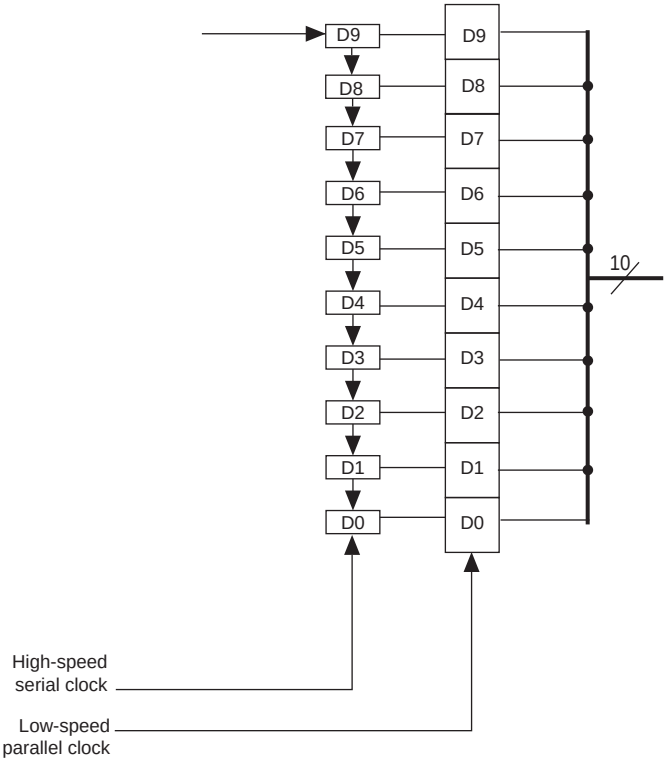
rx_locktodata	rx_locktorefclk	VCO (Lock to Mode)
0	0	Auto
0	1	Reference clock
1	x	Data

If the rx_locktorefclk and rx_locktodata ports are not used, the default is auto mode.

Deserializer (Serial-to-Parallel Converter)

The deserializer converts a serial bitstream into 8, 10, 16, or 20 bits of parallel data. The deserializer receives the LSB first. Figure 2–17 shows the deserializer.

Figure 2–17. Deserializer Note (1)



Note to Figure 2–17:
(1) This is a 10-bit deserializer. The deserializer can also convert 8, 16, or 20 bits of data.

Word Aligner

The deserializer block creates 8-, 10-, 16-, or 20-bit parallel data. The deserializer ignores protocol symbol boundaries when converting this data. Therefore, the boundaries of the transferred words are arbitrary. The word aligner aligns the incoming data based on specific byte or word boundaries. The word alignment module is clocked by the local receiver recovered clock during normal operation. All the data and programmed patterns are defined as big-endian (most significant word followed by least significant word). Most-significant-bit-first protocols such as SONET/SDH should reverse the bit order of word align patterns programmed.

This module detects word boundaries for the 8B/10B-based protocols, SONET, 16-bit, and 20-bit proprietary protocols. This module is also used to align to specific programmable patterns in PRBS7/23 test mode.

Pattern Detection

The programmable pattern detection logic can be programmed to align word boundaries using a single 7-, 8-, 10-, 16-, 20, or 32-bit pattern. The pattern detector can either do an exact match, or match the exact pattern and the complement of a given pattern. Once the programmed pattern is found, the data stream is aligned to have the pattern on the LSB portion of the data output bus.

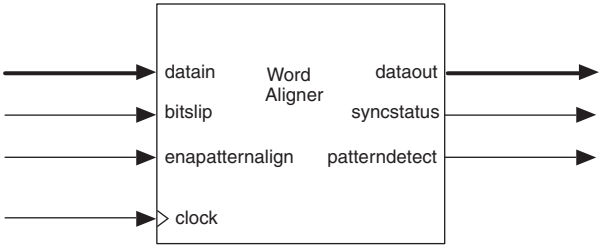
XAUI, GIGE, PCI Express, and Serial RapidIO standards have embedded state machines for symbol boundary synchronization. These standards use K28.5 as their 10-bit programmed comma pattern. Each of these standards uses different algorithms before signaling symbol boundary acquisition to the FPGA.

The pattern detection logic searches from the LSB to the most significant bit (MSB). If multiple patterns are found within the search window, the pattern in the lower portion of the data stream (corresponding to the pattern received earlier) is aligned and the rest of the matching patterns are ignored.

Once a pattern is detected and the data bus is aligned, the word boundary is locked. The two detection status signals (`rx_syncstatus` and `rx_patterndetect`) indicate that an alignment is complete.

Figure 2–18 is a block diagram of the word aligner.

Figure 2–18. Word Aligner



Control and Status Signals

The rx_enapatternalign signal is the FPGA control signal that enables word alignment in non-automatic modes. The rx_enapatternalign signal is not used in automatic modes (PCI Express, XAUI, GIGE, CPRI, and Serial RapidIO).

In manual alignment mode, after the rx_enapatternalign signal is activated, the rx_syncstatus signal goes high for one parallel clock cycle to indicate that the alignment pattern has been detected and the word boundary has been locked. If the rx_enapatternalign is deactivated, the rx_syncstatus signal acts as a re-synchronization signal to signify that the alignment pattern has been detected but not locked on a different word boundary.

When using the synchronization state machine, the rx_syncstatus signal indicates the link status. If the rx_syncstatus signal is high, link synchronization is achieved. If the rx_syncstatus signal is low, synchronization has not yet been achieved, or there were enough code group errors to lose synchronization.

In some modes, the rx_enapatternalign signal can be configured to operate as a rising edge signal.

For more information on manual alignment modes, refer to the *Stratix II GX Device Handbook*, volume 2.

When the rx_enapatternalign signal is sensitive to the rising edge, each rising edge triggers a new boundary alignment search, clearing the rx_syncstatus signal.

The rx_patterndetect signal pulses high during a new alignment, and also whenever the alignment pattern occurs on the current word boundary.

SONET/SDH

In all the SONET/SDH modes, you can configure the word aligner to either align to A1A2 or A1A1A2A2 patterns. Once the pattern is found, the word boundary is aligned and the word aligner asserts the rx_patterndetect signal for one clock cycle.

Programmable Run Length Violation

The word aligner supports a programmable run length violation counter. Whenever the number of the continuous '0' (or '1') exceeds a user programmable value, the `rx_rlv` signal goes high for a minimum pulse width of two recovered clock cycles. The maximum run values supported are shown in Table 2–7.

Table 2–7. Maximum Run Length (UI)

Mode	PMA Serialization			
	8 Bit	10 Bit	16 Bit	20 Bit
Single-Width	128	160	—	—
Double-Width	—	—	512	640

Running Disparity Check

The running disparity error `rx_disperr` and running disparity value `rx_runningdisp` are sent along with aligned data from the 8B/10B decoder to the FPGA. You can ignore or act on the reported running disparity value and running disparity error signals.

Bit-Slip Mode

The word aligner can operate in either pattern detection mode or in bit-slip mode.

The bit-slip mode provides the option to manually shift the word boundary through the FPGA. This feature is useful for:

- Longer synchronization patterns than the pattern detector can accommodate
- Scrambled data stream
- Input stream consisting of over-sampled data

This feature can be applied at 10-bit and 16-bit data widths.

The word aligner outputs a word boundary as it is received from the analog receiver after reset. You can examine the word and search its boundary in the FPGA. To do so, assert the `rx_bitslip` signal. The `rx_bitslip` signal should be toggled and held constant for at least two FPGA clock cycles.

For every rising edge of the `rx_bitslip` signal, the current word boundary is slipped by one bit. Every time a bit is slipped, the bit received earliest is lost. If bit slipping shifts a complete round of bus width, the word boundary is back to the original boundary.

The rx_syncstatus signal is not available in bit-slipping mode.

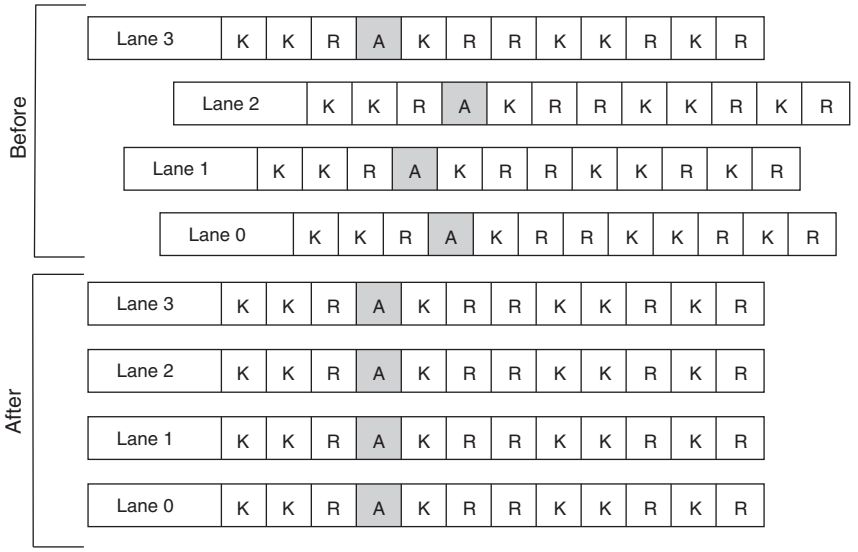
Channel Aligner

The channel aligner is available only in XAUI mode and aligns the signals of all four channels within a transceiver. The channel aligner follows the IEEE 802.3ae, clause 48 specification for channel bonding.

The channel aligner is a 16-word FIFO buffer with a state machine controlling the channel bonding process. The state machine looks for an /A/ (/K28.3/) in each channel, and aligns all the /A/ code groups in the transceiver. When four columns of /A/ (denoted by //A//) are detected, the rx_channelaligned signal goes high, signifying that all the channels in the transceiver have been aligned. The reception of four consecutive misaligned /A/ code groups restarts the channel alignment sequence and sends the rx_channelaligned signal low.

Figure 2–19 shows misaligned channels before the channel aligner and the aligned channels after the channel aligner.

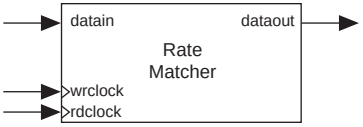
Figure 2–19. Before and After the Channel Aligner



Rate Matcher

Rate matcher is available in Basic, PCI Express, XAUI, and GIGE modes and consists of a 20-word deep FIFO buffer and a FIFO controller. Figure 2–20 shows the implementation of the rate matcher in the Stratix II GX device.

Figure 2–20. Rate Matcher



In a multi-crystal environment, the rate matcher compensates for up to a ± 300-PPM difference between the source and receiver clocks. Table 2–8 shows the standards supported and the PPM for the rate matcher tolerance.

Table 2–8. Rate Matcher PPM Support *Note (1)*

Standard	PPM
XAUI	± 100
PCI Express (PIPE)	± 300
GIGE	± 100
Basic Double-Width	± 300

Note to Table 2–8:

(1) Refer to the *Stratix II GX Transceiver User Guide* for the Altera®-defined scheme.

Basic Mode

In Basic mode, you can program the skip and control pattern for rate matching. In single-width Basic mode, there is no restriction on the deletion of a skip character in a cluster. The rate matcher deletes the skip characters as long as they are available. For insertion, the rate matcher inserts skip characters such that the number of skip characters at the output of rate matcher does not exceed five. In double-width mode, the rate matcher deletes skip character when they appear as pairs in the upper and lower bytes. There are no restrictions on the number of skip characters that are deleted. The rate matcher inserts skip characters as pairs.

GIGE Mode

In GIGE mode, the rate matcher adheres to the specifications in clause 36 of the IEEE 802.3 documentation for idle additions or removals. The rate matcher performs clock compensation only on /I2/ ordered sets, composed of a /K28.5/+ followed by a /D16.2/-. The rate matcher does not perform clock compensation on any other ordered set combinations. An /I2/ is added or deleted automatically based on the number of words in the FIFO buffer. A K28.4 is given at the control and data ports when the FIFO buffer is in an overflow or underflow condition.

XAUI Mode

In XAUI mode, the rate matcher adheres to clause 48 of the IEEE 802.3ae specification for clock rate compensation. The rate matcher performs clock compensation on columns of /R/ (/K28.0/), denoted by //R//. An //R// is added or deleted automatically based on the number of words in the FIFO buffer.

PCI Express Mode

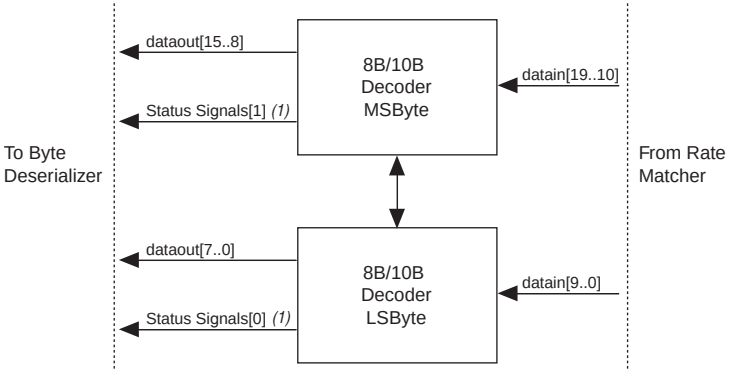
PCI Express mode operates at a data rate of 2.5 Gbps, and supports lane widths of ×1, ×2, ×4, and ×8. The rate matcher can support up to ± 300-PPM differences between the upstream transmitter and the receiver. The rate matcher looks for the skip ordered sets (SOS), which usually consist of a /K28.5/ comma followed by three /K28.0/ skip characters. The rate matcher deletes or inserts skip characters when necessary to prevent the rate matching FIFO buffer from overflowing or underflowing.

The Stratix II GX rate matcher in PCI Express mode has FIFO overflow and underflow protection. In the event of a FIFO overflow, the rate matcher deletes any data after the overflow condition to prevent FIFO pointer corruption until the rate matcher is not full. In an underflow condition, the rate matcher inserts 9'h1FE (/K30.7/) until the FIFO is not empty. These measures ensure that the FIFO can gracefully exit the overflow and underflow condition without requiring a FIFO reset.

8B/10B Decoder

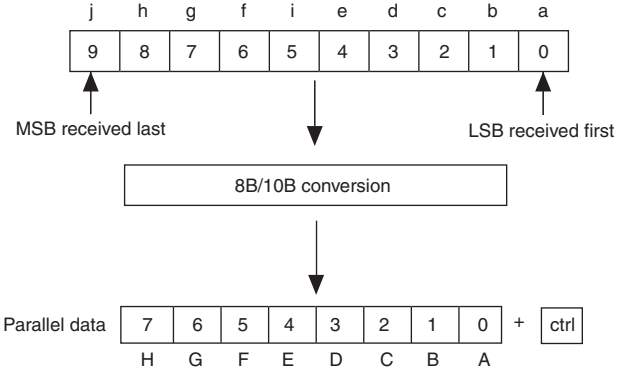
The 8B/10B decoder (Figure 2–21) is part of the Stratix II GX transceiver digital blocks (PCS) and lies in the receiver path between the rate matcher and the byte deserializer blocks. The 8B/10B decoder operates in single-width and double-width modes, and can be bypassed if the 8B/10B decoding is not necessary. In single-width mode, the 8B/10B decoder restores the 8-bit data + 1-bit control identifier from the 10-bit code. In double-width mode, there are two 8B/10B decoders in parallel, which restores the 16-bit (2 × 8-bit) data + 2-bit (2 × 1-bit) control identifier from the 20-bit (2 × 10-bit) code. This 8B/10B decoder conforms to the IEEE 802.3 1998 edition standards.

Figure 2–21. 8B/10B Decoder



The 8B/10B decoder in single-width mode translates the 10-bit encoded data into the 8-bit equivalent data or control code. The 10-bit code received must be from the supported Dx.y or Kx.y list with the proper disparity or error flags asserted. All 8B/10B control signals, such as disparity error or control detect, are pipelined with the data and edge-aligned with the data. Figure 2–22 shows how the 10-bit symbol is decoded in the 8-bit data + 1-bit control indicator.

Figure 2–22. 8B/10B Decoder Conversion

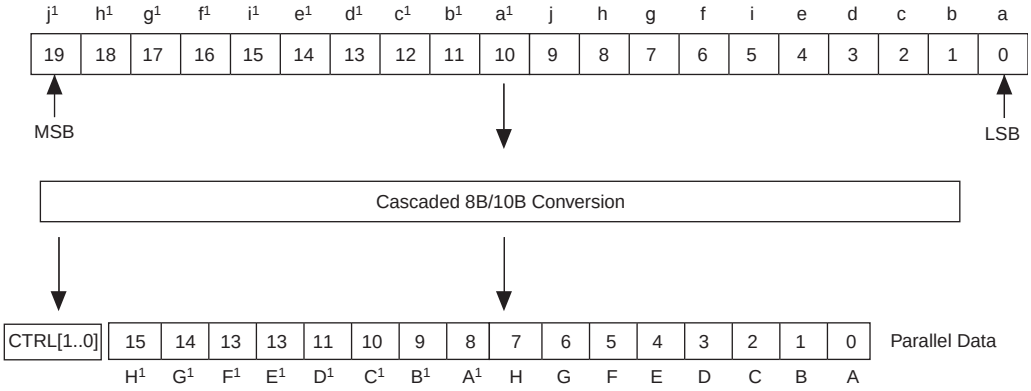


The 8B/10B decoder in double-width mode translates the 20-bit (2 × 10-bits) encoded code into the 16-bit (2 × 8-bits) equivalent data or control code. The 20-bit upper and lower symbols received must be from the supported Dx.y or Kx.y list with the proper disparity or error flags

asserted. All 8B/10B control signals, such as disparity error or control detect, are pipelined with the data in the Stratix II GX receiver block and are edge aligned with the data.

Figure 2–23 shows how the 20-bit code is decoded to the 16-bit data + 2-bit control indicator.

Figure 2–23. 20-Bit to 16-Bit Decoding Process



There are two optional error status ports available in the 8B/10B decoder, `rx_errdetect` and `rx_disperr`. These status signals are aligned with the code group in which the error occurred.

Receiver State Machine

The receiver state machine operates in Basic, GIGE, PCI Express, and XAUI modes. In GIGE mode, the receiver state machine replaces invalid code groups with K30.7. In XAUI mode, the receiver state machine translates the XAUI PCS code group to the XAUI XGMII code group.

Byte Deserializer

The byte deserializer widens the transceiver data path before the FPGA interface. This reduces the rate at which the received data needs to be clocked at in the FPGA logic. The byte deserializer block is available in both single- and double-width modes.

The byte deserializer converts the one- or two-byte interface into a two- or four-byte-wide data path from the transceiver to the FPGA logic (see Table 2–9). The FPGA interface has a limit of 250 MHz, so the byte deserializer is needed to widen the bus width at the FPGA interface and

reduce the interface speed. For example, at 6.375 Gbps, the transceiver logic has a double-byte-wide data path that runs at 318.75 MHz in a ×20 deserializer factor, which is above the maximum FPGA interface speed. When using the byte deserializer, the FPGA interface width doubles to 40-bits (36-bits when using the 8B/10B encoder) and the interface speed reduces to 159.375 MHz.

Table 2–9. Byte Deserializer Input and Output Widths

Input Data Width (Bits)	Deserialized Output Data Width to the FPGA (Bits)
20	40
16	32
10	20
8	16

Byte Ordering Block

The byte ordering block shifts the byte order. A pre-programmed byte in the input data stream is detected and placed in the least significant byte of the output stream. Subsequent bytes start appearing in the byte positions following the LSB. The byte ordering block inserts the programmed PAD characters to shift the byte order pattern to the LSB.

Based on the setting in the MegaWizard® Plug-In Manager, the byte ordering block can be enabled either by the `rx_syncstatus` signal or by the `rx_enabyteord` signal from the PLD. When the `rx_syncstatus` signal is used as enable, the byte ordering block reorders the data only for the first occurrence of the byte order pattern that is received after word alignment is completed. You must assert `rx_digitalreset` to perform byte ordering again. However, when the byte ordering block is controlled by `rx_enabyteord`, the byte ordering block can be controlled by the PLD logic dynamically. When you create your functional mode in the MegaWizard, you can select byte ordering block only if rate matcher is not selected.

Receiver Phase Compensation FIFO Buffer

The receiver phase compensation FIFO buffer resides in the transceiver block at the FPGA boundary and cannot be bypassed. This FIFO buffer compensates for phase differences and clock tree timing skew between the receiver clock domain within the transceiver and the receiver FPGA clock after it has transferred to the FPGA.

When the FIFO pointers initialize, the receiver domain clock must remain phase locked to receiver FPGA clock.

After resetting the receiver FIFO buffer, writing to the receiver FIFO buffer begins and continues on each parallel clock. The phase compensation FIFO buffer is eight words deep for PIPE mode and four words deep for all other modes.

Loopback Modes

The Stratix II GX transceiver has built-in loopback modes for debugging and testing. The loopback modes are configured in the Stratix II GX ALT2GXB megafunction in the Quartus II software. The available loopback modes are:

- Serial loopback
- Parallel loopback
- Reverse serial loopback
- Reverse serial loopback (pre-CDR)
- PCI Express PIPE reverse parallel loopback (available only in PIPE mode)

Serial Loopback

The serial loopback mode exercises all the transceiver logic, except for the input buffer. Serial loopback is available for all non-PIPE modes. The loopback function is dynamically enabled through the `rx_seriallpbken` port on a channel-by-channel basis.

In serial loopback mode, the data on the transmit side is sent by the PLD. A separate mode is available in the ALT2GXB megafunction under Basic protocol mode, in which PRBS data is generated and verified internally in the transceiver. The PRBS patterns available in this mode are shown in Table 2–10.

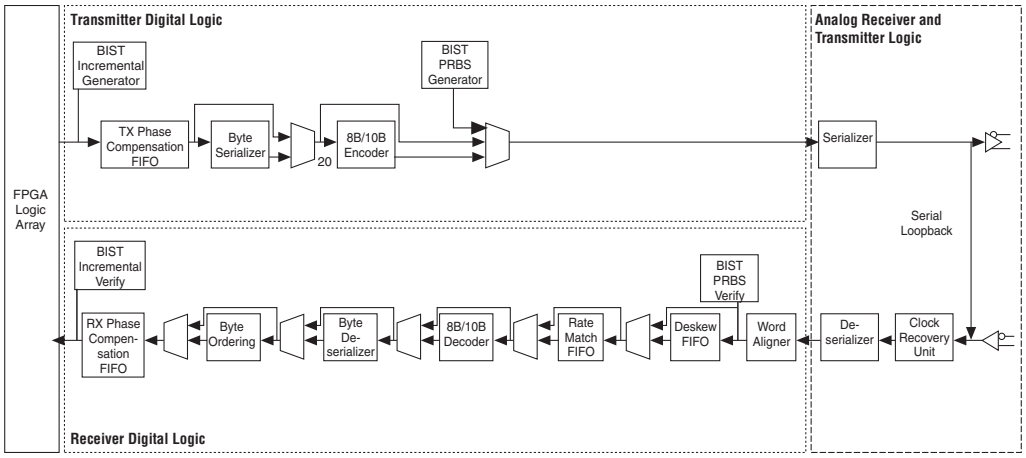
Table 2–10 shows the BIST data output and verifier alignment pattern.

Table 2–10. BIST Data Output and Verifier Alignment Pattern

Pattern	Polynomial	Parallel Data Width			
		8-Bit	10-Bit	16-Bit	20-Bit
PRBS-7	×7 + ×6 + 1				✓
PRBS-10	×10 + ×7 + 1		✓		

Figure 2–24 shows the data path in serial loopback mode.

Figure 2–24. Stratix II GX Block in Serial Loopback Mode with BIST and PRBS



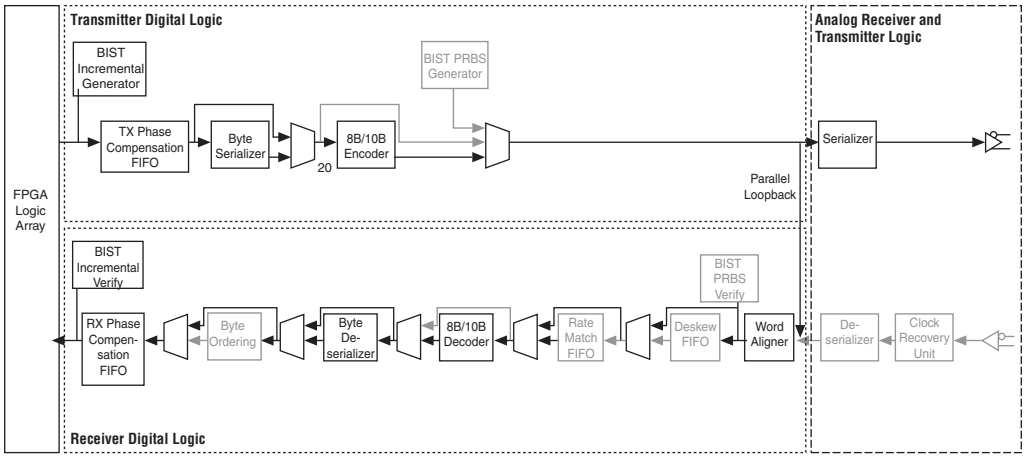
Parallel Loopback

The parallel loopback mode exercises the digital logic portion of the transceiver data path. The analog portions are not used in this loopback path, and the received high-speed serial data is not retimed. This protocol is available as one of the sub-protocols under Basic mode and can be used only for Basic double-width mode.

In this loopback mode, the data from the internally available BIST generator is transmitted. The data is looped back after the end of PCS and before the PMA. On the receive side, an internal BIST verifier checks for errors. This loopback enables you to verify the PCS block.

Figure 2–25 shows the data path in parallel loopback mode.

Figure 2–25. Stratix II GX Block in Parallel Loopback Mode

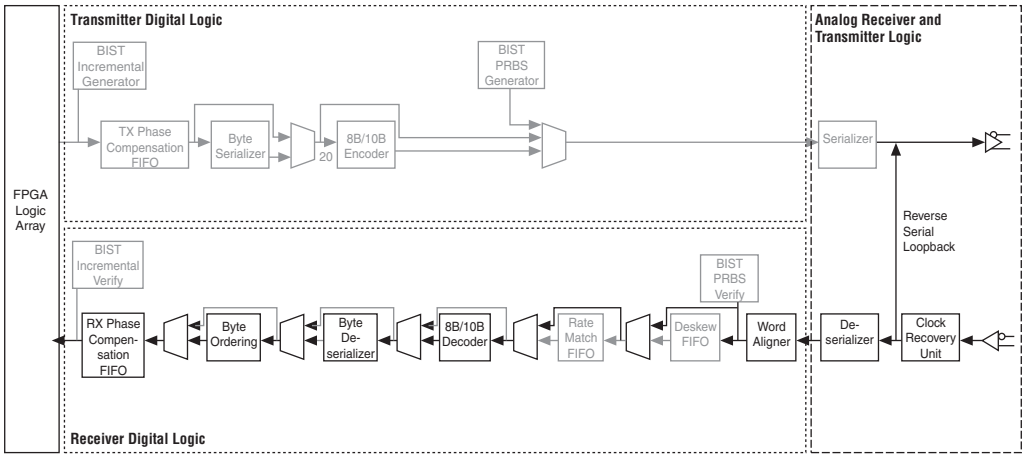


Reverse Serial Loopback

The reverse serial loopback mode uses the analog portion of the transceiver. An external source (pattern generator or transceiver) generates the source data. The high-speed serial source data arrives at the high-speed differential receiver input buffer, passes through the CRU unit, and the retimed serial data is looped back and transmitted though the high-speed differential transmitter output buffer.

Figure 2–26 shows the data path in reverse serial loopback mode.

Figure 2–26. Stratix II GX Block in Reverse Serial Loopback Mode

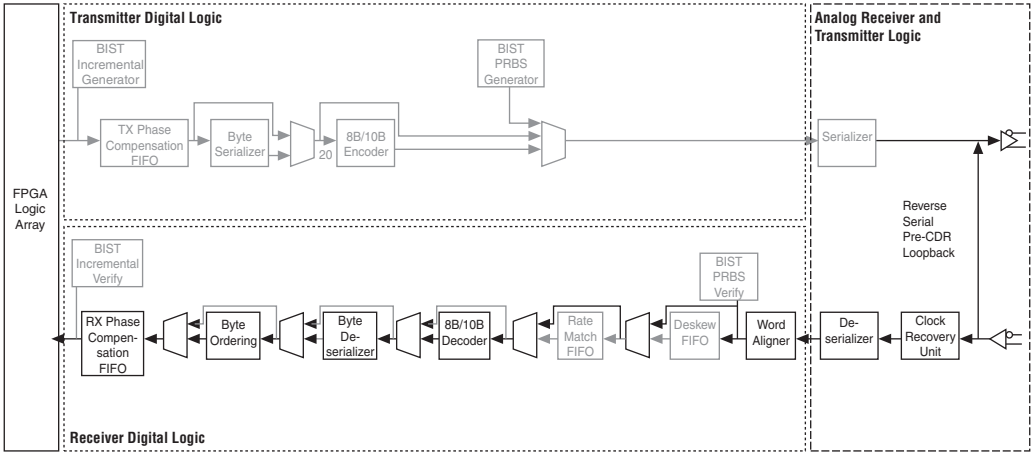


Reverse Serial Pre-CDR Loopback

The reverse serial pre-CDR loopback mode uses the analog portion of the transceiver. An external source (pattern generator or transceiver) generates the source data. The high-speed serial source data arrives at the high-speed differential receiver input buffer, loops back before the CRU unit, and is transmitted though the high-speed differential transmitter output buffer. It is for test or verification use only to verify the signal being received after the gain and equalization improvements of the input buffer. The signal at the output is not exactly what is received since the signal goes through the output buffer and the VOD is changed to the VOD setting level. The pre-emphasis settings have no effect.

Figure 2–27 show the Stratix II GX block in reverse serial pre-CDR loopback mode.

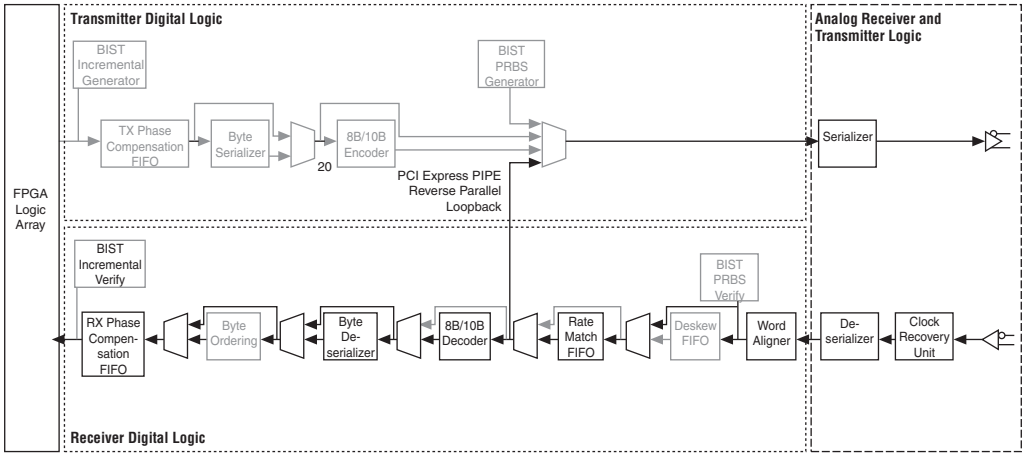
Figure 2–27. Stratix II GX Block in Reverse Serial Pre-CDR Loopback Mode



PCI Express PIPE Reverse Parallel Loopback

This loopback mode, available only in PIPE mode, can be dynamically enabled by the `tx_detectrxloopback` port of the PIPE interface. Figure 2–28 shows the datapath for this mode.

Figure 2–28. Stratix II GX Block in PCI Express PIPE Reverse Parallel Loopback Mode

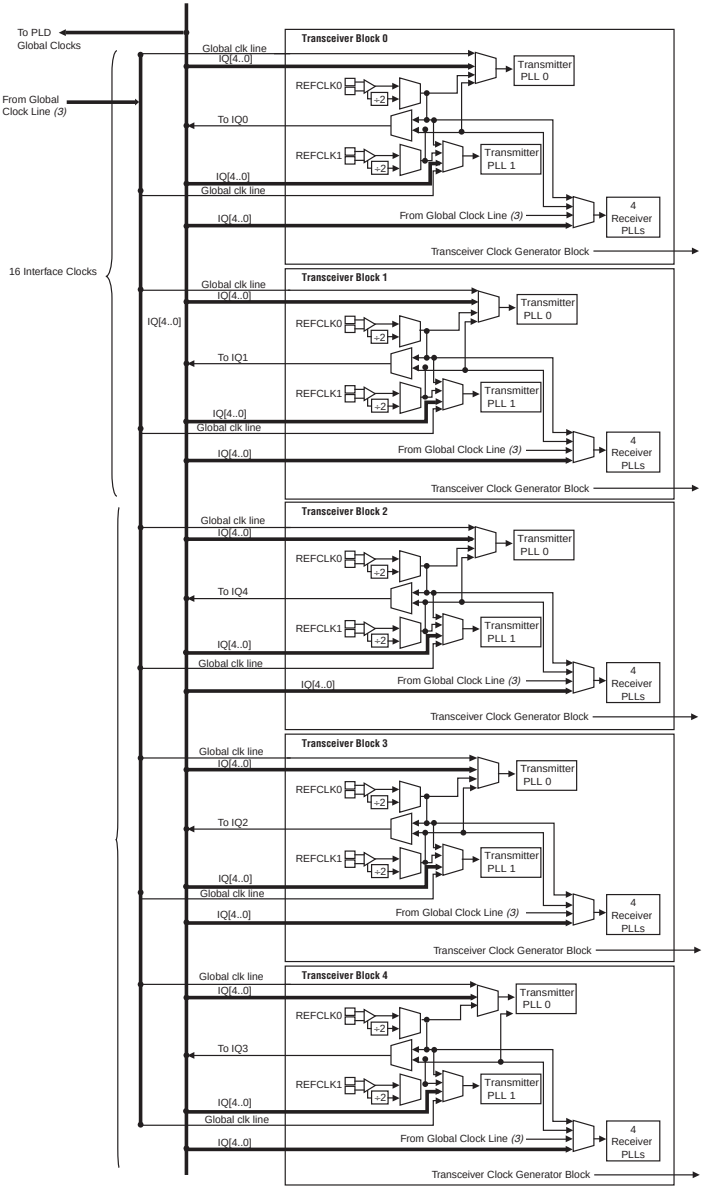


Transceiver Clocking

Each Stratix II GX device transceiver block contains two transmitter PLLs and four receiver PLLs. These PLLs can be driven by either of the two reference clocks per transceiver block. These REFCLK signals can drive all global clocks, transmitter PLL inputs, and all receiver PLL inputs. Subsequently, the transmitter PLL output can only drive global clock lines and the receiver PLL reference clock port. Only one of the two reference clocks in a quad can drive the Inter Quad (I/Q) lines to clock the PLLs in the other quads.

Figure 2–29 shows the inter-transceiver line connections as well as the global clock connections for the EP2SGX130 device.

Figure 2–29. EP2SGX130 Device Inter-Transceiver and Global Clock Connections

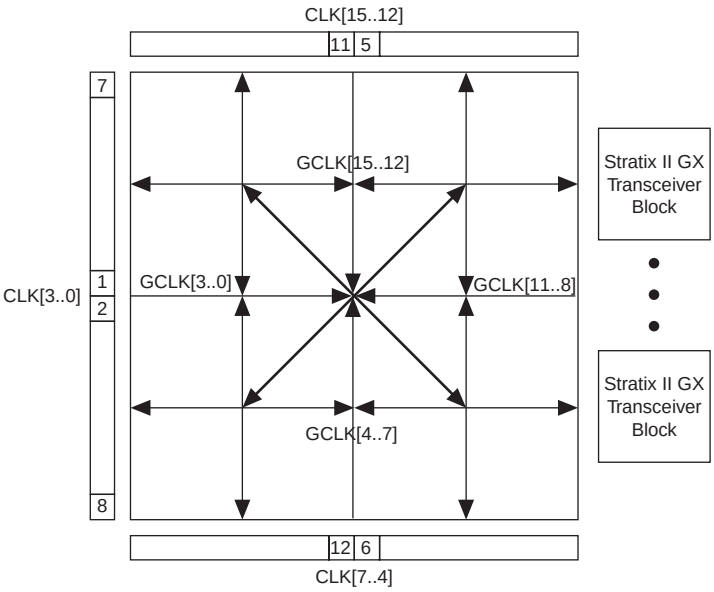


Notes to Figure 2–29:

(1) There are two transmitter PLLs in each transceiver block.
(2) There are four receiver PLLs in each transceiver block.
(3) The Global Clock line must be driven by an input pin.

The receiver PLL can also drive the regional clocks and regional routing adjacent to the associated transceiver block. Figure 2–30 shows which global clock resource can be used by the recovered clock. Figure 2–31 shows which regional clock resource can be used by the recovered clock.

Figure 2–30. Stratix II GX Receiver PLL Recovered Clock to Global Clock Connection *Notes (1), (2)*

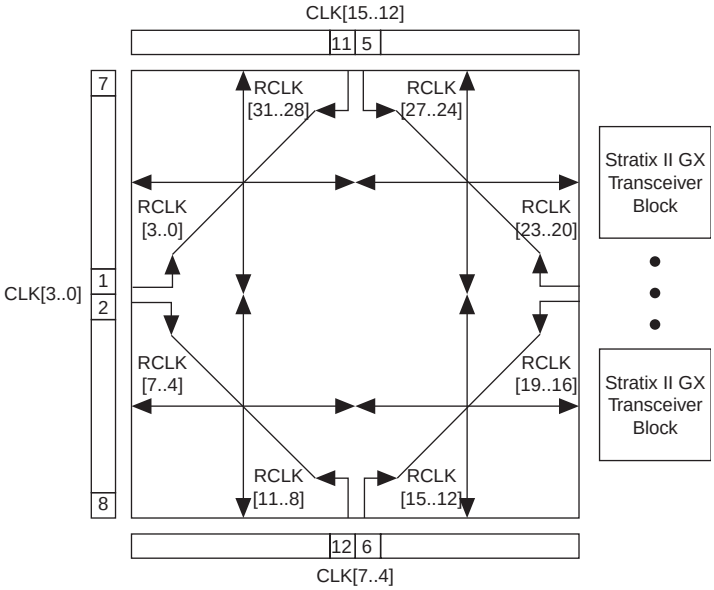


Notes to Figure 2–30:

(1) CLK# pins are clock pins and their associated number. These are pins for global and regional clocks.

(2) GCLK# pins are global clock pins.

Figure 2–31. Stratix II GX Receiver PLL Recovered Clock to Regional Clock Connection *Notes (1), (2)*



Notes to Figure 2–31:

(1) CLK# pins are clock pins and their associated number. These are pins for global and local clocks.

(2) RCLK# pins are regional clock pins.

Table 2–11 summarizes the possible clocking connections for the transceivers.

Table 2–11. Available Clocking Connections for Transceivers

Source	Destination				
	Transmitter PLL	Receiver PLL	Global Clock	Regional Clock	Inter-Transceiver Lines
REFCLK[1..0]	✓	✓	✓	✓	✓
Transmitter PLL			✓	✓	
Receiver PLL			✓	✓	
Global clock (driven from an input pin)	✓	✓			
Inter-transceiver lines	✓	✓			

Clock Resource for PLD-Transceiver Interface

For the regional or global clock network to route into the transceiver, a local route input output (LRIO) channel is required. Each LRIO clock region has up to eight clock paths and each transceiver block has a maximum of eight clock paths for connecting with LRIO clocks. These resources are limited and determine the number of clocks that can be used between the PLD and transceiver blocks. Table 2–12 shows the number of LRIO resources available for Stratix II GX devices with different numbers of transceiver blocks.

Tables 2–12 through 2–15 show the connection of the LRIO clock resource to the transceiver block.

Table 2–12. Available Clocking Connections for Transceivers in 2SGX30D

Region	Clock Resource		Transceiver	
	Global Clock	Regional Clock	Bank 13 8 Clock I/O	Bank 14 8 Clock I/O
Region0 8 LRIO clock	✓	RCLK 20-27	✓	
Region1 8 LRIO clock	✓	RCLK 12-19		✓

Table 2–13. Available Clocking Connections for Transceivers in 2SGX60E

Region	Clock Resource		Transceiver		
	Global Clock	Regional Clock	Bank 13 8 Clock I/O	Bank 14 8 Clock I/O	Bank 15 8 Clock I/O
Region0 8 LRIO clock	✓	RCLK 20-27	✓		
Region1 8 LRIO clock	✓	RCLK 20-27	✓	✓	
Region2 8 LRIO clock	✓	RCLK 12-19		✓	✓
Region3 8 LRIO clock	✓	RCLK 12-19			✓

Table 2–14. Available Clocking Connections for Transceivers in 2SGX90F

Region	Clock Resource		Transceiver			
	Global Clock	Regional Clock	Bank 13 8 Clock I/O	Bank 14 8 Clock I/O	Bank 15 8 clock I/O	Bank 16 8 Clock I/O
Region0 8 LRIO clock	✓	RCLK 20-27	✓			
Region1 8 LRIO clock	✓	RCLK 20-27		✓		
Region2 8 LRIO clock	✓	RCLK 12-19			✓	
Region3 8 LRIO clock	✓	RCLK 12-19				✓

.

Table 2–15. Available Clocking Connections for Transceivers in 2SGX130G

Region	Clock Resource		Transceiver				
	Global Clock	Regional Clock	Bank 13 8 Clock I/O	Bank 14 8 Clock I/O	Bank 15 8 clock I/O	Bank 16 8 Clock I/O	Bank 17 8 Clock I/O
Region0 8 LRIO clock	✓	RCLK 20-27	✓				
Region1 8 LRIO clock	✓	RCLK 20-27		✓			
Region2 8 LRIO clock	✓	RCLK 12-19			✓	✓	
Region3 8 LRIO clock	✓	RCLK 12-19				✓	✓

Other Transceiver Features

Other important features of the Stratix II GX transceivers are the power down and reset capabilities, external voltage reference and bias circuitry, and hot swapping.

Calibration Block

The Stratix II GX device uses the calibration block to calibrate the on-chip termination for the PLLs and their associated output buffers and the terminating resistors on the transceivers. The calibration block counters the effects of process, voltage, and temperature (PVT). The calibration block references a derived voltage across an external reference resistor to calibrate the on-chip termination resistors on the Stratix II GX device. The calibration block can be powered down. However, powering down the calibration block during operations may yield transmit and receive data errors.

Dynamic Reconfiguration

This feature allows you to dynamically reconfigure the PMA portion and the channel parameters, such as data rate and functional mode, of the Stratix II GX transceiver. The PMA reconfiguration allows you to quickly optimize the settings for the transceiver's PMA to achieve the intended bit error rate (BER).

The dynamic reconfiguration block can dynamically reconfigure the following PMA settings:

- Pre-emphasis settings
- Equalizer and DC gain settings
- Voltage Output Differential (V_{OD}) settings

The channel reconfiguration allows you to dynamically modify the data rate, local dividers, and the functional mode of the transceiver channel.

Refer to the *Stratix II GX Device Handbook*, volume 2, for more information.

The dynamic reconfiguration block requires an input clock between 2.5 MHz and 50 MHz. The clock for the dynamic reconfiguration block is derived from a high-speed clock and divided down using a counter.

Individual Power Down and Reset for the Transmitter and Receiver

Stratix II GX transceivers offer a power saving advantage with their ability to shut off functions that are not needed. The device can individually reset the receiver and transmitter blocks and the PLLs. The Stratix II GX device can either globally or individually power down and reset the transceiver. Table 2–16 shows the connectivity between the reset signals and the Stratix II GX transceiver blocks. These reset signals can be controlled from the FPGA or pins.

Table 2–16. Reset Signal Map to Stratix II GX Blocks

Reset Signal	Transmitter Phase Compensation FIFO Module/ Byte Serializer	Transmitter 8B/10B Encoder	Transmitter Serializer	Transmitter Analog Circuits	Transmitter PLL	Transmitter XAUI State Machine	BIST Generators	Receiver Deserializer	Receiver Word Aligner	Receiver Deskew FIFO Module	Receiver Rate Matcher	Receiver 8B/10B Decoder	Receiver Phase Comp FIFO Module/ Byte Deserializer	Receiver PLL / CRU	Receiver XAUI State Machine	BIST Verifiers	Receiver Analog Circuits
rx_digitalreset									✓	✓	✓	✓	✓		✓	✓	
rx_analogreset								✓						✓			✓
tx_digitalreset	✓	✓				✓	✓										
gxb_powerdown	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
gxb_enable	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Voltage Reference Capabilities

Stratix II GX transceivers provide voltage reference and bias circuitry. To set up internal bias for controlling the transmitter output driver voltage swings, as well as to provide voltage and current biasing for other analog circuitry, the device uses an internal bandgap voltage reference of 0.7 V. An external 2-KΩ resistor connected to ground generates a constant bias current (independent of power supply drift, process changes, or temperature variation). An on-chip resistor generates a tracking current that tracks on-chip resistor variation. These currents are mirrored and distributed to the analog circuitry in each channel.

For more information, refer to the *DC and Switching Characteristics* chapter in volume 1 of the *Stratix II GX Handbook*.

Applications and Protocols Supported with Stratix II GX Devices

Each Stratix II GX transceiver block is designed to operate at any serial bit rate from 600 Mbps to 6.375 Gbps per channel. The wide data rate range allows Stratix II GX transceivers to support a wide variety of standards and protocols, such as PCI Express, GIGE, SONET/SDH, SDI, OIF-CEI, and XAUI. Stratix II GX devices are ideal for many high-speed communication applications, such as high-speed backplanes, chip-to-chip bridges, and high-speed serial communications links.

Example Applications Support for Stratix II GX

Stratix II GX devices can be used for many applications, including:

- Traffic management with various levels of quality of service (QoS) and integrated serial backplane interconnect
- Multi-port single-protocol switching (for example, PCI Express, GIGE, XAUI switch, or SONET/SDH)

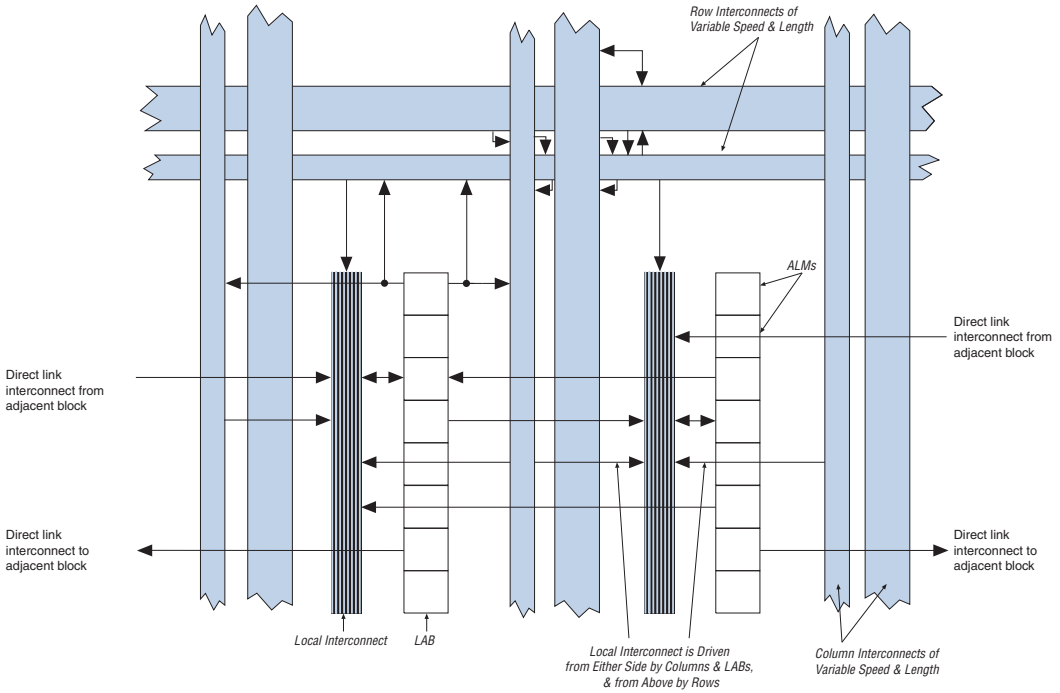
Logic Array Blocks

Each logic array block (LAB) consists of eight adaptive logic modules (ALMs), carry chains, shared arithmetic chains, LAB control signals, local interconnects, and register chain connection lines. The local interconnect transfers signals between ALMs in the same LAB. Register chain connections transfer the output of an ALM register to the adjacent ALM register in a LAB. The Quartus II Compiler places associated logic in a LAB or adjacent LABs, allowing the use of local, shared arithmetic chain, and register chain connections for performance and area efficiency. Table 2–17 shows Stratix II GX device resources. Figure 2–32 shows the Stratix II GX LAB structure.

Table 2–17. Stratix II GX Device Resources

Device	M512 RAM Columns/Blocks	M4K RAM Columns/Blocks	M-RAM Blocks	DSP Block Columns/Blocks	LAB Columns	LAB Rows
EP2SGX30	6/202	4/144	1	2/16	49	36
EP2SGX60	7/329	5/255	2	3/36	62	51
EP2SGX90	8/488	6/408	4	3/48	71	68
EP2SGX130	9/699	7/609	6	3/63	81	87

Figure 2–32. Stratix II GX LAB Structure

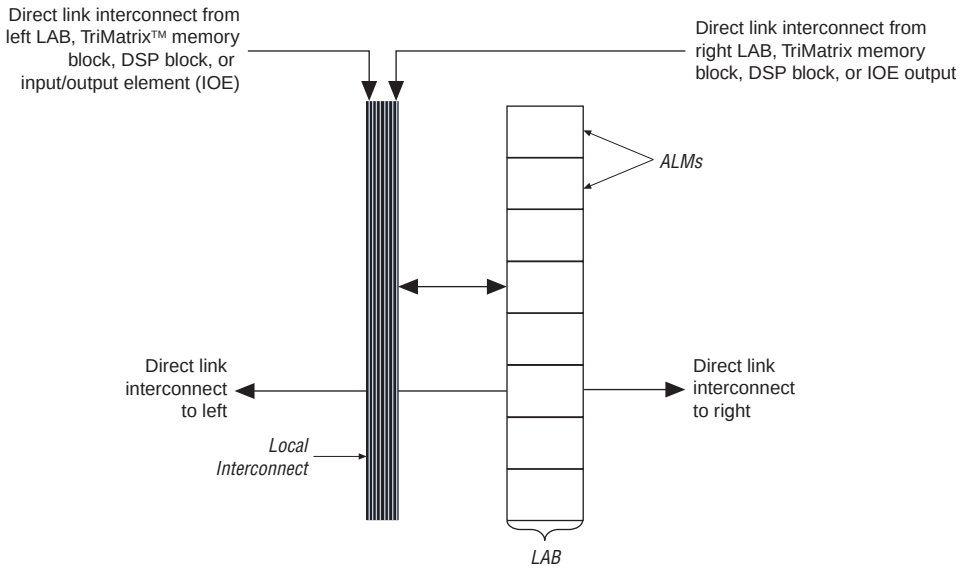


LAB Interconnects

The LAB local interconnect can drive all eight ALMs in the same LAB. It is driven by column and row interconnects and ALM outputs in the same LAB. Neighboring LABs, M512 RAM blocks, M4K RAM blocks, M-RAM blocks, or digital signal processing (DSP) blocks from the left and right can also drive a LAB's local interconnect through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each ALM can drive 24 ALMs through fast local and direct link interconnects.

Figure 2–33 shows the direct link connection.

Figure 2–33. Direct Link Connection



LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its ALMs. The control signals include three clocks, three clock enables, two asynchronous clears, synchronous clear, asynchronous preset/load, and synchronous load control signals, providing a maximum of 11 control signals at a time. Although synchronous load and clear signals are generally used when implementing counters, they can also be used with other functions.

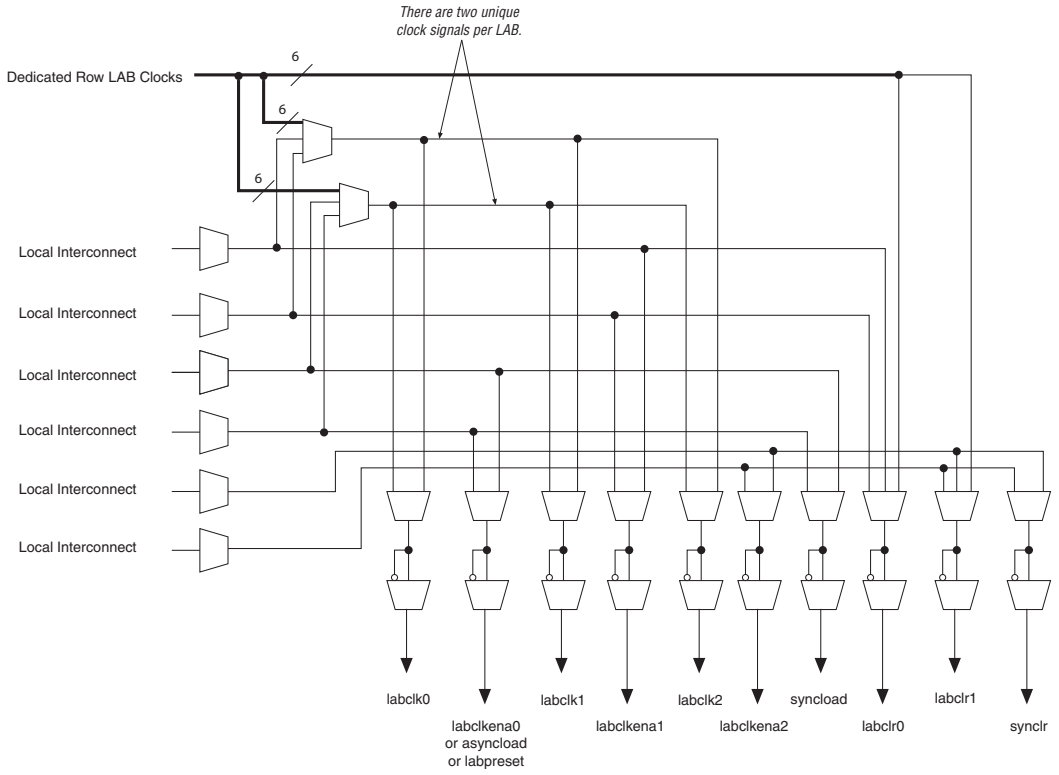
Each LAB can use three clocks and three clock enable signals. However, there can only be up to two unique clocks per LAB, as shown in the LAB control signal generation circuit in Figure 2–34. Each LAB's clock and clock enable signals are linked. For example, any ALM in a particular LAB using the `labclk1` signal also uses `labclkena1`. If the LAB uses both the rising and falling edges of a clock, it also uses two LAB-wide clock signals. De-asserting the clock enable signal turns off the corresponding LAB-wide clock. Each LAB can use two asynchronous clear signals and an asynchronous load/preset signal. The asynchronous

load acts as a preset when the asynchronous load data input is tied high. When the asynchronous load/preset signal is used, the `labclkena0` signal is no longer available.

The LAB row clocks `[5..0]` and LAB local interconnect generate the LAB-wide control signals. The MultiTrack™ interconnects have inherently low skew. This low skew allows the MultiTrack interconnects to distribute clock and control signals in addition to data.

Figure 2–34 shows the LAB control signal generation circuit.

Figure 2–34. LAB-Wide Control Signals



Adaptive Logic Modules

The basic building block of logic in the Stratix II GX architecture is the ALM. The ALM provides advanced features with efficient logic utilization. Each ALM contains a variety of look-up table (LUT)-based resources that can be divided between two adaptive LUTs (ALUTs). With up to eight inputs to the two ALUTs, one ALM can implement various combinations of two functions. This adaptability allows the ALM to be completely backward-compatible with four-input LUT architectures. One ALM can also implement any function of up to six inputs and certain seven-input functions.

In addition to the adaptive LUT-based resources, each ALM contains two programmable registers, two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. Through these dedicated resources, the ALM can efficiently implement various arithmetic functions and shift registers. Each ALM drives all types of interconnects: local, row, column, carry chain, shared arithmetic chain, register chain, and direct link interconnects. Figure 2–35 shows a high-level block diagram of the Stratix II GX ALM while Figure 2–36 shows a detailed view of all the connections in the ALM.

Figure 2–35. High-Level Block Diagram of the Stratix II GX ALM

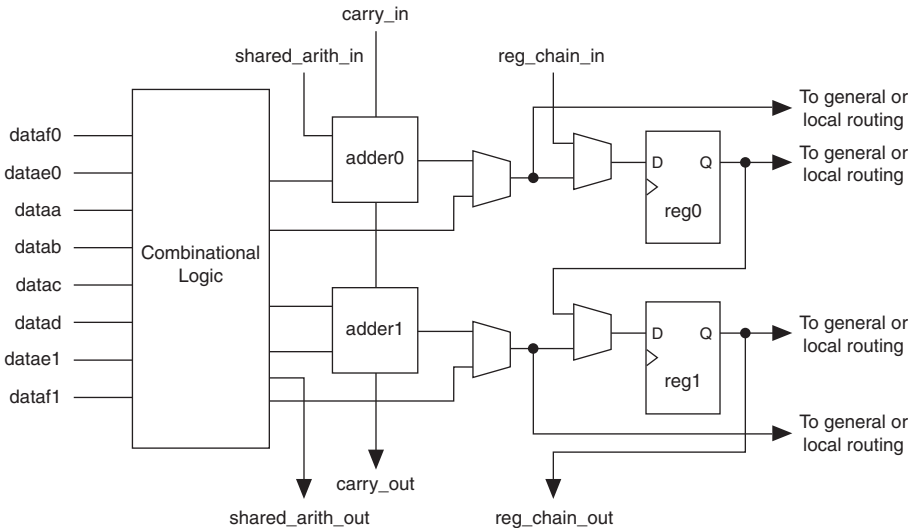
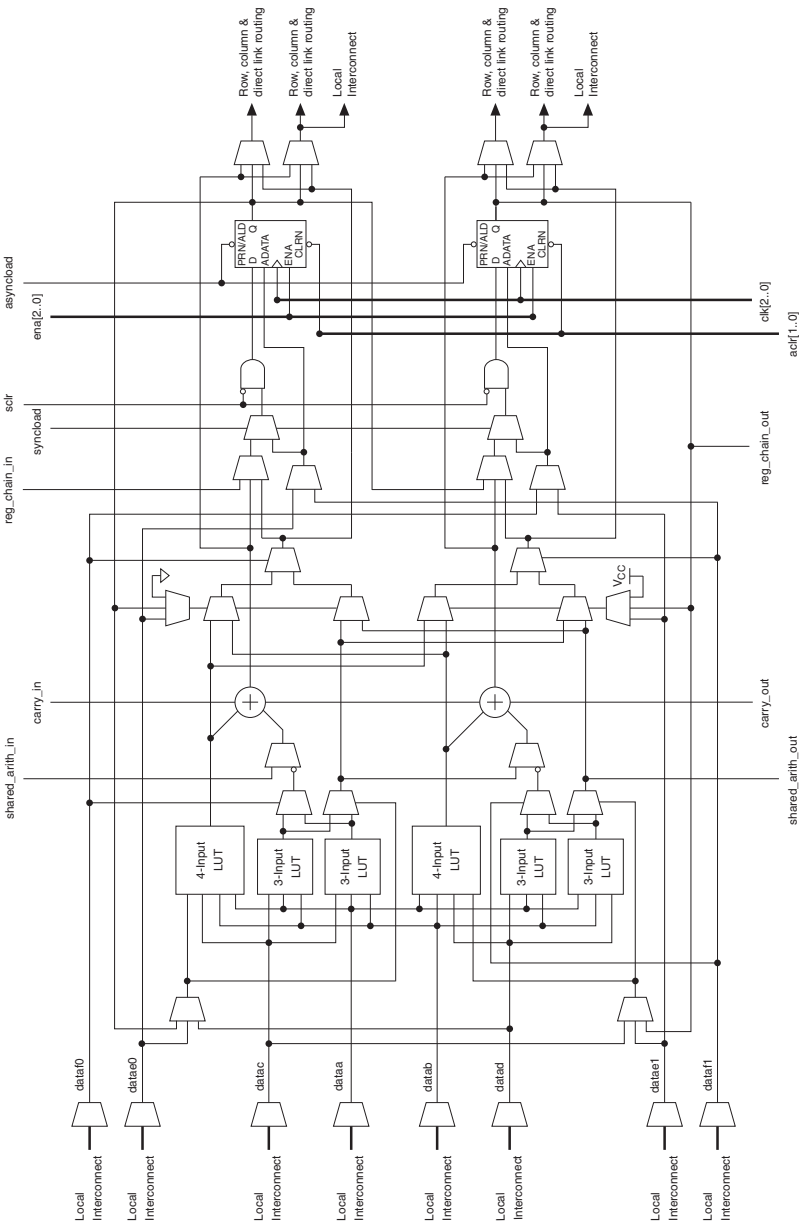


Figure 2–36. Stratix II GX ALM Details



One ALM contains two programmable registers. Each register has data, clock, clock enable, synchronous and asynchronous clear, asynchronous load data, and synchronous and asynchronous load/preset inputs.

Global signals, general-purpose I/O pins, or any internal logic can drive the register's clock and clear control signals. Either general-purpose I/O pins or internal logic can drive the clock enable, preset, asynchronous load, and asynchronous load data. The asynchronous load data input comes from the `datae` or `dataf` input of the ALM, which are the same inputs that can be used for register packing. For combinational functions, the register is bypassed and the output of the LUT drives directly to the outputs of the ALM.

Each ALM has two sets of outputs that drive the local, row, and column routing resources. The LUT, adder, or register output can drive these output drivers independently (see Figure 2–36). For each set of output drivers, two ALM outputs can drive column, row, or direct link routing connections, and one of these ALM outputs can also drive local interconnect resources. This allows the LUT or adder to drive one output while the register drives another output. This feature, called register packing, improves device utilization because the device can use the register and the combinational logic for unrelated functions. Another special packing mode allows the register output to feed back into the LUT of the same ALM so that the register is packed with its own fan-out LUT. This feature provides another mechanism for improved fitting. The ALM can also drive out registered and unregistered versions of the LUT or adder output.

See the *Stratix II Performance and Logic Efficiency Analysis White Paper* for more information on the efficiencies of the Stratix II GX ALM and comparisons with previous architectures.

ALM Operating Modes

The Stratix II GX ALM can operate in one of the following modes:

- Normal mode
- Extended LUT mode
- Arithmetic mode
- Shared arithmetic mode

Each mode uses ALM resources differently. Each mode has 11 available inputs to the ALM (see Figure 2–35)—the eight data inputs from the LAB local interconnect; carry-in from the previous ALM or LAB; the shared arithmetic chain connection from the previous ALM or LAB; and the register chain connection—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock,

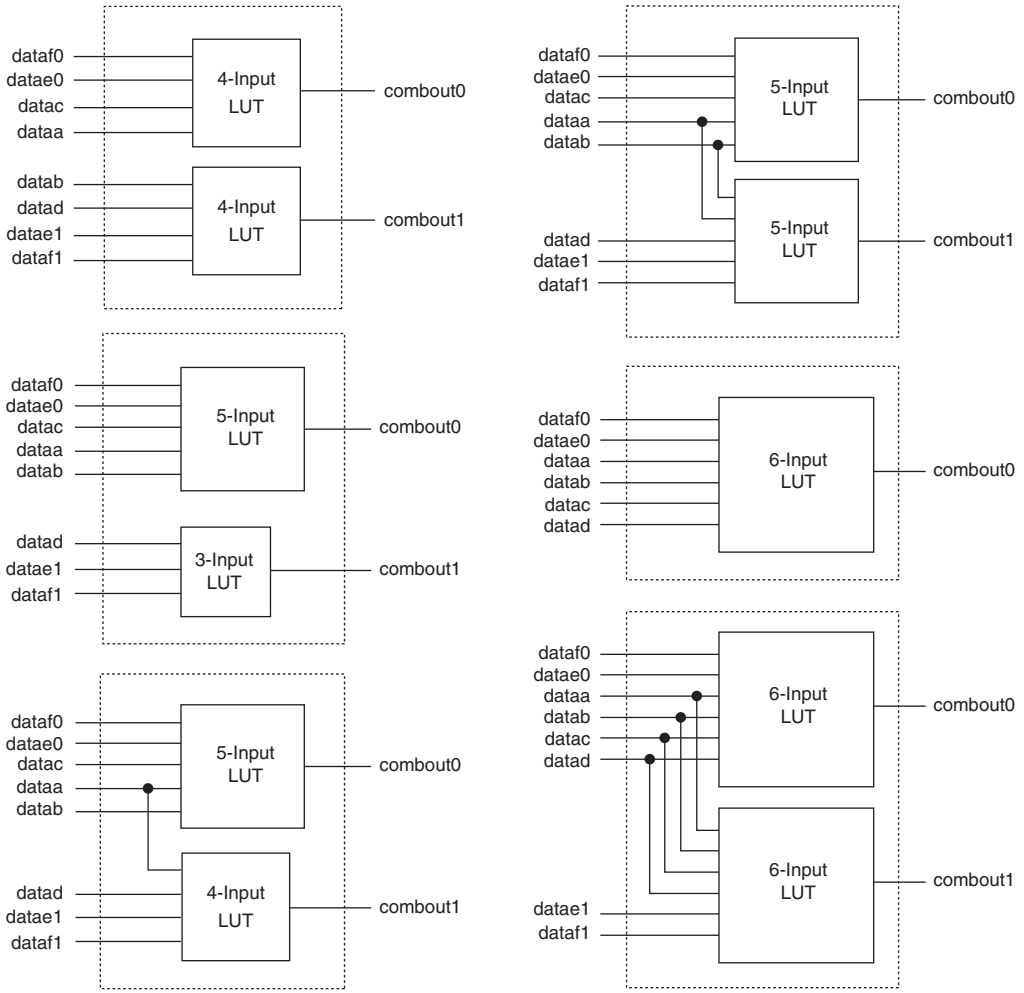
asynchronous clear, asynchronous preset/load, synchronous clear, synchronous load, and clock enable control for the register. These LAB wide signals are available in all ALM modes. Refer to "LAB Control Signals" on page 2–46 for more information on the LAB-wide control signals.

The Quartus II software and supported third-party synthesis tools, in conjunction with parameterized functions such as library of parameterized modules (LPM) functions, automatically choose the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions. If required, you can also create special-purpose functions that specify which ALM operating mode to use for optimal performance.

Normal Mode

The normal mode is suitable for general logic applications and combinational functions. In this mode, up to eight data inputs from the LAB local interconnect are inputs to the combinational logic. The normal mode allows two functions to be implemented in one Stratix II GX ALM, or an ALM to implement a single function of up to six inputs. The ALM can support certain combinations of completely independent functions and various combinations of functions which have common inputs. Figure 2–37 shows the supported LUT combinations in normal mode.

Figure 2–37. ALM in Normal Mode *Note (1)*



Note to Figure 2–37:

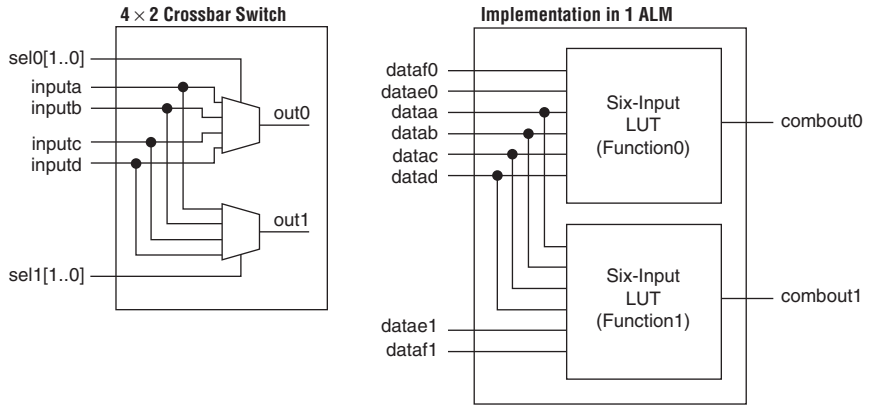
(1) Combinations of functions with less inputs than those shown are also supported. For example, combinations of functions with the following number of inputs are supported: 4 and 3, 3 and 3, 3 and 2, 5 and 2, etc.

The normal mode provides complete backward compatibility with four-input LUT architectures. Two independent functions of four inputs or less can be implemented in one Stratix II GX ALM. In addition, a five-input function and an independent three-input function can be implemented without sharing inputs.

To pack two five-input functions into one ALM, the functions must have at least two common inputs. The common inputs are dataa and datab. The combination of a four-input function with a five-input function requires one common input (either dataa or datab).

To implement two six-input functions in one ALM, four inputs must be shared and the combinational function must be the same. For example, a 4 × 2 crossbar switch (two 4-to-1 multiplexers with common inputs and unique select lines) can be implemented in one ALM, as shown in Figure 2–38. The shared inputs are dataa, datab, datac, and datad, while the unique select lines are datae0 and dataf0 for function0, and datae1 and dataf1 for function1. This crossbar switch consumes four LUTs in a four-input LUT-based architecture.

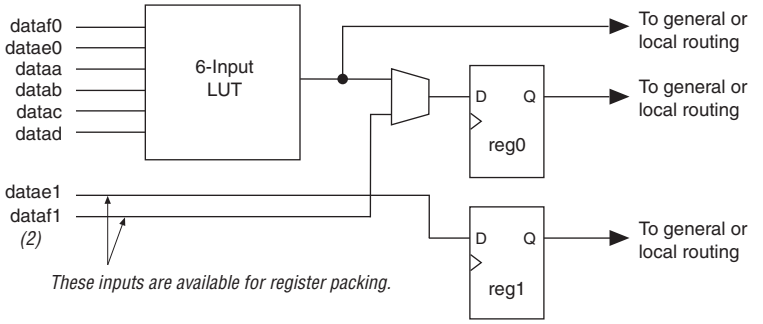
Figure 2–38. 4 × 2 Crossbar Switch Example



In a sparsely used device, functions that could be placed into one ALM can be implemented in separate ALMs. The Quartus II Compiler spreads a design out to achieve the best possible performance. As a device begins to fill up, the Quartus II software automatically utilizes the full potential of the Stratix II GX ALM. The Quartus II Compiler automatically searches for functions of common inputs or completely independent functions to be placed into one ALM and to make efficient use of the device resources. In addition, you can manually control resource usage by setting location assignments. Any six-input function can be implemented utilizing inputs dataa, datab, datac, datad, and either datae0 and dataf0 or datae1 and dataf1. If datae0 and dataf0 are utilized, the output is driven to register0, and/or register0 is bypassed and the data drives out to the interconnect using the top set of output drivers (see Figure 2–39). If datae1 and dataf1 are utilized, the output drives to register1 and/or bypasses register1 and drives to the interconnect

using the bottom set of output drivers. The Quartus II Compiler automatically selects the inputs to the LUT. Asynchronous load data for the register comes from the `datae` or `dataf` input of the ALM. ALMs in normal mode support register packing.

Figure 2–39. 6-Input Function in Normal Mode *Notes (1), (2)*



Notes to Figure 2–39:

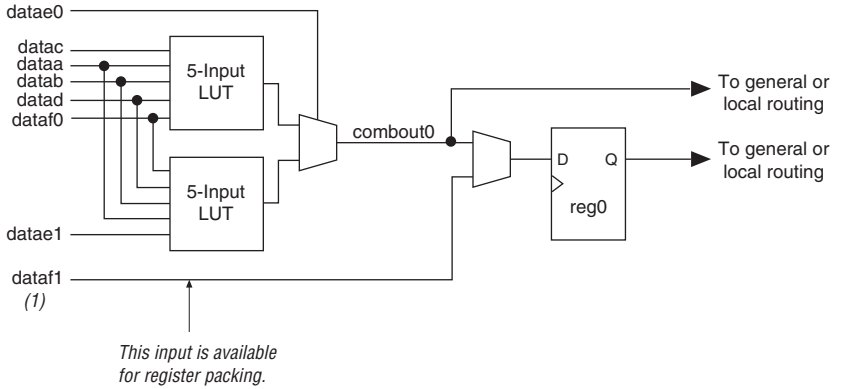
(1) If `datae1` and `dataf1` are used as inputs to the six-input function, `datae0` and `dataf0` are available for register packing.

(2) The `dataf1` input is available for register packing only if the six-input function is un-registered.

Extended LUT Mode

The extended LUT mode is used to implement a specific set of seven-input functions. The set must be a 2-to-1 multiplexer fed by two arbitrary five-input functions sharing four inputs. Figure 2–40 shows the template of supported seven-input functions utilizing extended LUT mode. In this mode, if the seven-input function is unregistered, the unused eighth input is available for register packing. Functions that fit into the template shown in Figure 2–40 occur naturally in designs. These functions often appear in designs as "if-else" statements in Verilog HDL or VHDL code.

Figure 2–40. Template for Supported Seven-Input Functions in Extended LUT Mode



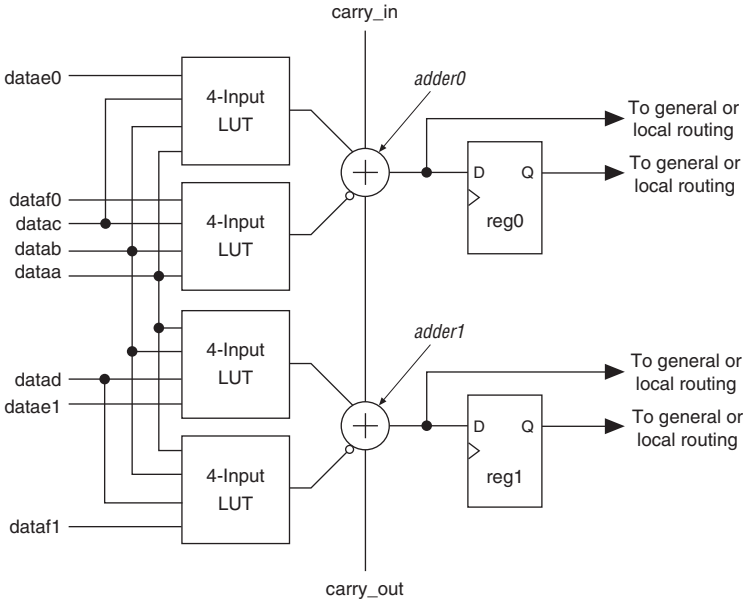
Note to Figure 2–40:

(1) If the seven-input function is un-registered, the unused eighth input is available for register packing. The second register, `reg1`, is not available.

Arithmetic Mode

The arithmetic mode is ideal for implementing adders, counters, accumulators, wide parity functions, and comparators. An ALM in arithmetic mode uses two sets of two four-input LUTs along with two dedicated full adders. The dedicated adders allow the LUTs to be available to perform pre-adder logic; therefore, each adder can add the output of two four-input functions. The four LUTs share the `dataa` and `datab` inputs. As shown in Figure 2–41, the carry-in signal feeds to `adder0`, and the carry-out from `adder0` feeds to carry-in of `adder1`. The carry-out from `adder1` drives to `adder0` of the next ALM in the LAB. ALMs in arithmetic mode can drive out registered and/or un-registered versions of the adder outputs.

Figure 2–41. ALM in Arithmetic Mode

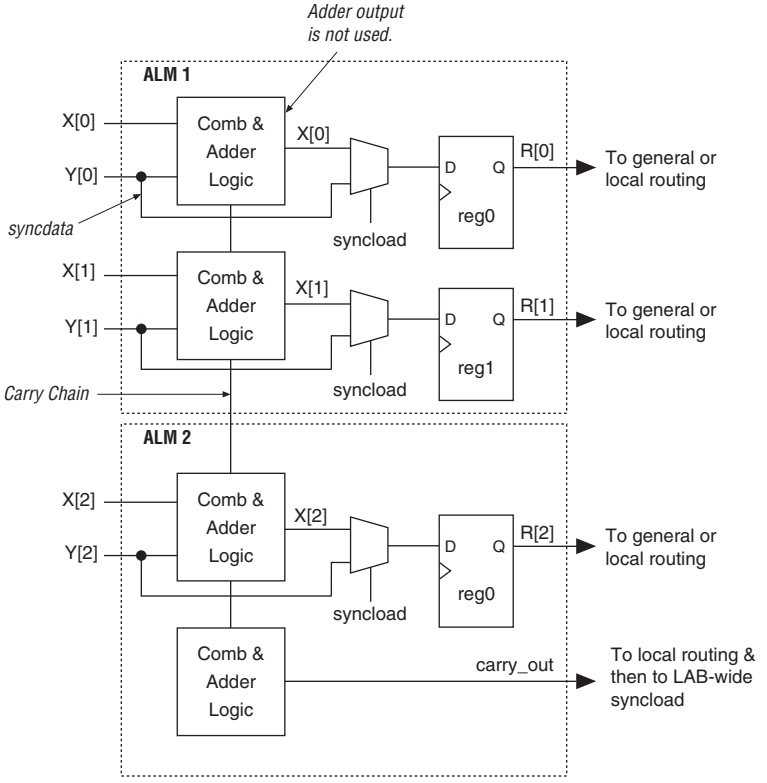


While operating in arithmetic mode, the ALM can support simultaneous use of the adder's carry output along with combinational logic outputs. In this operation, the adder output is ignored. This usage of the adder with the combinational logic output provides resource savings of up to 50% for functions that can use this ability. An example of such functionality is a conditional operation, such as the one shown in Figure 2–42. The equation for this example is:

R = (X < Y) ? Y : X

To implement this function, the adder is used to subtract 'Y' from 'X'. If 'X' is less than 'Y', the `carry_out` signal will be '1'. The `carry_out` signal is fed to an adder where it drives out to the LAB local interconnect. It then feeds to the LAB-wide `syncload` signal. When asserted, `syncload` selects the `syncdata` input. In this case, the data 'Y' drives the `syncdata` inputs to the registers. If 'X' is greater than or equal to 'Y', the `syncload` signal is de-asserted and 'X' drives the data port of the registers.

Figure 2–42. Conditional Operation Example



The arithmetic mode also offers clock enable, counter enable, synchronous up and down control, add and subtract control, synchronous clear, synchronous load. The LAB local interconnect data inputs generate the clock enable, counter enable, synchronous up and down and add and subtract control signals. These control signals may be used for the inputs that are shared between the four LUTs in the ALM. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. The Quartus II software automatically places any registers that are not used by the counter into other LABs.

Carry Chain

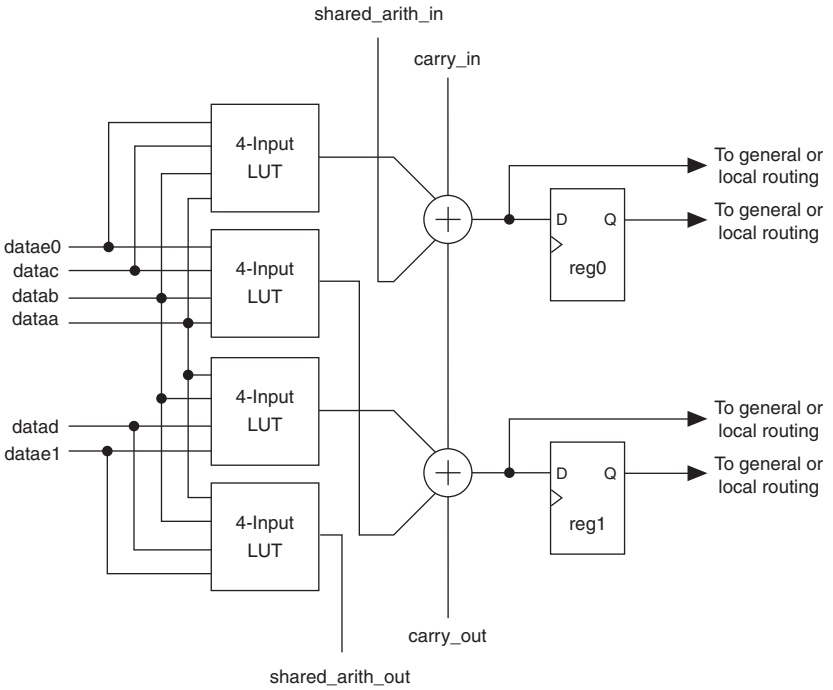
The carry chain provides a fast carry function between the dedicated adders in arithmetic or shared arithmetic mode. Carry chains can begin in either the first ALM or the fifth ALM in a LAB. The final carry-out signal is routed to an ALM, where it is fed to local, row, or column interconnects.

The Quartus II Compiler automatically creates carry chain logic during compilation, or you can create it manually during design entry. Parameterized functions, such as LPM functions, automatically take advantage of carry chains for the appropriate functions. The Quartus II Compiler creates carry chains longer than 16 (8 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically, allowing fast horizontal connections to TriMatrix memory and DSP blocks. A carry chain can continue as far as a full column. To avoid routing congestion in one small area of the device when a high fan-in arithmetic function is implemented, the LAB can support carry chains that only utilize either the top half or the bottom half of the LAB before connecting to the next LAB. The other half of the ALMs in the LAB is available for implementing narrower fan-in functions in normal mode. Carry chains that use the top four ALMs in the first LAB will carry into the top half of the ALMs in the next LAB within the column. Carry chains that use the bottom four ALMs in the first LAB will carry into the bottom half of the ALMs in the next LAB within the column. Every other column of the LABs are top-half bypassable, while the other LAB columns are bottom-half bypassable. Refer to "MultiTrack Interconnect" on page 2–63 for more information on carry chain interconnect.

Shared Arithmetic Mode

In shared arithmetic mode, the ALM can implement a three-input add. In this mode, the ALM is configured with four 4-input LUTs. Each LUT either computes the sum of three inputs or the carry of three inputs. The output of the carry computation is fed to the next adder (either to `adder1` in the same ALM or to `adder0` of the next ALM in the LAB) using a dedicated connection called the shared arithmetic chain. This shared arithmetic chain can significantly improve the performance of an adder tree by reducing the number of summation stages required to implement an adder tree. Figure 2–43 shows the ALM in shared arithmetic mode.

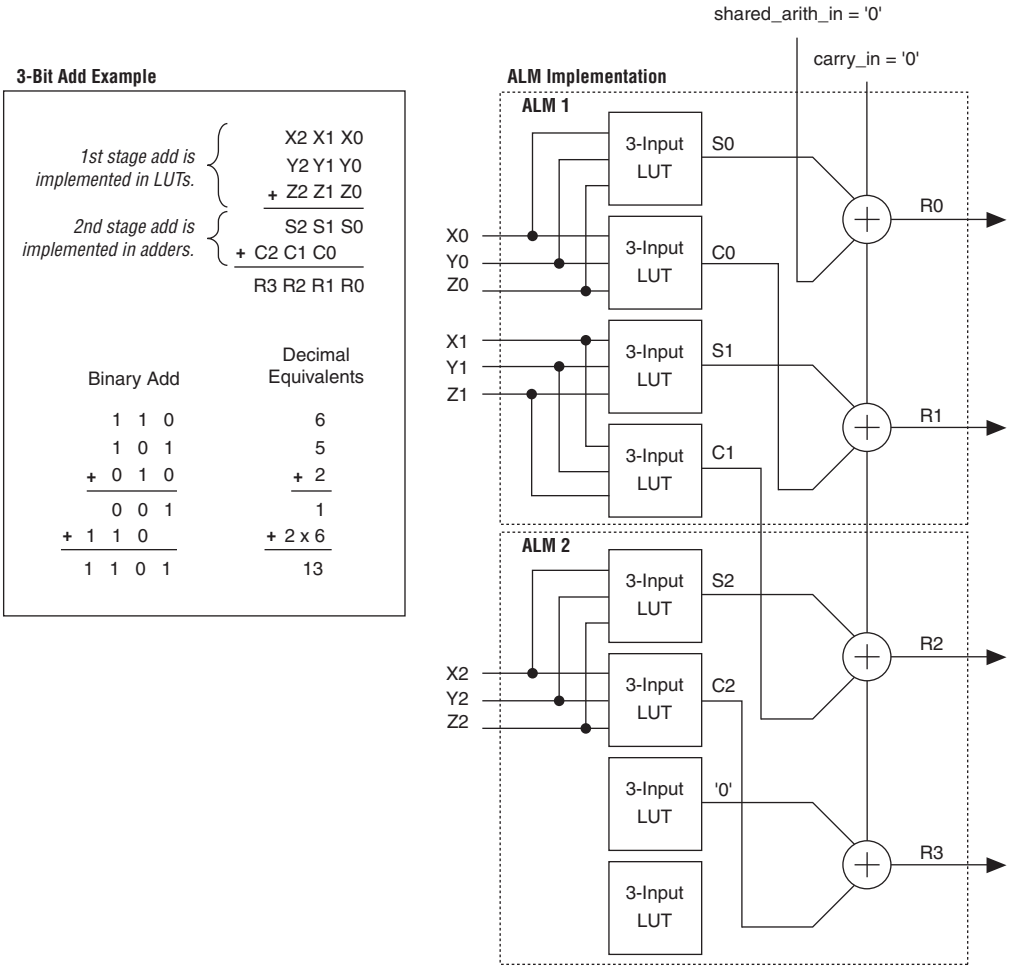
Figure 2–43. ALM in Shared Arithmetic Mode



Note to Figure 2–43:
(1) Inputs `dataf0` and `dataf1` are available for register packing in shared arithmetic mode.

Adder trees are used in many different applications. For example, the summation of the partial products in a logic-based multiplier can be implemented in a tree structure. Another example is a correlator function that can use a large adder tree to sum filtered data samples in a given time frame to recover or to de-spread data which was transmitted utilizing spread spectrum technology. An example of a three-bit add operation utilizing the shared arithmetic mode is shown in Figure 2–44. The partial sum (`S[2..0]`) and the partial carry (`C[2..0]`) is obtained using the LUTs, while the result (`R[2..0]`) is computed using the dedicated adders.

Figure 2–44. Example of a 3-Bit Add Utilizing Shared Arithmetic Mode



Shared Arithmetic Chain

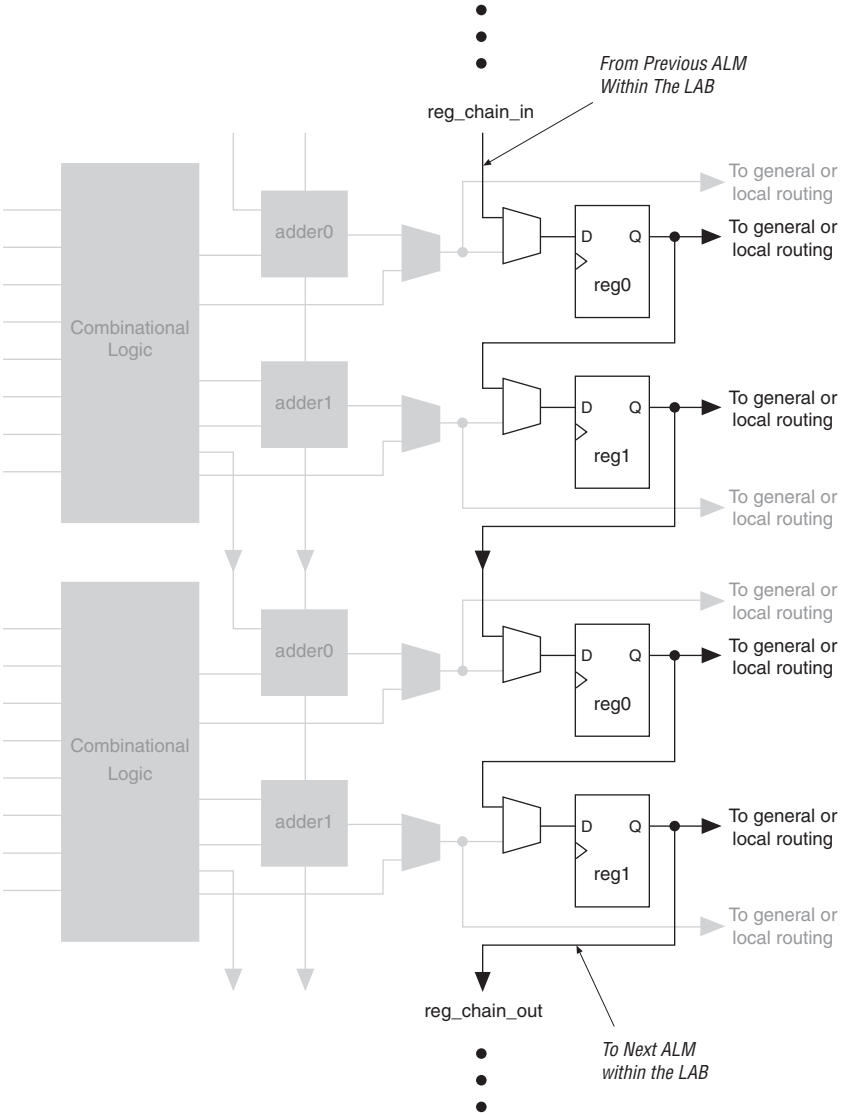
In addition to the dedicated carry chain routing, the shared arithmetic chain available in shared arithmetic mode allows the ALM to implement a three-input add, which significantly reduces the resources necessary to implement large adder trees or correlator functions. The shared arithmetic chains can begin in either the first or fifth ALM in a LAB. The Quartus II Compiler automatically links LABs to create shared arithmetic chains longer than 16 (8 ALMs in arithmetic or shared arithmetic mode). For enhanced fitting, a long shared arithmetic chain runs vertically

allowing fast horizontal connections to TriMatrix memory and DSP blocks. A shared arithmetic chain can continue as far as a full column. Similar to the carry chains, the shared arithmetic chains are also top- or bottom-half bypassable. This capability allows the shared arithmetic chain to cascade through half of the ALMs in a LAB while leaving the other half available for narrower fan-in functionality. Every other LAB column is top-half bypassable, while the other LAB columns are bottom-half bypassable. Refer to "MultiTrack Interconnect" on page 2–63 for more information on shared arithmetic chain interconnect.

Register Chain

In addition to the general routing outputs, the ALMs in a LAB have register chain outputs. The register chain routing allows registers in the same LAB to be cascaded together. The register chain interconnect allows a LAB to use LUTs for a single combinational function and the registers to be used for an unrelated shift register implementation. These resources speed up connections between ALMs while saving local interconnect resources (see Figure 2–45). The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance. See "MultiTrack Interconnect" on page 2–63 for more information about register chain interconnect.

Figure 2–45. Register Chain within a LAB *Note (1)*



Note to Figure 2–45:
(1) The combinational or adder logic can be utilized to implement an unrelated, un-registered function.

Clear and Preset Logic Control

LAB-wide signals control the logic for the register's clear and load/preset signals. The ALM directly supports an asynchronous clear and preset function. The register preset is achieved through the asynchronous load of a logic high. The direct asynchronous preset does not require a NOT gate push-back technique. Stratix II GX devices support simultaneous asynchronous load/preset and clear signals. An asynchronous clear signal takes precedence if both signals are asserted simultaneously. Each LAB supports up to two clears and one load/preset signal.

In addition to the clear and load/preset ports, Stratix II GX devices provide a device-wide reset pin (DEV_CLRn) that resets all registers in the device. An option set before compilation in the Quartus II software controls this pin. This device-wide reset overrides all other control signals.

MultiTrack Interconnect

In the Stratix II GX architecture, the MultiTrack interconnect structure with DirectDrive technology provides connections between ALMs, TriMatrix memory, DSP blocks, and device I/O pins. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different lengths and speeds used for inter- and intra-design block connectivity. The Quartus II Compiler automatically places critical design paths on faster interconnects to improve design performance.

DirectDrive technology is a deterministic routing technology that ensures identical routing resource usage for any function regardless of placement in the device. The MultiTrack interconnect and DirectDrive technology simplify the integration stage of block-based designing by eliminating the re-optimization cycles that typically follow design changes and additions.

The MultiTrack interconnect consists of row and column interconnects that span fixed distances. A routing structure with fixed length resources for all devices allows predictable and repeatable performance when migrating through different device densities. Dedicated row interconnects route signals to and from LABs, DSP blocks, and TriMatrix memory in the same row.

These row resources include:

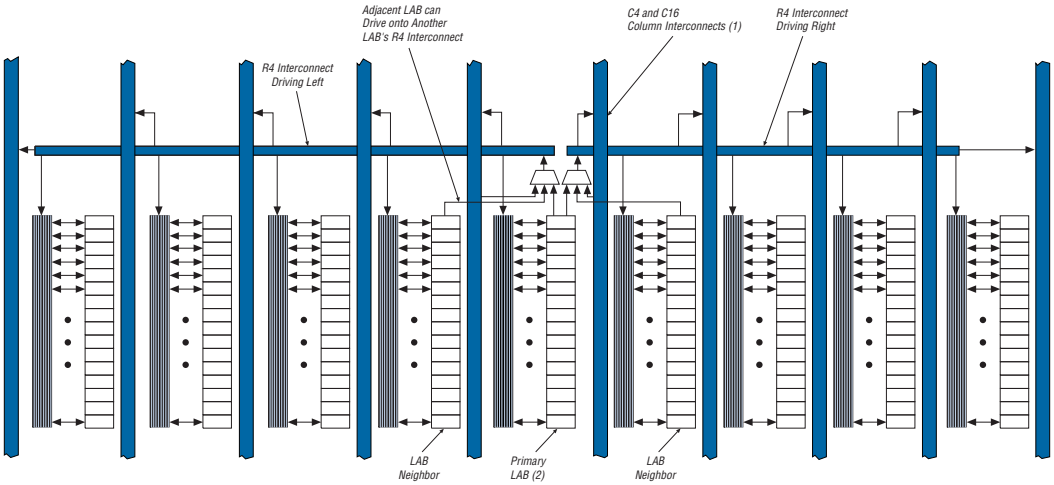
- Direct link interconnects between LABs and adjacent blocks
- R4 interconnects traversing four blocks to the right or left
- R24 row interconnects for high-speed access across the length of the device

The direct link interconnect allows a LAB, DSP block, or TriMatrix memory block to drive into the local interconnect of its left and right neighbors and then back into itself, providing fast communication between adjacent LABs and/or blocks without using row interconnect resources.

The R4 interconnects span four LABs, three LABs and one M512 RAM block, two LABs and one M4K RAM block, or two LABs and one DSP block to the right or left of a source LAB. These resources are used for fast row connections in a four-LAB region. Every LAB has its own set of R4 interconnects to drive either left or right. Figure 2–46 shows R4 interconnect connections from a LAB.

R4 interconnects can drive and be driven by DSP blocks and RAM blocks and row IOEs. For LAB interfacing, a primary LAB or LAB neighbor can drive a given R4 interconnect. For R4 interconnects that drive to the right, the primary LAB and right neighbor can drive onto the interconnect. For R4 interconnects that drive to the left, the primary LAB and its left neighbor can drive onto the interconnect. R4 interconnects can drive other R4 interconnects to extend the range of LABs they can drive. R4 interconnects can also drive C4 and C16 interconnects for connections from one row to another. Additionally, R4 interconnects can drive R24 interconnects.

Figure 2–46. R4 Interconnect Connections *Notes (1), (2), (3)*



Notes to Figure 2–46:

(1) C4 and C16 interconnects can drive R4 interconnects.
(2) This pattern is repeated for every LAB in the LAB row.
(3) The LABs in Figure 2–46 show the 16 possible logical outputs per LAB.

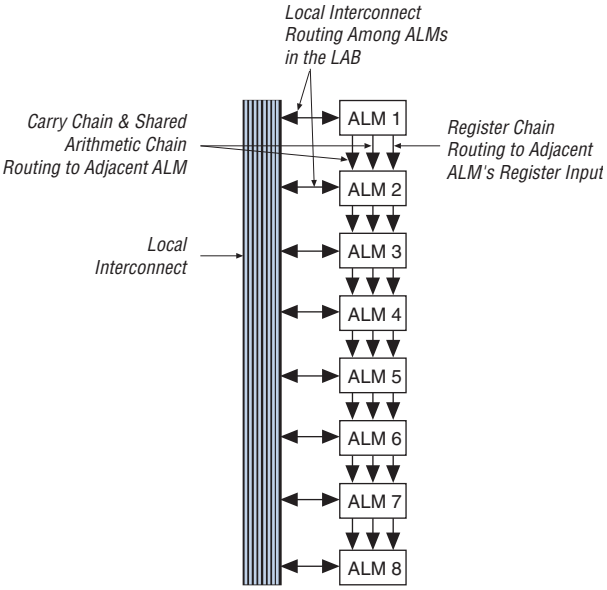
R24 row interconnects span 24 LABs and provide the fastest resource for long row connections between LABs, TriMatrix memory, DSP blocks, and Row IOEs. The R24 row interconnects can cross M-RAM blocks. R24 row interconnects drive to other row or column interconnects at every fourth LAB and do not drive directly to LAB local interconnects. R24 row interconnects drive LAB local interconnects via R4 and C4 interconnects. R24 interconnects can drive R24, R4, C16, and C4 interconnects. The column interconnect operates similarly to the row interconnect and vertically routes signals to and from LABs, TriMatrix memory, DSP blocks, and IOEs. Each column of LABs is served by a dedicated column interconnect.

These column resources include:

- Shared arithmetic chain interconnects in a LAB
- Carry chain interconnects in a LAB and from LAB to LAB
- Register chain interconnects in a LAB
- C4 interconnects traversing a distance of four blocks in an up and down direction
- C16 column interconnects for high-speed vertical routing through the device

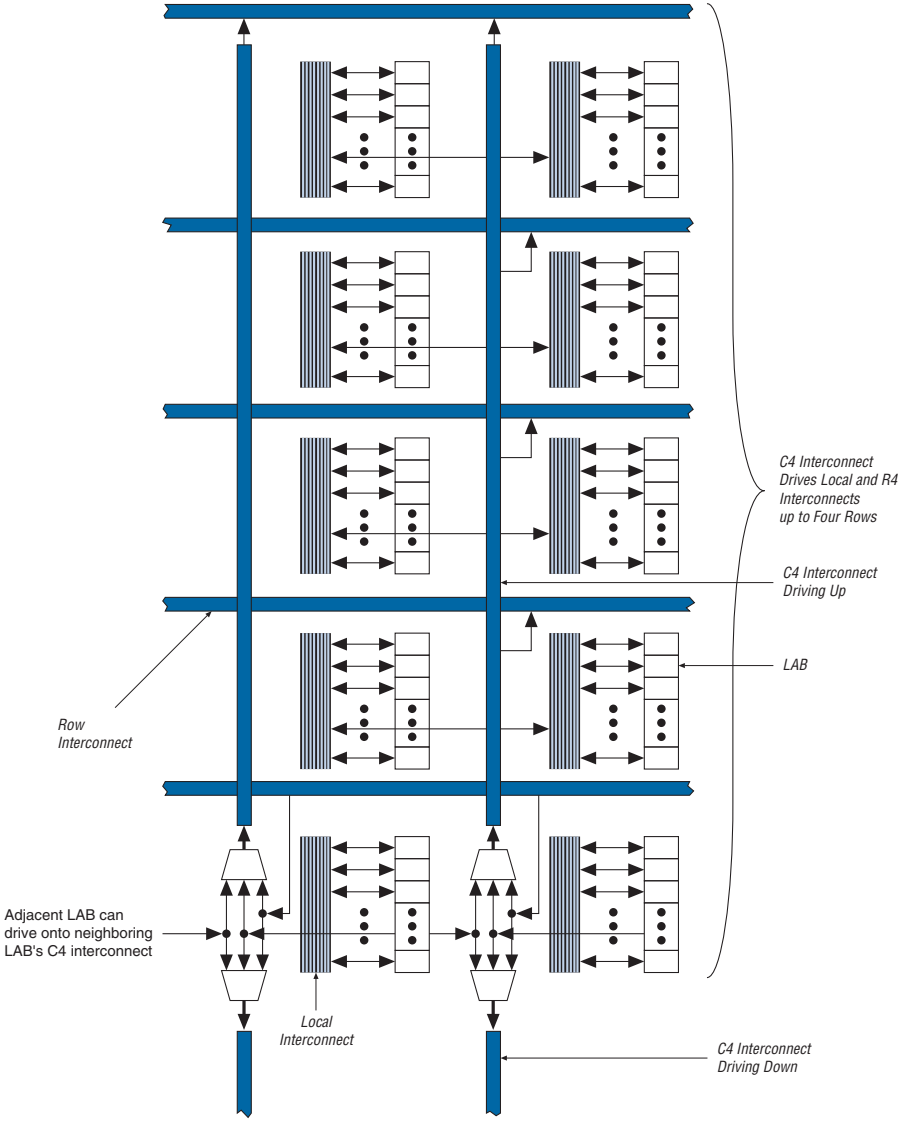
Stratix II GX devices include an enhanced interconnect structure in LABs for routing shared arithmetic chains and carry chains for efficient arithmetic functions. The register chain connection allows the register output of one ALM to connect directly to the register input of the next ALM in the LAB for fast shift registers. These ALM-to-ALM connections bypass the local interconnect. The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance. Figure 2–47 shows the shared arithmetic chain, carry chain, and register chain interconnects.

Figure 2–47. Shared Arithmetic Chain, Carry Chain and Register Chain Interconnects



The C4 interconnects span four LABs, M512, or M4K blocks up or down from a source LAB. Every LAB has its own set of C4 interconnects to drive either up or down. Figure 2–48 shows the C4 interconnect connections from a LAB in a column. The C4 interconnects can drive and be driven by all types of architecture blocks, including DSP blocks, TriMatrix memory blocks, and column and row IOEs. For LAB interconnection, a primary LAB or its LAB neighbor can drive a given C4 interconnect. C4 interconnects can drive each other to extend their range as well as drive row interconnects for column-to-column connections.

Figure 2–48. C4 Interconnect Connections *Note (1)*



Note to Figure 2–48:

(1) Each C4 interconnect can drive either up or down four rows.

C16 column interconnects span a length of 16 LABs and provide the fastest resource for long column connections between LABs, TriMatrix memory blocks, DSP blocks, and IOEs. C16 interconnects can cross M-RAM blocks and also drive to row and column interconnects at every fourth LAB. C16 interconnects drive LAB local interconnects via C4 and R4 interconnects and do not drive LAB local interconnects directly. All embedded blocks communicate with the logic array similar to LAB-to-LAB interfaces. Each block (that is, TriMatrix memory and DSP blocks) connects to row and column interconnects and has local interconnect regions driven by row and column interconnects. These blocks also have direct link interconnects for fast connections to and from a neighboring LAB. All blocks are fed by the row LAB clocks, `labclk[5..0]`.

Table 2–18 shows the Stratix II GX device's routing scheme.

Table 2–18. Stratix II GX Device Routing Scheme (Part 1 of 2)

Source	Destination															
	Shared Arithmetic Chain	Carry Chain	Register Chain	Local Interconnect	Direct Link Interconnect	R4 Interconnect	R24 Interconnect	C4 Interconnect	C16 Interconnect	ALM	M512 RAM Block	M4K RAM Block	M-RAM Block	DSP Blocks	Column IOE	Row IOE
Shared arithmetic chain										✓						
Carry chain										✓						
Register chain										✓						
Local interconnect										✓	✓	✓	✓	✓	✓	✓
Direct link interconnect				✓												
R4 interconnect				✓		✓	✓	✓	✓							
R24 interconnect						✓	✓	✓	✓							
C4 interconnect				✓		✓		✓								
C16 interconnect						✓	✓	✓	✓							
ALM	✓	✓	✓	✓	✓	✓		✓								
M512 RAM block				✓	✓	✓		✓								
M4K RAM block				✓	✓	✓		✓								
M-RAM block					✓	✓	✓	✓								
DSP blocks					✓	✓		✓								

Table 2–18. Stratix II GX Device Routing Scheme (Part 2 of 2)

Source	Destination															
	Shared Arithmetic Chain	Carry Chain	Register Chain	Local Interconnect	Direct Link Interconnect	R4 Interconnect	R24 Interconnect	C4 Interconnect	C16 Interconnect	ALM	M512 RAM Block	M4K RAM Block	M-RAM Block	DSP Blocks	Column IOE	Row IOE
Column IOE					✓			✓	✓							
Row IOE					✓	✓	✓	✓								

TriMatrix Memory

TriMatrix memory consists of three types of RAM blocks: M512, M4K, and M-RAM. Although these memory blocks are different, they can all implement various types of memory with or without parity, including true dual-port, simple dual-port, and single-port RAM, ROM, and FIFO buffers. Table 2–19 shows the size and features of the different RAM blocks.

Table 2–19. TriMatrix Memory Features (Part 1 of 2)

Memory Feature	M512 RAM Block (32 × 18 Bits)	M4K RAM Block (128 × 36 Bits)	M-RAM Block (4K × 144 Bits)
Maximum performance	500 MHz	550 MHz	420 MHz
True dual-port memory		✓	✓
Simple dual-port memory	✓	✓	✓
Single-port memory	✓	✓	✓
Shift register	✓	✓	
ROM	✓	✓	*(1)*
FIFO buffer	✓	✓	✓
Pack mode		✓	✓
Byte enable	✓	✓	✓
Address clock enable		✓	✓
Parity bits	✓	✓	✓
Mixed clock mode	✓	✓	✓
Memory initialization (**.mif**)	✓	✓	

Table 2–19. TriMatrix Memory Features (Part 2 of 2)

Memory Feature	M512 RAM Block (32 × 18 Bits)	M4K RAM Block (128 × 36 Bits)	M-RAM Block (4K × 144 Bits)
Simple dual-port memory mixed width support	✓	✓	✓
True dual-port memory mixed width support		✓	✓
Power-up conditions	Outputs cleared	Outputs cleared	Outputs unknown
Register clears	Output registers	Output registers	Output registers
Mixed-port read-during-write	Unknown output/old data	Unknown output/old data	Unknown output
Configurations	512 × 1 256 × 2 128 × 4 64 × 8 64 × 9 32 × 16 32 × 18	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 16K × 36 8K × 64 8K × 72 4K × 128 4K × 144

Note to Table 2–19:
(1) Violating the setup or hold time on the memory block address registers could corrupt memory contents. This applies to both read and write operations.

TriMatrix memory provides three different memory sizes for efficient application support. The Quartus II software automatically partitions the user-defined memory into the embedded memory blocks using the most efficient size combinations. You can also manually assign the memory to a specific block size or a mixture of block sizes.

M512 RAM Block

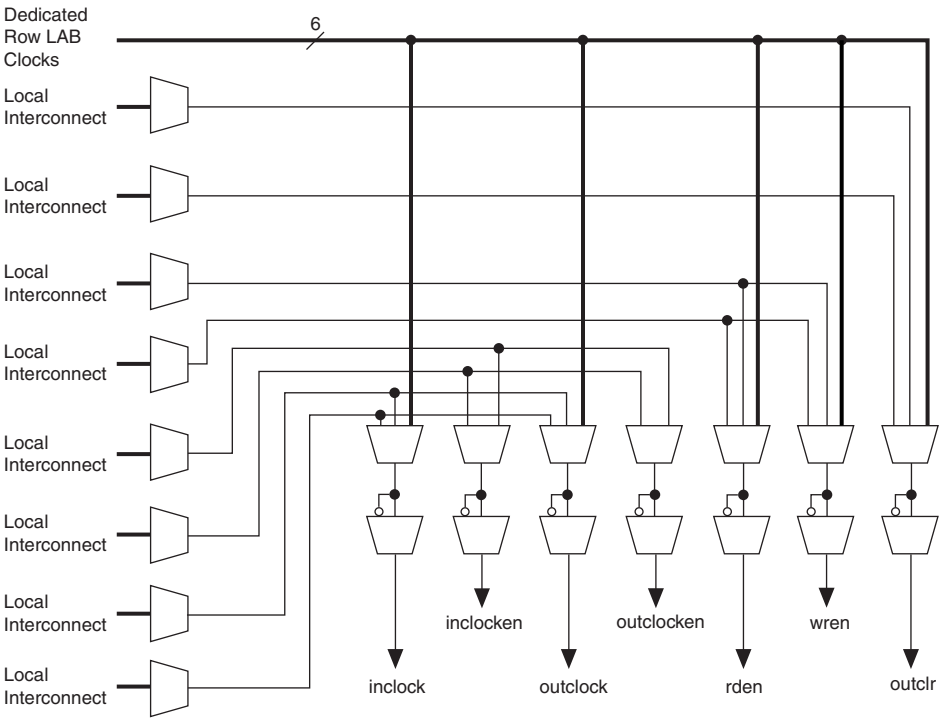
The M512 RAM block is a simple dual-port memory block and is useful for implementing small FIFO buffers, DSP, and clock domain transfer applications. Each block contains 576 RAM bits (including parity bits). M512 RAM blocks can be configured in the following modes:

- Simple dual-port RAM
- Single-port RAM
- FIFO
- ROM
- Shift register

When configured as RAM or ROM, you can use an initialization file to pre-load the memory contents.

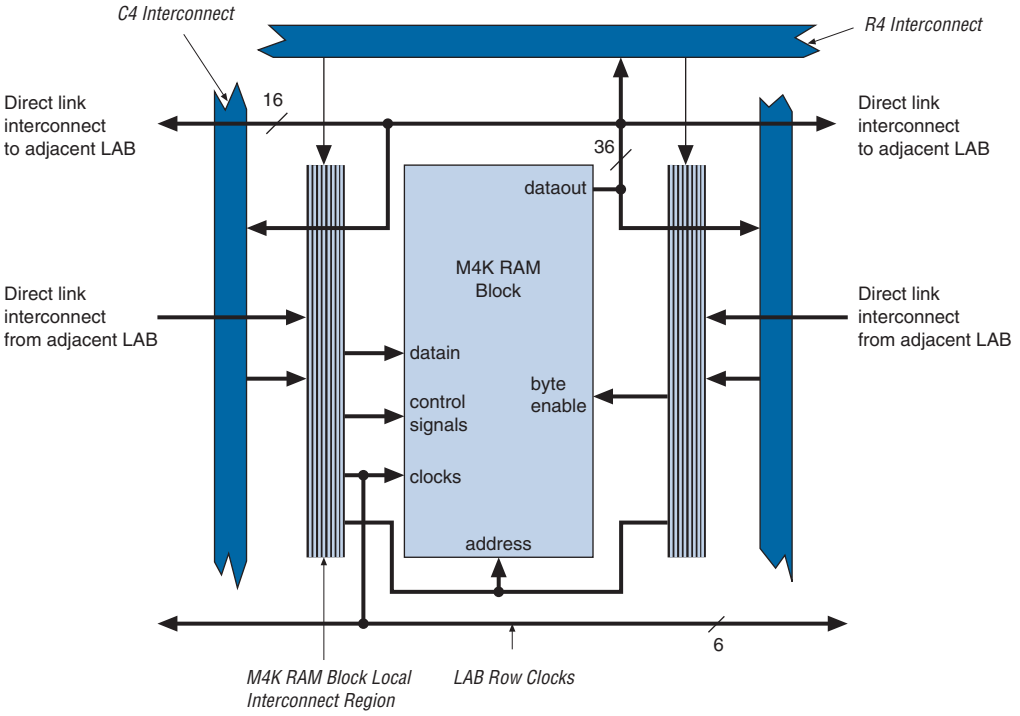
M512 RAM blocks can have different clocks on its inputs and outputs. The wren, datain, and write address registers are all clocked together from one of the two clocks feeding the block. The read address, rden, and output registers can be clocked by either of the two clocks driving the block, allowing the RAM block to operate in read and write or input and output clock modes. Only the output register can be bypassed. The six labclk signals or local interconnect can drive the inclock, outclock, wren, rden, and outclr signals. Because of the advanced interconnect between the LAB and M512 RAM blocks, ALMs can also control the wren and rden signals and the RAM clock, clock enable, and asynchronous clear signals. Figure 2–49 shows the M512 RAM block control signal generation logic.

Figure 2–49. M512 RAM Block Control Signals



The RAM blocks in Stratix II GX devices have local interconnects to allow ALMs and interconnects to drive into RAM blocks. The M512 RAM block local interconnect is driven by the R4, C4, and direct link interconnects from adjacent LABs. The M512 RAM blocks can communicate with LABs on either the left or right side through these row interconnects or with LAB columns on the left or right side with the column interconnects. The M512 RAM block has up to 16 direct link input connections from the left adjacent LABs and another 16 from the right adjacent LAB. M512 RAM outputs can also connect to left and right LABs through direct link interconnect. The M512 RAM block has equal opportunity for access and performance to and from LABs on either its left or right side. Figure 2–50 shows the M512 RAM block to logic array interface.

Figure 2–50. M512 RAM Block LAB Row Interface



M4K RAM Blocks

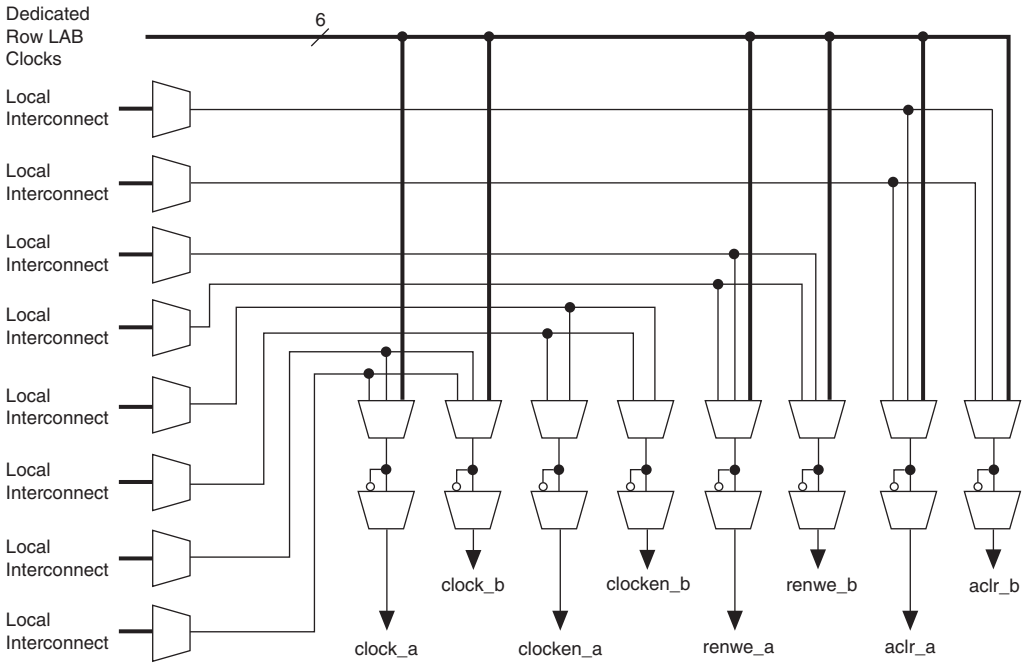
The M4K RAM block includes support for true dual-port RAM. The M4K RAM block is used to implement buffers for a wide variety of applications such as storing processor code, implementing lookup schemes, and implementing larger memory applications. Each block contains 4,608 RAM bits (including parity bits). M4K RAM blocks can be configured in the following modes:

- True dual-port RAM
- Simple dual-port RAM
- Single-port RAM
- FIFO
- ROM
- Shift register

When configured as RAM or ROM, you can use an initialization file to pre-load the memory contents.

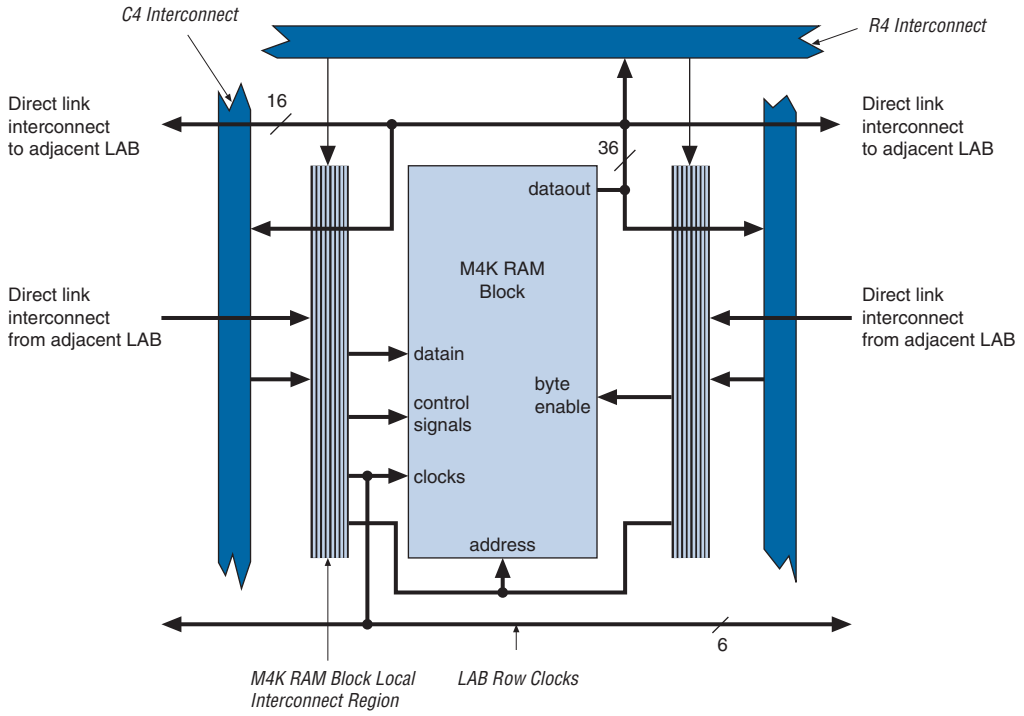
The M4K RAM blocks allow for different clocks on their inputs and outputs. Either of the two clocks feeding the block can clock M4K RAM block registers (`renwe`, `address`, `byte enable`, `datain`, and `output` registers). Only the `output` register can be bypassed. The six `labclk` signals or local interconnects can drive the control signals for the A and B ports of the M4K RAM block. ALMs can also control the `clock_a`, `clock_b`, `renwe_a`, `renwe_b`, `clr_a`, `clr_b`, `clocken_a`, and `clocken_b` signals, as shown in Figure 2–51.

Figure 2–51. M4K RAM Block Control Signals



The R4, C4, and direct link interconnects from adjacent LABs drive the M4K RAM block local interconnect. The M4K RAM blocks can communicate with LABs on either the left or right side through these row resources or with LAB columns on either the right or left with the column resources. Up to 16 direct link input connections to the M4K RAM block are possible from the left adjacent LABs and another 16 possible from the right adjacent LAB. M4K RAM block outputs can also connect to left and right LABs through direct link interconnect. Figure 2–52 shows the M4K RAM block to logic array interface.

Figure 2–52. M4K RAM Block LAB Row Interface



M-RAM Block

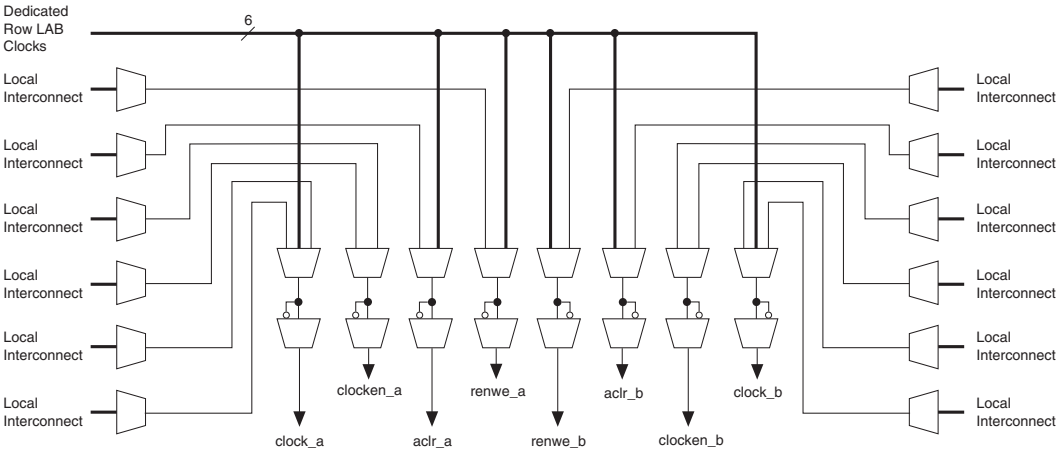
The largest TriMatrix memory block, the M-RAM block, is useful for applications where a large volume of data must be stored on-chip. Each block contains 589,824 RAM bits (including parity bits). The M-RAM block can be configured in the following modes:

- True dual-port RAM
- Simple dual-port RAM
- Single-port RAM
- FIFO

You cannot use an initialization file to initialize the contents of a M-RAM block. All M-RAM block contents power up to an undefined value. Only synchronous operation is supported in the M-RAM block, so all inputs are registered. Output registers can be bypassed.

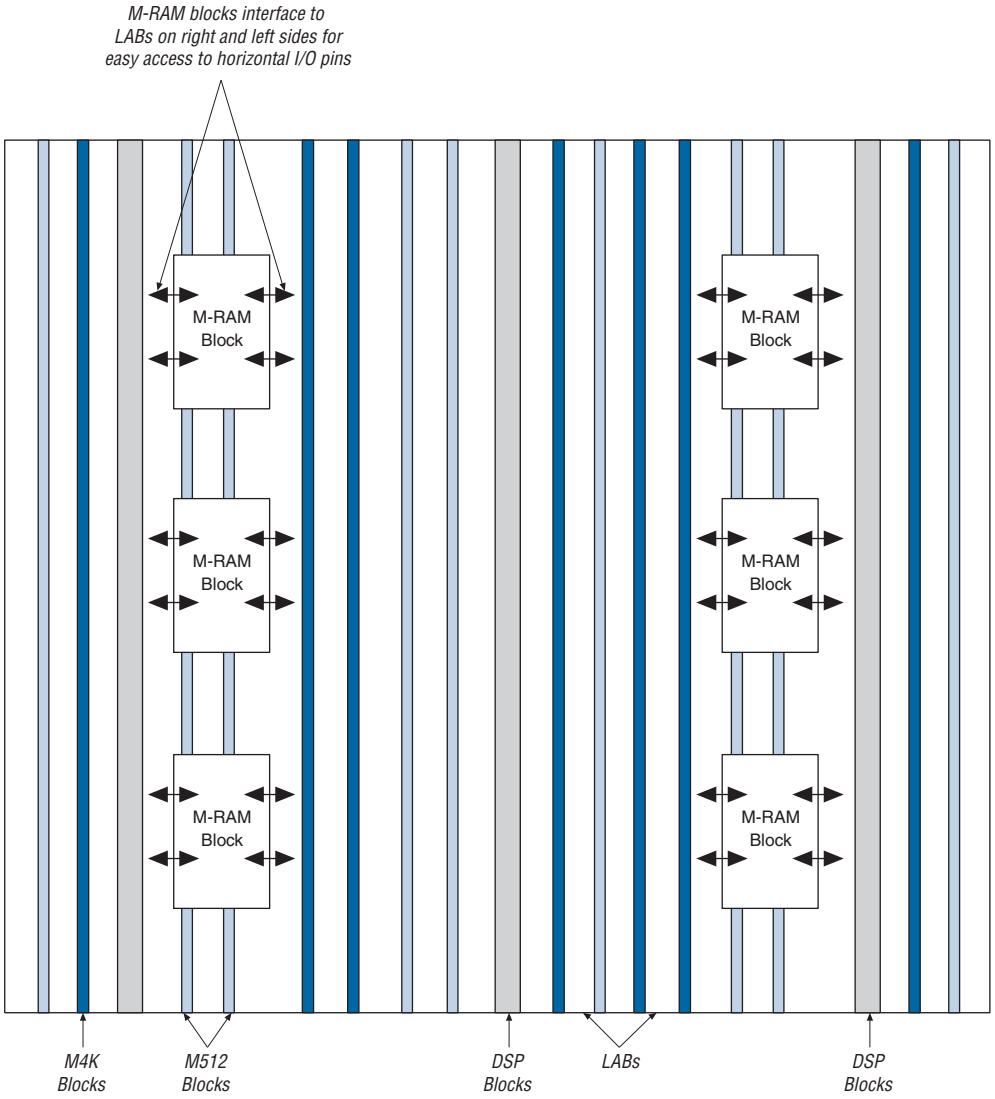
Similar to all RAM blocks, M-RAM blocks can have different clocks on their inputs and outputs. Either of the two clocks feeding the block can clock M-RAM block registers (renwe, address, byte enable, datain, and output registers). The output register can be bypassed. The six labclk signals or local interconnect can drive the control signals for the A and B ports of the M-RAM block. ALMs can also control the clock_a, clock_b, renwe_a, renwe_b, clr_a, clr_b, clocken_a, and clocken_b signals, as shown in Figure 2–53.

Figure 2–53. M-RAM Block Control Signals



The R4, R24, C4, and direct link interconnects from adjacent LABs on either the right or left side drive the M-RAM block local interconnect. Up to 16 direct link input connections to the M-RAM block are possible from the left adjacent LABs and another 16 possible from the right adjacent LAB. M-RAM block outputs can also connect to left and right LABs through direct link interconnect. Figure 2–54 shows an example floorplan for the EP2SGX130 device and the location of the M-RAM interfaces. Figures 2–55 and 2–56 show the interface between the M-RAM block and the logic array.

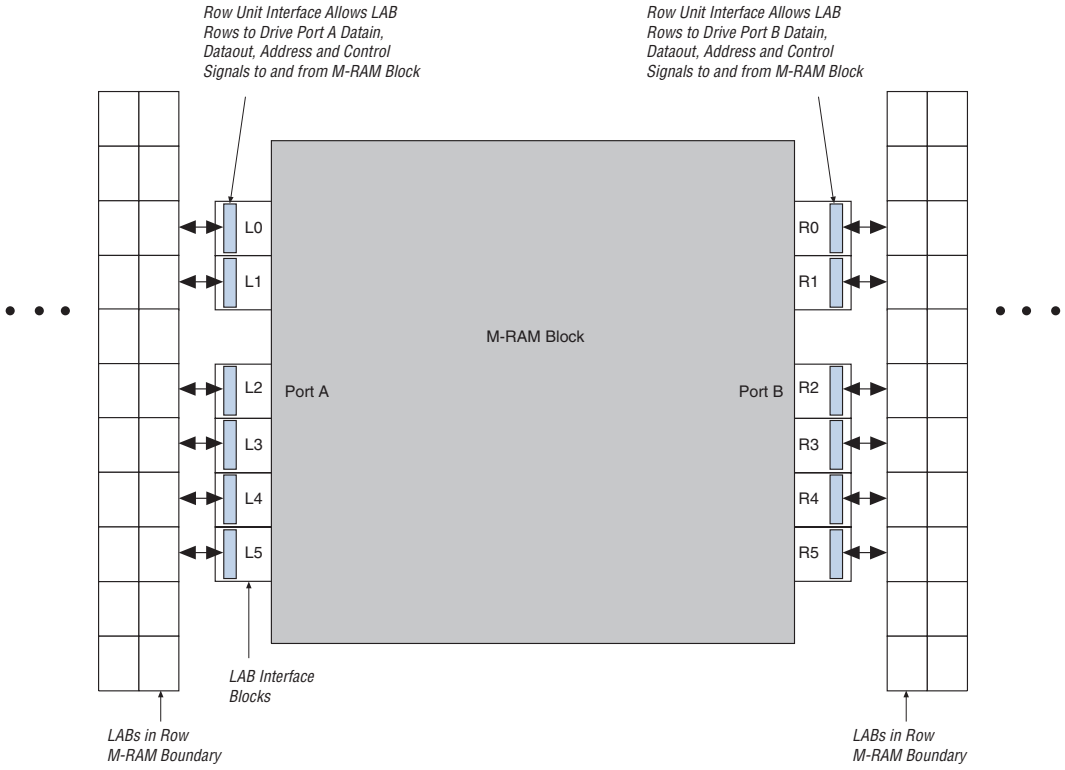
Figure 2–54. EP2SGX130 Device with M-RAM Interface Locations *Note (1)*



Note to Figure 2–54:

(1) The device shown is an EP2SGX130 device. The number and position of M-RAM blocks varies in other devices.

Figure 2–55. M-RAM Block LAB Row Interface *Note (1)*



Note to Figure 2–55:
(1) Only R24 and C16 interconnects cross the M-RAM block boundaries.

Figure 2–56. M-RAM Row Unit Interface to Interconnect

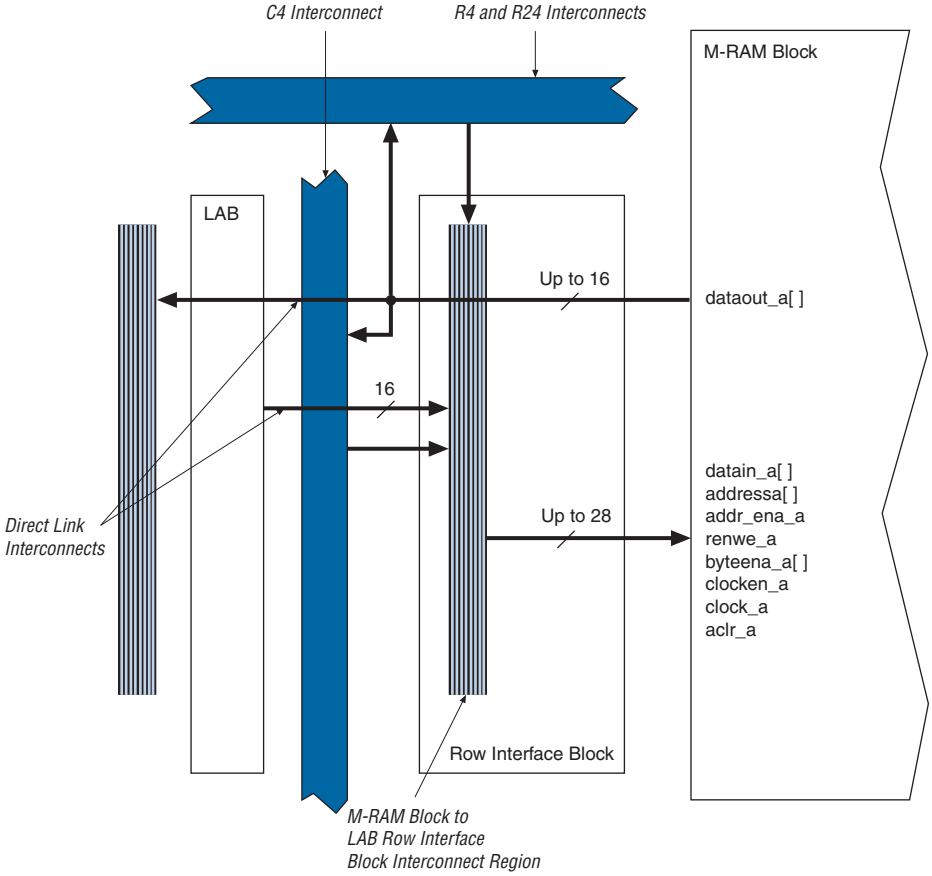


Table 2–20 shows the input and output data signal connections along with the address and control signal input connections to the row unit interfaces (L0 to L5 and R0 to R5).

Table 2–20. M-RAM Row Interface Unit Signals

Unit Interface Block	Input Signals	Output Signals
L0	datain_a[14..0] byteena_a[1..0]	dataout_a[11..0]
L1	datain_a[29..15] byteena_a[3..2]	dataout_a[23..12]
L2	datain_a[35..30] addressa[4..0] addr_ena_a clock_a clocken_a renwe_a aclr_a	dataout_a[35..24]
L3	addressa[15..5] datain_a[41..36]	dataout_a[47..36]
L4	datain_a[56..42] byteena_a[5..4]	dataout_a[59..48]
L5	datain_a[71..57] byteena_a[7..6]	dataout_a[71..60]
R0	datain_b[14..0] byteena_b[1..0]	dataout_b[11..0]
R1	datain_b[29..15] byteena_b[3..2]	dataout_b[23..12]
R2	datain_b[35..30] addressb[4..0] addr_ena_b clock_b clocken_b renwe_b aclr_b	dataout_b[35..24]
R3	addressb[15..5] datain_b[41..36]	dataout_b[47..36]
R4	datain_b[56..42] byteena_b[5..4]	dataout_b[59..48]
R5	datain_b[71..57] byteena_b[7..6]	dataout_b[71..60]

Refer to the *TriMatrix Embedded Memory Blocks in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information on TriMatrix memory.

Digital Signal Processing (DSP) Block

The most commonly used DSP functions are finite impulse response (FIR) filters, complex FIR filters, infinite impulse response (IIR) filters, fast Fourier transform (FFT) functions, direct cosine transform (DCT) functions, and correlators. All of these use the multiplier as the fundamental building block. Additionally, some applications need specialized operations such as multiply-add and multiply-accumulate operations. Stratix II GX devices provide DSP blocks to meet the arithmetic requirements of these functions.

Each Stratix II GX device has two to four columns of DSP blocks to efficiently implement DSP functions faster than ALM-based implementations. Stratix II GX devices have up to 24 DSP blocks per column (see Table 2–21). Each DSP block can be configured to support up to:

- Eight 9 × 9-bit multipliers
- Four 18 × 18-bit multipliers
- One 36 × 36-bit multiplier

As indicated, the Stratix II GX DSP block can support one 36 × 36-bit multiplier in a single DSP block, and is true for any combination of signed, unsigned, or mixed sign multiplications.

Figures 2–57 shows one of the columns with surrounding LAB rows.

Figure 2–57. DSP Blocks Arranged in Columns

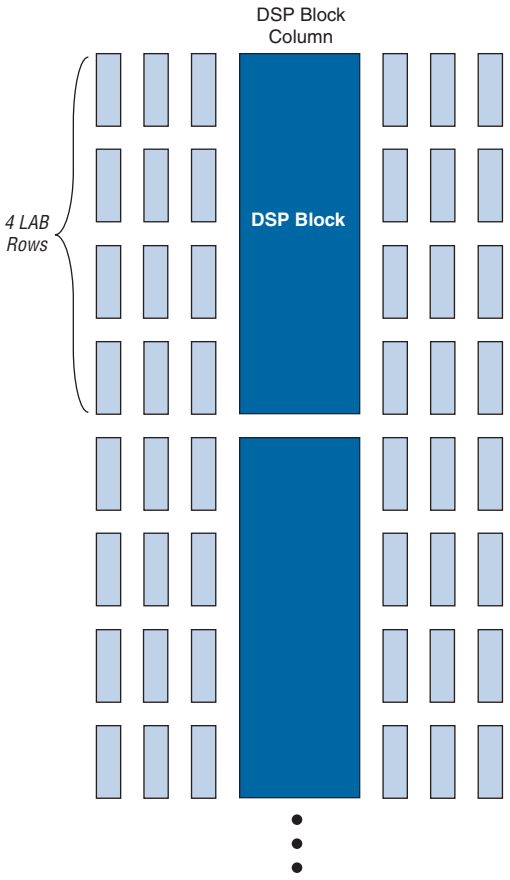


Table 2–21 shows the number of DSP blocks in each Stratix II GX device. DSP block multipliers can optionally feed an adder/subtractor or accumulator in the block, depending on the configuration, which makes routing to ALMs easier, saves ALM routing resources, and increases performance because all connections and blocks are in the DSP block.

Table 2–21. DSP Blocks in Stratix II GX Devices *Note (1)*

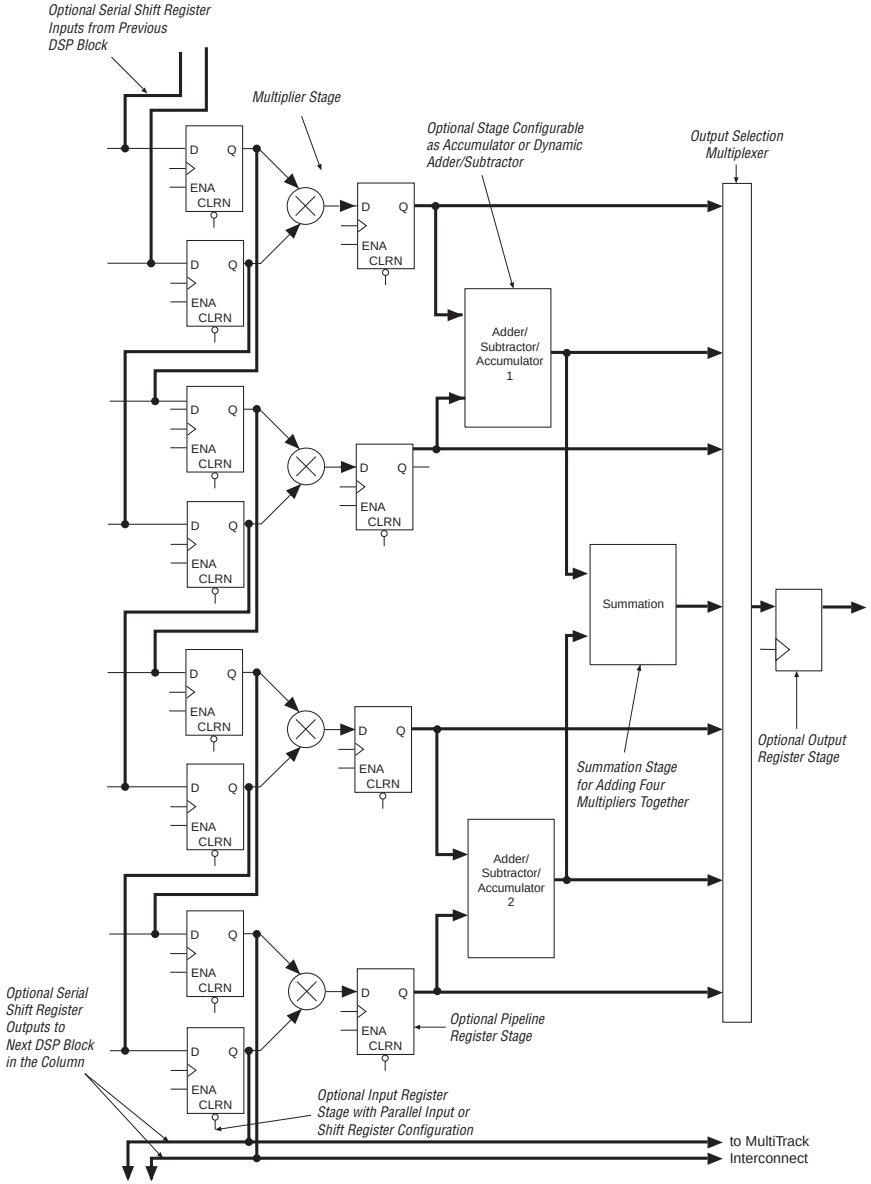
Device	DSP Blocks	Total 9 × 9 Multipliers	Total 18 × 18 Multipliers	Total 36 × 36 Multipliers
EP2SGX30	16	128	64	16
EP2SGX60	36	288	144	36
EP2SGX90	48	384	192	48
EP2SGX130	63	504	252	63

Note to Table 2–21:

(1) This list only shows functions that can fit into a single DSP block. Multiple DSP blocks can support larger multiplication functions.

Additionally, the DSP block input registers can efficiently implement shift registers for FIR filter applications, and DSP blocks support Q1.15 format rounding and saturation. Figure 2–58 shows the top-level diagram of the DSP block configured for 18 × 18-bit multiplier mode.

Figure 2–58. DSP Block Diagram for 18 × 18-Bit Configuration



Modes of Operation

The adder, subtractor, and accumulate functions of a DSP block have four modes of operation:

- Simple multiplier
- Multiply-accumulator
- Two-multipliers adder
- Four-multipliers adder

Table 2–22 shows the different number of multipliers possible in each DSP block mode according to size. These modes allow the DSP blocks to implement numerous applications for DSP including FFTs, complex FIR, FIR, 2D FIR filters, equalizers, IIR, correlators, matrix multiplication, and many other functions. The DSP blocks also support mixed modes and mixed multiplier sizes in the same block. For example, half of one DSP block can implement one 18 × 18-bit multiplier in multiply-accumulator mode, while the other half of the DSP block implements four 9 × 9-bit multipliers in simple multiplier mode.

Table 2–22. Multiplier Size and Configurations per DSP Block

DSP Block Mode	9 × 9	18 × 18	36 × 36
Multiplier	Eight multipliers with eight product outputs	Four multipliers with four product outputs	One multiplier with one product output
Multiply-accumulator	—	Two 52-bit multiply-accumulate blocks	—
Two-multipliers adder	Four two-multiplier adder (two 9 × 9 complex multiply)	Two two-multiplier adder (one 18 × 18 complex multiply)	—
Four-multipliers adder	Two four-multiplier adder	One four-multiplier adder	—

DSP Block Interface

The Stratix II GX device DSP block input registers can generate a shift register that can cascade down in the same DSP block column. Dedicated connections between DSP blocks provide fast connections between the shift register inputs to cascade the shift register chains. You can cascade registers within multiple DSP blocks for 9 × 9- or 18 × 18-bit FIR filters larger than four taps, with additional adder stages implemented in ALMs. If the DSP block is configured as 36 × 36 bits, the adder, subtractor, or accumulator stages are implemented in ALMs. Each DSP block can route the shift register chain out of the block to cascade multiple columns of DSP blocks.

The DSP block is divided into four block units that interface with four LAB rows on the left and right. Each block unit can be considered one complete 18 × 18-bit multiplier with 36 inputs and 36 outputs. A local interconnect region is associated with each DSP block. Like a LAB, this interconnect region can be fed with 16 direct link interconnects from the LAB to the left or right of the DSP block in the same row. R4 and C4 routing resources can access the DSP block's local interconnect region. The outputs also work similarly to LAB outputs. Eighteen outputs from the DSP block can drive to the left LAB through direct link interconnects and 18 can drive to the right LAB through direct link interconnects. All 36 outputs can drive to R4 and C4 routing interconnects. Outputs can drive right- or left-column routing.

Figures 2–59 and 2–60 show the DSP block interfaces to LAB rows.

Figure 2–59. DSP Block Interconnect Interface

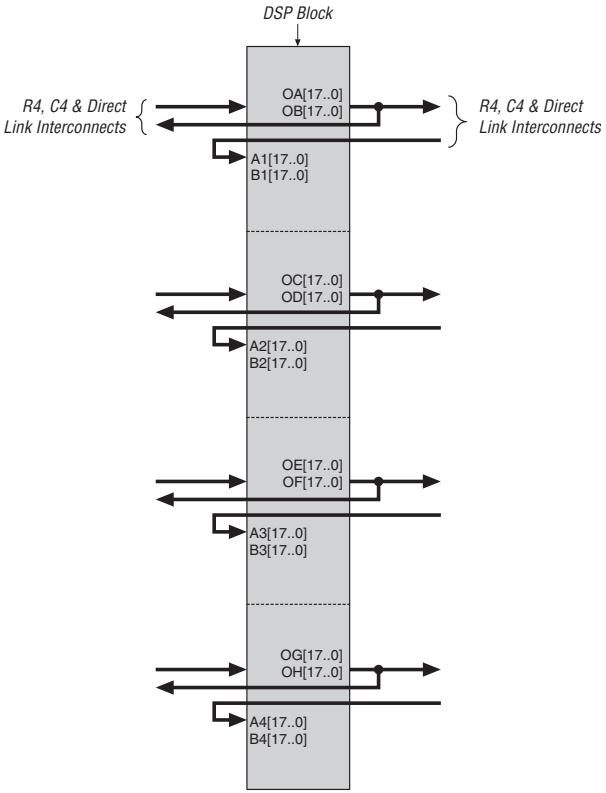
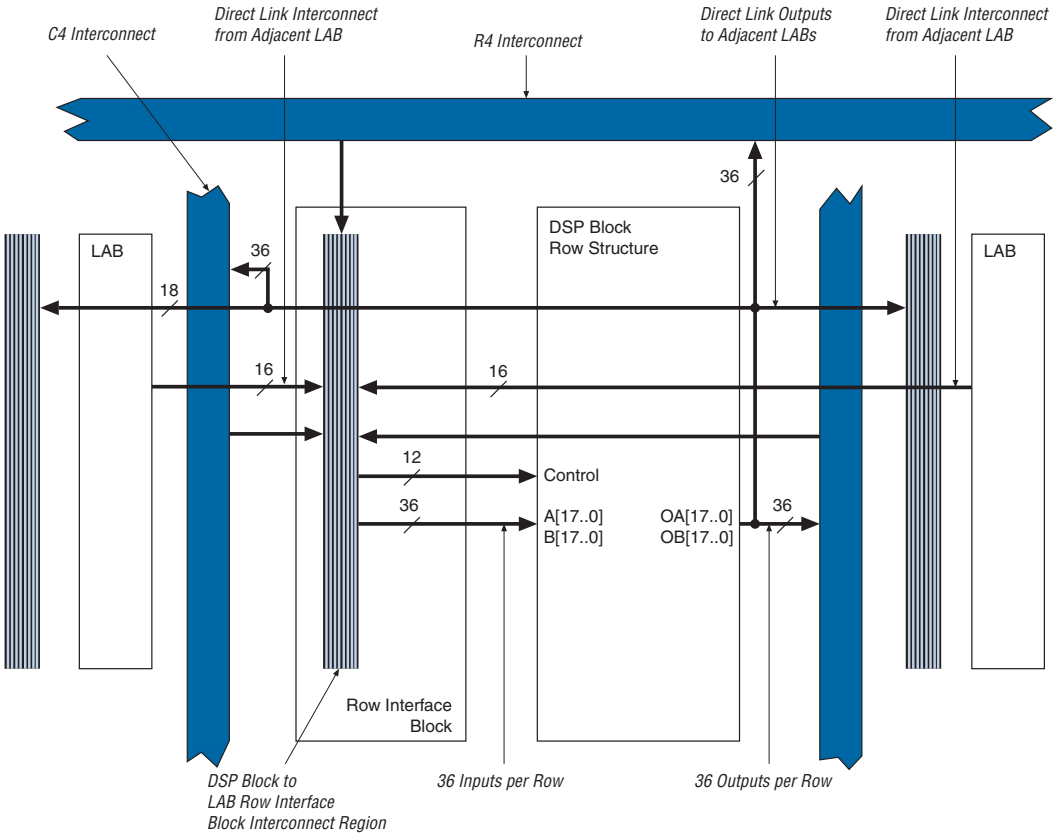


Figure 2–60. DSP Block Interface to Interconnect



A bus of 44 control signals feeds the entire DSP block. These signals include clocks, asynchronous clears, clock enables, signed and unsigned control signals, addition and subtraction control signals, rounding and saturation control signals, and accumulator synchronous loads. The clock signals are routed from LAB row clocks and are generated from specific LAB rows at the DSP block interface. The LAB row source for control signals, data inputs, and outputs is shown in Table 2–23.

Refer to the *DSP Blocks in Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information on DSP blocks.

Table 2–23. DSP Block Signal Sources and Destinations

LAB Row at Interface	Control Signals Generated	Data Inputs	Data Outputs
0	clock0 aclr0 ena0 mult01_saturate addnsub1_round/ accum_round addnsub1 signa sourcea sourceb	A1[17..0] B1[17..0]	OA[17..0] OB[17..0]
1	clock1 aclr1 ena1 accum_saturate mult01_round accum_sload sourcea sourceb mode0	A2[17..0] B2[17..0]	OC[17..0] OD[17..0]
2	clock2 aclr2 ena2 mult23_saturate addnsub3_round/ accum_round addnsub3 sign_b sourcea sourceb	A3[17..0] B3[17..0]	OE[17..0] OF[17..0]
3	clock3 aclr3 ena3 accum_saturate mult23_round accum_sload sourcea sourceb mode1	A4[17..0] B4[17..0]	OG[17..0] OH[17..0]

PLLs and Clock Networks

Stratix II GX devices provide a hierarchical clock structure and multiple phase-locked loops (PLLs) with advanced features. The large number of clocking resources in combination with the clock synthesis precision provided by enhanced and fast PLLs provides a complete clock management solution.

Global and Hierarchical Clocking

Stratix II GX devices provide 16 dedicated global clock networks and 32 regional clock networks (eight per device quadrant). These clocks are organized into a hierarchical clock structure that allows for up to 24 clocks per device region with low skew and delay. This hierarchical clocking scheme provides up to 48 unique clock domains in Stratix II GX devices.

There are 12 dedicated clock pins to drive either the global or regional clock networks. Four clock pins drive each side of the device, as shown in Figures 2–61 and 2–62. Internal logic and enhanced and fast PLL outputs can also drive the global and regional clock networks. Each global and regional clock has a clock control block, which controls the selection of the clock source and dynamically enables or disables the clock to reduce power consumption. Table 2–24 shows global and regional clock features.

Table 2–24. Global and Regional Clock Features

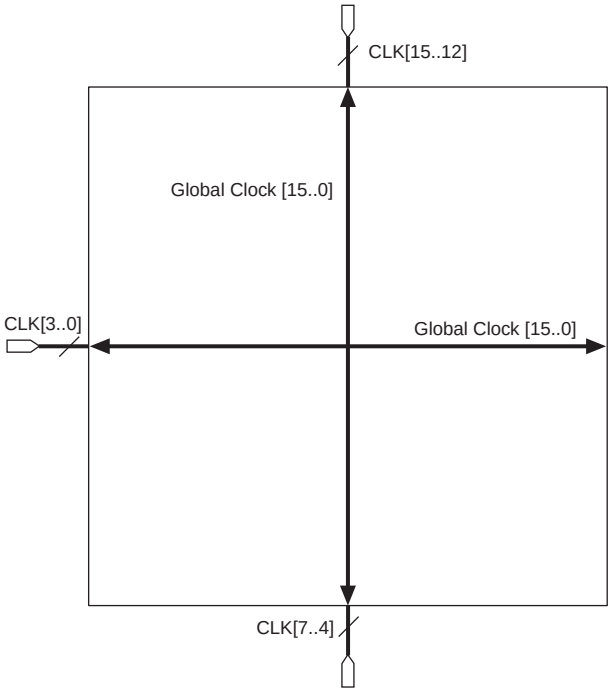
Feature	Global Clocks	Regional Clocks
Number per device	16	32
Number available per quadrant	16	8
Sources	Clock pins, PLL outputs, core routings, inter-transceiver clocks	Clock pins, PLL outputs, core routings, inter-transceiver clocks
Dynamic clock source selection	✓	—
Dynamic enable/disable	✓	✓

Global Clock Network

These clocks drive throughout the entire device, feeding all device quadrants. The global clock networks can be used as clock sources for all resources in the device IOEs, ALMs, DSP blocks, and all memory blocks. These resources can also be used for control signals, such as clock enables and synchronous or asynchronous clears fed from the external pin. The global clock networks can also be driven by internal logic for internally

generated global clocks and asynchronous clears, clock enables, or other control signals with large fanout. Figure 2–61 shows the 12 dedicated CLK pins driving global clock networks.

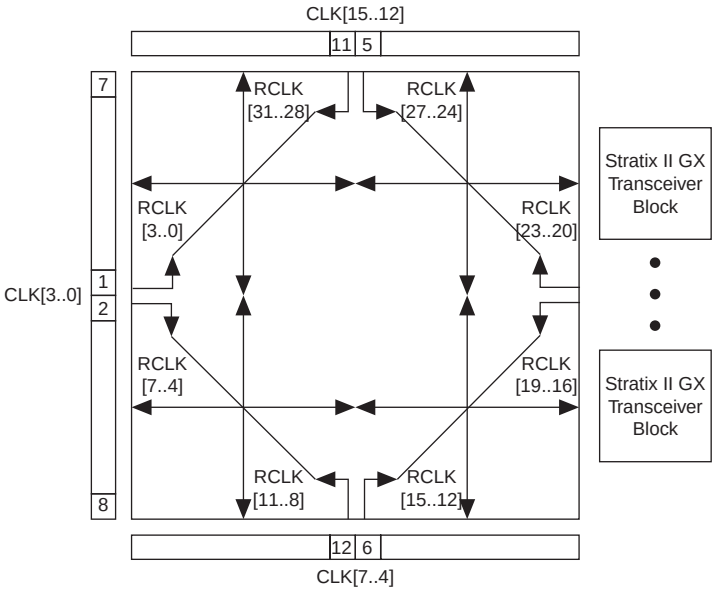
Figure 2–61. Global Clocking



Regional Clock Network

There are eight regional clock networks (RCLK[7..0]) in each quadrant of the Stratix II GX device that are driven by the dedicated CLK[15..12] and CLK[7..0] input pins, by PLL outputs, or by internal logic. The regional clock networks provide the lowest clock delay and skew for logic contained in a single quadrant. The CLK pins symmetrically drive the RCLK networks in a particular quadrant, as shown in Figure 2–62.

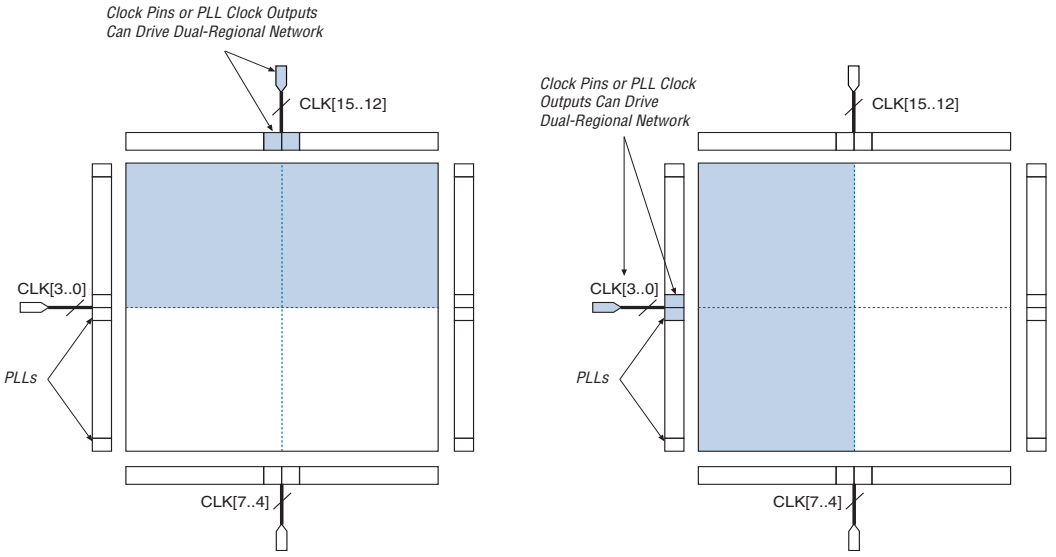
Figure 2–62. Regional Clocks



Dual-Regional Clock Network

A single source (CLK pin or PLL output) can generate a dual-regional clock by driving two regional clock network lines in adjacent quadrants (one from each quadrant), which allows logic that spans multiple quadrants to utilize the same low skew clock. The routing of this clock signal on an entire side has approximately the same speed but slightly higher clock skew when compared with a clock signal that drives a single quadrant. Internal logic-array routing can also drive a dual-regional clock. Clock pins and enhanced PLL outputs on the top and bottom can drive horizontal dual-regional clocks. Clock pins and fast PLL outputs on the left and right can drive vertical dual-regional clocks, as shown in Figure 2–63. Corner PLLs cannot drive dual-regional clocks.

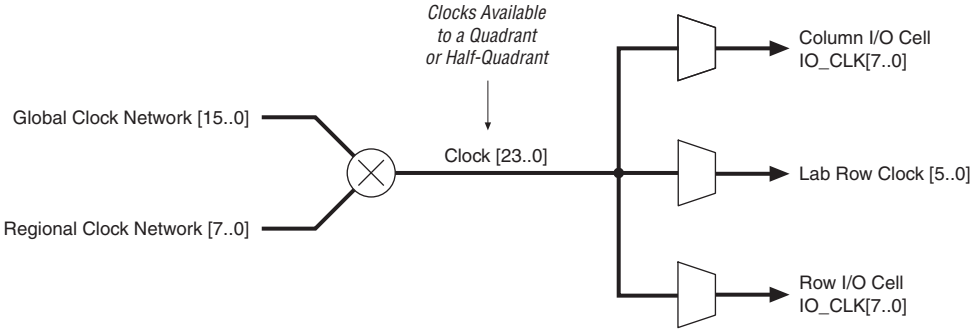
Figure 2–63. Dual-Regional Clocks



Combined Resources

Within each quadrant, there are 24 distinct dedicated clocking resources consisting of 16 global clock lines and 8 regional clock lines. Multiplexers are used with these clocks to form buses to drive LAB row clocks, column IOE clocks, or row IOE clocks. Another multiplexer is used at the LAB level to select three of the six row clocks to feed the ALM registers in the LAB (see Figure 2–64).

Figure 2–64. Hierarchical Clock Networks per Quadrant



IOE clocks have row and column block regions that are clocked by 8 I/O clock signals chosen from the 24 quadrant clock resources. Figures 2–65 and 2–66 show the quadrant relationship to the I/O clock regions.

Figure 2–65. EP2SGX30 Device I/O Clock Groups

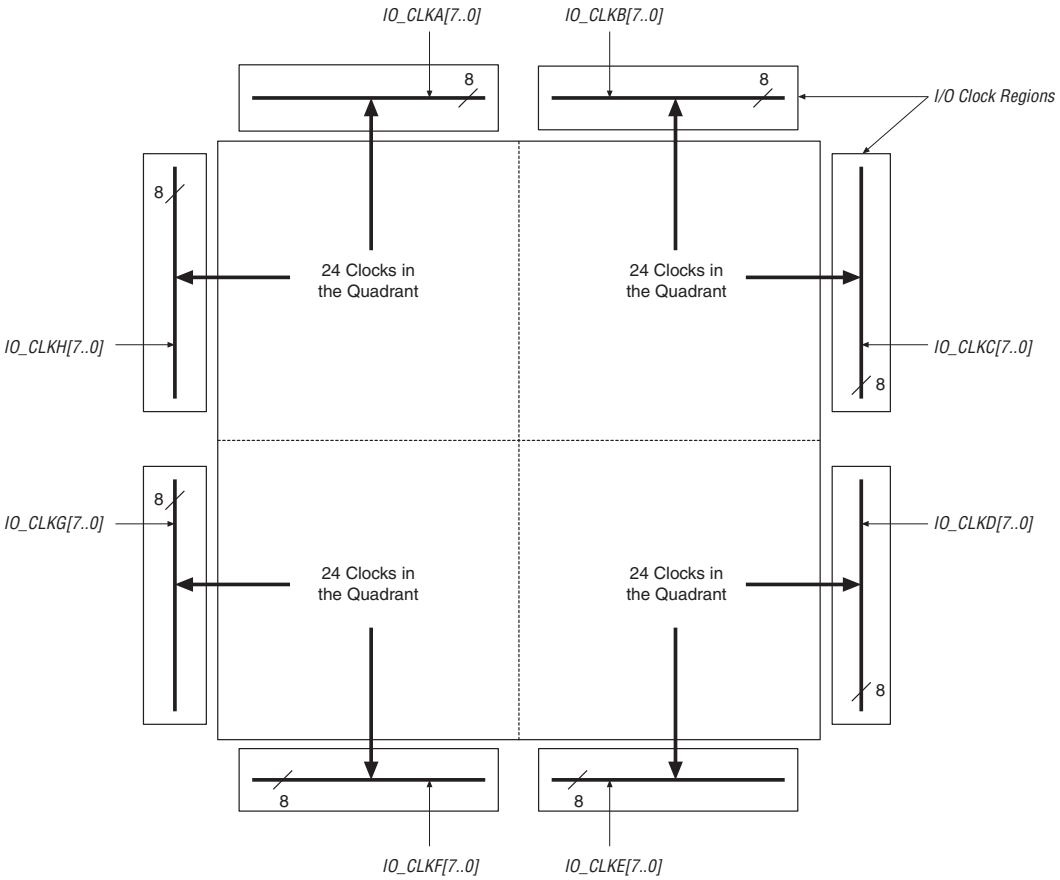
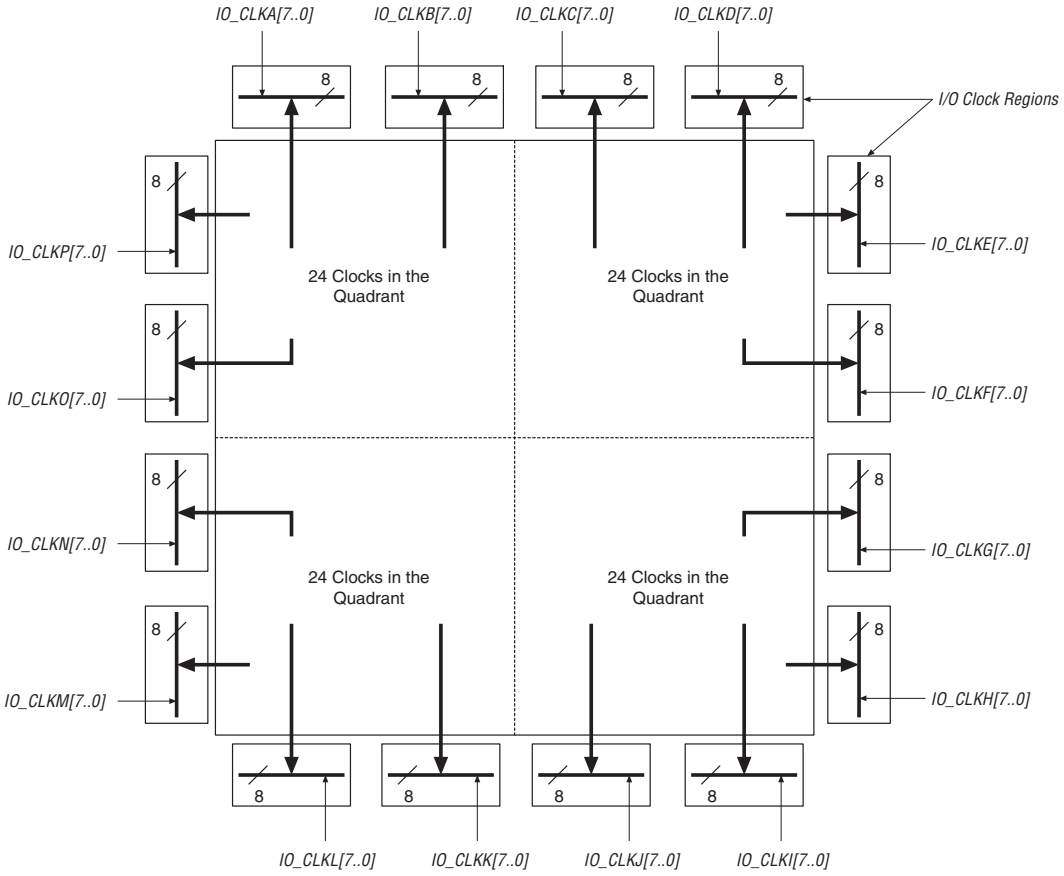


Figure 2–66. EP2SGX60, EP2SGX90 and EP2SGX130 Device I/O Clock Groups



You can use the Quartus II software to control whether a clock input pin drives either a global, regional, or dual-regional clock network. The Quartus II software automatically selects the clocking resources if not specified.

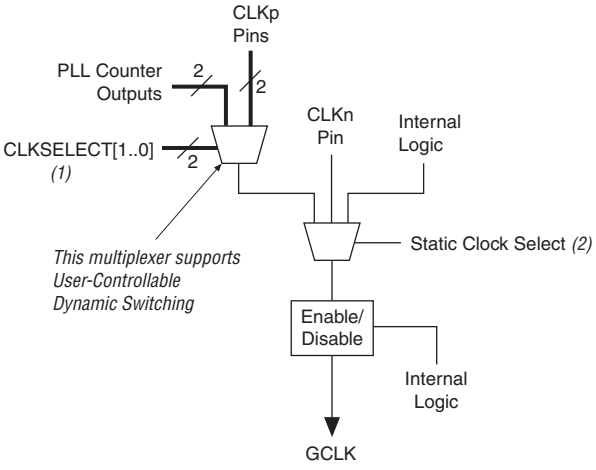
Clock Control Block

Each global clock, regional clock, and PLL external clock output has its own clock control block. The control block has two functions:

- Clock source selection (dynamic selection for global clocks)
- Clock power-down (dynamic clock enable or disable)

Figures 2–67 through 2–69 show the clock control block for the global clock, regional clock, and PLL external clock output, respectively.

Figure 2–67. Global Clock Control Blocks

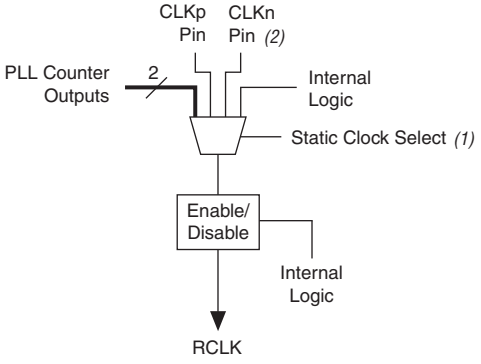


Notes to Figure 2–67:

(1) These clock select signals can be dynamically controlled through internal logic when the device is operating in user mode.

(2) These clock select signals can only be set through a configuration file (SRAM Object File [**.sof**] or Programmer Object File [**.pof**]) and cannot be dynamically controlled during user mode operation.

Figure 2–68. Regional Clock Control Blocks

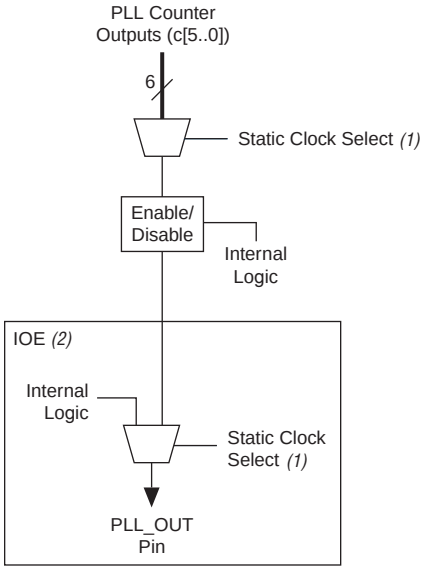


Notes to Figure 2–68:

(1) These clock select signals can only be set through a configuration file (**.sof** or **.pof**) and cannot be dynamically controlled during user mode operation.

(2) Only the `CLKn` pins on the top and bottom of the device feed to regional clock select.

Figure 2–69. External PLL Output Clock Control Blocks



Notes to Figure 2–69:

(1) These clock select signals can only be set through a configuration file (**.sof** or **.pof**) and cannot be dynamically controlled during user mode operation.

(2) The clock control block feeds to a multiplexer within the PLL_OUT pin's IOE. The PLL_OUT pin is a dual-purpose pin. Therefore, this multiplexer selects either an internal signal or the output of the clock control block.

For the global clock control block, the clock source selection can be controlled either statically or dynamically. You have the option of statically selecting the clock source by using the Quartus II software to set specific configuration bits in the configuration file (**.sof** or **.pof**) or you can control the selection dynamically by using internal logic to drive the multiplexer select inputs. When selecting statically, the clock source can be set to any of the inputs to the select multiplexer. When selecting the clock source dynamically, you can either select between two PLL outputs (such as the C0 or C1 outputs from one PLL), between two PLLs (such as the C0/C1 clock output of one PLL or the C0/C1 clock output of the other PLL), between two clock pins (such as CLK0 or CLK1), or between a combination of clock pins or PLL outputs.

For the regional and PLL_OUT clock control block, the clock source selection can only be controlled statically using configuration bits. Any of the inputs to the clock select multiplexer can be set as the clock source.

The Stratix II GX clock networks can be disabled (powered down) by both static and dynamic approaches. When a clock net is powered down, all the logic fed by the clock net is in an off-state, thereby reducing the overall power consumption of the device. The global and regional clock networks can be powered down statically through a setting in the configuration file (**.sof** or **.pof**). Clock networks that are not used are automatically powered down through configuration bit settings in the configuration file generated by the Quartus II software. The dynamic clock enable and disable feature allows the internal logic to control power up and down synchronously on GCLK and RCLK nets and PLL_OUT pins. This function is independent of the PLL and is applied directly on the clock network or PLL_OUT pin, as shown in Figures 2–67 through 2–69.

Enhanced and Fast PLLs

Stratix II GX devices provide robust clock management and synthesis using up to four enhanced PLLs and four fast PLLs. These PLLs increase performance and provide advanced clock interfacing and clock frequency synthesis. With features such as clock switchover, spread spectrum clocking, reconfigurable bandwidth, phase control, and reconfigurable phase shifting, the Stratix II GX device's enhanced PLLs provide you with complete control of clocks and system timing. The fast PLLs provide general purpose clocking with multiplication and phase shifting as well as high-speed outputs for high-speed differential I/O support. Enhanced and fast PLLs work together with the Stratix II GX high-speed I/O and advanced clock architecture to provide significant improvements in system performance and bandwidth.

The Quartus II software enables the PLLs and their features without requiring any external devices. Table 2–25 shows the PLLs available for each Stratix II GX device and their type.

Table 2–25. Stratix II GX Device PLL Availability *Notes (1), (2)*

Device	Fast PLLs								Enhanced PLLs			
	1	2	3 *(3)*	4 *(3)*	7	8	9 *(3)*	10 *(3)*	5	6	11	12
EP2SGX30	✓	✓							✓	✓		
EP2SGX60	✓	✓			✓	✓			✓	✓	✓	✓
EP2SGX90	✓	✓			✓	✓			✓	✓	✓	✓
EP2SGX130	✓	✓			✓	✓			✓	✓	✓	✓

Notes to Table 2–25:

(1) EP2SGX30C/D and EP2SGX60C/D devices only have two fast PLLs (1 and 2), but the connectivity from these two PLLs to the global and regional clock networks remains the same as shown. The EP2S60C/D devices only have two enhanced PLLs (5 and 6).

(2) The global or regional clocks in a fast PLL's quadrant can drive the fast PLL input. A dedicated clock input pin or other PLL must drive the global or regional source. The source cannot be driven by internally generated logic before driving the fast PLL.

(3) PLLs 3, 4, 9, and 10 are not available in Stratix II GX devices. However, these PLLs are listed in Table 2–25 because the Stratix II GX PLL numbering scheme is consistent with Stratix and Stratix II devices.

Table 2–26 shows the enhanced PLL and fast PLL features in Stratix II GX devices.

Table 2–26. Stratix II GX PLL Features

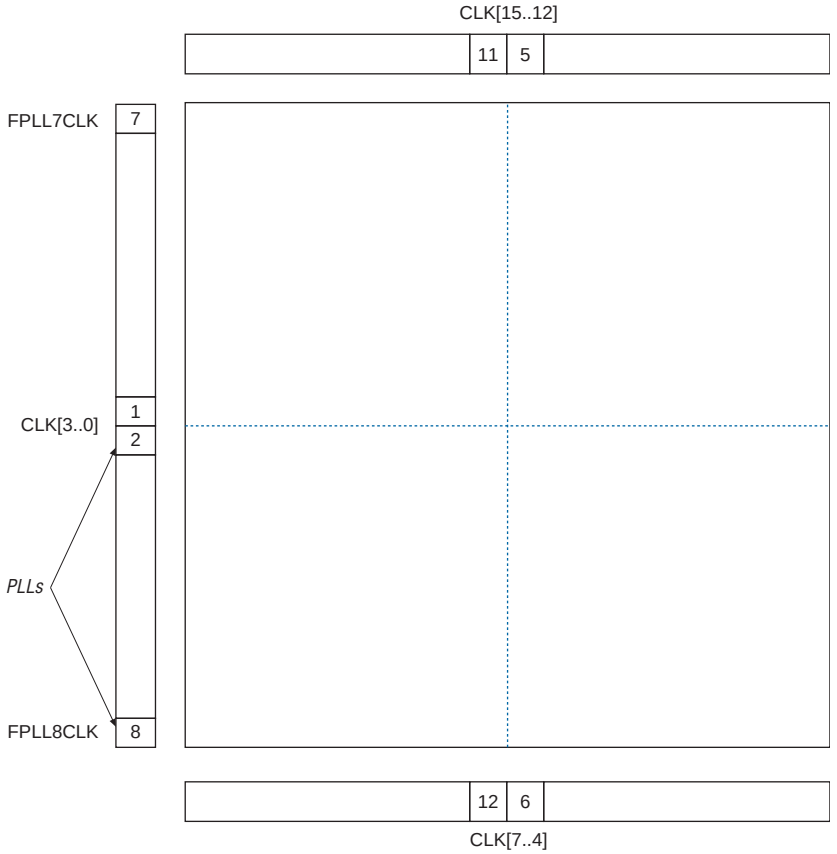
Feature	Enhanced PLL	Fast PLL
Clock multiplication and division	*m*/(*n* × post-scale counter) *(1)*	*m*/(*n* × post-scale counter) *(2)*
Phase shift	Down to 125-ps increments *(3)*, *(4)*	Down to 125-ps increments *(3)*, *(4)*
Clock switchover	✓	✓ *(5)*
PLL reconfiguration	✓	✓
Reconfigurable bandwidth	✓	✓
Spread spectrum clocking	✓	
Programmable duty cycle	✓	✓
Number of internal clock outputs	6	4
Number of external clock outputs	Three differential/six single-ended	*(6)*
Number of feedback clock inputs	One single-ended or differential *(7)*, *(8)*	

Notes to Table 2–26:

(1) For enhanced PLLs, *m*, *n* range from 1 to 256 and post-scale counters range from 1 to 512 with 50% duty cycle.
(2) For fast PLLs, *m*, and post-scale counters range from 1 to 32. The *n* counter ranges from 1 to 4.
(3) The smallest phase shift is determined by the voltage controlled oscillator (VCO) period divided by 8.
(4) For degree increments, Stratix II GX devices can shift all output frequencies in increments of at least 45. Smaller degree increments are possible depending on the frequency and divide parameters.
(5) Stratix II GX fast PLLs only support manual clock switchover.
(6) Fast PLLs can drive to any I/O pin as an external clock. For high-speed differential I/O pins, the device uses a data channel to generate `txclkout`.
(7) If the feedback input is used, you will lose one (or two, if f_{BIN} is differential) external clock output pin.
(8) Every Stratix II GX device has at least two enhanced PLLs with one single-ended or differential external feedback input per PLL.

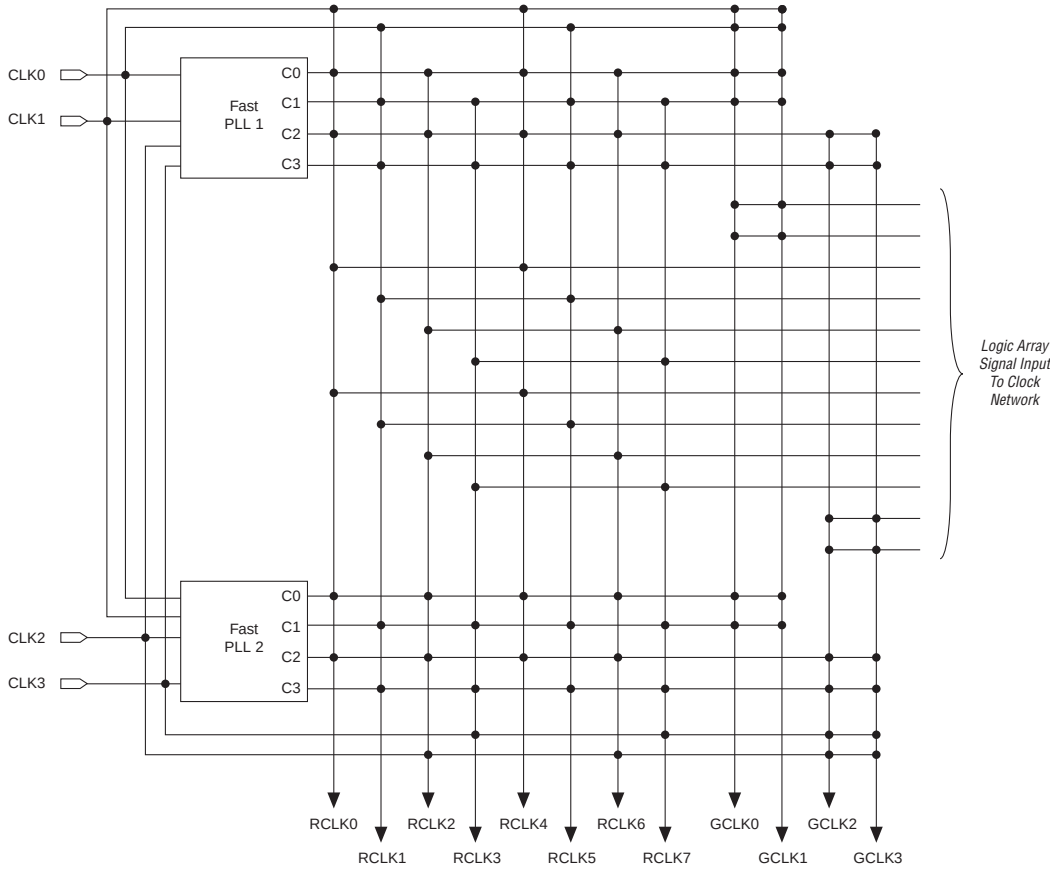
Figure 2–70 shows a top-level diagram of the Stratix II GX device and PLL floorplan.

Figure 2–70. PLL Locations



Figures 2–71 and 2–72 shows global and regional clocking from the fast PLL outputs and the side clock pins. The connections to the global and regional clocks from the fast PLL outputs, internal drivers, and the CLK pins on the left side of the device are shown in Table 2–27.

Figure 2–71. Global and Regional Clock Connections from Center Clock Pins and Fast PLL Outputs Notes (1), (2)

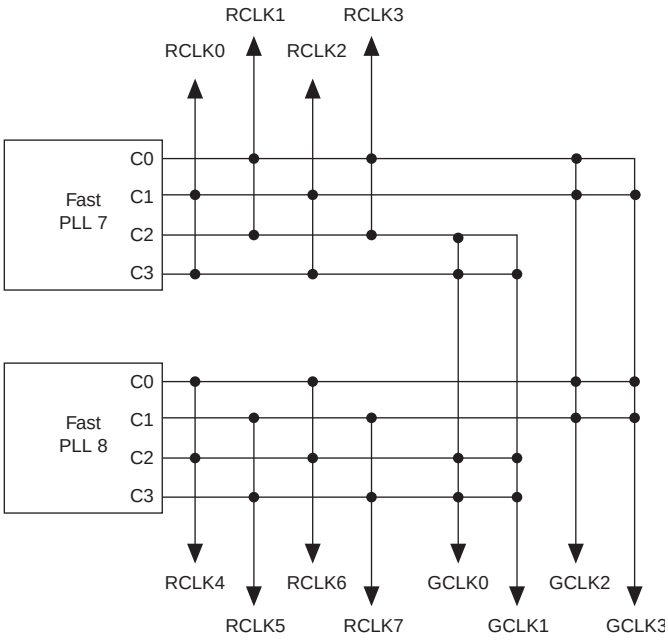


Notes to Figure 2–71:

(1) EP2SGX30C/D and P2SGX60C/D devices only have two fast PLLs (1 and 2) and two Enhanced PLLs (5 and 6), but the connectivity from these PLLs to the global and regional clock networks remains the same as shown.

(2) The global or regional clocks in a fast PLL's quadrant can drive the fast PLL input. A dedicated clock input pin or other PLL must drive the global or regional source. The source cannot be driven by internally generated logic before driving the fast PLL.

Figure 2–72. Global and Regional Clock Connections from Corner Clock Pins and Fast PLL Outputs Notes (1), (2)



Notes to Figure 2–72:

(1) The global or regional clocks in a fast PLL's quadrant can drive the fast PLL input. A dedicated clock input pin or other PLL must drive the global or regional source. The source cannot be driven by internally generated logic before driving the fast PLL.

(2) EP2SGX30C/D and EP2SGX60C/D devices only have two fast PLLs (1 and 2); they do not contain corner fast PLLs.

Table 2–27. Global and Regional Clock Connections from Left Side Clock Pins and Fast PLL Outputs (Part 1 of 3)

Left Side Global and Regional Clock Network Connectivity	CLK0	CLK1	CLK2	CLK3	RCLK0	RCLK1	RCLK2	RCLK3	RCLK4	RCLK5	RCLK6	RCLK7
Clock pins												
CLK0p	✓	✓			✓				✓			
CLK1p	✓	✓				✓				✓		
CLK2p			✓	✓			✓				✓	
CLK3p			✓	✓				✓				✓

Table 2–27. Global and Regional Clock Connections from Left Side Clock Pins and Fast PLL Outputs (Part 2 of 3)

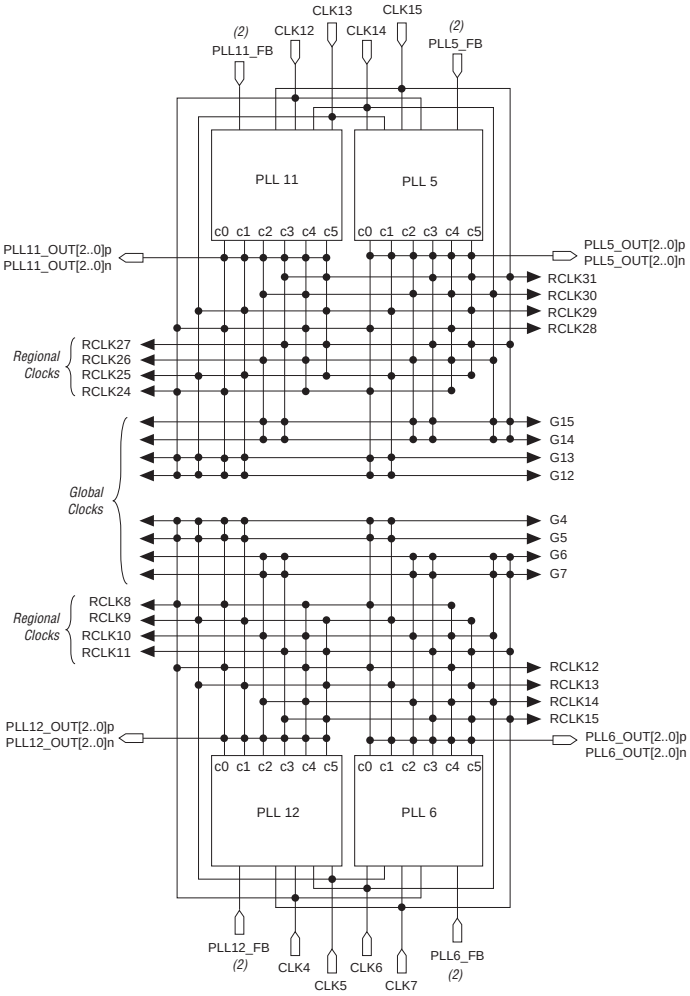
Left Side Global and Regional Clock Network Connectivity	CLK0	CLK1	CLK2	CLK3	RCLK0	RCLK1	RCLK2	RCLK3	RCLK4	RCLK5	RCLK6	RCLK7
Drivers from internal logic												
GCLKDRV0	✓	✓										
GCLKDRV1	✓	✓										
GCLKDRV2			✓	✓								
GCLKDRV3			✓	✓								
RCLKDRV0					✓				✓			
RCLKDRV1						✓				✓		
RCLKDRV2							✓				✓	
RCLKDRV3								✓				✓
RCLKDRV4					✓				✓			
RCLKDRV5						✓				✓		
RCLKDRV6							✓				✓	
RCLKDRV7								✓				✓
PLL 1 outputs												
c0	✓	✓			✓		✓		✓		✓	
c1	✓	✓				✓		✓		✓		✓
c2			✓	✓	✓		✓		✓		✓	
c3			✓	✓		✓		✓		✓		✓
PLL 2 outputs												
c0	✓	✓				✓		✓		✓		✓
c1	✓	✓			✓		✓		✓		✓	
c2			✓	✓		✓		✓		✓		✓
c3			✓	✓	✓		✓		✓		✓	
PLL 7 outputs												
c0			✓	✓		✓		✓				
c1			✓	✓	✓		✓					
c2	✓	✓				✓		✓				
c3	✓	✓			✓		✓					

Table 2–27. Global and Regional Clock Connections from Left Side Clock Pins and Fast PLL Outputs (Part 3 of 3)

Left Side Global and Regional Clock Network Connectivity	CLK0	CLK1	CLK2	CLK3	RCLK0	RCLK1	RCLK2	RCLK3	RCLK4	RCLK5	RCLK6	RCLK7
PLL 8 outputs												
c0			✓	✓					✓		✓	
c1			✓	✓						✓		✓
c2	✓	✓							✓		✓	
c3	✓	✓								✓		✓

Figure 2–73 shows the global and regional clocking from enhanced PLL outputs and top and bottom CLK pins.

Figure 2–73. Global and Regional Clock Connections from Top and Bottom Clock Pins and Enhanced PLL Outputs *Notes (1), (2)*



Notes to Figure 2–73:

(1) EP2SGX30C/D and EP2SGX60C/D devices only have two enhanced PLLs (5 and 6), but the connectivity from these two PLLs to the global and regional clock networks remains the same as shown.

(2) If the design uses the feedback input, you will lose one (or two, if FBIN is differential) external clock output pin.

The connections to the global and regional clocks from the top clock pins and enhanced PLL outputs are shown in Table 2–28. The connections to the clocks from the bottom clock pins are shown in Table 2–29.

Table 2–28. Global and Regional Clock Connections from Top Clock Pins and Enhanced PLL Outputs (Part 1 of 2)

Top Side Global and Regional Clock Network Connectivity	DLLCLK	CLK12	CLK13	CLK14	CLK15	RCLK24	RCLK25	RCLK26	RCLK27	RCLK28	RCLK29	RCLK30	RCLK31
Clock pins													
CLK12p	✓	✓	✓			✓				✓			
CLK13p	✓	✓	✓				✓				✓		
CLK14p	✓			✓	✓			✓				✓	
CLK15p	✓			✓	✓				✓				✓
CLK12n		✓				✓				✓			
CLK13n			✓				✓				✓		
CLK14n				✓				✓				✓	
CLK15n					✓				✓				✓
Drivers from internal logic													
GCLKDRV0		✓											
GCLKDRV1			✓										
GCLKDRV2				✓									
GCLKDRV3					✓								
RCLKDRV0						✓				✓			
RCLKDRV1							✓				✓		
RCLKDRV2								✓				✓	
RCLKDRV3									✓				✓
RCLKDRV4						✓				✓			
RCLKDRV5							✓				✓		
RCLKDRV6								✓				✓	
RCLKDRV7									✓				✓
Enhanced PLL5 outputs													
c0	✓	✓	✓			✓				✓			
c1	✓	✓	✓				✓				✓		

Table 2–28. Global and Regional Clock Connections from Top Clock Pins and Enhanced PLL Outputs (Part 2 of 2)

Top Side Global and Regional Clock Network Connectivity	DLLCLK	CLK12	CLK13	CLK14	CLK15	RCLK24	RCLK25	RCLK26	RCLK27	RCLK28	RCLK29	RCLK30	RCLK31
c2	✓			✓	✓			✓				✓	
c3	✓			✓	✓				✓				✓
c4	✓					✓		✓		✓		✓	
c5	✓						✓		✓		✓		✓
Enhanced PLL 11 outputs													
c0		✓	✓			✓				✓			
c1		✓	✓				✓				✓		
c2				✓	✓			✓				✓	
c3				✓	✓				✓				✓
c4						✓		✓		✓		✓	
c5							✓		✓		✓		✓

Table 2–29. Global and Regional Clock Connections from Bottom Clock Pins and Enhanced PLL Outputs (Part 1 of 2)

Bottom Side Global and Regional Clock Network Connectivity	DLLCLK	CLK4	CLK5	CLK6	CLK7	RCLK8	RCLK9	RCLK10	RCLK11	RCLK12	RCLK13	RCLK14	RCLK15
Clock pins													
CLK4p	✓	✓	✓			✓				✓			
CLK5p	✓	✓	✓				✓				✓		
CLK6p	✓			✓	✓			✓				✓	
CLK7p	✓			✓	✓				✓				✓
CLK4n		✓				✓				✓			
CLK5n			✓				✓				✓		
CLK6n				✓				✓				✓	
CLK7n					✓				✓				✓
Drivers from internal logic													
GCLKDRV0		✓											
GCLKDRV1			✓										

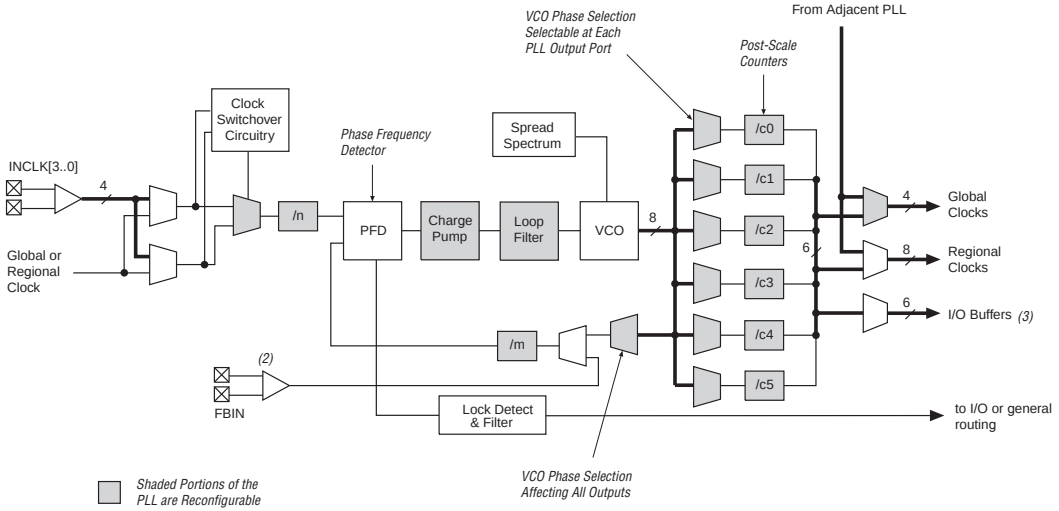
Table 2–29. Global and Regional Clock Connections from Bottom Clock Pins and Enhanced PLL Outputs (Part 2 of 2)

Bottom Side Global and Regional Clock Network Connectivity	DLLCLK	CLK4	CLK5	CLK6	CLK7	RCLK8	RCLK9	RCLK10	RCLK11	RCLK12	RCLK13	RCLK14	RCLK15
GCLKDRV2				✓									
GCLKDRV3					✓								
RCLKDRV0						✓				✓			
RCLKDRV1							✓				✓		
RCLKDRV2								✓				✓	
RCLKDRV3									✓				✓
RCLKDRV4						✓				✓			
RCLKDRV5							✓				✓		
RCLKDRV6								✓				✓	
RCLKDRV7									✓				✓
Enhanced PLL 6 outputs													
c0	✓	✓	✓			✓				✓			
c1	✓	✓	✓				✓				✓		
c2	✓			✓	✓			✓				✓	
c3	✓			✓	✓				✓				✓
c4	✓					✓		✓		✓		✓	
c5	✓						✓		✓		✓		✓
Enhanced PLL 12 outputs													
c0		✓	✓			✓				✓			
c1		✓	✓				✓				✓		
c2				✓	✓			✓				✓	
c3				✓	✓				✓				✓
c4						✓		✓		✓		✓	
c5							✓		✓		✓		✓

Enhanced PLLs

Stratix II GX devices contain up to four enhanced PLLs with advanced clock management features. These features include support for external clock feedback mode, spread-spectrum clocking, and counter cascading. Figure 2–74 shows a diagram of the enhanced PLL.

Figure 2–74. Stratix II GX Enhanced PLL *Note (1)*



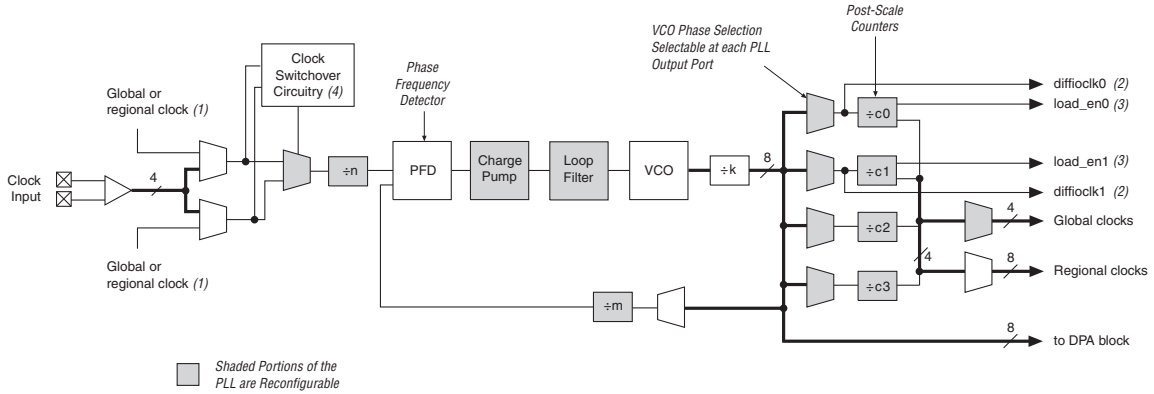
Notes to Figure 2–74:

(1) Each clock source can come from any of the four clock pins that are physically located on the same side of the device as the PLL.
(2) If the feedback input is used, you will lose one (or two, if FBIN is differential) external clock output pin.
(3) Each enhanced PLL has three differential external clock outputs or six single-ended external clock outputs.
(4) The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

Fast PLLs

Stratix II GX devices contain up to four fast PLLs with high-speed serial interfacing ability. The fast PLLs offer high-speed outputs to manage the high-speed differential I/O interfaces. Figure 2–75 shows a diagram of the fast PLL.

Figure 2–75. Stratix II GX Device Fast PLL



Notes to Figure 2–75:

(1) The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

(2) In high-speed differential I/O support mode, this high-speed PLL clock feeds the serializer/deserializer (SERDES) circuitry. Stratix II GX devices only support one rate of data transfer per fast PLL in high-speed differential I/O support mode.

(3) This signal is a differential I/O SERDES control signal.

(4) Stratix II GX fast PLLs only support manual clock switchover.

Refer to the *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information on enhanced and fast PLLs. Refer to "High-Speed Differential I/O with DPA Support" on page 2–136 for more information on high-speed differential I/O support.

I/O Structure

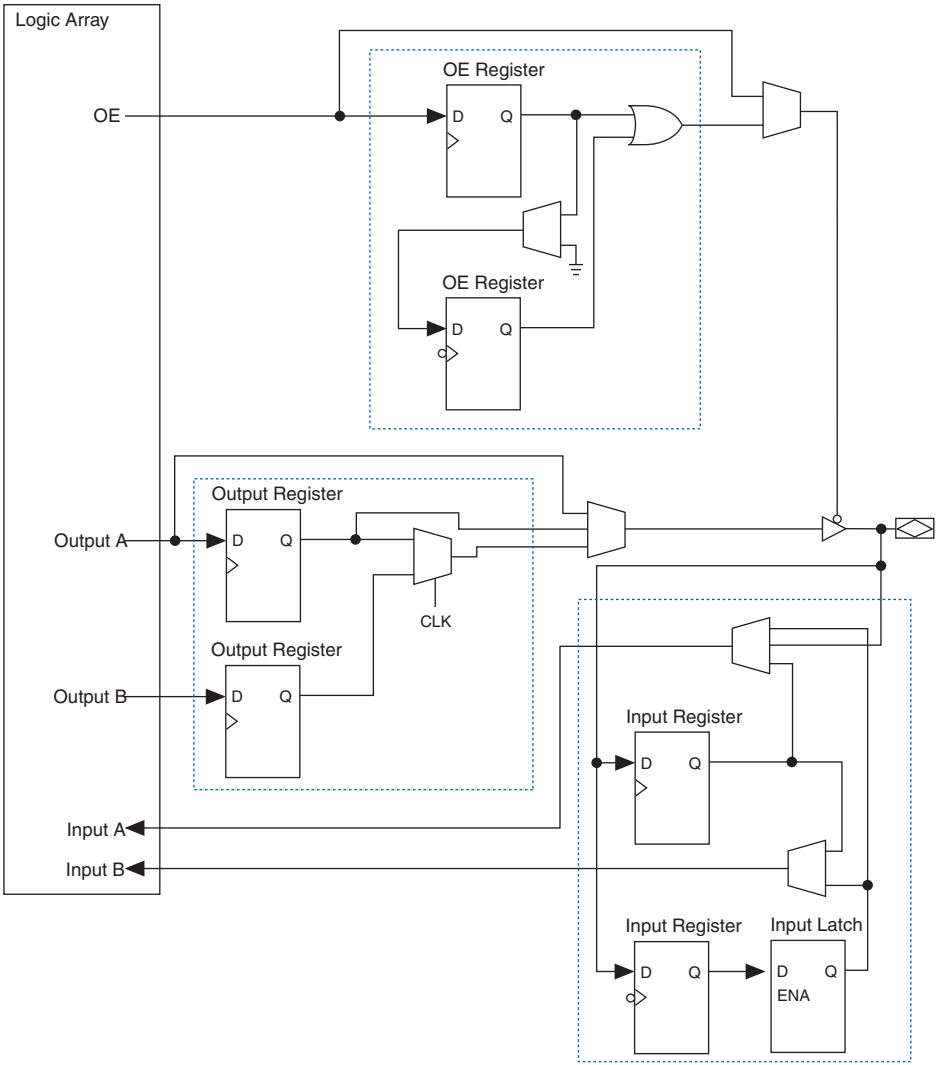
The Stratix II GX IOEs provide many features, including:

- Dedicated differential and single-ended I/O buffers
- 3.3-V, 64-bit, 66-MHz PCI compliance
- 3.3-V, 64-bit, 133-MHz PCI-X 1.0 compliance
- Joint Test Action Group (JTAG) boundary-scan test (BST) support
- On-chip driver series termination
- On-chip termination for differential standards
- Programmable pull-up during configuration
- Output drive strength control
- Tri-state buffers
- Bus-hold circuitry
- Programmable pull-up resistors
- Programmable input and output delays

- Open-drain outputs
- DQ and DQS I/O pins
- Double data rate (DDR) registers

The IOE in Stratix II GX devices contains a bidirectional I/O buffer, six registers, and a latch for a complete embedded bidirectional single data rate or DDR transfer. Figure 2–76 shows the Stratix II GX IOE structure. The IOE contains two input registers (plus a latch), two output registers, and two output enable registers. You can use both input registers and the latch to capture DDR input and both output registers to drive DDR outputs. Additionally, you can use the output enable (OE) register for fast clock-to-output enable timing. The negative edge-clocked OE register is used for DDR SDRAM interfacing. The Quartus II software automatically duplicates a single OE register that controls multiple output or bidirectional pins.

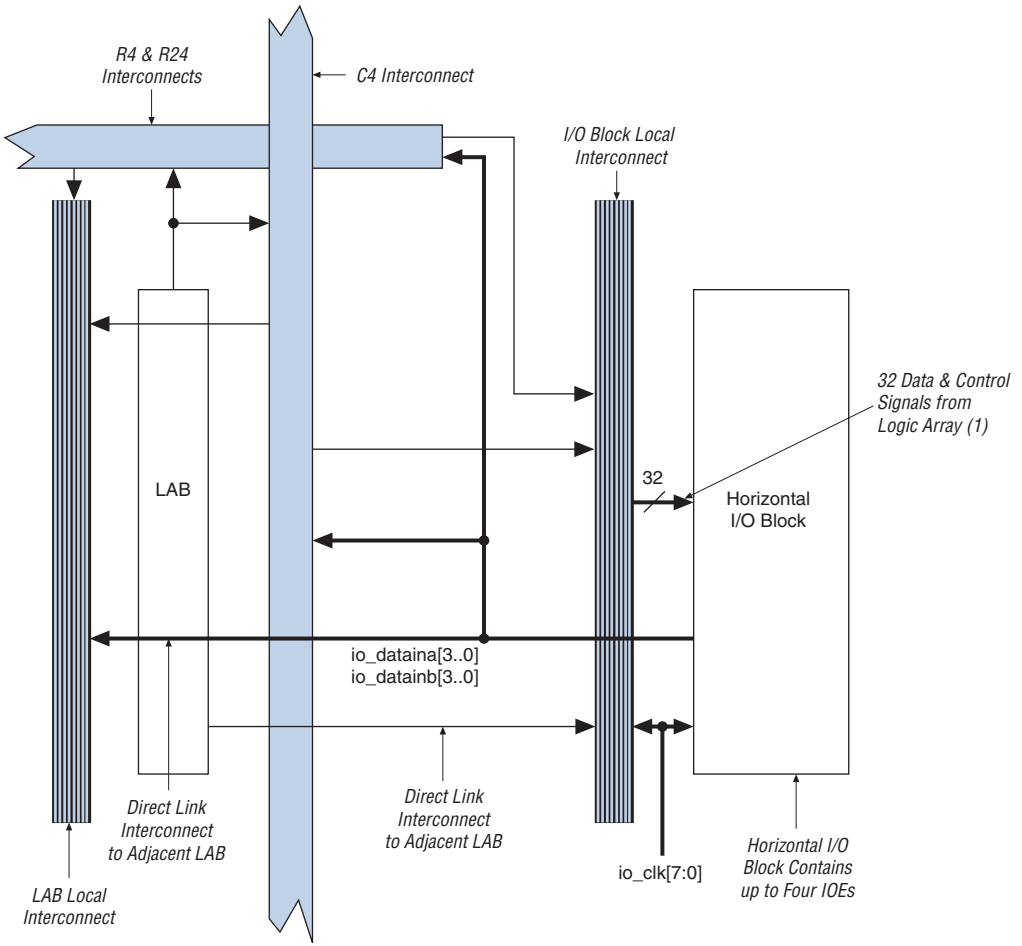
Figure 2–76. Stratix II GX IOE Structure



The IOEs are located in I/O blocks around the periphery of the Stratix II GX device. There are up to four IOEs per row I/O block and four IOEs per column I/O block. The row I/O blocks drive row, column, or direct link interconnects. The column I/O blocks drive column interconnects.

Figure 2–77 shows how a row I/O block connects to the logic array.

Figure 2–77. Row I/O Block Connection to the Interconnect

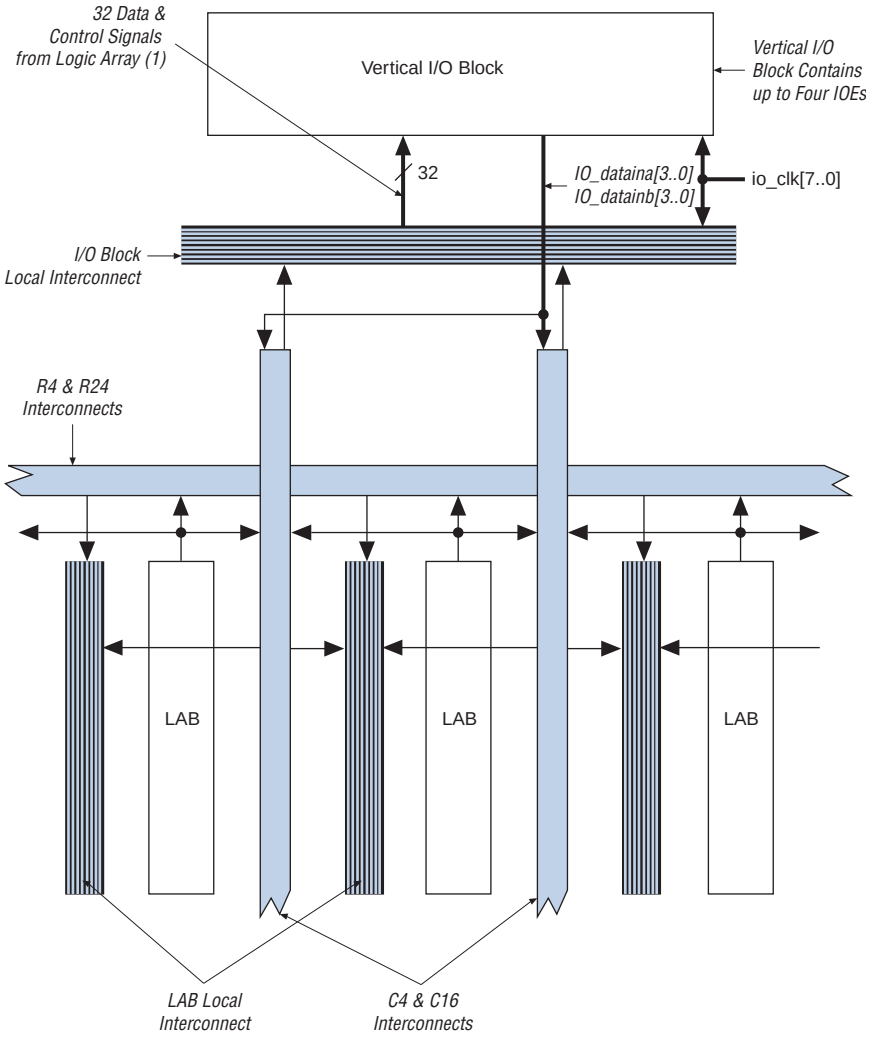


Note to Figure 2–77:

(1) The 32 data and control signals consist of eight data out lines: four lines each for DDR applications `io_dataouta[3..0]` and `io_dataoutb[3..0]`, four output enables `io_oe[3..0]`, four input clock enables `io_ce_in[3..0]`, four output clock enables `io_ce_out[3..0]`, four clocks `io_clk[3..0]`, four asynchronous clear and preset signals `io_aclr/apreset[3..0]`, and four synchronous clear and preset signals `io_sclr/spreset[3..0]`.

Figure 2–78 shows how a column I/O block connects to the logic array.

Figure 2–78. Column I/O Block Connection to the Interconnect



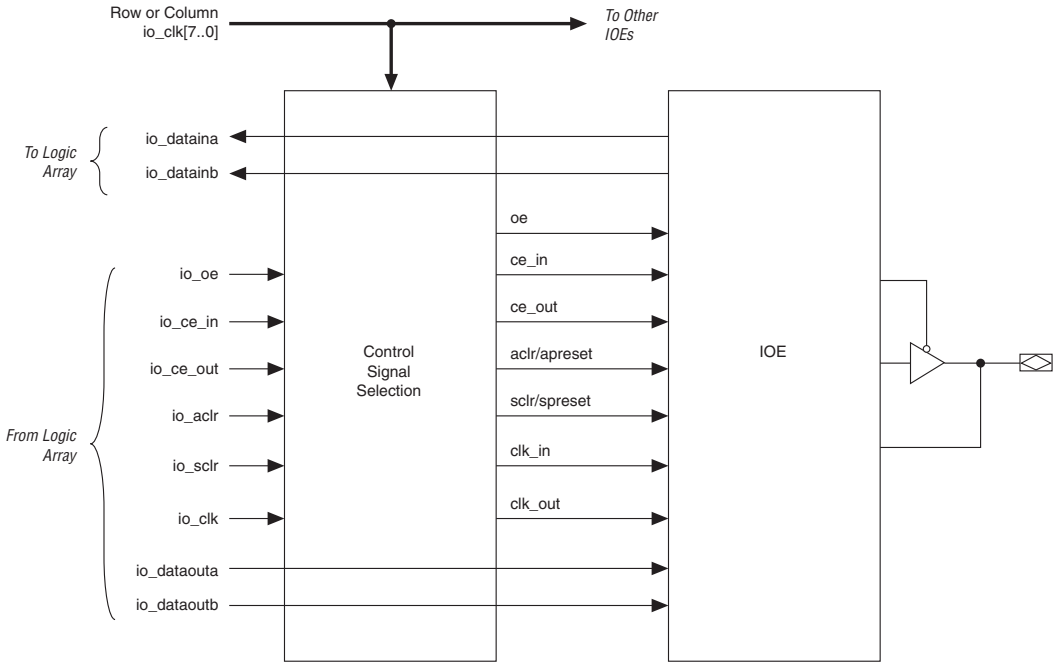
Note to Figure 2–78:

(1) The 32 data and control signals consist of eight data out lines: four lines each for DDR applications `io_dataouta[3..0]` and `io_dataoutb[3..0]`, four output enables `io_oe[3..0]`, four input clock enables `io_ce_in[3..0]`, four output clock enables `io_ce_out[3..0]`, four clocks `io_clk[3..0]`, four asynchronous clear and preset signals `io_aclr/apreset[3..0]`, and four synchronous clear and preset signals `io_sclr/spreset[3..0]`.

There are 32 control and data signals that feed each row or column I/O block. These control and data signals are driven from the logic array. The row or column IOE clocks, `io_clk[7..0]`, provide a dedicated routing resource for low-skew, high-speed clocks. I/O clocks are generated from global or regional clocks. Refer to "PLLs and Clock Networks" on page 2–89 for more information.

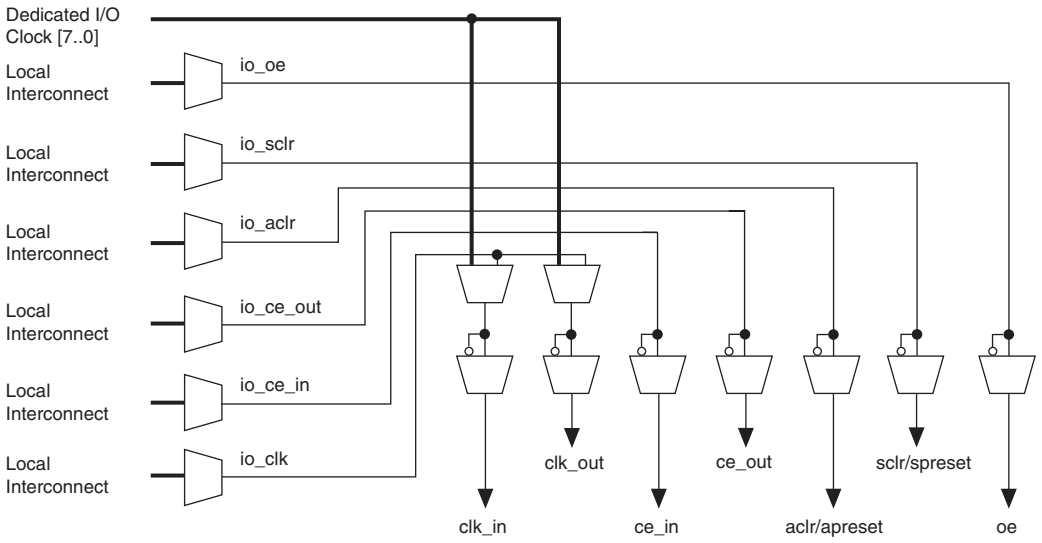
Figure 2–79 illustrates the signal paths through the I/O block.

Figure 2–79. Signal Path Through the I/O Block



Each IOE contains its own control signal selection for the following control signals: `oe`, `ce_in`, `ce_out`, `aclr/apreset`, `sclr/spreset`, `clk_in`, and `clk_out`. Figure 2–80 illustrates the control signal selection.

Figure 2–80. Control Signal Selection per IOE *Note (1)*

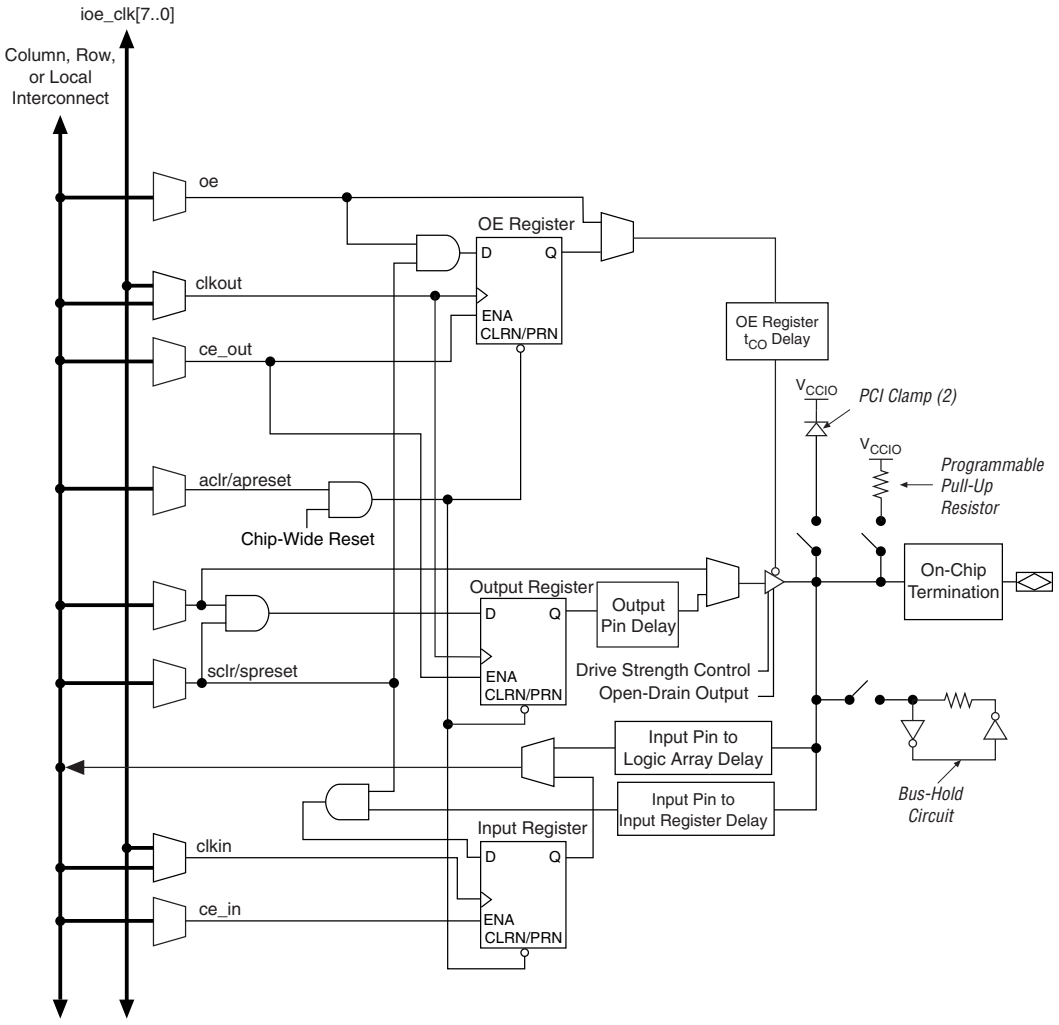


Note to Figure 2–80:

(1) Control signals `ce_in`, `ce_out`, `aclr/apreset`, `sclr/spreset`, and `oe` can be global signals even though their control selection multiplexers are not directly fed by the `ioe_clk[7..0]` signals. The `ioe_clk` signals can drive the I/O local interconnect, which then drives the control selection multiplexers.

In normal bidirectional operation, you can use the input register for input data requiring fast setup times. The input register can have its own clock input and clock enable separate from the OE and output registers. The output register can be used for data requiring fast clock-to-output performance. You can use the OE register for fast clock-to-output enable timing. The OE and output register share the same clock source and the same clock enable source from local interconnect in the associated LAB, dedicated I/O clocks, and the column and row interconnects. Figure 2–81 shows the IOE in bidirectional configuration.

Figure 2–81. Stratix II GX IOE in Bidirectional I/O Configuration *Note (1)*



Notes to Figure 2–81:
(1) All input signals to the IOE can be inverted at the IOE.
(2) The optional PCI clamp is only available on column I/O pins.

The Stratix II GX device IOE includes programmable delays that can be activated to ensure input IOE register-to-logic array register transfers, input pin-to-logic array register transfers, or output IOE register-to-pin transfers.

A path in which a pin directly drives a register can require the delay to ensure zero hold time, whereas a path in which a pin drives a register through combinational logic may not require the delay. Programmable delays exist for decreasing input-pin-to-logic-array and IOE input register delays. The Quartus II Compiler can program these delays to automatically minimize setup time while providing a zero hold time. Programmable delays can increase the register-to-pin delays for output and/or output enable registers. Programmable delays are no longer required to ensure zero hold times for logic array register-to-IOE register transfers. The Quartus II Compiler can create the zero hold time for these transfers. Table 2–30 shows the programmable delays for Stratix II GX devices.

Table 2–30. Stratix II GX Programmable Delay Chain

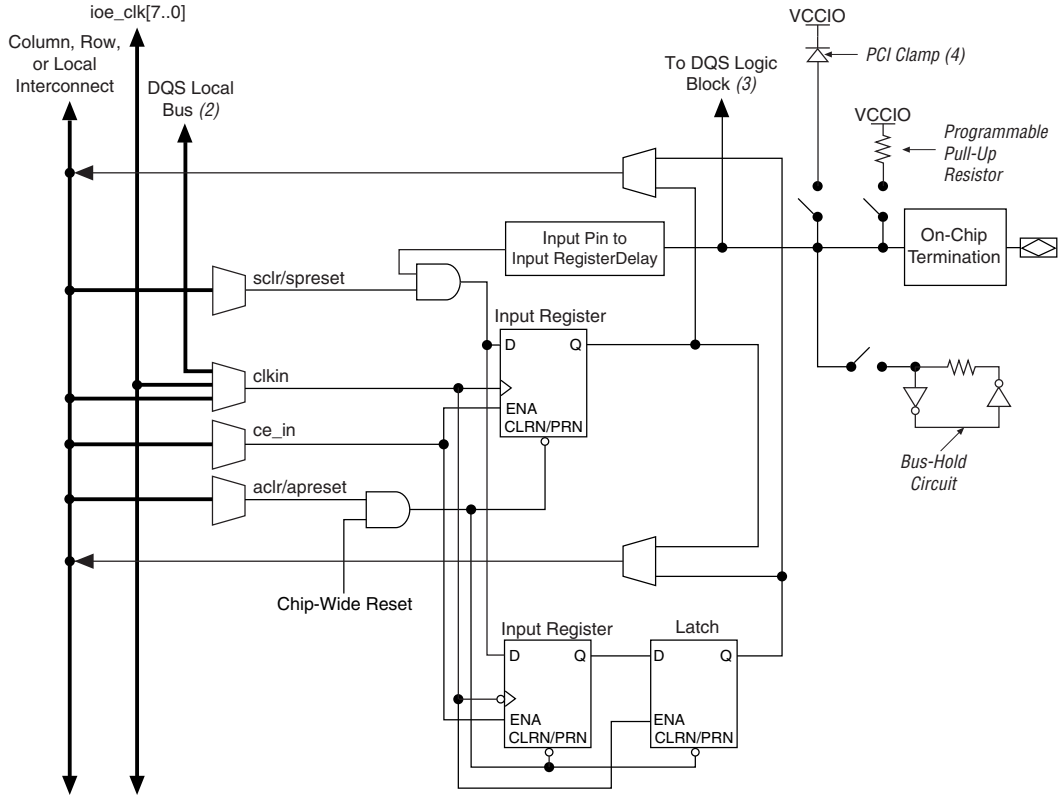
Programmable Delays	Quartus II Logic Option
Input pin to logic array delay	Input delay from pin to internal cells
Input pin to input register delay	Input delay from pin to input register
Output pin delay	Delay from output register to output pin
Output enable register t_{CO} delay	Delay to output enable pin

The IOE registers in Stratix II GX devices share the same source for clear or preset. You can program preset or clear for each individual IOE. You can also program the registers to power up high or low after configuration is complete. If programmed to power up low, an asynchronous clear can control the registers. If programmed to power up high, an asynchronous preset can control the registers. This feature prevents the inadvertent activation of another device's active-low input upon power-up. If one register in an IOE uses a preset or clear signal, all registers in the IOE must use that same signal if they require preset or clear. Additionally, a synchronous reset signal is available for the IOE registers.

Double Data Rate I/O Pins

Stratix II GX devices have six registers in the IOE, which support DDR interfacing by clocking data on both positive and negative clock edges. The IOEs in Stratix II GX devices support DDR inputs, DDR outputs, and bidirectional DDR modes. When using the IOE for DDR inputs, the two input registers clock double rate input data on alternating edges. An input latch is also used in the IOE for DDR input acquisition. The latch holds the data that is present during the clock high times, allowing both bits of data to be synchronous with the same clock edge (either rising or falling). Figure 2–82 shows an IOE configured for DDR input. Figure 2–83 shows the DDR input timing diagram.

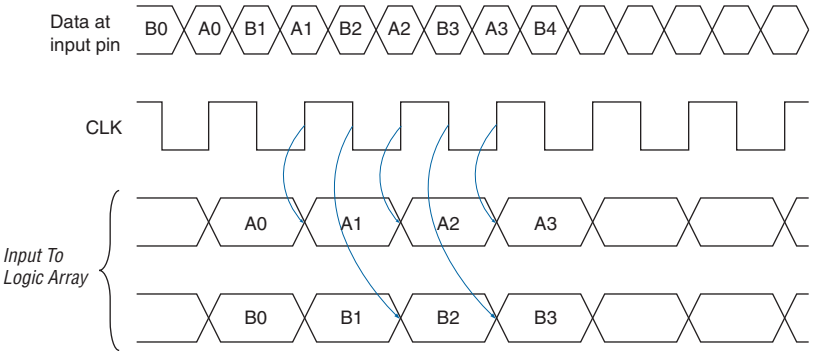
Figure 2–82. Stratix II GX IOE in DDR Input I/O Configuration *Note (1)*



Notes to Figure 2–82:

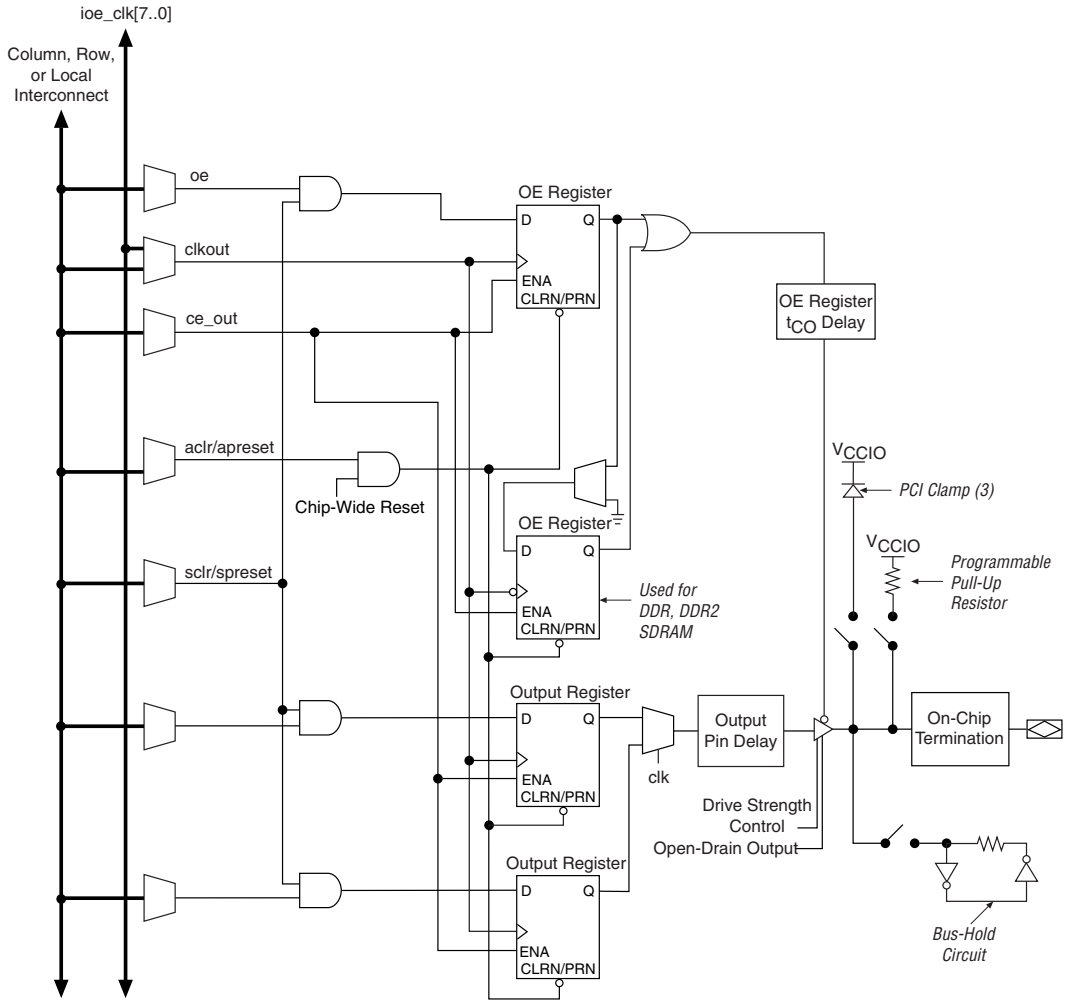
(1) All input signals to the IOE can be inverted at the IOE.
(2) This signal connection is only allowed on dedicated DQ function pins.
(3) This signal is for dedicated DQS function pins only.
(4) The optional PCI clamp is only available on column I/O pins.

Figure 2–83. Input Timing Diagram in DDR Mode



When using the IOE for DDR outputs, the two output registers are configured to clock two data paths from ALMs on rising clock edges. These output registers are multiplexed by the clock to drive the output pin at a ×2 rate. One output register clocks the first bit out on the clock high time, while the other output register clocks the second bit out on the clock low time. Figure 2–84 shows the IOE configured for DDR output. Figure 2–85 shows the DDR output timing diagram.

Figure 2–84. Stratix II GX IOE in DDR Output I/O Configuration *Notes (1), (2)*



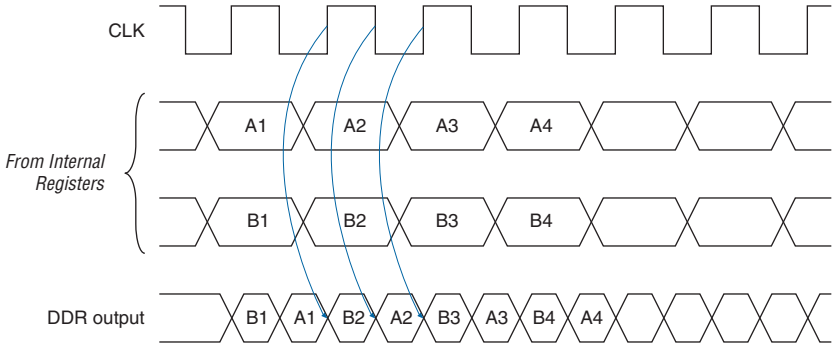
Notes to Figure 2–84:

(1) All input signals to the IOE can be inverted at the IOE.

(2) The tri-state buffer is active low. The DDIO megafunction represents the tri-state buffer as active-high with an inverter at the OE register data port.

(3) The optional PCI clamp is only available on column I/O pins.

Figure 2–85. Output Timing Diagram in DDR Mode



The Stratix II GX IOE operates in bidirectional DDR mode by combining the DDR input and DDR output configurations. The negative-edge-clocked OE register holds the OE signal inactive until the falling edge of the clock to meet DDR SDRAM timing requirements.

External RAM Interfacing

In addition to the six I/O registers in each IOE, Stratix II GX devices also have dedicated phase-shift circuitry for interfacing with external memory interfaces, including DDR and DDR2 SDRAM, QDR II SRAM, RLDRAM II, and SDR SDRAM. In every Stratix II GX device, the I/O banks at the top (banks 3 and 4) and bottom (banks 7 and 8) of the device support DQ and DQS signals with DQ bus modes of ×4, ×8/×9, ×16/×18, or ×32/×36. Table 2–31 shows the number of DQ and DQS buses that are supported per device.

Table 2–31. DQS and DQ Bus Mode Support

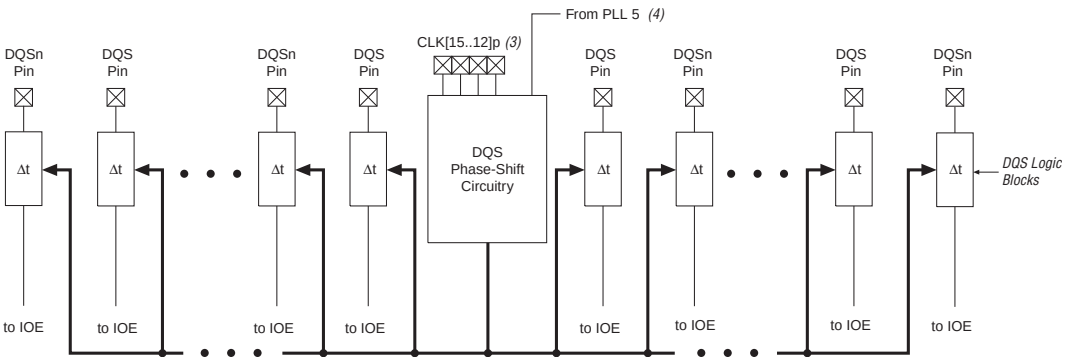
Device	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2SGX30	780-pin FineLine BGA	18	8	4	0
EP2SGX60	780-pin FineLine BGA	18	8	4	0
	1,152-pin FineLine BGA	36	18	8	4
EP2SGX90	1,152-pin FineLine BGA	36	18	8	4
	1,508-pin FineLine BGA	36	18	8	4
EP2SGX130	1,508-pin FineLine BGA	36	18	8	4

A compensated delay element on each DQS pin automatically aligns input DQS synchronization signals with the data window of their corresponding DQ data signals. The DQS signals drive a local DQS bus in the top and bottom I/O banks. This DQS bus is an additional resource to the I/O clocks and is used to clock DQ input registers with the DQS signal.

The Stratix II GX device has two phase-shifting reference circuits, one on the top and one on the bottom of the device. The circuit on the top controls the compensated delay elements for all DQS pins on the top. The circuit on the bottom controls the compensated delay elements for all DQS pins on the bottom.

Each phase-shifting reference circuit is driven by a system reference clock, which must have the same frequency as the DQS signal. Clock pins `CLK[15..12]p` feed the phase circuitry on the top of the device and clock pins `CLK[7..4]p` feed the phase circuitry on the bottom of the device. In addition, PLL clock outputs can also feed the phase-shifting reference circuits. Figure 2–86 shows the phase-shift reference circuit control of each DQS delay shift on the top of the device. This same circuit is duplicated on the bottom of the device.

Figure 2–86. DQS Phase-Shift Circuitry *Notes (1), (2)*



Notes to Figure 2–86:

(1) There are up to 18 pairs of DQS and DQSn pins available on the top or the bottom of the Stratix II GX device. There are up to 10 pairs on the right side and 8 pairs on the left side of the DQS phase-shift circuitry.

(2) The "t" module represents the DQS logic block.

(3) Clock pins `CLK[15..12]p` feed the phase-shift circuitry on the top of the device and clock pins `CLK[7..4]p` feed the phase circuitry on the bottom of the device. You can also use a PLL clock output as a reference clock to the phaseshift circuitry.

(4) You can only use PLL 5 to feed the DQS phase-shift circuitry on the top of the device and PLL 6 to feed the DQS phase-shift circuitry on the bottom of the device.

These dedicated circuits combined, with enhanced PLL clocking and phase-shift ability, provide a complete hardware solution for interfacing to high-speed memory.

For more information on external memory interfaces, refer to the *External Memory Interfaces in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Programmable Drive Strength

The output buffer for each Stratix II GX device I/O pin has a programmable drive strength control for certain I/O standards. The LVTTL, LVCMOS, SSTL, and HSTL standards have several levels of drive strength that you can control. The default setting used in the Quartus II software is the maximum current strength setting that is used to achieve maximum I/O performance. For all I/O standards, the minimum setting is the lowest drive strength that guarantees the I_{OH}/I_{OL} of the standard. Using minimum settings provides signal slew rate control to reduce system noise and signal overshoot.

Table 2–32 shows the possible settings for the I/O standards with drive strength control.

Table 2–32. Programmable Drive Strength *Note (1)*

I/O Standard	I_{OH} / I_{OL} Current Strength Setting (mA) for Column I/O Pins	I_{OH} / I_{OL} Current Strength Setting (mA) for Row I/O Pins
3.3-V LVTTL	24, 20, 16, 12, 8, 4	12, 8, 4
3.3-V LVCMOS	24, 20, 16, 12, 8, 4	8, 4
2.5-V LVTTL/LVCMOS	16, 12, 8, 4	12, 8, 4
1.8-V LVTTL/LVCMOS	12, 10, 8, 6, 4, 2	8, 6, 4, 2
1.5-V LVCMOS	8, 6, 4, 2	4, 2
SSTL-2 Class I	12, 8	12, 8
SSTL-2 Class II	24, 20, 16	16
SSTL-18 Class I	12, 10, 8, 6, 4	10, 8, 6, 4
SSTL-18 Class II	20, 18, 16, 8	—
HSTL-18 Class I	12, 10, 8, 6, 4	12, 10, 8, 6, 4
HSTL-18 Class II	20, 18, 16	—
HSTL-15 Class I	12, 10, 8, 6, 4	8, 6, 4
HSTL-15 Class II	20, 18, 16	—

Note to Table 2–32:

(1) The Quartus II software default current setting is the maximum setting for each I/O standard.

Open-Drain Output

Stratix II GX devices provide an optional open-drain (equivalent to an open collector) output for each I/O pin. This open-drain output enables the device to provide system-level control signals (for example, interrupt and write enable signals) that can be asserted by any of several devices.

Bus Hold

Each Stratix II GX device I/O pin provides an optional bus-hold feature. The bus-hold circuitry can hold the signal on an I/O pin at its last-driven state. Since the bus-hold feature holds the last-driven state of the pin until the next input signal is present, an external pull-up or pull-down resistor is not needed to hold a signal level when the bus is tri-stated.

The bus-hold circuitry also pulls undriven pins away from the input threshold voltage where noise can cause unintended high-frequency switching. You can select this feature individually for each I/O pin. The bus-hold output drives no higher than V_{CCIO} to prevent overdriving signals. If the bus-hold feature is enabled, the programmable pull-up option cannot be used. Disable the bus-hold feature when the I/O pin has been configured for differential signals.

The bus-hold circuitry uses a resistor with a nominal resistance (RBH) of approximately 7 kΩ to pull the signal level to the last-driven state.

Refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook* for the specific sustaining current driven through this resistor and overdrive current used to identify the next-driven input level. This information is provided for each V_{CCIO} voltage level.

The bus-hold circuitry is active only after configuration. When going into user mode, the bus-hold circuit captures the value on the pin present at the end of configuration.

Programmable Pull-Up Resistor

Each Stratix II GX device I/O pin provides an optional programmable pull-up resistor during user mode. If you enable this feature for an I/O pin, the pull-up resistor (typically 25 kΩ) holds the output to the V_{CCIO} level of the output pin's bank.

Programmable pull-up resistors are only supported on user I/O pins and are not supported on dedicated configuration pins, JTAG pins, or dedicated clock pins.

Advanced I/O Standard Support

The Stratix II GX device IOEs support the following I/O standards:

- 3.3-V LVTTL/LVCMOS
- 2.5-V LVTTL/LVCMOS
- 1.8-V LVTTL/LVCMOS
- 1.5-V LVCMOS
- 3.3-V PCI
- 3.3-V PCI-X mode 1
- LVDS
- LVPECL (on input and output clocks only)
- Differential 1.5-V HSTL class I and II
- Differential 1.8-V HSTL class I and II
- Differential SSTL-18 class I and II

- Differential SSTL-2 class I and II
- 1.2-V HSTL class I and II
- 1.5-V HSTL class I and II
- 1.8-V HSTL class I and II
- SSTL-2 class I and II
- SSTL-18 class I and II

Table 2–33 describes the I/O standards supported by Stratix II GX devices.

Table 2–33. Stratix II GX Supported I/O Standards

I/O Standard	Type	Input Reference Voltage (V_{REF}) (V)	Output Supply Voltage (V_{CCIO}) (V)	Board Termination Voltage (V_{TT}) (V)
LVTTL	Single-ended	—	3.3	—
LVCMOS	Single-ended	—	3.3	—
2.5 V	Single-ended	—	2.5	—
1.8 V	Single-ended	—	1.8	—
1.5-V LVCMOS	Single-ended	—	1.5	—
3.3-V PCI	Single-ended	—	3.3	—
3.3-V PCI-X mode 1	Single-ended	—	3.3	—
LVDS	Differential	—	2.5 *(3)*	—
LVPECL *(1)*	Differential	—	3.3	—
HyperTransport technology	Differential	—	2.5 *(3)*	—
Differential 1.5-V HSTL class I and II *(2)*	Differential	0.75	1.5	0.75
Differential 1.8-V HSTL class I and II *(2)*	Differential	0.90	1.8	0.90
Differential SSTL-18 class I and II *(2)*	Differential	0.90	1.8	0.90
Differential SSTL-2 class I and II *(2)*	Differential	1.25	2.5	1.25
1.2-V HSTL*(4)*	Voltage-referenced	0.6	1.2	0.6
1.5-V HSTL class I and II	Voltage-referenced	0.75	1.5	0.75
1.8-V HSTL class I and II	Voltage-referenced	0.9	1.8	0.9
SSTL-18 class I and II	Voltage-referenced	0.90	1.8	0.90

Table 2–33. Stratix II GX Supported I/O Standards

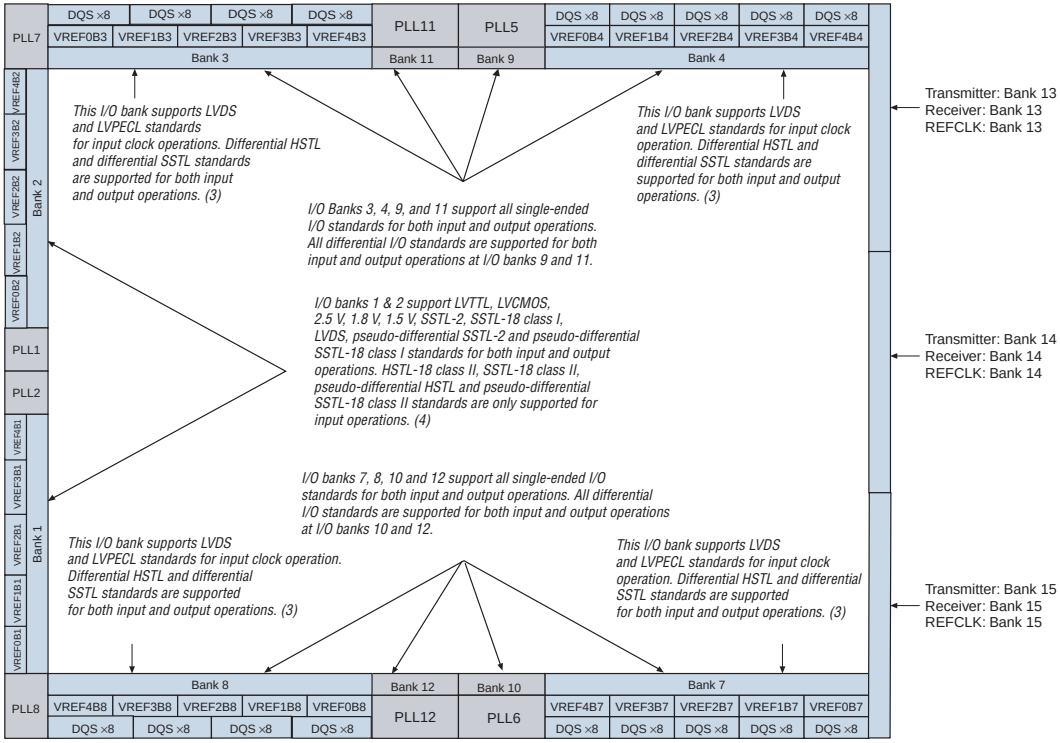
I/O Standard	Type	Input Reference Voltage (V_{REF}) (V)	Output Supply Voltage (V_{CCIO}) (V)	Board Termination Voltage (V_{TT}) (V)
SSTL-2 class I and II	Voltage-referenced	1.25	2.5	1.25

Notes to Table 2–33:
(1) This I/O standard is only available on input and output column clock pins.
(2) This I/O standard is only available on input clock pins and DQS pins in I/O banks 3, 4, 7, and 8, and output clock pins in I/O banks 9,10, 11, and 12.
(3) V_{CCIO} is 3.3 V when using this I/O standard in input and output column clock pins (in I/O banks 3, 4, 7, 8, 9, 10, 11, and 12).
(4) 1.2-V HSTL is only supported in I/O banks 4, 7, and 8.

For more information on I/O standards supported by Stratix II GX I/O banks, refer to the *Selectable I/O Standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Stratix II GX devices contain six I/O banks and four enhanced PLL external clock output banks, as shown in Figure 2–87. The two I/O banks on the left of the device contain circuitry to support source-synchronous, high-speed differential I/O for LVDS inputs and outputs. These banks support all Stratix II GX I/O standards except PCI or PCI-X I/O pins, and SSTL-18 class II and HSTL outputs. The top and bottom I/O banks support all single-ended I/O standards. Additionally, enhanced PLL external clock output banks allow clock output capabilities such as differential support for SSTL and HSTL.

Figure 2–87. Stratix II GX I/O Banks *Notes (1), (2)*



Notes to Figure 2–87:

(1) Figure 2–87 is a top view of the silicon die that corresponds to a reverse view for flip-chip packages. It is a graphical representation only.

(2) Depending on the size of the device, different device members have different numbers of V_{REF} groups. Refer to the pin list and the Quartus II software for exact locations.

(3) Banks 9 through 12 are enhanced PLL external clock output banks.

(4) Horizontal I/O banks feature SERDES and DPA circuitry for high-speed differential I/O standards. See the *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook 2* for more information on differential I/O standards.

Each I/O bank has its own VCCIO pins. A single device can support 1.5-, 1.8-, 2.5-, and 3.3-V interfaces; each bank can support a different V_{CCIO} level independently. Each bank also has dedicated VREF pins to support the voltage-referenced standards (such as SSTL-2).

Each I/O bank can support multiple standards with the same V_{CCIO} for input and output pins. Each bank can support one V_{REF} voltage level. For example, when V_{CCIO} is 3.3 V, a bank can support LVTTL, LVCMOS, and 3.3-V PCI for inputs and outputs.

On-Chip Termination

Stratix II GX devices provide differential (for the LVDS technology I/O standard) and series on-chip termination to reduce reflections and maintain signal integrity. On-chip termination simplifies board design by minimizing the number of external termination resistors required. Termination can be placed inside the package, eliminating small stubs that can still lead to reflections.

Stratix II GX devices provide four types of termination:

- Differential termination (R_D)
- Series termination (R_S) without calibration
- Series termination (R_S) with calibration
- Parallel termination (R_T) with calibration

Table 2–34 shows the Stratix II GX on-chip termination support per I/O bank.

Table 2–34. On-Chip Termination Support by I/O Banks (Part 1 of 2)

On-Chip Termination Support	I/O Standard Support	Top and Bottom Banks (3, 4, 7, 8)	Left Bank (1, 2)
Series termination without calibration	3.3-V LVTTL	✓	✓
	3.3-V LVCMOS	✓	✓
	2.5-V LVTTL	✓	✓
	2.5-V LVCMOS	✓	✓
	1.8-V LVTTL	✓	✓
	1.8-V LVCMOS	✓	✓
	1.5-V LVTTL	✓	✓
	1.5-V LVCMOS	✓	✓
	SSTL-2 class I and II	✓	✓
	SSTL-18 class I	✓	✓
	SSTL-18 class II	✓	—
	1.8-V HSTL class I	✓	✓
	1.8-V HSTL class II	✓	—
	1.5-V HSTL class I	✓	✓
	1.2-V HSTL	✓	—

Table 2–34. On-Chip Termination Support by I/O Banks (Part 2 of 2)

On-Chip Termination Support	I/O Standard Support	Top and Bottom Banks (3, 4, 7, 8)	Left Bank (1, 2)
Series termination with calibration	3.3-V LVTTL	✓	—
	3.3-V LVCMOS	✓	—
	2.5-V LVTTL	✓	—
	2.5-V LVCMOS	✓	—
	1.8-V LVTTL	✓	—
	1.8-V LVCMOS	✓	—
	1.5-V LVTTL	✓	—
	1.5-V LVCMOS	✓	—
	SSTL-2 class I and II	✓	—
	SSTL-18 class I and II	✓	—
	1.8-V HSTL class I	✓	—
	1.8-V HSTL class II	✓	—
	1.5-V HSTL class I	✓	—
	1.2-V HSTL	✓	—
Differential termination *(1)*	LVDS	—	✓
	HyperTransport technology	—	✓

Note to Table 2–34:

(1) Clock pins `CLK1` and `CLK3`, and pins `FPLL[7..8]CLK` do not support differential on-chip termination. Clock pins `CLK0` and `CLK2`, do support differential on-chip termination. Clock pins in the top and bottom banks (`CLK[4..7, 12..15]`) do not support differential on-chip termination.

Differential On-Chip Termination

Stratix II GX devices support internal differential termination with a nominal resistance value of 100 for LVDS input receiver buffers. LVPECL input signals (supported on clock pins only) require an external termination resistor. Differential on-chip termination is supported across the full range of supported differential data rates, as shown in the *High-Speed I/O Specifications* section of the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

For more information on differential on-chip termination, refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

For more information on tolerance specifications for differential on-chip termination, refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

On-Chip Series Termination without Calibration

Stratix II GX devices support driver impedance matching to provide the I/O driver with controlled output impedance that closely matches the impedance of the transmission line. As a result, reflections can be significantly reduced. Stratix II GX devices support on-chip series termination for single-ended I/O standards with typical R_S values of 25 and 50 Ω. Once matching impedance is selected, current drive strength is no longer selectable. Table 2–34 shows the list of output standards that support on-chip series termination without calibration.

For more information about series on-chip termination supported by Stratix II GX devices, refer to the *Selectable I/O Standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

For more information about tolerance specifications for on-chip termination without calibration, refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

On-Chip Series Termination with Calibration

Stratix II GX devices support on-chip series termination with calibration in column I/O pins in top and bottom banks. There is one calibration circuit for the top I/O banks and one circuit for the bottom I/O banks. Each on-chip series termination calibration circuit compares the total impedance of each I/O buffer to the external 25-Ω or 50-Ω resistors connected to the `RUP` and `RDN` pins, and dynamically enables or disables the transistors until they match. Calibration occurs at the end of device configuration. Once the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers.

For more information about series on-chip termination supported by Stratix II GX devices, refer to the *Selectable I/O Standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

For more information about tolerance specifications for on-chip termination with calibration, refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

On-Chip Parallel Termination with Calibration

Stratix II GX devices support on-chip parallel termination with calibration for column I/O pins only. There is one calibration circuit for the top I/O banks and one circuit for the bottom I/O banks. Each on-chip parallel termination calibration circuit compares the total impedance of each I/O buffer to the external 50-Ω resistors connected to the RUP and RDN pins and dynamically enables or disables the transistors until they match. Calibration occurs at the end of device configuration. Once the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers.

On-chip parallel termination with calibration is only supported for input pins.

For more information about on-chip termination supported by Stratix II devices, refer to the *Selectable I/O Standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

For more information about tolerance specifications for on-chip termination with calibration, refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook.*

MultiVolt I/O Interface

The Stratix II GX architecture supports the MultiVolt I/O interface feature that allows Stratix II GX devices in all packages to interface with systems of different supply voltages. The Stratix II GX VCCINT pins must always be connected to a 1.2-V power supply. With a 1.2-V V_{CCINT} level, input pins are 1.2-, 1.5-, 1.8-, 2.5-, and 3.3-V tolerant. The VCCIO pins can be connected to either a 1.2-, 1.5-, 1.8-, 2.5-, or 3.3-V power supply, depending on the output requirements. The output levels are compatible with systems of the same voltage as the power supply (for example, when VCCIO pins are connected to a 1.5-V power supply, the output levels are compatible with 1.5-V systems). The Stratix II GX VCCPD power pins must be connected to a 3.3-V power supply. These power pins are used to supply the pre-driver power to the output buffers, which increases the performance of the output pins. The VCCPD pins also power configuration input pins and JTAG input pins.

Table 2–35 summarizes Stratix II GX MultiVolt I/O support.

Table 2–35. Stratix II GX MultiVolt I/O Support *Note (1)*

V_{CCIO} (V)	Input Signal (V)					Output Signal (V)					
	1.2	1.5	1.8	2.5	3.3	1.2	1.5	1.8	2.5	3.3	5.0
1.2	(4)	✓ (2)	✓ (2)	✓ (2)	✓ (2)	✓ (4)	—	—	—	—	—
1.5	(4)	✓	✓	✓ (2)	✓ (2)	✓ (3)	✓	—	—	—	—
1.8	(4)	✓	✓	✓ (2)	✓ (2)	✓ (3)	✓ (3)	✓	—	—	—
2.5	(4)	—	—	✓	✓	✓ (3)	✓ (3)	✓ (3)	✓	—	—
3.3	(4)	—	—	✓	✓	✓ (3)	✓ (3)	✓ (3)	✓ (3)	✓	✓

Notes to Table 2–35:

(1) To drive inputs higher than V_{CCIO} but less than 4.0 V, disable the PCI clamping diode and select the **Allow LVTTL and LVCMOS input levels to overdrive input buffer** option in the Quartus II software.

(2) The pin current may be slightly higher than the default value. You must verify that the driving device's V_{OL} maximum and V_{OH} minimum voltages do not violate the applicable Stratix II GX V_{IL} maximum and V_{IH} minimum voltage specifications.

(3) Although V_{CCIO} specifies the voltage necessary for the Stratix II GX device to drive out, a receiving device powered at a different level can still interface with the Stratix II GX device if it has inputs that tolerate the V_{CCIO} value.

(4) Stratix II GX devices support 1.2-V HSTL. They do not support 1.2-V LVTTL and 1.2-V LVCMOS.

The TDO and nCEO pins are powered by V_{CCIO} of the bank that they reside. TDO is in I/O bank 4 and nCEO is in I/O bank 7. Ideally, the V_{CC} supplies for the I/O buffers of any two connected pins are at the same voltage level. This may not always be possible depending on the V_{CCIO} level of TDO and nCEO pins on master devices and the configuration voltage level chosen by V_{CCSEL} on slave devices. Master and slave devices can be in any position in the chain. Master indicates that it is driving out TDO or nCEO to a slave device. For multi-device passive configuration schemes, the nCEO pin of the master device drives the nCE pin of the slave device. The VCCSEL pin on the slave device selects which input buffer is used for nCE. When V_{CCSEL} is logic high, it selects the 1.8-V/1.5-V buffer powered by V_{CCIO}. When V_{CCSEL} is logic low, it selects the 3.3-V/2.5-V input buffer powered by V_{CCPD}. The ideal case is to have the V_{CCIO} of the nCEO bank in a master device match the V_{CCSEL} settings for the nCE input buffer of the slave device it is connected to, but that may not be possible depending on the application.

Table 2–36 contains board design recommendations to ensure that nCEO can successfully drive nCE for all power supply combinations.

Table 2–36. Board Design Recommendations for nCEO and nCE Input Buffer Power

nCE Input Buffer Power in I/O Bank 3	Stratix II GX nCEO V_{CCIO} Voltage Level in I/O Bank 7				
	V_{CCIO} = 3.3 V	V_{CCIO} = 2.5 V	V_{CCIO} = 1.8 V	V_{CCIO} = 1.5 V	V_{CCIO} = 1.2 V
VCCSEL high (V_{CCIO} Bank 3 = 1.5 V)	✓ (1), (2)	✓ (3), (4)	✓ (5)	✓	✓
VCCSEL high (V_{CCIO} Bank 3 = 1.8 V)	✓ (1), (2)	✓ (3), (4)	✓	✓	Level shifter required
VCCSEL low (nCE powered by V_{CCPD} = 3.3 V)	✓	✓ (4)	✓ (6)	Level shifter required	Level shifter required

Notes to Table 2–36:
(1) Input buffer is 3.3-V tolerant.
(2) The nCEO output buffer meets V_{OH} (MIN) = 2.4 V.
(3) Input buffer is 2.5-V tolerant.
(4) The nCEO output buffer meets V_{OH} (MIN) = 2.0 V.
(5) Input buffer is 1.8-V tolerant.
(6) An external 250-Ω pull-up resistor is not required, but recommended if signal levels on the board are not optimal.

For JTAG chains, the TDO pin of the first device drives the TDI pin of the second device in the chain. The V_{CCSEL} input on the JTAG input I/O cells (TCK, TMS, TDI, and TRST) is internally hardwired to GND selecting the 3.3-V/2.5-V input buffer powered by V_{CCPD}. The ideal case is to have the V_{CCIO} of the TDO bank from the first device match the V_{CCSEL} settings for TDI on the second device, but that may not be possible depending on the application. Table 2–37 contains board design recommendations to ensure proper JTAG chain operation.

Table 2–37. Supported TDO/TDI Voltage Combinations (Part 1 of 2)

Device	TDI Input Buffer Power	Stratix II GX TDO V_{CCIO} Voltage Level in I/O Bank 4				
		V_{CCIO} = 3.3 V	V_{CCIO} = 2.5 V	V_{CCIO} = 1.8 V	V_{CCIO} = 1.5 V	V_{CCIO} = 1.2 V
Stratix II GX	Always V_{CCPD} (3.3 V)	✓ (1)	✓ (2)	✓ (3)	Level shifter required	Level shifter required

Table 2–37. Supported TDO/TDI Voltage Combinations (Part 2 of 2)

Device	TDI Input Buffer Power	Stratix II GX TDO V_{CCIO} Voltage Level in I/O Bank 4				
		V_{CCIO} = 3.3 V	V_{CCIO} = 2.5 V	V_{CCIO} = 1.8 V	V_{CCIO} = 1.5 V	V_{CCIO} = 1.2 V
Non-Stratix II GX	VCC = 3.3 V	✓ (1)	✓ (2)	✓ (3)	Level shifter required	Level shifter required
	VCC = 2.5 V	✓ (1), (4)	✓ (2)	✓ (3)	Level shifter required	Level shifter required
	VCC = 1.8 V	✓ (1), (4)	✓ (2), (5)	✓	Level shifter required	Level shifter required
	VCC = 1.5 V	✓ (1), (4)	✓ (2), (5)	✓ (6)	✓	✓

Notes to Table 2–37:

(1) The TDO output buffer meets V_{OH} (MIN) = 2.4 V.
(2) The TDO output buffer meets V_{OH} (MIN) = 2.0 V.
(3) An external 250-Ω pull-up resistor is not required, but recommended if signal levels on the board are not optimal.
(4) Input buffer must be 3.3-V tolerant.
(5) Input buffer must be 2.5-V tolerant.
(6) Input buffer must be 1.8-V tolerant.

High-Speed Differential I/O with DPA Support

Stratix II GX devices contain dedicated circuitry for supporting differential standards at speeds up to 1 Gbps. The LVDS differential I/O standards are supported in the Stratix II GX device. In addition, the LVPECL I/O standard is supported on input and output clock pins on the top and bottom I/O banks.

The high-speed differential I/O circuitry supports the following high-speed I/O interconnect standards and applications:

- SPI-4 Phase 2 (POS-PHY Level 4)
- SFI-4
- Parallel RapidIO standard

There are two dedicated high-speed PLLs in the EP2SGX30 device and four dedicated high-speed PLLs in the EP2SGX60, EP2SGX90, and EP2SGX130 devices to multiply reference clocks and drive high-speed differential SERDES channels.

Tables 2–38 through 2–41 show the number of channels that each Fast PLL can clock in each of the Stratix II GX devices. In Tables 2–38 through 2–41, the first row for each transmitter or receiver provides the number of channels driven directly by the PLL. The second row below it shows the maximum channels a Fast PLL can drive if cross bank channels are used from the adjacent center Fast PLL. For example, in the 780-pin FineLine BGA EP2SGX30 device, PLL 1 can drive a maximum of

16 transmitter channels in I/O bank 1 or a maximum of 29 transmitter channels in I/O banks 1 and 2. The Quartus II software can also merge receiver and transmitter PLLs when a receiver is driving a transmitter. In this case, one fast PLL can drive both the maximum numbers of receiver and transmitter channels.

Table 2–38. EP2SGX30 Device Differential Channels *Note (1)*

Package	Transmitter/Receiver	Total Channels	Center Fast PLLs Package	
			PLL1	PLL2
780-pin FineLine BGA	Transmitter	29	16	13
	Receiver	31	17	14

Table 2–39. EP2SGX60 Device Differential Channels *Note (1)*

Package	Transmitter/Receiver	Total Channels	Center Fast PLLs		Corner Fast PLLs	
			PLL1	PLL2	PLL7	PLL8
780-pin FineLine BGA	Transmitter	29	16	13	—	—
	Receiver	31	17	14	—	—
1,152-pin FineLine BGA	Transmitter	42	21	21	21	21
	Receiver	42	21	21	21	21

Table 2–40. EP2SGX90 Device Differential Channels *Note (1)*

Package	Transmitter/Receiver	Total Channels	Center Fast PLLs		Corner Fast PLLs	
			PLL1	PLL2	PLL7	PLL8
1,152-pin FineLine BGA	Transmitter	45	23	22	23	22
	Receiver	47	23	24	23	24
1,508-pin FineLine BGA	Transmitter	59	30	29	29	29
	Receiver	59	30	29	29	29

Table 2–41. EP2SGX130 Device Differential Channels *Note (1)*

Package	Transmitter/Receiver	Total Channels	Center Fast PLLs		Corner Fast PLLs	
			PLL1	PLL2	PLL7	PLL8
1508-pin FineLine BGA	Transmitter	71	37	41	37	41
	Receiver	73	37	41	37	41

Note to Tables 2–38 through 2–41:
(1) The total number of receiver channels includes the four non-dedicated clock channels that can be optionally used as data channels.

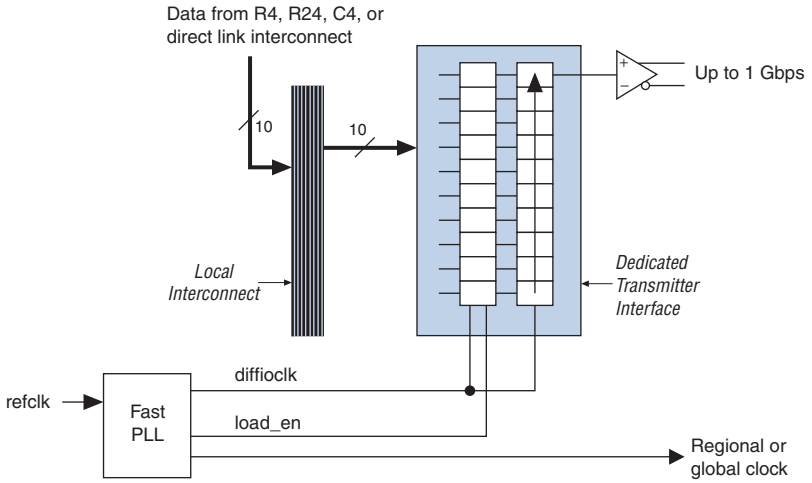
Therefore, the total number of channels is not the addition of the number of channels accessible by PLLs 1 and 2 with the number of channels accessible by PLLs 7 and 8.

Dedicated Circuitry with DPA Support

Stratix II GX devices support source-synchronous interfacing with LVDS signaling at up to 1 Gbps. Stratix II GX devices can transmit or receive serial channels along with a low-speed or high-speed clock.

The receiving device PLL multiplies the clock by an integer factor W = 1 through 32. The SERDES factor J determines the parallel data width to deserialize from receivers or to serialize for transmitters. The SERDES factor J can be set to 4, 5, 6, 7, 8, 9, or 10 and does not have to equal the PLL clock-multiplication W value. A design using the dynamic phase aligner also supports all of these J factor values. For a J factor of 1, the Stratix II GX device bypasses the SERDES block. For a J factor of 2, the Stratix II GX device bypasses the SERDES block, and the DDR input and output registers are used in the IOE. Figure 2–88 shows the block diagram of the Stratix II GX transmitter channel.

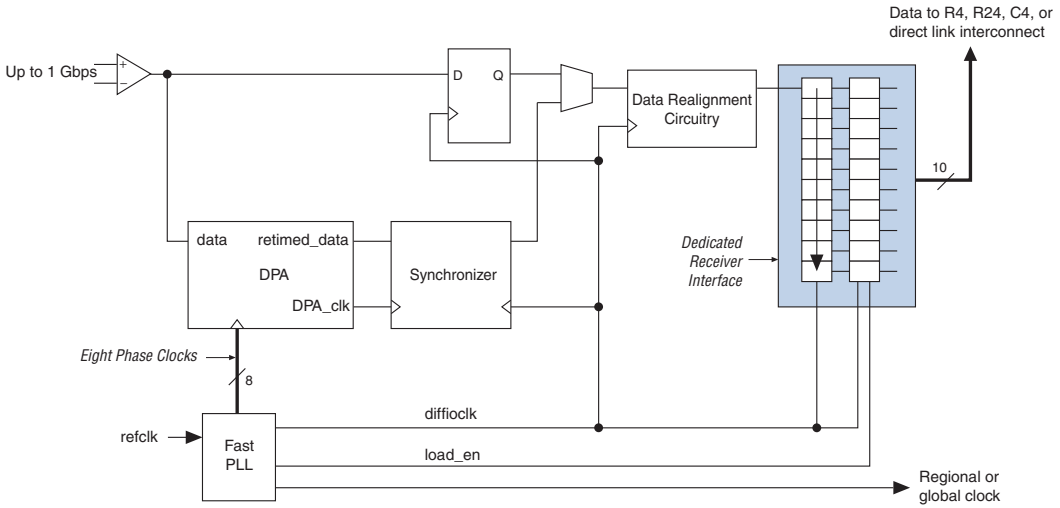
Figure 2–88. Stratix II GX Transmitter Channel



Each Stratix II GX receiver channel features a DPA block for phase detection and selection, a SERDES, a synchronizer, and a data realigner circuit. You can bypass the dynamic phase aligner without affecting the basic source-synchronous operation of the channel. In addition, you can dynamically switch between using the DPA block or bypassing the block via a control signal from the logic array.

Figure 2–89 shows the block diagram of the Stratix II GX receiver channel.

Figure 2–89. Stratix II GX Receiver Channel



An external pin or global or regional clock can drive the fast PLLs, which can output up to three clocks: two multiplied high-speed clocks to drive the SERDES block and/or external pin, and a low-speed clock to drive the logic array. In addition, eight phase-shifted clocks from the VCO can feed to the DPA circuitry.

For more information on the fast PLL, see the *PLLs in Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*.

The eight phase-shifted clocks from the fast PLL feed to the DPA block. The DPA block selects the closest phase to the center of the serial data eye to sample the incoming data. This allows the source-synchronous circuitry to capture incoming data correctly regardless of the channel-to-channel or clock-to-channel skew. The DPA block locks to a phase closest to the serial data phase. The phase-aligned DPA clock is used to write the data into the synchronizer.

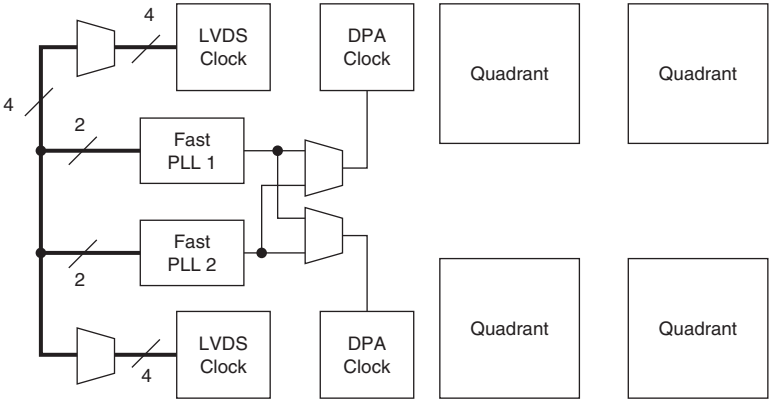
The synchronizer sits between the DPA block and the data realignment and SERDES circuitry. Since every channel utilizing the DPA block can have a different phase selected to sample the data, the synchronizer is needed to synchronize the data to the high-speed clock domain of the data realignment and the SERDES circuitry.

For high-speed source synchronous interfaces such as POS-PHY 4 and the Parallel RapidIO standard, the source synchronous clock rate is not a byte- or SERDES-rate multiple of the data rate. Byte alignment is necessary for these protocols because the source synchronous clock does not provide a byte or word boundary since the clock is one half the data rate, not one eighth. The Stratix II GX device's high-speed differential I/O circuitry provides dedicated data realignment circuitry for user-controlled byte boundary shifting. This simplifies designs while saving ALM resources. You can use an ALM-based state machine to signal the shift of receiver byte boundaries until a specified pattern is detected to indicate byte alignment.

Fast PLL and Channel Layout

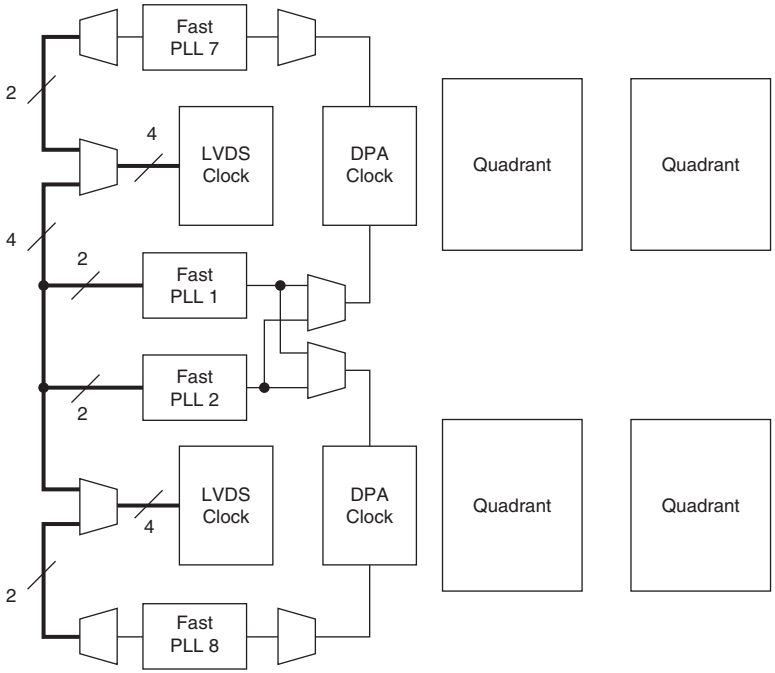
The receiver and transmitter channels are interleaved such that each I/O bank on the left side of the device has one receiver channel and one transmitter channel per LAB row. Figure 2–90 shows the fast PLL and channel layout in the EP2SGX30C/D and EP2SGX60C/D devices. Figure 2–91 shows the fast PLL and channel layout in EP2SGX60E, EP2SGX90E/F, and EP2SGX130G devices.

Figure 2–90. Fast PLL and Channel Layout in the EP2SGX30C/D and EP2SGX60C/D Devices *Note (1)*



Note to Figure 2–90:
(1) See Table 2–38 for the number of channels each device supports.

Figure 2–91. Fast PLL and Channel Layout in the EP2SGX60E to EP2SGX130 Devices *Note (1)*



Note to Figure 2–91:
(1) See Tables 2–39 through Tables 2–41 for the number of channels each device supports.

Referenced Documents

This chapter references the following documents:

- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Handbook*
- *DSP Blocks in Stratix II GX Devices* chapter in Volume 2 of the *Stratix II GX Device Handbook*
- *External Memory Interfaces in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*
- *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Selectable I/O Standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*
- *Stratix II GX Device Handbook*, volume 2
- *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Handbook*

- *Stratix II Performance and Logic Efficiency Analysis White Paper*
- *TriMatrix Embedded Memory Blocks in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*

Document Revision History

Table 2–42 shows the revision history for this chapter.

Table 2–42. Document Revision History (Part 1 of 6)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v2.2	Updated: • "Programmable Pull-Up Resistor" • "Reverse Serial Pre-CDR Loopback" • "Receiver Input Buffer" • "Pattern Detection" • "Control and Status Signals" • "Individual Power Down and Reset for the Transmitter and Receiver"	
	Updated: • Figure 2–14 • Figure 2–26 • Figure 2–27 • Figure 2–86 (notes only) • Figure 2–87	
	Updated: • Table 2–4 • Table 2–7	
	Removed note from Table 2–31.	
	Removed Tables 2-2, 2-7, and 2-8.	
	Minor text edits.	
August 2007, v2.1	Added "Reverse Serial Pre-CDR Loopback" section.	
	Updated Table 2–2.	
	Added "Referenced Documents" section.	

Table 2–42. Document Revision History (Part 2 of 6)

Date and Document Version	Changes Made	Summary of Changes
February 2007 v2.0	Added Chapter 02 “Stratix II GX Transceivers” to the beginning of Chapter 03 “Stratix II GX Architecture”. • Changed chapter number to Chapter 02.	Combined Chapter 02 “Stratix II GX Transceivers” and Chapter 03 “Stratix II GX Architecture” in the new Chapter 02 “Stratix II GX Architecture”
	Added the “Document Revision History” section to this chapter.	
	Moved the “Stratix II GX Transceiver Clocking” section to after the “Receiver Path” section.	

Table 2–42. Document Revision History (Part 3 of 6)

Date and Document Version	Changes Made	Summary of Changes
	Moved the "Transmit State Machine" section to after the "8B/10B Encoder" section.	
	Moved the "PCI Express Receiver Detect" and "PCI Express Electric Idles (or Individual Transmitter Tri-State)" sections to after the "Transmit Buffer" section.	
	Moved the "Dynamic Reconfiguration" section to the "Other Transceiver Features" section.	
	Moved the "Calibration Block", "Receiver PLL & CRU", and "Deserializer (Serial-to-Parallel Converter)" sections to the "Receiver Path" section.	
	Moved the "8B/10B Decoder" and "Receiver State Machine" sections to after the "Rate Matcher" section.	
	Moved the "Byte Ordering Block" section to after the "Byte Deserializer" section.	
	Updated the Clocking diagrams.	
	Added the "Clock Resource for PLD-Transceiver Interface" section.	
	Added the "On-Chip Parallel Termination with Calibration" section to the "On-Chip Termination" section.	
	Updated: • Table 2–2. • Table 2–10 • Table 2–14. • Table 2–3. • Table 2–5. • Table 2–8. • Table 2–13 • Table 2–18 • Table 2–19 • Table 2–29.	
	Updated Figures 2–3, 2–9, 2–24, 2–25, 2–28, 2–29, 2–60, 2–62.	
	Change 622 Mbps to 600 Mbps throughout the chapter.	

Table 2–42. Document Revision History (Part 4 of 6)

Date and Document Version	Changes Made	Summary of Changes
	Updated: ● "Transmitter PLLs" ● "Transmitter Phase Compensation FIFO Buffer" ● "8B/10B Encoder" ● "Byte Serializer" ● "Programmable Output Driver" ● "Receiver PLL & CRU" ● "Programmable Pre-Emphasis" ● "Receiver Input Buffer" ● "Control and Status Signals" ● "Programmable Run Length Violation" ● "Channel Aligner" ● "Basic Mode" ● "Byte Ordering Block" ● "Receiver Phase Compensation FIFO Buffer" ● "Loopback Modes" ● "Serial Loopback" ● "Parallel Loopback" ● "Regional Clock Network" ● "MultiVolt I/O Interface" ● "High-Speed Differential I/O with DPA Support"	
	Updated bulleted lists at the beginning of the "Transceivers" section.	
	Added reference to the "Transmit Buffer" section.	
	Deleted the Programmable V_{OD} table from the "Programmable Output Driver" section.	
	Changed "PLD Interface" heading to "Parallel Data Width" heading in Table 2–14.	
	Deleted "Global & Regional Clock Connections from Right Side Clock Pins & Fast PLL Outputs" table.	
	Updated notes to Tables 2–29 and 2–37.	
	Updated notes to Figures 2–72, 2–73 and 2–74.	
	Updated bulleted list in the "Advanced I/O Standard Support" section.	

Table 2–42. Document Revision History (Part 5 of 6)

Date and Document Version	Changes Made	Summary of Changes
Previous Chapter 02 changes: June 2006, v1.2	• Updated notes 1 and 2 in Figure 2–1. • Updated "Byte Serializer" section. • Updated Tables 2–4, 2–7, and 2–16. • Updated "Programmable Output Driver" section. • Updated Figure 2–12. • Updated "Programmable Pre-Emphasis" section. • Added Table 2–11. • Added "Dynamic Reconfiguration" section. • Added "Calibration Block" section. • Updated "Programmable Equalizer" section, including addition of Figure 2–18.	Updated input frequency range in Table 2–4.
Previous Chapter 02 changes: April 2006, v1.1	• Updated Figure 2–3. • Updated Figure 2–7. • Updated Table 2–4. • Updated "Transmit Buffer" section.	Updated input frequency range in Table 2–4.
Previous Chapter 02 changes: October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	
Previous Chapter 03 changes: August 2006, v1.4	• Updated Table 3–18 with note.	
Previous Chapter 03 changes: June 2006, v1.3	• Updated note 2 in Figure 3–41. • Updated column title in Table 3–21.	
Previous Chapter 03 changes: April 2006, v1.2	• Updated note 1 in Table 3–9. • Updated note 1 in Figure 3–40. • Updated note 2 in Figure 3–41. • Updated Table 3–16. • Updated Figure 3–56. • Updated Tables 3–19 through 3–22. • Updated Tables 3–25 and 3–26. • Updated "Fast PLL & Channel Layout" section.	Added 1,152-pin FineLine BGA package information for EP2SGX60 device in Table 3–16.

Table 2–42. Document Revision History (Part 6 of 6)

Date and Document Version	Changes Made	Summary of Changes
Previous Chapter 03 changes: December 2005 v1.1	Updated Figure 3–56.	
Previous Chapter 03 changes: October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	

3. Configuration & Testing

SIIGX51005-1.4

IEEE Std. 1149.1 JTAG Boundary-Scan Support

All Stratix® II GX devices provide Joint Test Action Group (JTAG) boundary-scan test (BST) circuitry that complies with the IEEE Std. 1149.1. You can perform JTAG boundary-scan testing either before or after, but not during configuration. Stratix II GX devices can also use the JTAG port for configuration with the Quartus® II software or hardware using either Jam Files (**.jam**) or Jam Byte-Code Files (**.jbc**).

Stratix II GX devices support IOE I/O standard setting reconfiguration through the JTAG BST chain. The JTAG chain can update the I/O standard for all input and output pins any time before or during user mode through the `CONFIG_IO` instruction. You can use this capability for JTAG testing before configuration when some of the Stratix II GX pins drive or receive from other devices on the board using voltage-referenced standards. Since the Stratix II GX device may not be configured before JTAG testing, the I/O pins may not be configured for appropriate electrical standards for chip-to-chip communication. Programming these I/O standards via JTAG allows you to fully test I/O connections to other devices.

A device operating in JTAG mode uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS`, and `TRST` pins have weak internal pull-up resistors. The JTAG input pins are powered by the 3.3-V `VCCPD` pins. The `TDO` output pin is powered by the `VCCIO` power supply in I/O bank 4.

Stratix II GX devices also use the JTAG port to monitor the logic operation of the device with the SignalTap® II embedded logic analyzer. Stratix II GX devices support the JTAG instructions shown in Table 3–1.

Stratix II GX devices must be within the first eight devices in a JTAG chain. All of these devices have the same JTAG controller. If any of the Stratix II GX devices appear after the eighth device in the JTAG chain, they will fail configuration. This does not affect SignalTap II embedded logic analysis.

Table 3–1. Stratix II GX JTAG Instructions

JTAG Instruction	Instruction Code	Description
SAMPLE/PRELOAD	00 0000 0101	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation and permits an initial data pattern to be output at the device pins. Also used by the SignalTap II embedded logic analyzer.
EXTEST *(1)*	00 0000 1111	Allows the external circuitry and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	11 1111 1111	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.
USERCODE	00 0000 0111	Selects the 32-bit USERCODE register and places it between the TDI and TDO pins, allowing the USERCODE to be serially shifted out of TDO.
IDCODE	00 0000 0110	Selects the IDCODE register and places it between TDI and TDO, allowing the IDCODE to be serially shifted out of TDO.
HIGHZ *(1)*	00 0000 1011	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation, while tri-stating all of the I/O pins.
CLAMP *(1)*	00 0000 1010	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding the I/O pins to a state defined by the data in the boundary-scan register.
ICR instructions		Used when configuring a Stratix II GX device via the JTAG port with a USB-Blaster™, MasterBlaster™, ByteBlasterMV™, or ByteBlaster II download cable, or when using a **.jam** or **.jbc** via an embedded processor or JRunner.
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is unaffected.
CONFIG_IO *(2)*	00 0000 1101	Allows configuration of I/O standards through the JTAG chain for JTAG testing. Can be executed before, during, or after configuration. Stops configuration if executed during configuration. Once issued, the CONFIG_IO instruction holds nSTATUS low to reset the configuration device. nSTATUS is held low until the IOE configuration register is loaded and the TAP controller state machine transitions to the UPDATE_DR state.
SignalTap II instructions		Monitors internal device operation with the SignalTap II embedded logic analyzer.

Notes to Table 3–1:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

(2) For more information on using the CONFIG_IO instruction, refer to the *MorphIO: An I/O Reconfiguration Solution for Altera Devices White Paper.*

The Stratix II GX device instruction register length is 10 bits and the USERCODE register length is 32 bits. Tables 3–2 and 3–3 show the boundary-scan register length and device IDCODE information for Stratix II GX devices.

Table 3–2. Stratix II GX Boundary-Scan Register Length

Device	Boundary-Scan Register Length
EP2SGX30	1,320
EP2SGX60	1,506
EP2SGX90	2,016
EP2SGX130	2,454

Table 3–3. 32-Bit Stratix II GX Device IDCODE

Device	IDCODE (32 Bits)			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit)
EP2SGX30	0000	0010 0000 1110 0001	000 0110 1110	1
EP2SGX60	0000	0010 0000 1110 0010	000 0110 1110	1
EP2SGX90	0000	0010 0000 1110 0011	000 0110 1110	1
EP2SGX130	0000	0010 0000 1110 0100	000 0110 1110	1

SignalTap II Embedded Logic Analyzer

Stratix II GX devices feature the SignalTap II embedded logic analyzer, which monitors design operation over a period of time through the IEEE Std. 1149.1 (JTAG) circuitry. You can analyze internal logic at speed without bringing internal signals to the I/O pins. This feature is particularly important for advanced packages, such as FineLine BGA packages, because it can be difficult to add a connection to a pin during the debugging process after a board is designed and manufactured.

Configuration

The logic, circuitry, and interconnects in the Stratix II GX architecture are configured with CMOS SRAM elements. Altera® FPGAs are reconfigurable and every device is tested with a high coverage production test program so you do not have to perform fault testing and can instead focus on simulation and design verification.

Stratix II GX devices are configured at system power-up with data stored in an Altera configuration device or provided by an external controller (for example, a MAX® II device or microprocessor). You can configure Stratix II GX devices using the fast passive parallel (FPP), active serial

(AS), passive serial (PS), passive parallel asynchronous (PPA), and JTAG configuration schemes. The Stratix II GX device's optimized interface allows microprocessors to configure it serially or in parallel and synchronously or asynchronously. The interface also enables microprocessors to treat Stratix II GX devices as memory and configure them by writing to a virtual memory location, making reconfiguration easy.

In addition to the number of configuration methods supported, Stratix II GX devices also offer the design security, decompression, and remote system upgrade features. The design security feature, using configuration bitstream encryption and advanced encryption standard (AES) technology, provides a mechanism to protect designs. The decompression feature allows Stratix II GX FPGAs to receive a compressed configuration bitstream and decompress this data in real-time, reducing storage requirements and configuration time. The remote system upgrade feature allows real-time system upgrades from remote locations of Stratix II GX designs. For more information, refer to the "Configuration Schemes" on page 3–6.

Operating Modes

The Stratix II GX architecture uses SRAM configuration elements that require configuration data to be loaded each time the circuit powers up. The process of physically loading the SRAM data into the device is called configuration. During initialization, which occurs immediately after configuration, the device resets registers, enables I/O pins, and begins to operate as a logic device. The I/O pins are tri-stated during power-up, and before and during configuration. Together, the configuration and initialization processes are called command mode. Normal device operation is called user mode.

SRAM configuration elements allow you to reconfigure Stratix II GX devices in-circuit by loading new configuration data into the device. With real-time reconfiguration, the device is forced into command mode with a device pin. The configuration process loads different configuration data, re-initializes the device, and resumes user-mode operation. You can perform in-field upgrades by distributing new configuration files either within the system or remotely.

The PORSEL pin is a dedicated input used to select power-on reset (POR) delay times of 12 ms or 100 ms during power up. When the PORSEL pin is connected to ground, the POR time is 100 ms. When the PORSEL pin is connected to V_{CC}, the POR time is 12 ms.

The nIO_PULLUP pin is a dedicated input that chooses whether the internal pull-up resistors on the user I/O pins and dual-purpose configuration I/O pins (nCSO, ASDO, DATA[7..0], nWS, nRS, RDYnBSY, nCS, CS, RUnLU, PGM[2..0], CLKUSR, INIT_DONE, DEV_OE, DEV_CLR) are on or off before and during configuration. A logic high (1.5, 1.8, 2.5, 3.3 V) turns off the weak internal pull-up resistors, while a logic low turns them on.

Stratix II GX devices also offer a new power supply, V_{CCPD}, which must be connected to 3.3 V in order to power the 3.3-V/2.5-V buffer available on the configuration input pins and JTAG pins. V_{CCPD} applies to all the JTAG input pins (TCK, TMS, TDI, and TRST) and the following configuration pins: nCONFIG, DCLK (when used as an input), nIO_PULLUP, DATA[7..0], RUnLU, nCE, nWS, nRS, CS, nCS, and CLKUSR. The VCCSEL pin allows the V_{CCIO} setting (of the banks where the configuration inputs reside) to be independent of the voltage required by the configuration inputs. Therefore, when selecting the V_{CCIO} voltage, you do not have to take the V_{IL} and V_{IH} levels driven to the configuration inputs into consideration. The configuration input pins, nCONFIG, DCLK (when used as an input), nIO_PULLUP, RUnLU, nCE, nWS, nRS, CS, nCS, and CLKUSR, have a dual buffer design: a 3.3-V/2.5-V input buffer and a 1.8-V/1.5-V input buffer. The V_{CCSEL} input pin selects which input buffer is used. The 3.3-V/2.5-V input buffer is powered by V_{CCPD}, while the 1.8-V/1.5-V input buffer is powered by V_{CCIO}.

V_{CCSEL} is sampled during power-up. Therefore, the V_{CCSEL} setting cannot change on-the-fly or during a reconfiguration. The V_{CCSEL} input buffer is powered by V_{CCINT} and must be hardwired to V_{CCPD} or ground. A logic high V_{CCSEL} connection selects the 1.8-V/1.5-V input buffer; a logic low selects the 3.3-V/2.5-V input buffer. V_{CCSEL} should be set to comply with the logic levels driven out of the configuration device or the MAX II microprocessor.

If the design must support configuration input voltages of 3.3 V/2.5 V, set V_{CCSEL} to a logic low. You can set the V_{CCIO} voltage of the I/O bank that contains the configuration inputs to any supported voltage. If the design must support configuration input voltages of 1.8 V/1.5 V, set V_{CCSEL} to a logic high and the V_{CCIO} of the bank that contains the configuration inputs to 1.8 V/1.5 V.

For more information on multi-volt support, including information on using TDO and nCEO in multi-volt systems, refer to the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Configuration Schemes

You can load the configuration data for a Stratix II GX device with one of five configuration schemes (refer to Table 3–4), chosen on the basis of the target application. You can use a configuration device, intelligent controller, or the JTAG port to configure a Stratix II GX device. A configuration device can automatically configure a Stratix II GX device at system power-up.

Multiple Stratix II GX devices can be configured in any of the five configuration schemes by connecting the configuration enable (nCE) and configuration enable output (nCEO) pins on each device. Stratix II GX FPGAs offer the following:

- Configuration data decompression to reduce configuration file storage
- Design security using configuration data encryption to protect designs
- Remote system upgrades for remotely updating Stratix II GX designs

Table 3–4 summarizes which configuration features can be used in each configuration scheme.

Refer to the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information about configuration schemes in Stratix II GX devices.

Table 3–4. Stratix II GX Configuration Features (Part 1 of 2)

Configuration Scheme	Configuration Method	Design Security	Decompression	Remote System Upgrade
FPP	MAX II device or microprocessor and flash device	✓ *(1)*	✓ *(1)*	✓
	Enhanced configuration device		✓ *(2)*	✓
AS	Serial configuration device	✓	✓	✓ *(3)*
PS	MAX II device or microprocessor and flash device	✓	✓	✓
	Enhanced configuration device	✓	✓	✓
	Download cable *(4)*	✓	✓	
PPA	MAX II device or microprocessor and flash device			✓

Table 3–4. Stratix II GX Configuration Features (Part 2 of 2)

Configuration Scheme	Configuration Method	Design Security	Decompression	Remote System Upgrade
JTAG	Download cable *(4)*			
	MAX II device or microprocessor and flash device			

Notes for Table 3–4:
(1) In these modes, the host system must send a DCLK that is 4× the data rate.
(2) The enhanced configuration device decompression feature is available, while the Stratix II GX decompression feature is not available.
(3) Only remote update mode is supported when using the AS configuration scheme. Local update mode is not supported.
(4) The supported download cables include the Altera USB-Blaster universal serial bus (USB) port download cable, MasterBlaster serial/USB communications cable, ByteBlaster II parallel port download cable, and the ByteBlasterMV parallel port download cable.

Device Security Using Configuration Bitstream Encryption

Stratix II and Stratix II GX FPGAs are the industry's first FPGAs with the ability to decrypt a configuration bitstream using the AES algorithm. When using the design security feature, a 128-bit security key is stored in the Stratix II GX FPGA. To successfully configure a Stratix II GX FPGA that has the design security feature enabled, the device must be configured with a configuration file that was encrypted using the same 128-bit security key. The security key can be stored in non-volatile memory inside the Stratix II GX device. This nonvolatile memory does not require any external devices, such as a battery back up, for storage.

An encrypted configuration file is the same size as a non-encrypted configuration file. When using a serial configuration scheme such as passive serial (PS) or active serial (AS), configuration time is the same whether or not the design security feature is enabled. If the fast passive parallel (FPP) scheme is used with the design security or decompression feature, a 4× DCLK is required. This results in a slower configuration time when compared to the configuration time of an FPGA that has neither the design security nor the decompression feature enabled. For more information about this feature, contact an Altera sales representative.

Device Configuration Data Decompression

Stratix II GX FPGAs support decompression of configuration data, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other

memory, and transmit this compressed bitstream to Stratix II GX FPGAs. During configuration, the Stratix II GX FPGA decompresses the bitstream in real time and programs its SRAM cells. Stratix II GX FPGAs support decompression in the FPP (when using a MAX II device or microprocessor and flash memory), AS, and PS configuration schemes. Decompression is not supported in the PPA configuration scheme nor in JTAG-based configuration.

Remote System Upgrades

Shortened design cycles, evolving standards, and system deployments in remote locations are difficult challenges faced by system designers. Stratix II GX devices can help effectively deal with these challenges with their inherent re programmability and dedicated circuitry to perform remote system updates. Remote system updates help deliver feature enhancements and bug fixes without costly recalls, reducing time to market, and extending product life.

Stratix II GX FPGAs feature dedicated remote system upgrade circuitry to facilitate remote system updates. Soft logic (Nios processor or user logic) implemented in the Stratix II GX device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information. This dedicated remote system upgrade circuitry avoids system downtime and is the critical component for successful remote system upgrades.

Remote system configuration is supported in the following Stratix II GX configuration schemes: FPP, AS, PS, and PPA. Remote system configuration can also be implemented in conjunction with Stratix II GX features such as real-time decompression of configuration data and design security using AES for secure and efficient field upgrades.

Refer to the *Remote System Upgrades with Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information about remote configuration in Stratix II GX devices.

Configuring Stratix II GX FPGAs with JRunner

The JRunner™ software driver configures Altera FPGAs, including Stratix II GX FPGAs, through the ByteBlaster II or ByteBlasterMV cables in JTAG mode. The programming input file supported is in Raw Binary File (**.rbf**) format. JRunner also requires a Chain Description File (**.cdf**)

generated by the Quartus II software. JRunner is targeted for embedded JTAG configuration. The source code is developed for the Windows NT operating system (OS), but can be customized to run on other platforms.

For more information on the JRunner software driver, refer to the *AN 414: An Embedded Solution for PLD JTAG Configuration* and the source files on the Altera web site (www.altera.com).

Programming Serial Configuration Devices with SRunner

A serial configuration device can be programmed in-system by an external microprocessor using SRunner. SRunner is a software driver developed for embedded serial configuration device programming that can be easily customized to fit into different embedded systems. SRunner reads a Raw Programming Data file (**.rpd**) and writes to serial configuration devices. The serial configuration device programming time using SRunner is comparable to the programming time when using the Quartus II software.

For more information about SRunner, refer to the *AN 418 SRunner: An Embedded Solution for Serial Configuration Device Programming* and the source code on the Altera web site.

For more information on programming serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS64, and EPCS128) Data Sheet* in the *Configuration Handbook*.

Configuring Stratix II FPGAs with the MicroBlaster Driver

The MicroBlaster software driver supports an RBF programming input file and is ideal for embedded FPP or PS configuration. The source code is developed for the Windows NT operating system, although it can be customized to run on other operating systems.

For more information on the MicroBlaster software driver, refer to the *Configuring the MicroBlaster Fast Passive Parallel Software Driver White Paper* or the *Configuring the MicroBlaster Passive Serial Software Driver White Paper* on the Altera web site.

PLL Reconfiguration

The phase-locked loops (PLLs) in the Stratix II GX device family support reconfiguration of their multiply, divide, VCO-phase selection, and bandwidth selection settings without reconfiguring the entire device. You can use either serial data from the logic array or regular I/O pins to program the PLL's counter settings in a serial chain. This option provides

considerable flexibility for frequency synthesis, allowing real-time variation of the PLL frequency and delay. The rest of the device is functional while reconfiguring the PLL.

See the *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information on Stratix II GX PLLs.

Temperature Sensing Diode (TSD)

Stratix II GX devices include a diode-connected transistor for use as a temperature sensor in power management. This diode is used with an external digital thermometer device. These devices steer bias current through the Stratix II GX diode, measuring forward voltage and converting this reading to temperature in the form of an 8-bit signed number (7 bits plus 1 sign bit). The external device's output represents the junction temperature of the Stratix II GX device and can be used for intelligent power management.

The diode requires two pins (`tempdiodep` and `tempdioden`) on the Stratix II GX device to connect to the external temperature-sensing device, as shown in Figure 3–1. The temperature sensing diode is a passive element and therefore can be used before the Stratix II GX device is powered.

Figure 3–1. External Temperature-Sensing Diode

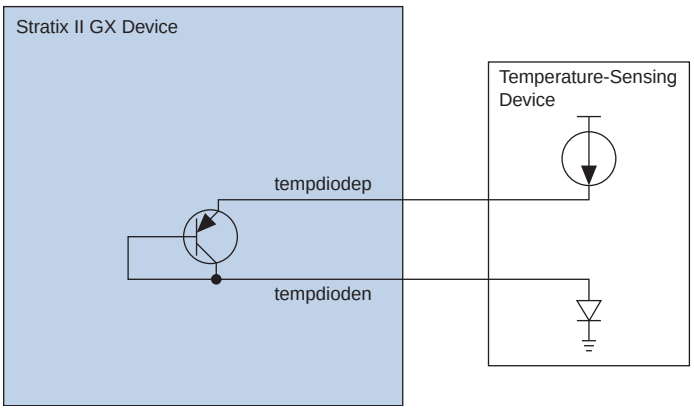


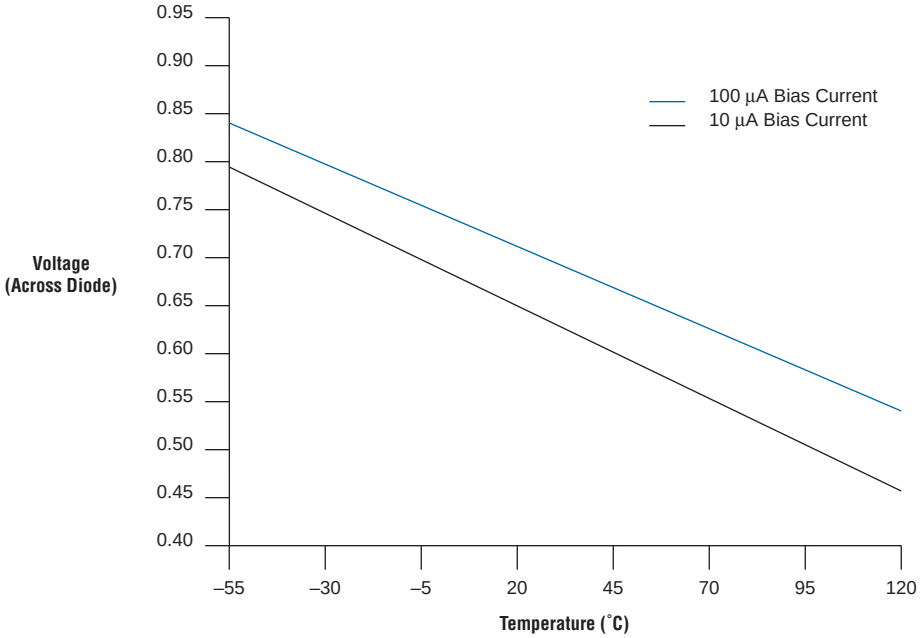
Table 3–5 shows the specifications for bias voltage and current of the Stratix II GX temperature sensing diode.

Table 3–5. Temperature-Sensing Diode Electrical Characteristics

Parameter	Minimum	Typical	Maximum	Unit
IBIAS high	80	100	120	µA
IBIAS low	8	10	12	µA
VBP - VBN	0.3		0.9	V
VBN		0.7		V
Series resistance			3	Ω

The temperature-sensing diode works for the entire operating range shown in Figure 3–2.

Figure 3–2. Temperature Versus Temperature-Sensing Diode Voltage



The temperature sensing diode is a very sensitive circuit which can be influenced by noise coupled from other traces on the board, and possibly within the device package itself, depending on device usage. The interfacing device registers temperature based on millivolts of difference as seen at the TSD. Switching I/O near the TSD pins can affect the temperature reading. Altera recommends you take temperature readings during periods of no activity in the device (for example, standby mode where no clocks are toggling in the device), such as when the nearby I/Os are at a DC state, and disable clock networks in the device.

Automated Single Event Upset (SEU) Detection

Stratix II GX devices offer on-chip circuitry for automated checking of single event upset (SEU) detection. Some applications that require the device to operate error free at high elevations or in close proximity to Earth's North or South Pole will require periodic checks to ensure continued data integrity. The error detection cyclic redundancy check (CRC) feature controlled by the **Device & Pin Options** dialog box in the Quartus II software uses a 32-bit CRC circuit to ensure data reliability and is one of the best options for mitigating SEU.

You can implement the error detection CRC feature with existing circuitry in Stratix II GX devices, eliminating the need for external logic. Stratix II GX devices compute CRC during configuration and checks the computed-CRC against an automatically computed CRC during normal operation. The `CRC_ERROR` pin reports a soft error when configuration SRAM data is corrupted, triggering device reconfiguration.

Custom-Built Circuitry

Dedicated circuitry is built into Stratix II GX devices to automatically perform error detection. This circuitry constantly checks for errors in the configuration SRAM cells while the device is in user mode. You can monitor one external pin for the error and use it to trigger a reconfiguration cycle. You can select the desired time between checks by adjusting a built-in clock divider.

Software Interface

Beginning with version 4.1 of the Quartus II software, you can turn on the automated error detection CRC feature in the **Device & Pin Options** dialog box. This dialog box allows you to enable the feature and set the internal frequency of the CRC between 400 kHz to 50 MHz. This controls the rate that the CRC circuitry verifies the internal configuration SRAM bits in the Stratix II GX FPGA.

For more information on CRC, refer to *AN 357: Error Detection Using CRC in Altera FPGA Devices*.

Referenced Documents

This chapter references the following documents:

- *AN 357: Error Detection Using CRC in Altera FPGA Devices*
- *AN 414: An Embedded Solution for PLD JTAG Configuration*
- *AN 418 SRunner: An Embedded Solution for Serial Configuration Device Programming*
- *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Configuring the MicroBlaster Fast Passive Parallel Software Driver White Paper*
- *Configuring the MicroBlaster Passive Serial Software Driver White Paper*
- *MorphIO: An I/O Reconfiguration Solution for Altera Devices White Paper*
- *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Remote System Upgrades with Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Serial Configuration Devices (EPCS1, EPCS4, EPCS64, and EPCS128) Data Sheet* in the *Configuration Handbook*
- *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Document Revision History

Table 3–6 shows the revision history for this chapter.

Table 3–6. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007 v1.4	Minor text edits.	—
August 2007 v1.3	Updated the note in the "IEEE Std. 1149.1 JTAG Boundary-Scan Support"	—
	Updated Table 3–3.	—
	Added the "Referenced Documents" section.	—
May 2007 v1.2	Updated the "Temperature Sensing Diode (TSD)" section.	—
February 2007 v1.1	Added the "Document Revision History" section to this chapter.	Added support information for the Stratix II GX device.
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	—

4. DC and Switching Characteristics

SIIGX51006-4.6

Operating Conditions

Stratix® II GX devices are offered in both commercial and industrial grades. Industrial devices are offered in -4 speed grade and commercial devices are offered in -3 (fastest), -4, and -5 speed grades.

Tables 4–1 through 4–51 provide information on absolute maximum ratings, recommended operating conditions, DC electrical characteristics, and other specifications for Stratix II GX devices.

Absolute Maximum Ratings

Table 4–1 contains the absolute maximum ratings for the Stratix II GX device family.

Table 4–1. Stratix II GX Device Absolute Maximum Ratings *Notes (1), (2), (3)*

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage	With respect to ground	–0.5	1.8	V
V_{CCIO}	Supply voltage	With respect to ground	–0.5	4.6	V
V_{CCPD}	Supply voltage	With respect to ground	–0.5	4.6	V
V_I	DC input voltage *(4)*		–0.5	4.6	V
I_{OUT}	DC output current, per pin		–25	40	mA
T_{STG}	Storage temperature	No bias	–65	150	C
T_J	Junction temperature	BGA packages under bias	–55	125	C

Notes to Table 4–1:

(1) See the *Operating Requirements for Altera Devices Data Sheet* for more information.

(2) Conditions beyond those listed in Table 4–1 may cause permanent damage to a device. Additionally, device operation at the absolute maximum ratings for extended periods of time may have adverse affects on the device.

(3) Supply voltage specifications apply to voltage readings taken at the device pins, not at the power supply.

(4) During transitions, the inputs may overshoot to the voltage shown in Table 4–2 based upon the input duty cycle. The DC case is equivalent to 100% duty cycle. During transitions, the inputs may undershoot to –2.0 V for input currents less than 100 mA and periods shorter than 20 ns.

Table 4–2. Maximum Duty Cycles in Voltage Transitions			
Symbol	**Parameter**	**Condition**	**Maximum Duty Cycles (%)** *(1)*
V_I	Maximum duty cycles in voltage transitions	V_I = 4.0 V	100
		V_I = 4.1 V	90
		V_I = 4.2 V	50
		V_I = 4.3 V	30
		V_I = 4.4 V	17
		V_I = 4.5 V	10

Note to Table 4–2:

(1) During transition, the inputs may overshoot to the voltages shown based on the input duty cycle. The duty cycle case is equivalent to 100% duty cycle.

Recommended Operating Conditions

Table 4–3 contains the Stratix II GX device family recommended operating conditions.

Table 4–3. Stratix II GX Device Recommended Operating Conditions (Part 1 of 2) Note (1)					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Unit**
V_{CCINT}	Supply voltage for internal logic and input buffers	100 µs ≤rise time ≤100 ms *(3)*	1.15	1.25	V
V_{CCIO}	Supply voltage for output buffers, 3.3-V operation	100 µs ≤rise time ≤100 ms *(3)*, *(6)*	3.135 (3.00)	3.465 (3.60)	V
	Supply voltage for output buffers, 2.5-V operation	100 µs ≤rise time ≤100 ms *(3)*	2.375	2.625	V
	Supply voltage for output buffers, 1.8-V operation	100 µs ≤rise time ≤100 ms *(3)*	1.71	1.89	V
	Supply voltage for output buffers, 1.5-V operation	100 µs ≤rise time ≤100 ms *(3)*	1.425	1.575	V
	Supply voltage for output buffers, 1.2-V operation	100 µs ≤rise time ≤100 ms *(3)*	1.15	1.25	V
V_{CCPD}	Supply voltage for pre-drivers as well as configuration and JTAG I/O buffers.	100 µs ≤rise time ≤100 ms *(4)*	3.135	3.465	V
V_I	Input voltage (see Table 4–2)	*(2)*, *(5)*	–0.5	4.0	V
V_O	Output voltage		0	V_{CCIO}	V

Table 4–3. Stratix II GX Device Recommended Operating Conditions (Part 2 of 2) Note (1)					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Unit**
T_J	Operating junction temperature	For commercial use	0	85	C
		For industrial use	–40	100	C

Notes to Table 4–3:
(1) Supply voltage specifications apply to voltage readings taken at the device pins, not at the power supply.
(2) During transitions, the inputs may overshoot to the voltage shown in Table 4–2 based upon the input duty cycle. The DC case is equivalent to 100% duty cycle. During transitions, the inputs may undershoot to –2.0 V for input currents less than 100 mA and periods shorter than 20 ns.
(3) Maximum V_{CC} rise time is 100 ms, and V_{CC} must rise monotonically from ground to V_{CC}.
(4) V_{CCPD} must ramp-up from 0 V to 3.3 V within 100 µs to 100 ms. If V_{CCPD} is not ramped up within this specified time, the Stratix II GX device will not configure successfully. If the system does not allow for a V_{CCPD} ramp-up time of 100 ms or less, hold `nCONFIG` low until all power supplies are reliable.
(5) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT}, V_{CCPD}, and V_{CCIO} are powered.
(6) V_{CCIO} maximum and minimum conditions for PCI and PCI-X are shown in parentheses.

Transceiver Block Characteristics

Tables 4–4 through 4–6 contain transceiver block specifications.

Table 4–4. Stratix II GX Transceiver Block Absolute Maximum Ratings Note (1)					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Units**
V_{CCA}	Transceiver block supply voltage	Commercial and industrial	–0.5	4.6	V
V_{CCP}	Transceiver block supply voltage	Commercial and industrial	–0.5	1.8	V
V_{CCR}	Transceiver block supply Voltage	Commercial and industrial	–0.5	1.8	V
V_{CCT}	Transceiver block supply voltage	Commercial and industrial	–0.5	1.8	V
V_{CCT_B}	Transceiver block supply voltage	Commercial and industrial	–0.5	1.8	V
V_{CCL}	Transceiver block supply voltage	Commercial and industrial	–0.5	1.8	V
V_{CCH_B}	Transceiver block supply voltage	Commercial and industrial	–0.5	2.4	V

Note to Table 4–4:
(1) The device can tolerate prolonged operation at this absolute maximum, as long as the maximum specification is not violated.

Table 4–5. Stratix II GX Transceiver Block Operating Conditions

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCA}	Transceiver block supply voltage	Commercial and industrial	3.135	3.3	3.465	V
V_{CCP}	Transceiver block supply voltage	Commercial and industrial	1.15	1.2	1.25	V
V_{CCR}	Transceiver block supply voltage	Commercial and industrial	1.15	1.2	1.25	V
V_{CCT}	Transceiver block supply voltage	Commercial and industrial	1.15	1.2	1.25	V
V_{CCT_B}	Transceiver block supply voltage	Commercial and industrial	1.15	1.2	1.25	V
V_{CCL}	Transceiver block supply voltage	Commercial and industrial	1.15	1.2	1.25	V
V_{CCH_B} *(2)*	Transceiver block supply voltage	Commercial and industrial	1.15	1.2	1.25	V
			1.425	1.5	1.575	V
R_{REF} *(1)*	Reference resistor	Commercial and industrial	2000 –1%	2000	2000 +1%	Ω

Notes to Table 4–5:
(1) The DC signal on this pin must be as clean as possible. Ensure that no noise is coupled to this pin.
(2) Refer to the *Stratix II GX Device Handbook*, volume 2, for more information.

Table 4–6. Stratix II GX Transceiver Block AC Specification (Part 1 of 6)

Symbol / Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Reference clock											
Input frequency from REFCLK input		50	-	622.08	50	-	622.08	50	-	622.08	MHz
Input frequency from PLD input		50	-	325	50	-	325	50	-	325	MHz
Input clock jitter		Refer to Table 4–20 on page 4–36 for the input jitter specifications for the reference clock.									
Absolute V_{MAX} for a REFCLK pin *(12)*		-	-	3.3	-	-	3.3	-	-	3.3	V

Table 4–6. Stratix II GX Transceiver Block AC Specification (Part 2 of 6)

Symbol / Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Absolute V_{MIN} for a REFCLK pin *(12)*		-0.3	-	-	-0.3	-	-	-0.3	-	-	V
Rise/fall time		-	0.2	-	-	0.2	-	-	0.2	-	UI
Duty cycle		40	-	60	40	-	60	40	-	60	%
Peak-to-peak differential input voltage		200	-	2000	200	-	2000	200	-	2000	mV
Spread-spectrum clocking		30 0 to -0.5%	-	33 0 to -0.5%	30 0 to -0.5%	-	33 0 to -0.5%	30 0 to -0.5%	-	33 0 to -0.5%	kHz
On-chip termination resistors		115 ±20%			115 ±20%			115 ±20%			Ω
V_{ICM} (AC coupled) *(12)*		1200 ±5%			1200 ±5%			1200 ±5%			mV
V_{ICM} (DC coupled) *(4)*		0.25	-	0.55	0.25	-	0.55	0.25	-	0.55	V
Rref		2000 ±1%			2000 ±1%			2000 ±1%			Ω
Transceiver Clocks											
Calibration block clock frequency		10	-	125	10	-	125	10	-	125	MHz
Calibration block minimum power-down pulse width		30	-	-	30	-	-	30	-	-	ns
Time taken for one-time calibration		-	-	8	-	-	8	-	-	8	ms
fixedclk clock frequency	PCI Express Receiver Detect	-	125	-	-	125	-	-	125	-	MHz
	Adaptive Equalization (AEQ)	2.5	-	125	2.5	-	125	-	-	-	MHz

Table 4–6. Stratix II GX Transceiver Block AC Specification (Part 3 of 6)

Symbol / Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
reconfig_clk clock frequency		2.5	-	50	2.5	-	50	2.5	-	50	MHz
Transceiver block minimum power-down pulse width		100	-	-	100	-	-	100	-	-	ns
Receiver											
Data rate		600	-	6375	600	-	5000	600	-	4250	Mbps
Absolute V_{MAX} for a receiver pin *(1)*		-	-	2.0	-	-	2.0	-	-	2.0	V
Absolute V_{MIN} for a receiver pin		-0.4	-	-	-0.4	-	-	-0.4	-	-	V
Maximum peak-to-peak differential input voltage V_{ID} (diff p-p)	V_{CM} = 0.85 V	-	-	3.3	-	-	3.3	-	-	3.3	V
Minimum peak-to-peak differential input voltage V_{ID} (diff p-p)	V_{CM} = 0.85 V DC Gain = ≥ 3 dB	160	-	-	160	-	-	160	-	-	mV
V_{ICM}	V_{ICM} = 0.85 V setting	850±10%			850±10%			850±10%			mV
	V_{ICM} = 1.2 V setting *(11)*	1200±10%			1200±10%			1200±10%			mV
On-chip termination resistors	100 Ω setting	100±15%			100±15%			100±15%			Ω
	120 Ω setting	120±15%			120±15%			120±15%			Ω
	150 Ω setting	150±15%			150±15%			150±15%			Ω
Bandwidth at 6.375 Gbps	BW = Low	-	20	-	-	-	-	-	-	-	MHz
	BW = Med	-	35	-	-	-	-	-	-	-	MHz
	BW = High	-	45	-	-	-	-	-	-	-	MHz

Table 4–6. Stratix II GX Transceiver Block AC Specification (Part 4 of 6)

Symbol / Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Bandwidth at 3.125 Gbps	BW = Low	-	30	-	-	30	-	-	30	-	MHz
	BW = Med	-	40	-	-	40	-	-	40	-	MHz
	BW = High	-	50	-	-	50	-	-	50	-	MHz
Bandwidth at 2.5 Gbps	BW = Low	-	35	-	-	35	-	-	35	-	MHz
	BW = Med	-	50	-	-	50	-	-	50	-	MHz
	BW = High	-	60	-	-	60	-	-	60	-	MHz
Return loss differential mode		100 MHz to 2.5 GHz (XAUI): -10 dB 50 MHz to 1.25 GHz (PCI-E): -10 dB 100 MHz to 4.875 GHz (OIF/CEI): -8dB 4.875 GHz to 10 GHz (OIF/CEI): 16.6 dB/decade slope									
Return loss common mode		100 MHz to 2.5 GHz (XAUI): -6 dB 50 MHz to 1.25 GHz (PCI-E): -6 dB 100 MHz to 4.875 GHz (OIF/CEI): -6dB 4.875 GHz to 10 GHz (OIF/CEI): 16.6 dB/decade slope									
Programmable PPM detector *(2)*		±62.5, 100, 125, 200, 250, 300, 500, 1000			±62.5, 100, 125, 200, 250, 300, 500, 1000			±62.5, 100, 125, 200, 250, 300, 500, 1000			ppm
Run length *(3)*, *(9)*		80			80			80			UI
Programmable equalization		-	-	16	-	-	16	-	-	16	dB
Signal detect/loss threshold *(4)*		65	-	175	65	-	175	65	-	175	mV
CDR LTR TIme *(5)*, *(9)*		-	-	75	-	-	75	-	-	75	us
CDR Minimum T1b *(6)*, *(9)*		15	-	-	15	-	-	15	-	-	us
LTD lock time *(7)*, *(9)*		0	100	4000	0	100	4000	0	100	4000	ns
Data lock time from `rx_freqlocked` *(8)*, *(9)*		-	-	4	-	-	4	-	-	4	us
Programmable DC gain		0, 3, 6			0, 3, 6			0, 3, 6			dB
Transmitter											

Table 4–6. Stratix II GX Transceiver Block AC Specification (Part 5 of 6)

Symbol / Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Data rate		600	-	6375	600	-	5000	600	-	4250	Mbps
V_{OCM}	V_{OCM} = 0.6 V setting	580±10%			580±10%			580±10%			mV
	V_{OCM} = 0.7 V setting	680±10%			680±10%			680±10%			mV
On-chip termination resistors	100 Ω setting	108±10%			108±10%			108±10%			Ω
	120 Ω setting	125±10%			125±10%			125±10%			Ω
	150 Ω setting	152±10%			152±10%			152±10%			Ω
Return loss differential mode	312 MHz to 625 MHz (XAUI): -10 dB 625 MHz to 3.125 GHz (XAUI): -10 dB/decade slope 50 MHz to 1.25 GHz (PCI-E): -10dB 100 MHz to 4.875 GHz (OIF/CEI): -8db 4.875 GHz to 10 GHz (OIF/CEI): 16.6 dB/decade slope										
Return loss common mode	50 MHz to 1.25 GHz (PCI-E): -6dB 100 MHz to 4.875 GHz (OIF/CEI): -6db 4.875 GHz to 10 GHz (OIF/CEI): 16.6 dB/decade slope										
Rise time		35	-	65	35	-	65	35	-	65	ps
Fall time		35	-	65	35	-	65	35	-	65	ps
Intra differential pair skew	V_{OD} = 800 mV	-	-	15	-	-	15	-	-	15	ps
Intra-transceiver block skew (x4)		-	-	100	-	-	100	-	-	100	ps
Inter-transceiver block skew (x8)		-	-	300	-	-	300	-	-	300	ps
TXPLL (TXPLL0 and TXPLL1)											
VCO frequency range (low gear)		500	-	1562.5	500	-	1562.5	500	-	1562.5	MHz
VCO frequency range (high gear)		1562.5		3187.5	1562.5		2500	1562.5	-	2125	MHz

Table 4–6. Stratix II GX Transceiver Block AC Specification (Part 6 of 6)

Symbol / Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Bandwidth at 6.375 Gbps	BW = Low	-	2	-	-	-	-	-	-	-	MHz
	BW = Med	-	3	-	-	-	-	-	-	-	MHz
	BW = High	-	7	-	-	-	-	-	-	-	MHz
Bandwidth at 3.125 Gbps	BW = Low	-	3	-	-	3	-	-	3	-	MHz
	BW = Med	-	5	-	-	5	-	-	5	-	MHz
	BW = High	-	9	-	-	9	-	-	9	-	MHz
Bandwidth at 2.5 Gbps	BW = Low	-	1	-	-	1	-	-	1	-	MHz
	BW = Med	-	2	-	-	2	--	-	2	-	MHz
	BW = High	-	4	-	-	4	-	-	4	-	MHz
TX PLL lock time from `gxb_powerdown` deassertion *(9)*, *(10)*		-	-	100	-	-	100	-	-	100	us
PLD-Transceiver Interface											
Interface speed		25	-	250	25	-	250	25	-	200	MHz
Digital Reset Pulse Width		Minimum is 2 parallel clock cycles									

Notes to Table 4–6:

(1) The device cannot tolerate prolonged operation at this absolute maximum. Refer to Figure 4–5 for more information.
(2) The rate matcher supports only up to +/-300 ppm.
(3) This parameter is measured by embedding the run length data in a PRBS sequence.
(4) This feature is only available in PCI-Express (PIPE) mode.
(5) Time taken to `rx_pll_locked` goes high from `rx_analogreset` deassertion. Refer to Figure 4–1.
(6) This is how long GXB needs to stay in LTR mode after `rx_pll_locked` is asserted and before `rx_locktodata` is asserted in manual mode. Refer to Figure 4–1.
(7) Time taken to recover valid data from GXB after `rx_locktodata` signal is asserted in manual mode. Measurement results are based on PRBS31, for native data rates only. Refer to Figure 4–1.
(8) Time taken to recover valid data from GXB after `rx_freqlocked` signal goes high in automatic mode. Measurement results are based on PRBS31, for native data rates only. Refer to Figure 4–1.
(9) Please refer to the Protocol Characterization documents for lock times specific to the protocols.
(10) Time taken to lock TX PLL from `gxb_powerdown` deassertion.
(11) The 1.2 V RX V_{ICM} setting is intended for DC-coupled LVDS links.
(12) For AC-coupled links, the on-chip biasing circuit is switched off before and during configuration. Make sure that input specifications are not violated during this period.

Figure 4–1 shows the lock time parameters in manual mode, Figure 4–2 shows the lock time parameters in automatic mode.

LTD = Lock to data
LTR = Lock to reference clock

Figure 4–1. Lock Time Parameters for Manual Mode

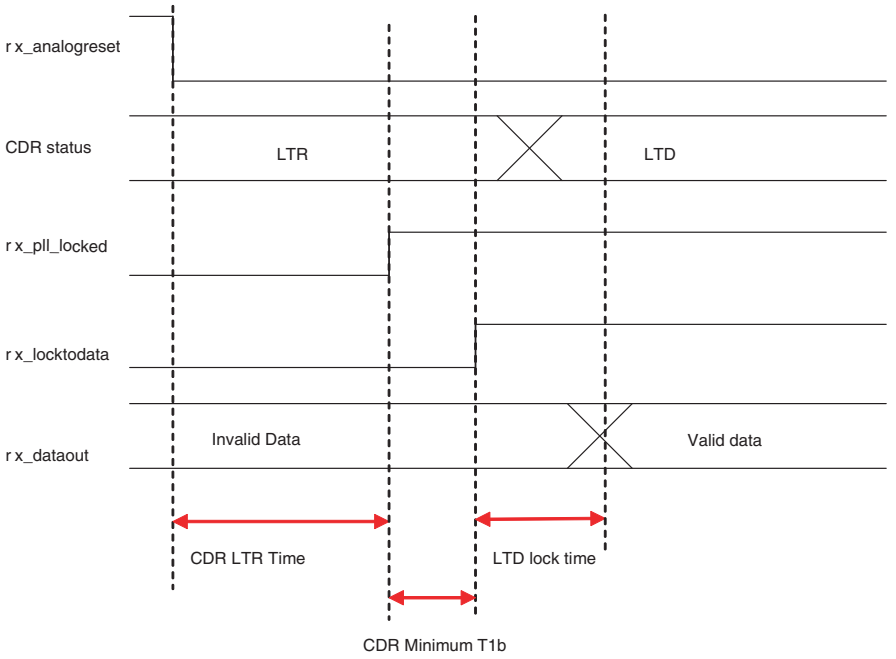
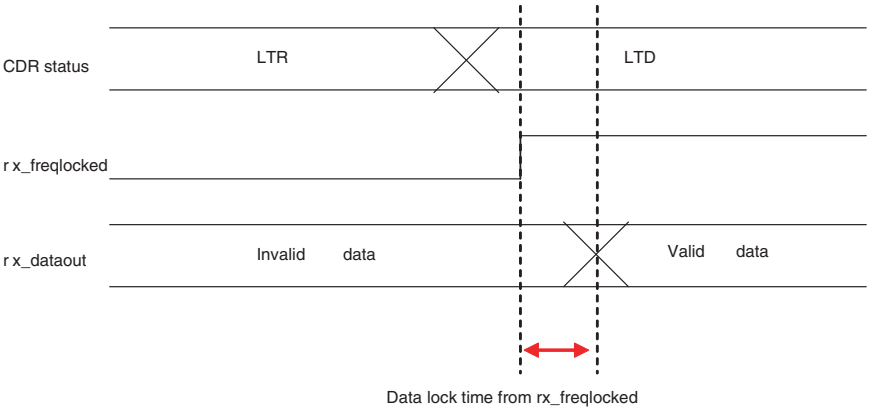


Figure 4–2. Lock Time Parameters for Automatic Mode



Figures 4–3 and 4–4 show differential receiver input and transmitter output waveforms, respectively.

Figure 4–3. Receiver Input Waveform

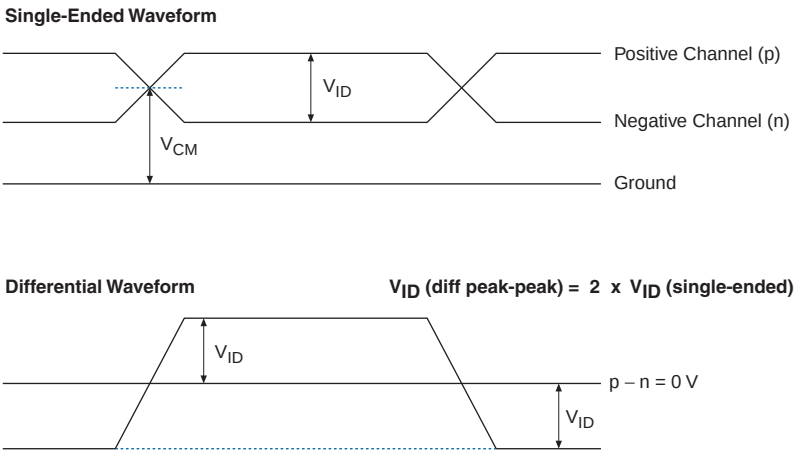


Figure 4–4. Transmitter Output Waveform

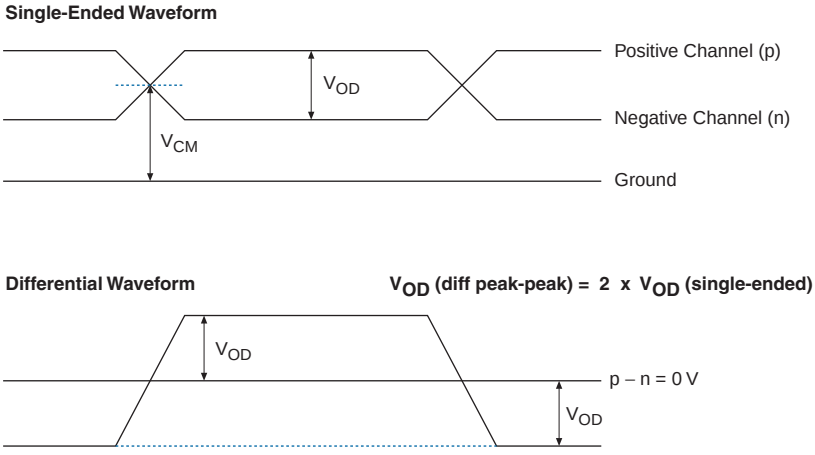
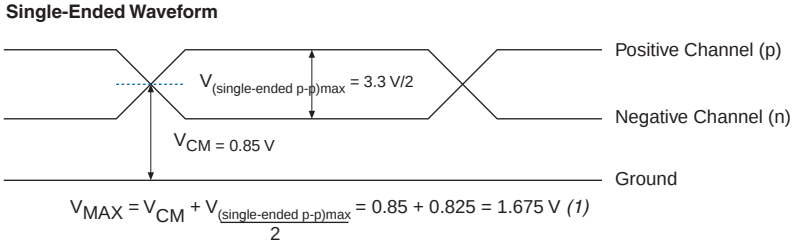


Figure 4–5. Maximum Receiver Input Pin Voltage



Note to Figure 4–5:
(1) The absolute V_{MAX} that the receiver input pins can tolerate is 2 V.

Tables 4–7 through 4–12 show the typical V_{OD} for data rates from 600 Mbps to 6.375 Gbps. The specification is for measurement at the package ball.

Table 4–7. Typical V_{OD} Setting, TX Term = 100 Ω Note (1)

V_{CCH} TX = 1.5 V	V_{OD} Setting (mV)						
	200	400	600	800	1000	1200	1400
V_{OD} Typical (mV)	220	430	625	830	1020	1200	1350

Note to Table 4–7:
(1) Applicable to data rates from 600 Mbps to 6.375 Gbps. Specification is for measurement at the package ball.

Table 4–8. Typical V_{OD} Setting, TX Term = 120 Ω Note (1)					
V_{CCH} TX = 1.5 V	**V_{OD} Setting (mV)**				
	240	**480**	**720**	**960**	**1200**
V_{OD} Typical (mV)	260	510	750	975	1200

Note to Table 4–8:
(1) Applicable to data rates from 600 Mbps to 6.375 Gbps. Specification is for measurement at the package ball.

Table 4–9. Typical V_{OD} Setting, TX Term = 150 Ω Note (1)				
V_{CCH} TX = 1.5 V	**V_{OD} Setting (mV)**			
	300	**600**	**900**	**1200**
V_{OD} Typical (mV)	325	625	920	1200

Note to Table 4–9:
(1) Applicable to data rates from 600 Mbps to 6.375 Gbps. Specification is for measurement at the package ball.

Table 4–10. Typical V_{OD} Setting, TX Term = 100 Ω Note (1)					
V_{CCH} TX = 1.2 V	**V_{OD} Setting (mV)**				
	320	**480**	**640**	**800**	**960**
V_{OD} Typical (mV)	344	500	664	816	960

Note to Table 4–10:
(1) Applicable to data rates from 600 Mbps to 3.125 Gbps. Specification is for measurement at the package ball.

Table 4–11. Typical V_{OD} Setting, TX Term = 120 Ω Note (1)					
V_{CCH} TX = 1.2 V	V_{OD} Setting (mV)				
	192	384	576	768	960
V_{OD} Typical (mV)	210	410	600	780	960

Note to Table 4–11:
(1) Applicable to data rates from 600 Mbps to 3.125 Gbps. Specification is for measurement at the package ball.

Table 4–12. Typical V_{OD} Setting, TX Term = 150 Ω Note (1)				
V_{CCH} TX = 1.2 V	V_{OD} Setting (mV)			
	240	480	720	960
V_{OD} Typical (mV)	260	500	730	960

Note to Table 4–12:
(1) Applicable to data rates from 600 Mbps to 3.125 Gbps. Specification is for measurement at the package ball.

Tables 4–13 through 4–18 show the typical first post-tap pre-emphasis.

Table 4–13. Typical Pre-Emphasis (First Post-Tap), Note (1) (Part 1 of 2)												
V_{CCH} TX = 1.5 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
	TX Term = 100 Ω											
400	24%	62%	112%	184%								
600		31%	56%	86%	122%	168%	230%	329%	457%			
800		20%	35%	53%	73%	96%	123%	156%	196%	237%	312%	387%
1000			23%	36%	49%	64%	79%	97%	118%	141%	165%	200%
1200			17%	25%	35%	45%	56%	68%	82%	95%	110%	125%

Table 4–13. Typical Pre-Emphasis (First Post-Tap), Note (1) (Part 2 of 2)

V_{CCH} TX = 1.5 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
1400				20%	26%	33%	41%	51%	58%	67%	77%	86%

Note to Table 4–13:
(1) Applicable to data rates from 600 Mbps to 6.375 Gbps. Specification is for measurement at the package ball.

Table 4–14. Typical Pre-Emphasis (First Post-Tap), Note (1)

V_{CCH} TX = 1.5 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
	TX Term = 120 Ω											
240	45%											
480		41%	76%	114%	166%	257%	355%					
720		23%	38%	55%	84%	108%	137%	179%	226%	280%	405%	477%
960		15%	24%	36%	47%	64%	80%	97%	122%	140%	170%	196%
1200			18%	22%	30%	41%	51%	63%	77%	86%	98%	116%

Note to Table 4–14:
(1) Applicable to data rates from 600 Mbps to 6.375 Gbps. Specification is for measurement at the package ball.

Table 4–15. Typical Pre-Emphasis (First Post-Tap), Note (1) (Part 1 of 2)

V_{CCH} TX = 1.5 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
	TX Term = 150 Ω											
300	32%	85%										

Table 4–15. Typical Pre-Emphasis (First Post-Tap), Note (1) (Part 2 of 2)

V_{CCH} TX = 1.5 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
600		33%	53%	80%	115%	157%	195%	294%	386%			
900		19%	28%	38%	56%	70%	86%	113%	133%	168%	196%	242%
1200			17%	22%	31%	40%	52%	62%	75%	86%	96%	112%

Note to Table 4–15:
(1) Applicable to data rates from 600 Mbps to 6.375 Gbps. Specification is for measurement at the package ball.

Table 4–16. Typical Pre-Emphasis (First Post-Tap), Note (1)

V_{CCH} TX = 1.2 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
	TX Term = 100 Ω											
320	24%	61%	114%									
480		31%	55%	86%	121%	170%	232%	333%				
640		20%	35%	54%	72%	95%	124%	157%	195%	233%	307%	373%
800			23%	36%	49%	64%	81%	97%	117%	140%	161%	195%
960			18%	25%	35%	44%	57%	69%	82%	94%	108%	127%

Note to Table 4–16:
(1) Applicable to data rates from 600 Mbps to 3.125 Gbps. Specification is for measurement at the package ball.

Table 4–17. Typical Pre-Emphasis (First Post-Tap), Note (1)

V_{CCH} TX = 1.2 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
	TX Term = 120 Ω											
192	45%											
384		41%	76%	114%	166%	257%	355%					
576		23%	38%	55%	84%	108%	137%	179%	226%	280%	405%	477%
768		15%	24%	36%	47%	64%	80%	97%	122%	140%	170%	196%
960			18%	22%	30%	41%	51%	63%	77%	86%	98%	116%

Note to Table 4–17:
(1) Applicable to data rates from 600 Mbps to 3.125 Gbps. Specification is for measurement at the package ball.

Table 4–18. Typical Pre-Emphasis (First Post-Tap), Note (1)

V_{CCH} TX = 1.2 V	First Post Tap Pre-Emphasis Level											
V_{OD} Setting (mV)	1	2	3	4	5	6	7	8	9	10	11	12
	TX Term = 150 Ω											
240	31%	85%										
480		32%	52%	78%	112%	152%	195%	275%				
720		19%	28%	37%	56%	68%	86%	108%	133%	169%	194%	239%
960			17%	22%	30%	39%	51%	59%	75%	85%	94%	109%

Note to Table 4–18:
(1) Applicable to data rates from 600 Mbps to 3.125 Gbps. Specification is for measurement at the package ball.

Table 4–19 shows the Stratix II GX transceiver block AC specifications.

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 1 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
SONET/SDH Transmit Jitter Generation *(7)*											
Peak-to-peak jitter at 622.08 Mbps	REFCLK = 77.76 MHz Pattern = PRBS23 V_{OD} = 800 mV No Pre-emphasis	-	-	0.1	-	-	0.1	-	-	0.1	UI
RMS jitter at 622.08 Mbps	REFCLK = 77.76 MHz Pattern = PRBS23 V_{OD} = 800 mV No Pre-emphasis	-	-	0.01	-	-	0.01	-	-	0.01	UI
Peak-to-peak jitter at 2488.32 Mbps	REFCLK = 155.52 MHz Pattern = PRBS23 V_{OD} = 800 mV No Pre-emphasis	-	-	0.1	-	-	0.1	-	-	0.1	UI
RMS jitter at 2488.32 Mbps	REFCLK = 155.52 MHz Pattern = PRBS23 V_{OD} = 800 mV No Pre-emphasis	-	-	0.01	-	-	0.01	-	-	0.01	UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 2 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
SONET/SDH Receiver Jitter Tolerance *(7)*											
Jitter tolerance at 622.08 Mbps	Jitter frequency = 0.03 KHz Pattern = PRBS23 No Equalization DC Gain = 0 dB		> 15			> 15			> 15		UI
	Jitter frequency = 25 KHZ Pattern = PRBS23 No Equalization DC Gain = 0 dB		> 1.5			> 1.5			> 1.5		UI
	Jitter frequency = 250 KHz Pattern = PRBS23 No Equalization DC Gain = 0 dB		> 0.15			> 0.15			> 0.15		UI
Jitter tolerance at 2488.32 MBps	Jitter frequency = 0.06 KHz Pattern = PRBS23 No Equalization DC Gain = 0 dB		> 15			> 15			> 15		UI
	Jitter frequency = 100 KHZ Pattern = PRBS23 No Equalization DC Gain = 0 dB		> 1.5			> 1.5			> 1.5		UI
	Jitter frequency = 1 MHz Pattern = PRBS23 No Equalization DC Gain = 0 dB		> 0.15			> 0.15			> 0.15		UI
	Jitter frequency = 10 MHz Pattern = PRBS23 No Equalization DC Gain = 0 dB		> 0.15			> 0.15			> 0.15		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 3 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Fibre Channel Transmit Jitter Generation *(8), (17)*											
Total jitter FC-1	REFCLK = 106.25 MHz Pattern = CRPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.23	-	-	0.23	-	-	0.23	UI
Deterministic jitter FC-1	REFCLK = 106.25 MHz Pattern = CRPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.11	-	-	0.11	-	-	0.11	UI
Total jitter FC-2	REFCLK = 106.25 MHz Pattern = CRPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.33	-	-	0.33	-	-	0.33	UI
Deterministic jitter FC-2	REFCLK = 106.25 MHz Pattern = CRPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.2	-	-	0.2	-	-	0.2	UI
Total jitter FC-4	REFCLK = 106.25 MHz Pattern = CRPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.52	-	-	0.52	-	-	0.52	UI
Deterministic jitter FC-4	REFCLK = 106.25 MHz Pattern = CRPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.33	-	-	0.33	-	-	0.33	UI
Fibre Channel Receiver Jitter Tolerance *(8), (18)*											
Deterministic jitter FC-1	Pattern = CJTPAT No Equalization DC Gain = 0 dB	> 0.37			> 0.37			> 0.37			UI
Random jitter FC-1	Pattern = CJTPAT No Equalization DC Gain = 0 dB	> 0.31			> 0.31			> 0.31			UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 4 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Sinusoidal jitter FC-1	Fc/25000	> 1.5			> 1.5			> 1.5			UI
	Fc/1667	> 0.1			> 0.1			> 0.1			UI
Deterministic jitter FC-2	Pattern = CJTPAT No Equalization DC Gain = 0 dB	> 0.33			> 0.33			> 0.33			UI
Random jitter FC-2	Pattern = CJTPAT No Equalization DC Gain = 0 dB	> 0.29			> 0.29			> 0.29			UI
Sinusoidal jitter FC-2	Fc/25000	> 1.5			> 1.5			> 1.5			UI
	Fc/1667	> 0.1			> 0.1			> 0.1			UI
Deterministic jitter FC-4	Pattern = CJTPAT No Equalization DC Gain = 0 dB	> 0.33			> 0.33			> 0.33			UI
Random jitter FC-4	Pattern = CJTPAT No Equalization DC Gain = 0 dB	> 0.29			> 0.29			> 0.29			UI
Sinusoidal jitter FC-4	Fc/25000	> 1.5			> 1.5			> 1.5			UI
	Fc/1667	> 0.1			> 0.1			> 0.1			UI
XAUI Transmit Jitter Generation *(9)*											
Total jitter at 3.125 Gbps	REFCLK = 156.25 MHz Pattern = CJPAT V_{OD} = 1200 mV No Pre-emphasis	-	-	0.3	-	-	0.3	-	-	0.3	UI
Deterministic jitter at 3.125 Gbps	REFCLK = 156.25 MHz Pattern = CJPAT V_{OD} = 1200 mV No Pre-emphasis	-	-	0.17	-	-	0.17	-	-	0.17	UI
XAUI Receiver Jitter Tolerance *(9)*											
Total jitter	Pattern = CJPAT No Equalization DC Gain = 3 dB	> 0.65			> 0.65			> 0.65			UI
Deterministic jitter	Pattern = CJPAT No Equalization DC Gain = 3 dB	> 0.37			> 0.37			> 0.37			UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 5 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Peak-to-peak jitter	Jitter frequency = 22.1 KHz		> 8.5			> 8.5			> 8.5		UI
Peak-to-peak jitter	Jitter frequency = 1.875 MHz		> 0.1			> 0.1			> 0.1		UI
Peak-to-peak jitter	Jitter frequency = 20 MHz		> 0.1			> 0.1			> 0.1		UI
PCI Express Transmit Jitter Generation *(10)*											
Total jitter at 2.5 Gbps	Compliance pattern V_{OD} = 800 mV Pre-emphasis (1st post-tap) = Setting 5	-	-	0.25	-	-	0.25	-	-	0.25	UI
PCI Express Receiver Jitter Tolerance *(10)*											
Total jitter at 2.5 Gbps	Compliance pattern No Equalization DC gain = 3 dB		> 0.6			> 0.6			> 0.6		UI
Serial RapidIO Transmit Jitter Generation *(11)*											
Deterministic Jitter (peak-to-peak)	Data Rate = 1.25, 2.5, 3.125 Gbps REFCLK = 125 MHz Pattern = CJPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.17	-	-	0.17	-	-	0.17	UI
Total Jitter (peak-to-peak)	Data Rate = 1.25, 2.5, 3.125 Gbps REFCLK = 125 MHz Pattern = CJPAT V_{OD} = 800 mV No Pre-emphasis	-	-	0.35	-	-	0.35	-	-	0.35	UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 6 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Serial RapidIO Receiver Jitter Tolerance *(11)*											
Deterministic Jitter Tolerance (peak-to-peak)	Data Rate = 1.25, 2.5, 3.125 Gbps REFCLK = 125 MHz Pattern = CJPAT Equalizer Setting = 0 for 1.25 Gbps Equalizer Setting = 6 for 2.5 Gbps Equalizer Setting = 6 for 3.125 Gbps	> 0.37			> 0.37			> 0.37			UI
Combined Deterministic and Random Jitter Tolerance (peak-to-peak)	Data Rate = 1.25, 2.5, 3.125 Gbps REFCLK = 125 MHz Pattern = CJPAT Equalizer Setting = 0 for 1.25 Gbps Equalizer Setting = 6 for 2.5 Gbps Equalizer Setting = 6 for 3.125 Gbps	> 0.55			> 0.55			> 0.55			UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 7 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Sinusoidal Jitter Tolerance (peak-to-peak)	Jitter Frequency = 22.1 KHz Data Rate = 1.25, 2.5, 3.125 Gbps REFCLK = 125 MHz Pattern = CJPAT Equalizer Setting = 0 for 1.25 Gbps Equalizer Setting = 6 for 2.5 Gbps Equalizer Setting = 6 for 3.125 Gbps		> 8.5			> 8.5			> 8.5		UI
	Jitter Frequency = 1.875 MHz Data Rate = 1.25, 2.5, 3.125 Gbps REFCLK = 125 MHz Pattern = CJPAT Equalizer Setting = 0 for 1.25 Gbps Equalizer Setting = 6 for 2.5 Gbps Equalizer Setting = 6 for 3.125 Gbps		> 0.1			> 0.1			> 0.1		UI
	Jitter Frequency = 20 MHz Data Rate = 1.25, 2.5, 3.125 Gbps REFCLK = 125 MHz Pattern = CJPAT Equalizer Setting = 0 for 1.25 Gbps Equalizer Setting = 6 for 2.5 Gbps Equalizer Setting = 6 for 3.125 Gbps		> 0.1			> 0.1			> 0.1		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 8 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
GIGE Transmit Jitter Generation *(12)*											
Deterministic Jitter (peak-to-peak)	Data Rate = 1.25 Gbps REFCLK = 125 MHz Pattern = CRPAT V_{OD} = 1400 mV No Pre-emphasis	-	-	0.14	-	-	0.14	-	-	0.14	UI
Total Jitter (peak-to-peak)	Data Rate = 1.25 Gbps REFCLK = 125 MHz Pattern = CRPAT V_{OD} = 1400 mV No Pre-emphasis	-	-	0.279	-	-	0.279	-	-	0.279	UI
GIGE Receiver Jitter Tolerance *(12)*											
Deterministic Jitter Tolerance (peak-to-peak)	Data Rate = 1.25 Gbps REFCLK = 125 MHz Pattern = CJPAT No Equalization	> 0.4			> 0.4			> 0.4			UI
Combined Deterministic and Random Jitter Tolerance (peak-to-peak)	Data Rate = 1.25 Gbps REFCLK = 125 MHz Pattern = CJPAT No Equalization	> 0.66			> 0.66			> 0.66			UI
HiGig Transmit Jitter Generation *(4), (13)*											
Deterministic Jitter (peak-to-peak)	Data Rate = 3.75 Gbps REFCLK = 187.5 MHz Pattern = CJPAT V_{OD} = 1200 mV No Pre-emphasis	-	-	0.17	-						UI
Total Jitter (peak-to-peak)	Data Rate = 3.75 Gbps REFCLK = 187.5 MHz Pattern = CJPAT V_{OD} = 1200 mV No Pre-emphasis	-	-	0.35	-						UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 9 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
HiGig Receiver Jitter Tolerance *(13)*											
Deterministic Jitter Tolerance (peak-to-peak)	Data Rate = 3.75 Gbps REFCLK = 187.5 MHz Pattern = CJPAT No Equalization DC Gain = 3 dB		> 0.37			-			-		UI
Combined Deterministic and Random Jitter Tolerance (peak-to-peak)	Data Rate = 3.75 Gbps REFCLK = 187.5 MHz Pattern = CJPAT No Equalization DC Gain = 3 dB		> 0.65			-			-		UI
	Jitter Frequency = 22.1 KHz Data Rate = 3.75 Gbps REFCLK = 187.5 MHz Pattern = CJPAT No Equalization DC Gain = 3 dB		> 8.5			-			-		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 10 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Sinusoidal Jitter Tolerance (peak-to-peak)	Jitter Frequency = 1.875 MHz Data Rate = 3.75 Gbps REFCLK = 187.5 MHz Pattern = CJPAT No Equalization DC Gain = 3 dB	> 0.1			-			-			UI
	Jitter Frequency = 20 MHz Data Rate = 3.75 Gbps REFCLK = 187.5 MHz Pattern = CJPAT No Equalization DC Gain = 3 dB	> 0.1			-			-			UI
(OIF) CEI Transmitter Jitter Generation *(14)*											
Total Jitter (peak-to-peak)	Data Rate = 6.375 Gbps REFCLK = 318.75 MHz Pattern = PRBS15 Vod=1000 mV *(5)* No Pre-emphasis BER = 10^{-12}			0.3			N/A			N/A	UI
(OIF) CEI Receiver Jitter Tolerance *(14)*											
Deterministic Jitter Tolerance (peak-to-peak)	Data Rate = 6.375 Gbps Pattern = PRBS31 Equalizer Setting = 15 DC Gain = 0 dB BER = 10^{-12}	> 0.675			N/A			N/A			UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 11 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Combined Deterministic and Random Jitter Tolerance (peak-to-peak)	Data Rate = 6.375 Gbps Pattern = PRBS31 Equalizer Setting = 15 DC Gain = 0 dB BER = 10^{-12}	> 0.988			N/A			N/A			UI
Sinusoidal Jitter Tolerance (peak-to-peak)	Jitter Frequency = 38.2 KHz Data Rate = 6.375 Gbps Pattern = PRBS31 Equalizer Setting = 15 DC Gain = 0 dB BER = 10^{-12}	> 5			N/A			N/A			UI
	Jitter Frequency = 3.82 MHz Data Rate=6.375 Gbps Pattern = PRBS31 Equalizer Setting = 15 DC Gain = 0 dB BER = 10^{-12}	> 0.05			N/A			N/A			UI
	Jitter Frequency = 20 MHz Data Rate = 6.375 Gbps Pattern = PRBS31 Equalizer Setting = 15 DC Gain = 0 dB BER = 10^{-12}	> 0.05			N/A			N/A			UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 12 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
CPRI Transmitter Jitter Generation *(15)*											
Deterministic Jitter (peak-to-peak)	Data Rate = 614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps REFCLK = 61.44 MHz for 614.4 Mbps and 1.2288 Gbps REFCLK = 122.88 MHz for 2.4576 Gbps Pattern = CJPAT Vod = 1400 mV No Pre-emphasis			0.14			0.14			N/A	UI
Total Jitter (peak-to-peak)	Data Rate = 614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps REFCLK = 61.44 MHz for 614.4 Mbps and 1.2288 Gbps REFCLK = 122.88 MHz for 2.4576 Gbps Pattern = CJPAT Vod = 1400 mV No Pre-emphasis			0.279			0.279			N/A	UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 13 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
CPRI Receiver Jitter Tolerance *(15)*											
Deterministic Jitter Tolerance (peak-to-peak)	Data Rate = 614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps REFCLK = 61.44 MHz for 614.4 Mbps REFCLK = 122.88 MHz for 1.2288 Gbps and 2.4576 Gbps Pattern = CJPAT Equalizer Setting = 6 DC Gain = 0 dB		> 0.4			> 0.4			N/A		UI
Combined Deterministic and Random Jitter Tolerance (peak-to-peak)	Data Rate = 614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps REFCLK = 61.44 MHz for 614.4 Mbps REFCLK = 122.88 MHz for 1.2288 Gbps and 2.4576 Gbps Pattern = CJPAT Equalizer Setting = 6 DC Gain = 0 dB		> 0.66			> 0.66			N/A		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 14 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Sinusoidal Jitter Tolerance (peak-to-peak) *(6)*	Jitter Frequency = 22.1 KHz Data Rate = 614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps REFCLK = 61.44 MHz for 614.4 Mbps REFCLK = 122.88 MHz for 1.2288 Gbps and 2.4576 Gbps Pattern = CJPAT Equalizer Setting = 6 DC Gain = 0 dB		> 8.5			> 8.5			N/A		UI
	Jitter Frequency = 1.875 MHz Data Rate = 614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps REFCLK = 61.44 MHz for 614.4 Mbps REFCLK = 122.88 MHz for 1.2288 Gbps and 2.4576 Gbps Pattern = CJPAT Equalizer Setting = 6 DC Gain = 0 dB		> 0.1			> 0.1			N/A		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 15 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Sinusoidal Jitter Tolerance (peak-to-peak) *(6)* (cont.)	Jitter Frequency = 20 MHz Data Rate = 614.4 Mbps, 1.2288 Gbps, 2.4576 Gbps REFCLK = 61.44 MHz for 614.4 Mbps REFCLK = 122.88 MHz for 1.2288 Gbps and 2.4576 Gbps Pattern = CJPAT Equalizer Setting = 6 DC Gain = 0 dB	> 0.1			> 0.1			N/A			UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 16 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
SDI Transmitter Jitter Generation *(16)*											
Alignment Jitter (peak-to-peak)	Data Rate = 1.485 Gbps (HD) REFCLK = 74.25 MHz Pattern = ColorBar Vod = 800 mV No Pre-emphasis Low-Frequency Roll-Off = 100 KHz		0.2			0.2			0.2		UI
	Data Rate = 2.97 Gbps (3G) REFCLK = 148.5 MHz Pattern = ColorBar Vod = 800 mV No Pre-emphasis Low-Frequency Roll-Off = 100 KHz		0.3			0.3			0.3		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 17 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
SDI Receiver Jitter Tolerance *(16)*											
Sinusoidal Jitter Tolerance (peak-to-peak)	Jitter Frequency = 15 KHz Data Rate = 2.97 Gbps (3G) REFCLK = 148.5 MHz Pattern = Single Line Scramble Color Bar No Equalization DC Gain = 0 dB		> 2			> 2			> 2		UI
	Jitter Frequency = 100 KHz Data Rate = 2.97 Gbps (3G) REFCLK = 148.5 MHz Pattern = Single Line Scramble Color Bar No Equalization DC Gain = 0 dB		> 0.3			> 0.3			> 0.3		UI
	Jitter Frequency = 148.5 MHz Data Rate = 2.97 Gbps (3G) REFCLK = 148.5 MHz Pattern = Single Line Scramble Color Bar No Equalization DC Gain = 0 dB		> 0.3			> 0.3			> 0.3		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 18 of 19)

Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Sinusoidal Jitter Tolerance (peak-to-peak)	Jitter Frequency = 20 KHz Data Rate = 1.485 Gbps (HD) REFCLK = 74.25 MHz Pattern = 75% Color Bar No Equalization DC Gain = 0 dB		> 1			> 1			> 1		UI
	Jitter Frequency = 100 KHz Data Rate = 1.485 Gbps (HD) REFCLK = 74.25 MHz Pattern = 75% Color Bar No Equalization DC Gain = 0 dB		> 0.2			> 0.2			> 0.2		UI
	Jitter Frequency = 148.5 MHz Data Rate = 1.485 Gbps (HD) REFCLK = 74.25 MHz Pattern = 75% Color Bar No Equalization DC Gain = 0 dB		> 0.2			> 0.2			> 0.2		UI

Table 4–19. Stratix II GX Transceiver Block AC Specification Notes (1), (2), (3) (Part 19 of 19)											
Symbol/ Description	Conditions	-3 Speed Commercial Speed Grade			-4 Speed Commercial and Industrial Speed Grade			-5 Speed Commercial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	

Notes to Table 4–19:

(1) Dedicated REFCLK pins were used to drive the input reference clocks.
(2) Jitter numbers specified are valid for the stated conditions only.
(3) Refer to the protocol characterization documents for detailed information.
(4) HiGig configuration is available in a -3 speed grade only. For more information, refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
(5) Stratix II GX transceivers meet CEI jitter generation specification of 0.3 UI for a V_{OD} range of 400 mV to 1000 mV.
(6) The Sinusoidal Jitter Tolerance Mask is defined only for low voltage (LV) variant of CPRI.
(7) The jitter numbers for SONET/SDH are compliant to the GR-253-CORE Issue 3 Specification.
(8) The jitter numbers for Fibre Channel are compliant to the FC-PI-4 Specification revision 6.10.
(9) The jitter numbers for XAUI are compliant to the IEEE802.3ae-2002 Specification.
(10) The jitter numbers for PCI Express are compliant to the PCIe Base Specification 2.0.
(11) The jitter numbers for Serial RapidIO are compliant to the RapidIO Specification 1.3.
(12) The jitter numbers for GIGE are compliant to the IEEE802.3-2002 Specification.
(13) The jitter numbers for HiGig are compliant to the IEEE802.3ae-2002 Specification.
(14) The jitter numbers for (OIF) CEI are compliant to the OIF-CEI-02.0 Specification.
(15) The jitter numbers for CPRI are compliant to the CPRI Specification V2.1.
(16) The HD-SDI and 3G-SDI jitter numbers are compliant to the SMPTE292M and SMPTE424M Specifications.
(17) The Fibre Channel transmitter jitter generation numbers are compliant to the specification at β_T interoperability point.
(18) The Fibre Channel receiver jitter tolerance numbers are compliant to the specification at β_R interoperability point.

Table 4–20 provides information on recommended input clock jitter for each mode.

Table 4–20. Recommended Input Clock Jitter (Part 1 of 2)						
Mode	Reference Clock (MHz)	Vectron LVPECL XO Type/Model	Frequency Range (MHz)	RMS Jitter (12 kHz to 20 MHz) (ps)	Period Jitter (Peak to Peak) (ps)	Phase Noise at 1 MHz (dB c/Hz)
PCI-E	100	VCC6-Q/R	10 to 270	0.3	23	-149.9957
(OIF) CEI PHY	156.25	VCC6-Q/R	10 to 270	0.3	23	-146.2169
	622.08	VCC6-Q	270 to 800	2	30	Not available
GIGE	62.5	VCC6-Q/R	10 to 270	0.3	23	-149.9957
	125	VCC6-Q/R	10 to 270	0.3	23	-146.9957
XAUI	156.25	VCC6-Q/R	10 to 270	0.3	23	-146.2169

Table 4–20. Recommended Input Clock Jitter (Part 2 of 2)

Mode	Reference Clock (MHz)	Vectron LVPECL XO Type/Model	Frequency Range (MHz)	RMS Jitter (12 kHz to 20 MHz) (ps)	Period Jitter (Peak to Peak) (ps)	Phase Noise at 1 MHz (dB c/Hz)
SONET/SDH OC-48	77.76	VCC6-Q/R	10 to 270	0.3	23	-149.5476
	155.52	VCC6-Q/R	10 to 270	0.3	23	-149.1903
	311.04	VCC6-Q	270 to 800	2	30	Not available
	622.08	VCC6-Q	270 to 800	2	30	Not available
SONET/SDH OC-12	62.2	VCC6-Q/R	10 to 270	0.3	23	-149.6289
	311	VCC6-Q	270 to 800	2	30	Not available
	77.76	VCC6-Q/R	10 to 270	0.3	23	-149.5476
	155.52	VCC6-Q/R	10 to 270	0.3	23	-149.1903
	622.08	VCC6-Q	270 to 800	2	30	Not available

Tables 4–21 and 4–22 show the transmitter and receiver PCS latency for each mode, respectively.

Table 4–21. PCS Latency (Part 1 of 2) *Note (1)*

Functional Mode	Configuration	Transmitter PCS Latency					
		TX PIPE	TX Phase Comp FIFO	Byte Serializer	TX State Machine	8B/10B Encoder	Sum *(2)*
XAUI		-	2-3	1	0.5	0.5	4-5
PIPE	×1, ×4, ×8 8-bit channel width	1	3-4	1	-	1	6-7
	×1, ×4, ×8 16-bit channel width	1	3-4	1	-	0.5	6-7
GIGE		-	2-3	1	-	1	4-5
SONET/SDH	OC-12	-	2-3	1	-	1	4-5
	OC-48	-	2-3	1	-	0.5	4-5
	OC-96	-	2-3	1	-	0.5	4-5
(OIF) CEI PHY		-	2-3	1	-	0.5	4-5
CPRI *(3)*	614 Mbps, 1.228 Gbps	-	2	1	-	1	4
	2.456 Gbps	-	2-3	1	-	1	4-5

Table 4–21. PCS Latency (Part 2 of 2) Note (1)

Functional Mode	Configuration	Transmitter PCS Latency					
		TX PIPE	TX Phase Comp FIFO	Byte Serializer	TX State Machine	8B/10B Encoder	Sum (2)
Serial RapidIO	1.25 Gbps, 2.5 Gbps, 3.125 Gbps	-	2-3	1	-	0.5	4-5
SDI	HD 10-bit channel width	-	2-3	1	-	1	4-5
	HD, 3G 20-bit channel width	-	2-3	1	-	0.5	4-5
BASIC Single Width	8-bit/10-bit channel width	-	2-3	1	-	1	4-5
	16-bit/20-bit channel width	-	2-3	1	-	0.5	4-5
BASIC Double Width	16-bit/20-bit channel width	-	2-3	1	-	1	4-5
	32-bit/40-bit channel width	-	2-3	1	-	0.5	4-5
	Parallel Loopback/ BIST	-	2-3	1	-	1	4-5

Notes to Table 4–21:

(1) The latency numbers are with respect to the PLD-transceiver interface clock cycles.

(2) The total latency number is rounded off in the Sum column.

(3) For CPRI 614 Mbps and 1.228 Gbps data rates, the Quartus II software customizes the PLD-transceiver interface clocking to achieve zero clock cycle uncertainty in the transmitter phase compensation FIFO latency. For more details, refer to the *CPRI Mode* section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–22. PCS Latency (Part 1 of 3) *Note (1)*

Functional Mode	Configuration	Receiver PCS Latency									
		Word Aligner	Deskew FIFO	Rate Matcher *(3)*	8B/10B Decoder	Receiver State Machine	Byte De-serializer	Byte Order	Receiver Phase Comp FIFO	Receiver PIPE	Sum *(2)*
XAUI		2-2.5	2-2.5	5.5-6.5	0.5	1	1	1	1-2	-	14-17
PIPE	×1, ×4, ×8 8-bit channel width	4-5	-	11-13	1	-	1	1	2-3	1	21-25
	×1, ×4, ×8 16-bit channel width	2-2.5	-	5.5-6.5	0.5	-	1	1	2-3	1	13-16
GIGE		4-5	-	11-13	1	-	1	1	1-2	-	19-23
SONET/ SDH	OC-12	6-7	-	-	1	-	1	1	1-2	-	10-12
	OC-48	3-3.5	-	-	0.5	-	1	1-2	1-2	-	7-9
	OC-96	2-2.5	-	-	0.5	-	1	1	1-2	-	6-7
(OIF) CEI PHY		2.5	-	-	0.5	-	1	1	1-2	-	6-7
CPRI *(4)*	614 Mbps, 1.228 Gbps	4-5	-	-	1	-	1	1	1	-	8-9
	2.456 Gbps	4-5	-	-	1	-	1	1	1-2	-	8-10
Serial RapidIO	1.25 Gbps, 2.5 Gbps, 3.125 Gbps	2-2.5	-	-	0.5	-	1	1	1-2	-	6-7
SDI	HD 10-bit channel width	5	-	-	1	-	1	1	1-2	-	9-10
	HD, 3G 20-bit channel width	2.5	-	-	0.5	-	1	1	1-2	-	6-7

Table 4–22. PCS Latency (Part 2 of 3) Note (1)

Functional Mode	Configuration	Receiver PCS Latency									
		Word Aligner	Deskew FIFO	Rate Matcher (3)	8B/10B Decoder	Receiver State Machine	Byte De-serializer	Byte Order	Receiver Phase Comp FIFO	Receiver PIPE	Sum (2)
BASIC Single Width	8/10-bit channel width; with Rate Matcher	4-5	-	11-13	1	-	1	1	1-2	1	19-23
	8/10-bit channel width; without Rate Matcher	4-5	-	-	1	-	1	1	1-2	-	8-10
	16/20-bit channel width; with Rate Matcher	2-2.5	-	5.5-6.5	0.5	-	1	1	1-2	-	11-14
	16/20-bit channel width; without Rate Matcher	2-2.5	-	-	0.5	-	1	1	1-2	-	6-7

Table 4–22. PCS Latency (Part 3 of 3) Note (1)

Functional Mode	Configuration	Receiver PCS Latency									
		Word Aligner	Deskew FIFO	Rate Matcher (3)	8B/10B Decoder	Receiver State Machine	Byte De-serializer	Byte Order	Receiver Phase Comp FIFO	Receiver PIPE	Sum (2)
BASIC Double Width	16/20-bit channel width; with Rate Matcher	4-5	-	11-13	1	-	1	1	1-2	-	19-23
	16/20-bit channel width; without Rate Matcher	4-5	-	-	1	-	1	1	1-2	-	8-10
	32/40-bit channel width; with Rate Matcher	2-2.5	-	5.5-6.5	0.5	-	1	1	1-2	-	11-14
	32/40-bit channel width; without Rate Matcher	2-2.5	-	-	0.5	-	1	1-3	1-2	-	6-9

Notes to Table 4–21:

(1) The latency numbers are with respect to the PLD-transceiver interface clock cycles.

(2) The total latency number is rounded off in the Sum column.

(3) The rate matcher latency shown is the steady state latency. Actual latency may vary depending on the skip ordered set gap allowed by the protocol, actual PPM difference between the reference clocks, and so forth.

(4) For CPRI 614 Mbps and 1.228 Gbps data rates, the Quartus II software customizes the PLD-transceiver interface clocking to achieve zero clock cycle uncertainty in the receiver phase compensation FIFO latency. For more details, refer to the *CPRI Mode* section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*

DC Electrical Characteristics

Table 4–23 shows the Stratix II GX device family DC electrical characteristics.

Table 4–23. Stratix II GX Device DC Operating Conditions (Part 1 of 2) *Note (1)*

Symbol	Parameter	Conditions	Device	Minimum	Typical	Maximum	Unit
I_I	Input pin leakage current	$V_I = V_{CCIOmax}$ to 0 V *(2)*	All	–10		10	µA
I_{OZ}	Tri-stated I/O pin leakage current	$V_O = V_{CCIOmax}$ to 0 V *(2)*	All	–10		10	µA
I_{CCINT0}	V_{CCINT} supply current (standby)	V_I = ground, no load, no toggling inputs T_J = 25 °C	EP2SGX30		0.30	*(3)*	A
			EP2SGX60		0.50	*(3)*	A
			EP2SGX90		0.62	*(3)*	A
			EP2SGX130		0.82	*(3)*	A
I_{CCPD0}	V_{CCPD} supply current (standby)	V_I = ground, no load, no toggling inputs T_J = 25 °C, V_{CCPD} = 3.3V	EP2SGX30		2.7	*(3)*	mA
			EP2SGX60		3.6	*(3)*	mA
			EP2SGX90		4.3	*(3)*	mA
			EP2SGX130		5.4	*(3)*	mA
I_{CCIO0}	V_{CCIO} supply current (standby)	V_I = ground, no load, no toggling inputs T_J = 25 °C	EP2SGX30		4.0	*(3)*	mA
			EP2SGX60		4.0	*(3)*	mA
			EP2SGX90		4.0	*(3)*	mA
			EP2SGX130		4.0	*(3)*	mA

Table 4–23. Stratix II GX Device DC Operating Conditions (Part 2 of 2) *Note (1)*

Symbol	Parameter	Conditions	Device	Minimum	Typical	Maximum	Unit
R_{CONF} *(4)*	Value of I/O pin pull-up resistor before and during configuration	Vi = 0, V_{CCIO} = 3.3 V		10	25	50	KOhm
		Vi = 0, V_{CCIO} = 2.5 V		15	35	70	KOhm
		Vi = 0, V_{CCIO} = 1.8 V		30	50	100	KOhm
		Vi = 0, V_{CCIO} = 1.5 V		40	75	150	KOhm
		Vi = 0, V_{CCIO} = 1.2 V		50	90	170	KOhm
	Recommended value of I/O pin external pull-down resistor before and during configuration				1	2	KOhm

Notes to Table 4–23:

(1) Typical values are for T_A = 25 °C, V_{CCINT} = 1.2 V, and V_{CCIO} = 1.5 V, 1.8 V, 2.5 V, and 3.3 V.

(2) This value is specified for normal device operation. The value may vary during power-up. This applies for all V_{CCIO} settings (3.3, 2.5, 1.8, and 1.5 V).

(3) Maximum values depend on the actual TJ and design utilization. See *PowerPlay Early Power Estimator (EPE) and Power Analyzer* or the *Quartus II PowerPlay Power Analyzer and Optimization Technology* (available at **www.altera.com**) for maximum values. See the section "Power Consumption" on page 4–59 for more information.

(4) Pin pull-up resistance values will lower if an external source drives the pin higher than V_{CCIO}.

I/O Standard Specifications

Tables 4–24 through 4–47 show the Stratix II GX device family I/O standard specifications.

Table 4–24. LVTTL Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output supply voltage		3.135	3.465	V
V_{IH}	High-level input voltage		1.7	4.0	V
V_{IL}	Low-level input voltage		–0.3	0.8	V
V_{OH}	High-level output voltage	I_{OH} = –4 mA *(2)*	2.4		V

Table 4–24. LVTTL Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{OL}	Low-level output voltage	I_{OL} = 4 mA *(2)*		0.45	V

Notes to Table 4–24:
(1) Stratix II GX devices comply to the narrow range for the supply voltage as specified in the EIA/JEDEC Standard, JESD8-B.
(2) This specification is supported across all the programmable drive strength settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–25. LVCMOS Specifications Note (1)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output supply voltage		3.135	3.465	V
V_{IH}	High-level input voltage		1.7	4.0	V
V_{IL}	Low-level input voltage		–0.3	0.8	V
V_{OH}	High-level output voltage	V_{CCIO} = 3.0, I_{OH} = –0.1 mA *(2)*	V_{CCIO} – 0.2		V
V_{OL}	Low-level output voltage	V_{CCIO} = 3.0, I_{OL} = 0.1 mA *(2)*		0.2	V

Notes to Table 4–25:
(1) Stratix II GX devices comply to the narrow range for the supply voltage as specified in the EIA/JEDEC Standard, JESD8-B.
(2) This specification is supported across all the programmable drive strength available for this I/O standard as shown in *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–26. 2.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output supply voltage		2.375	2.625	V
V_{IH}	High-level input voltage		1.7	4.0	V
V_{IL}	Low-level input voltage		–0.3	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –1 mA *(2)*	2.0		V
V_{OL}	Low-level output voltage	I_{OL} = 1 mA *(2)*		0.4	V

Notes to Table 4–26:
(1) The Stratix II GX device V_{CCIO} voltage level support of 2.5 to 5% is narrower than defined in the Normal Range of the EIA/JEDEC standard.
(2) This specification is supported across all the programmable drive settings available for this I/O standard as shown in *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–27. 1.8-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output supply voltage		1.71	1.89	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	2.25	V
V_{IL}	Low-level input voltage		–0.3	0.35 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 mA *(2)*	V_{CCIO} – 0.45		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(2)*		0.45	V

Notes to Table 4–27:

(1) The Stratix II GX device V_{CCIO} voltage level support of 1.8 to 5% is narrower than defined in the Normal Range of the EIA/JEDEC standard.

(2) This specification is supported across all the programmable drive settings available for this I/O standard as shown in *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–28. 1.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output supply voltage		1.425	1.575	V
V_{IH}	High-level input voltage		0.65 V_{CCIO}	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3	0.35 V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 mA *(2)*	0.75 V_{CCIO}		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(2)*		0.25 V_{CCIO}	V

Notes to Table 4–28:

(1) The Stratix II GX device V_{CCIO} voltage level support of 1.5 to 5% is narrower than defined in the Normal Range of the EIA/JEDEC standard.

(2) This specification is supported across all the programmable drive settings available for this I/O standard as shown in *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Figures 4–6 and 4–7 show receiver input and transmitter output waveforms, respectively, for all differential I/O standards (LVDS and LVPECL).

Figure 4–6. Receiver Input Waveforms for Differential I/O Standards

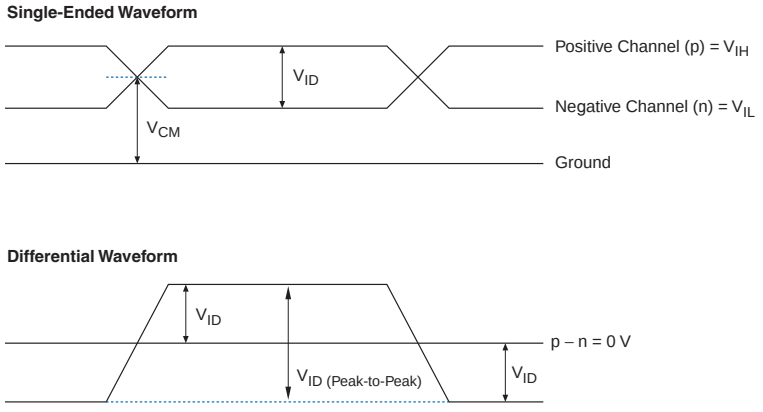


Figure 4–7. Transmitter Output Waveforms for Differential I/O Standards

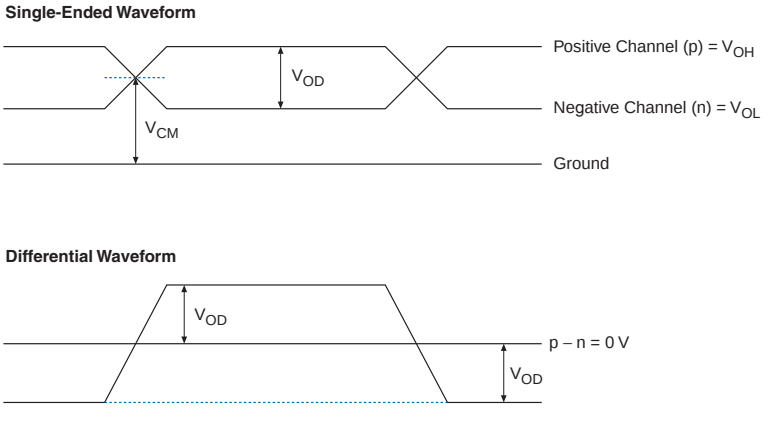


Table 4–29. 2.5-V LVDS I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage for left and right I/O banks (1, 2, 5, and 6)		2.375	2.5	2.625	V
V_{ID}	Input differential voltage swing (single-ended)		100	350	900	mV
V_{ICM}	Input common mode voltage		200	1,250	1,800	mV
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	250		450	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	1.125		1.375	V
R_L	Receiver differential input discrete resistor (external to Stratix II GX devices)		90	100	110	Ω

Table 4–30. 3.3-V LVDS I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO} *(1)*	I/O supply voltage for top and bottom PLL banks (9, 10, 11, and 12)		3.135	3.3	3.465	V
V_{ID}	Input differential voltage swing (single-ended)		100	350	900	mV
V_{ICM}	Input common mode voltage		200	1,250	1,800	mV
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	250		710	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	840		1,570	mV
R_L	Receiver differential input discrete resistor (external to Stratix II GX devices)		90	100	110	Ω

Note to Table 4–30:

(1) The top and bottom clock input differential buffers in I/O banks 3, 4, 7, and 8 are powered by V_{CCINT}, not V_{CCIO}. The PLL clock output/feedback differential buffers are powered by VCC_PLL_OUT. For differential clock output/feedback operation, connect VCC_PLL_OUT to 3.3 V.

Table 4–31. PCML Specifications *Note (1)*

Symbol	Parameter	References
Reference Clock		
3.3-V PCML 1.5-V PCML 1.2-V PCML	Reference clock supported PCML standards	
V_{ID}	Peak-to-peak differential input voltage	The specifications are located in the Reference Clock section of Table 4–6 on page 4–4. The specifications listed in Table 4–6 are applicable to PCML input standards.
V_{ICM}	Input common mode voltage	
R	On-chip termination resistors	
Receiver		
3.3-V PCML 1.5-V PCML 1.2-V PCML	Receiver supported PCML standards	
V_{ID}	Peak-to-peak differential input voltage	The specifications are located in the Receiver section of Table 4–6 on page 4–4. The specifications listed in Table 4–6 are applicable to PCML input standards.
V_{ICM}	Input common mode voltage	
R	On-chip termination resistors	
Transmitter		
1.5-V PCML 1.2-V PCML	Transmitter supported PCML standards	
V_{CCH}	Output buffer supply voltage	The specifications are located in Table 4–5 on page 4–4.
V_{OD}	Peak-to-peak differential output voltage	The specifications are located in Tables 4–7, 4–8, 4–9, 4–10, 4–11, and 4–12. The specifications listed in these tables are applicable to PCML output standards.
V_{OCM}	Output common mode voltage	The specifications are located in the Transmitter section of Table 4–6 on page 4–4. The specifications listed in Table 4–6 are applicable to PCML output standards.
R	On-chip termination resistors	

Note to Table 4–31:

(1) Stratix II GX devices support PCML input and output on GXB banks 13, 14, 15, 16, and 17. This table references Stratix II GX PCML specifications that are located in other sections of the *Stratix II GX Device Handbook*.

Table 4–32. LVPECL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO} *(1)*	I/O supply voltage		3.135	3.3	3.465	V
V_{ID}	Input differential voltage swing (single-ended)		300	600	1,000	mV
V_{ICM}	Input common mode voltage		1.0		2.5	V
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	525		970	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	1,650		2,250	mV
R_L	Receiver differential input resistor		90	100	110	Ω

Note to Table 4–32:

(1) The top and bottom clock input differential buffers in I/O banks 3, 4, 7, and 8 are powered by V_{CCINT}, not V_{CCIO}. The PLL clock output/feedback differential buffers are powered by VCC_PLL_OUT. For differential clock output/feedback operation, connect VCC_PLL_OUT to 3.3 V.

Table 4–33. 3.3-V PCI Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.0	3.3	3.6	V
V_{IH}	High-level input voltage		0.5 V_{CCIO}		V_{CCIO} + 0.5	V
V_{IL}	Low-level input voltage		–0.3		0.3 V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OUT} = –500 µA	0.9 V_{CCIO}			V
V_{OL}	Low-level output voltage	I_{OUT} = 1,500 µA			0.1 V_{CCIO}	V

Table 4–34. PCI-X Mode 1 Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.0		3.6	V
V_{IH}	High-level input voltage		0.5 V_{CCIO}		V_{CCIO} + 0.5	V
V_{IL}	Low-level input voltage		–0.3		0.35 V_{CCIO}	V
V_{IPU}	Input pull-up voltage		0.7 V_{CCIO}			V
V_{OH}	High-level output voltage	I_{OUT} = –500 µA	0.9 V_{CCIO}			V
V_{OL}	Low-level output voltage	I_{OUT} = 1,500 µA			0.1 V_{CCIO}	V

Table 4–35. SSTL-18 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.71	1.8	1.89	V
V_{REF}	Reference voltage		0.855	0.9	0.945	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{IH} (DC)	High-level DC input voltage		V_{REF} + 0.125			V
V_{IL} (DC)	Low-level DC input voltage				V_{REF} – 0.125	V
V_{IH} (AC)	High-level AC input voltage		V_{REF} + 0.25			V
V_{IL} (AC)	Low-level AC input voltage				V_{REF} – 0.25	V
V_{OH}	High-level output voltage	I_{OH} = –6.7 mA *(1)*	V_{TT} + 0.475			V
V_{OL}	Low-level output voltage	I_{OL} = 6.7 mA *(1)*			V_{TT} – 0.475	V

Note to Table 4–35:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–36. SSTL-18 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.71	1.8	1.89	V
V_{REF}	Reference voltage		0.855	0.9	0.945	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{IH} (DC)	High-level DC input voltage		V_{REF} + 0.125			V
V_{IL} (DC)	Low-level DC input voltage				V_{REF} – 0.125	V
V_{IH} (AC)	High-level AC input voltage		V_{REF} + 0.25			V
V_{IL} (AC)	Low-level AC input voltage				V_{REF} – 0.25	V
V_{OH}	High-level output voltage	I_{OH} = –13.4 mA *(1)*	V_{CCIO} – 0.28			V
V_{OL}	Low-level output voltage	I_{OL} = 13.4 mA *(1)*			0.28	V

Note to Table 4–36:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–37. SSTL-18 Class I and II Differential Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.71	1.8	1.89	V
V_{SWING} (DC)	DC differential input voltage		0.25			V
V_X (AC)	AC differential input cross point voltage		(V_{CCIO}/2) – 0.175		(V_{CCIO}/2) + 0.175	V
V_{SWING} (AC)	AC differential input voltage		0.5			V
V_{ISO}	Input clock signal offset voltage			0.5 V_{CCIO}		V
ΔV_{ISO}	Input clock signal offset voltage variation			200		mV
V_{OX} (AC)	AC differential cross point voltage		(V_{CCIO}/2) – 0.125		(V_{CCIO}/2) + 0.125	V

Table 4–38. SSTL-2 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		2.375	2.5	2.625	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage		1.188	1.25	1.313	V
V_{IH} (DC)	High-level DC input voltage		V_{REF} + 0.18		3.0	V
V_{IL} (DC)	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
V_{IH} (AC)	High-level AC input voltage		V_{REF} + 0.35			V
V_{IL} (AC)	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –8.1 mA *(1)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(1)*			V_{TT} – 0.57	V

Note to Table 4–38:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–39. SSTL-2 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		2.375	2.5	2.625	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage		1.188	1.25	1.313	V

Table 4–39. SSTL-2 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IH} (DC)	High-level DC input voltage		V_{REF} + 0.18		V_{CCIO} + 0.3	V
V_{IL} (DC)	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
V_{IH} (AC)	High-level AC input voltage		V_{REF} + 0.35			V
V_{IL} (AC)	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –16.4 mA *(1)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(1)*			V_{TT} – 0.76	V

Note to Table 4–39:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–40. SSTL-2 Class I and II Differential Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		2.375	2.5	2.625	V
V_{SWING} (DC)	DC differential input voltage		0.36			V
V_X (AC)	AC differential input cross point voltage		(V_{CCIO}/2) – 0.2		(V_{CCIO}/2) + 0.2	V
V_{SWING} (AC)	AC differential input voltage		0.7			V
V_{ISO}	Input clock signal offset voltage			0.5 V_{CCIO}		V
ΔV_{ISO}	Input clock signal offset voltage variation			200		mV
V_{OX} (AC)	AC differential output cross point voltage		(V_{CCIO}/2) – 0.2		(V_{CCIO}/2) + 0.2	V

Table 4–41. 1.2-V HSTL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.14	1.2	1.26	V
V_{REF}	Reference voltage		0.48 V_{CCIO}	0.5 V_{CCIO}	0.52 V_{CCIO}	V
V_{IH} (DC)	High-level DC input voltage		V_{REF} + 0.08		V_{CCIO} + 0.15	V
V_{IL} (DC)	Low-level DC input voltage		–0.15		V_{REF} – 0.08	V
V_{IH} (AC)	High-level AC input voltage		V_{REF} + 0.15		V_{CCIO} + 0.24	V
V_{IL} (AC)	Low-level AC input voltage		–0.24		V_{REF} – 0.15	V

Table 4–41. 1.2-V HSTL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{OH}	High-level output voltage	I_{OH} = 8 mA	V_{REF} + 0.15		V_{CCIO} + 0.15	V
V_{OL}	Low-level output voltage	I_{OH} = –8 mA	–0.15		V_{REF} – 0.15	V

Table 4–42. 1.5-V HSTL Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.425	1.5	1.575	V
V_{REF}	Input reference voltage		0.713	0.75	0.788	V
V_{TT}	Termination voltage		0.713	0.75	0.788	V
V_{IH} (DC)	DC high-level input voltage		V_{REF} + 0.1			V
V_{IL} (DC)	DC low-level input voltage		–0.3		V_{REF} – 0.1	V
V_{IH} (AC)	AC high-level input voltage		V_{REF} + 0.2			V
V_{IL} (AC)	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 8 mA *(1)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –8 mA *(1)*			0.4	V

Note to Table 4–42:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–43. 1.5-V HSTL Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.425	1.50	1.575	V
V_{REF}	Input reference voltage		0.713	0.75	0.788	V
V_{TT}	Termination voltage		0.713	0.75	0.788	V
V_{IH} (DC)	DC high-level input voltage		V_{REF} + 0.1			V
V_{IL} (DC)	DC low-level input voltage		–0.3		V_{REF} – 0.1	V
V_{IH} (AC)	AC high-level input voltage		V_{REF} + 0.2			V
V_{IL} (AC)	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 16 mA *(1)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(1)*			0.4	V

Note to Table 4–43:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–44. 1.5-V HSTL Class I and II Differential Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		1.425	1.5	1.575	V
V_{DIF} (DC)	DC input differential voltage		0.2			V
V_{CM} (DC)	DC common mode input voltage		0.68		0.9	V
V_{DIF} (AC)	AC differential input voltage		0.4			V
V_{OX} (AC)	AC differential cross point voltage		0.68		0.9	V

Table 4–45. 1.8-V HSTL Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.71	1.80	1.89	V
V_{REF}	Input reference voltage		0.85	0.90	0.95	V
V_{TT}	Termination voltage		0.85	0.90	0.95	V
V_{IH} (DC)	DC high-level input voltage		V_{REF} + 0.1			V
V_{IL} (DC)	DC low-level input voltage		–0.3		V_{REF} – 0.1	V
V_{IH} (AC)	AC high-level input voltage		V_{REF} + 0.2			V
V_{IL} (AC)	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 8 mA *(1)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –8 mA *(1)*			0.4	V

Note to Table 4–45:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–46. 1.8-V HSTL Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.71	1.80	1.89	V
V_{REF}	Input reference voltage		0.85	0.90	0.95	V
V_{TT}	Termination voltage		0.85	0.90	0.95	V
V_{IH} (DC)	DC high-level input voltage		V_{REF} + 0.1			V
V_{IL} (DC)	DC low-level input voltage		–0.3		V_{REF} – 0.1	V
V_{IH} (AC)	AC high-level input voltage		V_{REF} + 0.2			V
V_{IL} (AC)	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 16 mA *(1)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(1)*			0.4	V

Note to Table 4–46:

(1) This specification is supported across all the programmable drive settings available for this I/O standard as shown in the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Table 4–47. 1.8-V HSTL Class I and II Differential Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		1.71	1.80	1.89	V
V_{DIF} (DC)	DC input differential voltage		0.2			V
V_{CM} (DC)	DC common mode input voltage		0.78		1.12	V
V_{DIF} (AC)	AC differential input voltage		0.4			V
V_{OX} (AC)	AC differential cross point voltage		0.68		0.9	V

Bus Hold Specifications

Table 4–48 shows the Stratix II GX device family bus hold specifications.

Table 4–48. Bus Hold Parameters

Parameter	Conditions	V_{CCIO} Level										Unit
		1.2 V		1.5 V		1.8 V		2.5 V		3.3 V		
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
Low sustaining current	$V_{IN} > V_{IL}$ (maximum)	22.5		25		30		50		70		µA
High sustaining current	$V_{IN} < V_{IH}$ (minimum)	–22.5		–25		–30		–50		–70		µA
Low overdrive current	$0 V < V_{IN} < V_{CCIO}$		120		160		200		300		500	µA
High overdrive current	$0 V < V_{IN} < V_{CCIO}$		–120		–160		–200		–300		–500	µA
Bus-hold trip point		0.45	0.95	0.5	1.0	0.68	1.07	0.7	1.7	0.8	2.0	V

On-Chip Termination Specifications

Tables 4–49 and 4–50 define the specification for internal termination resistance tolerance when using series or differential on-chip termination.

Table 4–49. On-Chip Termination Specification for Top and Bottom I/O Banks (Part 1 of 2) *Notes (1), (2)*

Symbol	Description	Conditions	Resistance Tolerance		
			Commercial Max	Industrial Max	Unit
25-Ω R_S 3.3/2.5	Internal series termination with calibration (25-Ω setting)	V_{CCIO} = 3.3/2.5 V	±5	±10	%
	Internal series termination without calibration (25-Ω setting)	V_{CCIO} = 3.3/2.5 V	±30	±30	%
50-Ω R_S 3.3/2.5	Internal series termination with calibration (50-Ω setting)	V_{CCIO} = 3.3/2.5 V	±5	±10	%
	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 3.3/2.5 V	±30	± 30	%

Table 4–49. On-Chip Termination Specification for Top and Bottom I/O Banks (Part 2 of 2) *Notes (1), (2)*

Symbol	Description	Conditions	Resistance Tolerance		
			Commercial Max	Industrial Max	Unit
50-Ω R_T 2.5	Internal parallel termination with calibration (50-Ω setting)	V_{CCIO} = 1.8 V	±30	± 30	%
25-Ω R_S 1.8	Internal series termination with calibration (25-Ω setting)	V_{CCIO} = 1.8 V	±5	±10	%
	Internal series termination without calibration (25-Ω setting)	V_{CCIO} = 1.8 V	±30	±30	%
50-Ω R_S 1.8	Internal series termination with calibration (50-Ω setting)	V_{CCIO} = 1.8 V	±5	±10	%
	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 1.8 V	±30	±30	%
50-Ω R_T 1.8	Internal parallel termination with calibration (50-Ω setting)	V_{CCIO} = 1.8 V	±10	±15	%
50-Ω R_S 1.5	Internal series termination with calibration (50-Ω setting)	V_{CCIO} = 1.5 V	±8	±10	%
	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 1.5 V	±36	±36	%
50-Ω R_T 1.5	Internal parallel termination with calibration (50-Ω setting)	V_{CCIO} = 1.5 V	±10	±15	%
50-Ω R_S 1.2	Internal series termination with calibration (50-Ω setting)	V_{CCIO} = 1.2 V	±8	±10	%
	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 1.2 V	±50	±50	%
50-Ω R_T 1.2	Internal parallel termination with calibration (50-Ω setting)	V_{CCIO} = 1.2 V	±10	±15	%

Note for Table 4–49:

(1) The resistance tolerance for calibrated SOCT is for the moment of calibration. If the temperature or voltage changes over time, the tolerance may also change.

(2) On-chip parallel termination with calibration is only supported for input pins.

Table 4–50. Series and Differential On-Chip Termination Specification for Left I/O Banks Note (1)					
Symbol	Description	Conditions	Resistance Tolerance		
			Commercial Max	Industrial Max	Unit
25-Ω R_S 3.3/2.5	Internal series termination without calibration (25-Ω setting)	V_{CCIO} = 3.3/2.5V	±30	±30	%
50-Ω R_S 3.3/2.5/1.8	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 3.3/2.5/1.8V	±30	±30	%
50-Ω R_S 1.5	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 1.5V	±36	±36	%
R_D	Internal differential termination for LVDS (100-Ω setting)	V_{CCIO} = 2.5 V	±20	±25	%

Note to Table 4–50:
(1) On-chip parallel termination with calibration is only supported for input pins.

Pin Capacitance

Table 4–51 shows the Stratix II GX device family pin capacitance.

Table 4–51. Stratix II GX Device Capacitance Note (1)			
Symbol	Parameter	Typical	Unit
C_{IOTB}	Input capacitance on I/O pins in I/O banks 3, 4, 7, and 8.	5.0	pF
C_{IOL}	Input capacitance on I/O pins in I/O banks 1 and 2, including high-speed differential receiver and transmitter pins.	6.1	pF
C_{CLKTB}	Input capacitance on top/bottom clock input pins: `CLK[4..7]` and `CLK[12..15]`.	6.0	pF
C_{CLKL}	Input capacitance on left clock inputs: `CLK0` and `CLK2`.	6.1	pF
C_{CLKL+}	Input capacitance on left clock inputs: `CLK1` and `CLK3`.	3.3	pF
C_{OUTFB}	Input capacitance on dual-purpose clock output/feedback pins in PLL banks 11 and 12.	6.7	pF

Note to Table 4–51:
(1) Capacitance is sample-tested only. Capacitance is measured using time-domain reflections (TDR). Measurement accuracy is within ±0.5 pF.

Power Consumption

Altera offers two ways to calculate power for a design: the Excel-based PowerPlay early power estimator power calculator and the Quartus® II PowerPlay power analyzer feature.

The interactive Excel-based PowerPlay early power estimator is typically used prior to designing the FPGA in order to get an estimate of device power. The Quartus II PowerPlay power analyzer provides better quality estimates based on the specifics of the design after place-and-route is complete. The power analyzer can apply a combination of user-entered, simulation-derived and estimated signal activities which, combined with detailed circuit models, can yield very accurate power estimates.

In both cases, these calculations should only be used as an estimation of power, not as a specification.

For more information on PowerPlay tools, refer to the *PowerPlay Early Power Estimators (EPE) and Power Analyzer,* the *Quartus II PowerPlay Analysis and Optimization Technology*, and the *PowerPlay Power Analyzer* chapter in volume 3 of the *Quartus II Handbook*. The PowerPlay early power estimators are available on the Altera web site at **www.altera. com**.

See Table 4–23 on page 42 for typical I_{CC} standby specifications.

Timing Model

The DirectDrive technology and MultiTrack interconnect ensure predictable performance, accurate simulation, and accurate timing analysis across all Stratix II GX device densities and speed grades. This section describes and specifies the performance, internal, external, and PLL timing specifications.

All specifications are representative of worst-case supply voltage and junction temperature conditions.

Preliminary and Final Timing

Timing models can have either preliminary or final status. The Quartus II software issues an informational message during the design compilation if the timing models are preliminary. Table 4–52 shows the status of the Stratix II GX device timing models.

Preliminary status means the timing model is subject to change. Initially, timing numbers are created using simulation results, process data, and other known parameters. These tests are used to make the preliminary numbers as close to the actual timing parameters as possible.

Final timing numbers are based on actual device operation and testing. These numbers reflect the actual performance of the device under worst-case voltage and junction temperature conditions.

Table 4–52. Stratix II GX Device Timing Model Status

Device	Preliminary	Final
EP2SGX30		✓
EP2SGX60		✓
EP2SGX90		✓
EP2SGX130		✓

I/O Timing Measurement Methodology

Different I/O standards require different baseline loading techniques for reporting timing delays. Altera characterizes timing delays with the required termination for each I/O standard and with 0 pF (except for PCI and PCI-X which use 10 pF) loading and the timing is specified up to the output pin of the FPGA device. The Quartus II software calculates the I/O timing for each I/O standard with a default baseline loading as specified by the I/O standards.

The following measurements are made during device characterization. Altera measures clock-to-output delays (t_{CO}) at worst-case process, minimum voltage, and maximum temperature (PVT) for default loading conditions shown in Table 4–53. Use the following equations to calculate clock pin to output pin timing for Stratix II GX devices.

t_{CO} from clock pin to I/O pin = delay from clock pad to I/O output register + IOE output register clock-to-output delay + delay from output register to output pin + I/O output delay

t_{xz}/t_{zx} from clock pin to I/O pin = delay from clock pad to I/O output register + IOE output register clock-to-output delay + delay from output register to output pin + I/O output delay + output enable pin delay

Simulation using IBIS models is required to determine the delays on the PCB traces in addition to the output pin delay timing reported by the Quartus II software and the timing model in the device handbook.

1. Simulate the output driver of choice into the generalized test setup, using values from Table 4–53.
2. Record the time to V_{MEAS}.

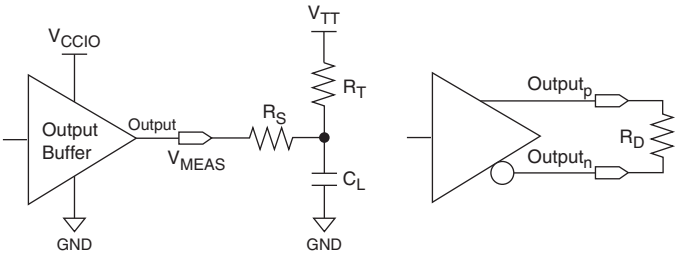
3. Simulate the output driver of choice into the actual PCB trace and load, using the appropriate IBIS model or capacitance value to represent the load.

4. Record the time to V_{MEAS}.

5. Compare the results of steps 2 and 4. The increase or decrease in delay should be added to or subtracted from the I/O Standard Output Adder delays to yield the actual worst-case propagation delay (clock-to-output) of the PCB trace.

The Quartus II software reports the timing with the conditions shown in Table 4–53 using the above equation. Figure 4–8 shows the model of the circuit that is represented by the output timing of the Quartus II software.

Figure 4–8. Output Delay Timing Reporting Setup Modeled by Quartus II



Notes to Figure 4–8:

(1) Output pin timing is reported at the output pin of the FPGA device. Additional delays for loading and board trace delay need to be accounted for with IBIS model simulations.
(2) V_{CCPD} is 3.085 V unless otherwise specified.
(3) V_{CCINT} is 1.12 V unless otherwise specified.

Table 4–53. Output Timing Measurement Methodology for Output Pins (Part 1 of 2) *Notes (1), (2), (3)*

I/O Standard	Loading and Termination						Measurement Point
	R_S (Ω)	R_D (Ω)	R_T (Ω)	V_{CCIO} (V)	V_{TT} (V)	C_L (pF)	V_{MEAS} (V)
LVTTL *(4)*				3.135		0	1.5675
LVCMOS *(4)*				3.135		0	1.5675
2.5 V *(4)*				2.375		0	1.1875
1.8 V *(4)*				1.710		0	0.855
1.5 V *(4)*				1.425		0	0.7125

Table 4–53. Output Timing Measurement Methodology for Output Pins (Part 2 of 2) Notes (1), (2), (3)

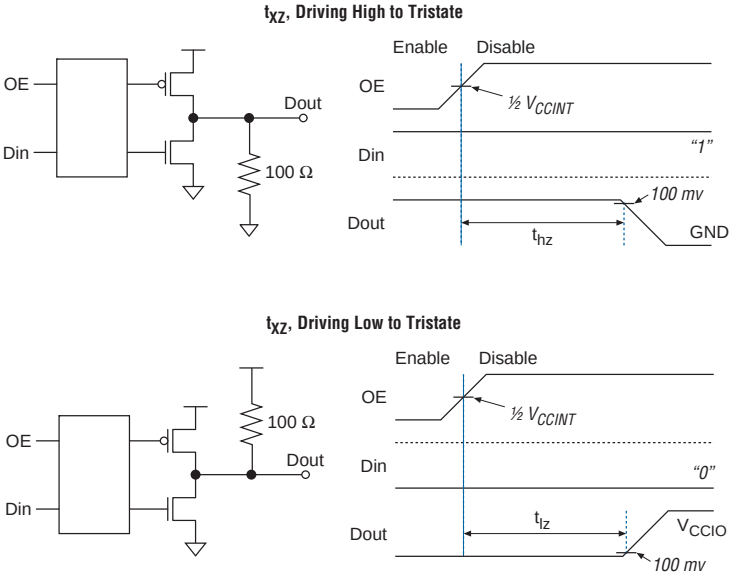
I/O Standard	Loading and Termination						Measurement Point
	R_S (Ω)	R_D (Ω)	R_T (Ω)	V_{CCIO} (V)	V_{TT} (V)	C_L (pF)	V_{MEAS} (V)
PCI (5)				2.970		10	1.485
PCI-X (5)				2.970		10	1.485
SSTL-2 Class I	25		50	2.325	1.123	0	1.1625
SSTL-2 Class II	25		25	2.325	1.123	0	1.1625
SSTL-18 Class I	25		50	1.660	0.790	0	0.83
SSTL-18 Class II	25		25	1.660	0.790	0	0.83
1.8-V HSTL Class I			50	1.660	0.790	0	0.83
1.8-V HSTL Class II			25	1.660	0.790	0	0.83
1.5-V HSTL Class I			50	1.375	0.648	0	0.6875
1.5-V HSTL Class II			25	1.375	0.648	0	0.6875
1.2-V HSTL with OCT				1.140		0	0.570
Differential SSTL-2 Class I	25		50	2.325	1.123	0	1.1625
Differential SSTL-2 Class II	25		25	2.325	1.123	0	1.1625
Differential SSTL-18 Class I	50		50	1.660	0.790	0	0.83
Differential SSTL-18 Class II	25		25	1.660	0.790	0	0.83
1.5-V differential HSTL Class I			50	1.375	0.648	0	0.6875
1.5-V differential HSTL Class II			25	1.375	0.648	0	0.6875
1.8-V differential HSTL Class I			50	1.660	0.790	0	0.83
1.8-V differential HSTL Class II			25	1.660	0.790	0	0.83
LVDS		100		2.325		0	1.1625
LVPECL		100		3.135		0	1.5675

Notes to Table 4–53:

(1) Input measurement point at internal node is 0.5 V_{CCINT}.
(2) Output measuring point for V_{MEAS} at buffer output is 0.5 V_{CCIO}.
(3) Input stimulus edge rate is 0 to V_{CC} in 0.2 ns (internal signal) from the driver preceding the I/O buffer.
(4) Less than 50-mV ripple on V_{CCIO} and V_{CCPD}, V_{CCINT} = 1.15 V with less than 30-mV ripple.
(5) V_{CCPD} = 2.97 V, less than 50-mV ripple on V_{CCIO} and V_{CCPD}, V_{CCINT} = 1.15 V.

Figures 4–9 and 4–10 show the measurement setup for output disable and output enable timing.

Figure 4–9. Measurement Setup for t_{xz} *Note (1)*



Note to Figure 4–9:

(1) V_{CCINT} is 1.12 V for this measurement.

Figure 4–10. Measurement Setup for t_{zx}

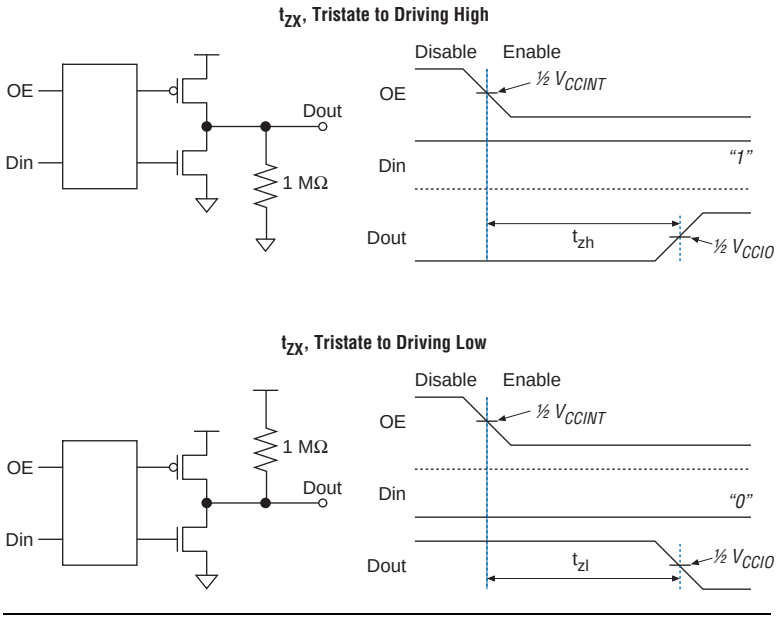


Table 4–54 specifies the input timing measurement setup.

Table 4–54. Timing Measurement Methodology for Input Pins (Part 1 of 2) *Notes (1), (2), (3), (4)*

I/O Standard	Measurement Conditions			Measurement Point
	V_{CCIO} (V)	V_{REF} (V)	Edge Rate (ns)	VMEAS (V)
LVTTL *(5)*	3.135		3.135	1.5675
LVCMOS *(5)*	3.135		3.135	1.5675
2.5 V *(5)*	2.375		2.375	1.1875
1.8 V *(5)*	1.710		1.710	0.855
1.5 V *(5)*	1.425		1.425	0.7125
PCI *(6)*	2.970		2.970	1.485
PCI-X *(6)*	2.970		2.970	1.485
SSTL-2 Class I	2.325	1.163	2.325	1.1625
SSTL-2 Class II	2.325	1.163	2.325	1.1625
SSTL-18 Class I	1.660	0.830	1.660	0.83
SSTL-18 Class II	1.660	0.830	1.660	0.83
1.8-V HSTL Class I	1.660	0.830	1.660	0.83

Table 4–54. Timing Measurement Methodology for Input Pins (Part 2 of 2) *Notes (1), (2), (3), (4)*

I/O Standard	Measurement Conditions			Measurement Point
	V_{CCIO} (V)	V_{REF} (V)	Edge Rate (ns)	VMEAS (V)
1.8-V HSTL Class II	1.660	0.830	1.660	0.83
1.5-V HSTL Class I	1.375	0.688	1.375	0.6875
1.5-V HSTL Class II	1.375	0.688	1.375	0.6875
1.2-V HSTL with OCT	1.140	0.570	1.140	0.570
Differential SSTL-2 Class I	2.325	1.163	2.325	1.1625
Differential SSTL-2 Class II	2.325	1.163	2.325	1.1625
Differential SSTL-18 Class I	1.660	0.830	1.660	0.83
Differential SSTL-18 Class II	1.660	0.830	1.660	0.83
1.5-V differential HSTL Class I	1.375	0.688	1.375	0.6875
1.5-V differential HSTL Class II	1.375	0.688	1.375	0.6875
1.8-V differential HSTL Class I	1.660	0.830	1.660	0.83
1.8-V differential HSTL Class II	1.660	0.830	1.660	0.83
LVDS	2.325		0.100	1.1625
LVPECL	3.135		0.100	1.5675

Notes to Table 4–54:

(1) Input buffer sees no load at buffer input.
(2) Input measuring point at buffer input is 0.5 V_{CCIO}.
(3) Output measuring point is 0.5 V_{CC} at internal node.
(4) Input edge rate is 1 V/ns.
(5) Less than 50-mV ripple on V_{CCIO} and V_{CCPD}, V_{CCINT} = 1.15 V with less than 30-mV ripple.
(6) V_{CCPD} = 2.97 V, less than 50-mV ripple on V_{CCIO} and V_{CCPD}, V_{CCINT} = 1.15 V.

Table 4–55 shows the Stratix II GX performance for some common designs. All performance values were obtained with the Quartus II software compilation of LPM or MegaCore functions for FIR and FFT designs.

Table 4–55. Stratix II GX Performance Notes (Part 1 of 3) *Note (1)*

Applications		Resources Used			Performance				
		ALUTs	TriMatrix Memory Blocks	DSP Blocks	-3 Speed Grade (2)	-3 Speed Grade (3)	-4 Speed Grade	-5 Speed Grade	Units
LE	16-to-1 multiplexer (4)	21	0	0	657.03	620.73	589.62	477.09	MHz
	32-to-1 multiplexer (4)	38	0	0	534.75	517.33	472.81	369.27	MHz
	16-bit counter	16	0	0	568.18	539.66	507.61	422.47	MHz
	64-bit counter	64	0	0	242.54	231.0	217.77	180.31	MHz
TriMatrix Memory M512 block	Simple dual-port RAM 32 x 18bit	0	1	0	500.0	476.19	447.22	373.13	MHz
	FIFO 32 x 18 bit	22	1	0	500.00	476.19	460.82	373.13	MHz
TriMatrix Memory M4K block	Simple dual-port RAM 128 x 36bit	0	1	0	540.54	515.46	483.09	401.6	MHz
	True dual-port RAM 128 x 18bit	0	1	0	540.54	515.46	483.09	401.6	MHz
	FIFO 128 x 36 bit	22	1	0	524.10	500.25	466.41	381.38	MHz

Table 4–55. Stratix II GX Performance Notes (Part 2 of 3) *Note (1)*

Applications		Resources Used			Performance				
		ALUTs	TriMatrix Memory Blocks	DSP Blocks	-3 Speed Grade (2)	-3 Speed Grade (3)	-4 Speed Grade	-5 Speed Grade	Units
TriMatrix Memory MegaRAM block	Single port RAM 4K x 144bit	0	1	0	349.65	333.33	313.47	261.09	MHz
	Simple dual-port RAM 4K x 144bit	0	1	0	420.16	400.0	375.93	313.47	MHz
	True dual-port RAM 4K x 144 bit	0	1	0	349.65	333.33	313.47	261.09	MHz
	Single port RAM 8K x 72 bit	0	1	0	354.6	337.83	317.46	263.85	MHz
	Simple dual-port RAM 8K x 72 bit	0	1	0	420.16	400.0	375.93	313.47	MHz
	True dual-port RAM 8K x 72 bit	0	1	0	349.65	333.33	313.47	261.09	MHz
	Single port RAM 16K x 36 bit	0	1	0	364.96	347.22	325.73	271.73	MHz
	Simple dual-port RAM 16K x 36 bit	0	1	0	420.16	400.0	375.93	313.47	MHz
	True dual-port RAM 16K x 36 bit	0	1	0	359.71	342.46	322.58	268.09	MHz
	Single port RAM 32K x 18 bit	0	1	0	364.96	347.22	325.73	271.73	MHz
	Simple dual-port RAM 32K x 18 bit	0	1	0	420.16	400.0	375.93	313.47	MHz
	True dual-port RAM 32K x 18 bit	0	1	0	359.71	342.46	322.58	268.09	MHz

Table 4–55. Stratix II GX Performance Notes (Part 3 of 3) *Note (1)*

Applications		Resources Used			Performance				
		ALUTs	TriMatrix Memory Blocks	DSP Blocks	-3 Speed Grade *(2)*	-3 Speed Grade *(3)*	-4 Speed Grade	-5 Speed Grade	Units
TriMatrix Memory MegaRAM block (cont.)	Single port RAM 64K x 9 bit	0	1	0	364.96	347.22	325.73	271.73	MHz
	Simple dual-port RAM 64K x 9 bit	0	1	0	420.16	400.0	375.93	313.47	MHz
	True dual-port RAM 64K x 9 bit	0	1	0	359.71	342.46	322.58	268.09	MHz
DSP block	9 x 9-bit multiplier *(5)*	0	0	1	430.29	409.16	385.2	320.1	MHz
	18 x 18-bit multiplier *(5)*	0	0	1	410.17	390.01	367.1	305.06	MHz
	18 x 18-bit multiplier *(7)*	0	0	1	450.04	428.08	403.22	335.12	MHz
	36 x 36-bit multiplier *(5)*	0	0	1	250.0	238.15	224.01	186.6	MHz
	36 x 36-bit multiplier *(6)*	0	0	1	410.17	390.01	367.1	305.06	MHz
	18-bit, 4-tap FIR filter	0	0	1	410.17	390.01	367.1	305.06	MHz

Notes to Table 4–55:

(1) These design performance numbers were obtained using the Quartus II software.
(2) This column refers to -3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(3) This column refers to -3 speed grades for EP2SGX130 devices.
(4) This application uses registered inputs and outputs.
(5) This application uses registered multiplier input and output stages within the DSP block.
(6) This application uses registered multiplier input, pipeline, and output stages within the DSP block.
(7) This application uses registered multiplier inputs with outputs of the multiplier stage feeding the accumulator or subtractor within the DSP block.

Internal Timing Parameters

Refer to Tables 4–56 through 4–61 for internal timing parameters.

Table 4–56. LE_FF Internal Timing Microparameters

Symbol	Parameter	-3 Speed Grade (1)		-3 Speed Grade (2)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{SU}	LE register setup time before clock	90		95		101		121		ps
t_{H}	LE register hold time after clock	149		157		167		200		ps
t_{CO}	LE register clock-to-output delay	62	94	62	99	62	105	62	127	ps
t_{CLR}	Minimum clear pulse width	204		214		227		273		ps
t_{PRE}	Minimum preset pulse width	204		214		227		273		ps
t_{CLKL}	Minimum clock low time	612		642		683		820		ps
t_{CLKH}	Minimum clock high time	612		642		683		820		ps
t_{LUT}		170	378	170	397	170	422	170	507	
t_{ADDER}		372	619	372	650	372	691	372	829	

Notes to Table 4–56:
(1) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(2) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–57. IOE Internal Timing Microparameters (Part 1 of 2)

Symbol	Parameter	-3 Speed Grade (1)		-3 Speed Grade (2)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{SU}	IOE input and output register setup time before clock	122		128		136		163		ps
t_{H}	IOE input and output register hold time after clock	72		75		80		96		ps
t_{CO}	IOE input and output register clock-to-output delay	101	169	101	177	101	188	101	226	ps

Table 4–57. IOE Internal Timing Microparameters (Part 2 of 2)

Symbol	Parameter	-3 Speed Grade (1)		-3 Speed Grade (2)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{PIN2COMBOUT\_R}$	Row input pin to IOE combinational output	410	760	410	798	410	848	410	1018	ps
$t_{PIN2COMBOUT\_C}$	Column input pin to IOE combinational output	428	787	428	825	428	878	428	1054	ps
$t_{COMBIN2PIN\_R}$	Row IOE data input to combinational output pin	1101	2026	1101	2127	1101	2261	1101	2439	ps
$t_{COMBIN2PIN\_C}$	Column IOE data input to combinational output pin	991	1854	991	1946	991	2069	991	2246	ps
t_{CLR}	Minimum clear pulse width	200		210		223		268		ps
t_{PRE}	Minimum preset pulse width	200		210		223		268		ps
t_{CLKL}	Minimum clock low time	600		630		669		804		ps
t_{CLKH}	Minimum clock high time	600		630		669		804		ps

(1) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(2) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–58. DSP Block Internal Timing Microparameters (Part 1 of 2)

Symbol	Parameter	-3 Speed Grade (1)		-3 Speed Grade (2)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{SU}	Input, pipeline, and output register setup time before clock	50		52		55		67		ps
t_{H}	Input, pipeline, and output register hold time after clock	180		189		200		241		ps
t_{CO}	Input, pipeline, and output register clock-to-output delay	0	0	0	0	0	0	0	0	ps

Table 4–58. DSP Block Internal Timing Microparameters (Part 2 of 2)

Symbol	Parameter	-3 Speed Grade (1)		-3 Speed Grade (2)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{INREG2PIPE9}$	Input register to DSP block pipeline register in 9 × 9-bit mode	1312	2030	1312	2131	1312	2266	1312	2720	ps
$t_{INREG2PIPE18}$	Input register to DSP block pipeline register in 18 × 18-bit mode	1302	2010	1302	2110	1302	2244	1302	2693	ps
$t_{INREG2PIPE36}$	Input register to DSP block pipeline register in 36 × 36-bit mode	1302	2010	1302	2110	1302	2244	1302	2693	ps
$t_{PIPE2OUTREG2ADD}$	DSP block pipeline register to output register delay in two-multipliers adder mode	924	1450	924	1522	924	1618	924	1943	ps
$t_{PIPE2OUTREG4ADD}$	DSP block pipeline register to output register delay in four-multipliers adder mode	1134	1850	1134	1942	1134	2065	1134	2479	ps
t_{PD9}	Combinational input to output delay for 9 × 9	2100	2880	2100	3024	2100	3214	2100	3859	ps
t_{PD18}	Combinational input to output delay for 18 × 18	2110	2990	2110	3139	2110	3337	2110	4006	ps
t_{PD36}	Combinational input to output delay for 36 × 36	2939	4450	2939	4672	2939	4967	2939	5962	ps
t_{CLR}	Minimum clear pulse width	2212		2322		2469		2964		ps
t_{CLKL}	Minimum clock low time	1190		1249		1328		1594		ps
t_{CLKH}	Minimum clock high time	1190		1249		1328		1594		ps

(1) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(2) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–59. M512 Block Internal Timing Microparameters (Part 1 of 2)

Symbol	Parameter	-3 Speed Grade (2)		-3 Speed Grade (3)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{M512RC}	Synchronous read cycle time	2089	2318	2089	2433	2089	2587	2089	3104	ps
$t_{M512WERESU}$	Write or read enable setup time before clock	22		23		24		29		ps
$t_{M512WEREH}$	Write or read enable hold time after clock	203		213		226		272		ps
$t_{M512DATASU}$	Data setup time before clock	22		23		24		29		ps
$t_{M512DATAH}$	Data hold time after clock	203		213		226		272		ps
$t_{M512WADDRSU}$	Write address setup time before clock	22		23		24		29		ps
$t_{M512WADDRH}$	Write address hold time after clock	203		213		226		272		ps
$t_{M512RADDRSU}$	Read address setup time before clock	22		23		24		29		ps
$t_{M512RADDRH}$	Read address hold time after clock	203		213		226		272		ps
$t_{M512DATACO1}$	Clock-to-output delay when using output registers	298	478	298	501	298	533	298	640	ps
$t_{M512DATACO2}$	Clock-to-output delay without output registers	2102	2345	2102	2461	2102	2616	2102	3141	ps
$t_{M512CLKL}$	Minimum clock low time	1315		1380		1468		1762		ps
$t_{M512CLKH}$	Minimum clock high time	1315		1380		1468		1762		ps

Table 4–59. M512 Block Internal Timing Microparameters (Part 2 of 2)

Symbol	Parameter	-3 Speed Grade (2)		-3 Speed Grade (3)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{M512CLR}$	Minimum clear pulse width	144		151		160		192		ps

(1) The M512 block f_{MAX} obtained using the Quartus II software does not necessarily equal to 1/TM512RC.
(2) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(3) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–60. M4K Block Internal Timing Microparameters (Part 1 of 2) *Note (1)*

Symbol	Parameter	-3 Speed Grade (2)		-3 Speed Grade (3)		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{M4KRC}	Synchronous read cycle time	1462	2240	1462	2351	1462	2500	1462	3000	ps
$t_{M4KWERESU}$	Write or read enable setup time before clock	22		23		24		29		ps
$t_{M4KWEREH}$	Write or read enable hold time after clock	203		213		226		272		ps
$t_{M4KBESU}$	Byte enable setup time before clock	22		23		24		29		ps
t_{M4KBEH}	Byte enable hold time after clock	203		213		226		272		ps
$t_{M4KDATAASU}$	A port data setup time before clock	22		23		24		29		ps
$t_{M4KDATAAH}$	A port data hold time after clock	203		213		226		272		ps
$t_{M4KADDRASU}$	A port address setup time before clock	22		23		24		29		ps
$t_{M4KADDRAH}$	A port address hold time after clock	203		213		226		272		ps
$t_{M4KDATABSU}$	B port data setup time before clock	22		23		24		29		ps

Table 4–60. M4K Block Internal Timing Microparameters (Part 2 of 2) *Note (1)*

Symbol	Parameter	-3 Speed Grade *(2)*		-3 Speed Grade *(3)*		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{M4KDATABH}$	B port data hold time after clock	203		213		226		272		ps
$t_{M4KRADDRBSU}$	B port address setup time before clock	22		23		24		29		ps
$t_{M4KRADDRBH}$	B port address hold time after clock	203		213		226		272		ps
$t_{M4KDATACO1}$	Clock-to-output delay when using output registers	334	524	334	549	334	584	334	701	ps
$t_{M4KDATACO2}$	Clock-to-output delay without output registers	1616	2453	1616	2574	1616	2737	1616	3286	ps
$t_{M4KCLKH}$	Minimum clock high time	1250		1312		1395		1675		ps
$t_{M4KCLKL}$	Minimum clock low time	1250		1312		1395		1675		ps
t_{M4KCLR}	Minimum clear pulse width	144		151		160		192		ps

(1) The M512 block f_{MAX} obtained using the Quartus II software does not necessarily equal to 1/TM4KRC.
(2) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(3) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–61. M-RAM Block Internal Timing Microparameters (Part 1 of 2) *Note (1)*

Symbol	Parameter	-3 Speed Grade *(2)*		-3 Speed Grade *(3)*		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{MEGARC}	Synchronous read cycle time	1866	2774	1866	2911	1866	3096	1866	3716	ps
$t_{MEGAWERESU}$	Write or read enable setup time before clock	144		151		160		192		ps
$t_{MEGAWEREH}$	Write or read enable hold time after clock	39		40		43		52		ps

Table 4–61. M-RAM Block Internal Timing Microparameters (Part 2 of 2) *Note (1)*

Symbol	Parameter	-3 Speed Grade *(2)*		-3 Speed Grade *(3)*		-4 Speed Grade		-5 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{MEGABESU}$	Byte enable setup time before clock	-9		-10		-11		-13		ps
$t_{MEGABEH}$	Byte enable hold time after clock	39		40		43		52		ps
$t_{MEGADATAASU}$	A port data setup time before clock	50		52		55		67		ps
$t_{MEGADATAAH}$	A port data hold time after clock	243		255		271		325		ps
$t_{MEGAADDRASU}$	A port address setup time before clock	589		618		657		789		ps
$t_{MEGAADDRAH}$	A port address hold time after clock	-347		-365		-388		-465		ps
$t_{MEGADATABSU}$	B port setup time before clock	50		52		55		67		ps
$t_{MEGADATABH}$	B port hold time after clock	243		255		271		325		ps
$t_{MEGAADDRBSU}$	B port address setup time before clock	589		618		657		789		ps
$t_{MEGAADDRBH}$	B port address hold time after clock	-347		-365		-388		-465		ps
$t_{MEGADATACO1}$	Clock-to-output delay when using output registers	480	715	480	749	480	797	480	957	ps
$t_{MEGADATACO2}$	Clock-to-output delay without output registers	1950	2899	1950	3042	1950	3235	1950	3884	ps
$t_{MEGACLKL}$	Minimum clock low time	1250		1312		1395		1675		ps
$t_{MEGACLKH}$	Minimum clock high time	1250		1312		1395		1675		ps
$t_{MEGACLR}$	Minimum clear pulse width	144		151		160		192		ps

(1) The M512 block f_{MAX} obtained using the Quartus II software does not necessarily equal to 1/TMEGARC.
(2) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(3) This column refers to –3 speed grades for EP2SGX130 devices.

Stratix II GX Clock Timing Parameters

See Tables 4–62 through 4–78 for Stratix II GX clock timing parameters.

Table 4–62. Stratix II GX Clock Timing Parameters

Symbol	Parameter
t_{CIN}	Delay from clock pad to I/O input register
t_{COUT}	Delay from clock pad to I/O output register
t_{PLLCIN}	Delay from PLL `inclk` pad to I/O input register
$t_{PLLCOUT}$	Delay from PLL `inclk` pad to I/O output register

EP2SGX30 Clock Timing Parameters

Tables 4–63 through 4–66 show the maximum clock timing parameters for EP2SGX30 devices.

Table 4–63. EP2SGX30 Column Pins Global Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.615	1.633	2.669	2.968	3.552	ns
t_{COUT}	1.450	1.468	2.427	2.698	3.228	ns
t_{PLLCIN}	0.11	0.129	0.428	0.466	0.547	ns
$t_{PLLCOUT}$	-0.055	-0.036	0.186	0.196	0.223	ns

Table 4–64. EP2SGX30 Row Pins Global Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.365	1.382	2.280	2.535	3.033	ns
t_{COUT}	1.370	1.387	2.276	2.531	3.028	ns
t_{PLLCIN}	-0.151	-0.136	0.043	0.037	0.032	ns
$t_{PLLCOUT}$	-0.146	-0.131	0.039	0.033	0.027	ns

Table 4–65. EP2SGX30 Column Pins Regional Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.493	1.507	2.522	2.806	3.364	ns
t_{COUT}	1.353	1.372	2.525	2.809	3.364	ns
t_{PLLCIN}	0.087	0.104	0.237	0.253	0.292	ns
$t_{PLLCOUT}$	-0.078	-0.061	0.237	0.253	0.29	ns

Table 4–66. EP2SGX30 Row Pins Regional Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.246	1.262	2.437	2.712	3.246	ns
t_{COUT}	1.251	1.267	2.437	2.712	3.246	ns
t_{PLLCIN}	-0.18	-0.167	0.215	0.229	0.263	ns
$t_{PLLCOUT}$	-0.175	-0.162	0.215	0.229	0.263	ns

EP2SGX60 Clock Timing Parameters

Tables 4–67 through 4–70 show the maximum clock timing parameters for EP2SGX60 devices.

Table 4–67. EP2SGX60 Column Pins Global Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.722	1.736	2.940	3.275	3.919	ns
t_{COUT}	1.557	1.571	2.698	3.005	3.595	ns
t_{PLLCIN}	0.037	0.051	0.474	0.521	0.613	ns
$t_{PLLCOUT}$	-0.128	-0.114	0.232	0.251	0.289	ns

Table 4–68. EP2SGX60 Row Pins Global Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.494	1.508	2.582	2.875	3.441	ns
t_{COUT}	1.499	1.513	2.578	2.871	3.436	ns
t_{PLLCIN}	-0.183	-0.168	0.116	0.122	0.135	ns
$t_{PLLCOUT}$	-0.178	-0.163	0.112	0.118	0.13	ns

Table 4–69. EP2SGX60 Column Pins Regional Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.577	1.591	2.736	3.048	3.648	ns
t_{COUT}	1.412	1.426	2.740	3.052	3.653	ns
t_{PLLCIN}	0.065	0.08	0.334	0.361	0.423	ns
$t_{PLLCOUT}$	-0.1	-0.085	0.334	0.361	0.423	ns

Table 4–70. EP2SGX60 Row Pins Regional Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.342	1.355	2.716	3.024	3.622	ns
t_{COUT}	1.347	1.360	2.716	3.024	3.622	ns
t_{PLLCIN}	-0.18	-0.166	0.326	0.352	0.412	ns
$t_{PLLCOUT}$	-0.175	-0.161	0.334	0.361	0.423	ns

EP2SGX90 Clock Timing Parameters

Tables 4–71 through 4–74 show the maximum clock timing parameters for EP2SGX90 devices.

Table 4–71. EP2SGX90 Column Pins Global Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.861	1.878	3.115	3.465	4.143	ns
t_{COUT}	1.696	1.713	2.873	3.195	3.819	ns
t_{PLLCIN}	-0.254	-0.237	0.171	0.179	0.206	ns
$t_{PLLCOUT}$	-0.419	-0.402	-0.071	-0.091	-0.118	ns

Table 4–72. EP2SGX90 Row Pins Global Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.634	1.650	2.768	3.076	3.678	ns
t_{COUT}	1.639	1.655	2.764	3.072	3.673	ns
t_{PLLCIN}	-0.481	-0.465	-0.189	-0.223	-0.279	ns
$t_{PLLCOUT}$	-0.476	-0.46	-0.193	-0.227	-0.284	ns

Table 4–73. EP2SGX90 Column Pins Regional Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.688	1.702	2.896	3.224	3.856	ns
t_{COUT}	1.551	1.569	2.893	3.220	3.851	ns
t_{PLLCIN}	-0.105	-0.089	0.224	0.241	0.254	ns
$t_{PLLCOUT}$	-0.27	-0.254	0.224	0.241	0.254	ns

Table 4–74. EP2SGX90 Row Pins Regional Clock Timing Parameters						
Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.444	1.461	2.792	3.108	3.716	ns
t_{COUT}	1.449	1.466	2.792	3.108	3.716	ns
t_{PLLCIN}	-0.348	-0.333	0.204	0.217	0.243	ns
$t_{PLLCOUT}$	-0.343	-0.328	0.212	0.217	0.254	ns

EP2SGX130 Clock Timing Parameters

Tables 4–75 through 4–78 show the maximum clock timing parameters for EP2SGX130 devices.

Table 4–75. EP2SGX130 Column Pins Global Clock Timing Parameters						
Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.980	1.998	3.491	3.706	4.434	ns
t_{COUT}	1.815	1.833	3.237	3.436	4.110	ns
t_{PLLCIN}	-0.027	-0.009	0.307	0.322	0.376	ns
$t_{PLLCOUT}$	-0.192	-0.174	0.053	0.052	0.052	ns

Table 4–76. EP2SGX130 Row Pins Global Clock Timing Parameters						
Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.741	1.759	3.112	3.303	3.950	ns
t_{COUT}	1.746	1.764	3.108	3.299	3.945	ns
t_{PLLCIN}	-0.261	-0.243	-0.089	-0.099	-0.129	ns
$t_{PLLCOUT}$	-0.256	-0.238	-0.093	-0.103	-0.134	ns

Table 4–77. EP2SGX130 Column Pins Regional Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.815	1.834	3.218	3.417	4.087	ns
t_{COUT}	1.650	1.669	3.218	3.417	4.087	ns
t_{PLLCIN}	0.116	0.134	0.349	0.364	0.426	ns
$t_{PLLCOUT}$	-0.049	-0.031	0.361	0.378	0.444	ns

Table 4–78. EP2SGX130 Row Pins Regional Clock Timing Parameters

Parameter	Fast Corner		-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Units
	Industrial	Commercial				
t_{CIN}	1.544	1.560	3.195	3.395	4.060	ns
t_{COUT}	1.549	1.565	3.195	3.395	4.060	ns
t_{PLLCIN}	-0.149	-0.132	0.34	0.356	0.417	ns
$t_{PLLCOUT}$	-0.144	-0.127	0.342	0.356	0.417	ns

Clock Network Skew Adders

The Quartus II software models skew within dedicated clock networks such as global and regional clocks. Therefore, the intra-clock network skew adder is not specified. Table 4–79 specifies the intra-clock skew between any two clock networks driving any registers in the Stratix II GX device.

Table 4–79. Clock Network Specifications (Part 1 of 2)

Name	Description	Min	Typ	Max	Unit
Clock skew adder EP2SGX30 *(1)*	Inter-clock network, same side			±50	ps
	Inter-clock network, entire chip			±100	ps
Clock skew adder EP2SGX60 *(1)*	Inter-clock network, same side			±50	ps
	Inter-clock network, entire chip			±100	ps
Clock skew adder EP2SGX90 *(1)*	Inter-clock network, same side			±55	ps
	Inter-clock network, entire chip			±110	ps

Table 4–79. Clock Network Specifications (Part 2 of 2)					
Name	**Description**	**Min**	**Typ**	**Max**	**Unit**
Clock skew adder EP2SGX130 *(1)*	Inter-clock network, same side			±63	ps
	Inter-clock network, entire chip			±125	ps

(1) This is in addition to intra-clock network skew, which is modeled in the Quartus II software.

IOE Programmable Delay

See Tables 4–80 and 4–81 for IOE programmable delay.

Table 4–80. Stratix II GX IOE Programmable Delay on Column Pins *Note (1)*													
Parameter	**Paths Affected**	**Available Settings**	**Minimum Timing**		**-3 Speed Grade** *(2)*		**-3 Speed Grade** *(3)*		**-4 Speed Grade**		**-5 Speed Grade**		**Unit**
			Min Offset	**Max Offset**	**Min Offset**	**Max Offset**	**Min Offset**	**Max Offset**	**Min Offset**	**Max Offset**	**Min Offset**	**Max Offset**	
Input delay from pin to internal cells	Pad to I/O dataout to core	8	0	1781	0	2881	0	3025	0	3217	0	3,860	ps
Input delay from pin to input register	Pad to I/O input register	64	0	2053	0	3275	0	3439	0	3657	0	4388	ps
Delay from output register to output pin	I/O output register to pad	2	0	332	0	500	0	525	0	559	0	670	ps
Output enable pin delay	t_{XZ}, t_{ZX}	2	0	320	0	483	0	507	0	539	0	647	ps

(1) The incremental values for the settings are generally linear. For the exact delay associated with each setting, use the latest version of the Quartus II software.
(2) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(3) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–81. Stratix II GX IOE Programmable Delay on Row Pins Note (1)													
Parameter	Paths Affected	Available Settings	Minimum Timing		-3 Speed Grade		-3 Speed Grade		-4 Speed Grade		-5 Speed Grade		Unit
			Min Offset	Max Offset	Min Offset	Max Offset	Min Offset	Max Offset	Min Offset	Max Offset	Min Offset	Max Offset	
Input delay from pin to internal cells	Pad to I/O dataout to logic array	8	0	1782	0	2876	0	3020	0	3212	0	3853	ps
Input delay from pin to input register	Pad to I/O input register	64	0	2054	0	3270	0	3434	0	3652	0	4381	ps
Delay from output register to output pin	I/O output register to pad	2	0	332	0	500	0	525	0	559	0	670	ps
Output enable pin delay	t_{XZ}, t_{ZX}	2	0	320	0	483	0	507	0	539	0	647	ps

(1) The incremental values for the settings are generally linear. For the exact delay associated with each setting, use the latest version of the Quartus II software.

Default Capacitive Loading of Different I/O Standards

See Table 4–82 for default capacitive loading of different I/O standards.

Table 4–82. Default Loading of Different I/O Standards for Stratix II GX Devices (Part 1 of 2)		
I/O Standard	Capacitive Load	Unit
LVTTL	0	pF
LVCMOS	0	pF
2.5 V	0	pF
1.8 V	0	pF
1.5 V	0	pF
PCI	10	pF
PCI-X	10	pF
SSTL-2 Class I	0	pF
SSTL-2 Class II	0	pF

Table 4–82. Default Loading of Different I/O Standards for Stratix II GX Devices (Part 2 of 2)

I/O Standard	Capacitive Load	Unit
SSTL-18 Class I	0	pF
SSTL-18 Class II	0	pF
1.5-V HSTL Class I	0	pF
1.5-V HSTL Class II	0	pF
1.8-V HSTL Class I	0	pF
1.8-V HSTL Class II	0	pF
Differential SSTL-2 Class I	0	pF
Differential SSTL-2 Class II	0	pF
Differential SSTL-18 Class I	0	pF
Differential SSTL-18 Class II	0	pF
1.5-V differential HSTL Class I	0	pF
1.5-V differential HSTL Class II	0	pF
1.8-V differential HSTL Class I	0	pF
1.8-V differential HSTL Class II	0	pF
LVDS	0	pF

I/O Delays

See Tables 4–83 through 4–87 for I/O delays.

Table 4–83. I/O Delay Parameters

Symbol	Parameter
t_{DIP}	Delay from I/O datain to output pad
t_{OP}	Delay from I/O output register to output pad
t_{PCOUT}	Delay from input pad to I/O dataout to core
t_{PI}	Delay from input pad to I/O input register

Table 4–84. Stratix II GX I/O Input Delay for Column Pins (Part 1 of 3)

I/O Standard	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(2)*	-3 Speed Grade *(3)*	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	t_{PI}	707	1223	1282	1364	1637	ps
	t_{PCOUT}	428	787	825	878	1054	ps

Table 4–84. Stratix II GX I/O Input Delay for Column Pins (Part 2 of 3)

I/O Standard	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(2)*	-3 Speed Grade *(3)*	-4 Speed Grade	-5 Speed Grade	Unit
2.5 V	t_{PI}	717	1210	1269	1349	1619	ps
	t_{PCOUT}	438	774	812	863	1036	ps
1.8 V	t_{PI}	783	1366	1433	1523	1829	ps
	t_{PCOUT}	504	930	976	1037	1246	ps
1.5 V	t_{PI}	786	1436	1506	1602	1922	ps
	t_{PCOUT}	507	1000	1049	1116	1339	ps
LVCMOS	t_{PI}	707	1223	1282	1364	1637	ps
	t_{PCOUT}	428	787	825	878	1054	ps
SSTL-2 Class I	t_{PI}	530	818	857	912	1094	ps
	t_{PCOUT}	251	382	400	426	511	ps
SSTL-2 Class II	t_{PI}	530	818	857	912	1094	ps
	t_{PCOUT}	251	382	400	426	511	ps
SSTL-18 Class I	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
SSTL-18 Class II	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
1.5-V HSTL Class I	t_{PI}	587	993	1041	1107	1329	ps
	t_{PCOUT}	308	557	584	621	746	ps
1.5-V HSTL Class II	t_{PI}	587	993	1041	1107	1329	ps
	t_{PCOUT}	308	557	584	621	746	ps
1.8-V HSTL Class I	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
1.8-V HSTL Class II	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
PCI	t_{PI}	712	1214	1273	1354	1625	ps
	t_{PCOUT}	433	778	816	868	1042	ps
PCI-X	t_{PI}	712	1214	1273	1354	1625	ps
	t_{PCOUT}	433	778	816	868	1042	ps
Differential SSTL-2 Class I *(1)*	t_{PI}	530	818	857	912	1094	ps
	t_{PCOUT}	251	382	400	426	511	ps

Table 4–84. Stratix II GX I/O Input Delay for Column Pins (Part 3 of 3)							
I/O Standard	**Parameter**	**Fast Corner Industrial/ Commercial**	**-3 Speed Grade** *(2)*	**-3 Speed Grade** *(3)*	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
Differential SSTL-2 Class II *(1)*	t_{PI}	530	818	857	912	1094	ps
	t_{PCOUT}	251	382	400	426	511	ps
Differential SSTL-18 Class I *(1)*	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
Differential SSTL-18 Class II *(1)*	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
1.8-V differential HSTL Class I *(1)*	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
1.8-V differential HSTL Class II *(1)*	t_{PI}	569	898	941	1001	1201	ps
	t_{PCOUT}	290	462	484	515	618	ps
1.5-V differential HSTL Class I *(1)*	t_{PI}	587	993	1041	1107	1329	ps
	t_{PCOUT}	308	557	584	621	746	ps
1.5-V differential HSTL Class II *(1)*	t_{PI}	587	993	1041	1107	1329	ps
	t_{PCOUT}	308	557	584	621	746	ps

(1) These I/O standards are only supported on DQS pins.
(2) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(3) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–85. Stratix II GX I/O Input Delay for Row Pins (Part 1 of 3)							
I/O Standard	**Parameter**	**Fast Corner Industrial/ Commercial**	**-3 Speed Grade** *(2)*	**-3 Speed Grade** *(3)*	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
LVTTL	t_{PI}	749	1287	1350	1435	1723	ps
	t_{PCOUT}	410	760	798	848	1018	ps
2.5 V	t_{PI}	761	1273	1335	1419	1704	ps
	t_{PCOUT}	422	746	783	832	999	ps
1.8 V	t_{PI}	827	1427	1497	1591	1911	ps
	t_{PCOUT}	488	900	945	1004	1206	ps
1.5 V	t_{PI}	830	1498	1571	1671	2006	ps
	t_{PCOUT}	491	971	1019	1084	1301	ps

Table 4–85. Stratix II GX I/O Input Delay for Row Pins (Part 2 of 3)

I/O Standard	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(2)*	-3 Speed Grade *(3)*	-4 Speed Grade	-5 Speed Grade	Unit
LVCMOS	t_{PI}	749	1287	1350	1435	1723	ps
	t_{PCOUT}	410	760	798	848	1018	ps
SSTL-2 Class I	t_{PI}	573	879	921	980	1176	ps
	t_{PCOUT}	234	352	369	393	471	ps
SSTL-2 Class II	t_{PI}	573	879	921	980	1176	ps
	t_{PCOUT}	234	352	369	393	471	ps
SSTL-18 Class I	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
SSTL-18 Class II	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
1.5-V HSTL Class I	t_{PI}	631	1056	1107	1177	1413	ps
	t_{PCOUT}	292	529	555	590	708	ps
1.5-V HSTL Class II	t_{PI}	631	1056	1107	1177	1413	ps
	t_{PCOUT}	292	529	555	590	708	ps
1.8-V HSTL Class I	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
1.8-V HSTL Class II	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
PCI	t_{PI}	830	1498	1571	1671	2006	ps
	t_{PCOUT}	491	971	1019	1084	1301	ps
PCI-X	t_{PI}	830	1498	1571	1671	2006	ps
	t_{PCOUT}	491	971	1019	1084	1301	ps
LVDS *(1)*	t_{PI}	540	948	994	1057	1269	ps
	t_{PCOUT}	201	421	442	470	564	ps
HyperTransport	t_{PI}	540	948	994	1057	1269	ps
	t_{PCOUT}	201	421	442	470	564	ps
Differential SSTL-2 Class I	t_{PI}	573	879	921	980	1176	ps
	t_{PCOUT}	234	352	369	393	471	ps
Differential SSTL-2 Class II	t_{PI}	573	879	921	980	1176	ps
	t_{PCOUT}	234	352	369	393	471	ps

Table 4–85. Stratix II GX I/O Input Delay for Row Pins (Part 3 of 3)

I/O Standard	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(2)*	-3 Speed Grade *(3)*	-4 Speed Grade	-5 Speed Grade	Unit
Differential SSTL-18 Class I	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
Differential SSTL-18 Class II	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
1.8-V differential HSTL Class I	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
1.8-V differential HSTL Class II	t_{PI}	605	960	1006	1070	1285	ps
	t_{PCOUT}	266	433	454	483	580	ps
1.5-V differential HSTL Class I	t_{PI}	631	1056	1107	1177	1413	ps
	t_{PCOUT}	292	529	555	590	708	ps
1.5-V differential HSTL Class II	t_{PI}	631	1056	1107	1177	1413	ps
	t_{PCOUT}	292	529	555	590	708	ps

(1) The parameters are only available on the left side of the device.
(2) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(3) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–86. Stratix II GX I/O Output Delay for Column Pins (Part 1 of 7)								
I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(3)*	-3 Speed Grade *(4)*	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	4 mA	t_{OP}	1236	2351	2467	2624	2820	ps
		t_{DIP}	1258	2417	2537	2698	2910	ps
	8 mA	t_{OP}	1091	2036	2136	2272	2448	ps
		t_{DIP}	1113	2102	2206	2346	2538	ps
	12 mA	t_{OP}	1024	2036	2136	2272	2448	ps
		t_{DIP}	1046	2102	2206	2346	2538	ps
	16 mA	t_{OP}	998	1893	1986	2112	2279	ps
		t_{DIP}	1020	1959	2056	2186	2369	ps
	20 mA	t_{OP}	976	1787	1875	1994	2154	ps
		t_{DIP}	998	1853	1945	2068	2244	ps
	24 mA *(1)*	t_{OP}	969	1788	1876	1995	2156	ps
		t_{DIP}	991	1854	1946	2069	2246	ps
LVCMOS	4 mA	t_{OP}	1091	2036	2136	2272	2448	ps
		t_{DIP}	1113	2102	2206	2346	2538	ps
	8 mA	t_{OP}	999	1786	1874	1993	2153	ps
		t_{DIP}	1021	1852	1944	2067	2243	ps
	12 mA	t_{OP}	971	1720	1805	1919	2075	ps
		t_{DIP}	993	1786	1875	1993	2165	ps
	16 mA	t_{OP}	978	1693	1776	1889	2043	ps
		t_{DIP}	1000	1759	1846	1963	2133	ps
	20 mA	t_{OP}	965	1677	1759	1871	2025	ps
		t_{DIP}	987	1743	1829	1945	2115	ps
	24 mA *(1)*	t_{OP}	954	1659	1741	1851	2003	ps
		t_{DIP}	976	1725	1811	1925	2093	ps

Table 4–86. Stratix II GX I/O Output Delay for Column Pins (Part 2 of 7)								
I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(3)*	-3 Speed Grade *(4)*	-4 Speed Grade	-5 Speed Grade	Unit
2.5 V	4 mA	t_{OP}	1053	2063	2165	2302	2480	ps
		t_{DIP}	1075	2129	2235	2376	2570	ps
	8 mA	t_{OP}	1001	1841	1932	2054	2218	ps
		t_{DIP}	1023	1907	2002	2128	2308	ps
	12 mA	t_{OP}	980	1742	1828	1944	2101	ps
		t_{DIP}	1002	1808	1898	2018	2191	ps
	16 mA *(1)*	t_{OP}	962	1679	1762	1873	2027	ps
		t_{DIP}	984	1745	1832	1947	2117	ps
1.8 V	2 mA	t_{OP}	1093	2904	3048	3241	3472	ps
		t_{DIP}	1115	2970	3118	3315	3562	ps
	4 mA	t_{OP}	1098	2248	2359	2509	2698	ps
		t_{DIP}	1120	2314	2429	2583	2788	ps
	6 mA	t_{OP}	1022	2024	2124	2258	2434	ps
		t_{DIP}	1044	2090	2194	2332	2524	ps
	8 mA	t_{OP}	1024	1947	2043	2172	2343	ps
		t_{DIP}	1046	2013	2113	2246	2433	ps
	10 mA	t_{OP}	978	1882	1975	2100	2266	ps
		t_{DIP}	1000	1948	2045	2174	2356	ps
	12 mA *(1)*	t_{OP}	979	1833	1923	2045	2209	ps
		t_{DIP}	1001	1899	1993	2119	2299	ps
1.5 V	2 mA	t_{OP}	1073	2505	2629	2795	3002	ps
		t_{DIP}	1095	2571	2699	2869	3092	ps
	4 mA	t_{OP}	1009	2023	2123	2257	2433	ps
		t_{DIP}	1031	2089	2193	2331	2523	ps
	6 mA	t_{OP}	1012	1923	2018	2146	2315	ps
		t_{DIP}	1034	1989	2088	2220	2405	ps
	8 mA *(1)*	t_{OP}	971	1878	1970	2095	2262	ps
		t_{DIP}	993	1944	2040	2169	2352	ps

Table 4–86. Stratix II GX I/O Output Delay for Column Pins (Part 3 of 7)								
I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(3)*	-3 Speed Grade *(4)*	-4 Speed Grade	-5 Speed Grade	Unit
SSTL-2 Class I	8 mA	t_{OP}	957	1715	1799	1913	2041	ps
		t_{DIP}	979	1781	1869	1987	2131	ps
	12 mA *(1)*	t_{OP}	940	1672	1754	1865	1991	ps
		t_{DIP}	962	1738	1824	1939	2081	ps
SSTL-2 Class II	16 mA	t_{OP}	918	1609	1688	1795	1918	ps
		t_{DIP}	940	1675	1758	1869	2008	ps
	20 mA	t_{OP}	919	1598	1676	1783	1905	ps
		t_{DIP}	941	1664	1746	1857	1995	ps
	24 mA *(1)*	t_{OP}	915	1596	1674	1781	1903	ps
		t_{DIP}	937	1662	1744	1855	1993	ps
SSTL-18 Class I	4 mA	t_{OP}	953	1690	1773	1886	2012	ps
		t_{DIP}	975	1756	1843	1960	2102	ps
	6 mA	t_{OP}	958	1656	1737	1848	1973	ps
		t_{DIP}	980	1722	1807	1922	2063	ps
	8 mA	t_{OP}	937	1640	1721	1830	1954	ps
		t_{DIP}	959	1706	1791	1904	2044	ps
	10 mA	t_{OP}	942	1638	1718	1827	1952	ps
		t_{DIP}	964	1704	1788	1901	2042	ps
	12 mA *(1)*	t_{OP}	936	1626	1706	1814	1938	ps
		t_{DIP}	958	1692	1776	1888	2028	ps
SSTL-18 Class II	8 mA	t_{OP}	925	1597	1675	1782	1904	ps
		t_{DIP}	947	1663	1745	1856	1994	ps
	16 mA	t_{OP}	937	1578	1655	1761	1882	ps
		t_{DIP}	959	1644	1725	1835	1972	ps
	18 mA	t_{OP}	933	1585	1663	1768	1890	ps
		t_{DIP}	955	1651	1733	1842	1980	ps
	20 mA *(1)*	t_{OP}	933	1583	1661	1766	1888	ps
		t_{DIP}	955	1649	1731	1840	1978	ps

Table 4–86. Stratix II GX I/O Output Delay for Column Pins (Part 4 of 7)								
I/O Standard	**Drive Strength**	**Parameter**	**Fast Corner Industrial/ Commercial**	**-3 Speed Grade** *(3)*	**-3 Speed Grade** *(4)*	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
1.8-V HSTL Class I	4 mA	t_{OP}	956	1608	1687	1794	1943	ps
		t_{DIP}	978	1674	1757	1868	2033	ps
	6 mA	t_{OP}	962	1595	1673	1779	1928	ps
		t_{DIP}	984	1661	1743	1853	2018	ps
	8 mA	t_{OP}	940	1586	1664	1769	1917	ps
		t_{DIP}	962	1652	1734	1843	2007	ps
	10 mA	t_{OP}	944	1591	1669	1775	1923	ps
		t_{DIP}	966	1657	1739	1849	2013	ps
	12 mA *(1)*	t_{OP}	936	1585	1663	1768	1916	ps
		t_{DIP}	958	1651	1733	1842	2006	ps
1.8-V HSTL Class II	16 mA	t_{OP}	919	1385	1453	1545	1680	ps
		t_{DIP}	941	1451	1523	1619	1770	ps
	18 mA	t_{OP}	921	1394	1462	1555	1691	ps
		t_{DIP}	943	1460	1532	1629	1781	ps
	20 mA *(1)*	t_{OP}	921	1402	1471	1564	1700	ps
		t_{DIP}	943	1468	1541	1638	1790	ps
1.5-V HSTL Class I	4 mA	t_{OP}	956	1607	1686	1793	1942	ps
		t_{DIP}	978	1673	1756	1867	2032	ps
	6 mA	t_{OP}	961	1588	1666	1772	1920	ps
		t_{DIP}	983	1654	1736	1846	2010	ps
	8 mA	t_{OP}	943	1590	1668	1774	1922	ps
		t_{DIP}	965	1656	1738	1848	2012	ps
	10 mA	t_{OP}	943	1592	1670	1776	1924	ps
		t_{DIP}	965	1658	1740	1850	2014	ps
	12 mA *(1)*	t_{OP}	937	1590	1668	1774	1922	ps
		t_{DIP}	959	1656	1738	1848	2012	ps

Table 4–86. Stratix II GX I/O Output Delay for Column Pins (Part 5 of 7)

I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(3)*	-3 Speed Grade *(4)*	-4 Speed Grade	-5 Speed Grade	Unit
1.5-V HSTL Class II	16 mA	t_{OP}	924	1431	1501	1596	1734	ps
		t_{DIP}	946	1497	1571	1670	1824	ps
	18 mA	t_{OP}	927	1439	1510	1605	1744	ps
		t_{DIP}	949	1505	1580	1679	1834	ps
	20 mA *(1)*	t_{OP}	929	1450	1521	1618	1757	ps
		t_{DIP}	951	1516	1591	1692	1847	ps
PCI	-	t_{OP}	1082	1956	2051	2176	2070	ps
		t_{DIP}	1104	2022	2121	2250	2160	ps
PCI-X	-	t_{OP}	1082	1956	2051	2176	2070	ps
		t_{DIP}	1104	2022	2121	2250	2160	ps
Differential SSTL-2 Class I *(2)*	8 mA	t_{OP}	957	1715	1799	1913	2041	ps
		t_{DIP}	979	1781	1869	1987	2131	ps
	12 mA	t_{OP}	940	1672	1754	1865	1991	ps
		t_{DIP}	962	1738	1824	1939	2081	ps
Differential SSTL-2 Class II *(2)*	16 mA	t_{OP}	918	1609	1688	1795	1918	ps
		t_{DIP}	940	1675	1758	1869	2008	ps
	20 mA	t_{OP}	919	1598	1676	1783	1905	ps
		t_{DIP}	941	1664	1746	1857	1995	ps
	24 mA	t_{OP}	915	1596	1674	1781	1903	ps
		t_{DIP}	937	1662	1744	1855	1993	ps
Differential SSTL-18 Class I *(2)*	4 mA	t_{OP}	953	1690	1773	1886	2012	ps
		t_{DIP}	975	1756	1843	1960	2102	ps
	6 mA	t_{OP}	958	1656	1737	1848	1973	ps
		t_{DIP}	980	1722	1807	1922	2063	ps
	8 mA	t_{OP}	937	1640	1721	1830	1954	ps
		t_{DIP}	959	1706	1791	1904	2044	ps
	10 mA	t_{OP}	942	1638	1718	1827	1952	ps
		t_{DIP}	964	1704	1788	1901	2042	ps
	12 mA	t_{OP}	936	1626	1706	1814	1938	ps
		t_{DIP}	958	1692	1776	1888	2028	ps

Table 4–86. Stratix II GX I/O Output Delay for Column Pins (Part 6 of 7)								
I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(3)*	-3 Speed Grade *(4)*	-4 Speed Grade	-5 Speed Grade	Unit
Differential SSTL-18 Class II *(2)*	8 mA	t_{OP}	925	1597	1675	1782	1904	ps
		t_{DIP}	947	1663	1745	1856	1994	ps
	16 mA	t_{OP}	937	1578	1655	1761	1882	ps
		t_{DIP}	959	1644	1725	1835	1972	ps
	18 mA	t_{OP}	933	1585	1663	1768	1890	ps
		t_{DIP}	955	1651	1733	1842	1980	ps
	20 mA	t_{OP}	933	1583	1661	1766	1888	ps
		t_{DIP}	955	1649	1731	1840	1978	ps
1.8-V differential HSTL Class I *(2)*	4 mA	t_{OP}	956	1608	1687	1794	1943	ps
		t_{DIP}	978	1674	1757	1868	2033	ps
	6 mA	t_{OP}	962	1595	1673	1779	1928	ps
		t_{DIP}	984	1661	1743	1853	2018	ps
	8 mA	t_{OP}	940	1586	1664	1769	1917	ps
		t_{DIP}	962	1652	1734	1843	2007	ps
	10 mA	t_{OP}	944	1591	1669	1775	1923	ps
		t_{DIP}	966	1657	1739	1849	2013	ps
	12 mA	t_{OP}	936	1585	1663	1768	1916	ps
		t_{DIP}	958	1651	1733	1842	2006	ps
1.8-V differential HSTL Class II *(2)*	16 mA	t_{OP}	919	1385	1453	1545	1680	ps
		t_{DIP}	941	1451	1523	1619	1770	ps
	18 mA	t_{OP}	921	1394	1462	1555	1691	ps
		t_{DIP}	943	1460	1532	1629	1781	ps
	20 mA	t_{OP}	921	1402	1471	1564	1700	ps
		t_{DIP}	943	1468	1541	1638	1790	ps

Table 4–86. Stratix II GX I/O Output Delay for Column Pins (Part 7 of 7)								
I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade (3)	-3 Speed Grade (4)	-4 Speed Grade	-5 Speed Grade	Unit
1.5-V differential HSTL Class I (2)	4 mA	t_{OP}	956	1607	1686	1793	1942	ps
		t_{DIP}	978	1673	1756	1867	2032	ps
	6 mA	t_{OP}	961	1588	1666	1772	1920	ps
		t_{DIP}	983	1654	1736	1846	2010	ps
	8 mA	t_{OP}	943	1590	1668	1774	1922	ps
		t_{DIP}	965	1656	1738	1848	2012	ps
	10 mA	t_{OP}	943	1592	1670	1776	1924	ps
		t_{DIP}	965	1658	1740	1850	2014	ps
	12 mA	t_{OP}	937	1590	1668	1774	1922	ps
		t_{DIP}	959	1656	1738	1848	2012	ps
1.5-V differential HSTL Class II (2)	16 mA	t_{OP}	924	1431	1501	1596	1734	ps
		t_{DIP}	946	1497	1571	1670	1824	ps
	18 mA	t_{OP}	927	1439	1510	1605	1744	ps
		t_{DIP}	949	1505	1580	1679	1834	ps
	20 mA	t_{OP}	929	1450	1521	1618	1757	ps
		t_{DIP}	951	1516	1591	1692	1847	ps

(1) This is the default setting in the Quartus II software.
(2) These I/O standards are only supported on DQS pins.
(3) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(4) This column refers to –3 speed grades for EP2SGX130 devices.

Table 4–87. Stratix II GX I/O Output Delay for Row Pins (Part 1 of 4)								
I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade (3)	-3 Speed Grade (4)	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	4 mA	t_{OP}	1328	2655	2786	2962	3189	ps
		t_{DIP}	1285	2600	2729	2902	3116	ps
	8 mA	t_{OP}	1200	2113	2217	2357	2549	ps
		t_{DIP}	1157	2058	2160	2297	2476	ps
	12 mA (1)	t_{OP}	1144	2081	2184	2321	2512	ps
		t_{DIP}	1101	2026	2127	2261	2439	ps

Table 4–87. Stratix II GX I/O Output Delay for Row Pins (Part 2 of 4)								
I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade (3)	-3 Speed Grade (4)	-4 Speed Grade	-5 Speed Grade	Unit
LVCMOS	4 mA	t_{OP}	1200	2113	2217	2357	2549	ps
		t_{DIP}	1157	2058	2160	2297	2476	ps
	8 mA (1)	t_{OP}	1094	1853	1944	2067	2243	ps
		t_{DIP}	1051	1798	1887	2007	2170	ps
	12 mA (1)	t_{OP}	1061	1723	1808	1922	2089	ps
		t_{DIP}	1018	1668	1751	1862	2016	ps
2.5 V	4 mA	t_{OP}	1183	2091	2194	2332	2523	ps
		t_{DIP}	1140	2036	2137	2272	2450	ps
	8 mA	t_{OP}	1080	1872	1964	2088	2265	ps
		t_{DIP}	1037	1817	1907	2028	2192	ps
	12 mA (1)	t_{OP}	1061	1775	1862	1980	2151	ps
		t_{DIP}	1018	1720	1805	1920	2078	ps
1.8 V	2 mA	t_{OP}	1253	2954	3100	3296	3542	ps
		t_{DIP}	1210	2899	3043	3236	3469	ps
	4 mA	t_{OP}	1242	2294	2407	2559	2763	ps
		t_{DIP}	1199	2239	2350	2499	2690	ps
	6 mA	t_{OP}	1131	2039	2140	2274	2462	ps
		t_{DIP}	1088	1984	2083	2214	2389	ps
	8 mA (1)	t_{OP}	1100	1942	2038	2166	2348	ps
		t_{DIP}	1057	1887	1981	2106	2275	ps
1.5 V	2 mA	t_{OP}	1213	2530	2655	2823	3041	ps
		t_{DIP}	1170	2475	2598	2763	2968	ps
	4 mA (1)	t_{OP}	1106	2020	2120	2253	2440	ps
		t_{DIP}	1063	1965	2063	2193	2367	ps
SSTL-2 Class I	8 mA	t_{OP}	1050	1759	1846	1962	2104	ps
		t_{DIP}	1007	1704	1789	1902	2031	ps
	12 mA (1)	t_{OP}	1026	1694	1777	1889	2028	ps
		t_{DIP}	983	1639	1720	1829	1955	ps
SSTL-2 Class II	16 mA (1)	t_{OP}	992	1581	1659	1763	1897	ps
		t_{DIP}	949	1526	1602	1703	1824	ps

Table 4–87. Stratix II GX I/O Output Delay for Row Pins (Part 3 of 4)

I/O Standard	Drive Strength	Parameter	Fast Corner Industrial/ Commercial	-3 Speed Grade *(3)*	-3 Speed Grade *(4)*	-4 Speed Grade	-5 Speed Grade	Unit
SSTL-18 Class I	4 mA	t_{OP}	1038	1709	1793	1906	2046	ps
		t_{DIP}	995	1654	1736	1846	1973	ps
	6 mA	t_{OP}	1042	1648	1729	1838	1975	ps
		t_{DIP}	999	1593	1672	1778	1902	ps
	8 mA	t_{OP}	1018	1633	1713	1821	1958	ps
		t_{DIP}	975	1578	1656	1761	1885	ps
	10 mA *(1)*	t_{OP}	1021	1615	1694	1801	1937	ps
		t_{DIP}	978	1560	1637	1741	1864	ps
1.8-V HSTL Class I	4 mA	t_{OP}	1019	1610	1689	1795	1956	ps
		t_{DIP}	976	1555	1632	1735	1883	ps
	6 mA	t_{OP}	1022	1580	1658	1762	1920	ps
		t_{DIP}	979	1525	1601	1702	1847	ps
	8 mA	t_{OP}	1004	1576	1653	1757	1916	ps
		t_{DIP}	961	1521	1596	1697	1843	ps
	10 mA	t_{OP}	1008	1567	1644	1747	1905	ps
		t_{DIP}	965	1512	1587	1687	1832	ps
	12 mA *(1)*	t_{OP}	999	1566	1643	1746	1904	ps
		t_{DIP}	956	1511	1586	1686	1831	ps
1.5-V HSTL Class I	4 mA	t_{OP}	1018	1591	1669	1774	1933	ps
		t_{DIP}	975	1536	1612	1714	1860	ps
	6 mA	t_{OP}	1021	1579	1657	1761	1919	ps
		t_{DIP}	978	1524	1600	1701	1846	ps
	8 mA *(1)*	t_{OP}	1006	1572	1649	1753	1911	ps
		t_{DIP}	963	1517	1592	1693	1838	ps
Differential SSTL-2 Class I	8 mA	t_{OP}	1050	1759	1846	1962	2104	ps
		t_{DIP}	1007	1704	1789	1902	2031	ps
	12 mA	t_{OP}	1026	1694	1777	1889	2028	ps
		t_{DIP}	983	1639	1720	1829	1955	ps
Differential SSTL-2 Class II	16 mA	t_{OP}	992	1581	1659	1763	1897	ps
		t_{DIP}	949	1526	1602	1703	1824	ps

Table 4–87. Stratix II GX I/O Output Delay for Row Pins (Part 4 of 4)								
I/O Standard	**Drive Strength**	**Parameter**	**Fast Corner Industrial/ Commercial**	**-3 Speed Grade** *(3)*	**-3 Speed Grade** *(4)*	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
Differential SSTL-18 Class I	4 mA	t_{OP}	1038	1709	1793	1906	2046	ps
		t_{DIP}	995	1654	1736	1846	1973	ps
	6 mA	t_{OP}	1042	1648	1729	1838	1975	ps
		t_{DIP}	999	1593	1672	1778	1902	ps
	8 mA	t_{OP}	1018	1633	1713	1821	1958	ps
		t_{DIP}	975	1578	1656	1761	1885	ps
	10 mA	t_{OP}	1021	1615	1694	1801	1937	ps
		t_{DIP}	978	1560	1637	1741	1864	ps
LVDS *(2)*	-	t_{OP}	1067	1723	1808	1922	2089	ps
		t_{DIP}	1024	1668	1751	1862	2016	ps
HyperTransport	-	t_{OP}	1053	1723	1808	1922	2089	ps
		t_{DIP}	1010	1668	1751	1862	2016	ps

(1) This is the default setting in the Quartus II software.
(2) The parameters are only available on the left side of the device.
(3) This column refers to –3 speed grades for EP2SGX30, EP2SGX60, and EP2SGX90 devices.
(4) This column refers to –3 speed grades for EP2SGX130 devices.

Maximum Input and Output Clock Toggle Rate

Maximum clock toggle rate is defined as the maximum frequency achievable for a clock type signal at an I/O pin. The I/O pin can be a regular I/O pin or a dedicated clock I/O pin.

The maximum clock toggle rate is different from the maximum data bit rate. If the maximum clock toggle rate on a regular I/O pin is 300 MHz, the maximum data bit rate for dual data rate (DDR) could be potentially as high as 600 Mbps on the same I/O pin.

Tables 4–88 through 4–90 specify the maximum input clock toggle rates. Tables 4–91 through 4–96 specify the maximum output clock toggle rates at 0 pF load. Table 4–97 specifies the derating factors for the output clock toggle rate for a non 0 pF load.

To calculate the output toggle rate for a non 0 pF load, use this formula:

The toggle rate for a non 0 pF load

= 1,000 / (1,000/ toggle rate at 0 pF load + derating factor × load value in pF /1,000)

For example, the output toggle rate at 0 pF load for SSTL-18 Class II 20 mA I/O standard is 550 MHz on a -3 device clock output pin. The derating factor is 94 ps/pF. For a 10 pF load the toggle rate is calculated as:

1,000 / (1,000/550 + 94 × 10 /1,000) = 363 (MHz)

Table 4–88 shows the maximum input clock toggle rates for Stratix II GX device column pins.

Table 4–88. Stratix II GX Maximum Input Clock Rate for Column I/O Pins (Part 1 of 2)

I/O Standard	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	500	500	450	MHz
2.5 V	500	500	450	MHz
1.8 V	500	500	450	MHz
1.5 V	500	500	450	MHz
LVCMOS	500	500	450	MHz
SSTL-2 Class I	500	500	500	MHz
SSTL-2 Class II	500	500	500	MHz
SSTL-18 Class I	500	500	500	MHz
SSTL-18 Class I I	500	500	500	MHz
1.5-V HSTL Class I	500	500	500	MHz
1.5-V HSTL Class I I	500	500	500	MHz
1.8-V HSTL Class I	500	500	500	MHz
1.8-V HSTL Class II	500	500	500	MHz
PCI	500	500	450	MHz
PCI-X	500	500	450	MHz
Differential SSTL-2 Class I	500	500	500	MHz
Differential SSTL-2 Class II	500	500	500	MHz
Differential SSTL-18 Class I	500	500	500	MHz

Table 4–88. Stratix II GX Maximum Input Clock Rate for Column I/O Pins (Part 2 of 2)				
I/O Standard	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
Differential SSTL-18 Class I I	500	500	500	MHz
1.8-V differential HSTL Class I	500	500	500	MHz
1.8-V differential HSTL Class II	500	500	500	MHz
1.5-V differential HSTL Class I	500	500	500	MHz
1.5-V differential HSTL Class I I	500	500	500	MHz
1.2-V HSTL	280	250	250	MHz
1.2-V differential HSTL	280	250	250	MHz

Table 4–89 shows the maximum input clock toggle rates for Stratix II GX device row pins.

Table 4–89. Stratix II GX Maximum Input Clock Rate for Row I/O Pins (Part 1 of 2)				
I/O Standard	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	500	500	450	MHz
2.5 V	500	500	450	MHz
1.8 V	500	500	450	MHz
1.5 V	500	500	450	MHz
LVCMOS	500	500	450	MHz
SSTL-2 Class I	500	500	500	MHz
SSTL-2 Class II	500	500	500	MHz
SSTL-18 Class I	500	500	500	MHz
SSTL-18 Class II	500	500	500	MHz
1.5-V HSTL Class I	500	500	500	MHz
1.5-V HSTL Class II	500	500	500	MHz
1.8-V HSTL Class I	500	500	500	MHz
1.8-V HSTL Class II	500	500	500	MHz
PCI	500	500	425	MHz
PCI-X	500	500	425	MHz
Differential SSTL-2 Class I	500	500	500	MHz

Table 4–89. Stratix II GX Maximum Input Clock Rate for Row I/O Pins (Part 2 of 2)				
I/O Standard	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
Differential SSTL-2 Class II	500	500	500	MHz
Differential SSTL-18 Class I	500	500	500	MHz
Differential SSTL-18 Class I I	500	500	500	MHz
1.8-V differential HSTL Class I	500	500	500	MHz
1.8-V differential HSTL Class I I	500	500	500	MHz
1.5-V differential HSTL Class I	500	500	500	MHz
1.5-V differential HSTL Class II	500	500	500	MHz
LVDS *(1)*	520	520	420	MHz
HyperTransport	520	520	420	MHz

(1) The parameters are only available on the left side of the device.

Table 4–90 shows the maximum input clock toggle rates for Stratix II GX device dedicated clock pins.

Table 4–90. Stratix II GX Maximum Input Clock Rate for Dedicated Clock Pins (Part 1 of 2)				
I/O Standard	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
LVTTL	500	500	400	MHz
2.5 V	500	500	400	MHz
1.8 V	500	500	400	MHz
1.5 V	500	500	400	MHz
LVCMOS	500	500	400	MHz
SSTL-2 Class I	500	500	500	MHz
SSTL-2 Class II	500	500	500	MHz
SSTL-18 Class I	500	500	500	MHz
SSTL-18 Class II	500	500	500	MHz
1.5-V HSTL Class I	500	500	500	MHz
1.5-V HSTL Class II	500	500	500	MHz
1.8-V HSTL CLass I	500	500	500	MHz

Table 4–90. Stratix II GX Maximum Input Clock Rate for Dedicated Clock Pins (Part 2 of 2)				
I/O Standard	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
1.8-V HSTL CLass I	500	500	500	MHz
PCI	500	500	400	MHz
PCI-X	500	500	400	MHz
Differential SSTL-2 Class I	500	500	500	MHz
Differential SSTL-2 Class II	500	500	500	MHz
Differential SSTL-18 Class I	500	500	500	MHz
Differential SSTL-18 Class II	500	500	500	MHz
1.8-V differential HSTL Class I	500	500	500	MHz
1.8-V differential HSTL Class II	500	500	500	MHz
1.5-V differential HSTL Class I	500	500	500	MHz
1.5-V differential HSTL Class I I	500	500	500	MHz
HyperTransport *(1)*	717	717	640	MHz
	450	450	400	MHz
LVPECL *(1)*, *(2)*	717	717	640	MHz
	450	450	400	MHz
LVDS *(1)*	717	717	640	MHz
	450	450	400	MHz

(1) The first set of numbers refers to the HIO dedicated clock pins. The second set of numbers refers to the VIO dedicated clock pins.
(2) LVPECL is only supported on column clock pins.

Table 4–91 shows the maximum output clock toggle rates for Stratix II GX device column pins.

Table 4–91. Stratix II GX Maximum Output Clock Rate for Column Pins (Part 1 of 3)

I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	4 mA	270	225	210	MHz
	8 mA	435	355	325	MHz
	12 mA	580	475	420	MHz
	16 mA	720	594	520	MHz
	20 mA	875	700	610	MHz
	24 mA *(1)*	1030	794	670	MHz
LVCMOS	4 mA	290	250	230	MHz
	8 mA	565	480	440	MHz
	12 mA	790	710	670	MHz
	16 mA	1020	925	875	MHz
	20 mA	1066	985	935	MHz
	24 mA *(1)*	1100	1040	1000	MHz
2.5 V	4 mA	230	194	180	MHz
	8 mA	430	380	380	MHz
	12 mA	630	575	550	MHz
	16 mA *(1)*	930	845	820	MHz
1.8 V	2 mA	120	109	104	MHz
	4 mA	285	250	230	MHz
	6 mA	450	390	360	MHz
	8 mA	660	570	520	MHz
	10 mA	905	805	755	MHz
	12 mA *(1)*	1131	1040	990	MHz
1.5 V	2 mA	244	200	180	MHz
	4 mA	470	370	325	MHz
	6 mA	550	430	375	MHz
	8 mA *(1)*	625	495	420	MHz
SSTL-2 Class I	8 mA	400	300	300	MHz
	12 mA *(1)*	400	400	350	MHz
SSTL-2 Class II	16 mA	350	350	300	MHz
	20 mA	400	350	350	MHz
	24 mA *(1)*	400	400	350	MHz

Table 4–91. Stratix II GX Maximum Output Clock Rate for Column Pins (Part 2 of 3)					
I/O Standard	**Drive Strength**	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
SSTL-18 Class I	4 mA	200	150	150	MHz
	6 mA	350	250	200	MHz
	8 mA	450	300	300	MHz
	10 mA	500	400	400	MHz
	12 mA *(1)*	700	550	400	MHz
SSTL-18 Class II	8 mA	200	200	150	MHz
	16 mA	400	350	350	MHz
	18 mA	450	400	400	MHz
	20 mA *(1)*	550	500	450	MHz
1.8-V HSTL Class I	4 mA	300	300	300	MHz
	6 mA	500	450	450	MHz
	8 mA	650	600	600	MHz
	10 mA	700	650	600	MHz
	12 mA *(1)*	700	700	650	MHz
1.8-V HSTL Class II	16 mA	500	500	450	MHz
	18 mA	550	500	500	MHz
	20 mA *(1)*	650	550	550	MHz
1.5-V HSTL Class I	4 mA	350	300	300	MHz
	6 mA	500	500	450	MHz
	8 mA	700	650	600	MHz
	10 mA	700	700	650	MHz
	12 mA *(1)*	700	700	700	MHz
1.5-V HSTL Class II	16 mA	600	600	550	MHz
	18 mA	650	600	600	MHz
	20 mA *(1)*	700	650	600	MHz
PCI	-	1000	790	670	MHz
PCI-X	-	1000	790	670	MHz
Differential SSTL-2 Class I	8 mA	400	300	300	MHz
	12 mA	400	400	350	MHz
Differential SSTL-2 Class II	16 mA	350	350	300	MHz
	20 mA	400	350	350	MHz
	24 mA	400	400	350	MHz

Table 4–91. Stratix II GX Maximum Output Clock Rate for Column Pins (Part 3 of 3)					
I/O Standard	**Drive Strength**	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
Differential SSTL-18 Class I	4 mA	200	150	150	MHz
	6 mA	350	250	200	MHz
	8 mA	450	300	300	MHz
	10 mA	500	400	400	MHz
	12 mA	700	550	400	MHz
Differential SSTL-18 Class II	8 mA	200	200	150	MHz
	16 mA	400	350	350	MHz
	18 mA	450	400	400	MHz
	20 mA	550	500	450	MHz
1.8-V HSTL differential Class I	4 mA	300	300	300	MHz
	6 mA	500	450	450	MHz
	8 mA	650	600	600	MHz
	10 mA	700	650	600	MHz
	12 mA	700	700	650	MHz
1.8-V HSTL differential Class II	16 mA	500	500	450	MHz
	18 mA	550	500	500	MHz
	20 mA	650	550	550	MHz
1.5-V HSTL differential Class I	4 mA	350	300	300	MHz
	6 mA	500	500	450	MHz
	8 mA	700	650	600	MHz
	10 mA	700	700	650	MHz
	12 mA	700	700	700	MHz
1.5-V HSTL differential Class II	16 mA	600	600	550	MHz
	18 mA	650	600	600	MHz
	20 mA	700	650	600	MHz

(1) This is the default setting in the Quartus II software.

Table 4–92 shows the maximum output clock toggle rates for Stratix II GX device row pins.

Table 4–92. Stratix II GX Maximum Output Clock Rate for Row Pins (Part 1 of 2)

I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	4 mA	270	225	210	MHz
	8 mA	435	355	325	MHz
	12 mA *(1)*	580	475	420	MHz
LVCMOS	4 mA	290	250	230	MHz
	8 mA	565	480	440	MHz
	12 mA *(1)*	350	350	297	MHz
2.5 V	4 mA	230	194	180	MHz
	8 mA	430	380	380	MHz
	12 mA *(1)*	630	575	550	MHz
1.8 V	2 mA	120	109	104	MHz
	4 mA	285	250	230	MHz
	6 mA	450	390	360	MHz
	8 mA *(1)*	660	570	520	MHz
1.5 V	2 mA	244	200	180	MHz
	4 mA *(1)*	470	370	325	MHz
SSTL-2 Class I	8 mA	400	300	300	MHz
	12 mA *(1)*	400	400	350	MHz
SSTL-2 Class II	16 mA	350	350	300	MHz
	20 mA *(1)*	350	350	297	MHz
SSTL-18 Class I	4 mA	200	150	150	MHz
	6 mA	350	250	200	MHz
	8 mA	450	300	300	MHz
	10 mA	500	400	400	MHz
	12 mA *(1)*	350	350	297	MHz
1.8-V HSTL Class I	4 mA	300	300	300	MHz
	6 mA	500	450	450	MHz
	8 mA	650	600	600	MHz
	10 mA	700	650	600	MHz
	12 mA *(1)*	700	700	650	MHz
1.5-V HSTL Class I	4 mA	350	300	300	MHz
	6 mA	500	500	450	MHz
	8 mA *(1)*	700	650	600	MHz

Table 4–92. Stratix II GX Maximum Output Clock Rate for Row Pins (Part 2 of 2)					
I/O Standard	**Drive Strength**	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
Differential SSTL-2 Class I	8 mA	400	300	300	MHz
	12 mA	400	400	350	MHz
Differential SSTL-2 Class II	16 mA *(1)*	350	350	300	MHz
Differential SSTL-18 Class I	4 mA	200	150	150	MHz
	6 mA	350	250	200	MHz
	8 mA	450	300	300	MHz
	10 mA *(1)*	500	400	400	MHz
LVDS	-	717	717	640	MHz
HyperTransport	-	717	717	640	MHz

(1) This is the default setting in Quartus II software.

Table 4–93 shows the maximum output clock toggle rate for Stratix II GX device dedicated clock pins.

Table 4–93. Stratix II GX Maximum Output Clock Rate for Dedicated Clock Pins (Part 1 of 4)					
I/O Standard	**Drive Strength**	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
LVTTL	4 mA	270	225	210	MHz
	8 mA	435	355	325	MHz
	12 mA	580	475	420	MHz
	16 mA	720	594	520	MHz
	20 mA	875	700	610	MHz
	24 mA *(1)*	1030	794	670	MHz
LVCMOS	4 mA	290	250	230	MHz
	8 mA	565	480	440	MHz
	12 mA	790	710	670	MHz
	16 mA	1020	925	875	MHz
	20 mA	1066	985	935	MHz
	24 mA *(1)*	1100	1040	1000	MHz

Table 4–93. Stratix II GX Maximum Output Clock Rate for Dedicated Clock Pins (Part 2 of 4)					
I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
2.5 V	4 mA	230	194	180	MHz
	8 mA	430	380	380	MHz
	12 mA	630	575	550	MHz
	16 mA *(1)*	930	845	820	MHz
1.8 V	2 mA	120	109	104	MHz
	4 mA	285	250	230	MHz
	6 mA	450	390	360	MHz
	8 mA	660	570	520	MHz
	10 mA	905	805	755	MHz
	12 mA *(1)*	1131	1040	990	MHz
1.5 V	2 mA	244	200	180	MHz
	4 mA	470	370	325	MHz
	6 mA	550	430	375	MHz
	8 mA *(1)*	625	495	420	MHz
SSTL-2 Class I	8 mA	400	300	300	MHz
	12 mA *(1)*	400	400	350	MHz
SSTL-2 Class II	16 mA	350	350	300	MHz
	20 mA	400	350	350	MHz
	24 mA *(1)*	400	400	350	MHz
SSTL-18 Class I	4 mA	200	150	150	MHz
	6 mA	350	250	200	MHz
	8 mA	450	300	300	MHz
	10 mA	500	400	400	MHz
	12 mA *(1)*	650	550	400	MHz
SSTL-18 Class II	8 mA	200	200	150	MHz
	16 mA	400	350	350	MHz
	18 mA	450	400	400	MHz
	20 mA *(1)*	550	500	450	MHz
1.8-V HSTL Class I	4 mA	300	300	300	MHz
	6 mA	500	450	450	MHz
	8 mA	650	600	600	MHz
	10 mA	700	650	600	MHz
	12 mA *(1)*	700	700	650	MHz

Table 4–93. Stratix II GX Maximum Output Clock Rate for Dedicated Clock Pins (Part 3 of 4)					
I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
1.8-V HSTL Class II	16 mA	500	500	450	MHz
	18 mA	550	500	500	MHz
	20 mA *(1)*	550	550	550	MHz
1.5-V HSTL Class I	4 mA	350	300	300	MHz
	6 mA	500	500	450	MHz
	8 mA	700	650	600	MHz
	10 mA	700	700	650	MHz
	12 mA *(1)*	700	700	700	MHz
1.5-V HSTL Class II	16 mA	600	600	550	MHz
	18 mA	650	600	600	MHz
	20 mA *(1)*	700	650	600	MHz
PCI	-	1000	790	670	MHz
PCI-X	-	1000	790	670	MHz
Differential SSTL-2 Class I	8 mA	400	300	300	MHz
	12 mA	400	400	350	MHz
Differential SSTL-2 Class II	16 mA	350	350	300	MHz
	20 mA	400	350	350	MHz
	24 mA	400	400	350	MHz
Differential SSTL-18 Class I	4 mA	200	150	150	MHz
	6 mA	350	250	200	MHz
	8 mA	450	300	300	MHz
	10 mA	500	400	400	MHz
	12 mA	650	550	400	MHz
Differential SSTL-18 Class II	8 mA	200	200	150	MHz
	16 mA	400	350	350	MHz
	18 mA	450	400	400	MHz
	20 mA	550	500	450	MHz
1.8-V differential Class I	4 mA	300	300	300	MHz
	6 mA	500	450	450	MHz
	8 mA	650	600	600	MHz
	10 mA	700	650	600	MHz
	12 mA	700	700	650	MHz

Table 4–93. Stratix II GX Maximum Output Clock Rate for Dedicated Clock Pins (Part 4 of 4)					
I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
1.8-V differential Class II	16 mA	500	500	450	MHz
	18 mA	550	500	500	MHz
	20 mA	550	550	550	MHz
1.5-V differential Class I	4 mA	350	300	300	MHz
	6 mA	500	500	450	MHz
	8 mA	700	650	600	MHz
	10 mA	700	700	650	MHz
	12 mA	700	700	700	MHz
1.5-V differential Class II	16 mA	600	600	550	MHz
	18 mA	650	600	600	MHz
	20 mA	700	650	600	MHz
HyperTransport	-	300	250	125	MHz
LVPECL	-	450	400	300	MHz

(1) This is the default setting in Quartus II software.

Table 4–94 shows the maximum output clock toggle rate for Stratix II GX device series-terminated column pins.

Table 4–94. Stratix II GX Maximum Output Clock Rate for Column Pins (Series Termination) (Part 1 of 2)					
I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	OCT_25_OHMS	400	400	350	MHz
	OCT_50_OHMS	400	400	350	MHz
LVCMOS	OCT_25_OHMS	350	350	300	MHz
	OCT_50_OHMS	350	350	300	MHz
2.5 V	OCT_25_OHMS	350	350	300	MHz
	OCT_50_OHMS	350	350	300	MHz
1.8 V	OCT_25_OHMS	700	550	450	MHz
	OCT_50_OHMS	700	550	450	MHz
1.5 V	OCT_50_OHMS	550	450	400	MHz
SSTL-2 Class I	OCT_50_OHMS	600	500	500	MHz
SSTL-2 Class II	OCT_25_OHMS	600	550	500	MHz

Table 4–94. Stratix II GX Maximum Output Clock Rate for Column Pins (Series Termination) (Part 2 of 2)

I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
SSTL-18 Class I	OCT_50_OHMS	560	400	350	MHz
SSTL-18 Class II	OCT_25_OHMS	550	500	450	MHz
1.5-V HSTL Class I	OCT_50_OHMS	600	550	500	MHz
1.8-V HSTL Class I	OCT_50_OHMS	650	600	600	MHz
1.8-V HSTL Class II	OCT_25_OHMS	500	500	450	MHz
Differential SSTL-2 Class I	OCT_50_OHMS	600	500	500	MHz
Differential SSTL-2 Class II	OCT_25_OHMS	600	550	500	MHz
Differential SSTL-18 Class I	OCT_50_OHMS	560	400	350	MHz
Differential SSTL-18 Class II	OCT_25_OHMS	550	500	450	MHz
1.8-V differential HSTL Class I	OCT_50_OHMS	650	600	600	MHz
1.8-V differential HSTL Class II	OCT_25_OHMS	500	500	450	MHz
1.5-V differential HSTL Class I	OCT_50_OHMS	600	550	500	MHz

Table 4–95 shows the maximum output clock toggle rate for Stratix II GX device series-terminated row pins.

Table 4–95. Stratix II GX Maximum Output Clock Rate for Row Pins (Series Termination) (Part 1 of 2)

I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
LVTTL	OCT_25_OHMS	400	400	350	MHz
	OCT_50_OHMS	400	400	350	MHz
LVCMOS	OCT_25_OHMS	350	350	300	MHz
	OCT_50_OHMS	350	350	300	MHz
2.5 V	OCT_25_OHMS	350	350	300	MHz
	OCT_50_OHMS	350	350	300	MHz
1.8 V	OCT_50_OHMS	700	550	450	MHz
1.5 V	OCT_50_OHMS	550	450	400	MHz

Table 4–95. Stratix II GX Maximum Output Clock Rate for Row Pins (Series Termination) (Part 2 of 2)					
I/O Standard	**Drive Strength**	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
SSTL-2 Class I	OCT_50_OHMS	600	500	500	MHz
SSTL-2 Class II	OCT_25_OHMS	600	550	500	MHz
SSTL-18 Class I	OCT_50_OHMS	590	400	350	MHz
1.5-V HSTL Class I	OCT_50_OHMS	600	550	500	MHz
1.8-V HSTL Class I	OCT_50_OHMS	650	600	600	MHz
Differential SSTL-2 Class I	OCT_50_OHMS	600	500	500	MHz
Differential SSTL-2 Class II	OCT_25_OHMS	600	550	500	MHz
Differential SSTL-18 Class I	OCT_50_OHMS	590	400	350	MHz
Differential HSTL-18 Class I	OCT_50_OHMS	650	600	600	MHz
Differential HSTL-15 Class I	OCT_50_OHMS	600	550	500	

Table 4–96 shows the maximum output clock toggle rate for Stratix II GX device series-terminated dedicated clock pins.

Table 4–96. Stratix II GX Maximum Output Clock Rate for Dedicated Clock Pins (Series Termination) (Part 1 of 2)					
I/O Standard	**Drive Strength**	**-3 Speed Grade**	**-4 Speed Grade**	**-5 Speed Grade**	**Unit**
LVTTL	OCT_25_OHMS	400	400	350	MHz
	OCT_50_OHMS	400	400	350	MHz
LVCMOS	OCT_25_OHMS	350	350	300	MHz
	OCT_50_OHMS	350	350	300	MHz
2.5 V	OCT_25_OHMS	350	350	300	MHz
	OCT_50_OHMS	350	350	300	MHz
1.8 V	OCT_25_OHMS	700	550	450	MHz
	OCT_50_OHMS	700	550	450	MHz
1.5 V	OCT_50_OHMS	550	450	400	MHz
SSTL-2 Class I	OCT_50_OHMS	600	500	500	MHz
SSTL-2 Class II	OCT_25_OHMS	600	550	500	MHz
SSTL-18 Class I	OCT_50_OHMS	450	400	350	MHz

Table 4–96. Stratix II GX Maximum Output Clock Rate for Dedicated Clock Pins (Series Termination) (Part 2 of 2)

I/O Standard	Drive Strength	-3 Speed Grade	-4 Speed Grade	-5 Speed Grade	Unit
SSTL-18 Class II	OCT_25_OHMS	550	500	450	MHz
1.5-V HSTL Class I	OCT_50_OHMS	600	550	500	MHz
1.8-V HSTL Class I	OCT_50_OHMS	650	600	600	MHz
1.8-V HSTL Class II	OCT_25_OHMS	500	500	450	MHz
DIfferential SSTL-2 Class I	OCT_50_OHMS	600	500	500	MHz
DIfferential SSTL-2 Class II	OCT_25_OHMS	600	550	500	MHz
DIfferential SSTL-18 Class I	OCT_50_OHMS	560	400	350	MHz
DIfferential SSTL-18 Class II	OCT_25_OHMS	550	500	450	MHz
1.8-V differential HSTL Class I	OCT_50_OHMS	650	600	600	MHz
1.8-V differential HSTL Class II	OCT_25_OHMS	500	500	450	MHz
1.5-V differential HSTL Class I	OCT_50_OHMS	600	550	500	MHz

Table 4–97 specifies the derating factors for the output clock toggle rate for a non 0 pF load.

Table 4–97. Maximum Output Clock Toggle Rate Derating Factors (Part 1 of 5)

I/O Standard	Drive Strength	Maximum Output Clock Toggle Rate Derating Factors (ps/pF)								
		Column I/O Pins			Row I/O Pins			Dedicated Clock Outputs		
		-3	-4	-5	-3	-4	-5	-3	-4	-5
3.3-V LVTTL	4 mA	478	510	510	478	510	510	466	510	510
	8 mA	260	333	333	260	333	333	291	333	333
	12 mA	213	247	247	213	247	247	211	247	247
	16 mA	136	197	197	-	-	-	166	197	197
	20 mA	138	187	187	-	-	-	154	187	187
	24 mA	134	177	177	-	-	-	143	177	177

Table 4–97. Maximum Output Clock Toggle Rate Derating Factors (Part 2 of 5)

I/O Standard	Drive Strength	Maximum Output Clock Toggle Rate Derating Factors (ps/pF)								
		Column I/O Pins			Row I/O Pins			Dedicated Clock Outputs		
		-3	-4	-5	-3	-4	-5	-3	-4	-5
3.3-V LVCMOS	4 mA	377	391	391	377	391	391	377	391	391
	8 mA	206	212	212	206	212	212	178	212	212
	12 mA	141	145	145	-	-	-	115	145	145
	16 mA	108	111	111	-	-	-	86	111	111
	20 mA	83	88	88	-	-	-	79	88	88
	24 mA	65	72	72	-	-	-	74	72	72
2.5-V LVTTL/ LVCMOS	4 mA	387	427	427	387	427	427	391	427	427
	8 mA	163	224	224	163	224	224	170	224	224
	12 mA	142	203	203	142	203	203	152	203	203
	16 mA	120	182	182	-	-	-	134	182	182
1.8-V LVTTL/ LVCMOS	2 mA	951	1,421	1,421	951	1,421	1,421	904	1,421	1,421
	4 mA	405	516	516	405	516	516	393	516	516
	6 mA	261	325	325	261	325	325	253	325	325
	8 mA	223	274	274	223	274	274	224	274	274
	10 mA	194	236	236	-	-	-	199	236	236
	12 mA	174	209	209	-	-	-	180	209	209
1.5-V LVTTL/ LVCMOS	2 mA	652	963	963	652	963	963	618	963	963
	4 mA	333	347	347	333	347	347	270	347	347
	6 mA	182	247	247	-	-	-	198	247	247
	8 mA	135	194	194	-	-	-	155	194	194
SSTL-2 Class I	8 mA	364	680	680	364	680	680	350	680	680
	12 mA	163	207	207	163	207	207	188	207	207
SSTL-2 Class II	16 mA	118	147	147	118	147	147	94	147	147
	20 mA	99	122	122	-	-	-	87	122	122
	24 mA	91	116	116	-	-	-	85	116	116
SSTL-18 Class I	4 mA	458	570	570	458	570	570	505	570	570
	6 mA	305	380	380	305	380	380	336	380	380
	8 mA	225	282	282	225	282	282	248	282	282
	10 mA	167	220	220	167	220	220	190	220	220
	12 mA	129	175	175	-	-	-	148	175	175

Table 4–97. Maximum Output Clock Toggle Rate Derating Factors (Part 3 of 5)

I/O Standard	Drive Strength	Maximum Output Clock Toggle Rate Derating Factors (ps/pF)								
		Column I/O Pins			Row I/O Pins			Dedicated Clock Outputs		
		-3	-4	-5	-3	-4	-5	-3	-4	-5
SSTL-18 Class II	8 mA	173	206	206	-	-	-	155	206	206
	16 mA	150	160	160	-	-	-	140	160	160
	18 mA	120	130	130	-	-	-	110	130	130
	20 mA	109	127	127	-	-	-	94	127	127
2.5-V SSTL-2 Class I	8 mA	364	680	680	364	680	680	350	680	680
	12 mA	163	207	207	163	207	207	188	207	207
2.5-V SSTL-2 Class II	16 mA	118	147	147	118	147	147	94	147	147
	20 mA	99	122	122	-	-	-	87	122	122
	24 mA	91	116	116	-	-	-	85	116	116
1.8-V SSTL-18 Class I	4 mA	458	570	570	458	570	570	505	570	570
	6 mA	305	380	380	305	380	380	336	380	380
	8 mA	225	282	282	225	282	282	248	282	282
	10 mA	167	220	220	167	220	220	190	220	220
	12 mA	129	175	175	-	-	-	148	175	175
1.8-V SSTL-18 Class II	8 mA	173	206	206	-	-	-	155	206	206
	16 mA	150	160	160	-	-	-	140	160	160
	18 mA	120	130	130	-	-	-	110	130	130
	20 mA	109	127	127	-	-	-	94	127	127
1.8-V HSTL Class I	4 mA	245	282	282	245	282	282	229	282	282
	6 mA	164	188	188	164	188	188	153	188	188
	8 mA	123	140	140	123	140	140	114	140	140
	10 mA	110	124	124	110	124	124	108	124	124
	12 mA	97	110	110	97	110	110	104	110	110
1.8-V HSTL Class II	16 mA	101	104	104	-	-	-	99	104	104
	18 mA	98	102	102	-	-	-	93	102	102
	20 mA	93	99	99	-	-	-	88	99	99
1.5-V HSTL Class I	4 mA	168	196	196	168	196	196	188	196	196
	6 mA	112	131	131	112	131	131	125	131	131
	8 mA	84	99	99	84	99	99	95	99	99
	10 mA	87	98	98	-	-	-	90	98	98
	12 mA	86	98	98	-	-	-	87	98	98

Table 4–97. Maximum Output Clock Toggle Rate Derating Factors (Part 4 of 5)

I/O Standard	Drive Strength	Maximum Output Clock Toggle Rate Derating Factors (ps/pF)								
		Column I/O Pins			Row I/O Pins			Dedicated Clock Outputs		
		-3	-4	-5	-3	-4	-5	-3	-4	-5
1.5-V HSTL Class II	16 mA	95	101	101	-	-	-	96	101	101
	18 mA	95	100	100	-	-	-	101	100	100
	20 mA	94	101	101	-	-	-	104	101	101
2.5-V differential SSTL Class II *(3)*	8 mA	364	680	680	-	-	-	350	680	680
	12 mA	163	207	207	-	-	-	188	207	207
	16 mA	118	147	147	-	-	-	94	147	147
	20 mA	99	122	122	-	-	-	87	122	122
	24 mA	91	116	116	-	-	-	85	116	116
1.8-V differential SSTL Class I *(3)*	4 mA	458	570	570	-	-	-	505	570	570
	6 mA	305	380	380	-	-	-	336	380	380
	8 mA	225	282	282	-	-	-	248	282	282
	10 mA	167	220	220	-	-	-	190	220	220
	12 mA	129	175	175	-	-	-	148	175	175
1.8-V differential SSTL Class II *(3)*	8 mA	173	206	206	-	-	-	155	206	206
	16 mA	150	160	160	-	-	-	140	160	160
	18 mA	120	130	130	-	-	-	110	130	130
	20 mA	109	127	127	-	-	-	94	127	127
1.8-V differential HSTL Class I *(3)*	4 mA	245	282	282	-	-	-	229	282	282
	6 mA	164	188	188	-	-	-	153	188	188
	8 mA	123	140	140	-	-	-	114	140	140
	10 mA	110	124	124	-	-	-	108	124	124
	12 mA	97	110	110	-	-	-	104	110	110
1.8-V differential HSTL Class II *(3)*	16 mA	101	104	104	-	-	-	99	104	104
	18 mA	98	102	102	-	-	-	93	102	102
	20 mA	93	99	99	-	-	-	88	99	99
1.5-V differential HSTL Class I *(3)*	4 mA	168	196	196	-	-	-	188	196	196
	6 mA	112	131	131	-	-	-	125	131	131
	8 mA	84	99	99	-	-	-	95	99	99
	10 mA	87	98	98	-	-	-	90	98	98
	12 mA	86	98	98	-	-	-	87	98	98

Table 4–97. Maximum Output Clock Toggle Rate Derating Factors (Part 5 of 5)

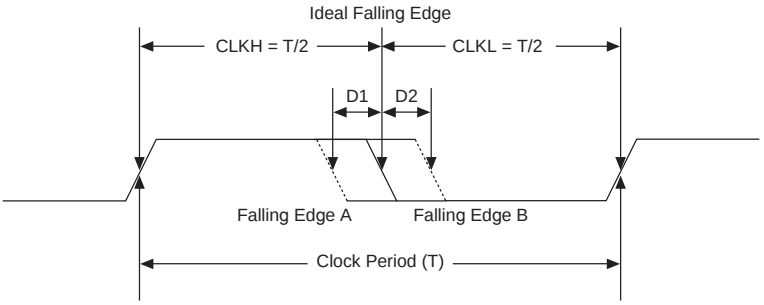
I/O Standard	Drive Strength	Maximum Output Clock Toggle Rate Derating Factors (ps/pF)								
		Column I/O Pins			Row I/O Pins			Dedicated Clock Outputs		
		-3	-4	-5	-3	-4	-5	-3	-4	-5
1.5-V differential HSTL Class II *(3)*	16 mA	95	101	101	-	-	-	96	101	101
	18 mA	95	100	100	-	-	-	101	100	100
	20 mA	94	101	101	-	-	-	104	101	101
3.3-V PCI		134	177	177	-	-	-	143	177	177
3.3-V PCI-X		134	177	177	-	-	-	143	177	177
LVDS		-	-	-	155 *(1)*	155 *(1)*	155 *(1)*	134	134	134
LVPECL *(4)*		-	-	-	-	-	-	134	134	134
3.3-V LVTTL	OCT 50 Ω	133	152	152	133	152	152	147	152	152
2.5-V LVTTL	OCT 50 Ω	207	274	274	207	274	274	235	274	274
1.8-V LVTTL	OCT 50 Ω	151	165	165	151	165	165	153	165	165
3.3-V LVCMOS	OCT 50 Ω	300	316	316	300	316	316	263	316	316
1.5-V LVCMOS	OCT 50 Ω	157	171	171	157	171	171	174	171	171
SSTL-2 Class I	OCT 50 Ω	121	134	134	121	134	134	77	134	134
SSTL-2 Class II	OCT 25 Ω	56	101	101	56	101	101	58	101	101
SSTL-18 Class I	OCT 50 Ω	100	123	123	100	123	123	106	123	123
SSTL-18 Class II	OCT 25 Ω	61	110	110	-	-	-	59	110	110
1.2-V HSTL *(2)*	OCT 50 Ω	95	-	-	-	-	-	95	-	-

(1) For LVDS output on row I/O pins the toggle rate derating factors apply to loads larger than 5 pF. In the derating calculation, subtract 5 pF from the intended load value in pF for the correct result. For a load less than or equal to 5 pF, refer to Tables 4–91 through 4–95 for output toggle rates.
(2) 1.2-V HSTL is only supported on column I/O pins on -3 devices.
(3) Differential HSTL and SSTL is only supported on column clock and DQS outputs.
(4) LVPECL is only supported on column clock outputs.

Duty Cycle Distortion

Duty cycle distortion (DCD) describes how much the falling edge of a clock is off from its ideal position. The ideal position is when both the clock high time (CLKH) and the clock low time (CLKL) equal half of the clock period (T), as shown in Figure 4–11. DCD is the deviation of the non-ideal falling edge from the ideal falling edge, such as D1 for the falling edge A and D2 for the falling edge B (see Figure 4–11). The maximum DCD for a clock is the larger value of D1 and D2.

Figure 4–11. Duty Cycle Distortion



DCD expressed in absolution derivation, for example, D1 or D2 in Figure 4–11, is clock-period independent. DCD can also be expressed as a percentage, and the percentage number is clock-period dependent. DCD as a percentage is defined as:

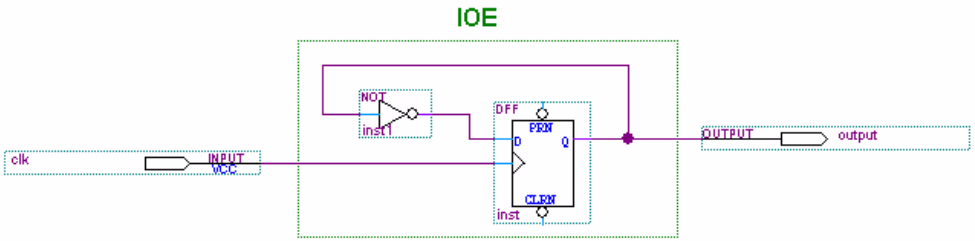
(T/2 – D1) / T (the low percentage boundary)

(T/2 + D2) / T (the high percentage boundary)

DCD Measurement Techniques

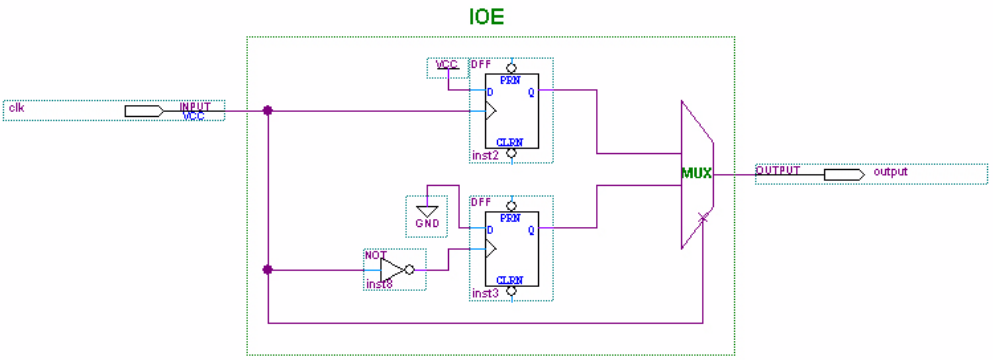
DCD is measured at an FPGA output pin driven by registers inside the corresponding I/O element (IOE) block. When the output is a single data rate signal (non-DDIO), only one edge of the register input clock (positive or negative) triggers output transitions (Figure 4–12). Therefore, any DCD present on the input clock signal or caused by the clock input buffer or different input I/O standard does not transfer to the output signal.

Figure 4–12. DCD Measurement Technique for Non-DDIO (Single-Data Rate) Outputs



However, when the output is a double data rate input/output (DDIO) signal, both edges of the input clock signal (positive and negative) trigger output transitions (Figure 4–13). Therefore, any distortion on the input clock and the input clock buffer affect the output DCD.

Figure 4–13. DCD Measurement Technique for DDIO (Double-Data Rate) Outputs



When an FPGA PLL generates the internal clock, the PLL output clocks the IOE block. As the PLL only monitors the positive edge of the reference clock input and internally re-creates the output clock signal, any DCD present on the reference clock is filtered out. Therefore, the DCD for a DDIO output with PLL in the clock path is better than the DCD for a DDIO output without PLL in the clock path.

Tables 4–98 through 4–105 show the maximum DCD in absolution derivation for different I/O standards on Stratix II GX devices. Examples are also provided that show how to calculate DCD as a percentage.

Table 4–98. Maximum DCD for Non-DDIO Output on Row I/O Pins

Row I/O Output Standard	Maximum DCD (ps) for Non-DDIO Output		
	-3 Devices	-4 and -5 Devices	Unit
3.3-V LVTTTL	245	275	ps
3.3-V LVCMOS	125	155	ps
2.5 V	105	135	ps
1.8 V	180	180	ps
1.5-V LVCMOS	165	195	ps
SSTL-2 Class I	115	145	ps
SSTL-2 Class II	95	125	ps
SSTL-18 Class I	55	85	ps
1.8-V HSTL Class I	80	100	ps
1.5-V HSTL Class I	85	115	ps
LVDS	55	80	ps

Here is an example for calculating the DCD as a percentage for a non-DDIO output on a row I/O on a -3 device:

If the non-DDIO output I/O standard is SSTL-2 Class II, the maximum DCD is 95 ps (see Table 4–99). If the clock frequency is 267 MHz, the clock period T is:

T = 1/ f = 1 / 267 MHz = 3.745 ns = 3,745 ps

To calculate the DCD as a percentage:

(T/2 – DCD) / T = (3,745 ps/2 – 95 ps) / 3,745 ps = 47.5% (for low boundary)

(T/2 + DCD) / T = (3,745 ps/2 + 95 ps) / 3,745 ps = 52.5% (for high boundary)

Therefore, the DCD percentage for the output clock at 267 MHz is from 47.5% to 52.5%.

Table 4–99. Maximum DCD for Non-DDIO Output on Column I/O Pins

Column I/O Output Standard I/O Standard	Maximum DCD (ps) for Non-DDIO Output		Unit
	-3 Devices	-4 and -5 Devices	
3.3-V LVTTL	190	220	ps
3.3-V LVCMOS	140	175	ps
2.5 V	125	155	ps
1.8 V	80	110	ps
1.5-V LVCMOS	185	215	ps
SSTL-2 Class I	105	135	ps
SSTL-2 Class II	100	130	ps
SSTL-18 Class I	90	115	ps
SSTL-18 Class II	70	100	ps
1.8-V HSTL Class I	80	110	ps
1.8-V HSTL Class II	80	110	ps
1.5-V HSTL Class I	85	115	ps
1.5-V HSTL Class II	50	80	ps
1.2-V HSTL-12	170	200	ps
LVPECL	55	80	ps

Table 4–100. Maximum DCD for DDIO Output on Row I/O Pins Without PLL in the Clock Path for -3 Devices *Note (1)*

Maximum DCD (ps) for Row DDIO Output I/O Standard	Input I/O Standard (No PLL in Clock Path)					Unit
	TTL/CMOS		SSTL-2	SSTL/HSTL	LVDS	
	3.3 and 2.5 V	1.8 and 1.5 V	2.5 V	1.8 and 1.5 V	3.3 V	
3.3-V LVTTL	260	380	145	145	110	ps
3.3-V LVCMOS	210	330	100	100	65	ps
2.5 V	195	315	85	85	75	ps
1.8 V	150	265	85	85	120	ps
1.5-V LVCMOS	255	370	140	140	105	ps
SSTL-2 Class I	175	295	65	65	70	ps
SSTL-2 Class II	170	290	60	60	75	ps
SSTL-18 Class I	155	275	55	50	90	ps
1.8-V HSTL Class I	150	270	60	60	95	ps
1.5-V HSTL Class I	150	270	55	55	90	ps
LVDS	180	180	180	180	180	ps

(1) The information in Table 4–100 assumes the input clock has zero DCD.

Here is an example for calculating the DCD in percentage for a DDIO output on a row I/O on a -3 device:

If the input I/O standard is 2.5-V SSTL-2 and the DDIO output I/O standard is SSTL-2 Class= II, the maximum DCD is 60 ps (see Table 4–100). If the clock frequency is 267 MHz, the clock period T is:

T = 1/ f = 1 / 267 MHz = 3.745 ns = 3,745 ps

Calculate the DCD as a percentage:

(T/2 – DCD) / T = (3,745 ps/2 – 60 ps) / 3745 ps = 48.4% (for low boundary)

(T/2 + DCD) / T = (3,745 ps/2 + 60 ps) / 3745 ps = 51.6% (for high boundary)

Therefore, the DCD percentage for the output clock is from 48.4% to 51.6%.

Table 4–101. Maximum DCD for DDIO Output on Row I/O Pins Without PLL in the Clock Path for -4 and -5 Devices *Note (1)*

Maximum DCD (ps) for Row DDIO Output I/O Standard	Input I/O Standard (No PLL in the Clock Path)					Unit
	TTL/CMOS		SSTL-2	SSTL/HSTL	LVDS	
	3.3/2.5V	1.8/1.5V	2.5V	1.8/1.5V	3.3V	
3.3-V LVTTL	440	495	170	160	105	ps
3.3-V LVCMOS	390	450	120	110	75	ps
2.5 V	375	430	105	95	90	ps
1.8 V	325	385	90	100	135	ps
1.5-V LVCMOS	430	490	160	155	100	ps
SSTL-2 Class I	355	410	85	75	85	ps
SSTL-2 Class II	350	405	80	70	90	ps
SSTL-18 Class I	335	390	65	65	105	ps
1.8-V HSTL Class I	330	385	60	70	110	ps
1.5-V HSTL Class I	330	390	60	70	105	ps
LVDS	180	180	180	180	180	ps

(1) Table 4–101 assumes the input clock has zero DCD.

Table 4–102. Maximum DCD for DDIO Output on Column I/O Pins Without PLL in the Clock Path for -3 Devices (Part 1 of 2) *Note (1)*

Maximum DCD (ps) for DDIO Column Output I/O Standard	Input IO Standard (No PLL in the Clock Path)					Unit
	TTL/CMOS		SSTL-2	SSTL/HSTL	HSTL12	
	3.3/2.5V	1.8/1.5V	2.5V	1.8/1.5V	1.2V	
3.3-V LVTTL	260	380	145	145	145	ps
3.3-V LVCMOS	210	330	100	100	100	ps
2.5 V	195	315	85	85	85	ps
1.8 V	150	265	85	85	85	ps
1.5-V LVCMOS	255	370	140	140	140	ps
SSTL-2 Class I	175	295	65	65	65	ps
SSTL-2 Class II	170	290	60	60	60	ps
SSTL-18 Class I	155	275	55	50	50	ps

Table 4–102. Maximum DCD for DDIO Output on Column I/O Pins Without PLL in the Clock Path for -3 Devices (Part 2 of 2) *Note (1)*

Maximum DCD (ps) for DDIO Column Output I/O Standard	Input IO Standard (No PLL in the Clock Path)					Unit
	TTL/CMOS		SSTL-2	SSTL/HSTL	HSTL12	
	3.3/2.5V	1.8/1.5V	2.5V	1.8/1.5V	1.2V	
SSTL-18 Class II	140	260	70	70	70	ps
1.8-V HSTL Class I	150	270	60	60	60	ps
1.8-V HSTL Class II	150	270	60	60	60	ps
1.5-V HSTL Class I	150	270	55	55	55	ps
1.5-V HSTL Class II	125	240	85	85	85	ps
1.2-V HSTL	240	360	155	155	155	ps
LVPECL	180	180	180	180	180	ps

(1) Table 4–102 assumes the input clock has zero DCD.

Table 4–103. Maximum DCD for DDIO Output on Column I/O Pins Without PLL in the Clock Path for -4 and -5 Devices *Note (1)*

Maximum DCD (ps) for DDIO Column Output I/O Standard	Input IO Standard (No PLL in the Clock Path)				Unit
	TTL/CMOS		SSTL-2	SSTL/HSTL	
	3.3/2.5V	1.8/1.5V	2.5V	1.8/1.5V	
3.3-V LVTTL	440	495	170	160	ps
3.3-V LVCMOS	390	450	120	110	ps
2.5 V	375	430	105	95	ps
1.8 V	325	385	90	100	ps
1.5-V LVCMOS	430	490	160	155	ps
SSTL-2 Class I	355	410	85	75	ps
SSTL-2 Class II	350	405	80	70	ps
SSTL-18 Class I	335	390	65	65	ps
SSTL-18 Class II	320	375	70	80	ps
1.8-V HSTL Class I	330	385	60	70	ps
1.8-V HSTL Class II	330	385	60	70	ps
1.5-V HSTL Class I	330	390	60	70	ps
1.5-V HSTL Class II	330	360	90	100	ps
LVPECL	180	180	180	180	ps

(1) Table 4–103 assumes the input clock has zero DCD.

Table 4–104. Maximum DCD for DDIO Output on Row I/O Pins With PLL in the Clock Path

Maximum DCD (ps) for Row DDIO Output I/O Standard	Stratix II GX Devices (PLL Output Feeding DDIO)		Unit
	-3 Device	-4 and -5 Device	
3.3-V LVTTL	110	105	ps
3.3-V LVCMOS	65	75	ps
2.5V	75	90	ps
1.8V	85	100	ps
1.5-V LVCMOS	105	100	ps
SSTL-2 Class I	65	75	ps
SSTL-2 Class II	60	70	ps
SSTL-18 Class I	50	65	ps
1.8-V HSTL Class I	50	70	ps
1.5-V HSTL Class I	55	70	ps
LVDS	180	180	ps

Table 4–105. Maximum DCD for DDIO Output on Column I/O Pins With PLL in the Clock Path (Part 1 of 2)

Maximum DCD (ps) for Column DDIO Output I/O Standard	Stratix II GX Devices (PLL Output Feeding DDIO)		Unit
	-3 Device	-4 and -5 Device	
3.3-V LVTTL	145	160	ps
3.3-V LVCMOS	100	110	ps
2.5V	85	95	ps
1.8V	85	100	ps
1.5-V LVCMOS	140	155	ps
SSTL-2 Class I	65	75	ps
SSTL-2 Class II	60	70	ps
SSTL-18 Class I	50	65	ps
SSTL-18 Class II	70	80	ps
1.8-V HSTL Class I	60	70	ps
1.8-V HSTL Class II	60	70	ps
1.5-V HSTL Class I	55	70	ps
1.5-V HSTL Class II	85	100	ps

Table 4–105. Maximum DCD for DDIO Output on Column I/O Pins With PLL in the Clock Path (Part 2 of 2)			
Maximum DCD (ps) for Column DDIO Output I/O Standard	Stratix II GX Devices (PLL Output Feeding DDIO)		Unit
	-3 Device	-4 and -5 Device	
1.2-V HSTL	155	155	ps
LVPECL	180	180	ps

High-Speed I/O Specifications

Table 4–106 provides high-speed timing specifications definitions.

Table 4–106. High-Speed Timing Specifications and Definitions	
High-Speed Timing Specifications	Definitions
t_C	High-speed receiver/transmitter input and output clock period.
f_{HSCLK}	High-speed receiver/transmitter input and output clock frequency.
J	Deserialization factor (width of parallel data bus).
W	PLL multiplication factor.
t_{RISE}	Low-to-high transmission time.
t_{FALL}	High-to-low transmission time.
Timing unit interval (TUI)	The timing budget allowed for skew, propagation delays, and data sampling window. (TUI = 1/(Receiver Input Clock Frequency × Multiplication Factor) = t_C/w).
f_{IN}	Fast PLL input clock frequency
f_{HSDR}	Maximum/minimum LVDS data transfer rate (f_{HSDR} = 1/TUI), non-DPA.
$f_{HSDRDPA}$	Maximum/minimum LVDS data transfer rate ($f_{HSDRDPA}$ = 1/TUI), DPA.
Channel-to-channel skew (TCCS)	The timing difference between the fastest and the slowest output edges including t_{CO} variation and clock skew across channels driven by the same fast PLL. The clock is included in the TCCS measurement.
Sampling window (SW)	The period of time during which the data must be valid in order to capture it correctly. The setup and hold times determine the ideal strobe position within the sampling window.
Input jitter	Peak-to-peak input jitter on high-speed PLLs.
Output jitter	Peak-to-peak output jitter on high-speed PLLs.
t_{DUTY}	Duty cycle on high-speed transmitter output clock.
t_{LOCK}	Lock time for high-speed transmitter and receiver PLLs.

Table 4–107 shows the high-speed I/O timing specifications for -3 speed grade Stratix II GX devices.

Table 4–107. High-Speed I/O Specifications for -3 Speed Grade *Notes (1), (2)*

Symbol	Conditions			-3 Speed Grade			Unit
				Min	Typ	Max	
$f_{IN} = f_{HSDR} / W$	W = 2 to 32 (LVDS, HyperTransport technology) *(3)*			16		520	MHz
	W = 1 (SERDES bypass, LVDS only)			16		500	MHz
	W = 1 (SERDES used, LVDS only)			150		717	MHz
f_{HSDR} (data rate)	J = 4 to 10 (LVDS, HyperTransport technology)			150		1,040	Mbps
	J = 2 (LVDS, HyperTransport technology)			*(4)*		760	Mbps
	J = 1 (LVDS only)			*(4)*		500	Mbps
$f_{HSDRDPA}$ (DPA data rate)	J = 4 to 10 (LVDS, HyperTransport technology)			150		1,040	Mbps
TCCS	All differential standards			-		200	ps
SW	All differential standards			330		-	ps
Output jitter						190	ps
Output t_{RISE}	All differential I/O standards					160	ps
Output t_{FALL}	All differential I/O standards					180	ps
t_{DUTY}				45	50	55	%
DPA run length						6,400	UI
DPA jitter tolerance *(5)*	Data channel peak-to-peak jitter			0.44			UI
DPA lock time							Number of repetitions
	SPI-4	0000000000 1111111111	10%	256			
	Parallel Rapid I/O	00001111	25%	256			
		10010000	50%	256			
	Miscellaneous	10101010	100%	256			
		01010101		256			

(1) When J = 4 to 10, the SERDES block is used.
(2) When J = 1 or 2, the SERDES block is bypassed.
(3) The input clock frequency and the W factor must satisfy the following fast PLL VCO specification: 150 ≤nput clock frequency × W ≤1,040.
(4) The minimum specification is dependent on the clock source (fast PLL, enhanced PLL, clock pin, and so on) and the clock routing resource (global, regional, or local) utilized. The I/O differential buffer and input register do not have a minimum toggle rate.
(5) For setup details, refer to the characterization report.

Table 4–108 shows the high-speed I/O timing specifications for -4 speed grade Stratix II GX devices.

Table 4–108. High-Speed I/O Specifications for -4 Speed Grade *Notes (1), (2)*

Symbol	Conditions			-4 Speed Grade			Unit
				Min	Typ	Max	
$f_{IN} = f_{HSDR} / W$	W = 2 to 32 (LVDS, HyperTransport technology) *(3)*			16		520	MHz
	W = 1 (SERDES bypass, LVDS only)			16		500	MHz
	W = 1 (SERDES used, LVDS only)			150		717	MHz
f_{HSDR} (data rate)	J = 4 to 10 (LVDS, HyperTransport technology)			150		1,040	Mbps
	J = 2 (LVDS, HyperTransport technology)			*(4)*		760	Mbps
	J = 1 (LVDS only)			*(4)*		500	Mbps
$f_{HSDRDPA}$ (DPA data rate)	J = 4 to 10 (LVDS, HyperTransport technology)			150		1,040	Mbps
TCCS	All differential standards			-		200	ps
SW	All differential standards			330		-	ps
Output jitter						190	ps
Output t_{RISE}	All differential I/O standards					160	ps
Output t_{FALL}	All differential I/O standards					180	ps
t_{DUTY}				45	50	55	%
DPA run length						6,400	UI
DPA jitter tolerance	Data channel peak-to-peak jitter			0.44			UI
DPA lock time							Number of repetitions
	SPI-4	0000000000 1111111111	10%	256			
	Parallel Rapid I/O	00001111	25%	256			
		10010000	50%	256			
	Miscellaneous	10101010	100%	256			
		01010101		256			

(1) When J = 4 to 10, the SERDES block is used.
(2) When J = 1 or 2, the SERDES block is bypassed.
(3) The input clock frequency and the W factor must satisfy the following fast PLL VCO specification: 150 ≤nput clock frequency × W ≤1,040.
(4) The minimum specification is dependent on the clock source (fast PLL, enhanced PLL, clock pin, and so on) and the clock routing resource (global, regional, or local) utilized. The I/O differential buffer and input register do not have a minimum toggle rate.

Table 4–109 shows the high-speed I/O timing specifications for -5 speed grade Stratix II GX devices.

Table 4–109. High-Speed I/O Specifications for -5 Speed Grade *Notes (1), (2)*

Symbol	Conditions			-5 Speed Grade			Unit
				Min	Typ	Max	
$f_{IN} = f_{HSDR} / W$	W = 2 to 32 (LVDS, HyperTransport technology) *(3)*			16		420	MHz
	W = 1 (SERDES bypass, LVDS only)			16		500	MHz
	W = 1 (SERDES used, LVDS only)			150		640	MHz
f_{HSDR} (data rate)	J = 4 to 10 (LVDS, HyperTransport technology)			150		840	Mbps
	J = 2 (LVDS, HyperTransport technology)			*(4)*		700	Mbps
	J = 1 (LVDS only)			*(4)*		500	Mbps
$f_{HSDRDPA}$ (DPA data rate)	J = 4 to 10 (LVDS, HyperTransport technology)			150		840	Mbps
TCCS	All differential I/O standards			-		200	ps
SW	All differential I/O standards			440		-	ps
Output jitter						190	ps
Output t_{RISE}	All differential I/O standards					290	ps
Output t_{FALL}	All differential I/O standards					290	ps
t_{DUTY}				45	50	55	%
DPA run length						6,400	UI
DPA jitter tolerance	Data channel peak-to-peak jitter			0.44			UI
DPA lock time							Number of repetitions
	SPI-4	0000000000 1111111111	10%	256			
	Parallel Rapid I/O	00001111	25%	256			
		10010000	50%	256			
	Miscellaneous	10101010	100%	256			
		01010101		256			

(1) When J = 4 to 10, the SERDES block is used.
(2) When J = 1 or 2, the SERDES block is bypassed.
(3) The input clock frequency and the W factor must satisfy the following fast PLL VCO specification: 150 ≤ input clock frequency × W ≤ 840.
(4) The minimum specification is dependent on the clock source (fast PLL, enhanced PLL, clock pin, and so on) and the clock routing resource (global, regional, or local) utilized. The I/O differential buffer and input register do not have a minimum toggle rate.

PLL Timing Specifications

Tables 4–110 and 4–111 describe the Stratix II GX PLL specifications when operating in both the commercial junction temperature range (0 to 85 C) and the industrial junction temperature range (–40 to 100 C), except for the clock switchover and phase-shift stepping features. These two features are only supported from the 0 to 100 C junction temperature range.

Table 4–110. Enhanced PLL Specifications (Part 1 of 2)

Name	Description	Min	Typ	Max	Unit
f_{IN}	Input clock frequency	4		500	MHz
f_{INPFD}	Input frequency to the PFD	4		420	MHz
f_{INDUTY}	Input clock duty cycle	40		60	%
f_{ENDUTY}	External feedback input clock duty cycle	40		60	%
$t_{INJITTER}$	Input or external feedback clock input jitter tolerance in terms of period jitter. Bandwidth ≤0.85 MHz		0.5		ns (peak-to-peak)
	Input or external feedback clock input jitter tolerance in terms of period jitter. Bandwidth > 0.85 MHz		1.0		ns (peak-to-peak)
$t_{OUTJITTER}$	Dedicated clock output period jitter			250 ps for ≥ 100 MHz outclk 25 mUI for < 100 MHz outclk	ps or mUI (p-p)
t_{FCOMP}	External feedback compensation time			10	ns
f_{OUT}	Output frequency for internal global or regional clock	1.5 *(2)*		550	MHz
$f_{OUTDUTY}$	Duty cycle for external clock output	45	50	55	%
$f_{SCANCLK}$	Scanclk frequency			100	MHz
$t_{CONFIGEPLL}$	Time required to reconfigure scan chains for EPLLs		$174/f_{SCANCLK}$		ns
f_{OUT_EXT}	PLL external clock output frequency	1.5 *(2)*		*(1)*	MHz
t_{LOCK}	Time required for the PLL to lock from the time it is enabled or the end of device configuration		0.03	1	ms
t_{DLOCK}	Time required for the PLL to lock dynamically after automatic clock switchover between two identical clock frequencies			1	ms
$f_{SWITCHOVER}$	Frequency range where the clock switchover performs properly	1.5	1	500	MHz
f_{CLBW}	PLL closed-loop bandwidth	0.13	1.2	16.9	MHz

Table 4–110. Enhanced PLL Specifications (Part 2 of 2)					
Name	**Description**	**Min**	**Typ**	**Max**	**Unit**
f_{VCO}	PLL VCO operating range for –3 and –4 speed grade devices	300		1,040	MHz
	PLL VCO operating range for –5 speed grade devices	300		840	MHz
f_{SS}	Spread-spectrum modulation frequency	100		500	kHz
% spread	Percent down spread for a given clock frequency	0.4	0.5	0.6	%
t_{PLL_PSERR}	Accuracy of PLL phase shift			±30	ps
t_{ARESET}	Minimum pulse width on `areset` signal.	10			ns
$t_{ARESET\_RECONFIG}$	Minimum pulse width on the `areset` signal when using PLL reconfiguration. Reset the PLL after `scandone` goes high.	500			ns
$t_{RECONFIGWAIT}$	The time required for the wait after the reconfiguration is done and the areset is applied.			2	us

(1) This is limited by the I/O f_{MAX}. See Tables 4–91 through 4–95 for the maximum.
(2) If the counter cascading feature of the PLL is utilized, there is no minimum output clock frequency.

Table 4–111. Fast PLL Specifications (Part 1 of 2)					
Name	**Description**	**Min**	**Typ**	**Max**	**Unit**
f_{IN}	Input clock frequency (for -3 and -4 speed grade devices)	16		717	MHz
	Input clock frequency (for -5 speed grade devices)	16		640	MHz
f_{INPFD}	Input frequency to the PFD	16		500	MHz
f_{INDUTY}	Input clock duty cycle	40		60	%
$t_{INJITTER}$	Input clock jitter tolerance in terms of period jitter. Bandwidth ≤2 MHz		0.5		ns (p-p)
	Input clock jitter tolerance in terms of period jitter. Bandwidth > 0.2 MHz		1.0		ns (p-p)

Table 4–111. Fast PLL Specifications (Part 2 of 2)					
Name	**Description**	**Min**	**Typ**	**Max**	**Unit**
f_{VCO}	Upper VCO frequency range for –3 and –4 speed grades	300		1,040	MHz
	Upper VCO frequency range for –5 speed grades	300		840	MHz
	Lower VCO frequency range for –3 and –4 speed grades	150		520	MHz
	Lower VCO frequency range for –5 speed grades	150		420	MHz
f_{OUT}	PLL output frequency to GCLK or RCLK	4.6875		550	MHz
	PLL output frequency to LVDS or DPA clock	150		1,040	MHz
f_{OUT_EXT}	PLL clock output frequency to regular I/O	4.6875		*(1)*	MHz
$t_{CONFIGPLL}$	Time required to reconfigure scan chains for fast PLLs		$75/f_{SCANCLK}$		ns
f_{CLBW}	PLL closed-loop bandwidth	1.16	5	28	MHz
t_{LOCK}	Time required for the PLL to lock from the time it is enabled or the end of the device configuration		0.03	1	ms
t_{PLL_PSERR}	Accuracy of PLL phase shift			±30	ps
t_{ARESET}	Minimum pulse width on areset signal.	10			ns
$t_{ARESET\_RECONFIG}$	Minimum pulse width on the areset signal when using PLL reconfiguration. Reset the PLL after scandone goes high.	500			ns

(1) This is limited by the I/O f_{MAX}. See Tables 4–91 through 4–95 for the maximum.

External Memory Interface Specifications

Tables 4–112 through 4–116 contain Stratix II GX device specifications for the dedicated circuitry used for interfacing with external memory devices.

Table 4–112. DLL Frequency Range Specifications (Part 1 of 2)		
Frequency Mode	**Frequency Range (MHz)**	**Resolution (Degrees)**
0	100 to 175	30
1	150 to 230	22.5
2	200 to 350 (–3 speed grade)	30
	200 to 310 (–4 and –5 speed grade)	30

Table 4–112. DLL Frequency Range Specifications (Part 2 of 2)		
Frequency Mode	**Frequency Range (MHz)**	**Resolution (Degrees)**
3	240 to 400 (–3 speed grade)	36
	240 to 350 (–4 and –5 speed grade)	36

Table 4–113. DQS Jitter Specifications for DLL-Delayed Clock (t_{DQS_JITTER}) Note (1)		
Number of DQS Delay Buffer Stages *(2)*	**Commercial (ps)**	**Industrial (ps)**
1	80	110
2	110	130
3	130	180
4	160	210

(1) Peak-to-peak period jitter on the phase-shifted DQS clock. For example, jitter on two delay stages under commercial conditions is 200 ps peak-to-peak or 100 ps.
(2) Delay stages used for requested DQS phase shift are reported in a project's Compilation Report in the Quartus II software.

Table 4–114. DQS Phase-Shift Error Specifications for DLL-Delayed Clock (t_{DQS_PSERR})			
Number of DQS Delay Buffer Stages *(1)*	**–3 Speed Grade (ps)**	**–4 Speed Grade (ps)**	**–5 Speed Grade (ps)**
1	25	30	35
2	50	60	70
3	75	90	105
4	100	120	140

(1) Delay stages used for request DQS phase shift are reported in a project's Compilation Report in the Quartus II software. For example, phase-shift error on two delay stages under -3 conditions is 50 ps peak-to-peak or 25 ps.

Table 4–115. DQS Bus Clock Skew Adder Specifications (t_{DQS}_CLOCK_SKEW_ADDER)	
Mode	**DQS Clock Skew Adder (ps)** *(1)*
4 DQ per DQS	40
9 DQ per DQS	70
18 DQ per DQS	75
36 DQ per DQS	95

(1) This skew specification is the absolute maximum and minimum skew. For example, skew on a 40 DQ group is 40 ps or 20 ps.

Table 4–116. DQS Phase Offset Delay Per Stage (ps) *Notes (1), (2), (3)*				
Speed Grade	**Positive Offset**		**Negative Offset**	
	Min	**Max**	**Min**	**Max**
-3	10	15	8	11
-4	10	15	8	11
-5	10	16	8	12

(1) The delay settings are linear.
(2) The valid settings for phase offset are -32 to +31.
(3) The typical value equals the average of the minimum and maximum values.

JTAG Timing Specifications

Figure 4–14 shows the timing requirements for the JTAG signals

Figure 4–14. Stratix II GX JTAG Waveforms.

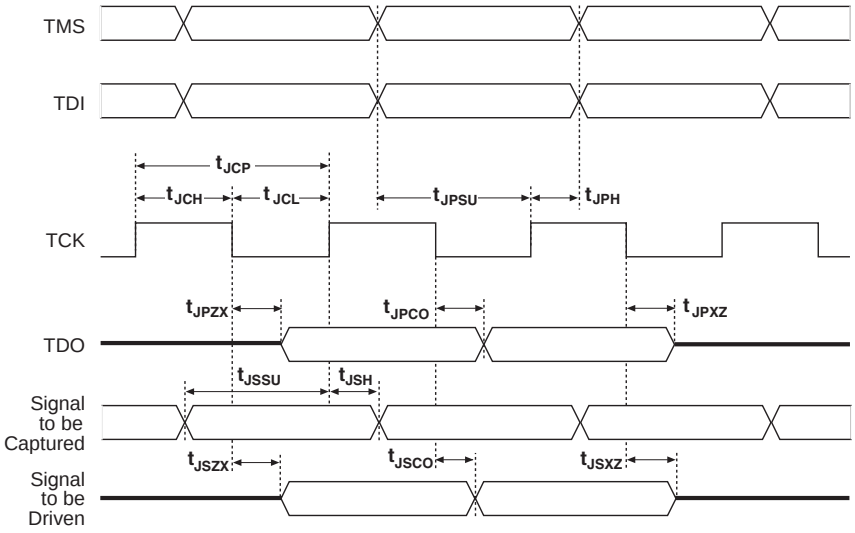


Table 4–117 shows the JTAG timing parameters and values for Stratix II GX devices.

Table 4–117. Stratix II GX JTAG Timing Parameters and Values

Symbol	Parameter	Min	Max	Unit
t_{JCP}	TCK clock period	30		ns
t_{JCH}	TCK clock high time	12		ns
t_{JCL}	TCK clock low time	12		ns
t_{JPSU}	JTAG port setup time	4		ns
t_{JPH}	JTAG port hold time	5		ns
t_{JPCO}	JTAG port clock to output		9	ns
t_{JPZX}	JTAG port high impedance to valid output		9	ns
t_{JPXZ}	JTAG port valid output to high impedance		9	ns
t_{JSSU}	Capture register setup time	4		ns
t_{JSH}	Capture register hold time	5		ns
t_{JSCO}	Update register clock to output		12	ns
t_{JSZX}	Update register high impedance to valid output		12	ns
t_{JSXZ}	Update register valid output to high impedance		12	ns

Referenced Documents

This chapter references the following documents:

- *Operating Requirements for Altera Devices Data Sheet*
- *PowerPlay Power Analyzer* chapter in volume 3 of the *Quartus II Handbook*.
- *PowerPlay Early Power Estimator (EPE) and Power Analyzer*
- *Quartus II PowerPlay Analysis and Optimization Technology*
- *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*
- Volume 2, *Stratix II GX Device Handbook*

Document Revision History

Table 6–105 shows the revision history for this chapter.

Table 4–118. Document Revision History (Part 1 of 5)

Date and Document Version	Changes Made	Summary of Changes
June 2009 v4.6	Replaced Table 4–31 Updated: • Table 4–5 • Table 4–6 • Table 4–7 • Table 4–8 • Table 4–9 • Table 4–10 • Table 4–11 • Table 4–12 • Table 4–13 • Table 4–14 • Table 4–15 • Table 4–16 • Table 4–17 • Table 4–18 • Table 4–20 • Table 4–50 • Table 4–95 • Table 4–105 • Table 4–110 • Table 4–111	
October 2007 v4.5	Updated: • Table 4–3 • Table 4–6 • Table 4–16 • Table 4–19 • Table 4–20 • Table 4–21 • Table 4–22 • Table 4–55 • Table 4–106 • Table 4–107 • Table 4–108 • Table 4–109 • Table 4–112	
	Updated title only in Tables 4–88 and 4–89.	
	Minor text edits.	

Table 4–118. Document Revision History (Part 2 of 5)		
Date and Document Version	**Changes Made**	**Summary of Changes**
August 2007 v4.4	Removed note “The data in this table is preliminary. Altera will provide a report upon completion of characterization of the Stratix II GX devices. Conditions for testing the silicon have not been determined.” from each table.	
	Removed note “The data in Tables xxx through xxx is preliminary. Altera will provide a report upon completion of characterization of the Stratix II GX devices. Conditions for testing the silicon have not been determined.” in the clock timing parameters sections.	
	Updated clock timing parameter Tables 4–63 through 4–78 (Table 4–75 was unchanged).	
	Updated Table 4–21 and added new Table 4–22.	
	Updated: ● Table 4–6 ● Table 4–16 ● Table 4–19 ● Table 4–49 ● Table 4–52 ● Table 4–107	
	Added note to Table 4–50.	
	Added: ● Figure 4–3 ● Figure 4–4 ● Figure 4–5	
	Added the “Referenced Documents” section.	
May 2007 v4.3	Changed 1.875 KHz to 1.875 MHz in Table 4–19, XAUI Receiver Jitter Tolerance section.	

Table 4–118. Document Revision History (Part 3 of 5)		
Date and Document Version	**Changes Made**	**Summary of Changes**
February 2007 v4.2	Added the “Document Revision History” section to this chapter.	Added support information for the Stratix II GX device.
	Updated Table 4–5: ● Removed last three lines ● Removed note 1 ● Added new note 4	
	Deleted table 6-6.	
	Replaced Table 4–6 with all new information.	
	Added Figures 4–1 and 4–2.	
	Added Tables 4–7 through 4–19.	
	Removed Figures 6-1 through 6-4.	
	Updated Table 4–22: ● Changed R_{CONF} information.	
	Updated Table 4–52 ● SSTL-18 Class I, column 1: changed 25 to 50.	
	Updated: ● Table 4–54 ● Table 4–87 ● Table 4–91 ● Table 4–94	
	Updated Tables 4–62 through 4–77	
	Updated Tables 4–79 and 4–80 ● Added “units” column	
	Updated Tables 4–83 through 4–86 ● Changed column title to “Fast Corner Industrial/Commercial”.	
	Updated Table 4–109. ● Added a new line to the bottom of the table.	
August 2006 v4.1	Update Table 6–75, Table 6–84, and Table 6–90.	

Table 4–118. Document Revision History (Part 4 of 5)

Date and Document Version	Changes Made	Summary of Changes
June 2006, v4.0	• Updated Table 6–5. • Updated Table 6–6. • Updated all values in Table 6–7. • Added Tables 6–8 and 6–9. • Added Figures 6–1 through 6–4. • Updated Table 6–18. • Updated Tables 6–85 through 6–96. • Added Table 6–80, Stratix II GX Maximum Output Clock Rate for Dedicated Clock Pins. • Updated Table 6–100. • In "I/O Timing Measurement Methodology" section, updated Table 6–42. • In "Internal Timing Parameters" section, updated Tables 6–43 through 6–48. • In "Stratix II GX Clock Timing Parameters" section, updated Tables 6–50 through 6–65. • In "IOE Programmable Delay" section, updated Tables 6–67 and 6–68. • In "I/O Delays" section, updated Tables 6–71 through 6–74. • In "Maximum Input & Output Clock Toggle Rate" section, updated Tables 6–75 through 6–83. • In "DCD Measurement Techniques" section, updated Tables 6–85 through 6–92. • In "High-Speed I/O Specifications" section, updated Tables 6–94 through 6–96. • In "External Memory Interface Specifications" section, updated Table 6–100.	• Removed rows for V_{ID}, V_{OD}, V_{ICM}, and V_{OCM} from Table 6–5. • Updated values for rx, tx, and `refclkb` in Table 6–6. • Removed table containing 1.2-V PCML I/O information. That information is in Table 6–7. • Added values to Table 6–100.

Table 4–118. Document Revision History (Part 5 of 5)		
Date and Document Version	**Changes Made**	**Summary of Changes**
April 2006, v3.0	• Updated Table 6–3. • Updated Table 6–5. • Updated Table 6–7. • Added Table 6–42. • Updated "Internal Timing Parameters" section (Tables 6–43 through 6–48). • Updated "Stratix II GX Clock Timing Parameters" section (Tables 6–49 through 6–65). • Updated "IOE Programmable Delay" section (Tables 6–67 and 6–68) • Updated "I/O Delays" section (Tables 6–71 through 6–74. • Updated "Maximum Input & Output Clock Toggle Rate" section. Replaced tables 6-73 and 6-74 with Tables 6–75 through 6–83. Input and output clock rates for row, column, and dedicated clock pins are now in separate tables.	
February 2006, v2.1	• Updated Tables 6–4 and 6–5. • Updated Tables 6–49 through 6–65 (removed column designations for industrial/commercial and removed industrial numbers).	
December 2005, v2.0	Updated timing numbers.	
October 2005 v1.1	• Updated Table 6–7. • Updated Table 6–38. • Updated 3.3-V PCML information and notes to Tables 6–73 through 6–76. • Minor textual changes throughout the document.	
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	

5. Reference and Ordering Information

SIIGX51007-1.3

Software

Stratix® II GX devices are supported by the Altera® Quartus® II design software, which provides a comprehensive environment for system-on-a-programmable-chip (SOPC) design. The Quartus II software includes HDL and schematic design entry, compilation and logic synthesis, full simulation and advanced timing analysis, SignalTap® II logic analyzer, and device configuration.

Refer to the *Quartus II Development Software Handbook* for more information on the Quartus II software features.

The Quartus II software supports the Windows XP/2000/NT, Sun Solaris 8/9, Linux Red Hat v7.3, Linux Red Hat Enterprise 3, and HP-UX operating systems. It also supports seamless integration with industry-leading EDA tools through the NativeLink interface.

Device Pin-Outs

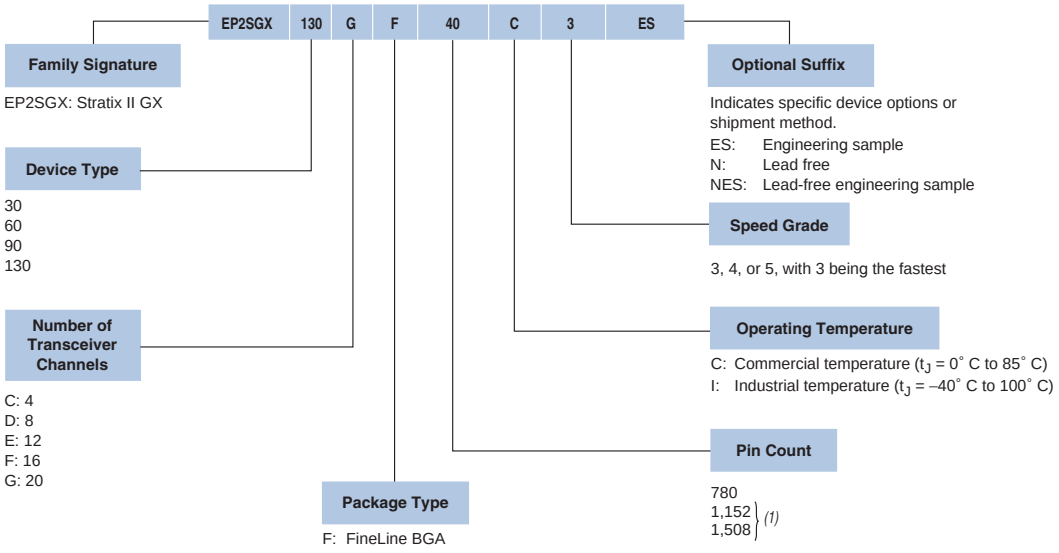
Stratix II GX device pin-outs (*Pin-Out Files for Altera Devices*) are available on the Altera web site at www.altera.com.

Ordering Information

Figure 5–1 describes the ordering codes for Stratix II GX devices.

For more information on a specific package, refer to the *Package Information for Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 5–1. Stratix II GX Device Packaging Ordering Information



(1) Product code notations for ES silicon for all EP2SGX130 family members (standard and lead free) and EP2SGX90 (lead free) use the following codings to denote pin count: 35 for 1152-pin devices and 40 for 1508-pin devices

Referenced Documents

This chapter references the following documents:

- *Package Information for Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Pin-Out Files for Altera Devices*
- *Quartus II Development Software Handbook*

Document Revision History

Table 5–1 shows the revision history for this chapter.

Table 5–1. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
August 2007 v1.3	Added the "Referenced Documents" section.	
	Minor text edits.	

Table 5–1. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
February 2007 v1.2	Added the "Document Revision History" section.	Added support information for the Stratix II GX device.
June 2006, v1.1	• Updated "Device Pin-Outs" section. • Updated Figure 7–1.	
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	

Stratix II GX Device Handbook, Volume 2

101 Innovation Drive
San Jose, CA 95134
www.altera.com

SIIGX5V2-4.3

Contents

Section I. Stratix II GX Transceiver User Guide

Chapter 1. Stratix II GX Transceiver Block Overview

Chapter 2. Stratix II GX Transceiver Architecture Overview

Chapter 3. Stratix II GX Dynamic Reconfiguration

Chapter 4. Stratix II GX ALT2GXB Megafunction User Guide

Chapter 5. Stratix II GX ALT2GXB_RECONFIG Megafunction User Guide

Chapter 6. Specifications & Additional Information

Section II. Clock Management

Chapter 7. PLLs in Stratix II and Stratix II GX Devices

Section III. Memory

Chapter 8. TriMatrix Embedded Memory Blocks in Stratix II and Stratix II GX Devices

Chapter 9. External Memory Interfaces in Stratix II and Stratix II GX Devices

Section IV. I/O Standards

Chapter 10. Selectable I/O Standards in Stratix II and Stratix II GX Devices

Chapter 11. High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices

Section V. Digital Signal Processing (DSP)

Chapter 12. DSP Blocks in Stratix II & Stratix II GX Devices

Section VI. Configuration & Remote System Upgrades

Chapter 13. Configuring Stratix II & Stratix II GX Devices

Chapter 14. Remote System Upgrades with Stratix II & Stratix II GX Devices

Chapter 15. IEEE 1149.1 (JTAG) Boundary-Scan Testing for Stratix II & Stratix II GX Devices

Section VII. PCB Layout Guidelines

Chapter 16. Package Information for Stratix II & Stratix II GX Devices

Chapter Revision Dates

The chapters in this book, *Stratix II GX Device Handbook, Volume 2*, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Stratix II GX Transceiver Block Overview
Revised: *October 2007*
Part number: *SIIGX52001-2.4*

Chapter 2. Stratix II GX Transceiver Architecture Overview
Revised: *October 2007*
Part number: *SIIGX52002-4.2*

Chapter 3. Stratix II GX Dynamic Reconfiguration
Revised: *October 2007*
Part number: *SIIGX52007-1.1*

Chapter 4. Stratix II GX ALT2GXB Megafunction User Guide
Revised: *October 2007*
Part number: *SIIGX52003-4.2*

Chapter 5. Stratix II GX ALT2GXB_RECONFIG Megafunction User Guide
Revised: *October 2007*
Part number: *SIIGX52006-1.4*

Chapter 6. Specifications & Additional Information
Revised: *October 2007*
Part number: *SIIGX52004-3.1*

Chapter 7. PLLs in Stratix II and Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52001-4.5*

Chapter 8. TriMatrix Embedded Memory Blocks in Stratix II and Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52002-4.5*

Chapter 9. External Memory Interfaces in Stratix II and Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52003-4.5*

Chapter 10. Selectable I/O Standards in Stratix II and Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52004-4.6*

Chapter 11. High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52005-2.3*

Chapter 12. DSP Blocks in Stratix II & Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52006-2.2*

Chapter 13. Configuring Stratix II & Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52007-4.5*

Chapter 14. Remote System Upgrades with Stratix II & Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52008-4.5*

Chapter 15. IEEE 1149.1 (JTAG) Boundary-Scan Testing for Stratix II & Stratix II GX Devices
Revised: *October 2007*
Part number: *SII52009-3.3*

Chapter 16. Package Information for Stratix II & Stratix II GX Devices
Revised: *May 2007*
Part number: *SII52010-4.3*

About this Handbook

This handbook provides comprehensive information about the Altera® Stratix® II GX family of devices.

How to Contact Altera

For the most up-to-date information about Altera products, refer to the following table.

Contact *(1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to table:
(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*

Visual Cue	Meaning
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>*.**pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
WARNING	The warning calls attention to a condition or possible situation that could cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

Section I. Stratix II GX Transceiver User Guide

This section provides information on the configuration modes for Stratix® II GX devices. It also includes information on testing, Stratix II GX port and parameter information, and pin constraint information.

This section includes the following chapters:

- Chapter 1, Stratix II GX Transceiver Block Overview
- Chapter 2, Stratix II GX Transceiver Architecture Overview
- Chapter 3, Stratix II GX Dynamic Reconfiguration
- Chapter 4, Stratix II GX ALT2GXB Megafunction User Guide
- Chapter 5, Stratix II GX ALT2GXB_RECONFIG Megafunction User Guide
- Chapter 6, Specifications & Additional Information

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

1. Stratix II GX Transceiver Block Overview

SIIGX52001-2.4

Introduction

Stratix® II GX devices combine highly advanced 6.375-Gigabits per second (Gbps) four-channel gigabit transceiver blocks with the industry's most advanced FPGA architecture. The Stratix II GX transceiver builds on the success of the Stratix GX family by offering higher data rate support and additional features that enable you to support a wide variety of standard and custom protocols. Each self-contained Stratix II GX gigabit transceiver block has a variety of embedded functions to implement commonly required tasks.

Building Blocks

Stratix II GX transceivers are structured into duplex four-channel groups called transceiver blocks. You can configure each channel within a transceiver block in either single-width or double-width mode. Single-width mode has an 8-bit/10-bit serializer/deserializer (SERDES) data path through the transceiver and supports data rates from 600 Mbps to 3.125 Gbps. Double-width mode has a 16-bit/20-bit SERDES data path through the transceiver and supports data rates from 1 Gbps to 6.375 Gbps. All blocks in the transceiver can operate in double-width mode, except deskew first-in first-out (FIFO), which is available only in single-width mode. The options for blocks available in the transceiver may differ depending on which mode (single or double) you use.

This documentation uses the terminology inter-transceiver block routing instead of inter-quad (IQ) routing, as seen in the Quartus II software.

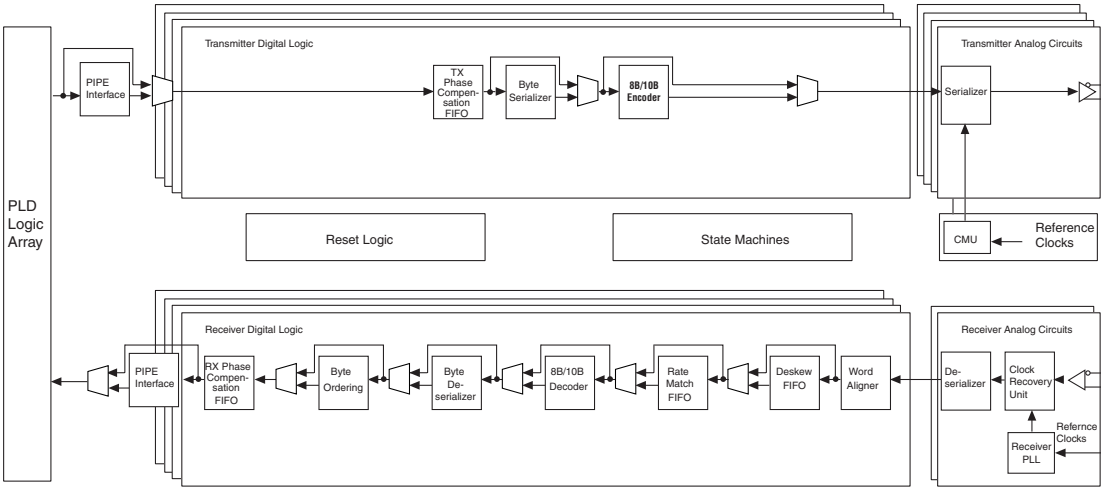
In addition to custom (Basic) modes, Stratix II GX transceivers support the following protocols:

- Physical Interface for PCI Express (PIPE) – single lane (×1), four lane (×4), and eight lane (×8)
- XAUI (10 Gigabit Attachment Unit Interface)
- GIGE (Gigabit Ethernet)
- SONET/SDH (Synchronous Optical NETwork) – OC-12, OC-48, and OC-96
- (OIF) CEI PHY Interface (Common Electrical I/O)
- Serial RapidIO (1.25 Gbps, 2.5 Gbps, and 3.125 Gbps)
- CPRI (Common Public Radio Interface)
- SDI (Serial Digital Interface) (HD-SDI and 3G-SDI)

Figure 1–1 shows a block diagram of the gigabit transceiver block in single-width mode. You enable or disable various optional modules based on the functional mode you select. The sections that follow Figure 1–1 give a brief description of each block.

For detailed information about each block, refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

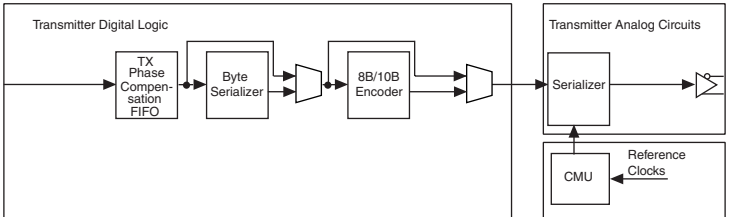
Figure 1–1. Stratix II GX Gigabit Transceiver Block Diagram



Transmitter Channel Overview

This section provides a brief description about the various components within the transmitter block (Figure 1–2). The modules are listed in order from the parallel logic array to the transmit buffer of the transmitter.

Figure 1–2. Stratix II GX Transmitter Block Diagram



Clock Multiplier Unit

Each gigabit transceiver block has a clock multiplier unit (CMU) to provide clocking flexibility and support a range of incoming data streams. Each CMU contains two transmitter phase-locked loops (PLLs) that generate the required clock frequencies based upon the synthesis of an input reference clock. Each transmitter PLL supports multiplication factors to allow the use of various input clock frequencies. Both transmitter PLLs are identical and support data ranges from 600 Mbps to 6.375 Gbps. However, each PLL can be configured to support different data rates. Each transmitter PLL drives up to four channels. In PIPE x8 mode, the transmitter PLL of the master transceiver block drives all eight channels. This CMU block is active in both single- and double-width modes and is powered down when not in use.

Phase Compensation FIFO Buffer

The transmitter data path has a dedicated phase compensation FIFO buffer that decouples phase variations between the FPGA and transceiver clock domains. This block is active in both single- and double-width modes and cannot be bypassed.

Byte Serializer

The byte serializer allows the programmable logic device (PLD) to run at half the rate of the transmit data path to allow the core to run at a lower frequency. Without the byte serializer, at the maximum data rate of 6.375 Gbps with a 20-bit serialization factor, the PLD-transceiver interface needs to run at 318.75 MHz. The PLD-transceiver interface can run at a maximum frequency of 250 MHz. With the byte serializer, the PLD-transceiver interface needs to run at 159.375 MHz. This block is available in both single- and double-width modes. In single-width mode, the PLD interface is either 16 or 20 bits when the byte serializer used. In double-width mode, using the byte serializer creates a PLD interface of 32 bits or 40 bits, depending on the serialization factor.

8B/10B Encoder

Many protocols use 8B/10B encoding. Stratix II GX devices have two dedicated 8B/10B encoders in each transmitter channel. This encoding technique ensures sufficient data transitions and a DC-balanced stream within the data signal for successful data recovery at the receiver. This block is available in single- and double-width modes. In single-width mode, one of the 8B/10B encoders is active. In double-width mode, both 8B/10B encoders are active and operate in a cascade mode. The 8B/10B encoder follows the IEEE 802.3 1998 edition standard for 8B/10B encoding.

Serializer

The serializer converts the incoming lower speed parallel signal from the transceiver's physical coding sublayer (PCS) to a high-speed serial signal on the transmit side. The serializer supports a variety of conversion factors, ensuring implementation flexibility. The serializer supports an 8- or 10-bit serialization factor in single-width mode and a 16- or 20-bit serialization factor in double-width mode. The serializer block also performs clock synthesis on the slow-speed clock for the parallel transmitter logic in the transceiver and PLD.

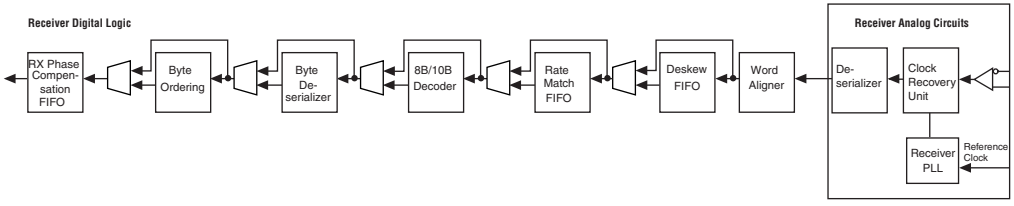
Transmitter Differential Output Buffers

The gigabit transceiver block differential output buffers support the 1.5-V PCML and 1.2-V PCML I/O standards and have a variety of features that improve system signal integrity. Programmable pre-emphasis helps compensate for high frequency losses. A variety of programmable voltage output differential (V_{OD}) settings allow noise margin tuning capabilities. Additionally, on-chip termination (OCT) provides the appropriate transmitter buffer termination for 100-, 120-, or 150-Ω transmission lines. The transmitter buffer circuit also contains a receiver-detect circuit for use with the PCI Express (PIPE) protocol to detect if a receiver is connected. The buffer can be tri-stated to reduce electromagnetic interference (EMI) and power consumption when not in use. In PIPE mode, the tri-state feature generates Electrical Idle.

Receiver Channel Overview

This section provides a brief description about the various components within the receiver block. The modules originate from the serial receiver buffer to the parallel FPGA interface (Figure 1–3).

Figure 1–3. Stratix II GX Receiver Block Diagram



Receiver Differential Input Buffers

Stratix II GX transceiver block differential input buffers support 1.5-V PCML and 1.2-V PCML I/O standards and have a variety of features that improve system signal integrity. Programmable equalization capabilities compensate for signal degradation across transmission mediums. Additionally, on-chip termination provides the appropriate receiver termination for 100-, 120-, or 150-Ω transmission lines. A signal detection block indicates if there is a valid signal at the receiver input.

Stratix II GX receiver input buffers also support the adaptive equalization (AEQ) capability to compensate for changing link characteristics.

Receiver PLL

The receiver PLL ramps the voltage controlled oscillator (VCO) to the frequency of the reference clock. Once that occurs, the clock recovery unit (CRU) controls the VCO. Each receiver channel in the transceiver has a dedicated receiver PLL that provides clocking flexibility and supports a range of data rates. These PLLs generate the required clock frequencies based upon the synthesis of an input reference clock.

Clock Recovery Unit

The Stratix II GX transceiver block CRU performs analog clock data recovery (CDR). The CRU recovers the embedded clock in the data stream to properly clock the incoming data. The recovered clock also clocks the reset of the receiver logic clock (`rx_digitalreset`) and is available in the PLD fabric.

Deserializer

The deserializer block converts the incoming data stream from a high-speed serial signal to a lower-speed parallel signal that can be processed in the FPGA logic array on the receive side. The deserializer supports a variety of conversion factors, ensuring implementation flexibility. The deserializer supports an 8- or 10-bit deserialization factor in the single-width mode and a 16- or 20-bit deserialization factor in double-width mode. The deserializer block also performs clock synthesis on the slow-speed clock from the CRU and forwards the recovered clock to the parallel receiver logic in the transceiver and for the PLD.

Word Aligner

The word aligner module contains a fully programmable pattern detector to identify specific patterns within the incoming data stream. The pattern detector includes recognition support for control code groups for 8B/10B encoded data and A1A2 or A1A1A2A2-type frame alignment patterns for scrambled data. Custom alignment patterns are also available. The word aligner can be bypassed in some functional modes.

In single-width mode, the following word-alignment options are available:

- Manual bit-slip mode
- Manual alignment to 7-, 10-, or 16-bit patterns
- Synchronization state machine that offers programmable hysteresis for synchronization.

In double-width mode, the following word-alignment options are available:

- Manual bit-slip mode
- Manual alignment to 7-, 8-, 10-, 16-, 20-, or 32-bit patterns

Channel Aligner (Deskew)

An embedded channel aligner aligns byte boundaries across multiple channels and synchronizes the data entering the logic array from the Gigabit transceiver block's four channels. The Stratix II GX channel aligner is optimized for a 10-Gigabit Ethernet XAUI four-channel implementation. The channel aligner includes the control circuitry and channel alignment character detection defined by the 10-Gigbit Attachment Unit Interface (XAUI) protocol.

This block is only available for the XAUI protocol and is disabled for all other protocols.

Rate Matcher

In CDR-based systems, the clock frequencies of the transmitting and receiving devices often do not match. This mismatch can cause the data to transmit at a rate slightly faster or slower than the receiving device can interpret. The Stratix II GX rate matcher resolves the frequency differences between the recovered clock and the FPGA logic array clock by inserting or deleting removable characters from the data stream, as defined by the transmission protocol, without compromising transmitted

data. The rate matcher block is available for single- and double-width Basic modes and for specific protocols—XAUI, Gigabit Ethernet (GIGE), and PCI Express (PIPE).

8B/10B Decoder

Various protocols use 8B/10B decoding. Stratix II GX devices have two dedicated 8B/10B decoders in each channel to support high data rates. This decoding technique ensures fast disparity and code group error detection. This block is available in single- and double-width modes. In single-width mode, only one of the 8B/10B decoders is active. In double-width mode, both 8B/10B decoders are active and operate in a cascade mode. The current running disparity can be sent to the PLD for each decoded code group. The 8B/10B decoder follows the IEEE 802.3 1998 edition standard for 8B/10B decoding.

Byte Deserializer

The byte deserializer widens the transceiver data path before the PLD interface to reduce the rate at which the received data must be clocked in the PLD logic. This byte deserializer block is available in both single- and double-width modes. In single-width mode, the PLD interface is either 16 or 20 bits when used. In double-width mode, using the byte deserializer creates a PLD interface of 32 or 40 bits, depending on your serialization factor.

Byte Ordering

Each receiver has an optional byte ordering block that is available in some functional modes when the byte deserializer is used. This block restores the expected word ordering if the byte deserialization of the data word does not match the expected word ordering after the byte deserializer block. This block is not available when the rate matcher is used (single- or double-width mode) because the rate matcher may alter the byte order by adding or deleting bytes. It is also not available when 8B/10B is used in single-width mode.

Receiver Phase Compensation FIFO Buffer

Each receiver data path has a dedicated phase compensation FIFO buffer that decouples phase variations between the FPGA and transceiver clock domains. This block is always used and cannot be bypassed.

PIPE Interface

The PIPE interface supports the PCI Express protocol. The PIPE interface simplifies and standardizes the back-end interface to the PCI Express physical layer. This block is automatically enabled in PIPE mode and is not available in any other mode.

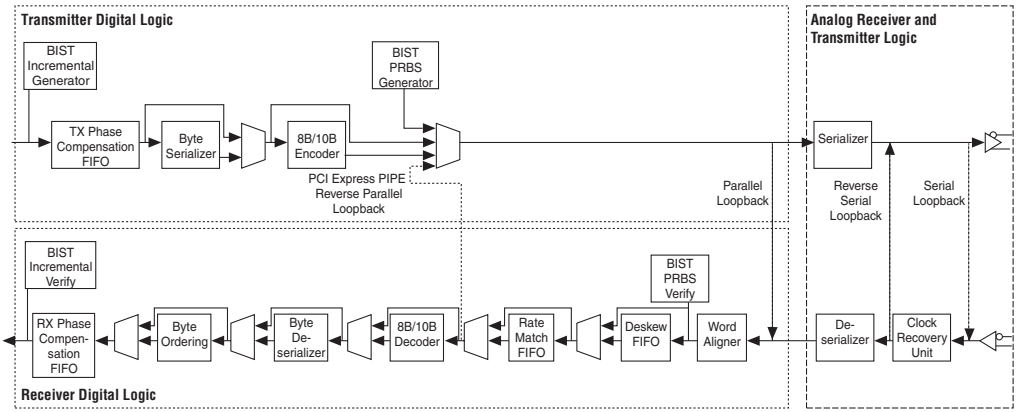
Loopback

There are four available loopback modes for diagnostic purposes. The following loopback modes are available:

- Serial loopback
- Reverse serial loopback
- Pre-CDR loopback
- Built-in self test (BIST) incremental test parallel loopback
- PCI Express (PIPE) reverse parallel loopback

Figure 1–4 shows the available loopback modes.

Figure 1–4. Loopback Modes



Built-In Self-Test

The gigabit transceiver block contains several features that simplify design verification. Embedded pattern generators and pattern verifiers provide a simple approach to board verification without the need to design additional logic in the PLD fabric. The BIST pseudo-random binary sequence (PRBS) and incremental pattern generators, along with their respective pattern verifiers, provide a full self-test path.

Reset and Power Down

Stratix II GX transceivers offer multiple reset signals to control separate ports of the transceiver channels and blocks. Each unused channel is automatically powered down to reduce power consumption. Additionally, there are dynamic power-down signals for each receiver and transmitter block.

Referenced Document

This chapter references the following document:

- *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Document Revision History

Table 1–1 shows the revision history for this chapter.

Table 1–1. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v2.4	Added note to "Receiver Differential Input Buffers" section.	—
	Updated bulleted list in"Building Blocks" and "Loopback" sections.	—
	Updated: • "Clock Multiplier Unit" • "Byte Serializer"	—
	Added "Referenced Document" section.	—
	Minor text edits.	—
August 2007 v2.3	Minor text edits.	—
February 2007 v2.2	Changed 622 Mbps to 600 Mbps in: • "Building Blocks" • "Clock Multiplier Unit"	—
	Changed 3.125 Gbps to 1 Gbps in "Building Blocks".	—
	Modified the following: • "Clock Multiplier Unit" • "Byte Serializer" • "8B/10B Encoder" • "Loopback"	—
	Updated Figure 1–3.	—

Table 1–1. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
April 2006, v2.1	• Minor change to Figures 1–1 and 1–3.	—
February 2006, v2.0	• Updated “Building Blocks” section. • Updated “Word Aligner” section. • Updated “Byte Ordering” section. • Updated “Loopback” section. • Updated “Built-In Self-Test” section.	—
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	—

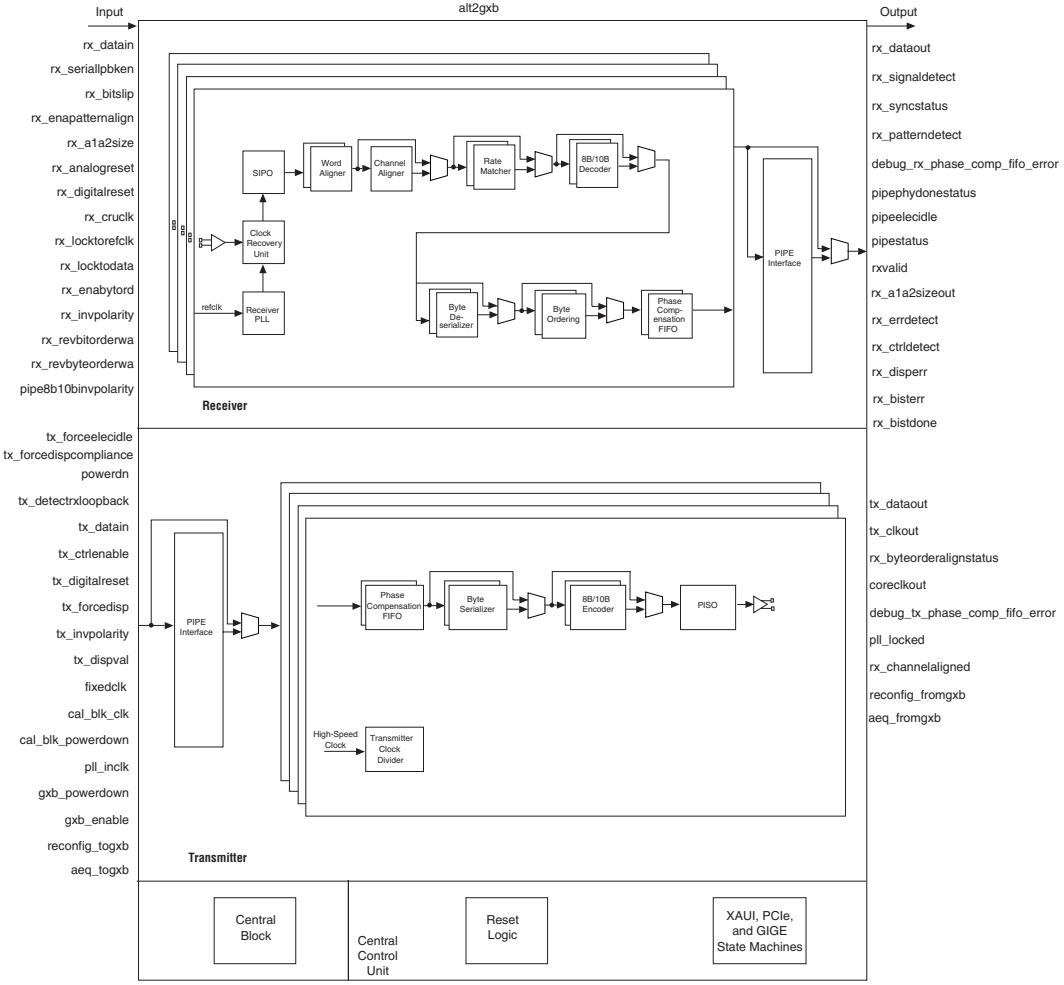
2. Stratix II GX Transceiver Architecture Overview

SIIGX52002-4.2

Introduction

This chapter provides detailed information about the architecture of Stratix® II GX devices. Figure 2–1 shows the Stratix II GX block diagram.

Figure 2–1. Stratix II GX Transceiver Block Diagram



Stratix II GX ALT2GXB Ports List

Table 2–1 provides information about the Stratix II GX ports.

Table 2–1. Stratix II GX ALT2GXB Ports (Part 1 of 7)

Port Name	Input/Output	Description	Scope
Receiver Physical Coding Sublayer (PCS) Ports			
rx_dataout	Output	Receiver parallel data output. The bus width depends on the channel width multiplied by the number of channels per instance.	
rx_clkout	Output	Recovered clock from the receiver channel.	Channel
rx_coreclk	Output	Optional read clock port for the receiver phase compensation first-in first-out (FIFO). If not selected, Quartus II software automatically selects rx_clkout/tx_clkout as the read clock for receiver phase compensation FIFO. If selected, you must drive this port with a clock that is frequency locked to rx_clkout/tx_clkout.	Channel
rx_enapatternalign	Input	Enables word aligner to align to the comma. This port can either be edge or level sensitive based on the word aligner mode. In the double-width mode, this port is only edge-sensitive.	Channel
rx_bitslip	Input	Word aligner bit-slip control. The word aligner slips a bit of the current word boundary every rising edge of this signal.	Channel
rx_rlv	Output	Run-length violation indicator. A high pulse is given when the run length has detected a violation.	Channel
rx_byteorderalignstatus	Output	From byte ordering block. A high pulse is given when the byte ordering block has successfully aligned the bytes of the PCS output.	Channel
pipe8b10binvpolarity	Input	Physical Interface for PCI Express (PIPE) polarity inversion at the 8B/10B decoder input. This port inverts the data at the input to the 8B/10B decoder.	Channel

Table 2–1. Stratix II GX ALT2GXB Ports (Part 2 of 7)

Port Name	Input/Output	Description	Scope
pipestatus	Output	PIPE receiver status port. In case of multiple status signals, the lower number signal takes precedence. 000 — Received data OK 001 — 1 skip added (not supported) 010 — 1 skip removed (not supported) 011 — Receiver detected 100 — 8B/10B decoder error 101 — Elastic buffer overflow 110 — Elastic buffer underflow 111 — Received disparity error.	Channel
pipephydonestatus	Output	PIPE indicates a mode transition completion—power transition and rx_detect. A pulse is given.	Channel
rx_pipedatavalid	Output	PIPE valid data indicator on the rx_dataout port.	Channel
pipeelecidle	Output	PIPE signal detect for PCI Express.	Channel
rx_digitalreset	Input	Reset port for the receiver PCS block. This port resets all the digital logic in the receiver channel. The minimum pulse width is two parallel clock cycles.	Channel
rx_bisterr	Output	Built-in self test (BIST) block error flag. This port latches high if an error is detected. Assertion of rx_digitalreset resets the BIST verifier, which clears the error flag.	Channel
rx_bistdone	Output	Built-in self test verifier done flag. This port goes high if the receiver finishes reception of the test sequence.	Channel
rx_ctrldetect	Output	Receiver control code indicator port. Indicates whether the data at the output of rx_dataout is a control or data word. Used with the 8B/10B decoder.	Channel
rx_errdetect	Output	8B/10B code group violation signal. Indicates that the data at the output of rx_dataout has a code violation or a disparity error. Used with disparity error signal to differentiate between a code group error and/or a disparity error. In addition, in XAUI mode, rx_errdetect is asserted in the corresponding byte position when ALT2GXB substitutes the received data with 9'b1FE because of XAUI protocol violations.	Channel

Table 2–1. Stratix II GX ALT2GXB Ports (Part 3 of 7)

Port Name	Input/Output	Description	Scope
rx_syncstatus	Output	Indicates when the word aligner either aligns to a new word boundary (in single-width mode the rx_patterndetect port is level sensitive), indicates that a resynchronization is needed (in single- or double-width mode the rx_patterndetect is edge sensitive), or indicates if synchronization is achieved or not (in single-width mode, the dedicated synchronization state machine is used).	Channel
rx_disperr	Output	8B/10B disparity error indicator port. Indicates that the data at the output of rx_dataout has a disparity error.	Channel
rx_patterndetect	Output	Indicates when the word aligner detects the alignment pattern in the current word boundary.	Channel
rx_a1a2size	Input	Available only in SONET/SDH OC-12 and OC-48 modes to select between one of the following two word alignment options: 0 – 16-bit A1A2 1 – 32-bit A1A1A2A2	Channel
rx_a1a2sizeout	Output	Available only in SONET/SDH OC-12 and OC-48 modes to indicate one of the following two word alignment options: 0 – 16-bit A1A2 1 – 32-bit A1A1A2A2	Channel
rx_invpolarity	Input	Available in all modes except (OIF) CEI PHY. Inverts the polarity of the received data at the input of the word aligner.	Channel
rx_revbitorderwa	Input	Available in Basic mode with bit-slip word alignment or dynamic reconfiguration enabled. Reverses the bit-order of the received data at a byte level at the output of the word aligner.	Channel
rx_revbyteorderwa	Input	Available in Basic double-width mode only. Swaps the MSByte and LSByte of the 16/20-bit data at the output of the word aligner.	Channel
rx_enabyteord	Input	Available in modes with byte ordering block enabled. Triggers the byte ordering block to perform byte alignment.	Channel
debug_rx_phase_comp_fifo_error	Output	Indicates receiver phase compensation FIFO overrun or underrun situation.	Channel

Table 2–1. Stratix II GX ALT2GXB Ports (Part 4 of 7)

Port Name	Input/Output	Description	Scope
Receiver Physical Media Attachment (PMA)			
rx_pll_locked	Output	Receiver PLL locked signal. Indicates if the receiver PLL is phase locked to the CRU reference clock.	Channel
rx_analogreset	Input	Receiver analog reset. Resets all analog circuits in the receiver PMA.	Channel
rx_freqlocked	Output	CRU mode indicator port. Indicates if the CRU is locked to data mode or locked to the reference clock mode. 0 – Receiver CRU is in lock-to-reference clock mode 1 – Receiver CRU is in lock-to-data mode	Channel
rx_signaldetect	Output	Signal detect port. In PIPE mode, indicates if a signal that meets the specified range is present at the input of the receiver buffer. In all other modes, rx_signaldetect is forced high and must not be used as an indication of a valid signal at receiver input.	Channel
rx_seriallpbken	Input	Serial loopback control port. 0 – normal data path, no serial loopback 1 – serial loopback	Channel
rx_locktodata	Input	Lock-to-data control for the CRU. Use with rx_locktorefclk.	Channel
rx_locktorefclk	Input	Lock-to-reference lock mode for the CRU. Use with rx_locktodata. rx_locktodata/rx_locktorefclk 0/0 – CRU is in automatic mode 0/1 – CRU is in lock-to-reference clock 1/0 – CRU is in lock-to-data mode 1/1 – CRU is in lock-to-data mode	Channel
rx_cruclk	Input	Receiver PLL/CRU reference clock.	Channel
Transmitter PCS			
tx_datain	Input	Transmitter parallel data input. The bus width is the channel width multiplied by the number of channels in the instance.	Channel
tx_clkout	Output	PLD logic array clock from the transceiver to the PLD. In a single-channel mode, there is one tx_clkout per channel.	Channel

Table 2–1. Stratix II GX ALT2GXB Ports (Part 5 of 7)

Port Name	Input/Output	Description	Scope
tx_coreclk	Input	Optional write clock port for the transmitter phase compensation FIFO. If not selected, Quartus II software automatically selects tx_clkout as the write clock for transmitter phase compensation FIFO. If selected, you must drive this port with a clock that is frequency locked to tx_clkout.	Channel
tx_detectrxloopback	Input	PIPE receiver detect / loopback pin. Depending on the power-down state the signal either activates receiver detect or loopback.	Channel
tx_forceelecidle	Input	PIPE Electrical Idle mode.	Channel
tx_forcedispcompliance	Input	PIPE forced negative disparity port for transmission of the compliance pattern. The pattern requires starting at a negative disparity. Assertion of this port at the first byte ensures that the first byte has a negative disparity. This port must be deasserted after the first byte.	Channel
powerdn	Input	PIPE power mode port. This port sets the power mode of the associated PCI Express channel. The power modes are as follows: 2'b00: P0 – Normal operation 2'b01: P0s – Low recover time latency, power saving state 2'b10: P1 – Longer recovery time (64 us max) latency, lower power state 2'b11: P2 – Lowest power state	Channel
tx_digitalreset	Input	Reset port for the transmitter PCS block. This port resets all the digital logic in the transmit channel. The minimum pulse width is two parallel clock cycles.	Channel
tx_ctrlenable	Input	Transmitter control code indicator port. Indicates whether the data at the tx_datain port is a control or data word. This port is used with the 8B/10B encoder.	Channel
tx_forcedisp	Input	Available in Basic mode with 8B/10B encoding enabled. Forces positive or negative disparity on the current symbol depending on the tx_dispval signal level.	Channel
tx_dispval	Input	Available in Basic mode with 8B/10B encoding enabled. A high forces negative starting running disparity on the current symbol and a LOW forces positive starting running disparity on the current symbol, provided tx_forcedisp signal is asserted.	Channel

Table 2–1. Stratix II GX ALT2GXB Ports (Part 6 of 7)

Port Name	Input/Output	Description	Scope
tx_invpolarity	Input	Available in all modes except (OIF) CEI PHY. Inverts the polarity of the data to be transmitted at the transmitter PCS-PMA interface (input to the serializer).	Channel
debug_tx_phase_comp_fifo_error	Output	Indicates transmitter phase compensation FIFO overrun or underrun situation.	Channel
Transmitter PMA			
fixedclk	Input	2.5-125 MHz clock for Adaptive Equalization (AEQ) feature. 125-MHz clock for receiver detect functionality in PCI Express (PIPE) mode.	Channel
Central Control Unit (CCU)			
rx_channelaligned	Output	10-Gigabit Attachment Unit Interface (XAUI) deskew FIFO aligned flag. This signal goes high after the channel aligner acquires channel alignment per the IEEE 802.3ae specification.	Transceiver block
coreclkout	Output	×4 mode output. This is the clock output from the central clock generation block. In ×8 mode, the central clock generator block from the lower transceiver generates this clock. For use with XAUI, PCI Express, ×4, and ×8 modes.	Transceiver block
reconfig_clk	Input	Input reference clock for the dynamic reconfiguration controller. The frequency range of this clock is 2.5 MHz to 50 MHz. The assigned clock uses global resources by default. This same clock should be connected to ALT2GXB.	
reconfig_togxb	Input	From reconfiguration controller for dynamic reconfiguration.	Transceiver block
reconfig_fromgxb	Output	To reconfiguration controller.	Transceiver block
aeq_togxb	Input	From reconfiguration controller for Adaptive Equalization.	Transceiver block
aeq_fromgxb	Output	To reconfiguration controller for Adaptive Equalization.	Transceiver block
CMU PMA			
gxb_powerdown	Input	Transceiver block reset and power down. This resets and powers down all circuits in the transceiver block. This does not affect the REFCLK buffers and reference clock lines.	Transceiver block

Table 2–1. Stratix II GX ALT2GXB Ports (Part 7 of 7)

Port Name	Input/Output	Description	Scope
pll_locked	Output	PLL locked indicator for the transmitter PLLs.	Transceiver block
pll_inclk	Input	Reference clocks for the transmitter PLLs.	Transceiver block
Calibration Block			
cal_blk_clk	Input	Calibration clock for the transceiver termination blocks. This clock supports frequencies from 10 MHz to 125 MHz.	Device
cal_blk_powerdown (active_low)	Input	Power-down signal for the calibration block. Assertion of this signal may interrupt data transmission and reception. Use this signal to recalibrate the termination resistors if temperature and/or voltage changes warrant it.	Device
External Signals			
tx_dataout	Output	Transmitter serial output port.	Channel
rx_datain	Input	Receiver serial input port.	Channel
rrefb *(1)*	Output	Reference resistor port. This port is always used and must be tied to a 2K-Ω resistor to ground. This port is highly sensitive to noise. There must be no noise coupled to this port.	Device
refclk *(1)*	Input	Dedicated reference clock inputs (two per transceiver block) for the transceiver. The buffer structure is similar to the receiver buffer, but the termination is not calibrated.	Transceiver block
gxb_enable	Input	Dedicated transceiver block enable pin. If instantiated, this port must be tied to the pll_ena input pin. A high level on this signal enables the transceiver block; a low level disables it.	Transceiver block

Note to Table 2–1:

(1) These are dedicated pins for the transceiver and do not appear in the MegaWizard® Plug-In Manager.

This chapter uses "transceiver block number" and "transceiver bank number" interchangeably. Table 2–2 maps transceiver block number to the Stratix II GX transceiver bank number.

Table 2–2. Transceiver Block Number to Transceiver Bank Number Mapping

Transceiver Block Number	Transceiver Bank Number
0	13
1	14
2	15
3	16
4	17

Transmitter Modules

This section describes the Stratix II GX transceiver's transmitter path. This section describes the following modules:

- Clock multiplier unit (CMU)
- Transmitter phase compensation FIFO buffer
- Byte deserializer
- 8B/10B encoder
- Serializer
- Transmitter buffer

Clock Multiplier Unit

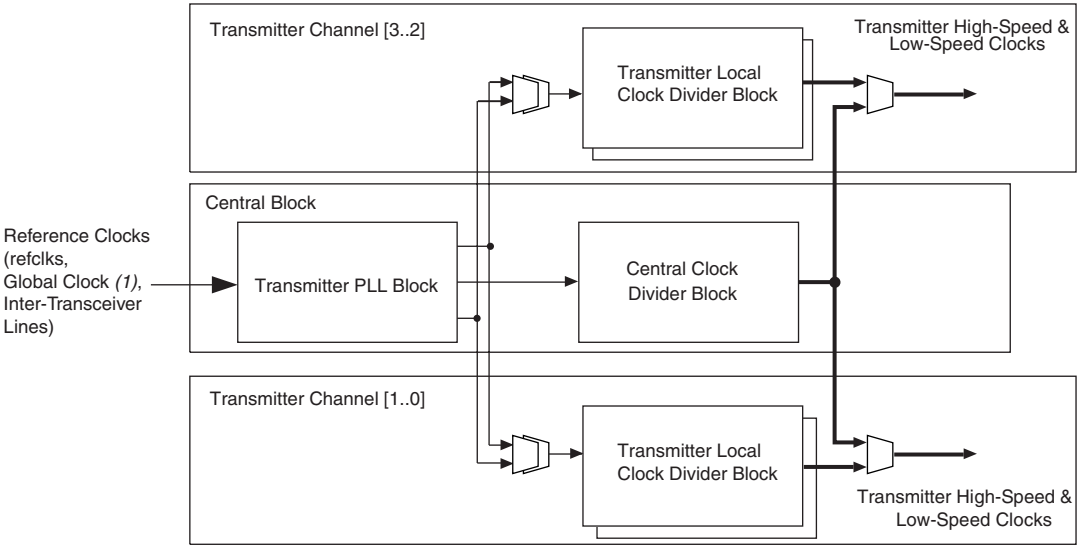
The CMU in Stratix II GX devices takes the reference clock from either the PLD or the dedicated reference clock inputs (`refclk0` and `refclk1`) and synthesizes the clocks that are used for the transmitter logic, serializer, receiver PLL reference clock, and PLD clocks.

Each transceiver block has its own CMU block that is further divided into three CMU sub blocks:

- Transmitter PLL block
- Central clock divider block
- Transmitter local clock divider block

The transmitter PLL block and central clock divider blocks are located in the central block of the transceiver block. A transmitter local clock divider block is located in each transmitter of the transceiver block. Each transceiver block has a dedicated CMU block and two dedicated reference clock inputs that feed the CMU (refer to Figure 2–2).

Figure 2–2. Clock Multiplier Unit Block Diagram



Note to Figure 2–2:
(1) The Global Clock line must be driven by an input pin.

The Quartus® II software simplifies the CMU settings. It sets most of the settings automatically for protocol modes; for example, PLL multiplication factors. You need provide only the data rate in the ALT2GXB MegaWizard Plug-In Manager and then select the input clock frequency.

Dedicated Reference Clock Pin Specifications

Table 2–3 shows the I/O standards allowed for the reference clock pins.

Table 2–3. Reference Clock Specifications (Part 1 of 2)

Protocol	I/O Standard	Coupling	Termination
Basic, XAUI, GIGE, SONET/SDH, (OIF) CEI PHY, Serial RapidIO, SDI, CPRI	1.2-V PCML, 1.5-V PCML, 3.3-V PCML, Differential LVPECL, LVDS	AC	On-chip

Table 2–3. Reference Clock Specifications (Part 2 of 2)

Protocol	I/O Standard	Coupling	Termination
PIPE	1.2-V PCML, 1.5-V PCML, 3.3-V PCML, Differential LVPECL, LVDS	AC	On-chip
	HCSL *(1)*	DC	External *(2)*

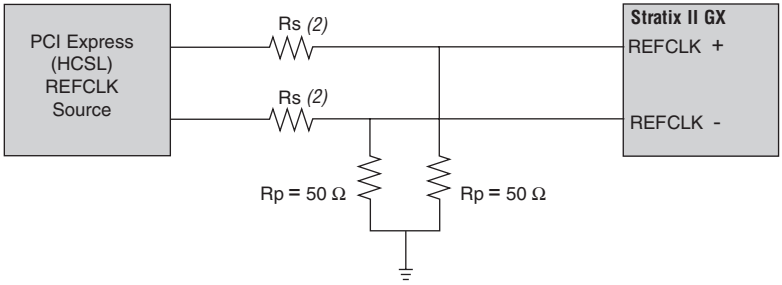
Notes to Table 2–3:

(1) In PIPE mode, you have the option of selecting the HCSL standard for the reference clock if compliance to PCI Express is required. The Quartus II software automatically selects DC coupling with external termination for the `REFCLK` signal if configured as HCSL.

(2) Refer to Figure 2–3 for an example termination scheme.

Figure 2–3 shows an example termination scheme for a reference clock signal when configured as HCSL.

Figure 2–3. DC Coupling and External Termination Scheme for PCI Express Reference Clock *Notes (1), (2)*



Notes to Figure 2–3:

(1) No biasing is required if the reference clock signals are generated from a clock source that conforms to the PCI Express specification.

(2) Select resistor values as recommended by the PCI Express clock source vendor.

Transmitter PLL Block

The transmitter PLL block contains two transmitter PLLs (transmitter PLL0 and transmitter PLL1) per transceiver block, as shown in Figure 2–4. The transmitter PLL block multiplies the reference clock to the frequency required to support the serial data rate. Transmitter PLL0 and transmitter PLL1 can support data rates from 600 Mbps to 6.375 Gbps. Each PLL has a dedicated locked signal (`pll_locked`) that is fed to the PLD logic array to indicate when the PLLs are locked to the reference clock.

You can use transmitter PLL0 and transmitter PLL1 individually (one PLL active at a time) to provide a base high-speed clock to the entire transceiver block, or simultaneously to provide support for the different data rates within the transceiver block that does not have a common base reference clock frequency. For example, one PLL can support a 1.25 Gbps data rate with a 125 MHz reference clock and the other PLL can support a 2.488 Gbps data rate with a 62.2 MHz reference clock.

You can use up to two reference clocks for the transmitter PLLs in a single transceiver block at any given time. The reference clocks can come from the following:

- Dedicated reference clock pins of the associated transceiver blocks (two total per transceiver block)
- PLD clock network (one per transceiver block, must be connected directly from an input clock pin and cannot be driven by user logic or enhanced PLL)
- Inter-transceiver lines (up to five total, one from each transceiver block)

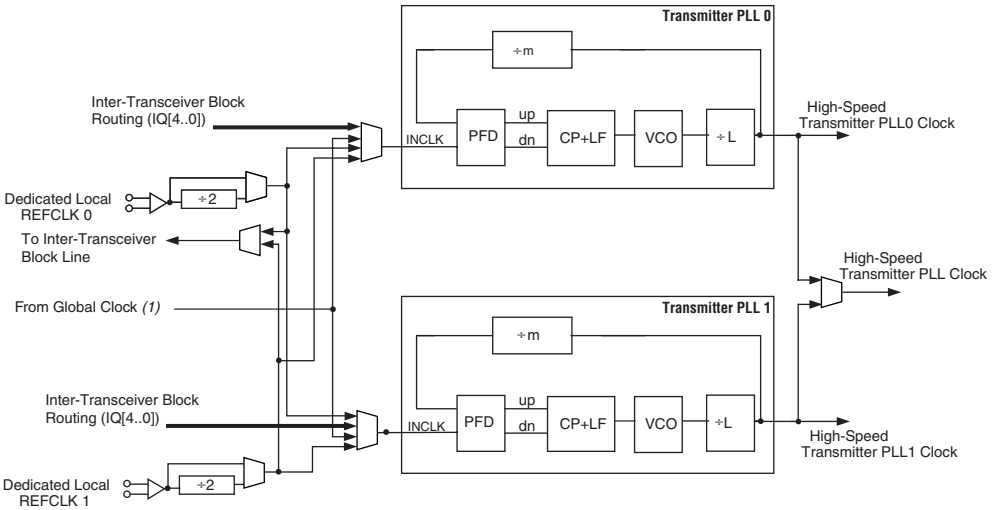
If you assign an I/O or a non-`REFCLK` clock pin to provide clock ONLY for the `pll_inclk/rx_cruclk` ports of the transceiver, the Quartus II software requires the following setting for the clock source in the assignment editor for successful compilation:

Assignment name: Stratix II GX REFCLK and termination setting

Value: Use as regular I/O.

Figure 2–4 shows the transmitter PLL block.

Figure 2–4. Transmitter PLL Block



Note to Figure 2–4:
(1) The Global Clock line must be driven by an input pin.

Transmitter PLLs

There are two transmitter PLLs in each CMU (transmitter PLL0 and transmitter PLL1). Transmitter PLL0 and transmitter PLL1 receive the reference clock from one of five inter-transceiver lines (refer to "Inter-Transceiver Line Routing" on page 2–19 for more information), a global PLD clock driven from a clock input pin, or from the dedicated reference clock `REFCLK0` or `REFCLK1` (both reference clock pins can drive either transmitter PLL0 or transmitter PLL1). You can divide the reference clocks from the `REFCLK` pins by two to support higher reference clock frequencies.

Transmitter PLL0 and transmitter PLL1 have half-rate VCOs that operate at half the rate of the serial data stream. The range of these VCOs are from 500 MHz to 3.1875 GHz to support a native data rate of 1 Gbps to 6.375 GHz. Lower data rates (600 Mbps to 1 Gbps) are supported via additional clock dividers (refer to "Clock Synthesis" on page 2–16 for more information).

The PLL contains two multiplier blocks in the PLL feedback loop to multiply the reference clock to support the required data rate. The Quartus II software automatically selects the values for all the dividers. You must input a data rate and select the input clock frequency.

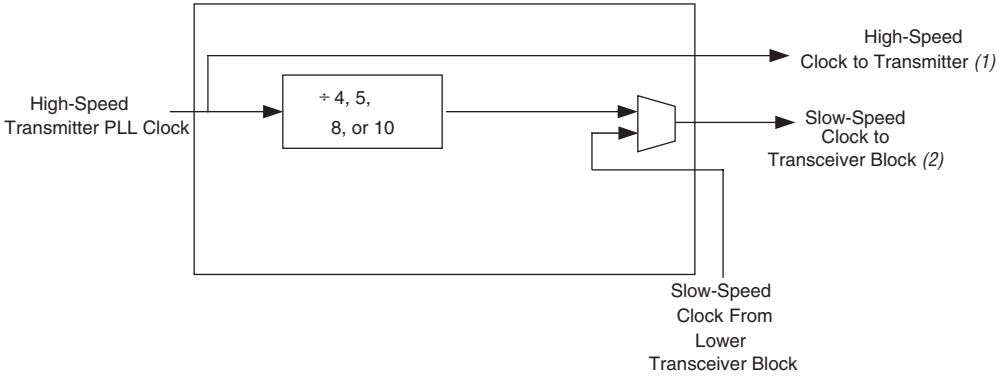
The PLL output feeds the central clock divider block through the high-speed transmitter PLL clock multiplexer or feeds the transmitter local clock divider block in each transmitter channel through the high-speed transmitter PLL clocks.

Central Clock Divider Block

The central clock divider block is located in the central block of the transceiver block (refer to Figure 2–2). This block provides the high-speed clock for the serializer and the low-speed clock for the transceiver's PCS logic within the transceiver block in a four-lane mode. In Physical Interface for PCI Express (PIPE) ×8 mode, the central clock divider block also provides the high-speed clock and low-speed clock for the adjacent upper transceiver block and provides the high- and low-speed clocks to the associated transceiver block. The PLLs, central clock divider block, and the transmitter local clock dividers are powered down in the adjacent upper transceiver block in an eight-lane configuration.

Figure 2–5 shows the central clock divider block. The / 4, /5, /8, and /10 block generates the slow-speed clock based on the serialization factor. In an eight-lane configuration in PIPE mode, the slow-speed clock is multiplexed from the lower transceiver block. The high-speed clock goes directly into each channel's serializer through a clock multiplexer.

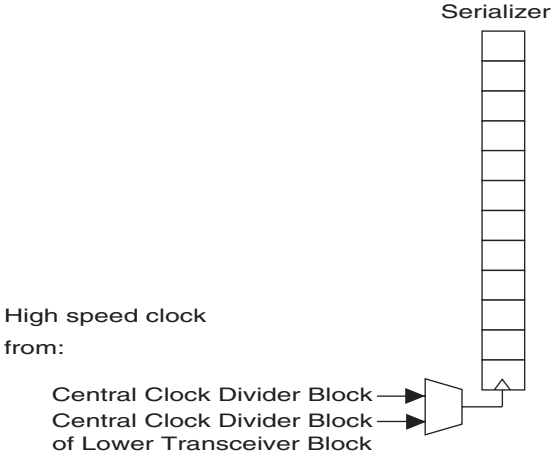
Figure 2–5. Central Clock Divider Block



Notes to Figure 2–5:
(1) Feeds the transmitter within the transceiver and the above adjacent transceiver block.
(2) Feeds the PCS logic.

Figure 2–6 shows the clock selection for the serializer.

Figure 2–6. Serializer High-Speed Clock Connection



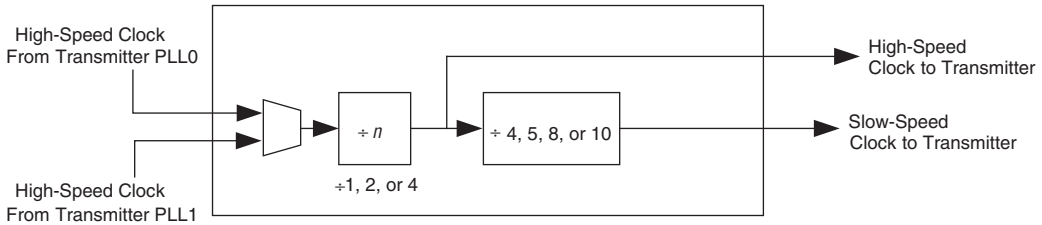
The central clock divider block feeds all the channels in the transceiver block and, in PIPE ×8 mode, it also feeds the adjacent upper transceiver block. This ensures that the serializer in each channel outputs the same bit number at the same time and minimizes the channel-to-channel skew.

Transmitter Local Clock Divider Block

The TX local clock divider blocks are located in each transmitter channels of the transceiver block. The purpose of this block is to provide the high-speed clock for the serializer and the low-speed clock for the transmitter data path and the PLD for all the transmitters within the transceiver block. This allows for each of the transmitter channels to run at different rates. The /n divider offers a /1, /2, and /4 factors to provide capability to reduce base frequency of the driving PLL to a half or a quarter rate. This allows each transmitter channel to run at a /1, /2, or /4 of the original data rate.

Figure 2–7 shows the transmitter local clock divider block.

Figure 2–7. Transmitter Local Clock Divider Block



Each transmitter local clock divider block is operated independently, so there is no guarantee that each channel sends out the same bit at the same time.

Clock Synthesis

Each PLL in a transceiver block receives a reference clock and generates a high-speed clock that is forwarded to the clock generator blocks. There are two types of clock generators:

- the transmitter local clock divider block
- the central clock divider block

The transmitter local clock divider block resides in the transmit channel and synthesizes the high-speed serial clock (used by the serializer) and slow-speed clock (used by the transmitter's PCS logic). The central clock divider block resides in the transceiver block outside the transmit or receive channels. This block synthesizes the high-speed serial clock (used by the serializer) and slow-speed clock (used by the transceiver block PCS logic—transmitter and receiver (if the rate matcher is used). The PLD

clock is also supplied by the central clock divider block and goes through the divide-by-two block (located in the central block of the transceiver block) if the byte serializer/deserializer is used.

The PLLs in the transceiver have half rate VCOs that run at half the rate of the data stream. When in the individual channel mode, the slow-speed clocks for the transmitter logic and the serializer need only be / 4 or a / 5 divider to support a ×8 and ×10 serialization factor. The ×16 and ×20 serialization factor is supported by the / 8 and / 10 clock divider. Table 2–4 shows the divider settings for achieving the available serialization factor.

Table 2–4. Serialization Factor and Divider Settings

Serialization Factor	Divider Setting
× 8	/ 4
× 10	/ 5
× 16	/ 8
× 20	/ 10

In the four-lane mode, the central clock divider block supplies all the necessary clocks for the entire transceiver block.

The reference clock ranges from 50 MHz to 622.08 MHz. The phase frequency detector (PFD) has a minimum frequency limit of 50 MHz and a maximum frequency limit of 325 MHz.

The `refclk` pre-divider (/2) is available if you use the dedicated `refclk` pins for the input reference clock. The `refclk` pre-divider is required if **one** of the following conditions is satisfied:

1. If the input clock frequency is greater than 325 MHz.
2. For functional modes with a data rate less than 3.125 Gbps (the data rate is specified in the **what is the data rate?** option in the **General** tab of the ALT2GXB MegaWizard):
   - If the input clock frequency is greater than or equal to 100 MHz AND
   - If the ratio of data rate to input clock frequency is 4, 5, or 25

3. For functional modes with an data rate greater than 3.125 Gbps:

   - If the input clock frequency is greater than or equal to 100 MHz AND
   - If the ratio of data rate to input clock frequency is 8, 10, or 25

Table 2–5 shows the `refclk` pre-divider and the available PLL multiplication factors.

Table 2–5. Multiplication Values as a Function of the Reference Clock Source to the Transmitter PLL

Transmitter PLL Reference Clock Source	Reference Clock Pre-Divider	/ M *(2)*	/ L
Inter-transceiver routing	1, 2 *(1)*	1, 4, 5, 8, 10, 16, 20, 25	1, 2, 4
Dedicated local reference clock	1, 2	1,4,5,8,10,16,20, 25	1, 2, 4

Notes to Table 2–5:

(1) The / 2 is achieved by using the pre-divider on the driving `REFCLK` pin.

(2) The M, L counters are automatically selected by the Quartus II software based on the selected data rate and protocol and the reference clock frequency.

Table 2–6 shows the multiplication values for Basic mode.

Table 2–6. Multiplication Values Allowed in Basic Mode

Protocol Functional Mode	Reference Clock Pre-Divider	/ M
Basic single width	1, 2	4, 5, 8, 10, 16, 20, 25
Basic double width (1 Gbps to 3.125 Gbps)	1, 2	4, 5, 8, 10, 16, 20, 25
Basic double width (> 3.125 Gbps to 6.375 Gbps)	1, 2	8, 10, 16, 20, 25

Transmitter PLL Bandwidth Setting

The Stratix II GX transmitter PLLs in the transceiver offer a programmable bandwidth setting. The bandwidth of a PLL is the measure of its ability to track the input clock and jitter. It is determined by the –3dB frequency of the closed-loop gain of the PLL.

There are three bandwidth settings: high, medium, and low. The high bandwidth setting filters out internal noise from the VCO because it tracks the input clock above the frequency of the internal VCO noise. With the low bandwidth setting, if the noise on the input reference clock

is greater than the internal noise of the VCO, the PLL filters out the noise above the –3dB frequency of the closed-loop gain of the PLL. The medium bandwidth setting is a compromise between the high and low settings.

The –3dB frequencies for these settings can vary because of the non-linear nature and frequency dependencies of the circuit.

Inter-Transceiver Line Routing

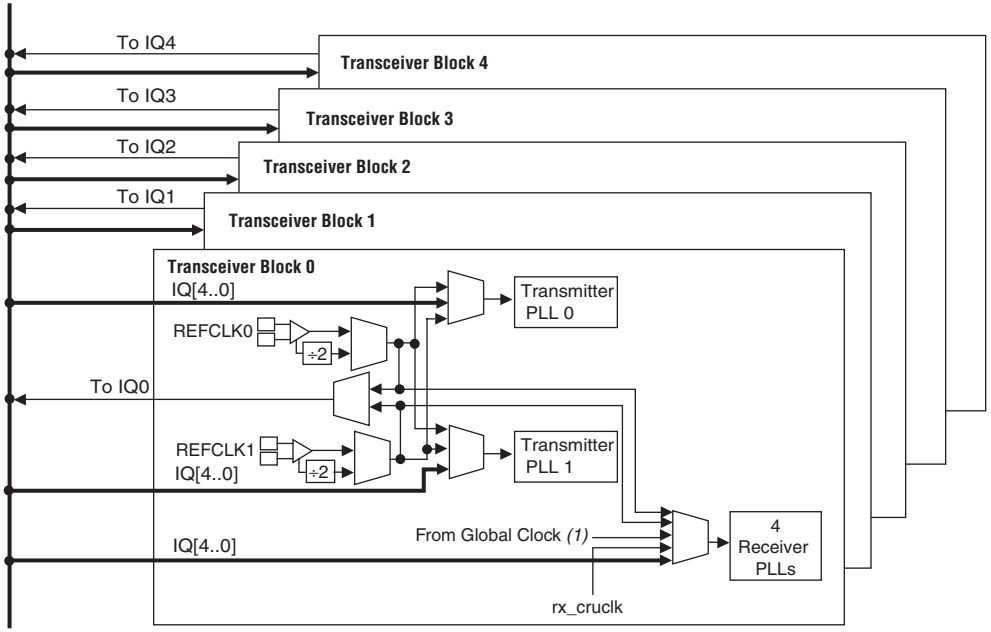
The inter-transceiver lines allow the REFCLK of one transceiver block to drive other transceiver blocks. There are a maximum of five inter-transceiver clock routing lines available in each device in the Stratix II GX family. Each transceiver block can drive one inter-transceiver line from either one of its associated REFCLK pins. The inter-transceiver lines can drive any or all of the transmitter PLLs and receiver PLLs in the device. The inter-transceiver lines offer flexibility when multiple channels in separate transceiver blocks share a common reference clock frequency. The inter-transceiver lines only distribute the reference clock and cannot be used to bond channels because each PLL and clock dividers operate independently. If you select an input reference clock frequency such that it requires the use of the REFCLK pre-divider, you cannot use the clock on REFCLK pins to drive PLD logic.

The inter-transceiver lines also drive the reference clock from the REFCLK pins into the PLD, which reduces the need to drive multiple clocks of the same frequency into the device.

The Quartus II software automatically uses the appropriate inter-transceiver line if the transceiver block is being clocked by the dedicated reference clock (REFCLK) pin of another transceiver block.

Figure 2–8 shows the inter-transceiver line interface to the transceivers in the gigabit transceiver blocks and to the PLD. The connections of transceiver block 0 are shown. The other connections are the same, with the exception of the inter-transceiver line number that the transceiver block drives.

Figure 2–8. Inter-Transceiver Lines of a Five Transceiver Block Device (2SGX130G)



Note to Figure 2–8:
(1) The Global Clock line must be driven by an input pin.

Transceiver Clock Distribution

This section describes single-lane, four-lane, and eight-lane configurations for the high-speed and low-speed transceiver clocks. All protocol support falls in the single-lane configuration except for the four-lane and eight-lane PIPE mode and XAUI. The four-lane PIPE mode uses the four-lane configuration. The eight-lane PIPE mode uses the eight-lane configuration.

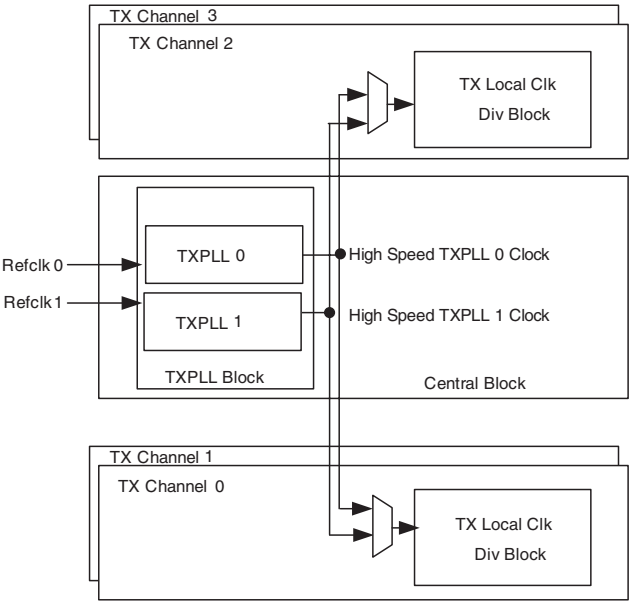
Single Lane

In a single-lane configuration, the PLLs in the central block supply the high speed clock and the clock generation blocks in each transmitter channel divides down that clock to the frequency needed to support its particular data rate. In this configuration, two separate clocks can be supplied through the central block to provide support for two separate base frequencies. The transmitter clock generation blocks can divide those down to create additional frequencies for specific data rate

requirements. Each of the four transmitter channels can operate at a different data rate with the use of the individual transmitter local clock dividers and both Transmitter PLL0 and Transmitter PLL1.

If you instantiate four channels and are not in PIPE ×4, XAUI, or Basic single-width mode with ×4 clocking, the Quartus II software automatically chooses the single-lane configuration.

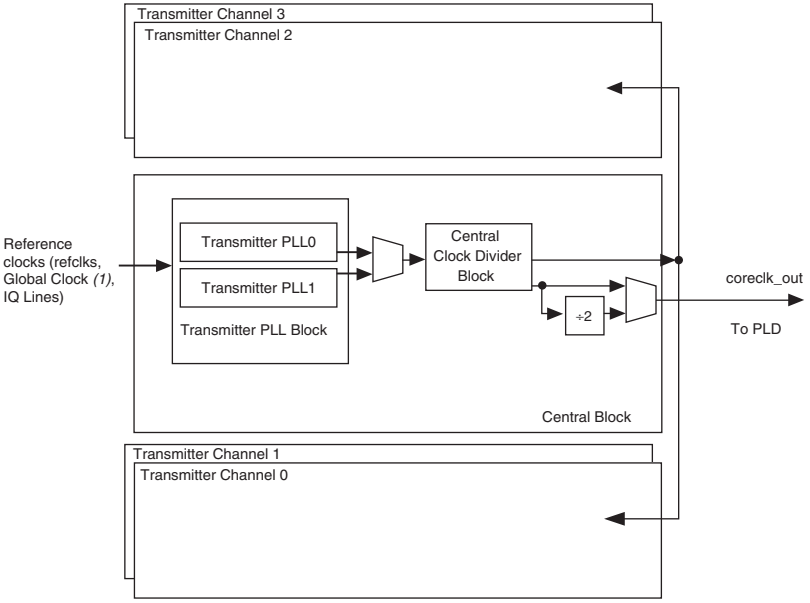
Figure 2–9. Clock Distribution for Individual Channel Configuration



Four-Lane Mode

In a four-lane configuration (Figure 2–10), the central block generates the parallel and serial clocks that feed the transmitter channels within the transceiver. All channels in a transceiver must operate at the same data rate. This configuration is only supported in PIPE ×4, XAUI and Basic single-width mode with ×4 clocking.

Figure 2–10. Clock Distribution for a Four-Lane Configuration

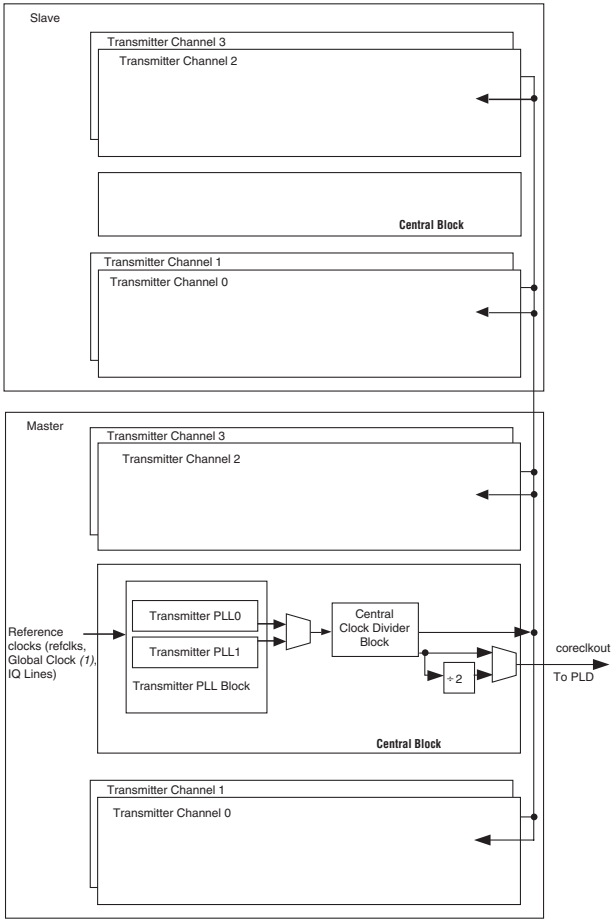


Note to Figure 2–10:
(1) The Global Clock line must be driven by an input pin.

Eight-Lane Mode

The eight-lane mode (refer to Figure 2–11) is reserved for PIPE ×8 use only. The central block of the lower transceiver supplies the parallel and serial clocks for all eight transmitter channels. The clock distribution uses a dedicated eight-lane clocking routing that offers low skew for transmitter channel-to-channel skew specification. The high- and low-speed clocks are forwarded using this dedicated eight-lane clocking tree. The central block of the upper transceiver block and all the transmitter clock generation blocks are unused and are powered down in this mode. The clock to the PLD (`coreclkout`) is generated by the central clock generation block of the master transceiver block (the lower transceiver block).

Figure 2–11. Clock Distribution for Eight-Lane Mode

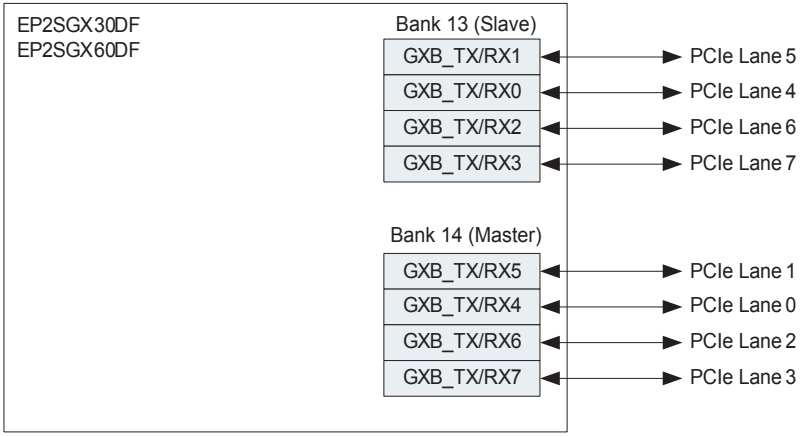


Note to Figure 2–11:
(1) The Global Clock line must be driven by an input pin.

Only designated lower transceiver blocks can be used as a master (transceiver blocks 1 and 3), and designated upper transceiver blocks (transceiver blocks 0 and 2) can be used as a slave as long as they are coupled to the lower master transceiver block. The Quartus II software automatically utilizes the correct transceiver blocks in a ×8 mode if you do not assign placement. If you do not place the master and slave transceiver blocks accordingly (through pin assignments), a no fit error occurs.

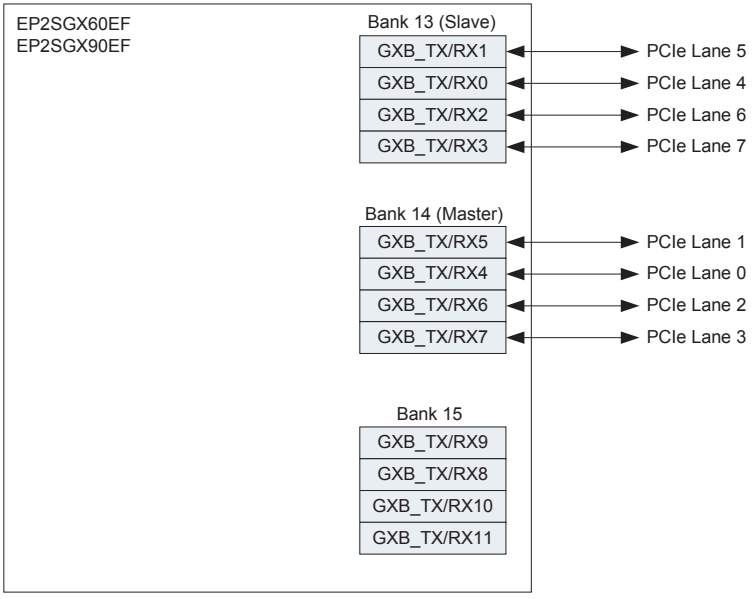
Single transceiver block devices (EP2SGX30C and EP2SGX60C) cannot be used for PCI-E ×8 mode. Figure 2–12 shows how double transceiver block devices (EP2SGX30D and EP2SGX60D) are configured for PCI-E ×8 mode.

Figure 2–12. Two Transceiver Block Device With One ×8 PCI-E Link



The three transceiver block devices (EP2SGX60E and EP2SGX90E) support only one PCI-E ×8 link. Figure 2–13 shows the PCI-E ×8 configuration.

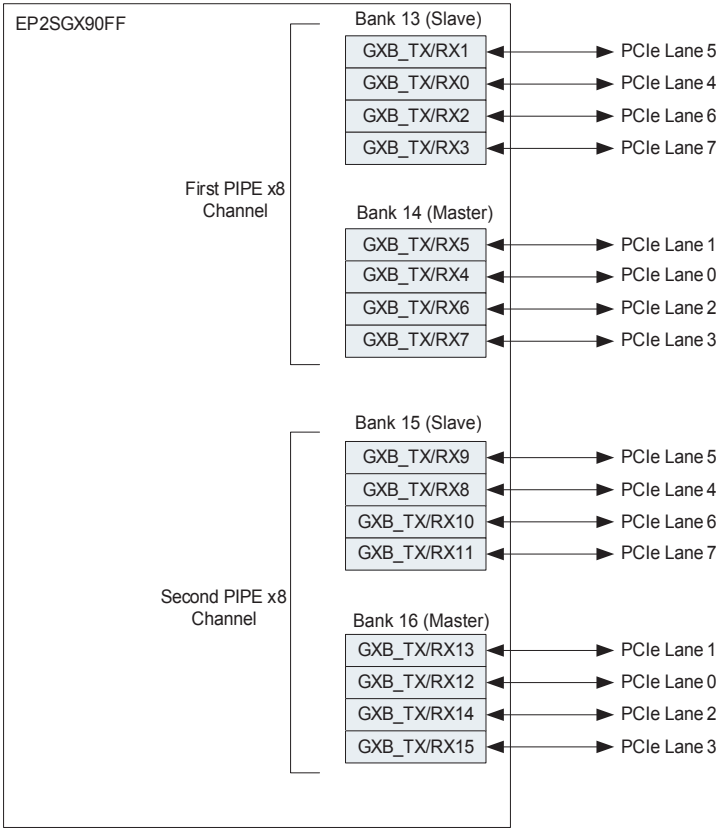
Figure 2–13. Three Transceiver Block Device With One ×8 PCI-E Link *Note (1)*



Note to Figure 2–13:
(1) Transceiver Bank 15 can be active and used to support other protocols.

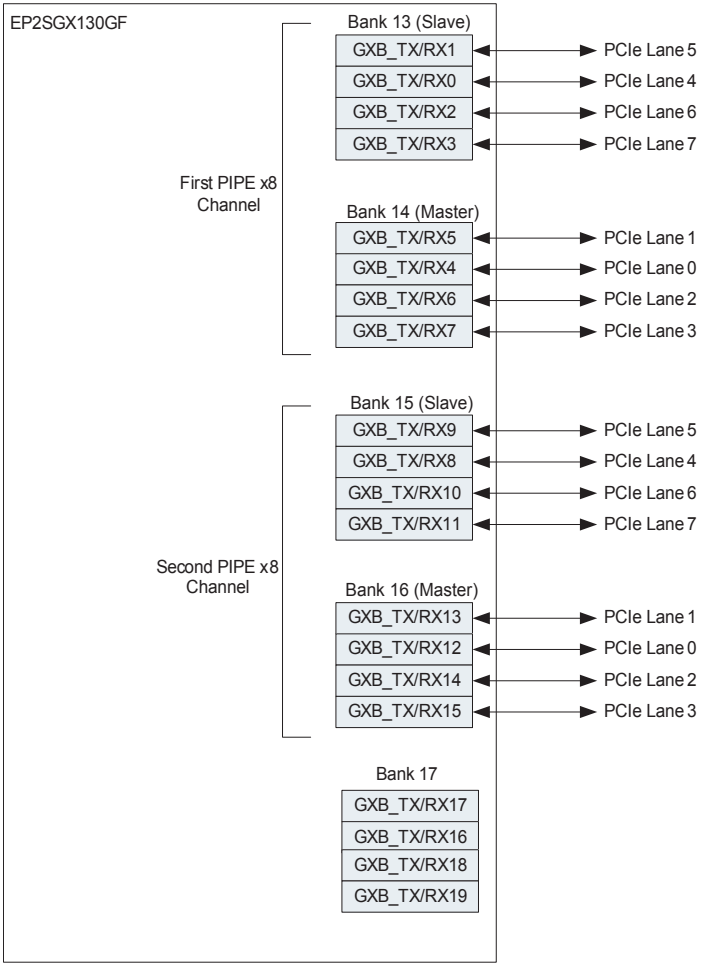
A four transceiver block device (EP2SGX90F) supports up to two PCI-E ×8 links (refer to Figure 2–14). The transceiver block pairs are blocks 0 and 1 and blocks 2 and 3. If only one PCI-E ×8 link is used, the other transceiver blocks can be active and be used to support other protocols.

Figure 2–14. Four Transceiver Block Device With Two ×8 PCI-E Links



A five transceiver device (EP2SGX130G) supports up to two PCI-E ×8 links (refer to Figure 2–15). The transceiver block pairs are the same as in the four transceiver device—blocks 0 and 1 and blocks 2 and 3. Block 4 is not used for PCI-E ×8 mode. If any transceiver blocks are not used in the PCI-E ×8 mode, they can be used to support any other protocol.

Figure 2–15. Five Transceiver Block Device With Two ×8 PCI-E Links *Note (1)*



Note to Figure 2–15:
(1) Transceiver Bank 17 can be active and used to support other protocols.

Channel Clock Distribution

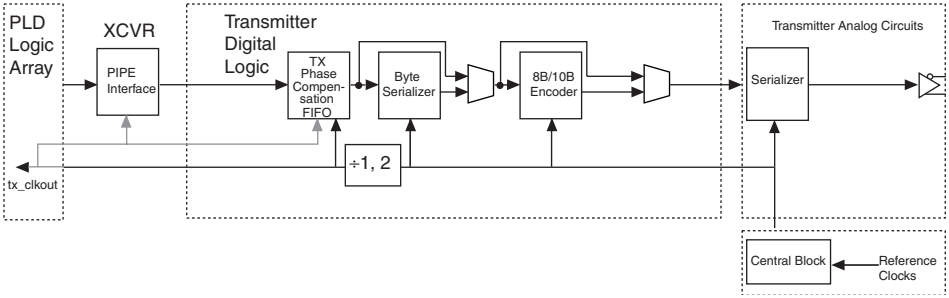
This section describes clocking within each channel for:

- Individual channels in Basic (without ×4 clocking enabled), SONET/SDH, PIPE x1, GIGE, (OIF) CEI PHY Interface (with low-jitter option disabled), Serial RapidIO, SDI, and CPRI modes
- Bonded channels in XAUI, PIPE ×4, PIPE ×8, Basic (with ×4 clocking enabled), and (OIF) CEI PHY Interface (with low-jitter option enabled) modes

Individual Channels Clocking

In individual channel modes, the transmitter logic is clocked by the slow-speed clock from the clock divider block. The transmitter phase compensation FIFO buffer and the PIPE interface (in PIPE mode) are clocked by the `tx_clkout` clock of the channel that is fed back to the transmitter channel from the PLD logic. Figure 2–16 shows the clock routing for the transmitter channel.

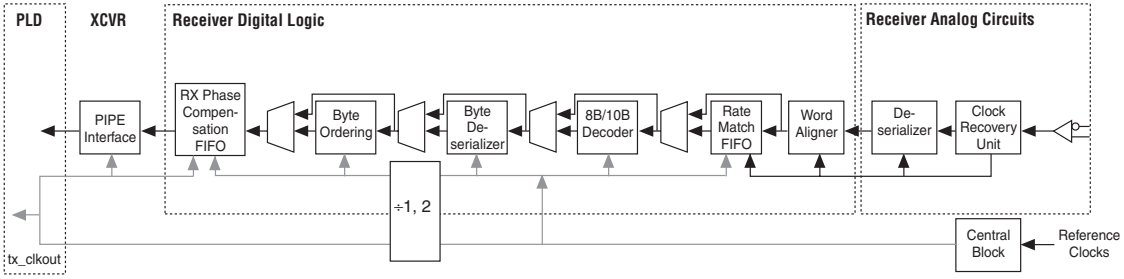
Figure 2–16. Individual Channel Transmitter Logic Clocking



The receiver logic clocking has two clocking methods: one when rate matching is used and the other when rate matching is not used.

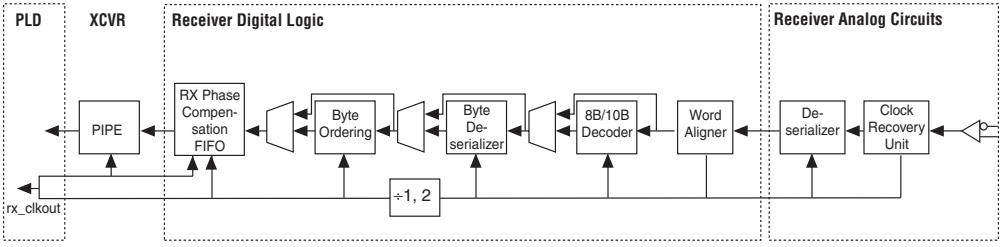
If rate matching is used (PIPE, GIGE, and Basic modes), the receiver logic from the serializer to the rate matcher is clocked by the recovered clock from its associated channel. The rest of the logic is clocked by the slow clock from the clock divider block of its associated channel. The read side of the phase compensation FIFO buffer and the PIPE interface (for PIPE mode) is clocked by the `tx_clkout` fed back through the PLD logic. Figure 2–17 shows the clocking of the receiver logic with the rate matcher.

Figure 2–17. Individual Channel Receiver Logic Clocking With Rate Matching



If rate matching is not used (Basic, SONET/SDH, CPRI, (OIF) CEI PHY Interface, SDI, Serial RapidIO modes), then the receiver logic is clocked by the recovered clock of its associated channel (Figure 2–18). The receiver phase compensation FIFO buffer's read port is clocked by the recovered clock that is fed back from the PLD logic array as `rx_clkout`.

Figure 2–18. Individual Channel Receiver Logic Clocking Without Rate Matching



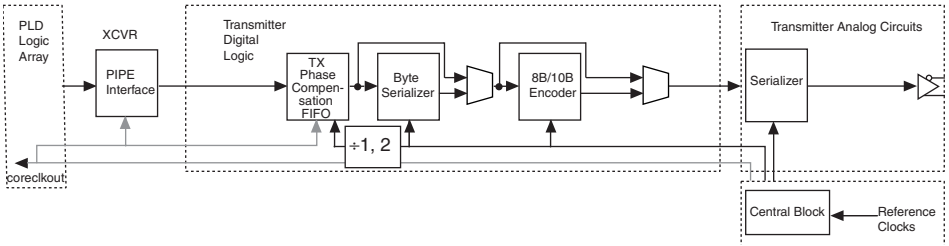
Transmitter Clocking (Bonded Channels)

The clocking in bonded channel modes (Figure 2–19) is different from that of the individual channel. All the transmitters are synchronized to the same transmitter PLL and clock divider from the central block. In ×4 bonded channel modes, the central clock divider of the transceiver block clocks all 4 channels. In PIPE ×8 bonded channel mode, the central clock divider of the master transceiver block clocks all 8 channels.

The transmitter logic up to the read port of the transmitter phase compensation FIFO buffer is clocked by the slow-speed clock from the central block. The PIPE interface and the write port of the transmitter phase compensation FIFO buffer is clocked by the `coreclkout` signal routed from the PLD. In the PIPE ×8 slave transceiver, the central block of the associated transceiver is not active and the transmitter logic to the read port of the transmitter phase compensation FIFO buffer is clocked by

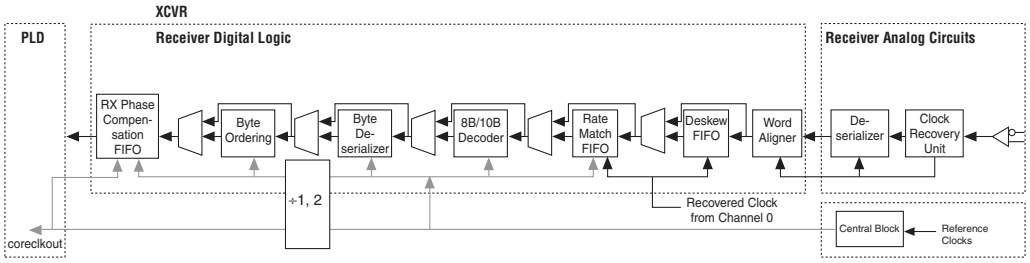
the slow-speed clock from the master transceiver. The PIPE interface and the write port of the transmitter phase compensation FIFO buffer is clocked by the `coreclkout` signal of the master transceiver. The slave transceiver does not output a `coreclkout` signal because the CMU is powered down.

Figure 2–19. Transmitter Channel Clocking in Transceiver Mode



For the receiver logic, in XAUI mode (Figure 2–20), the local recovered clock feeds the logic up to the write clock of the deskew FIFO buffer. The recovered clock from `channel[0]` feeds the read clock of the deskew FIFO buffer and the write port of the rate matcher. The slow clock from the central block feeds the rest of the logic up to the write port of the phase compensation FIFO buffer. The `coreclkout` signal routed through the PLD from the central block feeds the read side of the phase compensation FIFO buffer.

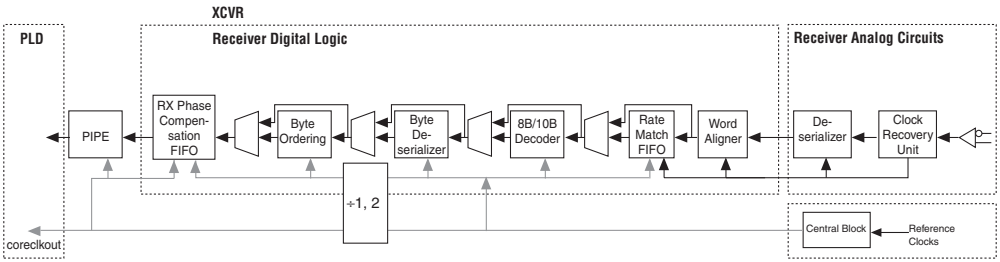
Figure 2–20. Receiver Channel Clocking in XAUI Mode



In the PIPE ×4 and PIPE ×8 modes (Figure 2–21), the local recovered clock feeds the logic up to the write port of the rate matcher FIFO buffer. The slow clock from the central block feeds the rest of the logic up to the write port of the phase compensation FIFO buffer. The `coreclkout` signal routed through the PLD from the central block feeds the read side of the phase compensation FIFO buffer. In PIPE ×8, the slave transceiver takes

the clocks from the central block of the master transceiver. The individual channel clock distribution of the slave transceiver is the same as the master transceiver.

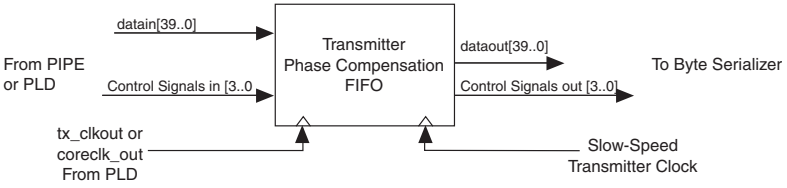
Figure 2–21. Receiver Channel PIPE ×4 and × 8 Modes



Transmitter Phase Compensation FIFO Buffer

The transmitter phase compensation FIFO buffer (Figure 2–22) is located at the device's logic array interface in the transmitter block and compensates for phase difference between the transmitter clock and the clock from the PLD. The transmitter phase compensation FIFO buffer operates in two modes: low latency and high latency. In the low latency mode, the FIFO buffer is four words deep. The Quartus II software chooses low latency mode automatically for every mode except the PCI-Express PIPE mode (which automatically uses high latency mode). In high latency mode, the FIFO buffer is eight words deep.

Figure 2–22. Transmitter Phase Compensation FIFO Buffer



The phase compensation FIFO buffer's read port is clocked by the transmitter PLL clock. The write clock is fed by `tx_clkout` of the associated channel in a single-channel configuration. The FIFO buffer's write clock is clocked by `coreclkout` in the four- or eight-channel configuration.

The transmitter phase compensation FIFO buffer is always used and cannot be bypassed. The input to the transmitter phase compensation FIFO buffer is the data from the PLD logic array or the PIPE interface in PIPE mode.

Transmitter Phase Compensation FIFO Error Flag

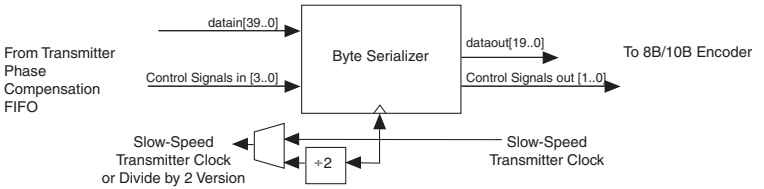
The write port of the transmitter phase compensation FIFO can be clocked by either transmitter PLL output clock (`tx_clkout` or `coreclkout`) or a PLD clock. The read port is always clocked by the transmitter PLL output clock. In all configurations, the write clock and the read clock must have 0 PPM difference to avoid overrun/underflow of the phase compensation FIFO.

An optional `debug_tx_phase_comp_fifo_error` port is available in all modes to indicate transmitter phase compensation FIFO overrun/underflow condition. The `debug_tx_phase_comp_fifo_error` is asserted high when the phase compensation FIFO gets either full or empty. This feature is useful to verify the phase compensation FIFO overrun/underflow condition as a probable cause of link errors.

Byte Serializer

The byte serializer (Figure 2–23) converts the two- or four-byte interface into a one- or two-byte-wide data path for the transceiver from the PLD interface. The PLD interface has a limit of 250 MHz, so the byte serializer is required to stripe the parallel data into the single- or double-wide transceiver data path. At 6.375 Gbps, the transceiver logic has a double-byte-wide data path that runs at 318.75 MHz in a ×20 serializer factor, which is above the maximum PLD interface speed. By using the byte serializer, the PLD interface width doubles to 40-bits (36 bits when using the 8B/10B encoder) and the interface speed drops to 159.375 MHz.

Figure 2–23. Byte Serializer



The byte serializer takes in a 40-, 32-, 20-, or 16-bit-wide input from the phase compensation FIFO buffer and serializes it to 20, 16, 10, or 8 bits, respectively (refer to Table 2–7). At the same time, the clock frequency is doubled.

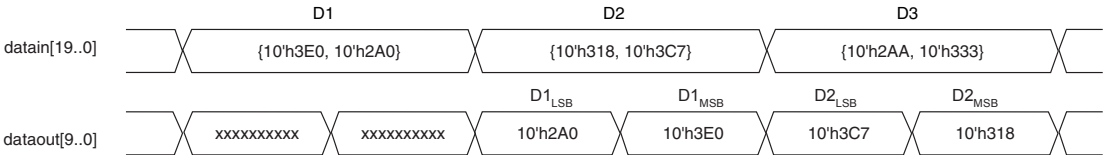
Table 2–7. Byte Serializer Input and Output Data Widths

Input Data Width (Bits)	Output Data Width After Byte Serialization (Bits)
40	20
32	16
20	10
16	8

After serialization, the byte serializer transmits the least significant byte to the most significant byte. Always use the transmitter digital reset to reset the byte serializer FIFO pointers whenever the transmitter PLL loses lock. Refer to "Reset Control and Power Down" on page 2–214 for further details on the reset sequence.

Figure 2–24 shows byte serializer input and output signals when serializing a 20-bit input to 10 bits. The `tx_datain` signal is the input from the FPGA's logic array that has already passed through the transmitter phase compensation FIFO buffer.

Figure 2–24. Transmitter Byte Serializer



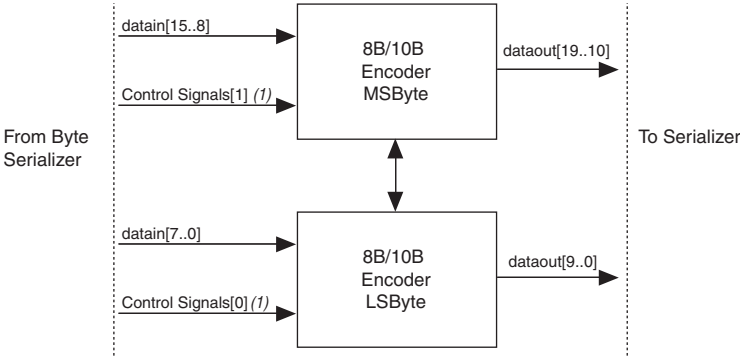
In Figure 2–24, the LSB is transmitted before the MSB in the transmitter byte serializer. For the input of D1, the output is $D1_{LSB}$ and then $D1_{MSB}$.

8B/10B Encoder

The 8B/10B encoder (refer to Figure 2–25) is part of the Stratix II GX transceiver digital blocks and lies between the byte serializer and the serializer. The 8B/10B encoder operates in two modes: single-width and double-width and can be bypassed if the 8B/10B encoder is not used. In single-width mode, the 8B/10B encoder generates a 10-bit code group

from the 8-bit data and 1-bit control identifier. In double-width mode, there are two 8B/10B encoders that are cascaded together to generate two 10-bit code groups from two 8-bit data and their respective control identifiers. The 8B/10B encoder conforms to the IEEE 802.3 1998 edition standards.

Figure 2–25. 8B/10B Encoder

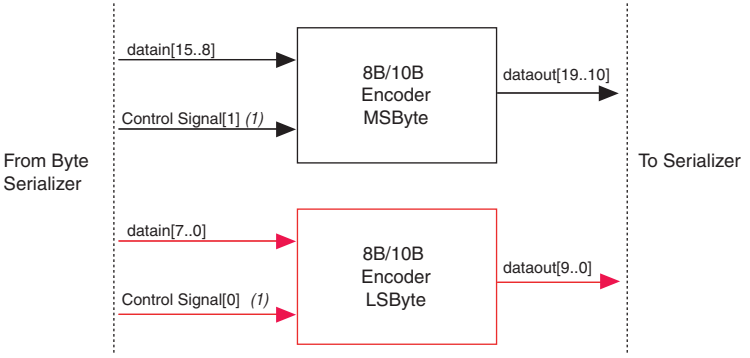


Note to Figure 2–25:
(1) The control signal is `tx_ctrlenable`.

Single-Width Mode

The 8B/10B encoder data path active in single-width mode is highlighted in Figure 2–26.

Figure 2–26. 8B/10B Encoder, Single-Width Mode



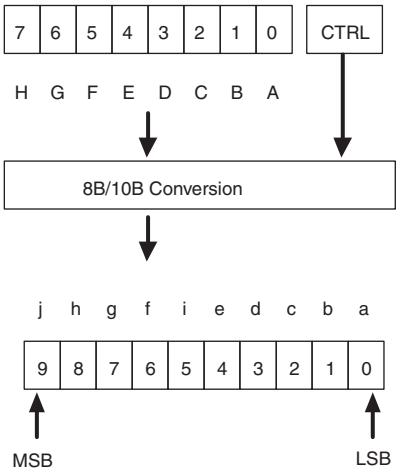
Note to Figure 2–26:
(1) The control signal is `tx_ctrlenable`.

10-Bit Encoding

In single-width mode, the 8B/10B encoder translates the 8-bit data or 8-bit control character (as qualified by the control identifier) to a 10-bit code group with proper disparity. Figure 2–27 shows the conversion format. The LSB is transmitted first.

For additional information regarding the 8B/10B code itself, refer to the *Specifications & Additional Information* chapter in volume 2 of the *Stratix II GX Handbook*.

Figure 2–27. 8B/10B Conversion Format



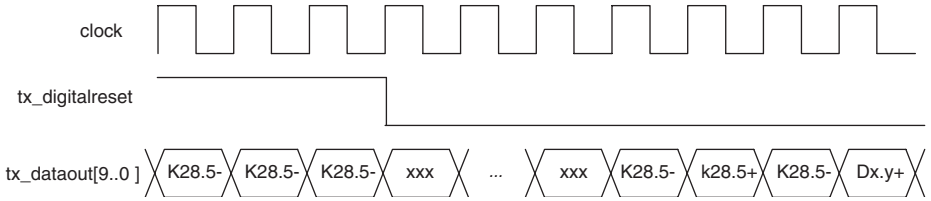
Reset Condition

The `tx_digitalreset` signal resets the 8B/10B encoder. During reset, the running disparity registers and the data registers are cleared. The 8B/10B encoder outputs a K28.5 pattern from the RD- column continuously until `tx_digitalreset` goes low. The input data and `tx_ctrlenable` signals are ignored during the reset state. Once out of reset, the 8B/10B encoder starts with a negative disparity (RD-) and transmits three K28.5 code groups for synchronizing before it starts encoding and transmitting the data on the `tx_datain` port.

While the `tx_digitalreset` signal is asserted, the 8B/10B decoder receives errors in the form of an invalid code error, synchronization error, control detect, and/or disparity error.

Figure 2–28 shows the 8B/10B encoder's reset behavior. When in reset (tx_digitalreset is high), a K28.5- (K28.5 10-bit code group from the RD- column) is sent continuously until tx_digitalreset is low. The transmitter channel pipelining causes some "don't cares (10'hxxx)" until the first of three K28.5 is sent. User data follows the third K28.5.

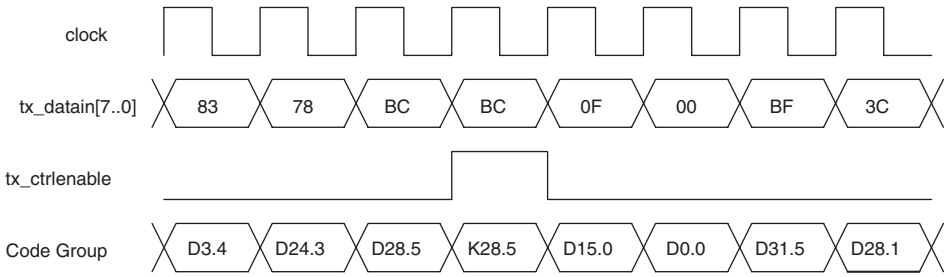
Figure 2–28. Transmitter Output During Reset Conditions



Control Code Encoding

The tx_ctrlenable port identifies if the 8-bit data at the tx_datain port is to be encoded as a control word (Kx.y). If this port is not used, there is no way to send control words out. When tx_ctrlenable is low, the byte at tx_datain port of the transceiver is encoded as data (Dx.y). When tx_ctrlenable is high, the data at the tx_datain port is encoded as a Kx.y code group. The waveform in Figure 2–29 shows that the 2nd 0xBC is encoded as a control word (K28.5). The rest of the tx_datain bytes are encoded as data (Dx.y).

Figure 2–29. Control Word Identification Waveform



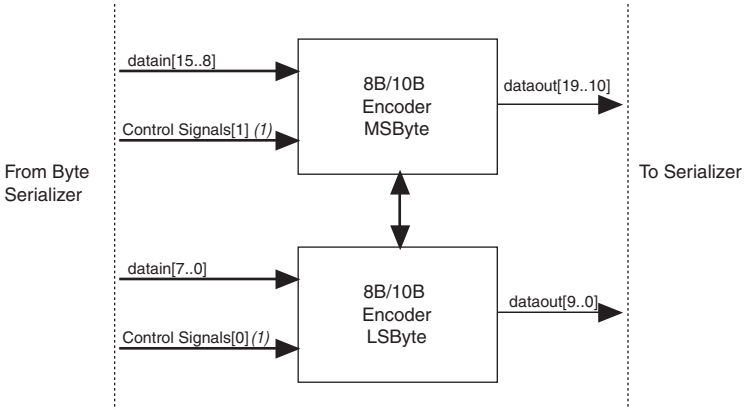
The 8B/10B encoder does not check to see if the code word entered is one of the 12 valid codes. If you enter an invalid control code, the resultant 10-bit code group may be encoded as an invalid code (does not map to a valid Dx.y or Kx.y code), or unintended valid Dx.y code, depending on the value entered.

It is possible for an 8B/10B decoder to decode an invalid control word encoded into a valid Dx.y code without asserting any code error flags. For example, depending on the current running disparity, the invalid code K24.1 (`tx_datain` = 8'h38 + `tx_ctrl` = 1'b1) can be encoded to 10'b0110001100 (0×18C), which is equivalent to a D24.6+ (8'hD8 from the RD+ column). Altera recommends that you do not send invalid control words.

Double-Width Mode

In double-width mode, the 8B/10B encoder operates in a cascaded mode. The least significant byte is transmitted prior to the most significant byte. Figure 2–30 shows the active 8B/10B encoder blocks in double-width mode.

Figure 2–30. Active 8B/10B Encoder Blocks in Double-Width Mode

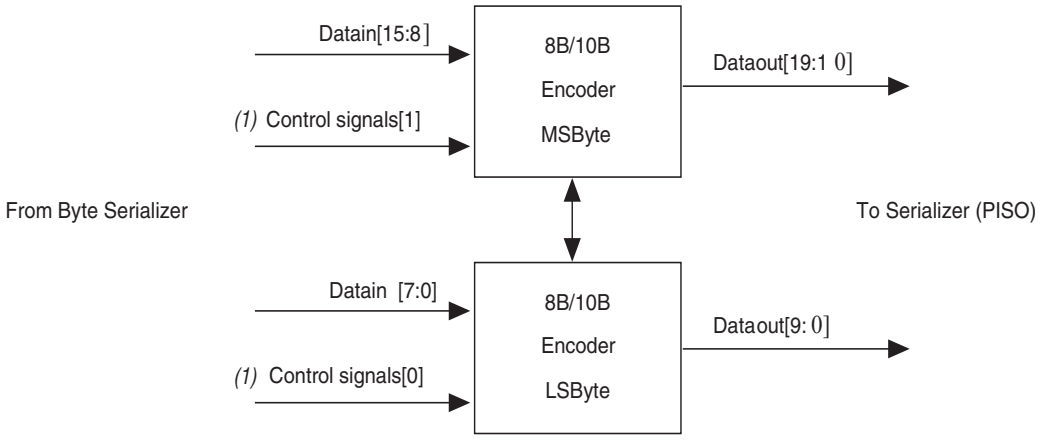


Note to Figure 2–30:
(1) The control signal is `tx_ctrlenable`.

20-Bit Encoding
In double-width mode, the cascaded 8B/10B encoders generate two 10-bit code groups from two 8-bit data and their respective control identifiers. The 8B/10B encoder forwards the current running disparity value from the LSByte encoder to the MSByte encoder to calculate the disparity of the symbol going into the MSByte encoder. The MSByte encoder's ending running disparity is then fed back to the LSByte encoder on the next clock cycle.

Refer to Figure 2–31 for the conversion format. The LSB is transmitted first. Figure 2–31 shows the 20-bit encoding.

Figure 2–31. 8B/10B Conversion Format



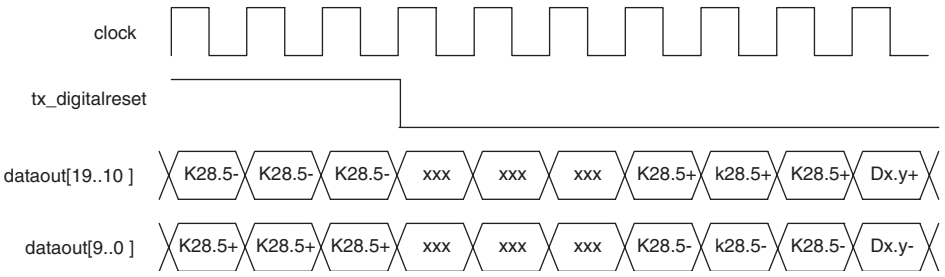
Reset Condition

The `tx_digitalreset` signal resets the 8B/10B encoder. During reset, the running disparity registers and the data registers are cleared. Also, the 8B/10B encoder outputs a K28.5 pattern with proper disparity continuously until `tx_digitalreset` goes low. The `tx_datain` and `tx_ctrlenable` ports are ignored during the reset state. Once out of reset, the 8B/10B encoder starts the LSByte with a negative disparity (RD-) bias and the MSByte with a positive disparity (RD+) and transmits six K28.5 code groups (three on the LSByte and three on the MSByte encoder) for synchronizing before it starts encoding and transmitting the data on `tx_datain`.

If the reset signal for the 8B/10B encoder is asserted, the 8B/10B decoder receiving the data may receive an invalid code error, synchronization error, control detect, and/or disparity error while `tx_digitalreset` is high.

Figure 2–32 shows the reset behavior of the 8B/10B encoder. When in reset (`tx_digitalreset` is high) a K28.5- code group is sent continuously until `tx_digitalreset` is low. Transmitter channel pipelining causes some "don't cares" (10'hxxx) until the first K28.5 is sent (Figure 2–32 shows six don't cares, but the number can vary). Both LSByte and MSByte transmit three K28.5 code groups each before the data at the `tx_datain` port is encoded and sent out.

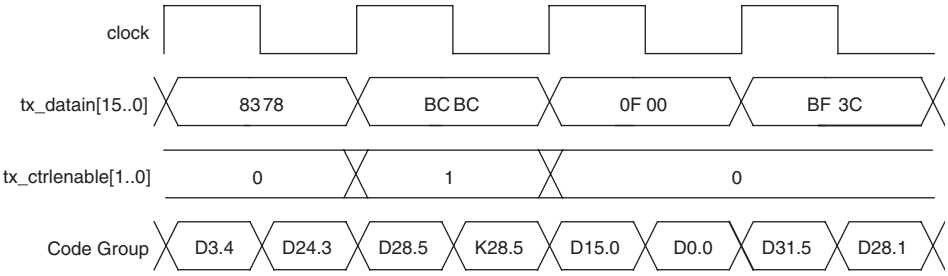
Figure 2–32. Transmitter Output During Reset Conditions



Control Code Encoding

The `tx_ctrlenable` port identifies which 8-bit data is encoded as a control word. If this port is not used, control words cannot be sent. In double-width mode, the `tx_ctrlenable` port has two bits. The lower bit is associated with the LSByte and the upper bit is associated with the MSByte. When `tx_ctrlenable` is low, the byte at the `tx_datain` port of the transceiver is encoded as data (Dx.y), otherwise, it is encoded as a control code (Kx.y). Figure 2–33 shows that the lower byte of the second byte of `tx_datain` (8'hBC) is encoded as a control code as identified by a high on the lower `tx_ctrlenable` bit's second clock cycle.

Figure 2–33. Control Word Identification Waveform



The 8B/10B encoder does not check to see if the code word entered is one of the 12 valid codes. If an invalid control code is entered, the resultant 10-bit code may be encoded as an invalid code (does not map to a valid Dx.y or Kx.y code), or unintended valid Dx.y code, depending on the value entered.

It is possible for an 8B/10B decoder to decode an invalid control word encoded into a valid Dx.y code without asserting any code error flags. For example, depending on the current running disparity, the invalid code

K24.1 (tx_datain = 8'h38 + tx_ctrl = 1'b1) can be encoded to 10'b0110001100 (0x18C), which is equivalent to a D24.6+ (8'hD8 from the RD+ column). Altera recommends that you do not send invalid control words.

Transmitter Force Disparity

Upon power on or reset, the 8B/10B encoder has a negative disparity and will choose the 10-bit code from the RD- column. The Transmitter Force Disparity feature allows altering the running disparity via the tx_forcedisp and tx_dispval ports.

Two optional ports namely tx_forcedisp and tx_dispval ports are available in 8B/10B enabled Basic single-width and Basic double-width modes. In double-width mode, both tx_forcedisp and tx_dispval signals have two bits. The low bit is associated to the LSByte and the high bit is associated to the MSByte. A high value on the tx_forcedisp bit will change the disparity value of the data to the value indicated by the associated tx_dispval bit. If the tx_forcedisp bit is low, then tx_dispval is ignored and the current running disparity is not altered. Forcing disparity can either maintain the current running disparity calculations if the forced disparity value (on the tx_dispval bit) happens to match the current running disparity, or flip the current running disparity calculations if it does not. If the forced disparity flips the current running disparity, the down stream 8B/10B decoder may detect a disparity error which should be tolerated by the down stream device.

Figure 2–34 shows the current running disparity being altered in Basic single-width mode by forcing a positive disparity on a negative disparity K28.5. In this example, a series of K28.5 code groups are continuously being sent. The stream alternates between a positive ending running disparity (RD+) K28.5 and a negative ending running disparity (RD-) K28.5 as governed by the 8B/10B encoder to maintain a neutral overall disparity. The current running disparity at time n+3 indicates that the K28.5 in time n+4 should be encoded with a negative disparity. Since the tx_forcedisp is high at time n+4, and tx_dispval is also high, the K28.5 at time n+4 is encoded as a positive disparity code group. As the tx_forcedisp is low at n+5 the K28.5 will take the current running disparity of n+4 and encode the K28.5 in time n+5 with a negative disparity. If the tx_forcedisp were driven high at time n+5, that K28.5 would also be encoded with positive disparity.

Figure 2–34. Transmitter Force Disparity Feature in Single-Width Mode

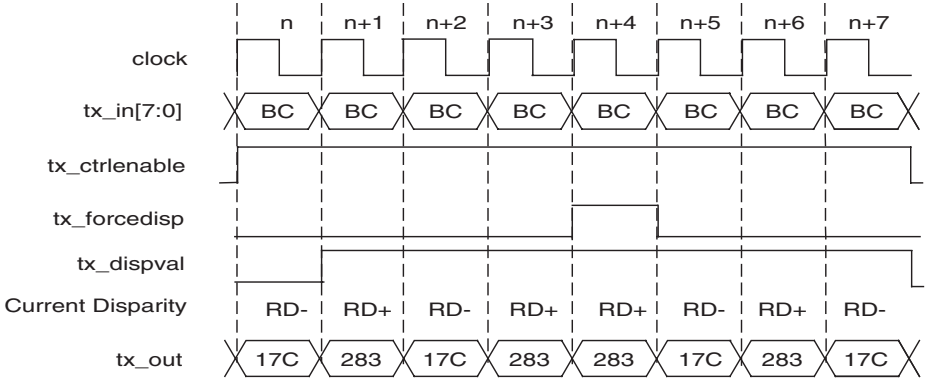
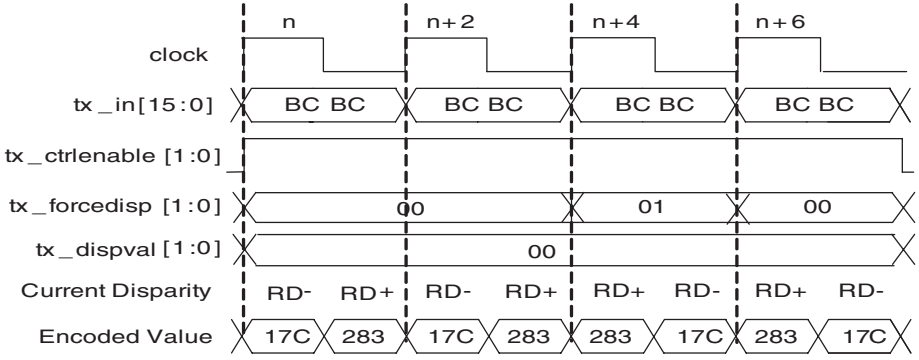


Figure 2–35 shows the current running disparity being altered in Basic double-width mode by forcing a positive disparity on a negative disparity K28.5. In this example, a series of K28.5 are continuously being sent. The stream alternates between a positive ending running disparity (RD+) K28.5 and a negative ending running disparity (RD-) K28.5 as governed by the 8B/10B encoder to maintain a neutral overall disparity. The current running disparity at the end of time n+2 indicates that the K28.5 at the low byte position in time n+4 should be encoded with a positive disparity. Since the tx_forcedisp is high at time n+4, the signal level of tx_dispval is used to convert the lower byte K28.5 to be encoded as a negative disparity code word. As the upper bit of tx_forcedisp is low at n+4 the high byte K28.5 will take the current running disparity dictated by the low byte and encode the upper byte K28.5 with a positive disparity. If the upper bit of tx_forcedisp were driven high in time n+4, the upper byte K28.5 in time n+4 will be encoded with negative disparity.

Figure 2–35. Transmitter Force Disparity Feature in Double-Width Mode



Transmitter Polarity Inversion

The positive and negative signals of a serial differential link might accidentally be swapped during board layout. Solutions like a board re-spin or major updates to the PLD logic can prove expensive. The transmitter polarity inversion feature is provided to correct this situation.

An optional `tx_invpolarity` port is available in all single-width and double-width modes except (OIF) CEI PHY to dynamically enable the transmitter polarity inversion feature. In single-width modes, a high value on the `tx_invpolarity` port inverts the polarity of every bit of the 8-bit or 10-bit input data word to the serializer in the transmitter data path. In double-width modes, a high value on the `tx_invpolarity` port inverts the polarity of every bit of the 16-bit or 20-bit input data word to the serializer in the transmitter data path. Since inverting the polarity of each bit has the same effect as swapping the positive and negative signals of the differential link, correct data is seen by the receiver. The `tx_invpolarity` is a dynamic signal and may cause initial disparity errors at the receiver of an 8B/10B encoded link. The downstream system must be able to tolerate these disparity errors.

Figure 2–36 illustrates the transmitter polarity inversion feature in a single-width 10-bit wide data path configuration.

Figure 2–36. Transmitter Polarity Inversion in Single-Width Mode

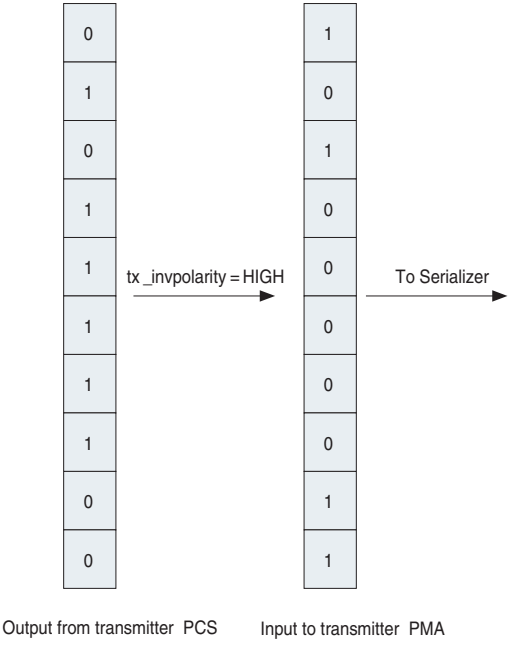
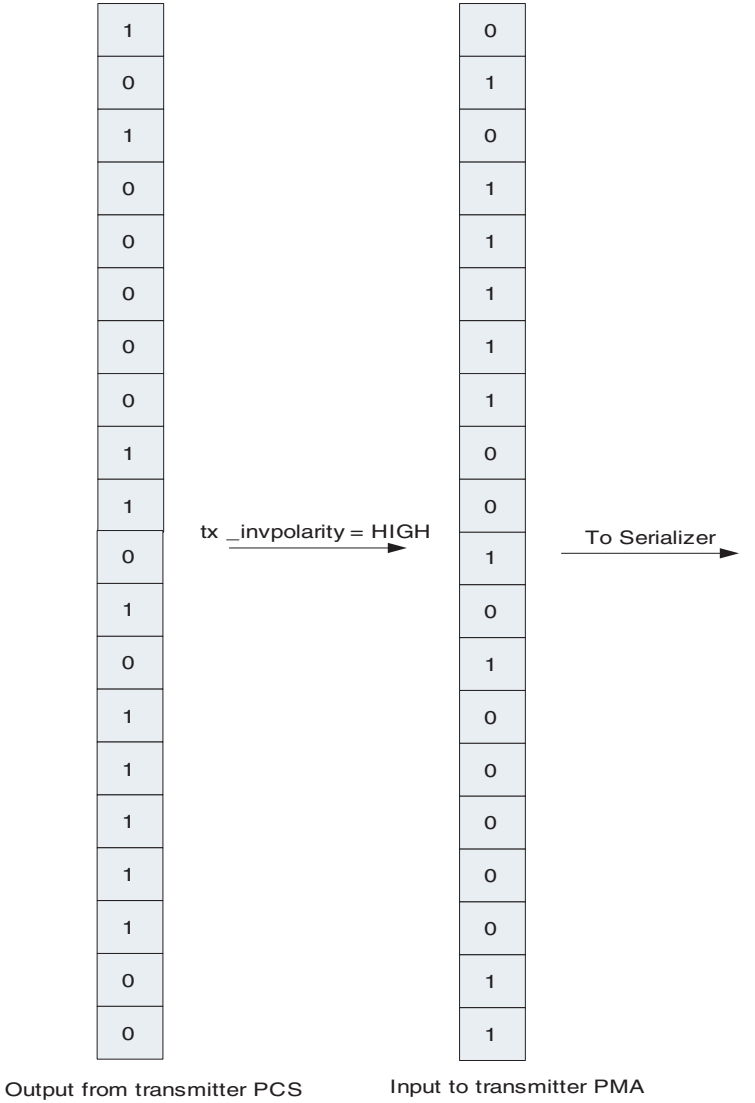


Figure 2–37 illustrates the transmitter polarity inversion feature in double-width 20-bit wide data path configuration.

Figure 2–37. Transmitter Polarity Inversion in Double-Width Mode



Transmitter Bit Reversal

By default, the Stratix II GX transmitted bit order is LSBit to MSBit. In single-width mode, the least significant bit of the 8/10-bit data word is transmitted first and the most significant bit is transmitted last. In double-width mode, the least significant bit of the 16/20-bit data word is transmitted first and the most significant bit is transmitted last. The Transmitter Bit Reversal feature allows reversing the transmitted bit order as MSBit to LSBit.

If the Transmitter Bit Reversal feature is enabled in Basic single-width mode, the 8-bit D[7:0] or 10-bit D[9:0] data at the input of the serializer gets rewired to D[0:7] or D[0:9], respectively. If the Transmitter Bit Reversal feature is enabled in Basic double-width mode, the 16-bit D[15:0] or 20-bit D[19:0] data at the input of the serializer gets rewired to D[0:15] or D[0:19], respectively. Flipping the parallel data using this feature and transmitting LSBit to MSBit effectively provides MSBit to LSBit transmission.

Figure 2–38 illustrates the transmitter bit reversal feature in Basic single-width 10-bit wide data path configuration.

Figure 2–38. Transmitter Bit Reversal in Single-Width Mode

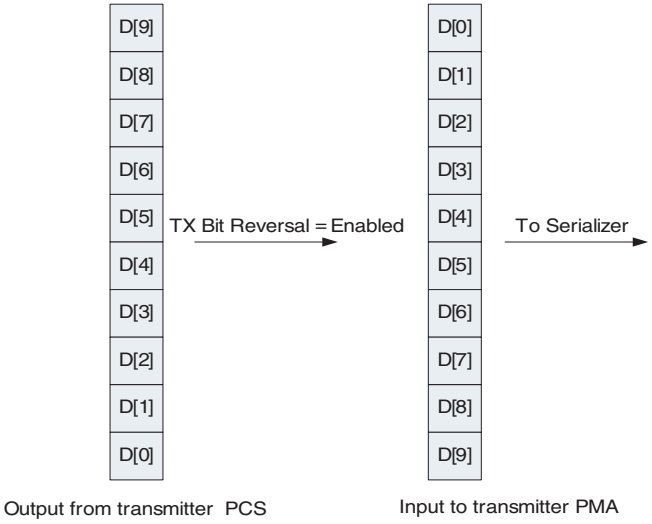
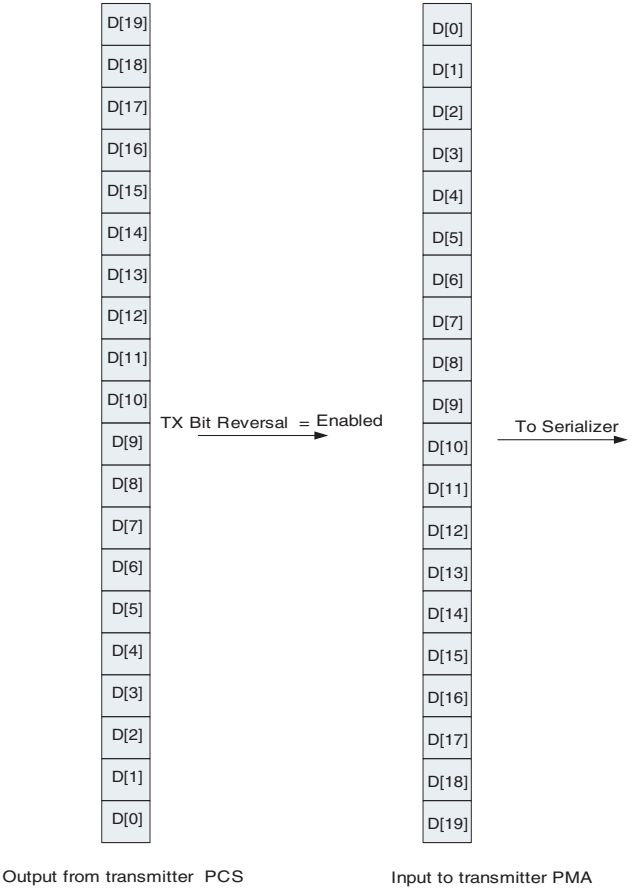


Figure 2–39 illustrates the transmitter bit reversal feature in Basic double-width 20-bit wide data path configuration.

Figure 2–39. Transmitter Bit Reversal in Double-Width Mode



Serializer

The serializer block converts parallel data to serial data at the transmitter output buffer. The serializer block supports 8-bit (Figure 2–40), 10-bit, 16-bit, and 20-bit words. The 8-bit and 10-bit operations are for use in the single-width mode and support the data rate range from 600 Mbps to 3.125 Gbps. The 16-bit and 20-bit operations are for the double-width mode and support the data rate range from 1 Gbps to 6.375 Gbps.

The serializer block drives the serial data to the output buffer as shown in the figure below. The serializer block drives the serial bit-stream at a data rate range of 600 Mbps to 6.375 Gbps. The serializer block natively transmits the LSB of the word first.

Figure 2–40. Serializer Block In 8-bit Mode

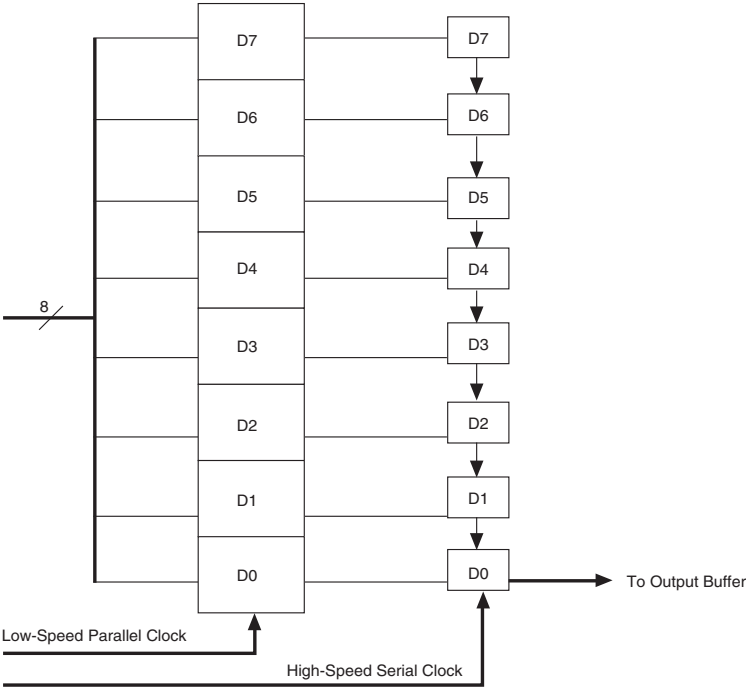
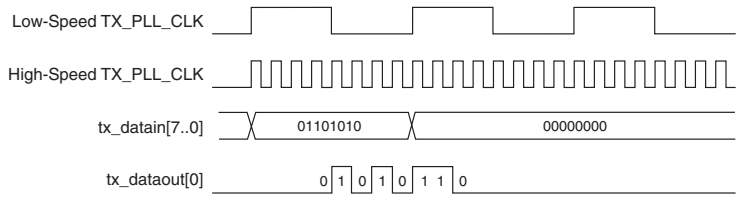


Figure 2–41 shows the serial bit order of the serializer block output. In this example a constant 8'h6A (01101010) value is serialized, and the serial data is transmitted from LSB to MSB.

Figure 2–41. Serializer Bit Order



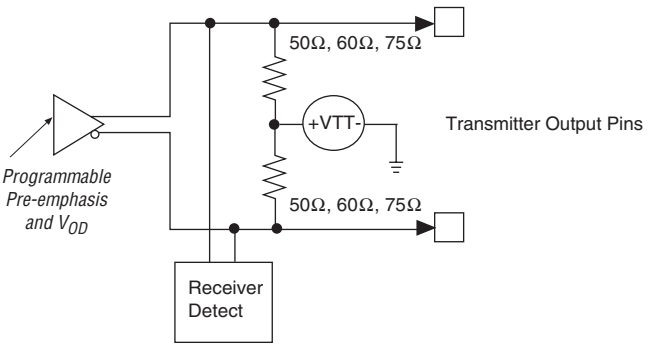
In the individual channel mode, the serializer block supplies the PLD and transceiver parallel clock. The serializer takes the ÷8 or ÷10 parallel clock from the clock divider block and distributes it to the transmitter's PCS logic in the associated transmitter channel. In single-width mode, the clock is unaltered. In double-width mode, the serializer block creates a ÷16 or ÷20 clock from the clock provided by the clock divider block, depending on the serialization factor.

In Quartus II software version 7.1 and later, basic single width allows 8-bit serialization (disable 8B/10B).

Transmitter Buffer

The Stratix II GX transmitter buffers support 1.2-V and 1.5-V pseudo current mode logic (PCML) up to 6.375 Gbps and can drive 40 inches of FR4 trace across two connectors. The transmitter buffer (refer to Figure 2–42) has additional circuitry to improve signal integrity—programmable output voltage, programmable three-tap pre-emphasis circuit, and internal termination circuitry—and the capability to detect the presence of a downstream receiver.

Figure 2–42. Transmitter Buffer



Programmable Voltage Output Differential (V_{OD})

Stratix II GX device's allow you to customize the differential output voltage (V_{OD}) to handle different trace lengths, various backplanes, and receiver requirements (refer to Figure 2–43). You select the V_{OD} from a range between 200 and 1,400 mV, as shown in Table 2–8.

Figure 2–43. V_{OD} (Differential) Signal Level

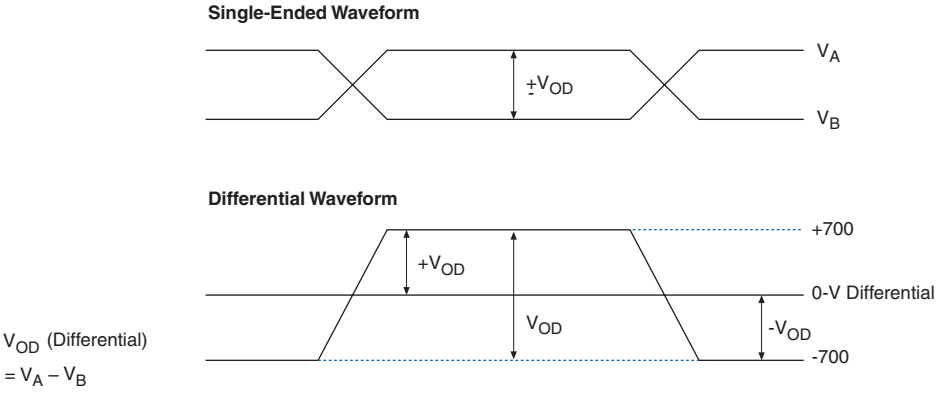


Table 2–8 shows the differential output voltage V_{OD} setting per supply voltage for each of the on-chip transmitter programmable termination values.

Table 2–8. Programmable V_{OD}

V_{OD} Differential Peak to Peak					
1.2-V VCC			1.5-V VCC		
100-Ω (mV)	120-Ω (mV)	150-Ω (mV)	100-Ω (mV)	120-Ω (mV)	150-Ω (mV)
—	192	240	200	240	300
320	384	480	400	480	600
480	576	720	600	720	900
640	768	960	800	960	1,200
800	960	—	1,000	1,200	—
960	—	—	1,200	—	—
—	—	—	1,400	—	—

You set the V_{OD} values in the MegaWizard.

The transmitter buffer is powered by either a 1.2-V or a 1.5-V power supply. You choose the transmitter buffer power (V_{CCH}) of 1.2 V or 1.5 V through the ALT2GXB MegaWizard Plug-In Manager (the **What is the transmit buffer power (V_{CCH})?** option). The transmitter buffer power supply in Stratix II GX devices is transceiver based. The 1.2-V power supply supports the 1.2-V PCML standard.

You specify the V_{OD} settings either through the MegaWizard or dynamically using the dynamic reconfiguration controller. Refer to "Introduction" on page 2–1 for more information

Programmable Pre-Emphasis

The programmable pre-emphasis module in each transmit buffer boosts the high frequencies in the transmit data signal, which may be attenuated in the transmission media. Using pre-emphasis can maximize the data eye opening at the far-end receiver.

The transmission line's transfer function can be represented in the frequency domain as a low-pass filter. Any frequency components below the –3dB frequency pass through with minimal losses. Frequency components greater than the –3dB frequency are attenuated. This variation in frequency response yields data-dependent jitter and other ISI effects. By applying pre-emphasis, the high frequency components are boosted, that is, pre-emphasized. Pre-emphasis equalizes the frequency response at the receiver so the difference between the low-frequency and high-frequency components are reduced, which minimizes the ISI effects from the transmission medium.

The pre-emphasis requirements increase as data rates through legacy backplanes increase. The Stratix II GX transmitter buffer employs a pre-emphasis circuit with up to 477% of pre-emphasis to correct for losses in the transmission medium.

You set pre-emphasis settings through a slider menu in the ALT2GXB MegaWizard Plug-In Manager. Specify the pre-emphasis settings (pre-emphasis control pre-tap, pre-emphasis 1st post tap, and pre-emphasis 2nd post tap) through the MegaWizard or dynamically using the dynamic reconfiguration controller. To enable the dynamic reconfiguration controller, you must first enable the option for dynamic reconfiguration in the ALT2GXB MegaWizard. After you enable that option, you must configure the settings in the ALT2GXB_RECONFIG MegaWizard Plug-In Manager (Stratix II GX device family) in the Quartus II software. Refer to "Introduction" on page 2–1 for more information.

Programmable Transmitter Termination

The Stratix II GX transmitter buffer includes programmable on-chip differential termination of 100 Ω, 120 Ω, or 150 Ω. The resistance is adjusted by the on-chip calibration circuit in the calibration block (refer to "Calibration Blocks" on page 2–229 for more information), which compensates for temperature, voltage, and process changes. The Stratix II GX transmitter buffers in the transceiver are current mode

drivers, so the resultant V_{OD} is a function of the transmitter termination value. Refer to "Programmable Voltage Output Differential (VOD)" on page 2–48 for more information regarding resultant V_{OD} values. You can disable the on-chip termination to use external termination. If you select external termination, the transmitter common mode is also tri-stated.

You select transmitter termination in the Quartus II software (in the Assignments menu choose Assignment Organizer, then Options for Individual Nodes Only, and then Stratix II GX Termination Value). By default, the value is 100 Ω If you plan to use 100-Ω termination, the Quartus II software automatically sets it during the compilation.

You set the transmitter termination setting through a pull-down menu in the ALT2GXB MegaWizard

Programmable Common Mode

You can set the common mode in Stratix II GX devices to 600 mV or 700 mV. Use the 600-mV setting with the 1.2-V PCML standard. Use the 700-mV setting for the 1.5-V PCML standard. Table 2–9 shows the available common mode settings for 1.2-V/1.5-V V_{CCH}.

Table 2–9. Common Mode Settings for the Available PCML Standards *Note (1)*

Common Mode Settings	1.2-V PCML Standard	1.5-V PCML Standard	Data Rates (Mbps)
600 mV	✓	✓	600 to 3125
600 mV	—	✓	3126 to 6375
700 mV	—	✓	600 to 3125

Note to Table 2–9:

(1) The PIPE protocol only allows 1.2-V PCML with 600 mV common mode. It does not allow a 1.5-V PCML and 700 mV combination.

PCI Express Receiver Detect

The Stratix II GX transmitter buffer has a built-in receiver detection circuit for use in the PIPE mode. This circuit detects if there is a receiver downstream by sending out a pulse on the common mode of the transmitter and monitoring the reflection. This mode requires the transmitter buffer to be tri-stated (in Electrical Idle mode) and the use of on-chip termination and a 125 MHz `fixedclk` signal.

This feature is only available in the PIPE mode and you enable it by setting the tx_forceelecidle and tx_detectrxloopback ports to 1'b1. You must set the powerdn port to 2'b10 to place the transmitter in the PCI-Express P1 power-down state. The results of the receiver detect is encoded on the pipestatus port.

PCI Express Electrical Idle

The Stratix II GX transmitter buffer supports PCI Express Electrical Idle (or individual transmitter tri-state). This feature is only active in the PIPE mode. The tx_forceelecidle port puts the transmitter buffer in Electrical Idle mode. This port is available in all PCI Express power-down modes and has a specific use in each mode. Table 2–10 shows the usage in each power mode.

Table 2–10. Power Mode Usage

Power Mode	Usage
P0	tx_forceelecidle must be asserted. If this signal is deasserted, it indicates that there is valid data.
P1	tx_forceelecidle must be asserted.
P2	When deasserted, the beacon signal must be transmitted. Refer to "PCI Express (PIPE) Mode" on page 2–150.

Receiver Modules

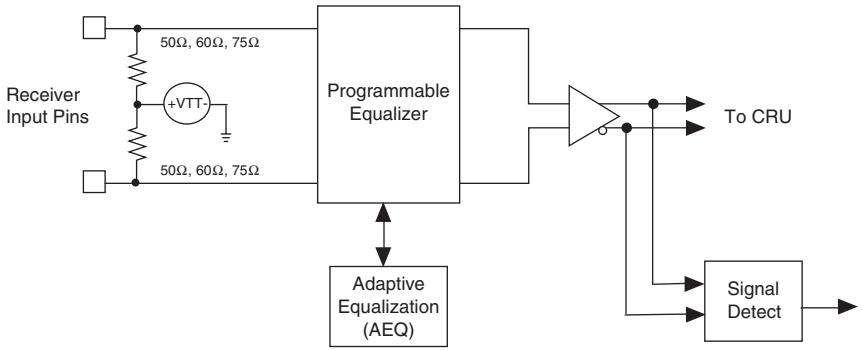
This section describes the Stratix II GX transceiver's receiver path. This section describes the following modules:

- Receiver buffer
- Receiver PLL
- Clock recovery unit
- Deserializer
- Word aligner
- Channel aligner (deskew)
- Rate matcher
- 8B/10B decoder
- Byte deserializer
- Byte Ordering
- Receiver phase compensation FIFO buffer

Receiver Buffer

The Stratix II GX receiver buffers support 1.2-V, 1.5-V, 3.3-V PCML (pseudo-current mode logic), differential LVPECL and LVDS I/O standards. The receiver buffers support data rates from 600 Mbps to 6.375 Gbps and are capable of compensating up to 40 inches of FR4 trace across two connectors. The receiver buffer (Figure 2–44) has additional circuitry to improve signal integrity, including a programmable equalization circuit and internal termination circuitry. Through a signal detect circuit, the receiver buffers can also detect if a signal of predefined amplitude exists at the receiver.

Figure 2–44. Receiver Buffer



Programmable Receiver Termination

The Stratix II GX receiver buffer has an optional programmable on-chip differential termination of 100 Ω, 120 Ω, or 150 Ω. You can set the receiver termination resistance setting using one of the two ways.

- Set the receiver termination resistance option in the MegaWizard if on-chip termination is used. The settings allowed are 100 Ω, 120 Ω, and 150 Ω. If the design requires external receive termination, check the **Use external Receiver termination** option.
- You make the differential termination assignment per pin in the Quartus II software (in the Assignments menu, choose Assignment Organizer, then Options for Individual Nodes Only, and then Stratix II GX GXB Termination Value).

Verify and set the receiver termination settings before compilation.

Signal Threshold Detection Circuit

The signal detect feature supported only in PIPE mode. The signal detect/loss threshold detector senses if the specified voltage level exists at the receiver buffer. This detector has a hysteresis response, that filters out any high-frequency ringing caused by inter-symbol interference or high-frequency losses in the transmission medium. The `rx_signaldetect` signal indicates if a signal conforms to the signal detection settings. A high level indicates that the signal conforms to the settings, a low level indicates that the signal does not conform to the settings.

The signal detect levels are to be determined by characterization. The signal detect levels may vary because of changing data patterns.

The signal/detect loss threshold detector also switches the receiver PLL/CRU from lock to reference mode to lock to data mode. The lock to reference and lock to data modes dictate whether the VCO of the clock recovery unit (CRU) is trained by the reference clock or by the data stream. Refer to "Lock-to-Reference and Lock-to-Data Modes" on page 2–64 for more information regarding this process.

You can bypass the signal/detect loss threshold detection circuit by choosing the Forced Signal Detect option in the MegaWizard. This is useful in lossy environments where the voltage thresholds might not meet the lowest voltage threshold setting. Forcing this signal high enables the receiver PLL to switch from VCO training based on the reference clock to the incoming data without detecting a valid voltage threshold.

Receiver Common Mode

Stratix II GX transceivers support the receiver buffer common mode voltage of 0.85 V and 1.2 V.

For AC-coupled links, Altera recommends selecting 0.85 V as the receiver buffer common mode voltage. For DC-coupled links, refer to "DC Coupling" on page 2–55 for recommendations on selecting the receiver common mode voltage.

Programmable Equalization

The Stratix II GX device offers an equalization circuit in each gigabit transceiver block receiver channel to increase noise margins and help reduce the effects of high-frequency losses. The programmable equalizer compensates for the high-frequency losses that distort the signal and reduces the noise margin of the transmission medium by equalizing the frequency response. There are 16 equalizer control settings allowed for a Stratix II GX device (including a setting with no equalization). In addition

to equalization, Stratix II GX devices offer an equalizer DC gain option. There are three legal settings for DC gain. You specify the equalizer settings (Equalization Settings and DC Gain) either through the MegaWizard or dynamically using the dynamic reconfiguration controller. To enable the dynamic reconfiguration controller, you first enable the option for dynamic configuration in the ALT2GXB MegaWizard Plug-In Manager. After you enable that option, you must configure the settings in the ALT2GXB_RECONFIG MegaWizard Plug-In Manager (Stratix II GX device family) in the Quartus II software. Refer to "Introduction" on page 2–1 for more information.

The transmission line's transfer function can be represented in the frequency domain as a low-pass filter. Any frequency components below the –3-dB frequency pass through with minimal losses. Frequency components that are greater than the –3-dB frequency are attenuated. This variation in frequency response yields data-dependant jitter and other ISI effects. By applying equalization, the low frequency components are attenuated. This equalizes the frequency response such that the delta between the low frequency and high frequency components are reduced, which in return minimizes the ISI effects from the transmission medium.

Stratix II GX receiver also offers the Adaptive Equalization (AEQ) feature. If you enable this feature, the equalizer circuit automatically selects and adjusts appropriate equalization levels, depending on the changing link characteristics. You can enable this feature through the dynamic reconfiguration controller. For more information, refer to the *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

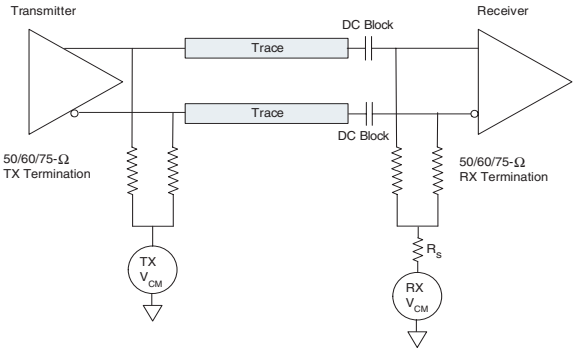
DC Coupling

A high-speed serial link can either be AC-coupled or DC-coupled, depending on the serial protocol being implemented. While most of the serial protocols require links to support AC-coupling, protocols like Common Electrical I/O (CEI) optionally allow DC coupling.

In an AC coupled link, the DC blocking capacitor blocks the transmitter DC common mode voltage (TX V_{CM}). The Stratix II GX receiver buffer allows a common mode voltage (RX V_{CM}) setting of 0.85 V and 1.2 V. The on-chip receiver termination and bias circuitry automatically restores the selected common mode voltage. If you select external termination, you must ensure that the link is terminated properly to the selected common mode voltage.

Figure 2–45 shows an AC coupled link.

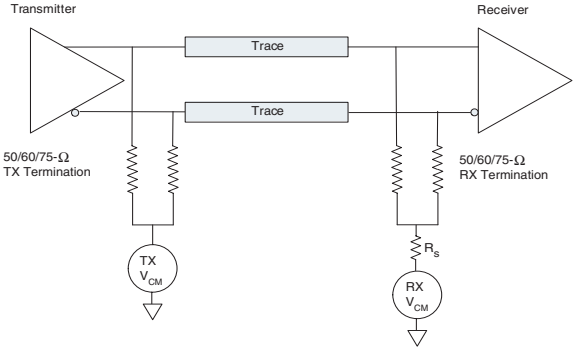
Figure 2–45. AC Coupled Link



In a DC coupled link, the transmitter DC common mode voltage is seen unblocked at the receiver buffer. The trace common mode voltage depends on the transmitter common mode voltage and the receiver common mode voltage. The external or on-chip receiver termination and bias circuitry must ensure compatibility between the transmitter and the receiver common mode voltage.

Figure 2–46 shows a DC coupled link.

Figure 2–46. DC Coupled Link



The following protocols mandate AC coupled links:

- PCI Express (PIPE)
- Gigabit Ethernet
- Serial RapidIO
- CPRI
- XAUI
- SDI

You may choose to DC couple the high-speed link for the following functional modes only:

- Basic Single and Double Width
- SONET/SDH
- CEI

The following sections discuss DC coupling requirements for a high-speed link with a Stratix II GX device used as the transmitter, receiver, or both. Specifically, the following link configurations are discussed:

- Stratix II GX Transmitter (PCML) to Stratix II GX Receiver (PCML)
- Stratix II GX Transmitter (PCML) to Stratix GX Receiver (PCML)
- Stratix GX Transmitter (PCML) to Stratix II GX Receiver (PCML)
- LVDS Transmitter to Stratix II GX Receiver (PCML)

Stratix II GX Transmitter (PCML) to Stratix II GX Receiver (PCML)

Figure 2–47 shows a typical Stratix II GX to Stratix II GX DC coupled link.

Figure 2–47. Stratix II GX to Stratix II GX DC Coupling

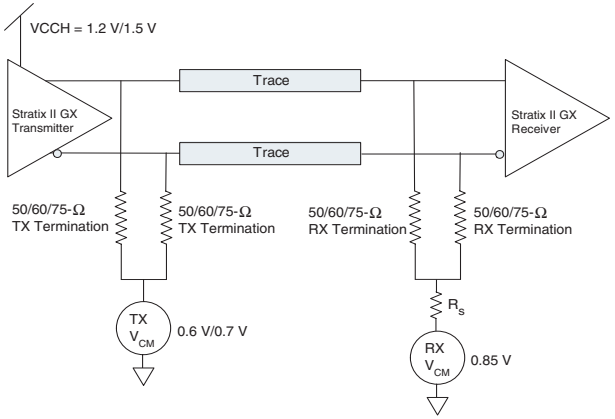


Table 2–11 shows the allowed transmitter and receiver settings in a Stratix II GX to Stratix II GX DC coupled link.

Table 2–11. Settings for a Stratix II GX to Stratix II GX DC Coupled Link

Transmitter (Stratix II GX) Settings				Receiver (Stratix II GX) Settings		
Data Rate	VCCH *(1)*	TX V_{CM} *(1)*	Differential Termination	Data Rate	RX V_{CM}	Differential Termination
600-6375 Mbps	1.2 V/1.5 V	0.6 V/0.7 V	100/120/150-Ω	600-6375 Mbps	0.85 V	100/120/150-Ω

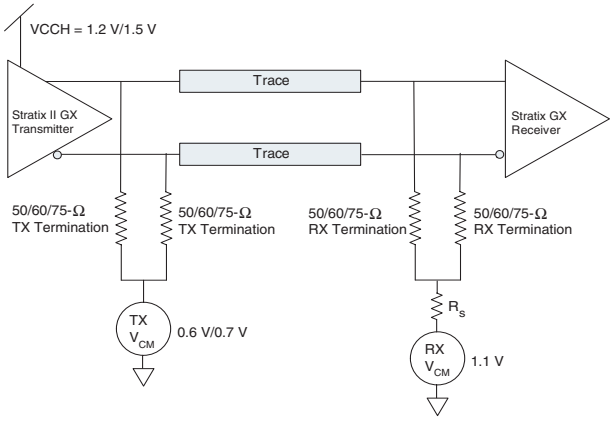
Note to Table 2–10:

(1) VCCH = 1.2 V with TX Vcm = 0.6 V can support data rates from 600 Mbps to 3125 Mbps. VCCH = 1.5 V with TX Vcm = 0.7 V can support data rates from 600 Mbps to 3125 Mbps. VCCH = 1.5 V with TX Vcm = 0.6 V can support data rates from 600 Mbps to 6375 Mbps.

Stratix II GX Transmitter (PCML) to Stratix GX Receiver (PCML)

Figure 2–48 shows a typical Stratix II GX to Stratix GX DC coupled link.

Figure 2–48. Stratix II GX to Stratix GX DC Coupling Note (1)



Note to Figure 2–48:

(1) When DC coupling to a Stratix GX receiver, you must select the **Enable Stratix GX to Stratix GX DC coupling** option for the Stratix GX receiver.

Table 2–12 shows the allowed transmitter and receiver settings in a Stratix II GX to Stratix GX DC coupled link.

Table 2–12. Settings for a Stratix II GX to Stratix GX DC Coupled Link

Transmitter (Stratix II GX) Settings				Receiver (Stratix GX) Settings		
Data Rate	**VCCH** *(1)*	**TX V_{CM}** *(1)*	**Differential Termination**	**Data Rate**	**RX V_{CM}**	**Differential Termination**
600-3187.5 Mbps	1.5 V (1.5 V PCML)	0.6 V/0.7 V	100/120/150-Ω	600-3187.5 Mbps	1.1 V	100/120/150-Ω

Note to Table 2–12:

(1) VCCH = 1.5 V with TX Vcm = 0.7 V can support data rates from 600 Mbps to 3125 Mbps. VCCH = 1.5 V with TX Vcm = 0.6 V can support data rates from 600 Mbps to 3187.5 Mbps.

Stratix GX Transmitter (PCML) to Stratix II GX Receiver (PCML)

Figure 2–49 shows a typical Stratix GX to Stratix II GX DC coupled link.

Figure 2–49. Stratix GX to Stratix II GX DC Coupling

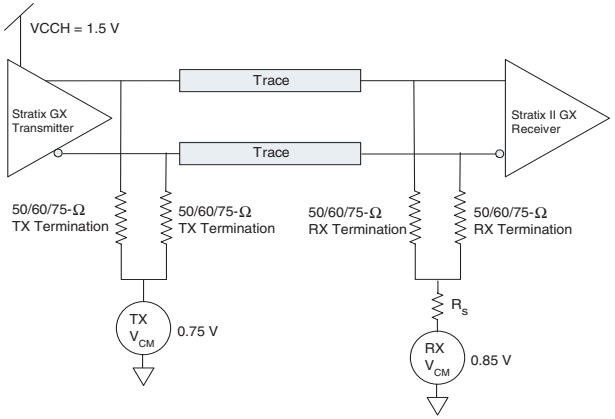


Table 2–13 shows the allowed transmitter and receiver settings in a Stratix GX to Stratix II GX DC coupled link.

Table 2–13. Settings for a Stratix GX to Stratix II GX DC Coupled Link

Transmitter (Stratix GX) Settings				Receiver (Stratix II GX) Settings			
Data Rate	VCCH	TX V_{CM}	Differential Termination	Data Rate	I/O Standard	RX V_{CM}	Differential Termination
600-3187.5 Mbps	1.5 V	0.75 V	100/120/150-Ω	600-3187.5 Mbps	1.5 V PCML	0.85 V	100/120/150-Ω

LVDS Transmitter to Stratix II GX Receiver (PCML)

Figure 2–50 shows a typical LVDS transmitter to Stratix II GX receiver DC coupled link.

Figure 2–50. LVDS to Stratix II GX DC Coupling

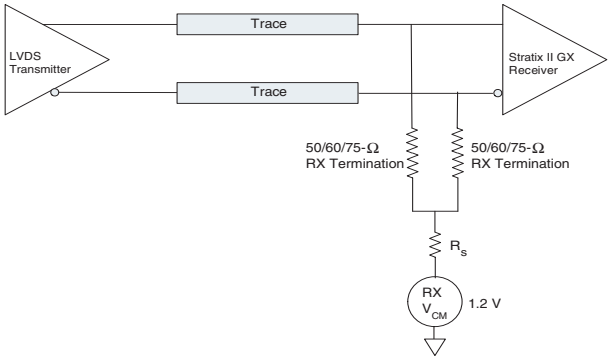


Table 2–14 shows the allowed transmitter and receiver settings in a LVDS to Stratix II GX DC coupled link.

Table 2–14. Settings for a LVDS to Stratix II GX DC Coupled Link *Note (1)*

Receiver (Stratix II GX) Settings		
RX V_{CM}	Differential Termination	R_S
1.2 V	100-Ω	88-Ω ±10%

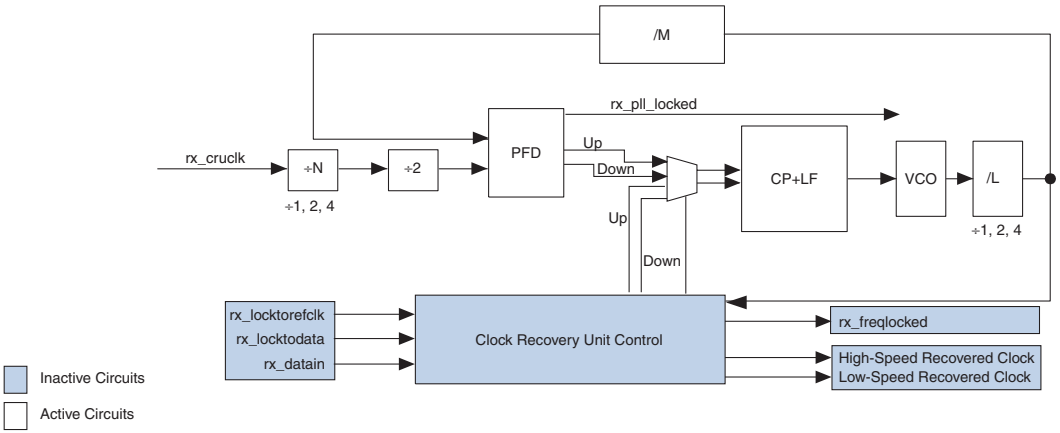
Note to Table 2–14:

(1) When DC coupling an LVDS transmitter to the Stratix II GX receiver, use RX Vcm = 1.2 V and series resistance value Rs = 88-Ω to verify compliance to the LVDS specification.

Receiver PLL

Each transceiver block contains four receiver PLLs, which receive the reference clock to train the VCO used by the CRU to match the phase and frequency of the reference clock. Figure 2–51 shows the block diagram for the lock-to-reference portion as the receiver PLL is active. Table 2–15 lists some of the PLL specifications. Table 2–16 lists the available /M and /L values within the receiver PLL.

Figure 2–51. Receiver PLL Block Diagram



Note to Figure 2–51:

(1) Values of /M and /L counters are specified in Table 2–16. The Quartus II software selects these values automatically based on the data rate and the selected reference clock frequencies.

This section focuses on the receiver PLL in lock-to-reference mode only (the receiver is not active in lock-to-data mode). The lock-to-data mode is discussed in the section "Clock Recovery Unit" on page 2–63. For information on the operation between the lock-to-reference and lock-to-data modes, refer to "Lock-to-Reference and Lock-to-Data Modes" on page 2–64.

The receiver PLL has an optional lock indicator, `rx_pll_locked`, which indicates when the receiver PLL is phase and frequency locked to the reference clock. The `rx_pll_locked` is an active high signal. A high signal indicates that the PLL is phase and frequency-locked to a reference clock, a low signal indicates that the PLL is not locked to the reference clock. If the CRU is locked to the incoming data, the `rx_pll_locked` port may toggle (assert and deassert) because the phase and/or frequency differences between the recovered clock and the reference clock might be large enough to trigger a loss of lock. This is an expected behavior because the receiver PLL is inactive in the lock-to-data mode

and the rx_pll_locked signal is ignored when the CRU is in lock-to-data mode. Table 2–15 lists the specifications for the clock recovery unit. Table 2–16 lists the available /M and /L values within the receiver PLL.

Table 2–15. Clock Recovery Unit Specifications

Parameter	Specifications
Input reference frequency range	50 MHz to 622.08 MHz
Data rate support	600 Mbps to 6.375 Gbps

Table 2–16. Available /M and /L Values Within the Receiver PLL

RX PLL Reference Clock Source	/M	/L
rx_cruclk	1, 4, 5, 8, 10, 16, 20, 25	1, 2, 4

Clock Synthesis

The maximum input frequency of the receiver PLL's phase frequency detector (PFD) is 325 MHz. To achieve a reference clock frequency above this limitation, the divide by 2 pre-divider on the dedicated local REFCLK path is automatically enabled by the Quartus II software. This divides the reference clock frequency by a factor of 2, and the /M PLL multiplier multiplies this pre-divided clock to yield the configured data rate. For example, in a situation with a data rate of 2,488 Mbps and a reference clock of 622 MHz, the reference clock must be assigned to the REFCLK port where the 622-MHz reference clock can be divided by 2, yielding a 311-MHz clock at the PFD. The VCO runs at half the data rate, so the selected multiplication factor should yield a 1244 MHz high-speed clock. The Quartus II software automatically selects a multiplication factor of 4 in this case to generate a 1244 MHz clock from the pre-divided 311 MHz clock.

If the /2 pre-divider is used, the reference clock must be fed by a dedicated reference clock input (REFCLK) pin. Otherwise the Quartus II compiler gives a fitter error.

The pre-divider and the multiplication factors are automatically set by the Quartus II software. The MegaWizard takes the data rate input and provides a list of the available reference clock frequencies that fall within the supported multiplication factors that you can select.

PPM Frequency Threshold Detector

The PPM frequency threshold detector senses whether the incoming reference clock to the clock recovery unit (CRU) and the PLL VCO of the CRU are within a prescribed PPM tolerance range. Valid parameters are 62.5, 100, 125, 200, 250, 300, 500, or 1,000 PPM. The default parameter, if no assignments are made, is 1,000 PPM. The output of the PPM frequency threshold detector is one of the variables that assert the `rx_freqlocked` signal. Refer to "Automatic Lock Mode" on page 2–64 for more details regarding the `rx_freqlocked` signal.

Receiver Bandwidth Type

The Stratix II GX receiver PLL in the CRU offers a programmable bandwidth setting. The PLL bandwidth is the measure of its ability to track the input data and jitter. The bandwidth is determined by the –3dB frequency of the closed-loop gain of the PLL.

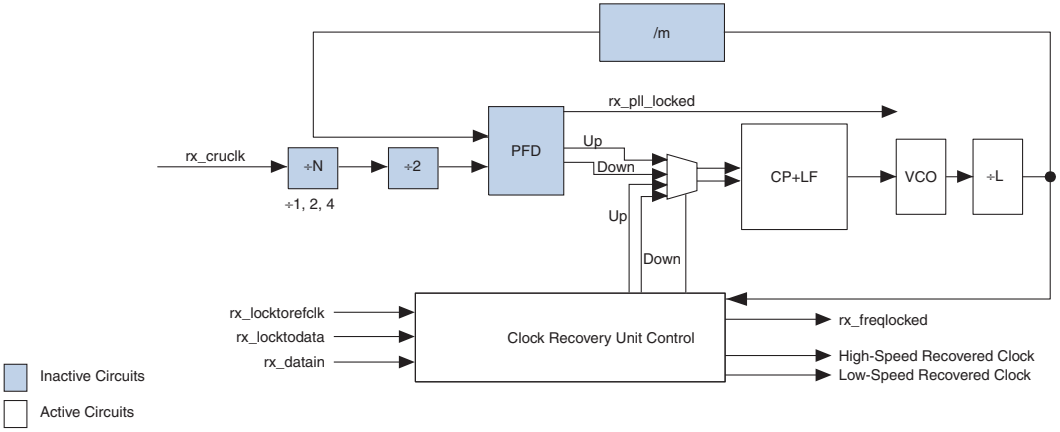
A higher bandwidth setting helps reject noise from the VCO and power supplies. A low bandwidth setting filters out more high-frequency data input jitter.

Valid receiver bandwidth settings are low, medium, or high. The –3dB frequencies for these settings vary because of the non-linear nature and data dependencies of the circuit. You can vary the bandwidth to adjust and customize the performance on specific systems.

Clock Recovery Unit

The CRU (refer to Figure 2–52) in each Stratix II GX transceiver channel recovers the clock from the serial data stream on `rx_datain`. You can set the CRU to automatically or manually lock to the data phase and frequency to match the bit transition to eliminate any clock-to-data skew or to keep the receiver PLL locked to the reference clock (lock-to-data or lock-to-reference mode). The CRU generates two clocks: a high-speed clock that feeds the deserializer and a slow-speed clock that feeds the rest of the receiver's digital logic.

Figure 2–52. Clock Recovery Unit



Lock-to-Reference and Lock-to-Data Modes

The lock-to-reference and lock-to-data modes describe the receiver PLL and the CRU. The receiver PLL is active in the lock-to-reference mode and the CRU is active in the lock-to-data mode. The switch between the two modes can be done automatically or manually.

Automatic Lock Mode

The CRU, by default, initially locks to the CRU reference clock (lock-to-reference mode) until switching over to the incoming data (lock-to-data mode). The switch to lock-to-data mode is indicated by the assertion of the `rx_freqlocked` signal. The `rx_freqlocked` signal only indicates the current mode of CRU (lock-to-data or lock-to-reference). After switching into lock-to-data mode (assertion of the `rx_freqlocked` signal), the CRU unit needs time to acquire phase lock to the incoming data stream.

For automatic transition from the lock-to-reference mode to the lock-to-data mode, the following conditions must be met:

- The serial data at the receiver input buffer is within the prescribed voltage signal loss threshold.
- The CRU PLL is within the prescribed PPM frequency threshold setting (62.5, 100, 125, 200, 250, 300 , 500, or 1,000 PPM) of the CRU reference clock.
- The reference clock and CRU PLL output are phase matched (phases are within approximately 0.08 UI).

When the receiver PLL and CRU is in lock-to-reference mode, the PPM detector, phase detector, and signal-detect circuits monitor the relationship of the reference clock to the receiver PLL and VCO output. If the frequency difference is within the prescribed PPM setting, the amplitude is within the prescribed limits, and the phase is within 0.08 UI, then the CRU switches to the lock-to-data mode.

In lock-to-data mode, the PLL uses a phase detector to keep the recovered clock aligned properly with the data. If the PLL does not stay locked to data because of problems such as frequency drift or severe amplitude attenuation, the receiver PLL locks back to the reference clock of the CRU to train the VCO. When the device is in lock-to-data mode (`rx_freqlocked` is asserted), the CRU is trying to align with incoming data and there is no phase relationship with the reference clock.

The `rx_freqlocked` signal indicates which mode the CRU is in (either lock-to-data or lock-to-reference mode). In lock-to-data mode, the `rx_freqlocked` signal is high. In lock-to-reference mode, the `rx_freqlocked` signal is low.

In lock-to-data mode, the `rx_freqlocked` signal is asserted and the `rx_pll_locked` signal loses its significance because the `rx_pll_locked` signal indicates that the CRU has locked to the reference clock. When the CRU is in lock-to-data mode, the phase of the VCO may differ from the reference clock, which may deassert the `rx_pll_locked` signal. You should ignore the `rx_pll_locked` signal when the `rx_freqlocked` signal is high.

In automatic lock mode, there are two conditions that force the CRU to fall out of lock-to-data mode.

- The serial data at the receiver input buffer is not within the prescribed voltage signal loss threshold. This condition is ignored if the Force signal detection option in the MegaWizard is enabled.
- The CRU PLL is not within the prescribed PPM frequency threshold setting (62.5, 100, 125, 200, 250, 300, 500, or 1,000 PPM) of the CRU reference clock.

If the CRU falls out of lock-to-data mode, the `rx_freqlocked` signal is deasserted. You can also deassert the `rx_freqlocked` signal by asserting either `rx_analogreset` (powers down the receiver) or `gxb_powerdown` (powers down all four channels of the transceiver block.

Manual Lock Options

Two optional input pins (rx_locktorefclk and rx_locktodata) allow you to control whether the CRU PLL automatically or manually switches between lock-to-reference mode and lock-to-data modes. This enables you to bypass the default automatic switchover circuitry if either rx_locktorefclk or rx_locktodata is instantiated.

When the rx_locktorefclk signal is asserted, it forces the CRU PLL to lock to the reference clock (rx_cruclk). Asserting rx_locktodata forces the CRU PLL to lock to data. This occurs whether the CRU is ready or not. When both signals are asserted, the rx_locktodata signal takes precedence over the rx_locktorefclk signal.

The signal loss threshold detector, PPM threshold frequency detector, and phase relationship detector reaction times may be too long for some applications. You can manually control the CRU to reduce CRU lock times using the rx_locktorefclk and rx_locktodata ports. Using the manual mode may reduce the time it takes for the CRU to switch from lock-to-reference mode to lock-to-data mode. You can assert the rx_locktorefclk to initially train the CRU. The rx_locktodata signal should be asserted after training the CRU.

When the rx_locktorefclk signal is asserted, the rx_freqlocked signal does not have any significance because it is low, indicating that the CRU is in lock-to-reference mode. If lock-to-data mode is asserted, the rx_freqlocked signal is always asserted, indicating that the CRU is in lock-to-data mode. When both signals are asserted, lock-to-data mode takes precedence. If both signals are deasserted, the CRU switchover is in automatic mode. Table 2–17 shows a summary of the control signals.

Table 2–17. CRU User Control Lock Signals

rx_locktorefclk	rx_locktodata	CRU Mode
1	0	Lock-to-reference clock
x	1	Lock to data
0	0	Automatic

Deserializer

The deserializer block converts incoming high-speed serial data streams to 8-, 10-, 16-, or 20-bit-wide parallel data synchronized to the recovered clock of the CRU. Use the 8- and 10-bit operations, which support a data rate from 600 Mbps to 3.125 Gbps, in the single-width mode. Use the 16- and 20-bit operations, which support a data rate from 1 Gbps to 6.375 Gbps, for the double-width mode.

The deserializer block drives the parallel data to the pattern detector and word aligner, as shown in Figure 2–53. The deserializer block output bus data rate is the input data rate divided by the width of the output data bus. For example, for a 10-bit bus and a serial input data rate of 2.5 Gbps, the parallel data rate is 2.5 ÷ 10 or 250 MHz. The first bit into the deserializer block is the LSB of the data bus out.

Figure 2–53. Deserializer Block in 8-Bit Mode

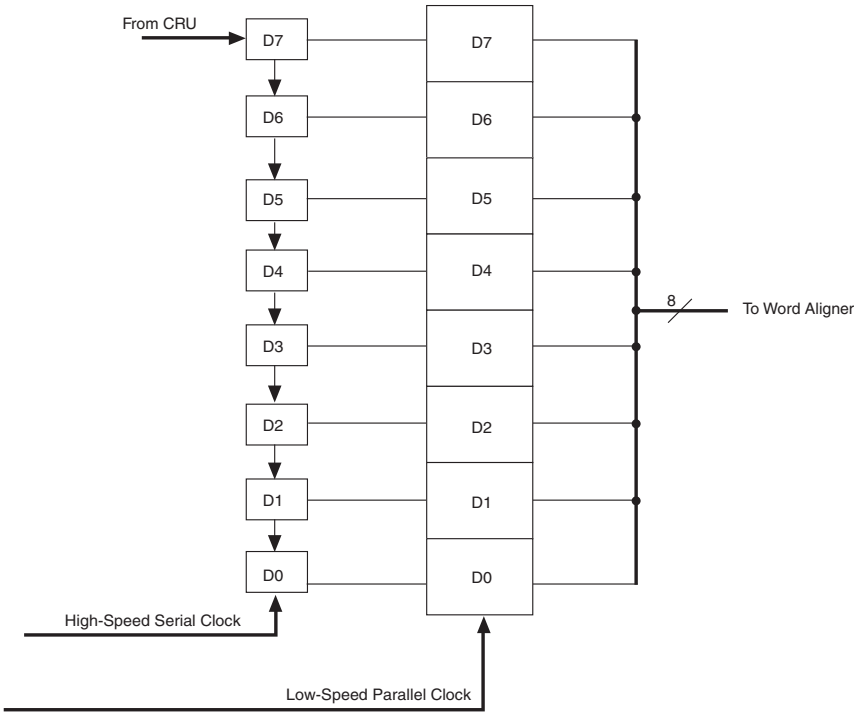
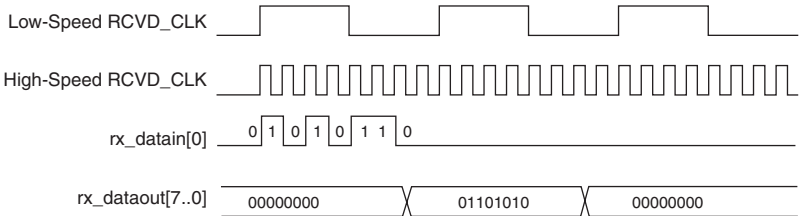


Figure 2–54 shows the serial bit order of the deserializer block input and the parallel data out of the deserializer block. Figure 2–54 shows a serial stream (01101010) deserialized into a value 8'h6A (01101010). The serial data is received LSB to MSB.

In Quartus II software version 7.1 and later, basic single width allows 8-bit deserializer (disable 8B/10B).

Figure 2–54. Deserializer Bit Order



Generic Receiver Polarity Inversion

The positive and negative signals of a serial differential link are often erroneously swapped during board layout. Solutions like a board re-spin or major updates to the PLD logic can prove expensive. The receiver polarity inversion feature is provided to correct this situation.

An optional `rx_invpolarity` port is available in all single-width and double-width modes except (OIF) CEI PHY to dynamically enable the receiver polarity inversion feature. In single-width modes, a high value on the `rx_invpolarity` port inverts the polarity of every bit of the 8-bit or 10-bit input data word to the word aligner in the receiver data path. In double-width modes, a high value on the `rx_invpolarity` port inverts the polarity of every bit of the 16-bit or 20-bit input data word to the word aligner in the receiver data path. Since inverting the polarity of each bit has the same effect as swapping the positive and negative signals of the differential link, correct data is seen by the receiver. The `rx_invpolarity` is a dynamic signal and may cause initial disparity errors in an 8B/10B encoded link. The downstream system must be able to tolerate these disparity errors.

The generic receiver polarity inversion feature is different from the PIPE 8B/10B polarity inversion feature. The generic receiver polarity inversion feature inverts the polarity of the data bits at the input of the word aligner. The PIPE 8B/10B polarity inversion feature inverts the polarity of the data bits at the input of the 8B/10B decoder and is available only in PIPE mode. Enabling the generic receiver polarity inversion and the PIPE 8B/10B polarity inversion simultaneously is not allowed in PIPE mode.

Figure 2–55 illustrates the receiver polarity inversion feature in single-width 10-bit wide data path configuration.

Figure 2–55. Receiver Polarity Inversion in Single-Width Mode

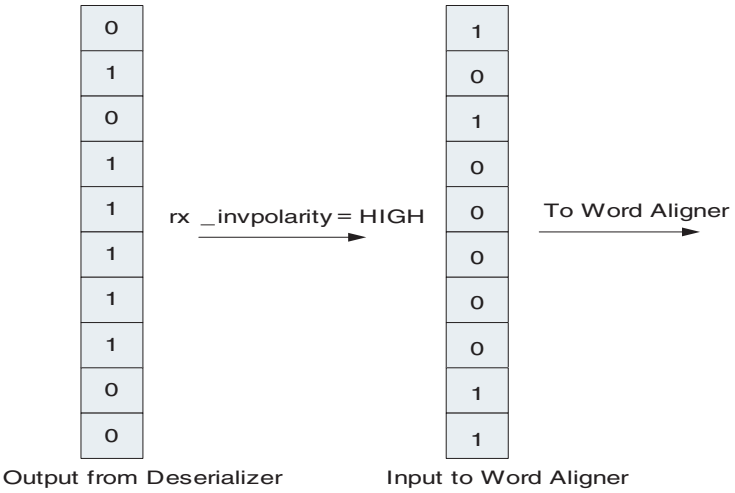
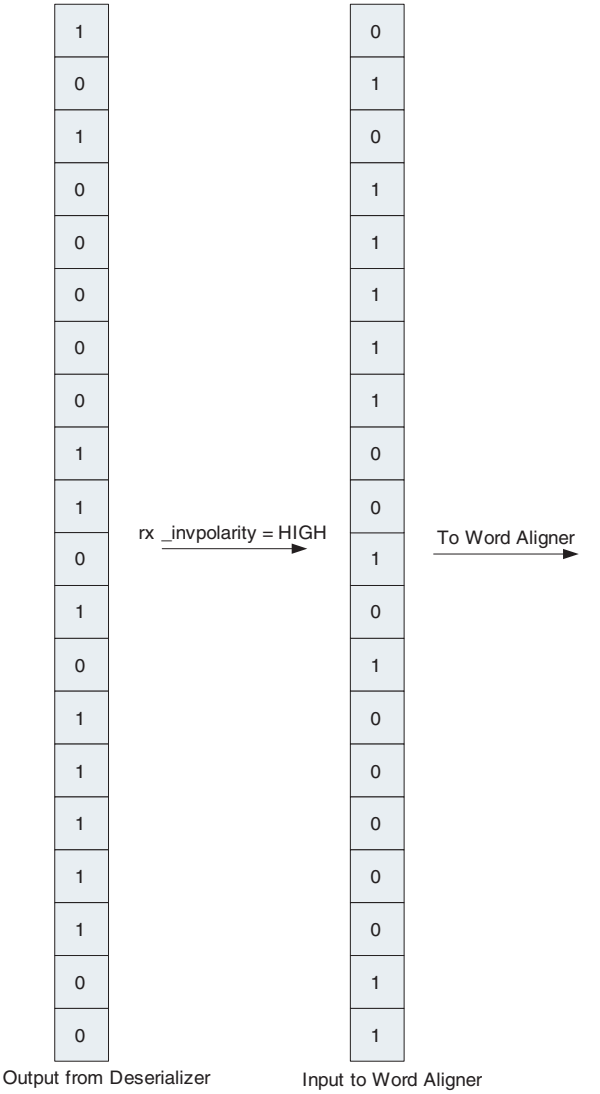


Figure 2–56 illustrates the receiver polarity inversion feature in double-width 20-bit wide data path configuration.

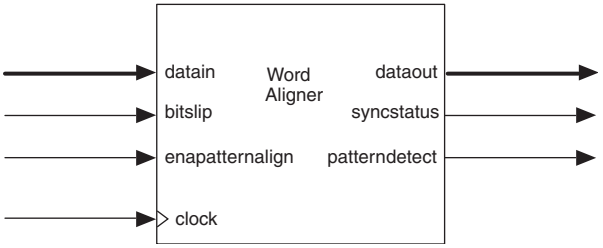
Figure 2–56. Receiver Polarity Inversion in Double-Width Mode



Word Aligner

The word aligner (refer to Figure 2–57) is part of the Stratix II GX transceiver digital blocks and is located in the receiver path between the deserializer and the de-skew FIFO buffer. The word aligner restores the byte boundary of the upstream transmitter based on a programmable alignment pattern that appears in the serial data stream.

Figure 2–57. Word Aligner



The word aligner block consists of four main sub-blocks:

- Aligner block
- Pattern detect block
- Manual bit-slip block
- Run-length checker

There are three modes in which the word aligner works: single-width mode, double-width mode, and automatic synchronization state machine mode. The following sections explain each of the blocks in each mode of operation. The word aligner cannot be bypassed and must be used. However, you can use the `rx_enapatternalign` port to set the word alignment to not align to the pattern.

Single-Width Mode

In single-width mode, there are three blocks active in the word aligner:

- Pattern detector
- Manual word aligner
- Automatic synchronization state machine

The pattern detector detects if the pattern exists in the current word boundary. The manual alignment identifies the alignment pattern across the byte boundaries and aligns to the correct byte boundary. The synchronization state machine detects the number of alignment patterns and good code groups for synchronization and goes out of

synchronization if code group errors (bad code groups) are detected. Figure 2–58 and Table 2–18 show the modes available in the single-width mode and the supported alignment modes.

Figure 2–58. Word Aligner Components in Single-Width Mode

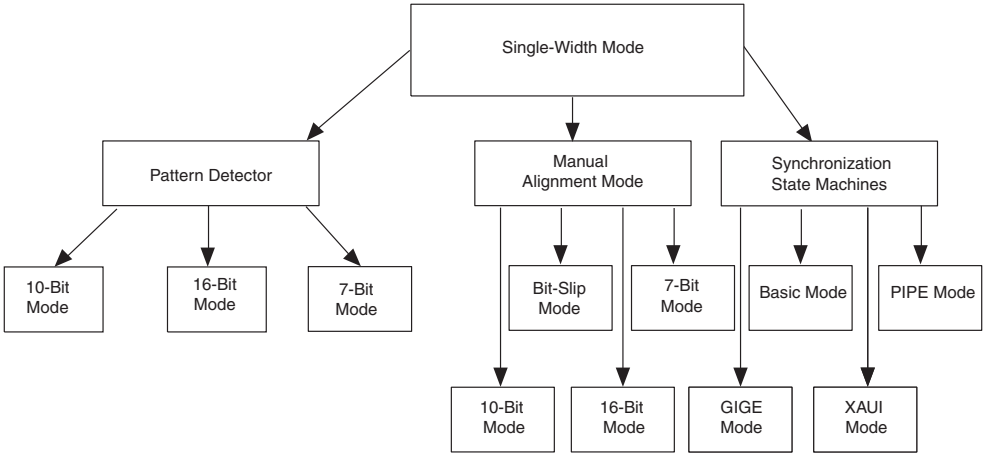


Table 2–18. Word Alignment Modes

Word Alignment Mode	Effective Mode	Control Signals	Status Signals
Synchronization state machine	PCI Express, XAUI, GIGE, Serial RapidIO, CPRI, or Basic	Automatically controlled to adhere to the specified standard or by user entered parameter	`rx_syncstatus` `rx_patterndetect`
Manual 7- and 10-bit alignment mode	Alignment to detected pattern when allowed by the `rx_enapatternalign` signal	`rx_enapatternalign`	`rx_syncstatus` `rx_patterndetect`
Manual 16-bit alignment mode	Alignment to detected pattern when allowed by the `rx_enapatternalign` signal	`rx_enapatternalign`	`rx_syncstatus` `rx_patterndetect`
Manual bit-slipping alignment mode	Manual bit slip controlled by the PLD logic array	`rx_bitslip`	`rx_patterndetect`

Pattern Detector Module

The pattern detector matches a pre-defined alignment pattern to the current byte boundary. When the pattern detector locates the alignment pattern, the optional rx_patterndetect signal is asserted for the duration of one clock cycle to signify that the alignment pattern exists in the current word boundary. The pattern detector module only indicates that the signal exists and does not modify the word boundary. Modification of the word boundary is discussed in the sections "Manual Alignment Modes" on page 2–75 and "Synchronization State Machine Mode" on page 2–81.

In the MegaWizard, you can program a 7-bit, 10-bit, or 16-bit pattern for the pattern detector to recognize. The pattern used for pattern matching is automatically derived from the word alignment pattern in the MegaWizard. For the 7-bit and 10-bit patterns, the actual alignment pattern specified in the MegaWizard and its complement are checked. For the 16-bit alignment pattern, only the actual pattern is checked. Table 2–19 shows the supported alignment patterns.

Table 2–19. Supported Alignment Patterns

Pattern Detect Mode	Supported Protocols	Pattern Checked
7 bit	Basic, CPRI	Actual and complement
10 bit	Basic, XAUI, GIGE, CPRI, Serial RapidIO, and PIPE	Actual and complement
Two consecutive 8-bit characters	SONET/SDH	Actual

In 8B/10B encoded data, actual and complement pattern indicates positive and negative disparities.

Each bit in the rx_patterndetect and rx_syncstatus signal indicates the status of the group of 8 or 10 bits in the rx_dataout port. For the rx_dataout width of 8/10, 16/20, and 32/40 bits, the rx_patterndetect and rx_syncstatus widths are 1, 2, and 4 bits, respectively. An example of corresponding signals for rx_dataout (widths of 16 and 32) is shown in Table 2–20.

Table 2–20. Corresponding Signals for rx_dataout (Part 1 of 2)

Data Width	Signal	Corresponding Signal
16	rx_patterndetect[1]	rx_dataout[15:8]
	rx_patterndetect[0]	rx_dataout[7:0]

Table 2–20. Corresponding Signals for rx_dataout (Part 2 of 2)

Data Width	Signal	Corresponding Signal
32	rx_patterndetect[3]	rx_dataout[31:24]
	rx_patterndetect[2]	rx_dataout[23:16]
	rx_patterndetect[1]	rx_dataout[15:8]
	rx_patterndetect[0]	rx_dataout[7:0]

7-Bit Pattern Mode

In the 7-bit pattern detection mode (use this mode with 8B/10B code), the pattern detector matches the seven LSBs of the 10-bit alignment pattern, which you specified in your ALT2GXB custom megafunction variation, in the current word boundary. Both positive and negative disparities are also checked in this mode.

The 7-bit pattern mode can mask out the three MSBs of the data, which allows the pattern detector to recognize multiple alignment patterns. For example, in the 8B/10B encoded data, a /K28.5/ (b'0011111010), /K28.1/ (b'0011111001), and /K28.7/ (b'0011111000) share seven common LSBs. Masking the three MSBs allows the pattern detector to resolve all three alignment patterns and indicate them on the `rx_patterndetect` port.

In 7-bit pattern mode, the word aligner still aligns to a 10-bit word boundary. The specified 7-bit pattern forms the least significant seven bits of the 10-bit word.

10-Bit Pattern Mode

In the 10-bit pattern detection mode (use this mode with 8B/10B code), the module matches the 10-bit alignment pattern you specified in your ALT2GXB custom megafunction variation with the data and its complement in the current word boundary. Both positive and negative disparities are checked by the pattern checker in this mode. For example, if you specify a /K28.5/ (b'0011111010) pattern as the comma, `rx_patterndetect` is asserted if b'0011111010 or b'1100000101 is detected in the incoming data.

16-bit Pattern Mode

You specify the 16-bit alignment pattern in the MegaWizard and it is oriented with the MSB first and the LSB last. A1 represents the least significant byte, which consists of bits `[7..0]`. A2 represents the most significant byte, which consists of bits `[15..8]`. Therefore, the alignment pattern is specified as [A2,A1] in the MegaWizard. Only the actual alignment pattern specified in the MegaWizard is detected in this mode.

The pattern detector is defaulted to the A1A2 mode. Only the positive disparity is detected in this mode.

Manual Alignment Modes

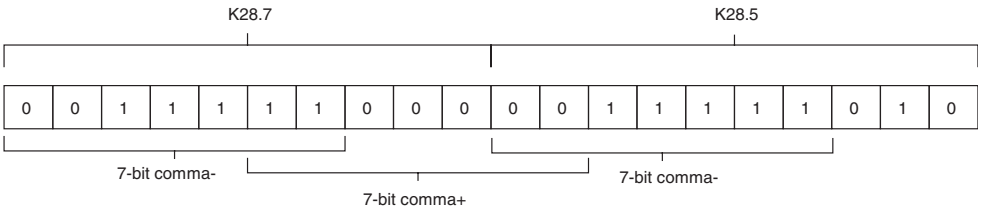
The word aligner has three manual alignment modes (7-, 10-, and 16-bits) when the transceiver data path is in single-width mode. The 10- and 7-bit alignment modes are used with 8B/10B code and the 16-bit alignment mode is for scrambled or non-scrambled data.

7-bit Alignment Mode

In the 7-bit alignment mode (use the 8B/10B encoded data with this mode), the module looks for the 7-bit alignment pattern you specified in the MegaWizard Plug-In Manager in the incoming data stream. The 7-bit alignment mode is useful because it can mask out the three most significant bits of the data, which allows the word aligner to align to multiple alignment patterns. For example, in the 8B/10B encoded data, a /K28.5/ (b'0011111010), /K28.1/ (b'0011111001), and /K28.7/ (b'0011111000) share seven common LSBs. Masking the three MSBs allows the word aligner to resolve all three alignment patterns synchronized to it. The word aligner places the boundary of the 7-bit pattern in the LSByte position with bit positions `[0..7]`. The true and complement of the patterns is checked.

Use the `rx_enapatternalign` port to enable the 7-bit manual word alignment mode. When the `rx_enapatternalign` signal is high, the word aligner detects the specified alignment patterns and realigns the byte boundary if needed. The `rx_syncstatus` port is asserted for one parallel clock cycle to signify that the word boundary was detected across the current word boundary and has synchronized to the new boundary, if a rising edge was detected previously on the `rx_enapatternalign` port. You must differentiate if the acquired byte boundary is correct, because the 7-bit pattern can appear between word boundaries. For example, in the standard 7-bit alignment pattern -7'b1111100, if a K28.7 is followed by a K28.5, the 7-bit alignment pattern appears on K28.7, between K28.7 and K28.5, and also again in K28.5 (refer to Figure 2–59).

Figure 2–59. Cross Boundary 7-Bit Comma When /K28.7 is Followed by /K28.5



Manual 10-Bit Alignment Mode

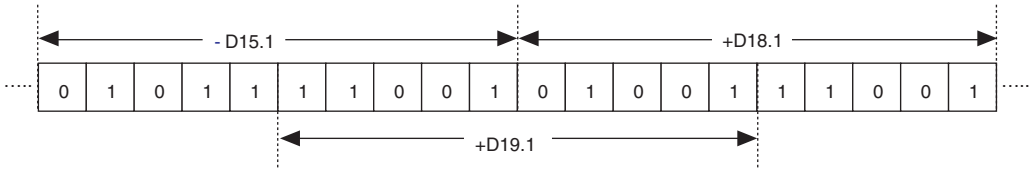
You can configure the word aligner to align to a 10-bit word boundary. The internal word alignment circuitry shifts to the correct word boundary if the alignment pattern specified in the pattern detector is detected in the data stream.

The rx_enapatternalign port enables the word alignment in the manual 10-bit alignment mode. When the rx_enapatternalign signal is high, the word aligner detects the specified alignment pattern and realigns the byte boundary if necessary. The rx_syncstatus port is asserted for one parallel clock cycle to signify that the word boundary has been detected across the word boundary and has synchronized to the new boundary.

The rx_enapatternalign signal is held high if the alignment pattern is known to be unique and does not appear across the byte boundaries of other data. For example, if an 8B/10B encoding scheme guarantees that the /K28.5/ code group is a unique pattern in the data stream, the rx_enapatternalign port is held at a constant high.

If the alignment pattern can exist between word boundaries, the rx_enapatternalign port must be controlled by the user logic in the PLD to avoid false word alignment. For example, assume that 8B/10B is used and a /+D19.1/ (b'110010 1001) character is specified as the alignment pattern. In this case, a false word boundary is detected if a /-D15.1/ (b'010111 1001) is followed by a /+D18.1/ (b'010011 1001). Refer to Figure 2–60.

Figure 2–60. False Word Boundary Alignment if Alignment Pattern Exists Across Word Boundaries, Single Width



In this example, the rx_enapatternalign signal is deasserted after the word aligner locates the initial word alignment to prevent false word boundary alignment. When the rx_enapatternalign signal is deasserted, the current word boundary is locked even if the alignment pattern is detected across different boundaries. In this case, the rx_syncstatus acts as a re-synchronization signal to signify that the

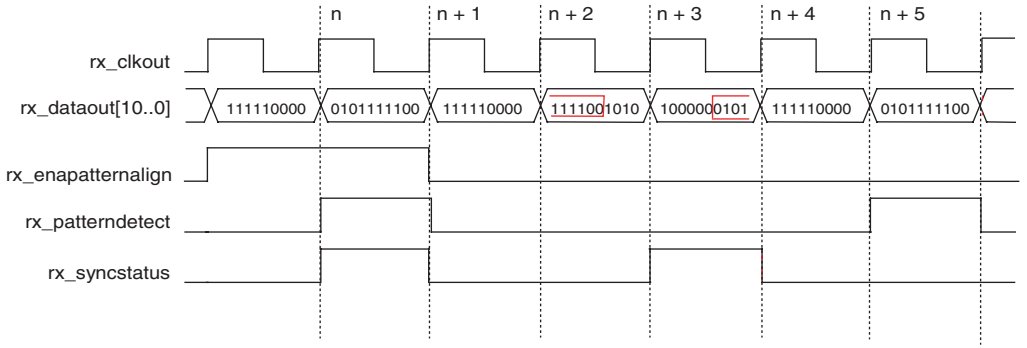
alignment pattern was detected, but the boundary is different than the current boundary. You must monitor this signal and reassert the rx_enapatternalign signal if realignment is desired.

Figure 2–61 shows an example of how the word aligner signals interact in 10-bit alignment mode. In this example, a /K28.5/ (10'b0011111010) is specified as the alignment pattern. The rx_enapatternalign signal is held high at time *n*, so alignment occurs whenever a alignment pattern exists in the pattern. The rx_patterndetect signal is asserted for one clock cycle to signify that the pattern exists on the re-aligned boundary. The rx_syncstatus signal is also asserted for one clock cycle to signify that the boundary has been synchronized. At time *n* + 1, the rx_enapatternalign signal is deasserted to instruct the word aligner to lock the current word boundary.

The alignment pattern is detected at time *n* + 2, but it exists on a different boundary than the current locked boundary. The bit orientation of the Stratix II GX device is LSB to MSB, so the alignment pattern exists across time *n* + 2 and *n* + 3 (refer to Figure 2–61). In this condition the rx_patterndetect remains low because the alignment pattern does not exist on the current word boundary, but the rx_syncstatus signal is asserted for one clock cycle to signify a resynchronization condition. This means that the alignment pattern has been detected across another word boundary.

The user logic design in the PLD must decide whether or not to assert the rx_enapatternalign to re-initiate the word alignment process. At time *n* + 5 the rx_patterndetect signal is asserted for one clock cycle to signify that the alignment pattern has been detected on the current word boundary.

Figure 2–61. Word Aligner Symbol Interaction in 10-Bit Manual Alignment Mode



Manual SONET/SDH Alignment Mode (Two Consecutive 8-bit Characters (A1A2) or Four Consecutive 8-bit Characters (A1A1A2A2))
The word aligner can be configured to align to a 16-bit word boundary in SONET/SDH protocol mode. In the SONET/SDH protocol mode, the word aligner either aligns to two consecutive 8-bit characters (A1A2) or four consecutive 8-bit characters (A1A1A2A2). The `rx_a1a2size` signal can be used to differentiate between the two and four consecutive modes. The word aligner aligns to the A1A2 pattern when the `rx_a1a2size` signal is held low "0," or to the A1A1A2A2 when `rx_a1a2size` is high "1." The `rx_a1a2sizeout` port sends the state of the `rx_a1a2size` signal as seen by the word aligner back to the PLD logic array.

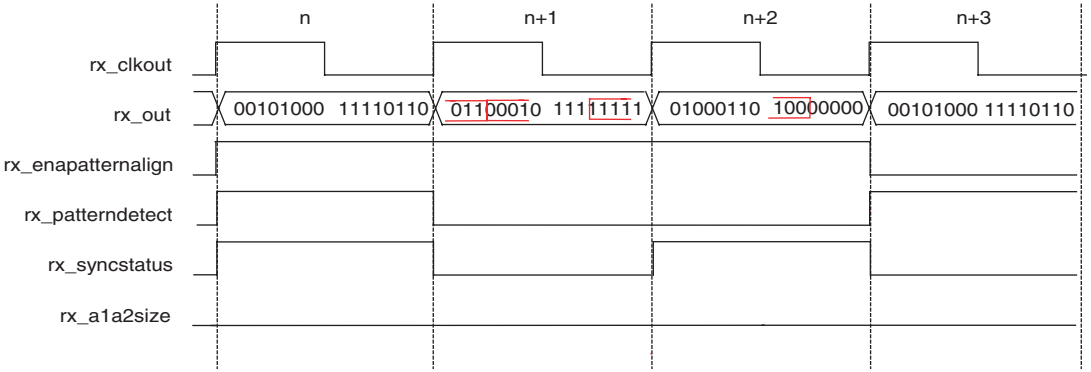
Word alignment is enabled or re-enabled by the `rx_enapatternalign` signal, but the behavior is different than that described for the 7-bit or 10-bit manual mode in single-width configuration. In the 7/10-bit mode the byte boundary can be dynamically changed if the `rx_enapatternalign` signal is held high. However, in the SONET/SDH mode the byte boundary is locked after the first alignment pattern is detected and aligned after the rising edge of the `rx_enapatternalign` signal. If the byte boundary changes, the `rx_enapatternalign` signal must be deasserted and reasserted to re-enable the alignment circuit. This feature is valuable in SONET/SDH because the data is scrambled and not encoded. The alignment pattern can potentially exist across byte boundaries and can trigger a false realignment. In SONET/SDH the byte boundary should be aligned and locked at the beginning of a SONET/SDH frame since the A1A2 alignment pattern resides in the framing section at the beginning of the transport overhead.

Initially, the word aligner locks onto the first alignment pattern detected. In this scenario the `rx_patterndetect` signal is asserted for one clock cycle to signify that the alignment pattern has been aligned. The `rx_syncstatus` signal is asserted for a clock cycle to signify that the word boundary has been synchronized. After the word boundary has been locked, regardless of whether the `rx_enapatternalign` signal is held high or low, the `rx_syncstatus` signal asserts for one clock cycle whenever the alignment pattern is detected across a different byte boundary. The `rx_syncstatus` signal operates in this resynchronization state until a rising edge is detected on `rx_enapatternalign`.

Figure 2–62 shows an example of how the word aligner signals interact in SONET/SDH alignment mode for an A1A2 pattern. For this example, a SONET/SDH A1A2 framing pattern uses 16'hF628 (16'b1111011000101000) with the reverse bit ordering. This option reverses

the bit order so that the data can be transmitted and received MSB to LSB. If the bit reversal option in the MegaWizard Plug-In Manager is not used, the transceiver transmits and receives LSB to MSB.

Figure 2–62. Word Aligner Symbol Interaction in SONET/SDH A1A2 Manual Alignment Mode



The `rx_a1a2size` signal is held low, which sets the SONET/SDH alignment mode to A1A2. The `rx_enapatternalign` signal is toggled high at time *n*, so the aligner locks to the boundary of the next present alignment pattern. The A1A2 alignment pattern appears on the `rx_dataout` port during this period. The alignment pattern exists, so the `rx_patterndetect` and `rx_syncstatus` signals are asserted for one clock cycle to signify that the A1A2 alignment pattern has been detected and the word boundary has been locked. The A1A2 alignment pattern appears again across word boundaries in during periods *n*+1, and *n*+2. The `rx_enapatternalign` signal is held high, but the word aligner does not re-align the byte boundary as it would in 10-bit manual alignment mode. Instead the `rx_syncstatus` signal is asserted for one clock cycle to signify a re-synchronization condition. You must deassert and reassert the `rx_enapatternalign` signal to re-trigger the word aligner and align on the next alignment pattern. The next transition occurs at time *n*+3, where `rx_enapatternalign` is deasserted and the A1A2 pattern is present on the `rx_dataout` port. The word aligner then asserts the `rx_patterndetect` signal for one clock cycle to flag the detection of the alignment pattern on the current word boundary.

Manual 16-bit Alignment Mode

You enable the 16-bit alignment mode in the single-width mode. This mode aligns to the 16-bit alignment pattern you specified in the MegaWizard.

The byte boundary is locked after the first alignment pattern is detected and after the rising edge of the rx_enapatternalign port. If the byte boundary changes, the rx_enapatternalign port is deasserted and reasserted to enable the alignment circuit to search for and align to the next available alignment pattern.

On the rising edge of the rx_enapatternalign port, the word aligner locks onto the first alignment pattern detected. In this scenario, rx_patterndetect is asserted to signify that the alignment pattern has been aligned. The rx_syncstatus signal is also asserted for one clock cycle to signify that the word boundary has been synchronized. After the word boundary is locked, whether or not rx_enapatternalign is held high or low, the rx_syncstatus signal asserts for one clock cycle whenever the alignment pattern is detected across a different byte boundary. The rx_syncstatus signal operates in this resynchronization state until a rising edge is detected on the rx_enapatternalign port.

Manual Bit-Slip Alignment Mode

You can also achieve word alignment by enabling the manual bit-slip option in the MegaWizard. With this option enabled, the transceiver shifts the word boundary MSB to LSB one bit every parallel clock cycle. The transceiver shifts the word boundary every time the bit-slipping circuitry detects a rising edge of the rx_bitslip signal. At each rising edge of the rx_bitslip signal, the word boundary slips one bit. The bit that arrives at the receiver first is skipped. When the word boundary matches the alignment pattern you specified in the MegaWizard, the rx_patterndetect signal is asserted for one clock cycle. You must implement the logic in the PLD logic array to control the bit-slip circuitry.

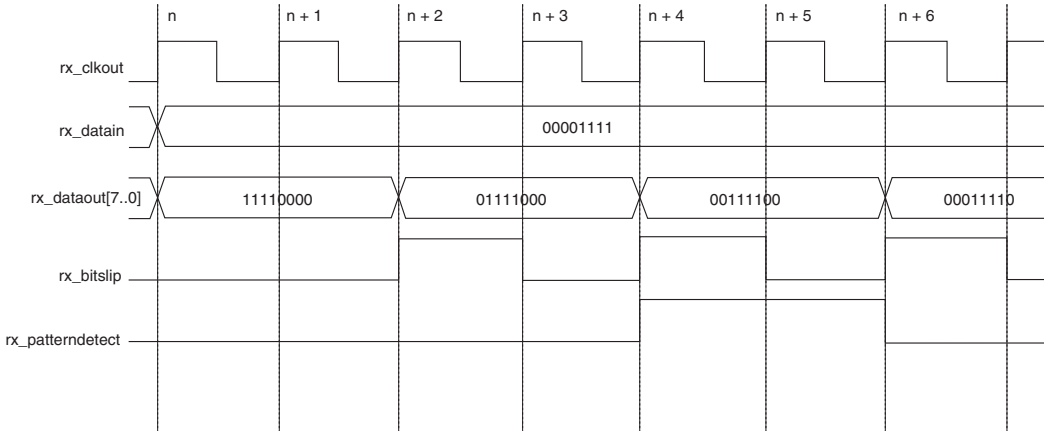
The bit slipper is useful if the alignment pattern changes dynamically when the Stratix II GX device is in user mode. You can implement the controller in the logic array, so you can build a custom controller to dynamically change the alignment pattern without needing to reprogram the Stratix II GX device.

Figure 2–63 shows an example of how the word aligner signals interact in the manual bit-slip alignment mode. For this example, 8'b00111100 is specified as the alignment pattern and an 8'b11110000 value is held at the rx_datain port.

Every rising edge on the rx_bitslip port causes the rx_dataout data to shift one bit from the MSB to the LSB by default. This is shown at time $n + 2$ where the 8'b11110000 data is shifted to a value of 8'b01111000. At this state the rx_patterndetect signal is held low because the specified alignment pattern does not exist in the current word boundary.

The rx_bitslip is disabled at time *n* + 3 and re-enabled at time *n* + 4. The output of the rx_dataout now matches the specified alignment pattern, thus the rx_patterndetect signal is asserted for one clock cycle. At time *n* + 5, the rx_patterndetect signal is still asserted because the alignment pattern still exists in the current word boundary. Finally, at time *n* + 6 the rx_dataout boundary is shifted again and the rx_patterndetect signal is deasserted to signify that the word boundary does not contain the alignment pattern.

Figure 2–63. Word Aligner Symbol Interaction in Manual Bit-Slip Mode

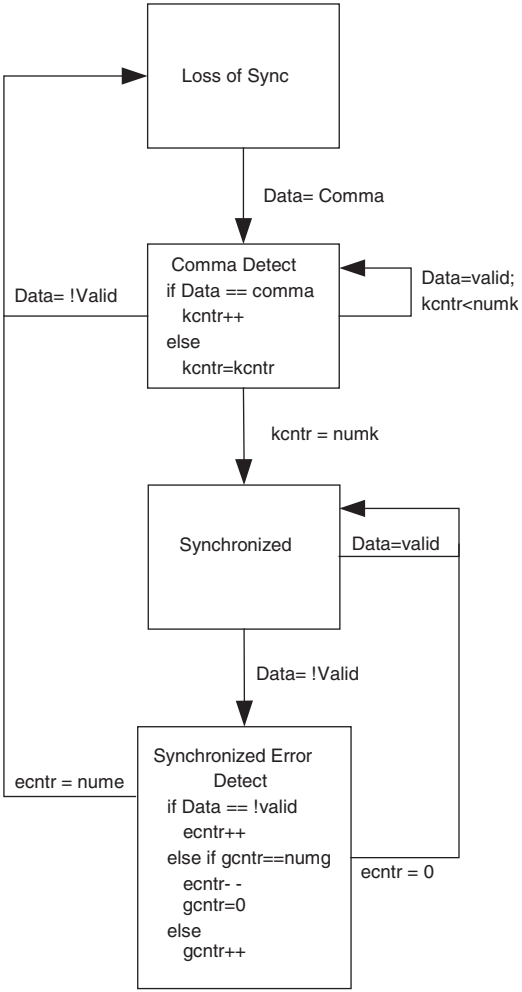


Synchronization State Machine Mode

In single-width mode, you can choose to have the link synchronization handled by a state machine. Unlike the manual alignment mode where there is no built-in hysteresis to go into or fall out of synchronization, the synchronization state machine offers automatic detection of a valid number of alignment patterns and synchronization and detection of code group errors for automatically falling out of synchronization. The synchronization state machine is available in the Basic (single-width mode only), XAUI, GIGE, and PIPE modes. For the XAUI, GIGE, and PIPE modes, the number of alignment patterns, consecutive code groups, and bad code groups are fixed. You must use the 8B/10B code for the synchronization state machine. In XAUI, GIGE, and PIPE modes, the 8B/10B encoder/decoder is embedded in the transceiver data path. In Basic single-width mode, you can configure the MegaWizard to either use or bypass the 8B/10B encoder/decoder in the transceiver. If the synchronization state machine is enabled and the 8B/10B encoder/decoder is bypassed, the 8B/10B encoder/decoder logic must be implemented outside the transceiver as a requirement for using the synchronization state machine.

In Basic mode, you can configure the state machine to suit a variety of standard and custom protocols. In the MegaWizard, you can program the number of alignment patterns to acquire link synchronization. You can program the number of bad code groups to fall out of synchronization. You can program the number of good code groups to negate a bad code group. You enter these values in the MegaWizard. The `rx_syncstatus` port indicates the link status. A high level indicates link synchronization is achieved, a low level indicates that synchronization has not yet been achieved or that there were enough code group errors to fall out of synchronization. Figure 2–64 shows a flowchart of the synchronization state machine.

Figure 2–64. Word Aligner Synchronization State Machine Flow Chart



The maximum value for the number of valid alignment patterns and good code groups is 256. The maximum value of invalid or bad code groups to fall out of synchronization is 8. For example, if 3 is set for the number of good code groups, then when 3 consecutive good code groups are detected after a bad code group, the effect of the bad code group on synchronization is negated. This does not negate the bad code group that actually triggers the loss of synchronization. To negate a loss of synchronization, the protocol-defined number of alignment patterns must be received.

When either XAUI or GIGE mode is used, the synchronization and word alignment is handled automatically by a built-in state machine that adheres to either the IEEE 802.3ae or IEEE 802.3 synchronization specifications, respectively. If you specify either standard, the alignment pattern is automatically defaulted to /K28.5/ (b'0011111010).

When you specify the XAUI protocol, code-group synchronization is achieved upon the reception of four /K28.5/ commas. Each comma can be followed by any number of valid code groups. Invalid code groups are not allowed during the synchronization stage. When code-group synchronization is achieved the optional `rx_syncstatus` signal is asserted. Refer to clause 47-48 of the IEEE P802.3ae standard or "XAUI Mode" on page 2–167 for more information regarding the operation of the synchronization phase.

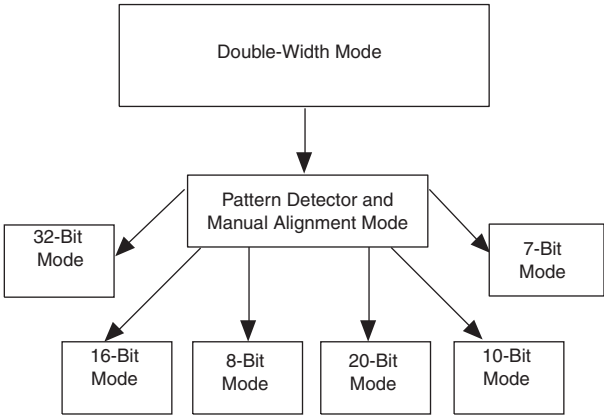
If you specify the GIGE protocol, code-group synchronization is achieved upon the reception of three consecutive ordered sets. An ordered set starts with the /K28.5/ comma and can be followed by an odd number of valid data code groups. Invalid code groups are not allowed during the reception of three ordered-sets. When code-group synchronization is achieved the optional `rx_syncstatus` signal is asserted.

In PIPE mode, lane synchronization is achieved when the word aligner sees 4 good /K28.5/ commas and 16 good code groups. This is accomplished through the reception of 4 good PCI Express training sequences (TS1 or TS2). The PCI-Express fast training sequence (FTS) can also be used to achieve lane or link synchronization, but requires at least five of these training sequences. The `rx_syncstatus` signal is asserted when synchronization is achieved and is deasserted when the word aligner receives 23 code group errors.

Double-Width Mode

In the double-width mode, there are two blocks active in the word aligner: the pattern detector and manual alignment mode. The pattern detector detects if the pattern exists in the current word boundary. The manual alignment identifies the alignment pattern across the byte boundaries and aligns to the correct byte boundary. There are no synchronization state machines available for the double-width mode.

Figure 2–65. Word Aligner Components, Double-Width Mode



Pattern Detector Module

The pattern detector matches a pre-defined comma to the current word boundary. If the pattern detector locates the correct alignment pattern, the optional `rx_patterndetect` signal is asserted for the duration of one clock cycle to signify that the alignment pattern exists in the current word boundary. The pattern detector module only indicates that the signal exists and does not modify the word boundary.

You can program and then specify in the MegaWizard a 7-bit, 8-bit, 10-bit, 16-bit, 20-bit, or 32-bit, pattern for the pattern detector to recognize in double-width mode. For the 7-bit, 10-bit, and 20-bit patterns, the actual and complement of the alignment patterns are checked and used with 8B/10B coding. For the 8-bit, 16-bit, and 32-bit alignment patterns, only the actual patterns are checked, not the complements. These patterns are used for scrambled coding or non-encoded data.

7-Bit Pattern Mode

In the 7-bit pattern detection mode, the pattern detector matches the seven least significant bits of the 10-bit alignment pattern in the LSByte as specified in the MegaWizard in the current word boundary. The pattern detector checks both positive and negative disparities in this mode.

The 7-bit pattern mode can mask out the three most significant bits of the data, which allows the pattern detector to recognize multiple alignment patterns. For example, in the 8B/10B encoded data, a /K28.5/ (b'0011111010), /K28.1/ (b'0011111001), and /K28.7/ (b'0011111000) share seven common LSBs, so masking the three MSBs allows the pattern

detector to resolve all three alignment patterns and indicate it on the rx_patterndetect port. If the alignment pattern appears on the MSByte, no actions are taken by the pattern detect module.

8-Bit Pattern Mode

You can enable a single 8-bit character (A1) detection in the MegaWizard. In this mode, the pattern detector detects a single 8-bit alignment pattern character in the LSByte of the data path. If the alignment pattern appears in the MSByte, no action is taken by the pattern detector.

10-Bit Pattern Mode

In the 10-bit pattern detection mode, the pattern detector matches the 10-bit alignment pattern you specified in the MegaWizard to the LSByte data and its complement in the current word boundary. The pattern detector checks both positive and negative disparities in this mode. For example, if you specified a /K28.5/ (b'0011111010) pattern as the alignment pattern, the rx_patterndetect signal is asserted if b'0011111010 or b'1100000101 is detected in the incoming data. If the alignment pattern appears on the MSByte, no action is taken by the pattern detector.

16-Bit Pattern Mode

You enable the two consecutive 8-bit characters (A1A2).

You specify the 16-bit alignment pattern, which has the bit orientation of [MSB..LSB], in the MegaWizard. A1 represents the least significant byte, which consists of bits [7..0]. A2 represents the most significant byte, which consists of bits [15..8]. Therefore, the alignment pattern is specified as [A2,A1] in the MegaWizard. Only the actual alignment pattern you specified in the MegaWizard is detected in this mode.

20-Bit Pattern Mode

In the 20-bit pattern detection mode, the pattern detector matches the 20-bit comma (K1K2) you specified in the MegaWizard to the incoming data stream. The pattern detector checks the true and complement of the pattern. For example, if you specify a /K28.5/ and /K28.0/ (10'b0011111010, 10'b0011110100) pattern as the alignment pattern, the rx_patterndetect signal is asserted if 10'b0011111010 or 10'b1100000101 and 10'b0011110100 or 10'b1100001011 are detected in the incoming data. You do not need to enter the correct disparity in the MegaWizard, because the true and complement of each code group is checked. In this mode, only Kx.y codes are used as the true and complement to represent the same code group (but different disparity). The Dx.y code group does not necessarily use its true and complement to represent the same code group.

32-Bit Pattern Mode
You enable the four consecutive 8-bit character (A1A2A3A4 or A1A1A2A2) detection in the MegaWizard. You specify the 32-bit alignment pattern in the MegaWizard and it is oriented with the MSB first and the LSB last. A1 represents the least significant byte, which consists of bits `[7..0]`. A4 represents the most significant byte, which consists of bits `[31..24]`. Therefore, the alignment pattern is specified as [A4,A3,A2,A1] in the MegaWizard. Only the actual alignment pattern you specified in the MegaWizard is detected in this mode.

Manual Alignment Modes

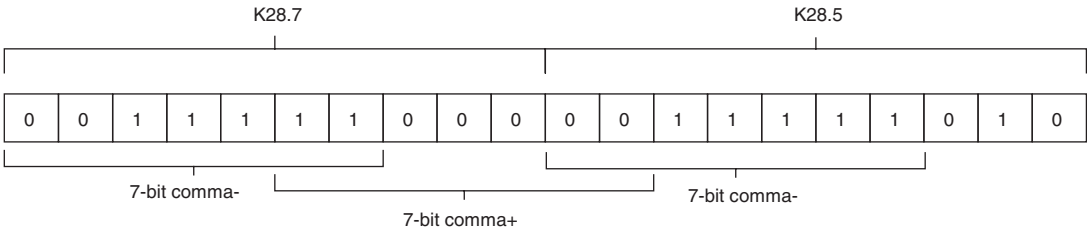
The word aligner has six manual alignment modes (7-, 8- 10-, 16-, 20- and 32-bits) when the transceiver data path is in double-width mode. The 7-, 10-, and 20-bit alignment modes are used with 8B/10B encoded data. Both the actual and complement of the alignment pattern are checked for these modes. The 8-, 16-, and 32-bit alignment modes are for scrambled or non-scrambled data. Only the actual alignment pattern is checked for these modes.

Manual 7-bit Alignment Mode
In the 7-bit alignment mode (use the 8B/10B encoded data with this mode), the module looks for the 7-bit alignment pattern you specified in the MegaWizard Plug-In Manager in the incoming data stream. The 7-bit alignment mode is useful because it can mask out the three most significant bits of the data, which allows the word aligner to align to multiple alignment patterns. For example, in the 8B/10B encoded data, a /K28.5/ (b'0011111010), /K28.1/ (b'0011111001), and /K28.7/ (b'0011111000) share seven common LSBs. Masking the three MSBs allows the word aligner to resolve all three alignment patterns synchronized to it. The word aligner places the boundary of the 7-bit pattern in the LSByte position with bit positions `[0..7]`. The true and complement of the patterns is checked.

In 7-bit manual word alignment mode, the word aligner looks for the 7-bit alignment pattern after detecting a rising edge on the `rx_enapatternalign` signal. On finding the alignment pattern, the word aligner locks the word boundary and asserts the `rx_syncstatus` signal. The `rx_syncstatus` signal remains high until it sees another rising edge on the `rx_enapatternalign`. After detecting a rising edge on the `rx_enapatternalign` signal, the word aligner starts looking for the 7-bit word alignment pattern again and asserts the `rx_syncstatus` signal once it finds the 7-bit alignment pattern. You must differentiate if the acquired byte boundary is correct, because the 7-bit pattern can appear between word boundaries. For example, in the standard 7-bit

alignment pattern -7'b1111100, if a K28.7 is followed by a K28.5, the 7-bit alignment pattern appears on K28.7, between K28.7 and K28.5, and also again in K28.5 (refer to Figure 2–66).

Figure 2–66. Cross Boundary 7-Bit Comma When /K28.7 is Followed by /K28.5



Manual 8-bit Alignment Mode

You can enable the 8-bit alignment mode in the double-width mode. This mode aligns to the 8-bit alignment pattern you specified in the MegaWizard.

The byte boundary is locked after the first alignment pattern is detected and after the rising edge of the `rx_enapatternalign` signal. The detected pattern is placed in the LSByte of the 16-bit word. If the byte boundary changes, the `rx_enapatternalign` port must be deasserted and reasserted to enable the alignment circuit to search for and align to the next available alignment pattern. On the rising edge of the `rx_enapatternalign` signal, the word aligner locks onto the first alignment pattern detected and places the detected pattern in the data stream on the LSByte position. In this scenario, `rx_patterndetect` is asserted to signify that the alignment pattern has been aligned. The `rx_syncstatus` signal is also asserted to signify that the word boundary has been synchronized.

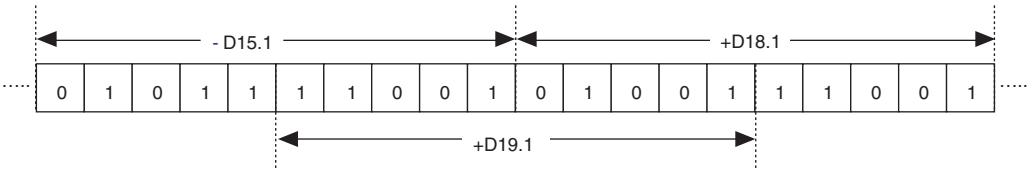
Manual 10-Bit Alignment Mode

You can configure the word aligner to align to a 10-bit word boundary. The internal word alignment circuitry shifts to the correct word boundary if the alignment pattern you specified in the pattern detector is detected in the data stream. The word aligner then puts the alignment pattern in the LSByte of the data path.

In 10-bit manual word alignment mode, the word aligner looks for the 10-bit alignment pattern after detecting a rising edge on the `rx_enapatternalign` signal. On finding the alignment pattern, the word aligner locks the word boundary and asserts the `rx_syncstatus` signal. The `rx_syncstatus` signal remains high until it sees another rising edge on the `rx_enapatternalign`. After detecting a rising edge

on the rx_enapatternalign signal, the word aligner starts looking for the 10-bit word alignment pattern again and asserts the rx_syncstatus signal once it finds the 10-bit alignment pattern. The rx_enapatternalign port only operates in an edge-sensitive fashion in double-width mode. You must deassert the rx_enapatternalign signal and assert it again for re-alignment. Altera recommends using the /K28.5/ code group as one of the control codes for this alignment pattern. For example, assume that 8B/10B coding is used and a /+D19.1/ (b'110010 1001) character is specified as the alignment pattern. In that case, a false word boundary is detected if a /-D15.1/ (b'010111 1001) is followed by a /+D18.1/ (b'010011 1001). Refer to Figure 2–67.

Figure 2–67. False Word Boundary Alignment if Alignment Pattern Exists Across Word Boundaries, Double Width

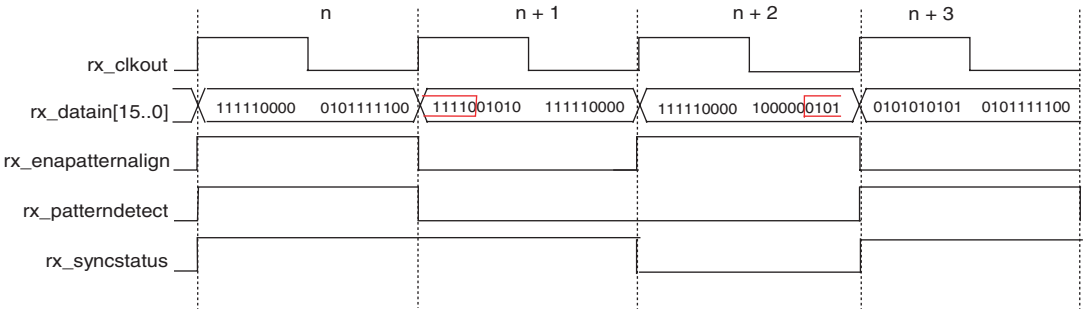


If there is no rising edge on the rx_enapatternalign port, the current word boundary is locked, even if the alignment pattern is detected across different boundaries.

Figure 2–68 shows an example of how the word aligner signals interact in 10-bit alignment mode. For this example, a /K28.5/ (10'b0011111010) is specified as the alignment pattern. The rx_enapatternalign signal is as a rising edge at time *n*, alignment occurs whenever an alignment pattern exists in the pattern. The rx_patterndetect signal is asserted for one clock cycle to signify that the pattern exists on the re-aligned boundary. The rx_syncstatus signal also gets asserted to signify that the boundary has been synchronized. At time *n* + 1, the rx_enapatternalign signal is deasserted and no rising edge occurs, which instructs the word aligner to lock the current word boundary. The alignment pattern is detected at time *n* + 2, but it exists on a different boundary than the current locked boundary. Figure 2–68 shows that the alignment pattern exists across time *n* + 1 and *n* + 2. In this condition the rx_patterndetect signal remains low because the alignment pattern does not exist on the current word boundary. It is up to the user logic to decide whether or not to assert the rx_enapatternalign signal to re-initiate the word alignment process. At time *n* + 3, the rx_patterndetect signal is asserted for one clock cycle to signify that

the alignment pattern has been detected on the LSByte of the current word boundary. If the pattern exists on the MSByte, the rx_syncstatus signal goes high.

Figure 2–68. Word Aligner Symbols Interacting in 10-Bit Manual Alignment Mode



Manual 16-Bit Alignment Mode

You can enable the 16-bit alignment mode in the double-width mode. This mode aligns to the 16-bit alignment pattern you specified in the MegaWizard.

The byte boundary is locked after the pattern detector detects the first alignment pattern and then after the rising edge of the rx_enapatternalign port. If the byte boundary changes, the rx_enapatternalign port must be deasserted and reasserted to enable the alignment circuit to search for and align to the next available alignment pattern. On the rising edge of the rx_enapatternalign signal, the word aligner locks onto the first alignment pattern detected. In this scenario the rx_patterndetect signal is asserted to signify that the alignment pattern has been aligned. The rx_syncstatus signal is also asserted to signify that the word boundary has been synchronized.

Manual 20-bit Alignment Mode

In the 20-bit alignment mode, the pattern detector looks for the 20-bit alignment pattern (K1K2) you specified in the MegaWizard in the incoming data stream. The pattern detector checks the true and complement of the pattern. For example, if you specified a /K28.5/ and /K28.0/ (10'b0011111010, 10'b0011110100) pattern as the alignment pattern, the byte boundary is set to the pattern boundary when 10'b0011111010 or 10'b1100000101 and 10'b0011110100 or 10'b1100001011 are detected in the incoming data. It is not necessary to enter the correct disparity in the MegaWizard because the true and complement of each code group is checked automatically by the pattern detector. Do not use Dx.y codes as the alignment pattern in the 20-bit alignment mode.

In 20-bit manual word alignment mode, the word aligner looks for the 20-bit alignment pattern after detecting a rising edge on the rx_enapatternalign signal. On finding the alignment pattern, the word aligner locks the word boundary and asserts the rx_syncstatus signal. The rx_syncstatus signal remains high until it sees another rising edge on the rx_enapatternalign. After detecting a rising edge on the rx_enapatternalign signal, the word aligner starts looking for the 20-bit word alignment pattern again and asserts the rx_syncstatus signal once it finds the 20-bit alignment pattern. The rx_enapatternalign port can only operate in an edge-sensitive fashion in double-width mode. Deassertion of the rx_enapatternalign port is necessary for realignment. Altera recommends that you include the /K28.5/ code group as one of the control codes in this alignment pattern.

Manual 32-Bit Alignment Mode

You can enable the 32-bit alignment mode in the double-width mode only. This mode aligns to the 32-bit alignment pattern you specified in the MegaWizard.

The byte boundary is locked after the first alignment pattern is detected and then after the rising edge of the rx_enapatternalign port. If the byte boundary changes, the rx_enapatternalign port must be deasserted and reasserted to enable the alignment circuit to search for and align to the next available alignment pattern. On the rising edge of rx_enapatternalign, the word aligner locks onto the first alignment pattern detected. In this scenario, the rx_patterndetect is asserted to signify that the alignment pattern has been aligned. The rx_syncstatus signal is asserted to signify that the word boundary has been synchronized.

Manual Bit-Slipping Alignment Mode

In the double-width mode, word alignment is also achieved by enabling the manual bit-slip option in the MegaWizard. This mode operates the same way as the bit slip in the single-width mode. With this option enabled, the transceiver shifts the word boundary one bit from the MSB to the LSB every parallel clock cycle. This occurs every time the bit-slipping circuitry detects a rising edge of the rx_bitslip signal. At each rising edge of rx_bitslip, the word boundary slips one bit. The bit that arrives at the receiver first is skipped. When the word boundary matches what you specified as the alignment pattern in the MegaWizard, the rx_patterndetect signal is asserted for one clock cycle. You must implement the logic in the PLD logic array to control the bit-slip circuitry.

Run-Length Violation Detection Circuit

The programmable run-length violation circuit resides in the word aligner block and detects consecutive 1s or 0s in the data. If the data stream exceeds the preset maximum number of consecutive 1s or 0s, the violation is signified by the assertion of the `rx_rlv` signal.

This signal is not synchronized to the parallel data and appears in the logic array earlier than the run-length violation data. To ensure that the PLD can latch this signal in systems where there are frequency variations between the recovered clock and the PLD logic array clock, the `rx_rlv` signal is asserted for a minimum of two clock cycles in single-width modes and a minimum of three clock cycles in double-width modes. The `rx_rlv` signal may be asserted longer, depending on the run-length of the received data.

In single-width mode, the run-length violation circuit detects up to a run length of 128 (for an 8-bit deserialization factor) or 160 (for a 10-bit deserialization factor). The settings are in increments of 4 or 5 for the 8-bit or 10-bit deserialization factors, respectively.

In double-width mode, the run-length violation circuit maximum run-length detection is 512 (with a run-length increment of 8) and 640 (with a run-length increment of 10) for the 16-bit and 20-bit deserialization factors, respectively.

Table 2–21 summarizes the detection capabilities of the run-length violation circuit.

Table 2–21. Run-Length Violation Circuit Minimum and Maximum Range

Data Path	Deserialization Factor	Run-Length Violation Detector Range	
		Minimum	Maximum
Single-width	8	4	128
	10	5	160
Double-width	16	8	512
	20	10	640

Receiver Bit Reversal

By default, the Stratix II GX receiver assumes an LSBit to MSBit transmission. If the transmission order is MSBit to LSBit, then the receiver will put out the bit-flipped version of the data on the PLD interface. The Receiver Bit Reversal feature is available to correct this situation.

The Receiver Bit Reversal feature is available only in Basic single-width and Basic double-width modes. If the Receiver Bit Reversal feature is enabled in Basic single-width mode, the 10-bit data `D[9:0]` at the output of the word aligner gets rewired to `D[0:9]`. If the Receiver Bit Reversal feature is enabled in Basic double-width mode without the 8B/10B decoder, the MSByte `D[15:8]` and LSByte `D[7:0]` at the output of the word aligner get rewired to `D[8:15]` and `D[0:7]`, respectively. If the Receiver Bit Reversal feature is enabled in Basic double-width mode with the 8B/10B decoder, the MSByte `D[19:0]` and LSByte `D[9:0]` at the output of the word aligner get rewired to `D[0:19]` and `D[0:9]`, respectively. Flipping the parallel data using this feature allows the receiver to put out the correctly bit-ordered data on the PLD interface in case of MSBit to LSBit transmission.

Since the receiver bit reversal is done at the output of the word aligner, a dynamic bit reversal would also require a reversal of word alignment pattern. As a result, the Receiver Bit Reversal feature is dynamic only if the receiver is dynamically reconfigurable (allows changing the word alignment pattern dynamically) or uses manual bit slip alignment mode (no word alignment pattern). The Receiver Bit Reversal feature is static in all other Basic mode configurations and can be enabled through the MegaWizard Plug-In. In configurations where this feature is dynamic, an `rx_revbitordwa` port is available to control the bit reversal dynamically. A high on the `rx_revbitordwa` port reverses the bit order at the input of the word aligner.

Figure 2–69 illustrates the receiver bit reversal feature in Basic single-width 10-bit wide data path configuration.

Figure 2–69. Receiver Bit Reversal in Single-Width Mode

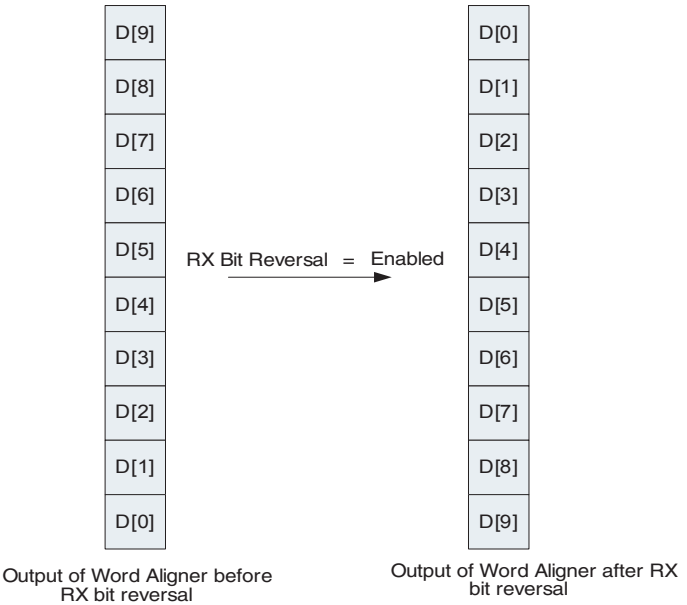
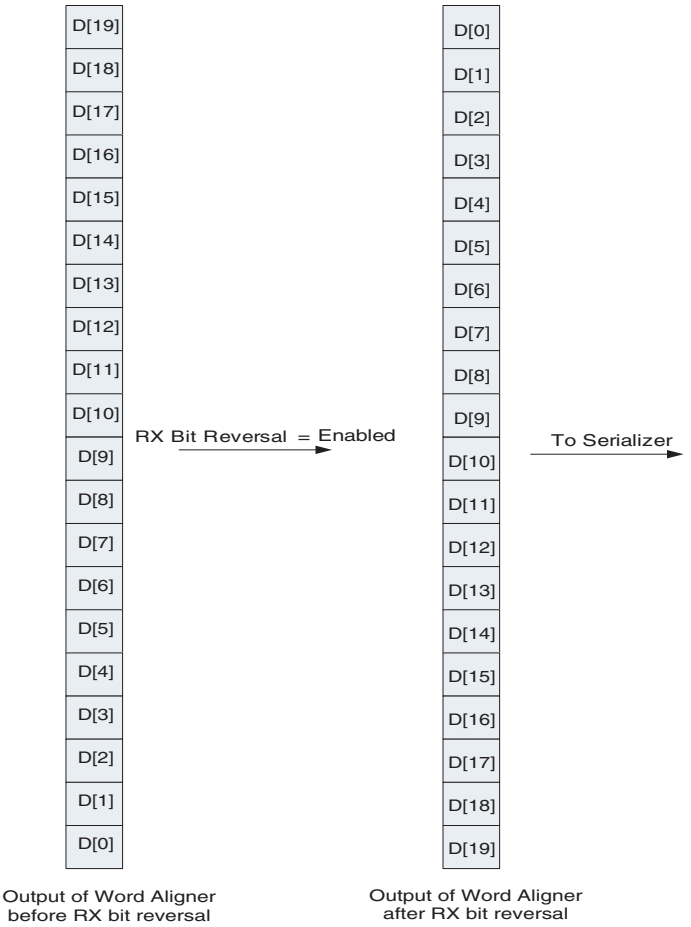


Figure 2–70 illustrates the receiver bit reversal feature in Basic double-width 20-bit wide data path configuration.

Figure 2–70. Receiver Bit Reversal in Double-Width Mode



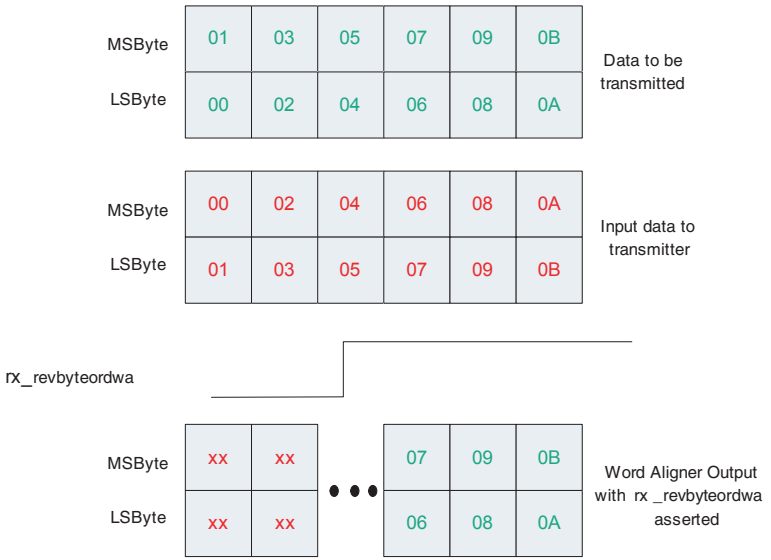
Receiver Byte Reversal

The MSByte and LSByte of the input data to the transmitter are often erroneously swapped. The receiver byte reversal feature is available to correct this situation.

An optional port `rx_revbyteordwa` is available only in Basic double-width mode to enable receiver byte reversal. In 8B/10B enabled mode, a high value on `rx_revbyteordwa` swaps the 10-bit MSByte and LSByte of the 20-bit word at the output of the word aligner in the receiver data path. In non 8B/10B enabled mode, a high value on `rx_revbyteordwa` swaps the 8-bit MSByte and LSByte of the 16-bit word at the output of the word aligner in the receiver data path. This compensates for the erroneous swapping at the transmitter and hence corrects the data received by the downstream systems. The `rx_revbyteorderwa` is a dynamic signal and may cause an initial disparity error at the receiver of an 8B/10B encoded link. The downstream system must be able to tolerate this disparity error.

Figure 2–71 illustrates the receiver byte reversal feature.

Figure 2–71. Receiver Byte Reversal Feature



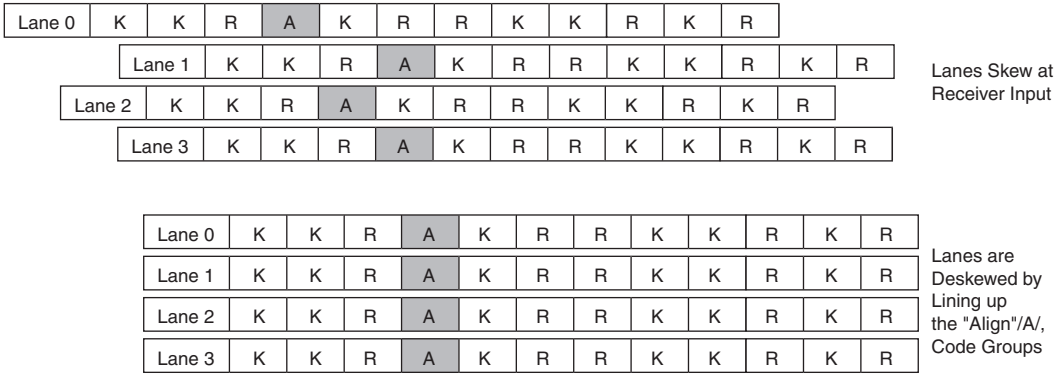
Channel Aligner (Deskew)

The channel aligner is automatically used when implementing the XAUI protocol to ensure that the channels are aligned with respect to each other. The channel aligner uses a 16-word-deep FIFO buffer.

The channel aligner is only available in the XAUI mode.

It is possible for ordered sets to be misaligned with respect to one another because of board skew or differences between the independent clock recoveries per serial lane. Channel alignment, also referred to as deskew or channel bonding, realigns the ordered sets by using the alignment code group, referred to as /A/. The /A/ code group is transmitted simultaneously on all four lanes, constituting an ||A|| ordered set, during idles or inter-packet gaps (IPG). XAUI receivers use these code groups to resolve any lane-to-lane skew. Skew between the lanes can be up to 40 UI (12.8ns) as specified in the standard, which relaxes the board design constraints. Figure 2–72 shows lane skew at the receiver input and how the deskew circuitry uses the /A/ code group to deskew the channels.

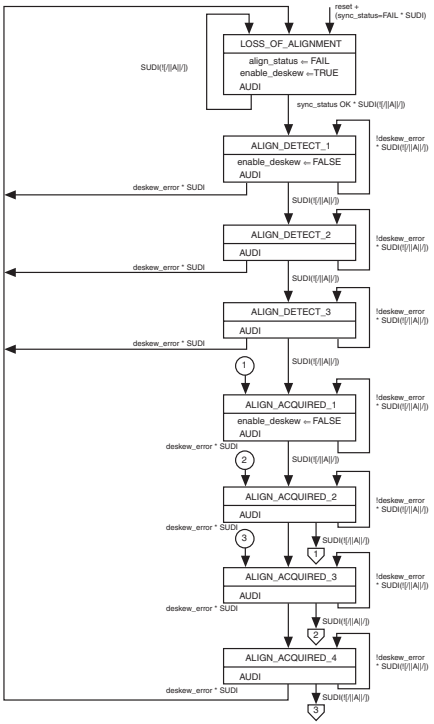
Figure 2–72. Lane Deskew With /A/ Code Group



Stratix II GX devices manage XAUI channel alignment with a dedicated deskew macro that consists of a 16-word-deep FIFO buffer that is controlled by a XAUI deskew state machine. The XAUI deskew state machine first looks for the /A/ code group within each channel. When the XAUI deskew state machine detects /A/ in each channel, the deskew FIFO buffer is enabled. The deskew state machine now monitors the reception of /A/ code groups. When four aligned /A/ code groups are received, the `rx_channelaligned` signal is asserted. The deskew state machine continues to monitor the reception of /A/ code groups and de-asserts the `rx_channelaligned` signal if alignment conditions are

lost. This built-in deskew macro is only enabled for the XAUI protocol. The PCS deskew state diagram specified in clause 48 of the IEEE P802.3ae is shown in Figure 2–73.

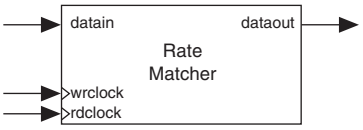
Figure 2–73. IEEE 802.3ae PCS Deskew State Diagram



Rate Matcher

The rate matcher (Figure 2–74) compensates for clock frequency differences between the upstream transmitter and the local receiver. The rate matcher operates in five modes: GIGE, XAUI, PIPE, Basic single-width mode, and Basic double-width mode.

Figure 2–74. Rate Matcher



The transceiver block can operate in multi-crystal environments, which can tolerate frequency variations of ± 300 PPM between crystals. Stratix II GX devices have embedded circuitry to perform clock rate compensation. Clock rate compensation is achieved by inserting or removing skip characters from the IPG or idle streams. This process is called rate matching or clock rate compensation.

The rate matcher in the transceiver consists of a 20-word-deep FIFO buffer and necessary logic to detect and perform the insertion and deletion functions.

XAUI

The rate matcher in XAUI mode operates in a synchronized ×4 mode and supports up to a ±100 PPM clock difference between the upstream transmitter and receiver. In this mode, the rate matcher can insert or delete a column of /R/ characters as denoted by the ||R|| designation, depending on whether the FIFO buffer is approaching an empty or full condition. The rate matcher does not operate until the XAUI synchronization state machine achieves word alignment and channel alignment. Until that point, the rate matcher is not active (read and write pointers do not move).

If the ||R|| code words are not received on all channels, rate matching does not occur and may lead to over/underflow conditions in the rate matching FIFO buffer. If this situation occurs, the data output of the receiver outputs a constant 9'h19C (8'h9C on the `rx_dataout` output and 1'b1 on the `rx_ctrldetect` output) in lane 0 (rest of the lane are data 8'h00). The receiver digital reset must be asserted and the lanes resynchronized before data can be received.

GIGE

The rate matcher in GIGE mode operates in a channel-by-channel mode and supports up to a ± 100 PPM clock difference between the upstream transmitter and the receiver. The rate matcher either inserts or deletes the /I2/ ordered set depending on whether the FIFO buffer is approaching an empty or full condition. The /I2/ order set consists of a /K28.5+/ code group and a /D16.2-/ code group (the sign after the code group signifies the running disparity at the end of the code group). The rate matcher in GIGE mode waits until the GIGE synchronization state machine achieves synchronization. Once synchronization is achieved, the rate matcher is active.

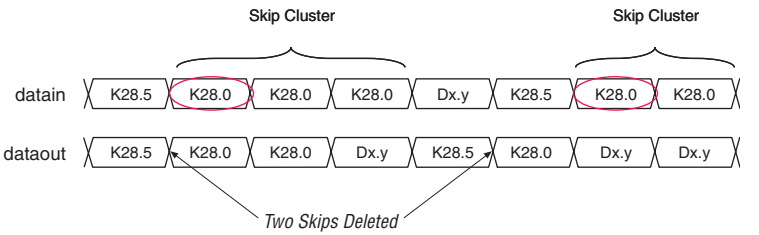
In the event the rate matching FIFO buffer in the GIGE mode approaches overflow or underflow, the transceiver outputs a sequence code group (9'h19C)—8'h9C on `rx_dataout` and 1'b1 on `rx_ctrldetect`. The `rx_digitalreset` signal must be asserted to reset the rate matcher FIFO buffer.

PIPE Mode

In PIPE mode, the rate matcher supports up to ±300 PPM (600 PPM total) differences between the upstream transmitter and the receiver. The rate matcher looks for the skip ordered set, which is usually a /K28.5/ comma followed by three /K28.0/ skip characters. The rate matcher deletes or inserts skip characters when necessary to prevent the rate-matching FIFO buffer from overflowing or underflowing.

The rate matcher can delete only one skip character in a consecutive cluster of skip characters in PIPE mode only. Figure 2–69 shows a PIPE mode rate matcher deletion of two skip characters.

Figure 2–75. PIPE Mode With Two Deletions (One Per Cluster)



The rate matcher can perform skip character insertion one insertion per skip cluster in PIPE mode. There is no limit on the consecutive number of skip characters allowed per skip cluster.

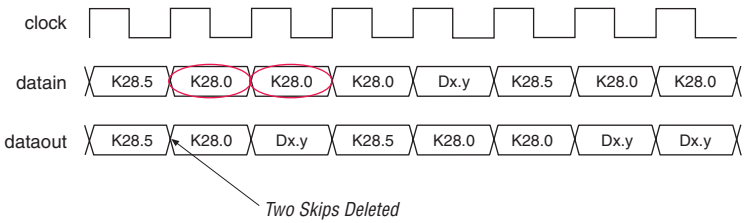
The Stratix II GX rate matcher in PIPE mode has FIFO buffer overflow and underflow protection. In the event of a FIFO buffer overflow, the rate matcher deletes any data after the overflow condition to prevent FIFO pointer corruption until the rate matcher is not full. In an underflow condition, the rate matcher inserts 9'h1FE (/K30.7/) until the FIFO buffer is not empty. These measures ensure that the FIFO buffer gracefully exits the overflow and underflow condition without requiring a FIFO buffer reset.

Single-Width General Rate Matching

In Basic single-width mode, the rate matcher supports up to ±300 PPM differences between the upstream transmitter and the receiver. The rate matcher looks for the skip ordered set, which is a /K28.5/ comma followed by three programmable neutral disparity skip characters (for example, /K28.0/). For general rate matching, you can customize the SOS to support a variety of protocols, including custom protocols. The SOS must contain a valid control code group (Kx.y), followed by any neutral disparity skip code group (any Kx.y or Dx.y of neutral disparity, for example, K28.0). The rate matcher deletes or inserts skip characters when necessary to prevent the rate matching FIFO buffer from overflowing or underflowing.

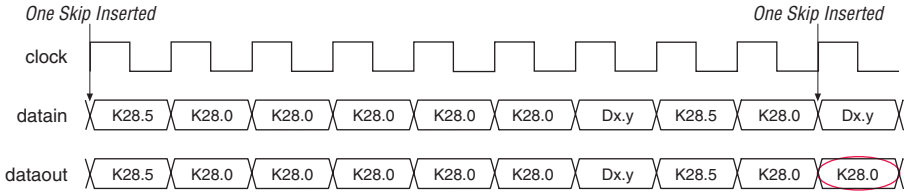
The rate matcher in single-width mode can delete any number of skip characters as necessary in a cluster as long as there are skip characters to delete. There are no restrictions regarding deleting more than one skip character in a cluster of skip characters. Figure 2–76 shows an example of a single-width mode rate matcher deletion of two skip characters. Although the skip characters are programmable, the /K28.0/ control group is used for illustration purposes.

Figure 2–76. Single-Width Mode Deletion of Two Skip Characters



The rate matcher inserts skip characters as required for rate matching. For a given skip ordered set, the rate matcher inserts skip characters so that the total number of consecutive skip characters does not exceed five at the output of the rate matching FIFO buffer. Figure 2–77 shows an example where a skip character insertion is made on the second set of skip ordered sets because the first set has the maximum number of skip characters.

Figure 2–77. Single-Width Mode Insertion of a Skip Character



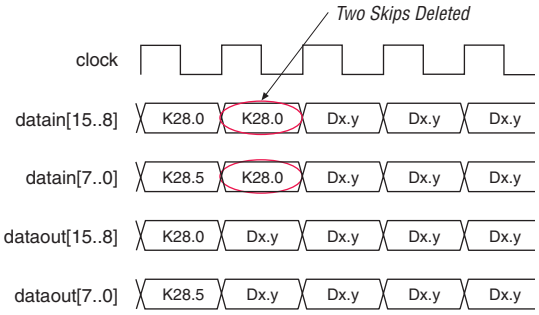
The Stratix II GX rate matcher in single-width mode has FIFO buffer overflow and underflow protection. In the event of a FIFO buffer overflow the rate matcher deletes any data after the overflow condition to prevent FIFO buffer pointer corruption until the rate matcher is not full. In an underflow condition, the rate matcher inserts 9'h1FE (/K30.7) until the FIFO buffer is not empty. These measures ensure that the FIFO buffer gracefully exits the overflow and underflow condition without requiring a FIFO buffer reset.

Double-Width General Rate Matching

In double-width mode, the rate matcher can support up to ±300 PPM differences between the upstream transmitter and the receiver. The rate matcher looks for the skip ordered set, which is usually a /K28.5/ comma followed by programmable neutral disparity skip characters (for example, /K28.0/). For general rate matching, you can customize the SOS to support a variety of protocols, including custom protocols. The SOS must contain a valid control code group (Kx.y), followed by any neutral disparity skip code group (any Kx.y or Dx.y of neutral disparity, for example, K28.0). The rate matcher deletes or inserts dual skip characters when necessary to prevent the rate matching FIFO buffer from overflowing or underflowing.

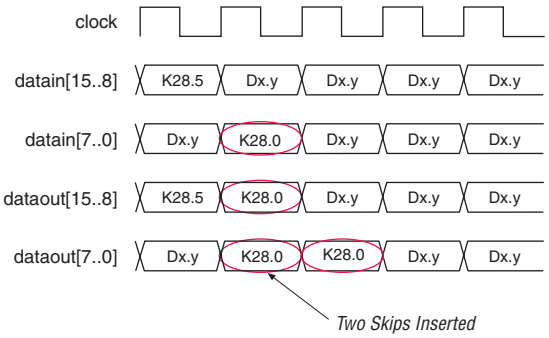
The rate matcher deletes skip characters by pairs when they appear on the upper and lower bytes at the same time. There are no other restrictions for the deletion of skip characters. Figure 2–78 shows an example of deleting two skip characters.

Figure 2–78. Deletion of Skip Characters in Double-Width Mode



The rate matcher inserts skip characters by pairs on the upper and lower byte (Figure 2–79). The insertion occurs after a /K28.5/ or the programmed control code group is detected by the rate matcher. If the comma is detected on the lower byte, the high byte must be a skip character. The insertion happens on the next double byte, if needed. If the comma appears on the upper byte, and the skip character is in the next lower byte, the insertion occurs on the double byte after the comma character, if needed.

Figure 2–79. Insertion of Skip Characters in Double-Width Mode

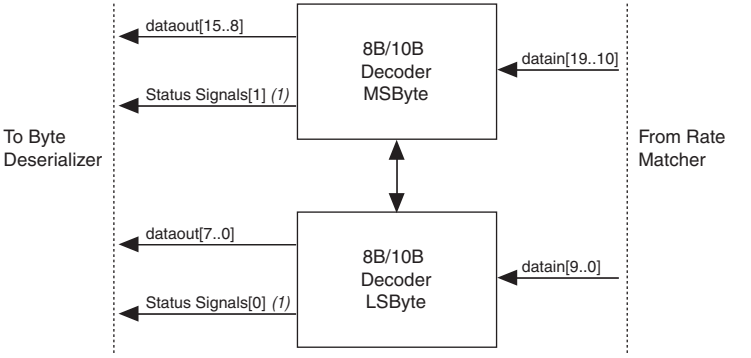


8B/10B Decoder

The 8B/10B decoder (Figure 2–80) is part of the Stratix II GX transceiver digital blocks and lies in the receiver path between the rate matcher and the byte deserializer blocks. The 8B/10B decoder operates in two modes: single-width and double-width modes and can be bypassed if 8B/10B decoding is not needed. In single-width mode, the 8B/10B decoder

restores the 8-bit data + 1-bit control identifier from the 10-bit code. In double-width mode, there are two 8B/10B decoders cascaded together, which restores the 16-bit (2× 8-bit) data + 2-bit (2× 1-bit) control identifier from the 20-bit (2× 10-bit) code. This 8B/10B decoder conforms to the IEEE 802.3 1998 edition standards.

Figure 2–80. 8B/10B Decoder

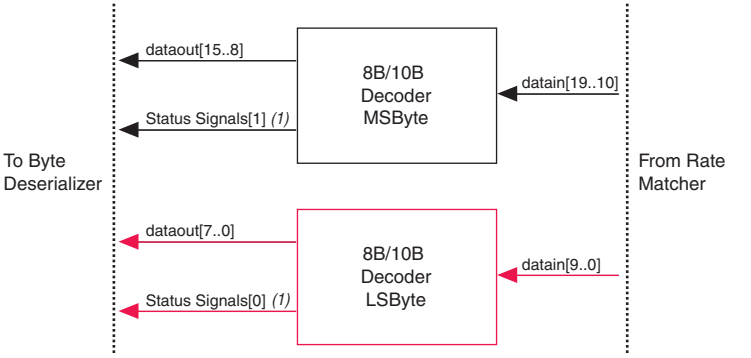


Note to Figure 2–80:
(1) Status signals include `rx_ctrldetect`, `rx_disperr`, and `rxerrdectect`.

Single-Width Mode

In single-width mode, the Stratix II GX 8B/10B decoder operates in a similar fashion as the Stratix GX 8B/10B decoder. The highlighted data path in Figure 2–81 is active in the single-width mode.

Figure 2–81. Active Data Path in Single-Width Mode

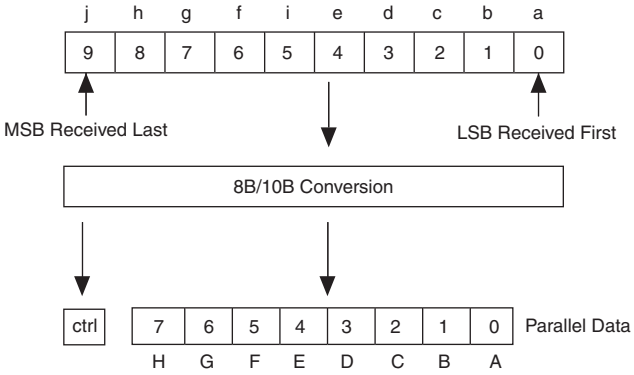


Note to Figure 2–81:
(1) Status signals include `rx_ctrldetect`, `rx_disperr`, and `rxerrdectect`.

10-Bit Decoding

The 8B/10B decoder in single-width mode translates the 10-bit encoded code into the 8-bit equivalent data or control code. The 10-bit code received must be from the supported Dx.y or Kx.y list with the proper disparity or error flag asserted. All 8B/10B control signals (disparity error, control detect, code error, and so on) are pipelined and edge-aligned with the data. Figure 2–82 shows how the 10-bit symbol is decoded to the 8-bit data, plus a 1-bit control indicator.

Figure 2–82. 10-Bit to 8-Bit Conversion



Code Error Detect

The `rx_errdetect` signal indicates when the code received contains an error. This port is optional but, if not in use, there is no way to detect if a code received is valid or not. The `rx_errdetect` signal goes high if a code received is an invalid code or if it has a disparity error. If a code is received that is not part of the valid Dx.y or Kx.y list, the `rx_errdetect` signal goes high. This signal is aligned to the invalid code word received at the PLD logic array.

In GIGE, XAUI, and PIPE mode, the invalid code is replaced by a /K30.7/ code (8'hFE on `rx_dataout` + 1'b1 on `rx_ctrldetect`). In all other modes, the value of the invalid code value can vary and should be ignored.

Disparity Error Detector

The 8B/10B decoder detects disparity errors based on which 10-bit code it received. The disparity error is indicated at the optional `rx_disperr` port.

Refer to the *Specifications & Additional Information* chapter in volume 2 of the *Stratix II GX Device Handbook* for information on the disparity calculation.

If negative disparity is calculated for the last 10-bit code, a neutral or positive disparity 10-bit code is expected. If the 8B/10B decoder does not receive a neutral or positive disparity 10-bit code, the `rx_disperr` signal goes high, indicating that the code received had a disparity error.

If a positive disparity is calculated, a neutral or negative disparity 10-bit code is expected. The `rx_disperr` signal goes high if the code received is not as expected.

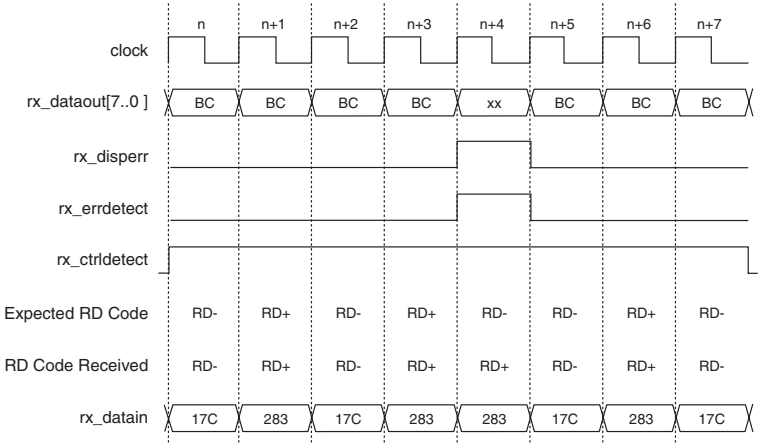
When `rx_disperr` is high, `rx_errdetect` also goes high.

The detection of the disparity error might be delayed, depending on the data that follows the actual disparity error. The 8B/10B control codes terminate propagation of the disparity error. Any disparity errors propagated stop at the control code, terminating that disparity error.

In GIGE and XAUI modes, the code that contains a disparity error is replaced by a /K30.7/ code (8'hFE on `rx_dataout` + `rx_ctrldetect`). In all other modes, the code with incorrect disparity should be treated as an invalid code and ignored.

Figure 2–83 shows a case where the disparity is violated. A K28.5 code has an 8-bit value of 8'hbc and a 10-bit value that depends on the disparity calculation at the point of the generation of the K28.5 code. The 10-bit value is 10'b0011111010 (10'h17c) for RD- or 10'b1100000101 (10'h283) for RD+. If the running disparity at time *n* - 1 is negative, the expected code at time *n* must be from the RD- column. A K28.5 does not have a balanced 10-bit code (equal number of 1s and 0s), so the expected RD code must toggle back and forth between RD- and RD+. At time *n* + 3, the 8B/10B decoder received a RD+ K28.5 code (10'h283), which makes the current running disparity negative. At time *n* + 4, because the current disparity is negative, a K28.5 from the RD- column is expected, but a K28.5 code from the RD+ is received instead. This prompts `rx_disperr` to go high during time *n* + 4 to indicate that this particular K28.5 code had a disparity error. The current running disparity at the end of time *n* + 4 is negative because a K28.5 from the RD+ column was received. Based on the current running disparity at the end of time *n* + 5, a positive disparity K28.5 code (from the RD-) column is expected at time *n* + 5.

Figure 2–83. Disparity Error

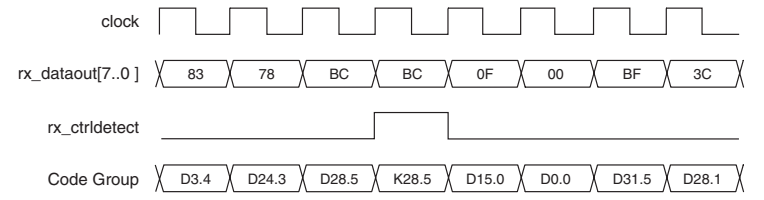


Control Detect

The 8B/10B decoder differentiates between data and control codes through the `rx_ctrldetect` port. This port is optional but, if not in use, there is no way to differentiate a Dx.y from a Kx.y.

Figure 2–84 shows an example waveform demonstrating the receipt of a K28.5 code (BC + ctrl). The `rx_ctrldetect=1'b1` is aligned with 8'hbc, indicating that it is a control code. The rest of the codes received are Dx.y code groups.

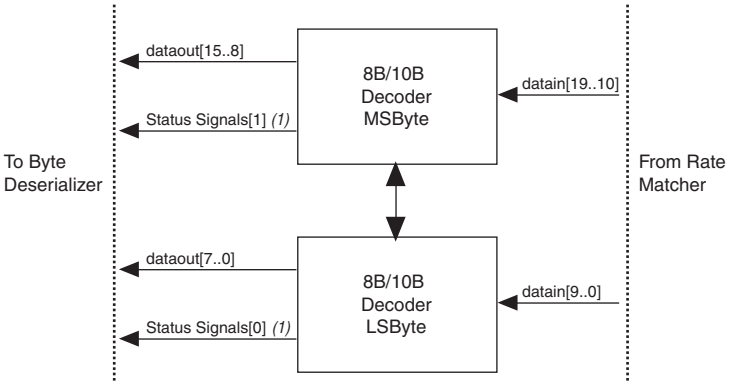
Figure 2–84. Control Code Detection



Double-Width Mode

In double-width mode, the dual 8B/10B decoder operates in cascaded fashion. The LSByte is received first, followed by the MSByte. The highlighted data path in Figure 2–85 is active in the double-width mode.

Figure 2–85. Active Data Path in Double-Width Mode

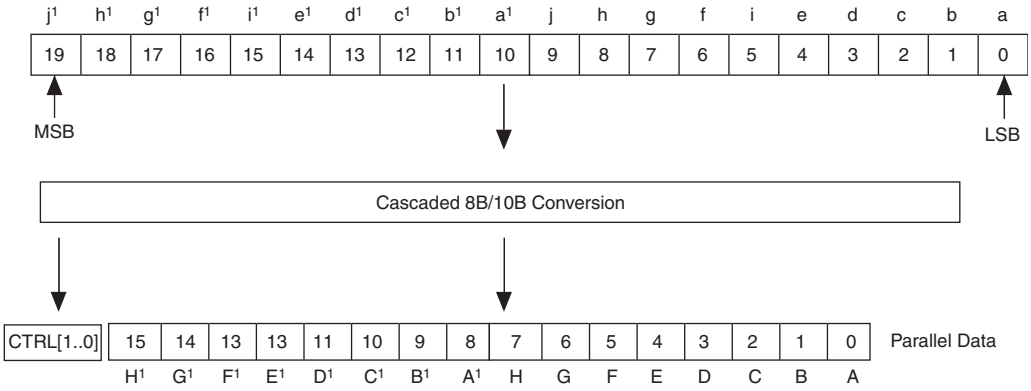


Note to Figure 2–85:
(1) Status signals include `rx_ctrldetect`, `rx_disperr`, and `rxerrdectect`.

20-Bit Decoding

The 8B/10B decoder in double-width mode translates the 20-bit (2× 10-bits) encoded code into the 16-bit (2× 8-bits) equivalent data or control code. The 20-bit upper and lower symbols received must be from the supported Dx.y or Kx.y list with the proper disparity or error flags asserted. All 8B/10B control signals (disparity error, control detect, and code error) are pipelined with the data in the Stratix II GX receiver block and are edge-aligned with the data. Figure 2–86 shows how the 20-bit code is decoded to the 16-bit data plus a 2-bit control indicator.

Figure 2–86. 20-Bit to 16-Bit Conversion



Code Error detect

The `rx_errdetect` signal indicates when a code received contains an error. This port is optional but, if not in use, there is no way to detect if the code received is valid or not. The `rx_errdetect` signal goes high if a code received is an invalid code or if it has a disparity error. If a code is received that is not part of the valid Dx.y or Kx.y list, the `rx_errdetect` signal goes high. This signal is aligned to the invalid code word received at the PLD logic array.

In double-width mode, the `rx_errdetect` signal is 2-bits wide in the PCS portion of the transceiver. The lower bit indicates if the LSByte contains a code error, the upper bit indicates if the MSByte contains a code error. The value of the invalid code can vary and should be ignored.

Disparity Error Detector

The 8B/10B decoder in double-width mode forwards the current running disparity value from the LSByte decoder to the MSByte decoder to check the disparity of the symbol going into the MSByte decoder. The MSByte decoder's ending running disparity is then fed back to the LSByte decoder on the next clock cycle.

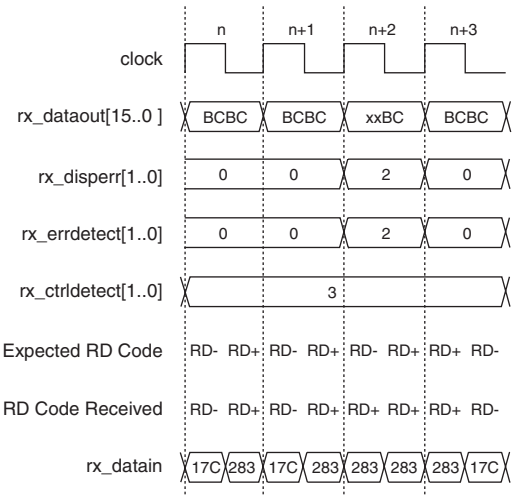
Refer to the *Specifications & Additional Information* chapter in volume 2 of the *Stratix II GX Device Handbook* for information on the disparity calculation.

The rx_disperr port is 2-bits wide in the PCS. When high, the lower bit indicates if the LSByte decoder detected a disparity error in the low byte code conversion. When high, the upper bit indicates if the MSByte decoder detected a disparity error in the high byte code conversion. If both of the rx_disperr bits are low, there is no error.

The detection of the disparity error might be delayed, depending on the data that follows the actual disparity error. The 8B/10B control codes terminate any propagation of the disparity error. Any disparity errors propagated assert rx_disperr on the control code byte, terminating that disparity error.

Figure 2–87 shows a case where the disparity is violated. A K28.5 code has an 8-bit value of 8'hbc and a 10-bit value that depends on the disparity calculation at the point of the generation of the K28.5 code. The 10-bit value is 10'b0011111010 (10'h17c) for RD- or 10'b1100000101 (10'h283) for RD+. This example uses double-width mode and the 20-bit codes are split into two 10-bit codes for clarity. The expected running disparity is indicated for each 10-bit code. At time *n*, rx_datain receives 10'h283 first and the decoded version goes on the LSByte of rx_dataout. At time *n* + 2, the high byte received a K28.5 code of incorrect disparity. The upper bits of the rx_disperr and rx_errdetect ports are asserted, resulting in the 2'h2 values shown in Figure 2–87.

Figure 2–87. Disparity Error



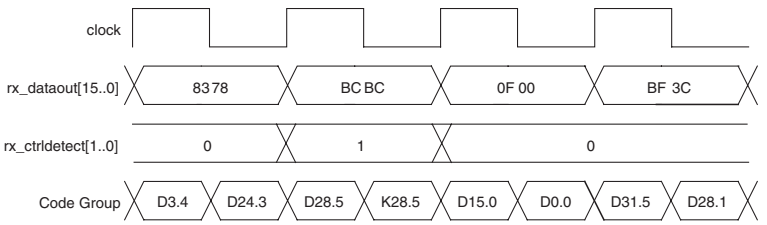
Control Detect

The 8B/10B decoder indicates to the PLD logic array the difference between data and control codes through the `rx_ctrldetect` port. This port is optional but, if not in use, there is no way to differentiate a Dx.y from a Kx.y.

In double-width mode, the `rx_ctrldetect` port is 2-bits wide inside the PCS. The lower bit indicates if the LSByte is a control word or data, and the upper bit indicates if the MSByte is a control word or data. When a control word is decoded, the corresponding `rx_ctrldetect` bit goes high. For data, the corresponding `rx_ctrldetect` bit goes low. The `rx_ctrldetect` port in the PLD logic array is edge-aligned with the code group it is associated with.

Figure 2–88 shows an example waveform that shows the receipt of a K28.5 code (BC + ctrl). The `rx_ctrldetect=2'b1` is aligned with 8'hbc on the LSByte, indicating that it is a control code. The 8'hbc on the MSByte is a data, not a control word. The rest of the codes received are Dk.y control codes.

Figure 2–88. Control Code Detection



Reset

The reset for the 8B/10B decoder block is derived from the receiver digital reset (`rx_digitalreset`). When `rx_digitalreset` is asserted, the 8B/10B decoder block resets. In reset, the disparity registers are cleared and the outputs of the 8B/10B decoder block are driven low. After reset, the 8B/10B decoder starts with either a positive or negative disparity, depending on the disparity of the data it receives. The decoder calculates the initial running disparity based on the first valid code received.

The receiver block must be word aligned after reset before the 8B/10B decoder can decode valid data or control codes. If word alignment has not been achieved, the data from the 8B/10B decoder is discarded and considered invalid.

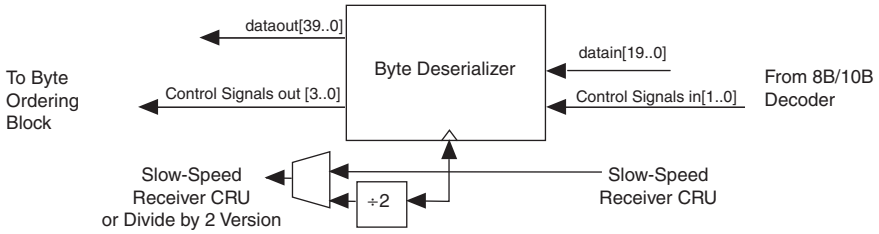
Polarity Inversion

The 8B/10B decoder has a PCI Express compatible polarity inversion on the data bus prior to 8B/10B decoding. This polarity inversion inverts the bits of the incoming data stream prior to the 8B/10B decoding block to fix potential P-N polarity inversion on the differential input buffer. You use the optional `pipe8b10binvpolarity` port to invert the inputs to the 8B/10B decoder dynamically from the PLD.

Byte Deserializer

Use the byte deserializer (Figure 2–89) to convert the one- or two-byte interface into a two- or four-byte-wide data path from the transceiver to the PLD logic (refer to Table 2–22). The PLD interface has a limit of 250 MHz, so the byte deserializer is needed to widen the bus width at the PLD interface and reduce the interface speed. For example, at 6.375 Gbps, the transceiver logic has a double-byte-wide data path that runs at 318.75 MHz in a ×20 deserializer factor, which is above the maximum PLD interface speed.

Figure 2–89. Byte Deserializer



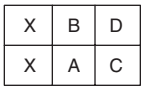
When using the byte deserializer, the PLD interface width doubles to 40-bits (36-bits when using the 8B/10B encoder) and the interface speed drops to 159.375 MHz.

Table 2–22. Byte Deserializer Input and Output Widths

Input Data Width (Bits)	Deserialized Output Data Width to the FPGA Logic Array (Bits)
20	40
16	32
10	20
8	16

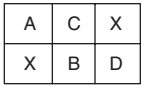
If you use the byte deserializer, the byte ordering might be different than what you intended. Figure 2–90 shows the byte deserializer operating in single-width mode. The expected data pattern is A at the lower byte, followed by B at the upper byte. C and D follow in the next lower and upper bytes, respectively.

Figure 2–90. Intended Transmitter Pattern



The receiver may receive the intended transmitter pattern or slip a byte, as shown in Figure 2–91, where A arrives when the byte deserializer is stuffing the upper byte instead of stuffing the lower byte. This is a nondeterministic swap, because it depends on PLL lock times and link delay.

Figure 2–91. Incorrect Byte Position at Receiver After Byte Deserializer



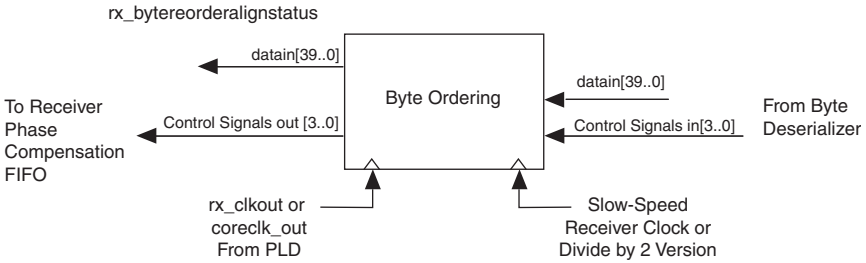
You can use the byte ordering block or a byte reordering circuit to restore the byte order to the expected pattern.

Byte Ordering

The Stratix II GX device has a dedicated byte ordering circuit on each receiver to obtain a certain byte order on multiple lanes. This circuit is used in conjunction with the byte deserializer block. The byte deserializer doubles the number of lanes for each receiver. If you use the single-width mode, the receiver output at the PLD interface 8-bits or 10-bits (single lane) if the byte deserializer is not used, and 16-bit or 20-bit (dual lanes) if it is used. If you use double-width mode, the receiver's output at the PLD interface is 16-bits or 20-bits (dual lanes) or 36-bits or 40-bits (transceiver block lanes) if the byte deserializer is used. The nature of the byte deserializer block does not lend itself to preserving the lane striping of the source transmitter. The least significant byte of the transmitter may be received in a different location. Refer to "Byte Deserializer" on page 2–112 for more details on how the bytes can be re-ordered. The byte ordering block ensures that the correct lane striping is kept at the receiver output.

You cannot use the byte ordering block in conjunction with the rate matcher because it disturbs the byte ordering by adding or deleting bytes because of the data and clock PPM offset.

Figure 2–92. Byte Ordering Block



Word Alignment Based on Byte Ordering

In word alignment based on byte ordering, the byte ordering block performs lane alignment after the word aligner achieves byte alignment. The byte ordering block is triggered by the rising edge of the `rx_syncstatus` signal. To achieve lane alignment, the byte reordering block monitors the data stream for the alignment patterns. When the byte reordering block finds the correct alignment pattern, it inserts the programmable pad byte in the data stream until the alignment pattern can be placed in the LSByte position (lane 0). When the alignment pattern is placed in the LSByte position, the byte ordering process is complete and the status signal `rx_byteorderalignstatus` asserts (stays high). If the alignment pattern is already in the LSByte position, the byte ordering block detects this, considers the byte ordering process complete, and asserts the `rx_byteorderalignstatus` signal.Byte ordering is not performed again, even if the alignment byte exists in the data stream, until the channel is reset by the `rx_digitalreset` port (`rx_analogreset` and `gxb_powerdown` also reset the receiver channel).

Figure 2–93 shows how the byte ordering block works in a double-width mode four-lane configuration (four-byte-wide interface). The alignment character, denoted by the "A" character, goes into the byte ordering block in lane two. The byte ordering block inserts two pad bytes, denoted by PD, delaying the alignment byte until it appears in the LSByte position (lane 0).

Figure 2–93. Double-Width Byte Ordering With Two Pad Byte Inserts

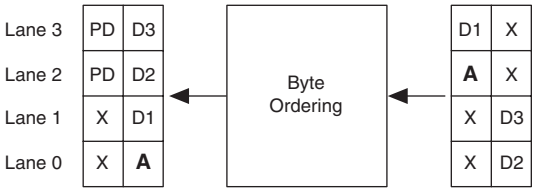
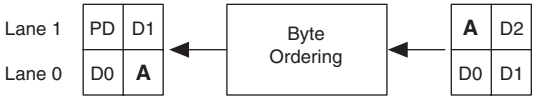


Figure 2–94 shows the byte ordering block in single-width mode two-lane configuration. In Figure 2–94, the alignment pattern "A" is in the MSByte position (lane 1). The byte ordering block inserts a pad character to force the alignment character to the LSByte position (lane 0). If the alignment pattern already exists in lane 0, the byte ordering process completes without any ordering done because it is unnecessary. After the ordering process is complete, the `rx_byteorderalignstatus` signal asserts and stays high until `rx_digitalreset`, `rx_analogreset`, or `gxb_powerdown` is asserted to reset the byte ordering block.

Figure 2–94. Single-Width Byte Ordering With One Pad Insert



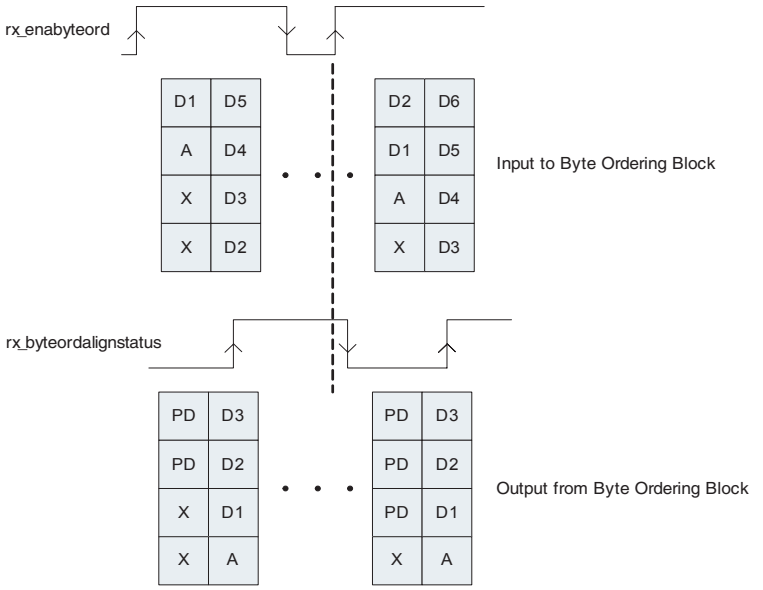
PLD-Controlled Byte Ordering

Unlike word alignment based byte ordering, PLD-controlled byte ordering provides control to the user logic to restore correct byte order at the receiver. When enabled, an `rx_enabyteord` port is available at the PLD interface. A rising edge on the `rx_enabyteord` port triggers the byte ordering block. The byte ordering block looks for the user-programmed byte ordering pattern in the data stream from the byte deserializer. When the byte reordering block finds the byte ordering pattern, it inserts the user-programmed pad byte in the data stream until the byte ordering pattern can be placed in the LSByte position. When the byte ordering pattern is placed in the LSByte position, the byte ordering process is complete and the status signal `rx_byteorderalignstatus` is asserted (stays high). If the alignment pattern is already in the LSByte position, the byte ordering block does not add any pad byte, considers the byte ordering process complete, and asserts the `rx_byteorderalignstatus` signal.

Unlike word alignment based byte ordering, PLD-controlled byte ordering does not require resetting the channel to re-trigger the byte ordering process. A rising edge on the rx_enabyteord signal re-triggers the process by de-asserting the rx_byteorderalignstatus signal. The byte ordering block starts looking for the byte ordering pattern again and adds pad bytes as necessary to achieve byte ordering. Once it completes the byte ordering process, it asserts the rx_byteorderalignstatus signal.

Figure 2–95 shows PLD-controlled byte ordering in Basic double-width Mode.

Figure 2–95. User-Controlled Byte Ordering in Double-Width Mode



After the first rising edge of the rx_enabyteord signal in Figure 2–95, the byte ordering block finds the byte ordering pattern A in the second most significant byte. It adds two pad bytes PD to push the byte ordering pattern to the least significant byte position and asserts the rx_byteorderalignstatus signal. After the second rising edge of the rx_enabyteord signal, rx_byteorderalignstatus is de-asserted and the byte ordering block starts looking for byte ordering pattern A. It finds the byte ordering pattern A in the second least significant byte position and adds three pad bytes PD. The byte ordering pattern A now appears at the least significant byte position and rx_byteordalignstatus is asserted.

Two critical aspects related to PLD-controlled byte ordering process are:

- What to choose as the byte ordering pattern
- When to assert the rx_enabyteord signal

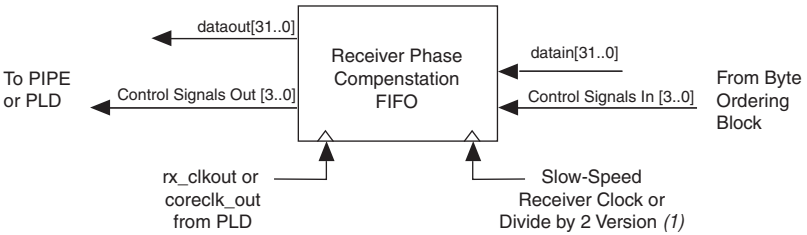
In Stratix II GX configurations, PLD-controlled byte ordering is available only in SONET/SDH OC-48 mode or Basic double-width mode. In SONET/SDH OC-48 mode, byte A2 of the A1A2 word alignment pattern is automatically selected as the byte ordering pattern. In Basic Double-Width mode, you programs the byte ordering pattern while configuring the transceiver using the MegaWizard Plug-In Manager. Since the byte ordering block is designed to place the byte ordering pattern at the LSByte position, you must select a pattern that appears at the LSByte position at the source. This ensures that when the byte ordering block pushes the byte ordering pattern byte to the LSByte position at the receiver, the data is correctly byte ordered. Ideally, if this pattern is unique and is guaranteed to appear only at the LSByte position at the source, the instance at which the rx_enabyteord signal is asserted becomes irrelevant. For example, in packet-based 8B/10B encoded links, you could choose the Start of Packet (SOP) byte as the byte ordering pattern if it is a unique control code (say K28.0). In non 8B/10B scrambled data links, it may be difficult to find a unique pattern since there is a possibility of the pattern appearing in the scrambled payload and causing the byte ordering block to add pad bytes incorrectly. In such cases, the instance at which the rx_enabyteord signal is asserted becomes critical.

The rx_enabyteord signal must be asserted after the word aligner has aligned to the correct word boundary. This ensures that the byte ordering block does not find a byte ordering pattern between the word boundaries. If the rx_enabyteord signal is asserted before the intended byte ordering byte appears at the receiver, then the byte ordering block will add necessary pad bytes to achieve correct byte ordering. If the rx_enabyteord signal is asserted before the unintended data byte that matches the byte ordering pattern, then the byte ordering block may incorrectly add pad bytes and assert the rx_byteorderalignstatus signal. In the SONET/SDH OC-48 configuration, since the receiver anticipates the byte ordering pattern A2 every 125 µs, the rx_enabyteord signal assertion can be easily timed to avoid incorrect byte ordering. In Basic Double-Width mode, it is up to you to either select a unique byte ordering pattern or an appropriate instance to assert rx_enabyteord, depending on the dynamics of the implemented protocol.

Receiver Phase Compensation FIFO Buffer

The receiver phase compensation FIFO buffer (Figure 2–96) is located at the FPGA logic array interface in the receiver block and is used to compensate for phase difference between the receiver clock and the clock from the PLD. The receiver phase compensation FIFO buffer operates in two modes: low latency and high latency. In low latency mode, the FIFO buffer is four words deep. The Quartus II software chooses the low latency mode automatically for every mode except the PCI-Express PIPE mode (which automatically uses high latency mode). In high latency mode, the FIFO buffer is eight words deep.

Figure 2–96. Receiver Phase Compensation FIFO Buffer



Note to Figure 2–96:

(1) The receiver clock can either be the recovered clock or the transmitter CMU clock, depending on whether the rate matcher is used or not.

In Basic mode, the write port is clocked by the recovered clock from the CRU. This clock is half the rate if the byte deserializer is used. The read clock is clocked by the associated channel's recovered clock.

The receiver phase compensation FIFO is always used and cannot be bypassed.

In four-channel (×4) and eight-channel (×8) bonding modes, all the read pointers are derived from a common source so that there is no need to synchronize the data of each channel in the PLD logic.

Receiver Phase Compensation FIFO Error Flag

Depending on the transceiver configuration, the write port of the receiver phase compensation FIFO can be clocked by either the recovered clock (`rx_clkout`) or transmitter PLL output clock (`tx_clkout` or `coreclkout`). The read port can be clocked by the recovered clock (`rx_clkout`), transmitter PLL output clock (`tx_clkout` or

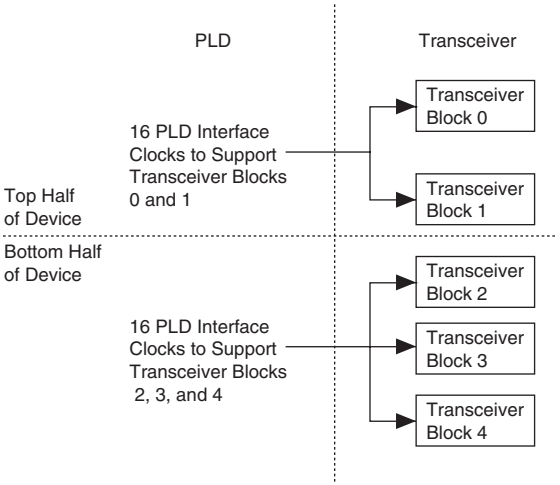
coreclkout) or a PLD clock. In all configurations, the write clock and the read clock must have 0 PPM difference to avoid overrun/underflow of the phase compensation FIFO.

An optional debug_rx_phase_comp_fifo_error port is available in all modes to indicate receiver phase compensation FIFO overrun/underflow condition. debug_rx_phase_comp_fifo_error is asserted high when the phase compensation FIFO gets either full or empty. This feature is useful to verify the phase compensation FIFO overrun/underflow condition as a probable cause of link errors.

PLD-Transceiver Interface Clocking

There are 32 PLD interface clocks available between the PLD logic and the transceiver blocks. The 32 PLD interface clocks are divided equally between the top and bottom half of the device (16 PLD interface clocks for the top half of the device and 16 PLD interface clocks for the lower half). The PLD interface clocks are used as the receiver and transmitter phase compensation FIFO clocks. Figure 2–97 shows the PLD clock interface.

Figure 2–97. PLD Interface Clock (2SGX130G)



The following clock inputs utilize the PLD interface clocks:

- rx_cruclk (if driven from the PLD clock tree)
- Pll_inclk (if driven from the PLD clock tree)
- tx_coreclk
- rx_coreclk
- Cal_blk_clk

The rx_cruclk and the pll_inclk are reference clocks to the transceiver receiver PLL and transmitter PLL. These two ports can take the reference clock from the dedicated REFCLK pins or from the PLD global clock pins. If the PLD global clock pins are used to feed the transceiver PLLs, a PLD interface clock will be used for each independent reference clock feeding the transceiver.

Each transceiver block has one possible PLD connection to pll_inclk and four possible connections to rx_cruclk. Only one reference clock frequency can be fed from the PLD to each transceiver block. The receiver PLLs of each channel can possibly have a different reference clock frequency as long as there are PLD interface clocks available.

The tx_coreclk and rx_coreclk are input clocks to the transmitter and receiver phase compensation FIFOs, respectively. By default, the Quartus II software automatically routes the tx_clkout or coreclk_out to the tx_coreclk, and the rx_clkout, tx_clkout, or coreclk_out to the rx_coreclk port, depending on the transceiver block and channel configuration as listed in the above section.

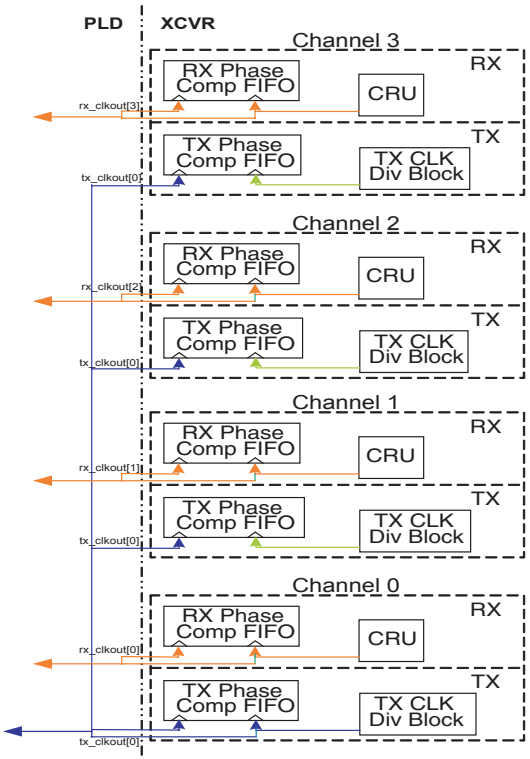
There are options to route other PLD clocks to the tx_coreclk and rx_coreclk ports. The non-transceiver clocks that feed these ports are required to be frequency locked (0 ppm) to the transceiver output clocks of the associated channel or transceiver block, depending on the configuration. The method of using this option is discussed in a later section.

The cal_blk_clk feeds the calibration block.

If a single clock from the PLD feeds multiple ports listed above, then only one PLD interface clock will be used. It is recommended that whenever possible, utilize a common clock. This will save PLD clock resources and PLD interface resources.

Each transceiver block (with all channels running in the same configuration), by default, uses a minimum of five PLD interface clocks as seen in Figure 2–98. This is with each rx_coreclk clocked by the associated rx_clkout of each RX channel, and since all the TX channels are the same, the Quartus II software will automatically route tx_clkout[0] to all the tx_coreclk inputs. The reference clock can use the dedicated REFCLK pins to save on PLD interface clocks. The Quartus II software does not cross the transceiver block boundary when combining like TX channels. Also, the Quartus II software does not combine RX clocks automatically.

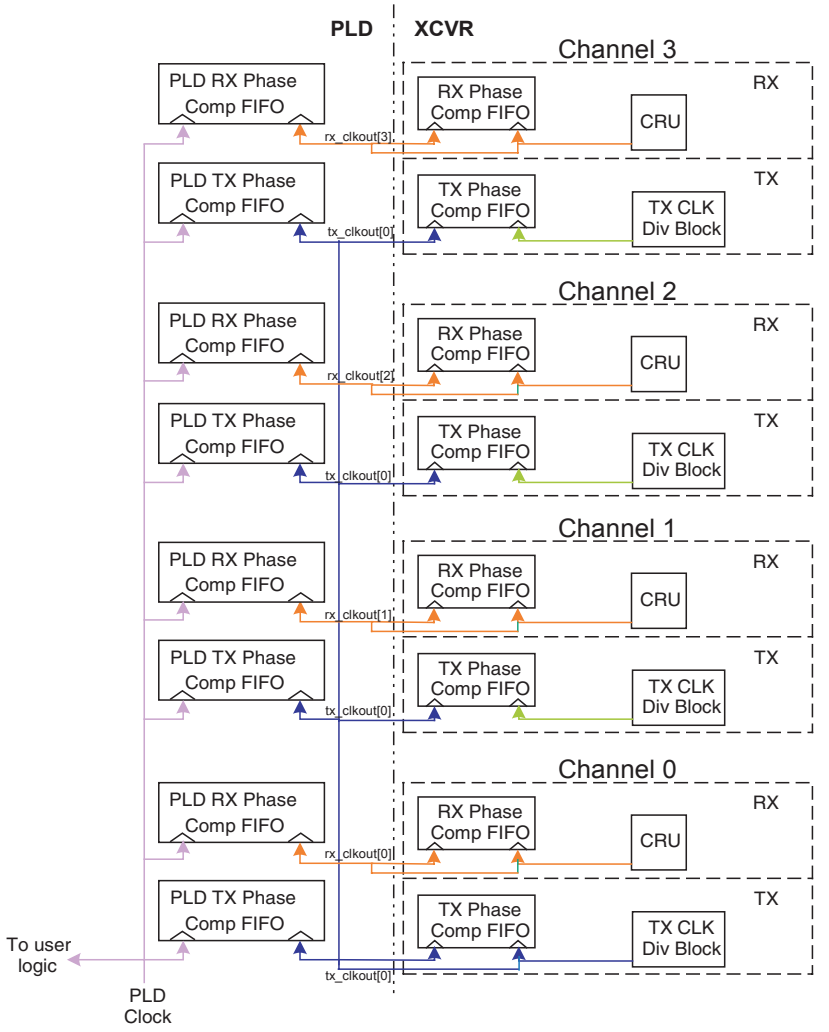
Figure 2–98. Minimum PLD Interface Clock Utilization



Note that in this configuration, the user logic needs to be clocked by the `tx_clkout[0]` for the TX path and the individual RX channel logic needs to be clocked by its associated `rx_clkout`.

If you use the default configuration, and the user logic is clocked by another clock than the transceiver clock associated with that channel, a PLD phase compensation must decouple the phase difference. Figure 2–99 shows of TX and RX PLD phase compensation FIFOs decoupling the user logic from the transceiver.

Figure 2–99. Phase Compensation FIFO Implementation in PLD Logic



If more TX channels are used across transceiver blocks, and/or RX channels are also configured in a similar fashion and have the same data rate and PLD output clock frequency, you will need to manually connect the `tx_coreclk` and `rx_coreclk` ports.

The PLD interface clock utilization can be further reduced by directly feeding the tx_coreclk and/or the rx_coreclk ports of like channels directly with a single clock. The source clock must be frequency locked to the associated transceiver output clock. Any PPM difference results in data corruption.

To help guard against incorrect usage, the use of the tx_coreclk and rx_coreclk options requires clock assignments in the assignment organizer. If no assignments are used, the Quartus II software will give a compilation error.

There are 4 settings to enable the PLD interface clocking options:

- **Stratix II GX GXB Shared Clock Group Setting**
- **Stratix II GX GXB Shared Clock Group Driver Setting**
- **Stratix II GX 0PPM Clock Group Setting**
- **Stratix II GX 0PPM Clock Group Driver Setting**

As the name indicates, there are two main settings, each with a driver and clock group setting. When specifying clock groups, an integer identifier is used as the group name in order to differentiate other clock group settings from one another.

The **Stratix II GX GXB Shared Clock Group Setting** is the safest assignment. The Quartus II compiler will analyze the netlist during compilation to ensure TX channel members are derived from the same source. The Quartus II software will give a fitting error for incompatible assignments. The software cannot check for the output of the RX frequency locked to the driving clock as the exact frequency is dictated by the upstream transmitter's source clock. It will be up to you to ensure that the rx_coreclk is derived from the same source clock as the upstream transmitter.

The **Stratix II GX GXB Shared Clock Group Driver Setting** assignment must be made to the source channel of the tx_clkout or coreclk_out. Specifying anything except the TX channels (the source for the tx_clkout or coreclk_out) will result in a fitter error. If the source clock is not from tx_clkout or coreclk_out (for example, the source is from rx_clkout or from a PLD clock input), the 0 PPM setting must be used instead.

For example, in a synchronous system, the TX and RX are of the same data rate and configuration. The clock output of the channel 0 is used (but any TX clock output can be used). The **Stratix II GX GXB Shared Clock Group Driver Setting** is made in the assignment editor on the tx_dataout[0] name. You can use a group identifier value of "1" to identify the group that this driver feeds. The **Stratix II GX GXB shared**

Clock Group Setting is made to all the `rx_datain` channels that the `tx_dataout[0]` output clock drives. (note that the other `tx_dataout` channels do not need an assignment as the Quartus II software automatically groups the like transmitters in a transceiver block). A group identifier value of "1" is also made to the `rx_datain` assignments.

A breakdown of the assignment in the assignment editor is shown in Table 2–23:

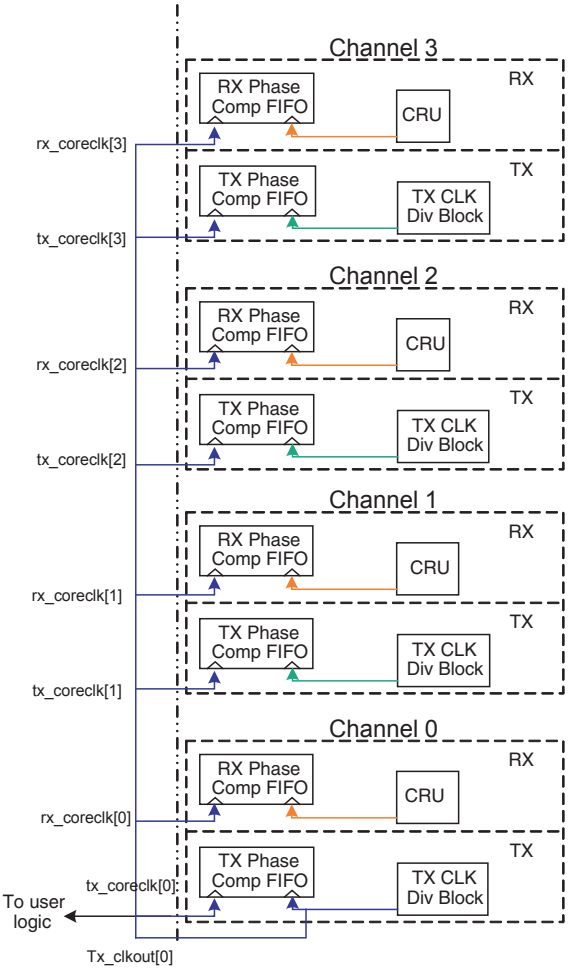
Table 2–23. Assignment Editor

To:	`tx_dataout[0]`
Assignment name:	Stratix II GX GXB Shared Clock Group Driver Setting
Value:	1
To:	`rx_datain[]` (note that the [] signifies the entire `rx_datain` group)
Assignment name:	Stratix II GX GXB Shared Clock Group Setting
Value:	1

Figure 2–100 shows the clocking configuration of the example.

For ×4 transceiver block configurations, the `coreclk_out` can replace the `tx_clkout[0]` but the driver assignment still remains `tx_dataout[0]`.

Figure 2–100. tx_clkout[0] Feeding All the tx_coreclk and rx_coreclk Posts of a Transceiver Block



The **Stratix II GX 0PPM Clock Group Setting** is for more advanced users that know the clocking configuration of the entire system and wants to reduce the PLD global clock resource and PLD interface clock resource utilization.

The Quartus II compiler does not perform any checking on the clock source. It is up to you to ensure that there is no frequency difference from the associated transceiver clock of the group and the driving clock to the `tx_coreclk` and `rx_coreclk` ports.

The **Stratix II GX 0PPM Clock Group Driver Setting** can be made to any of the transceiver output clocks (tx_clkout, rx_clkout, and coreclk_out) as well as any PLD clock input pins, transceiver dedicated REFCLK pin, or PLD PLL output. User logic cannot be used as a driver. As with the shared clock group setting, the driver setting for the transceiver output clocks is made to the associated channel. For example, for tx_clkout or coreclk_out, the transmitter channel name is specified. For the rx_clkout being the driver, the receiver channel name of the associated rx_clkout is specified. For the PLD input clock pins and the transceiver REFCLK pins, the name of the clock pin can be specified. For the PLL output, the PLL clock output port of the PLL can be found in the node finder and entered as the driver name. An integer value is specified for the group identification.

The **Stratix II GX 0 PPM Clock Group Setting** is made to the TX or RX channel names.

A breakdown of the assignment in the assignment editor is shown in Table 2–24:

Table 2–24. Assignment Editor

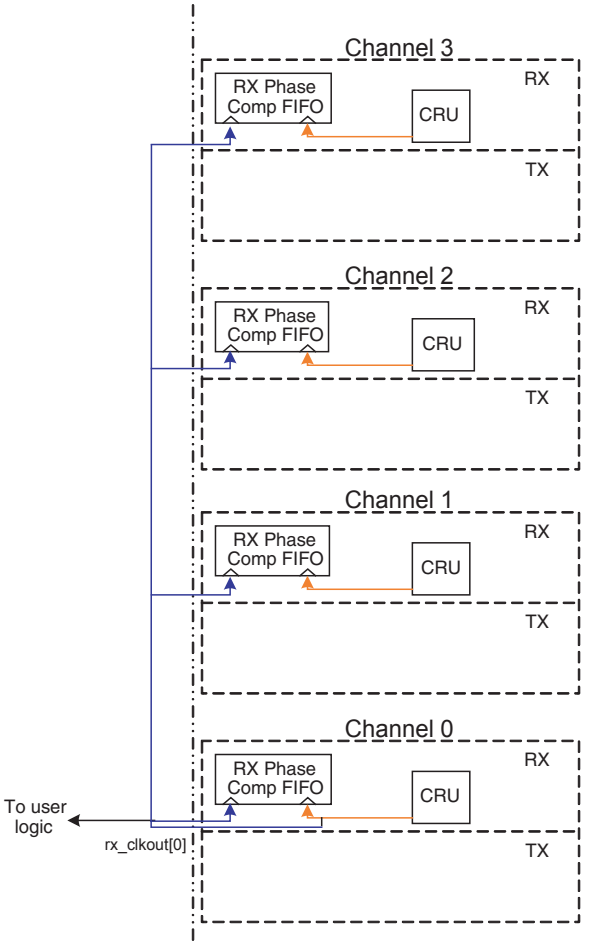
To:	tx_dataout, rx_datain, pld_clk_pin_name, refclk_pin, and pll_outclk
Assignment name:	Stratix II GX GXB 0 PPM Clock Group Driver Setting
Value:	1
To:	rx_datain[] and tx_dataout[]
Assignment name:	Stratix II GX GXB 0 PPM Clock Group Setting
Value:	1

The following are examples of clocking configurations that can use the 0PPM assignment:

All RX channels are configured the same and one recovered clock is feeding the rx_coreclk as shown in Figure 2–101. Though this example shows all the RX channels reside in a transceiver block, rx_clkout[0] of this transceiver block can feed other RX channels of other transceiver blocks. Note that this example is not showing any TX channels. If the rx_clkout is used as a driver, it can only feed RX channels. If these channels are used in a duplex mode, the tx_clkout from the TX channels should be used as the driver and feed the TX as well as the RX phase compensation FIFOs. The rx_clkout cannot feed the tx_coreclk ports.

It is important to note that asserting the `rx_analogreset` of the RX channel associated with the driver clock will flatline the clock. All logic and RX Phase Compensation FIFOs read port that it feeds will not be receiving a clock during analog reset of the driving channel. If the reset state machine is clocked by the driver clock, the reset state machine will hang and may not come out of reset. All the RX channels will need to go through a digital reset in order to restore the phase compensation FIFO pointers.

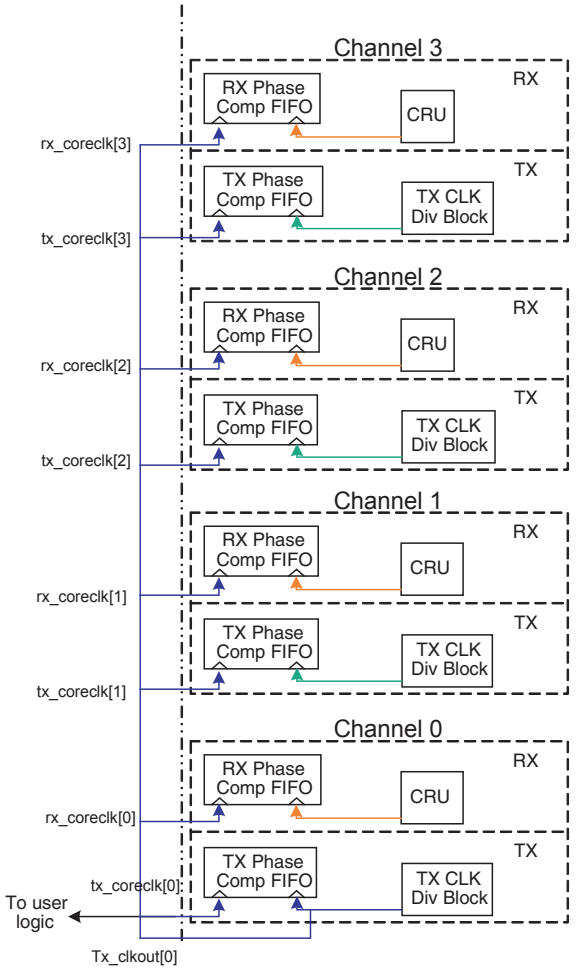
Figure 2–101. rx_clkout[0] Feeding All Receivers of the Same Transceiver Block



The next example is similar to the shared clock example of a tx_clkout feeding all the RX and TX channels, except that with the 0PPM setting, the tx_clkout can drive across transceiver blocks, as shown in Figure 2–102. The upstream device feeding the RX channels must be frequency locked to the tx_clkout used.

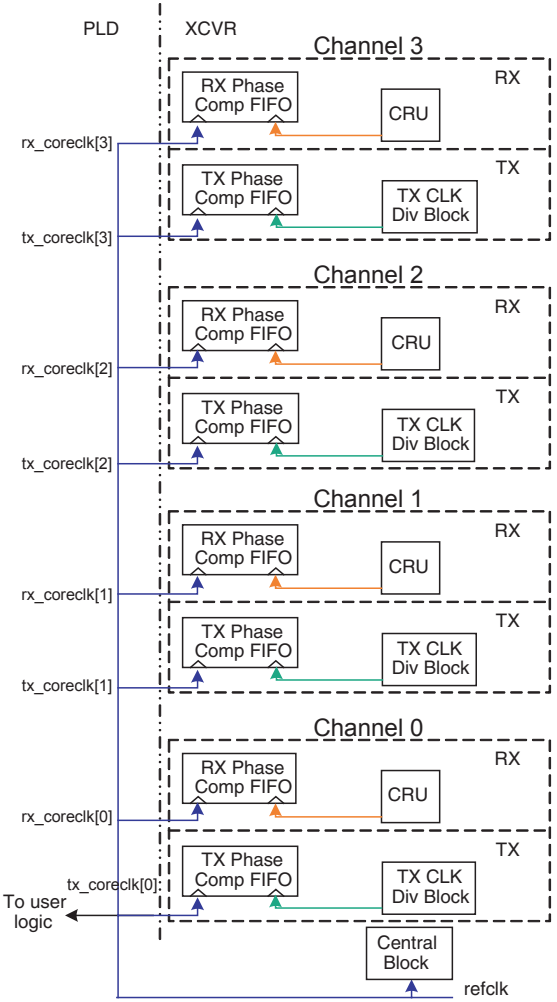
As with the RX channel example above, it is important to note that powering down the transceiver block where the driving channel resides will flatline the tx_clkout. All logic and the write ports of all the TX phase compensation FIFO will the driving clock feeds will flatline. A digital reset must be done on all channels after a driving transceiver block power down event.

Figure 2–102. tx_clockout[0] Feeding TX and RX Phase Compensation FIFOs Across Multiple Transceiver Blocks



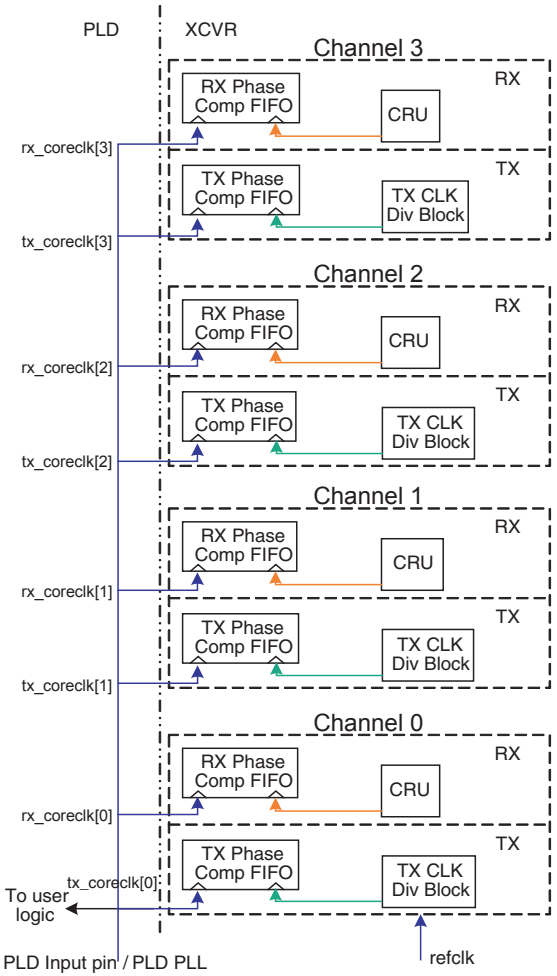
The next example features the dedicated REFCLK feeding the tx_coreclk and rx_coreclk ports as well as supplying the transceiver with the reference clock (Figure 2–103). This requires that the frequency of the reference clock at the REFCLK pin be of the same frequency as the transceiver output clocks of the associated channels. Any frequency difference yields corruption of data.

Figure 2–103. Dedicated REFCLK Feeding PCFIFO Clock Ports



Another example is where the PLD input clock or PLL output is feeding the `tx_coreclk` and `rx_coreclk` ports. Note that the driver clock must be the same frequency as the transceiver output clocks. Also, though this example shows the channels within a single transceiver block, the 0PPM setting will also allow TX and/or RX channel PCFIFOs of multiple transceiver blocks to be clocked by a common clock.

Figure 2–104. PLD-Driven Clock Feeding the Transceiver Phase Compensation FIFOs



PLD Interface Clock Resources

For the regional or global clock network to route into the transceiver, a local route input output (LRIO) channel is required. Each LRIO clock region has up to eight clock paths and each transceiver block has maximum of eight clock paths for connecting with LRIO clocks. These resources are limited and determine the number of clocks that can be used between the PLD and transceiver blocks. Tables 2–25 through 2–28 give the number of LRIO resources available for Stratix II GX devices with different numbers of transceiver blocks.

Table 2–25. Available Clocking Connections for Transceivers in 2SGX30D

	Clock Resource		Transceiver	
	Global Clock	Regional Clock	Bank13 8 Clock I/O	Bank14 8 Clock I/O
Region0 8 LRIO Clock	✓	RCLK 20-27	✓	
Region1 8 LRIO Clock	✓	RCLK 12-1		✓

Table 2–26. Available Clocking Connections for Transceivers in 2SGX60E

	Clock Resource		Transceiver		
	Global Clock	Regional Clock	Bank13 8 Clock I/O	Bank14 8 Clock I/O	Bank15 8 Clock I/O
Region0 8 LRIO Clock	✓	RCLK 20-27	✓		
Region1 8 LRIO Clock	✓	RCLK 20-27	✓	✓	
Region2 8 LRIO Clock	✓	RCLK 12-19		✓	✓
Region3 8 LRIO Clock	✓	RCLK 12-19			✓

Table 2–27. Available Clocking Connections for Transceivers in 2SGX90F (Part 1 of 2)

	Clock Resource		Transceiver			
	Global Clock	Regional Clock	Bank13 8 Clock I/O	Bank14 8 Clock I/O	Bank15 8 Clock I/O	Bank16 8 Clock I/O
Region0 8 LRIO Clock	✓	RCLK 20-27	✓			
Region1 8 LRIO Clock	✓	RCLK 20-27		✓		

Table 2–27. Available Clocking Connections for Transceivers in 2SGX90F (Part 2 of 2)

	Clock Resource		Transceiver			
	Global Clock	Regional Clock	Bank13 8 Clock I/O	Bank14 8 Clock I/O	Bank15 8 Clock I/O	Bank16 8 Clock I/O
Region2 8 LRIO Clock	✓	RCLK 12-19			✓	
Region3 8 LRIO Clock	✓	RCLK 12-19				✓

Table 2–28. Available Clocking Connections for Transceivers in 2SGX130G

	Clock Resource		Transceiver				
	Global Clock	Regional Clock	Bank13 8 Clock I/O	Bank14 8 Clock I/O	Bank15 8 Clock I/O	Bank16 8 Clock I/O	Bank17 8 Clock I/O
Region0 8 LRIO Clock	✓	RCLK 20-27	✓				
Region1 8 LRIO Clock	✓	RCLK 20-27					
Region2 8 LRIO Clock	✓	RCLK 12-19			✓	✓	
Region3 8 LRIO Clock	✓	RCLK 12-19				✓	✓

Clock Signal
provides information about clock signal on the transceiver block.

Table 2–29. Clock Port List (Part 1 of 2)

Clock Name	Maximum Number per Transceiver Block	Notes
pll_inclk	2	*(1)*
rx_cruclk	4	rx_cruclk can share resources with pll_inclk
tx_clkout / tx_coreclk	4	
rx_clkout / rx_coreclk	4	

Table 2–29. Clock Port List (Part 2 of 2)

Clock Name	Maximum Number per Transceiver Block	Notes
reconfig_clock	1	Used only for dynamic reconfiguration.
cal_blk_clk	1	One for all transceiver blocks.
fixedclk	1	PCI-Express (PIPE) mode only.
coreclkout	1	Used only for four-lane configurations.

Note to Table 2–29:

(1) Altera recommends using the REFCLK pin for pll_inclk. The REFCLK pin uses inter-transceiver line; therefore, there is no need to use the LRIO clock resource. This usage helps with performance and resource.

Example of clock usage

In a five transceiver block device or using channel reconfiguration, it is possible to exceed the number of LRIO clocks available.

Table 2–30 shows an example of LRIO clock resource usage in a EP2SGX130G device. In this case, the Quartus II software does not give errors for transceiver configurations.

Table 2–30. Example of LRIO Clock Resource Usage (Part 1 of 2)

	Clock Name	Number of Signals in the Transceiver Block	LRIO Resource Usage	Routing
Bank13 XAUI	pll_inclk	1	0	Using one REFCLK pin.
	rx_cruclk	4	0	Connect to pll_inclk.
	coreclkout	1	1	To PLD fabric.
Region0 LRIO Clock			1/8	
Bank14 GIGE 4ch	pll_inclk	1	0	Using one REFCLK pin.
	rx_cruclk	4	0	Connect to pll_inclk.
	tx_clkout	4	1	To PLD fabric.
Region1 LRIO Clock			1/8	

Table 2–30. Example of LRIO Clock Resource Usage (Part 2 of 2)

	Clock Name	Number of Signals in the Transceiver Block	LRIO Resource Usage	Routing
Bank15 Basic 4ch	pll_inclk	1	0	Using one REFCLK pin.
	rx_cruclk	4	0	Connect to pll_inclk.
	rx_clkout	4	4	To PLD fabric.
	tx_clkout	4	1	To PLD fabric.
	cal_blk_clk	1	1	From PLD fabric.
Bank16 Basic 4ch	pll_inclk	1	0	Using one REFCLK pin.
	rx_cruclk	4	0	Connect to pll_inclk.
	rx_clkout	4	4	To PLD fabric.
	tx_clkout	4	1	To PLD fabric.
Bank17 PCIE ×4	pll_inclk	1	0	Using one REFCLK pin.
	rx_cruclk	4	0	Connect to pll_inclk.
	coreclkout	1	1	To PLD fabric.
	fixedclk	1	1	From PLD fabric.
Region2 LRIO Clock			8/8 (or 7/8, 6/8)	
Region3 LRIO Clock			5/8 (or 6/8, 7/8)	

Multiple Protocols and Data Rates in a Transceiver Block

Stratix II GX supports multiple protocols and/or data rates in a single transceiver block. This allows for better utilization of the channels and power savings. There can be up to four independent data rates supported and up to two separate frequencies for the TX channels and up to four separate frequencies for the RX channels within a single transceiver block.

On the TX side, the TX local dividers and the two TXPLLs in the central block can be used together or separately to achieve multiple data rates and/or protocols in a transceiver block. Refer to "Transmitter Local Clock Divider Block" on page 2–16 and "Central Clock Divider Block" on page 2–14 for more information regarding the TX local divider blocks and the dual TXPLL configuration.

On the RX side, it is possible to have up to four receiver channels to be of different data rates and configurations as long as there are enough PLD interface clocks to support the channels. Since each receiver channel contains a dedicated RXPLL, there are no data rate or configuration restrictions.

Quartus II software automatically combines multiple ALT2GXB Megafunction instances into a transceiver block if possible. If there is a particular placement required, Altera recommends that you force the placement via TX and/or RX channel pin assignment in the assignment editor. Quartus II software checks to see if the desired placement is possible. If not, the Quartus II software sends a fitter error.

Since the TX channel parameters are the deciding factors on whether the channels can be combined in a single transceiver block, the following sections will concentrate on the TX side.

Transceiver Block-Based Controls

First, in order to combine channels into a single transceiver block, the transceiver block-based control signals must be driven from the same source.

The following is a list of transceiver block-based control signals:

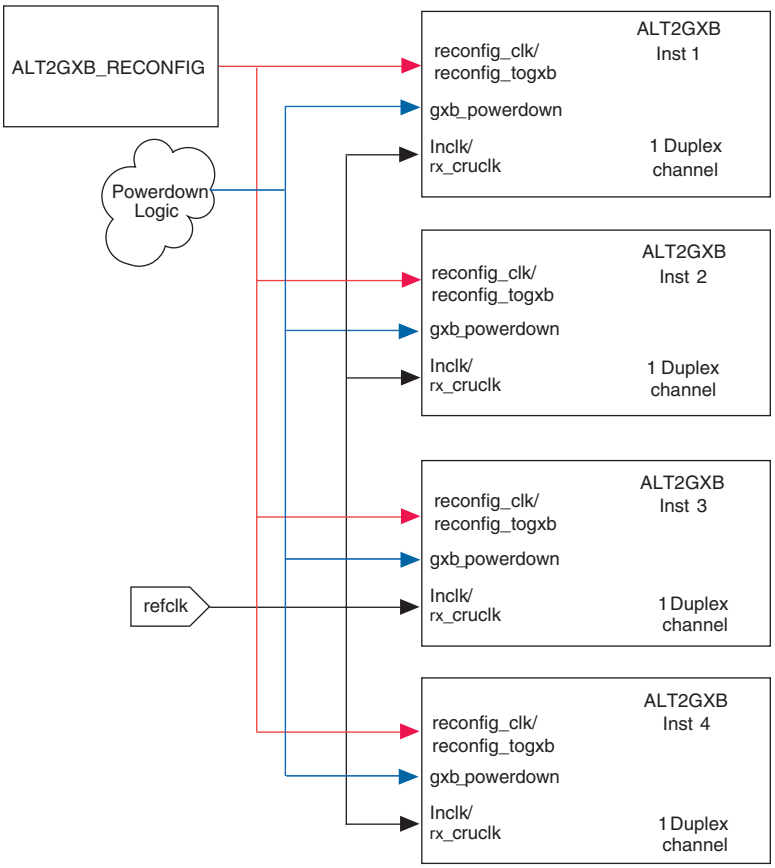
- `gxb_powerdown`
- `reconfig_clk`
- `reconfig_togxb`

The `gxb_powerdown` signal of all the instances that are to be combined in a single transceiver block must be connected to a single point; for example, the same input pin or same logic. Any driving logic differences will prevent the instance from being combined in a single transceiver block.

If you use dynamic reconfiguration, only one dynamic reconfiguration controller is allowed to drive the instances to be combined in a single transceiver block. If you use multiple dynamic reconfiguration controllers, the Quartus II software will not be able to combine the instances into a single transceiver block as the PLD logic cannot be combined. All the `reconfig_clk` and `reconfig_togxb` ports need to be tied to a single dynamic reconfiguration block.

Figure 2–105 is an example of correct transceiver block-control signal connection.

Figure 2–105. Transceiver Block-Control Signal Connections



TXPLL Sharing

Multiple channels in using the Basic protocol can share a common TXPLL. Their data path configuration can differ as long as they are within the same width mode; for example, if all operate in single-width mode or if all operate in double-width mode. Mixing single-width and double-width modes is not allowed for TXPLL sharing.

The data rate range of single-width versus double-width mode is listed in Table 2–31. If the desired data rate range falls within a mode using the TX local divider factors (/1, /2, or /4), you can use a single TXPLL to support the desired data rate range. If the desired data rate range straddles two modes, TXPLL sharing cannot be done. They can still be in the same transceiver block, but you will need to use two TXPLLs.

Table 2–31. Single-Width Versus Double-Width Data Rate Range

Mode	Minimum Data Rate	Maximum Data Rate
Single Width	600 Mbps	3.125 Gbps
Double Width	1 Gbps	6.375 Gbps

In order to share a TXPLL, their PLL configurations are required to be the same. The following is a list of TXPLL parameters that must be identical:

- PLL bandwidth
- Primary data rate
- Reference clock frequency
- Pre-divider on the dedicated REFCLK (if applicable)

Though the TXPLL settings must be identical, the TX local divider settings on the channels can vary. You will need to set up the TXPLL for the highest data rate and use the dividers to drop the data rate down on the slower channels to /4 or /2 of the primary data rate. Use the ALT2GXB MegaWizard to make your changes.

Non-Basic protocol modes cannot share a TXPLL unless they are all of the same protocol and sub protocol.

TXPLL sharing is restricted to the channels within a transceiver block.

Also, the analog buffer voltage setting VCCH must be the same across all the channels in the transceiver block. The Quartus II software will not allow channels with different VCCH settings into the same transceiver block.

The following is an example of instances that can be combined into a single transceiver block. We have two 4 Gbps, one 2 Gbps, and one 1 Gbps links. Since the target data rate is either a /1, /2, or /4 division factor from the highest data rate, it is possible to combine this into a single transceiver block. It is assumed that the VCCH, transceiver block signals

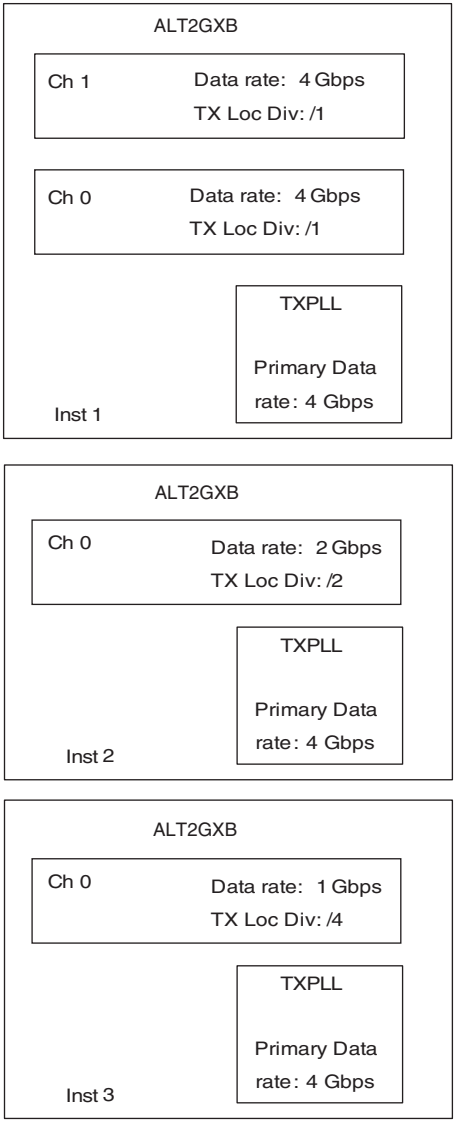
(for example, gxb_powerdown), Dynamic Reconfig ports, reference clock frequency, and PLL bandwidth of each instance are the same and/or are driven from the same point.

Each instance must be created with the same primary data rate set a 4 Gbps so that the TXPLLs can be combined. Use the local dividers to achieve the desired data rate for each channel. Because the goal is to have all four channels in a transceiver block driven off of a single TXPLL, the reference clock will need to be from the same point - either from a single dedicated REFCLK pin or from

Though the example shows that there are two 4 Gbps channels configured in a single ALT2GXB instance, duplicate channels may not need to be configured in the same instance. If the two 4 Gbps channels were to be configured in separate instances, the resultant transceiver block configuration will not have changed.

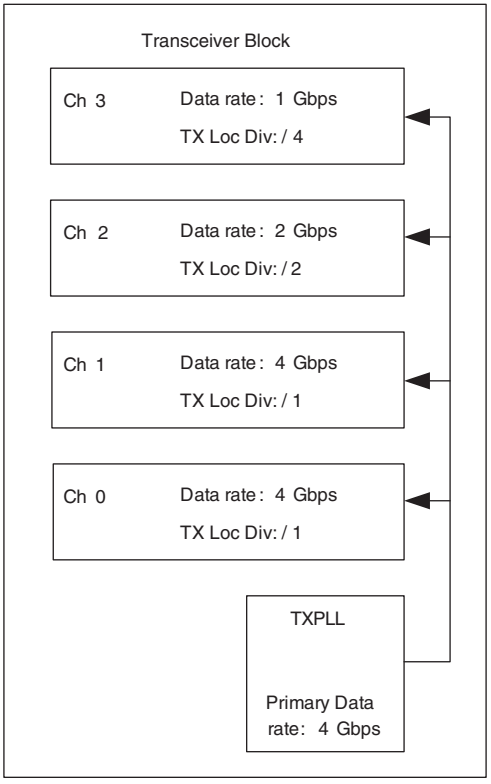
Figure 2–106 shows an example of TXPLL sharing compatible instances.

Figure 2–106. TXPLL Sharing Compatible Instances



The instances shown in Figure 2–106 net the transceiver block configuration shown in Figure 2–107 after compilation. The specific data rate on the channel location may differ depending on either the placement algorithm or your assignments. Since all the channels in this transceiver block are utilized, the second TXPLL is not used. However, you can use the second dedicated REFCLK input to feed the IQ lines or the PLD logic if you do not use the first dedicated REFCLK to drive the IQ lines or PLD logic.

Figure 2–107. Resultant Transceiver Block Configuration After Combining Instances



Using Two TXPLLs

If the desired data rates and/or protocols cannot utilize a single TXPLL within a single transceiver block, the other TXPLL within that transceiver block may be able to support the additional data rates and/or protocol configuration.

This is useful if combining channels operating in single-width and double-width modes, or two Quartus II software-defined protocols (for example, Basic, GIGE, SONET/SDH, SDI, or PCI Express [PIPE] modes) in the same transceiver block. You can configure channels in the same protocol group from individual ALT2GXB instances, or you can configure the whole group in a single instance.

The example in Figure 2–108 will not be able to share a single TXPLL due to incompatible primary data rates. Since there are only two different primary data rates, you can merge the four channels into a single transceiver block. It is assumed that the channels sharing the same TXPLL will meet the requirements described in the above example. For the channels not sharing the same TXPLL, the PLL parameters may be different; for example, different data rate, reference clock input frequency, reference clock input pin, and PLL bandwidth. Also, the primary data rates need not be a multiple of the other TXPLL primary data rate. The transceiver block-based signals will have to be the same across all channels.

In addition to the primary data rate differences, the mode of operations also differs which require the use of separate TXPLLs. The 4 Gbps channels operate in a double-width mode while the 2 Gbps and 1 Gbps channels operate in a single-width mode (due to the sub 3.125 Gbps primary data rate).

Figure 2–108. TXPLL Sharing Incompatible Instances

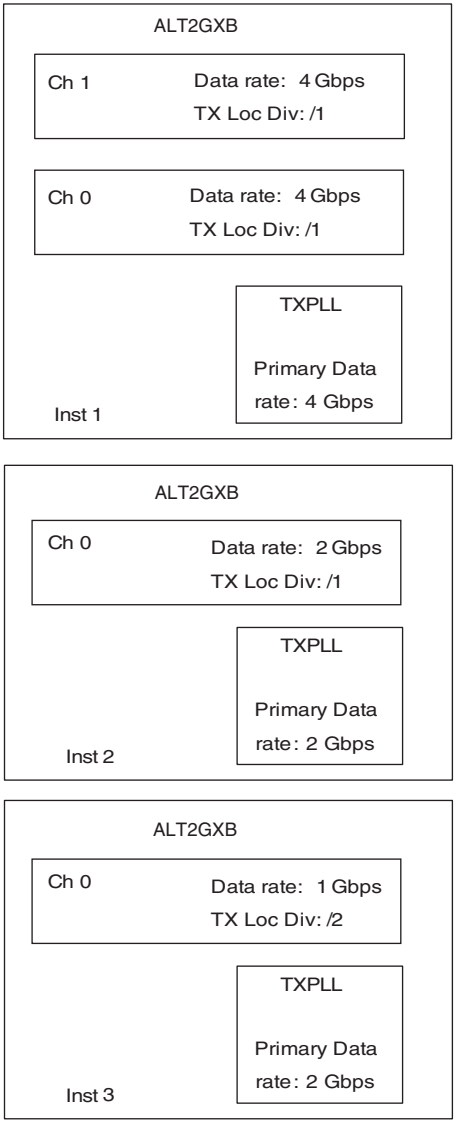
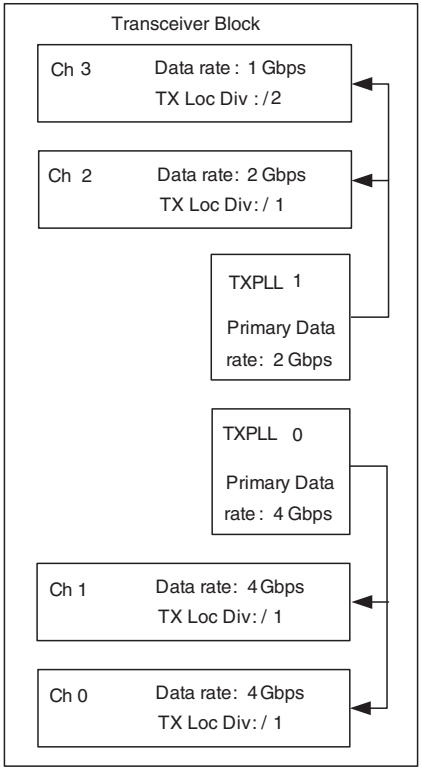


Figure 2–109 is the resultant transceiver block configuration after combining the above instances. Since the two TXPLLs primary data rates can be derived from the same reference clock frequency, one reference clock input is needed. If the reference clock frequency differs, or if the primary data rates differ (for example, 2.488 Gbps instead of 2 Gbps), two reference clocks will be needed.

If the reference clock is driven from the PLD, only one connection exists for each transceiver block. Both TXPLLs in a transceiver block cannot be driven from separate PLD clock pins. If the reference clock frequencies are the same, it is possible to drive both TXPLLs in a transceiver block from a single PLD clock pin (or the dedicated REFCLK pin).

Figure 2–109. Resultant Transceiver Block Configuration with Two TXPLLs



Combining RX Channels in a Transceiver Block

Receiver channels of different configuration and data rates can be combined in to a single transceiver block. Since each receiver channel contains a dedicated RXPLL, there are no data rate or configuration restrictions. However, there are some restrictions. One restriction is for the dedicate reference clock usage. All the RX channels referenced off of the same dedicated REFCLK pin must have the same input frequency and the same usage of the REFCLK pre-divider. The other restriction is that since this is a RX-only configuration, the use of the rate matcher block is not allowed. If rate matching is needed, you will need to implement the rate matching FIFO in the PLD core.

It is possible to have up to four receiver channels to be different data rates and configurations, as long as there are enough PLD interface clocks to support the channels. As with the TX channels, each receiver channel can be configured from separate ALT2GXB instances or the entire group (as long as all the RX channels are the same) can be configured in one instance.

Combining RX and TX channels in a Transceiver Block

You can combine duplex channels and/or a mixture of TX and RX channel instances into a single transceiver block. For combining the duplex channel configuration, the TX rules and restrictions is a superset of the RX side and is covered in "TXPLL Sharing" on page 2–137 and "Using Two TXPLLs" on page 2–142.

The rules and restrictions for a combination of separate RX and TX channel instances in the same transceiver block are the same as outlined in "TXPLL Sharing" on page 2–137 and "Using Two TXPLLs" on page 2–142 for the transmitter and "Combining RX Channels in a Transceiver Block" on page 2–145.

Native Modes

The Stratix II GX transceiver operates in one of nine native modes:

- Basic
  - Single-width mode (600 Mbps to 3.125 Gbps)
  - Double-width mode (1 Gbps to 6.375 Gbps)
- PCI Express (PIPE) mode (2.5 Gbps)
- XAUI (3.125 Gbps up to "HiGig" 3.75 Gbps)
- GIGE (1.25 Gbps)
- SONET/SDH mode (OC-12, OC-48, and OC-96)
- (OIF) CEI PHY Interface (>3.135 Gbps to 6.375 Gbps)
- Serial RapidIO (1.25 Gbps, 2.5 Gbps, 3.125 Gbps)
- SDI (HD, 3G)
- CPRI (614 Mbps, 1.228 Gbps, 2.456 Gbps)

Basic Single-Width Mode

Use the Basic single-width mode for custom protocols that are not part of the pre-defined supported protocols, for example PIPE. With some restrictions, the following PCS blocks are available:

- Transmitter phase compensation FIFO buffer
- Transmitter byte serializer
- 8B/10B encoder
- Word aligner
- Rate matcher
- 8B/10B decoder
- Byte deserializer
- Byte ordering block
- Receiver phase compensation FIFO buffer

The byte ordering block is available only in reverse serial loopback configuration in Basic mode. The rate matcher is coupled with the 8B/10B code groups, which requires the use of the 8B/10B encoder or decoder either in the PCS or PLD logic array.

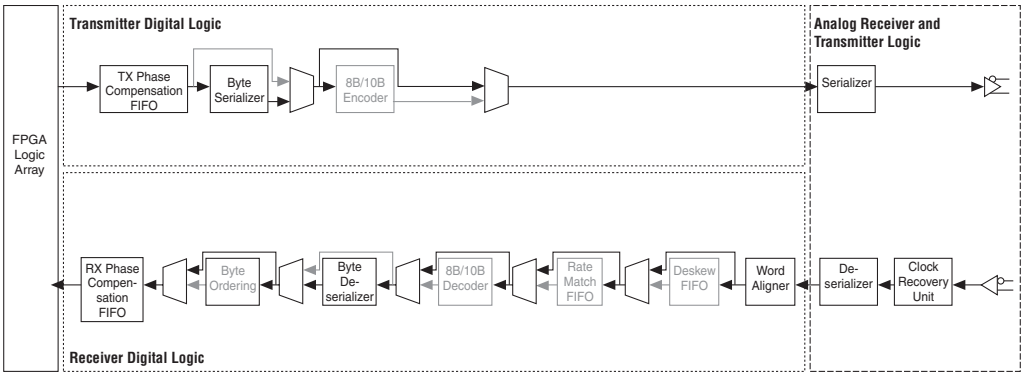
Basic Single-Width Mode with x4 Clocking

In basic single-width mode, the ALT2GXB MegaWizard provides a ×4 option under the **Which subprotocol will you be using?** option. If you select this option, all four transmitter channels within the transceiver block are clocked by clocks generated from the central clock divider block (refer to "Transmitter Clocking (Bonded Channels)" on page 2–29). The low-speed clock from the central clock divider block clocks the bonded transmitter PCS logic in all four channels. This reduces the transmitter channel-to-channel skew within the transceiver block. Each receiver channel within the transceiver block is clocked individually by the recovered clock from its own clock recovery unit (CRU).

Configuring transceivers in this mode yields low transmitter channel-to-channel skew within a transceiver block. It does not provide skew reduction for channels placed across transceiver blocks.

Figure 2–110 shows the data path in this mode.

Figure 2–110. Basic Single-Width Mode with ×4 Clocking



The transmitter data path consists of a 16-bit PLD-transceiver interface, transmitter phase compensation FIFO, 16:8-bit byte serializer, and 8:1 serializer.

The receiver data path consists of the clock recovery unit (CRU), 1:8 deserializer, bit-slip word aligner, 8:16 byte deserializer, receiver phase compensation FIFO, and 16-bit Transceiver-PLD interface.

Transceiver Placement Limitations

If one or more channels in a transceiver block are configured to Basic single-width mode with ×4 clocking option enabled, the remaining channels in that transceiver block must either have the same configuration or must be unused. All used channels within a transceiver block configured to this mode must also run at the same data rate. All channels within the transceiver block configured to this mode must be instantiated using the same ALT2GXB MegaWizard instance.

Figures 2–111 and 2–112 show examples of legal and illegal transceiver placements with respect to the Basic single-width mode with ×4 clocking enabled.

Figure 2–111. Examples of Legal Transceiver Placement

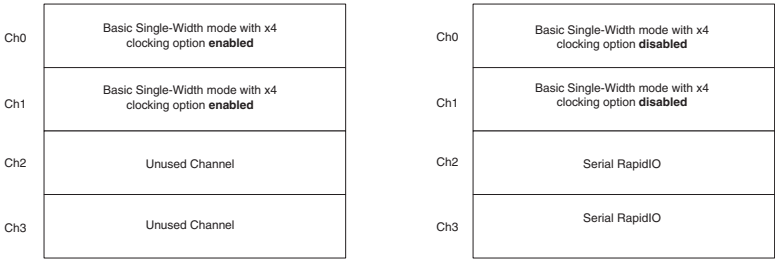
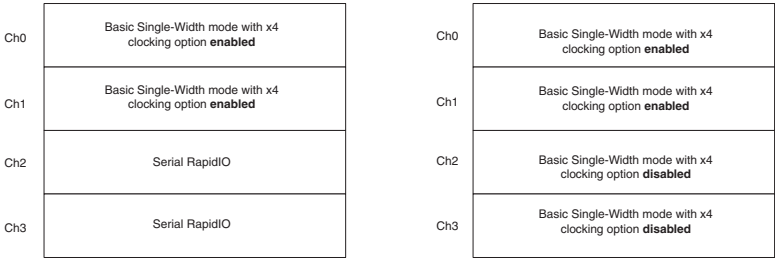


Figure 2–112. Examples of Illegal Transceiver Placement



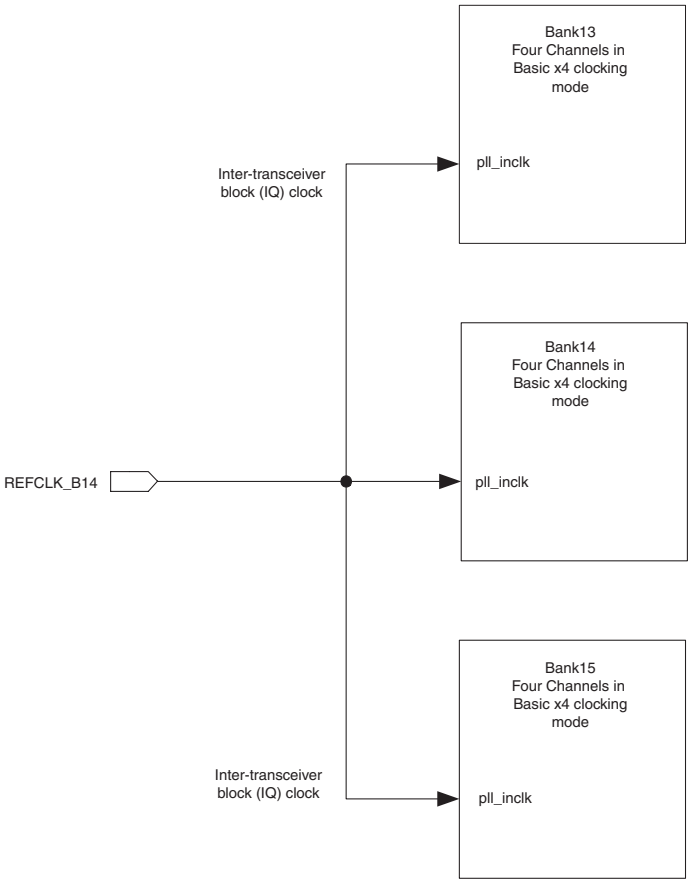
Clocking and Reset Recommendations

To minimize the transmitter channel to channel skew across transceiver blocks, Altera recommends:

- Using the dedicated `REFCLK` pins of the centrally located transceiver block in your design to provide the input reference clock for all transceiver blocks. This reduces the skew on the input reference clock driving the CMU PLL in each transceiver block. For example, in a design with 12 channels placed across Banks 13, 14, and 15, use the `REFCLK` pins of Bank 14 to provide the input reference clock. In a design with 16 channels placed across Banks 13, 14, 15, and 16, use the `REFCLK` pins of either Bank 14 or 15.
- De-asserting the `tx_digitalreset` signal of all used transceiver blocks simultaneously after `pll_locked` signal from all active transceiver blocks goes high.

Figure 2–113 shows the recommended clocking for 12 transceiver channels across transceiver banks 13, 14, and 15 in the EP2SGX90EF1152 device.

Figure 2–113. Clocking Recommendations to Minimize Transmitter Channel-To-Channel Skew



Basic Double-Width Mode

Use Basic double-width mode for custom protocols that are not part of the pre-defined supported protocols, for example, PIPE. With some restrictions, the following PCS blocks are available:

- Transmitter phase compensation FIFO buffer
- Transmitter byte serializer

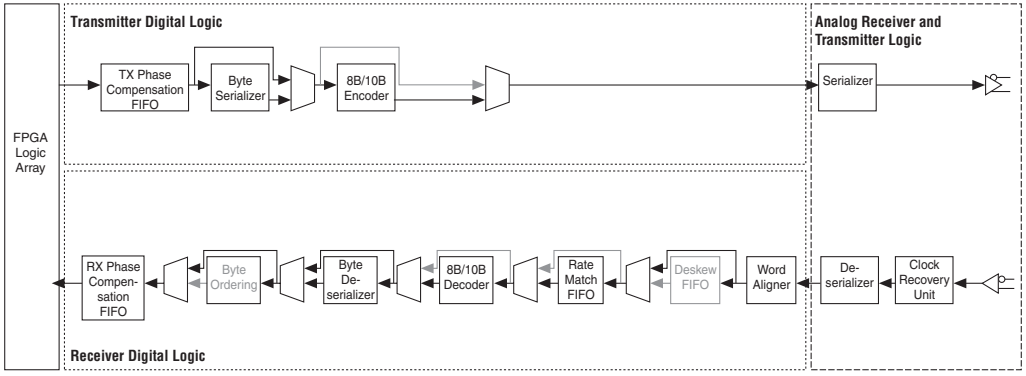
- 8B/10B encoder
- Word aligner
- Rate matcher
- 8B/10B decoder
- Byte deserializer
- Byte ordering block
- Receiver phase compensation FIFO buffer

The rate matcher is not available with the byte ordering block and vice versa. Because the rate matcher removes one byte at a time, the ordering of the blocks changes if rate matching occurs. The rate matcher is coupled with the 8B/10B code groups, which require the use of the 8B/10B encoder or decoder either in the PCS or PLD logic array.

PCI Express (PIPE) Mode

PCI Express is an evolution of Peripheral Component Interconnect (PCI). PCI is bandwidth-limited for today's applications because it relies on synchronous single-ended type signaling with a wide multi-drop data bus (refer to Figure 2–114). Clock and data-trace matching is required with PCI. PCI Express uses differential serial signaling with an embedded clock to enable an effective through-put of 2 Gbps per link to circumvent the limitations of PCI. PCI Express operates in ×1, ×4, ×8, ×16, and ×32 configurations and is also backward compatible with PCI on a software and driver level.

Figure 2–114. PIPE Mode



Stratix II GX devices support the PIPE standard in ×1, ×4, and ×8 configurations.

The Stratix II GX device has dedicated circuits to support the PCI Express protocol, including the following:

- 8B/10B encoder and decoder
- Rate matcher, which supports a multi-crystal environment up to ±300 PPM (600 PPM total) clock difference
- PIPE interface (Physical Interface for PCI Express)
- Receiver detection
- Beacon transmit capability
- Loopback
- Inversion
- Disparity control

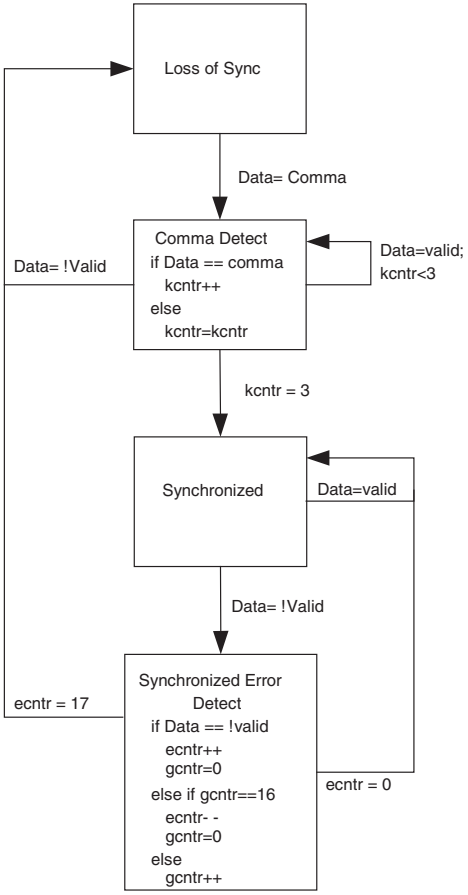
The PHY state machines, except for rate matching, are not included in the transceiver. Those state machines can be created in the PLD logic. This mode of operation is called the PIPE mode.The PIPE mode has a separate reset sequence. Refer to "Reset Sequence for PIPE Mode" on page 2–218 for more information.

The equalizer DC gain value in the MegaWizard Plug-In Manager for PIPE mode is set to a default value of 1. If the equalizer DC gain is controlled by the ALT2GXB_RECONFIG controller, the `rx_eqdcgain` input to the ALT2GXB_RECONFIG controller should be tied to "01" to be PCI E-compliant. Refer to the *ALT2GXB Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information.

Synchronization

In PIPE mode, the synchronization automatically occurs when the receiver receives 4 good /K28.5/ commas and 16 good code groups. Synchronization can be accomplished through the reception of four good PCI Express training sequences (TS1 or TS2) or four fast training sequences. Figure 2–115 shows a state diagram of the PCI-E synchronization.

Figure 2–115. PIPE Mode Synchronization State Machine



Tables 2–32 and 2–33 describe the TS1 and TS2 training sequences, respectively. A PCI Express fast training sequence consists of a /K28.5/, followed by three /K28.1/ code group.

If there is one code group error during the synchronization process, resynchronization must be performed.

Table 2–32. PCI Express TS1 Ordered Set

Symbol Number	Allowed Values	Encoded Values	Description
0		K28.5	Comma code group for symbol alignment
1	0–255	D0.0–D31.7, and K23.7	Link number with component
2	0–31	D0.0–D31.0, and K23.7	Lane number within port
3	0–255	D0.0–D31.7	N_FTS. The number of fast training ordered sets required by the receiver to obtain reliable bit and symbol lock.
4	2	D2.0	Data rate identifier Bit 0–Reserved, set to 0 Bit 1 = 1, generation 1 (2.5 Gbps) data rate supported Bit 2..7–Reserved, set to 0
5	Bit 0 = 0, 1 Bit 1 = 0, 1 Bit 2 = 0, 1 Bit 3 = 0, 1 Bit 4..7 = 0	D0.0, D1.0, D2.0, D4.0, and D8.0	Training control **Bit 0–Hot reset** Bit 0 = 0, deassert Bit 0 = 1, assert **Bit 1–Disable link** Bit 1 = 0, deassert Bit 1 = 1, assert **Bit 1–Loopback** Bit 2 = 0, deassert Bit 2 = 1, assert **Bit 3–Disable scrambling** Bit 3 = 0, deassert Bit 3 = 1, assert **Bit 4..7–Reserved** Bit 0 = 0, deassert Set to 0
6–15		D10.2	TS1 identifier

Table 2–33. PCI Express TS2 Ordered Set

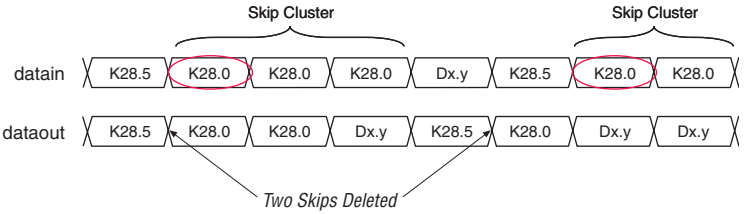
Symbol Number	Allowed Values	Encoded Values	Description
0		K28.5	Comma code group for symbol alignment.
1	0–255	D0.0–D31.7, and K23.7	Link number with component.
2	0–31	D0.0–D31.0, and K23.7	Lane number within port.
3	0–255	D0.0–D31.7	N_FTS. The number of fast training ordered sets required by the receiver to obtain reliable bit and symbol lock.
4	2	D2.0	Data rate identifier Bit 0–Reserved, set to 0 Bit 1 = 1, generation 1 (2.5 Gbps) data rate supported Bit 2..7–Reserved, set to 0
5	Bit 0 = 0, 1 Bit 1 = 0, 1 Bit 2 = 0, 1 Bit 3 = 0, 1 Bit 4..7 = 0	D0.0, D1.0, D2.0, D4.0, and D8.0	Training control **Bit 0–Hot reset** Bit 0 = 0, deassert Bit 0 = 1, assert **Bit 1–Disable link** Bit 1 = 0, deassert Bit 1 = 1, assert **Bit 1–Loopback** Bit 2 = 0, deassert Bit 2 = 1, assert **Bit 3–Disable scrambling** Bit 3 = 0, deassert Bit 3 = 1, assert **Bit 4..7–Reserved** Bit 0 = 0, deassert Set to 0
6–15		D5.2	TS2 identifier

Rate Matching

In PIPE mode, the rate matcher supports up to ±300 PPM (600 PPM total) differences between the upstream transmitter and the receiver. The rate matcher looks for skip ordered sets, which usually contain a /K28.5/ comma followed by three /K28.0/ skip characters. The rate matcher deletes or inserts skip characters when necessary to prevent the rate matching FIFO buffer from overflowing or underflowing.

The rate matcher can delete skip characters on only one skip character in a consecutive cluster of skip characters. Figure 2–116 shows an example of a PIPE mode rate matcher deletion of two skip characters.

Figure 2–116. PIPE Mode With Two Deletions (One Deletion Per Cluster)



The rate matcher can insert skip characters one insertion per skip cluster. There is no limit on the consecutive number of skip characters allowed per skip cluster.

The Stratix II GX rate matcher in PIPE mode has FIFO buffer overflow and underflow protection. In the event of a FIFO buffer overflow, the rate matcher deletes any data after the overflow condition to prevent FIFO pointer corruption until the rate matcher is not full. In an underflow condition, the rate matcher inserts 9'h1FE (/K30.7/) until the FIFO buffer is not empty. These measures ensure that the FIFO buffer can gracefully exit the overflow/underflow condition without requiring a FIFO reset.

Power State

There are four supported power states in Stratix II GX devices: P0, P0s, P1, and P2. P0 is the normal power state. P0s is a low recovery time power state that is lower than P0. P1 is a lower power state than P0s and have higher latency to come out of this state. P2 is the lowest power state supported by this mode.

The powerdn port transitions the transceiver into different power states. The encoded value is shown in Table 2–34. The pipephydonestatus signal reacts to the powerdn request and pulses high for one parallel clock cycle.

There are specific functions that are performed at each of the power states. The power-down states are for PIPE emulation. The transceiver does not go into actual power saving mode, with the exception of the transmitter buffer for Electrical Idle. This shouldn't matter because the power consumption for the PLD logic is much greater than the transceiver power consumption. Powering down the transceiver does not save much when compared to the overall power consumption of the entire device. Table 2–34 shows each power state and its function.

Table 2–34. Power State Functions and Descriptions

Power State	powerdn	Function	Description
P0	2'b00	Transmits normal data, transmits Electrical Idle, or enters into loopback mode.	Normal operation mode
P0s	2'b01	Only transmits Electrical Idle.	Low recovery time power saving state
P1	2'b10	Transmitter buffer is powered down and can do a receiver detect while in this state.	High recovery time power saving state
P2	2'b11	Transmits Electrical Idle or a beacon to wake up the downstream receiver.	Lowest power saving state

There are two signals associated with the power states: tx_detectrxloopback and tx_forceelecidle. The tx_detectrxloopback signal controls whether the channel goes into loopback when the power state is in P0 or receiver detect when in P1 state. This signal does not have any affect in any other power states. The tx_forceelecidle signal governs when the transmitter goes into an Electrical Idle state. The tx_forceelecidle signal is asserted in P0s and P1 states and deasserted in P0 state. In P2 state, under normal conditions, the tx_forceelecidle signal is asserted and then deasserted when the beacon signal must be sent out, signifying the intent to exit the P2 power-down state. Table 2–35 shows the behavior of the tx_detectrxloopback and tx_forceelecidle signals in the power states.

Table 2–35. Power States and Functions Allowed in Each Power State		
Power State	**tx_detectrxloopback**	**tx_forceelecidle**
P0	0: normal mode 1: data path in loopback mode	0: Must be deasserted. 1: Illegal mode
P0s	Don't care	0: Illegal mode 1: Must be asserted in this state
P1	0: Electrical Idle 1: receiver detect	0: Illegal mode 1: Must be asserted in this state
P2	Don't care	Deasserted in this state for sending beacon. Otherwise asserted.

Receiver Status

The PIPE interface for PCI Express has a receiver status indicator that reports the status of the PHY (PCS and PMA). The receiver status is communicated to the PLD logic by the 3-bit `pipestatus` port. This port enumerates the status as shown in Table 2–36. If more than one event occurs at the same time, the signal is resolved with the higher priority status.The skip character added and removed flags (3'b001 and 3'b010) are not supported. The `pipestatus` port may be encoded to 3b'001 and 3'b010, which should be ignored. It does not indicate that a skip has been added or removed and should be considered the same as 3'b000—received data. If the upper MAC layer must know when a skip character was added or removed, Altera recommends monitoring the number of skip characters received. The transmitter should send three skip characters in a standard skip-ordered set.

Table 2–36. pipestatus Description and Priority		
pipestatus	**Description**	**Priority**
3'b000	Received data	6
3'b001	One skip character added (not supported)	N/A
3'b010	One skip character removed (not supported)	N/A
3'b011	Receiver detected	1
3'b100	8B/10B decoder error	2
3'b101	Elastic buffer overflow	3
3'b110	Elastic buffer underflow	4
3'b111	Received disparity error	5

An additional status port, rx_pipedatavalid, indicates that the data on the rx_dataout port is valid. This signal is equivalent to the rx_syncstatus port. The rx_pipedatavalid port operates in parallel with the pipestatus signal.

Receiver Detect

The receiver detect circuitry is available for PCI Express applications. The receiver detect circuitry is only available in the P1 power state and is set through the tx_detectrxloopback port and requires a 125 MHz fixedclk signal. In the P1 power state, a high on tx_detectrxloopback port triggers the receiver detect circuitry to alter the transmitter buffer common mode voltage. The sudden change in common mode voltage effectively appears as a step voltage at the serial link. If a receiver (that complies to PCI-Express input impedance requirements) is present at the far end, the time constant of the step voltage is higher. If a receiver is not present or powered down, the time constant of the step voltage is lower. The receiver detect circuitry snoops the serial line (tx_dataout) for the time constant of the step voltage to detect the presence of the receiver at the far end.

Upon receiver detect, the pipestatus port indicates if a receiver is detected or not. There is some latency after asserting the tx_detectrxloopback signal, before the receiver detection is indicated. The tx_forceelecidle port must be asserted at least 10 parallel clock cycles prior to the tx_detectrxloopback to ensure that the transmitter buffer is tri-stated.

Beacon Transmission

The beacon is an optional 30 kHz to 500 MHz in-band signal that wakes the receiver from a P2 power state. This signal is optional, and the Stratix II GX device does not have dedicated beacon transmission circuitry. The Stratix II GX device supports the transmission of the beacon signal through a 10-bit encoded code word that has a five 1's pulse (for example, K28.5).

Because the beacon signal is a pulse that ranges from 2 ns to 500 ns, sending out a K28.5 at 2.5 Gbps meets the lower requirement with its five 1's pulse. (Though other 8B/10B code groups might meet the beacon requirement, this document uses the K28.5 control code group as the beacon signal.) The beacon transmission takes place only in the P2 power state. The tx_forceelecidle port controls when the transmitter is in Electrical Idle or not. This port must be deasserted in order to transmit the K28.5 code group.

Compliance Pattern Transmission Support

PCI Express has an option to transmit a compliance pattern for testing purposes. The compliance pattern must be transmitted with a negative disparity. In PIPE mode, you set the negative disparity with the `tx_forcedispcompliance` port.

Asserting the `tx_forcedispcompliance` port sets the associated byte in the `tx_datain` port to be encoded, by the 8B/10B encoder, to a negative disparity. If a wider PLD interface is used, only the LSByte is encoded with a negative disparity. The `tx_forcedispcompliance` port must be deasserted after the first byte of the compliance pattern is clocked into the transceiver.

The compliance pattern generator is not part of the Stratix II GX transceiver and must be designed using the PLD logic. However, you can set the beginning of the disparity of the compliance pattern to negative by asserting the `tx_forcedispcompliance` port.

NTFS Fast Recovery IP (NFRI)

The PCI-E specification fast training sequences (FTS) are used for bit and byte synchronization to transition from L0s state to L0 (Stratix II GX P0s to P0) power states. The PCI-E base specification states that the required time period for this transaction be within 16 ns to 4 us. Currently, the default PIPE ALT2GXB settings do not meet these requirements. Therefore, Altera developed NTFS fast recovery IP (NFRI), a soft IP that enables the receiver to transition from the P0s to the P0 state within 4 us. The Quartus II software creates this NFRI soft IP when the **Enable fast recovery mode** option is selected in the ALT2GXB MegaWizard Plug-In Manager. This option is available from the Quartus II software version 6.0, SP1.

FTS ordered sets are used by the receiver to detect exit from Electrical Idle (EIdle) and align the receiver's bit/symbol receiver circuitry to the incoming data. If the FTS time period (4 us) expires prior to the receiver obtaining alignment and deskew on all lanes, the receiver transitions to the recovery state (the PCI Express [PIPE] ALT2GXB performs only word alignment. The deskew operation should be done in the user logic). Following the electrical idle condition, the Stratix II GX device requires 255 FTS sequences to recover valid data.

PIPE Mode Default Settings

In the PIPE mode default settings, the receiver PLL is in automatic lock mode. The PLL moves from lock to reference mode to lock to data mode based on the `rx_freqlocked` being asserted. For the `rx_freqlocked` signal to be asserted, the CRU PLL clock should be within the PPM threshold settings of the CRU reference clock.

The PPM detector checks the PPM threshold settings by comparing the CRU PLL clock output with the reference clock for approximately 32768 clock cycles. For a 250 MHz PLD interface clock frequency, this comparison time period exceeds 4 us, which violates the PCI-E specification. The NTFS soft IP overcomes the restriction.

Enable Fast Recovery Mode Option

When you select the **Enable fast recovery mode** option, consider the following:

- NFRI is created in the PLD side for each PCI-E channel
- NFRI is a soft IP, so it consumes logic resources
- This block is self-contained, so no input/output ports are available to access the soft IP

The NFRI takes control of the `rx_locktorefclk` and `rx_locktodata` signals on the ALT2GXB transceiver and therefore avoids the delay of the PPM detector discussed in "PIPE Mode Default Settings" on page 2–160.

If you select the `rx_locktorefclk` and `rx_locktodata` signals in the MegaWizard Plug-In Manager, the **Enable fast recovery mode** option cannot be used.

NTFS Fast Recovery IP (NFRI) in Software Versions before Quartus II 7.2

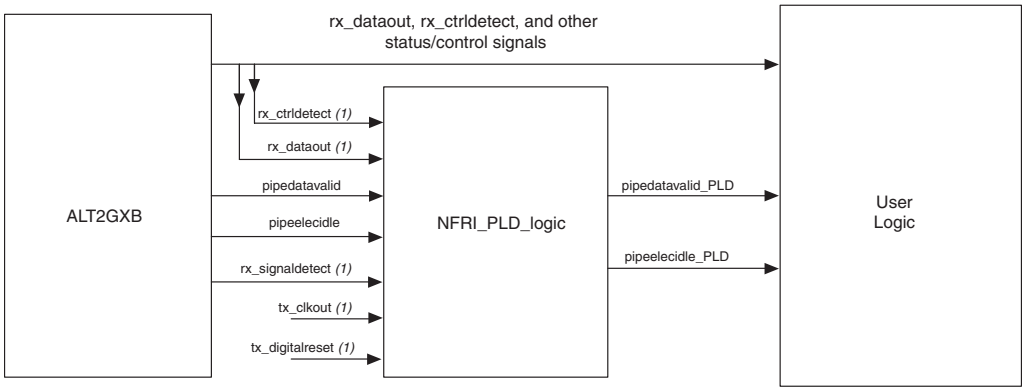
This section discusses the solution to control the sequence of events around the results of the 4 us timing requirement during the P0s to P0 state transition. This is only applicable for software versions before Quartus II 7.2 with the transceiver channel configured in PCI Express (PIPE) mode and the optional **fast recovery mode** enabled in the ALT2GXB MegaWizard. Use this for synchronous and non-synchronous PCI Express (PIPE) modes. The solution requires that you use the data in user logic in the core only if the data is valid.

In the Quartus II software version 7.2, the solution is built into the software when you enable the **fast recovery mode** option in the ALT2GXB MegaWizard.

Some logic can be added per channel which controls the circuit with different possible results. The proposed logic in the PLD is referred to as **NFRI_PLD_logic**. The output signals generated by the **NFRI_PLD_logic** are referred to as `pipeelecidle_PLD` and `pipedatavalid_PLD`.

Figure 2–117 shows the top-level block diagram of the overall system.

Figure 2–117. NFRI_PLD_LOGIC Top Level Diagram



Note to Figure 2–117:
(1) This signal is also provided to the user logic.

The proposed logic runs with the `tx_clkout`. To reset this logic, use the same signal that connects to the `tx_digitalreset` port of the PCI-Express (PIPE) ALT2GXB instance.

The **NFRI_PLD_logic** contains:

- A state machine sequence to generate the `pipeelecidle_PLD` and `pipedatavalid_PLD` signals.
- `4us_timer`: A user-implemented timer in the PLD logic to count 4 us. This timer represents the maximum time period to transition from P0s to P0 per the protocol specification.
- `3.2us_timer`: A user-implemented timer in the PLD logic to count 3.2 us. This timer represents the minimum time period to wait before looking for valid data.

Sequence to Generate pipeelecidle_PLD and pipedatavalid_PLD Signals

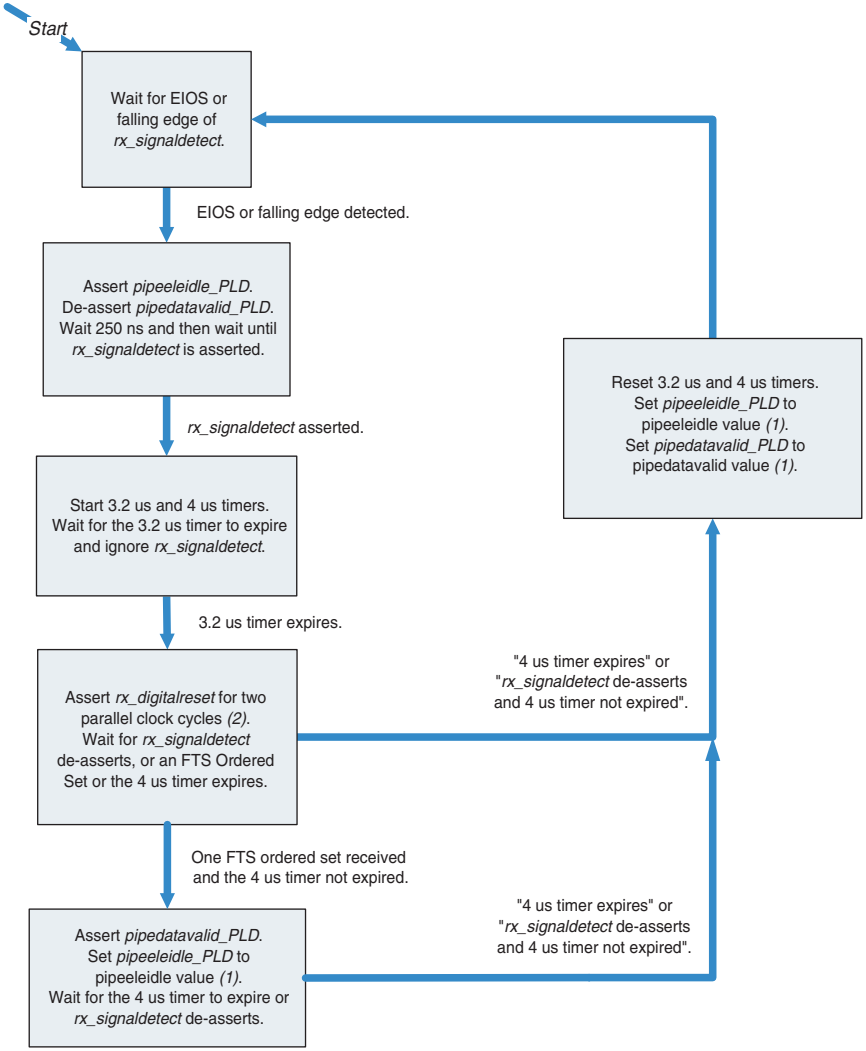
To generate the pipeelecidle_PLD and pipedatavalid_PLD signals, follow these steps:

1. When you detect EIOS ordered set or the falling edge of rx_signaldetect:
   - assert pipeelecidle_PLD
   - deassert pipedatavalid_PLD
2. Wait 250 ns. This is the minimum time required for rx_signaldetect to get deasserted when there is no valid signal at the receive input.
3. Wait for rx_signaldetect to get asserted.
4. Start the 3.2us_timer and 4us_timer. Ignore rx_signaldetect during this step.
5. Wait for the 3.2us_timer to expire. Ignore rx-signaldetect during this step.
6. If one FTS ordered set is received and the 4us_timer has NOT expired:
   - Set pipeelecidle_PLD to the pipeelecidle value (that is, forward the pipeelecidle value from the ALT2GXB to the user logic).
   - Assert pipedatavalid_PLD.
   - Reset and pause the 3.2us_timer and 4us_timer.
   - Return to Step 1.
7. If rx_signaldetect gets deasserted and the 4us_timer has NOT expired:
   - Reset and pause the 3.2us_timer and 4us_timer.
   - Set pipeelecidle_PLD to the pipeelecidle value (that is, forward the pipeelecidle value from the ALT2GXB to the user logic). Set the pipedatavalid_PLD to the pipedatavalid value (that is, forward the pipedatavalid value from the ALT2GXB to the user logic).
   - Return to Step 1.

8. If the `4us_timer` has expired:
   - Reset and pause the `3.2us_timer` and `4us_timer`.
   - Set the `pipeelecidle_PLD` to the pipeelecidle value (that is, forward the pipeelecidle value from the ALT2GXB to the user logic).
   - Set `pipedatavalid_PLD` to the pipedatavalid value (that is, forward the pipedatavalid value from the ALT2GXB to the user logic).
   - Return to Step 1.

Figure 2–118 shows the NFRI_PLD_logic state machine.

Figure 2–118. NFRI_PLD_logic State Machine



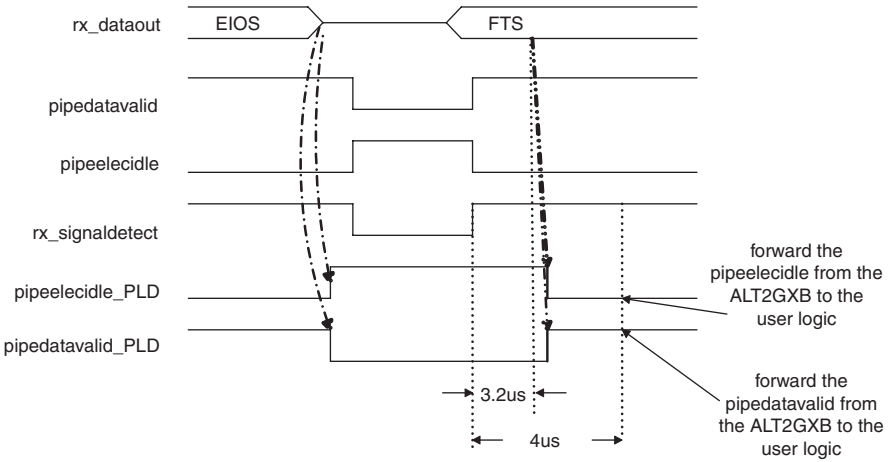
Notes to Figure 2–118:

(1) Forward the ALT2GXB value to the `NFRI_PLD_logic` output. For example, forward the pipedatavalid value from the ALT2GXB to the `pipedatavalid_PLD` for use in user logic.

(2) After the `3.2us_timer` expires, it is important to reset `rx_digitalreset` for two parallel clock cycles. For example, now that correct data is coming through, the FIFO should be cleared and the pointers reset to the middle. Following this reset, look for the FTS ordered set. For other cases to assert `rx_diditalreset`, refer to "Reset Sequence for PIPE Mode" on page 2–218.

Figure 2–119 shows the timing diagram for the signals generated by the NFRI_PLD_logic.

Figure 2–119. Timing for the Signals Generated by the NFRI_PLD_logic



Low-Latency PIPE mode

The Stratix II GX receiver data path employs a rate match FIFO in PIPE mode to compensate up to ± 300 PPM difference between the upstream transmitter and the local receiver reference clock. The rate match FIFO adds a latency of 12-16 parallel clock cycles to the link. The low-latency PIPE mode allows bypassing the rate match FIFO in synchronous systems that derive the transmitter and receiver reference clocks from the same source. You can bypass the rate match FIFO by not selecting the **Enable Rate match FIFO** option in the ALT2GXB MegaWizard Plug-In Manager.

You can bypass the rate match FIFO in single-lane (×1), four-lane (×4) and eight-lane (×8) PIPE modes. In normal PIPE mode, the receiver blocks following the rate match FIFO are clocked by `tx_clkout` (×1 mode) or `coreclkout` (×4 and ×8 modes) of the local port. In low-latency mode, since the rate match FIFO is bypassed, these receiver blocks are clocked by the recovered clocks of the respective channels. As a result, the channels in a multi-lane (×4 or ×8) low-latency PIPE mode are unbonded.

Except for the rate match FIFO being bypassed and the resulting changes in transceiver internal clocking, the low-latency PIPE shares the same data path and state machines as the normal PIPE mode. However, some features supported in normal PIPE mode are not supported in low-latency PIPE mode.

Unlike regular PIPE mode, the Quartus II software in low-latency PIPE mode does not automatically clock the transmitter and receiver phase compensation FIFO write and read clock with the `tx_clkout` signal from channel 0. You must use the **Shared Clock Grouping** or **0 PPM Clock Grouping** assignments to manually clock the phase compensation FIFOs. Refer to "PLD-Transceiver Interface Clocking" on page 2–119 for more information on clock grouping.

PIPE Reverse Parallel Loopback

In normal PIPE mode, if the transceiver is in P1 power state, a high value on the `tx_rxdetectloop` signal forces a reverse parallel loopback as discussed in "PCI Express PIPE Reverse Parallel Loopback" on page 2–206. Parallel data at the output of the receiver rate match FIFO gets looped back to the input of the transmitter serializer.

In low-latency PIPE mode, since the rate match FIFO is bypassed, this feature is not supported. A high value on the `tx_rxdetectloop` signal, when the transceiver is in P1 power state, will not force it to perform reverse parallel loopback.

Link Width Negotiation

In normal multi-lane (×4 and ×8) PIPE configuration, the receiver phase compensation FIFO control signals (for example, `write/read enable`) are shared among all lanes within the link. As a result, all lanes are truly bonded and the lane-lane skew meets the PCI Express specification.

In low-latency PIPE configuration, the receiver phase compensation FIFO of individual lanes do not share control signals. The write port of the receiver phase compensation FIFO of each lane is clocked by its recovered clock. As a result, the lanes within a link are not bonded. You should perform external lane de-skewing to ensure proper link width negotiation.

PIPESTATUS Signal

Since the rate match FIFO is bypassed in low-latency PIPE mode, status signal combinations related to the rate match FIFO on `pipestatus[2:0]` port become irrelevant and must not be interpreted (Table 2–37).

Table 2–37. Normal and Low-Latency PIPE Status (Part 1 of 2)

pipestatus[2:0]	Normal PIPE	Low-Latency PIPE
000	Received Data OK	Received Data OK
001	Not supported	Not supported

Table 2–37. Normal and Low-Latency PIPE Status (Part 2 of 2)

pipestatus[2:0]	Normal PIPE	Low-Latency PIPE
010	Not supported	Not supported
011	Receiver Detected	Receiver Detected
100	8B/10B Decoder Error	8B/10B Decoder Error
101	Elastic Buffer Overflow	Not supported
110	Elastic Buffer Underflow	Not supported
111	Received Disparity Error	Received Disparity Error

XAUI Mode

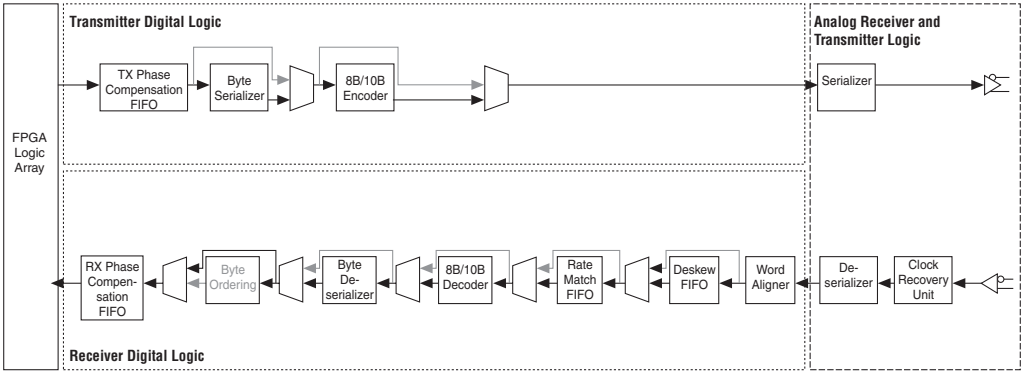
This section briefly introduces the XAUI standard (refer to Figure 2–120) and the code groups and ordered sets associated with this self-managed interface. For full details on the XAUI standard, refer to clause 47 and 48 in the 10 Gigabit Ethernet standard (IEEE 802.3ae).

Stratix II GX devices contain embedded macros dedicated to the XAUI protocol, including synchronization, channel deskew, rate matching, XGMII Extender Sublayer (XGXS) to 10 Gigabit Media Independent Interface (XGMII) and XGMII to XGXS code-group conversion macros.

For HiGig, the Stratix II GX XAUI data rate protocol has been extended from 3.125 Gbps up to 3.75 Gbps. For HiGig data rates, select the XAUI protocol and type in the increased data rate.

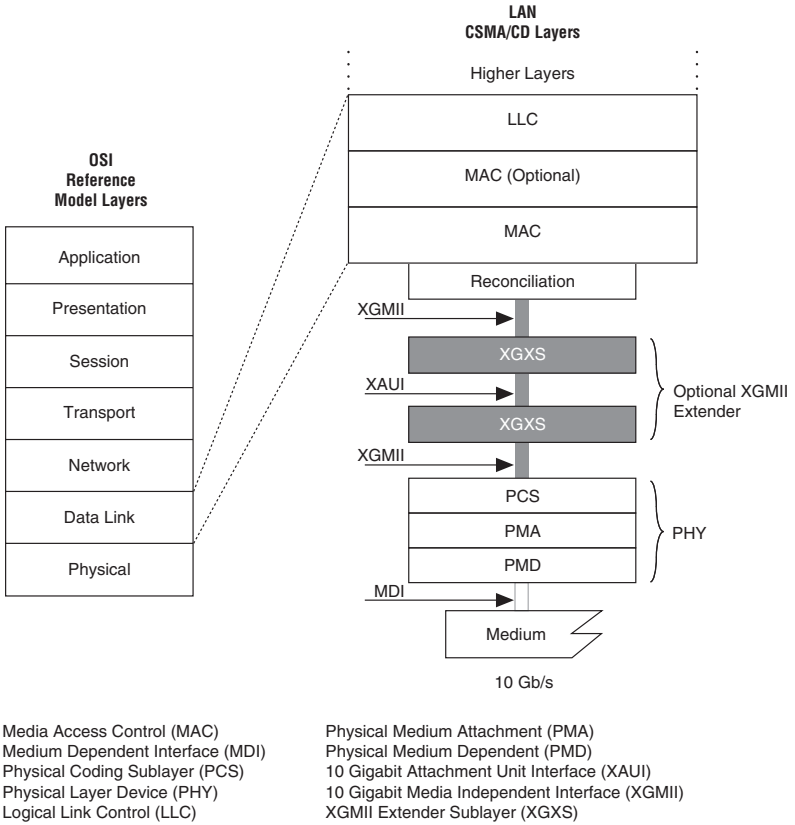
The XAUI standard is an optional self-managed interface that is inserted between the reconciliation sublayer and the PHY layer to transparently extend the physical reach of XGMII.

Figure 2–120. XAUI Mode



XAUI addresses several physical limitations of XGMII. XGMII signaling is based on the HSTL Class I single-ended IO standard, which has an electrical distance limitation of approximately 7 cm. XAUI utilizes a low-voltage differential signaling method, so the electrical limitation is increased to approximately 50 cm. Another advantage of XAUI is the simplification of backplane and board trace routing. XGMII is composed of 32 transmit channels, 32 receive channels, 1 transmit clock, 1 receive clock, 4 transmitter control characters, and 4 receive control characters for a total of a 74-pin wide interface. XAUI consists of 4 differential transmitter channels and 4 differential receiver channels for a total of a 16-pin-wide interface. This reduction in pin count significantly simplifies the routing process in the layout design. Figure 2–121 shows the relationships between the XGMII and XAUI layers.

Figure 2–121. XGMII and XAUI Relationship



The XGMII interface consists of four lanes of 8 bits. At the transmit side of the XAUI interface, the data and control characters are converted within the XGXS into an 8B/10B encoded data stream. Each data stream is then transmitted across a single differential pair running at 3.125 Gbps (3.75 Gbps for HiGig). At the XAUI receiver, the incoming data is decoded and mapped back to the 32 bit XGMII format. This provides a transparent extension of the physical reach of the XGMII and also reduces the interface pin count.

XAUI functions as a self-managed interface because code group synchronization, channel deskew, and clock domain decoupling is handled with no upper layer support requirements. This functionality is

based on the PCS code groups that are used during the IPG time and idle periods. PCS code groups are mapped by the XGXS to XGMII characters specified in Table 2–38.

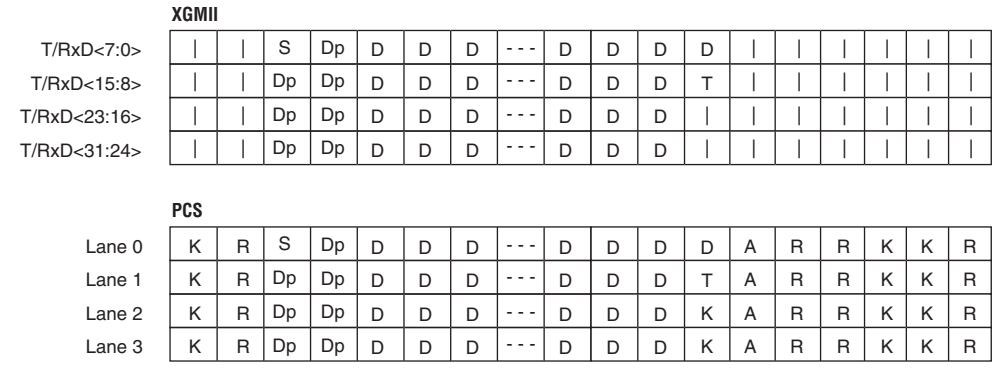
Table 2–38. XGMII Character to PCS Code-Group Mapping

XGMII TXC	XGMII TXD *(1)*	PCS Code Group	Description
0	00 through FF	Dxx.y	Normal data transmission
1	07	K28.0, K28.3, or K28.5	Idle in ||I||
1	07	K28.5	Idle in ||T||
1	9C	K28.4	Sequence
1	FB	K27.7	Start
1	FD	K29.7	Terminate
1	FE	K30.7	Error
1	Other value		Reserved XGMII character
1	Any other value	K30.7	Deleted XGMII character

Note to Table 2–38:
(1) Values in TXD column are in hexadecimal.

Figure 2–122 shows an example of the mapping between XGMII characters and the PCS code groups that are used in XAUI. The idle characters are mapped to a pseudo-random sequence of /A/, /R/, and /K/ code groups.

Figure 2–122. XGMII Character to PCS Code-Group Mapping



The PCS code-groups are sent via PCS ordered sets. PCS ordered sets consist of combinations of special and data code groups defined as a column of code groups. These ordered sets are composed of four code groups beginning in lane 0. Table 2–39 lists the defined idle ordered sets (||I||) that are used for the self-managed properties of XAUI.

Table 2–39. Defined Idle Ordered Set

Code	Ordered Set	Number of Code Groups	Encoding
||I||	Idle		Substitute for XGMII Idle
||K||	Synchronization column	4	/K28.5/K28.5/K28.5/K28.5
||R||	Skip column	4	/K28.0/K28.0/K28.0/K28.0
||A||	Align column	4	/K28.3/K28.3/K28.3/K28.3

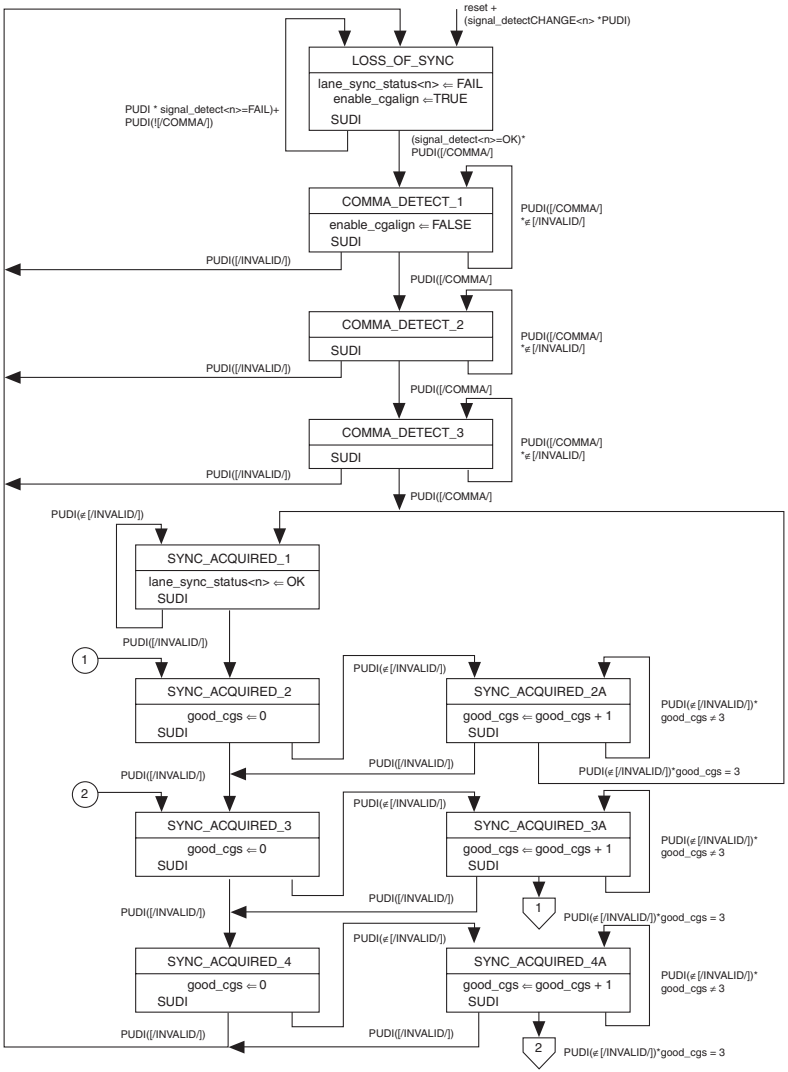
Synchronization ||K|| (Word Aligner)

XAUI uses an embedded clocking scheme that re-times the data that can potentially alter the code-group boundary. The boundaries of the code groups are re-aligned through a synchronization process specified in clause 48 of the IEEE P802.3ae standard, which states that synchronization is achieved upon the reception of four /K28.5/ commas. Each comma can be followed by any number of valid code groups. Invalid code groups are not allowed during the synchronization stage.

When you configure Stratix II GX devices to the XAUI protocol, the built-in pattern detector, word aligner, and XAUI state machines adhere to the PCS synchronization specification. After all the conditions for synchronization have been met, the `rx_syncstatus` signal is asserted and only de-asserts if synchronization is lost.

Figure 2–123 shows the PCS synchronization state diagram specified in clause 48 of the IEEE P802.3ae.

Figure 2–123. IEEE 802.3ae PCS Synchronization State Diagram

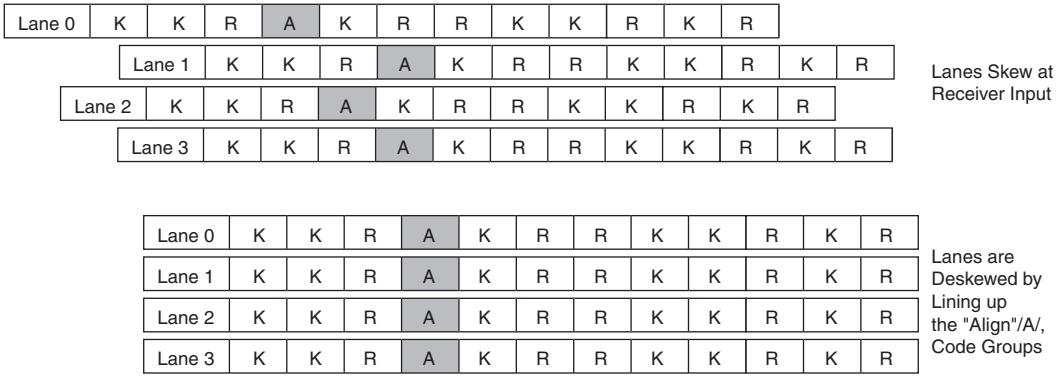


Channel Alignment ||A|| (Deskew)

It is possible for ordered sets to be misaligned with respect to one another because of board skew or differences between the independent clock recoveries per serial lane. Channel alignment, also referred to as deskew or channel bonding, realigns the ordered sets by using the alignment code group, referred to as /A/. The /A/ code group is transmitted simultaneously on all four lanes, constituting an ||A|| ordered set, during idles or IPG. XAUI receivers use these code groups to resolve any lane-to-lane skew. Skew between the lanes can be up to 40 UI (12.8 ns) as specified in the standard, which relaxes the board design constraints.

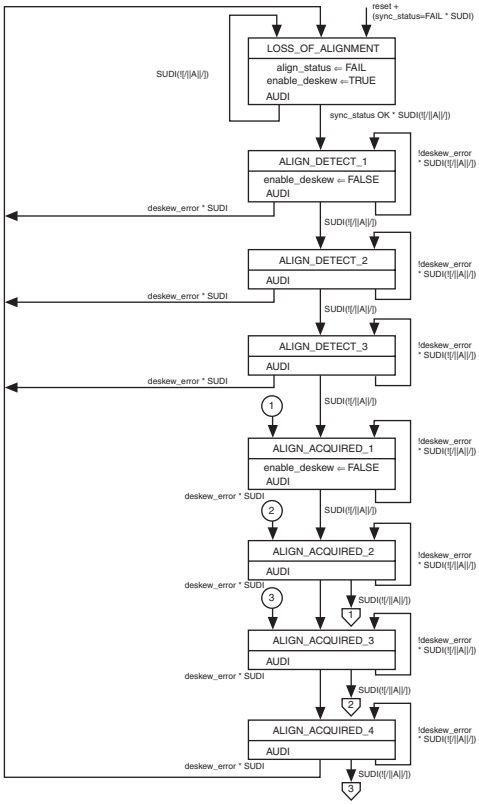
Figure 2–124 shows lane skew at the receiver input and how the deskew circuitry uses the /A/ code group to deskew the channels.

Figure 2–124. Lane Deskew With the /A/ Code Group



Stratix II GX devices manage XAUI channel alignment with a dedicated deskew macro that consists of a 16-word-deep FIFO buffer that is controlled by a XAUI deskew state machine. The XAUI deskew state machine first looks for the /A/ code group within each channel. When the XAUI deskew state machine detects /A/ in each channel, the deskew FIFO buffer is enabled. The deskew state machine now monitors the reception of /A/ code groups. When four aligned /A/ code groups have been received the `rx_channelaligned` is asserted. The deskew state machine continues to monitor the reception of /A/ code groups and de-asserts the `rx_channelaligned` signal if alignment conditions are lost. This built-in deskew macro is only enabled for the XAUI protocol. Figure 2–125 shows the PCS deskew state diagram specified in clause 48 of the IEEE P802.3ae.

Figure 2–125. IEEE 802.3ae PCS Deskew State Diagram



Clock Compensation ||R|| (Rate Matcher)

XAUI can operate in multi-crystal environments, which can tolerate frequency variations of ±100 PPM between crystals. Stratix II GX devices contain embedded circuitry to perform clock rate compensation, which is achieved by inserting or removing the PCS SKIP code group (/R/) from the IPG or idle stream. This process is called rate matching and is sometimes referred to as clock rate compensation.

The rate matcher in Stratix II GX devices consists of a 12-word-deep FIFO buffer along with control logic that you can configure to support XAUI, GIGE, or custom modes. In XAUI mode the controller begins to write data into the FIFO buffer whenever the `rx_channelaligned` signal is asserted. Within the control logic there is a FIFO counter that keeps track of the read and write executions. When the FIFO counter reaches a value of greater than nine, the receivers delete the /R/ code-group

simultaneously across all channels during IPG or idle conditions. If the FIFO counter is less than five, the receivers insert the /R/ code-group simultaneously across all channels during IPG or idle conditions.

This circuitry compensates for ±100 PPM frequency variations.

PCS Code Group to XGMII Character Mapping

In XAUI mode, the 8B/10B decoder in Stratix II GX devices is controlled by a global receiver state machine that maps various PCS code groups into specific 8-bit XGMII codes. Table 2–40 lists the PCS code group to XGMII character mapping.

Table 2–40. PCS Code Group to XGMII Character Mapping

XGMII RXC	XGMII RXD *(1)*	PCS Code Group	Description
0	00 through FF	Dxx.y	Normal data transmission
1	07	K28.0, K28.3, or K28.5	Idle in ||I||
1	07	K28.5	Idle in ||T||
1	9C	K28.4	Sequence
1	FB	K27.7	Start
1	FD	K29.7	Terminate
1	FE	K30.7	Error
1	FE	Invalid code group	Received code group

Note to Table 2–40:
(1) Values in RXD column are in hexadecimal.

XGMII Character to PCS Code-Group Mapping

In XAUI mode, the 8B/10B encoder in Stratix II GX devices is controlled by a global transmitter state machine that maps various 8-bit XGMII codes to 10-bit PCS code groups. This state machine complies with the IEEE 802.3ae PCS transmit specification. Figure 2–126 shows the PCS transmit source state diagram specified in clause 48 of the IEEE P802.3ae.

Figure 2–126. IEEE 802.3ae PCS Transmit Source State Diagram

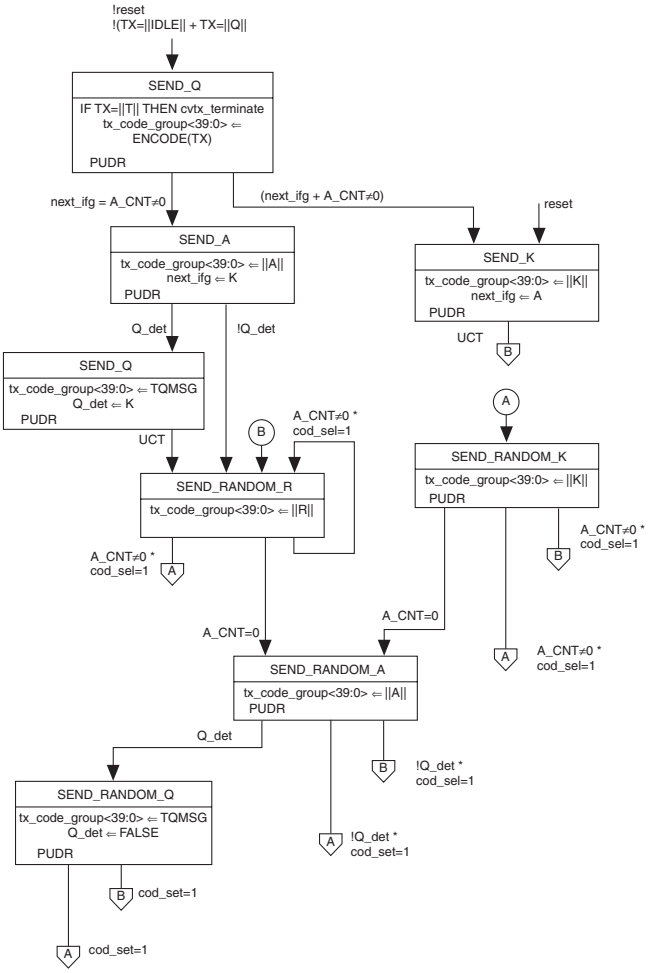


Table 2–41 lists the XGMII character to PCS code-group mapping.

Table 2–41. XGMII Character to PCS Code-Group Mapping

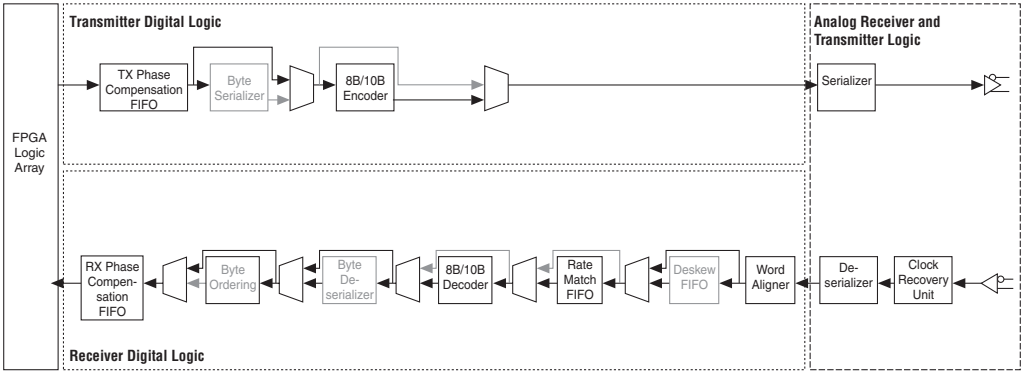
XGMII TXC	XGMII TXD *(1)*	PCS Code Group	Description
0	00 through FF	Dxx.y	Normal data transmission
1	07	K28.0, K28.3, or K28.5	Idle in ||I||
1	07	K28.5	Idle in ||T||
1	9C	K28.4	Sequence
1	FB	K27.7	Start
1	FD	K29.7	Terminate
1	FE	K30.7	Error
1	Other value		Reserved XGMII character
1	Any other value	K30.7	Invalid XGMII character

Note to Table 2–41:
(1) Values in TXD column are in hexadecimal.

GIGE Mode

The Gigabit Media Independent Interface (GMII) is an intermediate, or transition, layer that interfaces various mediums with the media access control (MAC) in a GIGE system (refer to Figure 2–127). The GMII is divided into three sublayers: the PCS, the PMA, and the physical medium dependent (PMD) layers. GMII offers data rates up to 1000 Mbps at either half- or full-duplex modes.

Figure 2–127. GIGE Mode



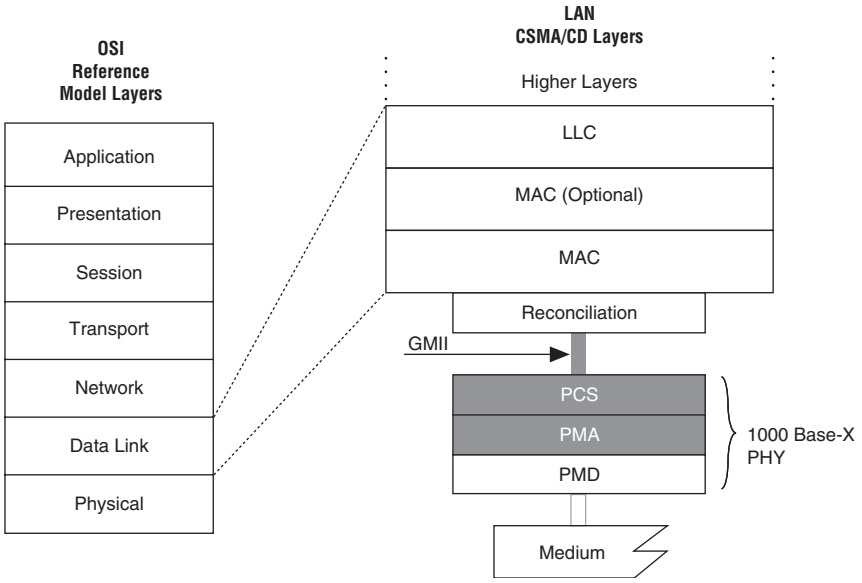
The PCS sublayer provides synchronization, encoding and decoding, and rate matching services to the MAC. The PCS also provides auto negotiation to the network to negotiate speeds, carrier, and collision detect signals.

The PMA sublayer provides the PCS with a media-independent interface that a variety of serial physical media can be connected to. This layer handles the serialization and deserialization of the data.

The PMD sublayer defines actual physical attachment, such as connectors for different media types.

Figure 2–128 shows the GMII position relative to the OSI reference model.

Figure 2–128. GMII Position Relative to the OSI Reference Model



The Stratix II GX transceiver can be used for the PCS and the PMA layers of the GMII. Stratix II GX devices in GIGE mode use the 8B/10B encoder/decoder, rate matcher, and synchronizer built-in hard macros. The rate matcher and synchronizer have a dedicated state machine governing their functions. This state machine is only active in GIGE mode.

Table 2–42 shows the code groups used in the GIGE protocol. If required for your design, you must implement the remaining functions of the PCS—auto negotiation, collision detect, and carrier detect—in user logic or external circuits.

Table 2–42. GIGE Code Groups (Part 1 of 2)

Code	Ordered Set	Number of Code Groups	Encoding
/C/	Configuration		Alternating /C1/ and /C2/
/C1/	Configuration 1	4	/K28.5/D21.5/`Config_Reg` *(1)*

Table 2–42. GIGE Code Groups (Part 2 of 2)

Code	Ordered Set	Number of Code Groups	Encoding
/C2/	Configuration 2	4	/K28.5/D2.2/Config_Reg *(1)*
/I/	IDLE		Correcting /I1/, Preserving /I2/
/I1/	IDLE 1	2	/K28.5/D5.6
/I2/	IDLE 2	2	/K28.5/D16.2
	Encapsulation		
/R/	Carrier_Extend	1	/K23.7/
/S/	Start_of_Packet	1	/K27.7/
/T/	End_of_Packet	1	/K29.7/
/V/	Error_Propagation	1	/K30.7/

Note to Table 2–42:
(1) Two data code groups representing the Config_Reg value.

Synchronization (Word Aligner)

Synchronization is required in GIGE mode to align the byte boundary of the receiver to the byte boundary of the transmitter, because the Stratix II GX transceiver block uses a non-source-synchronous serial stream. To correctly align the byte boundary at the receiver, a unique synchronization pattern must be received that does not occur between any Dx.y and/or Kx.y code combinations. A /K28.5/ 10-bit comma is used for this purpose.

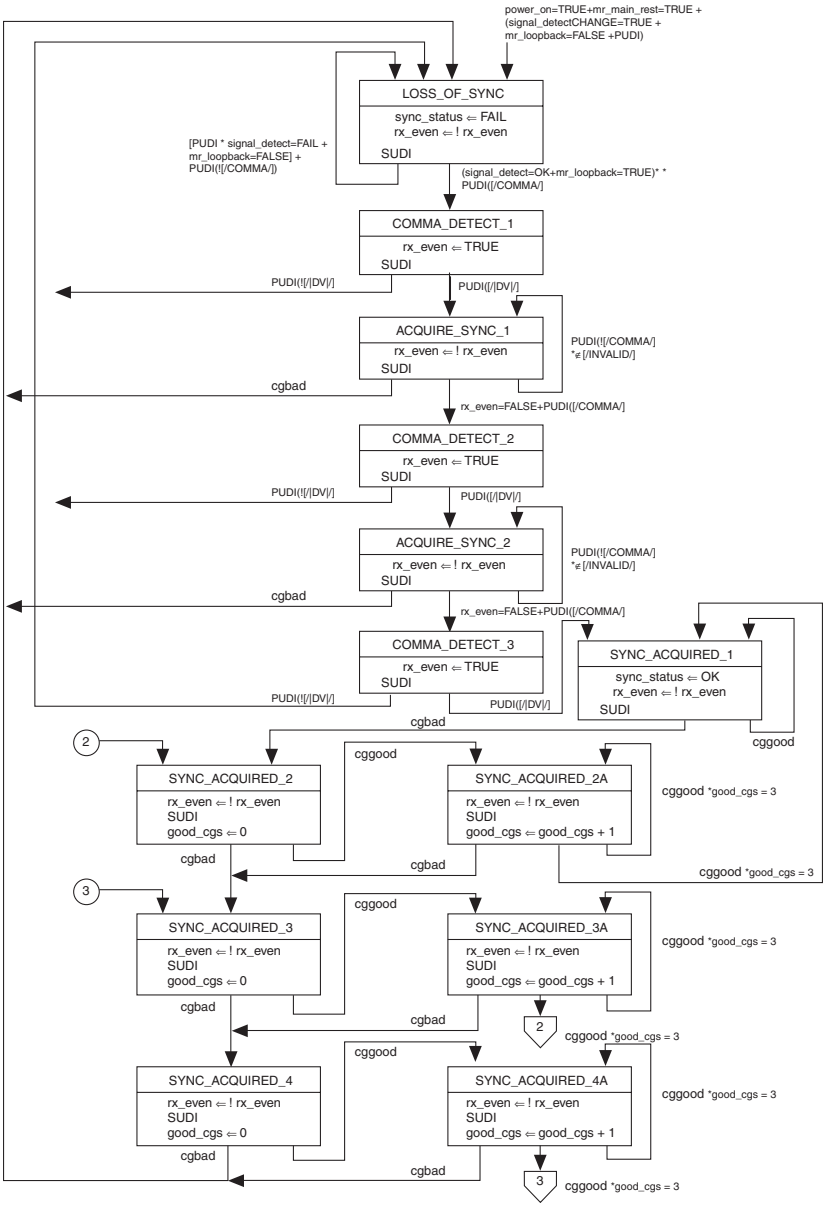
Synchronization is achieved when the receiver sees three consecutive ordered sets. An ordered set defined for synchronization is a /K28.5/ comma followed by any odd number of valid /Dx.y/ code (/Dx.y/ denotes any valid data code group). Although you can have a number of synchronization patterns based on the synchronization rule, three /K28.5/ followed by one /Dx.y/ code is the fastest synchronization pattern.

Once the synchronization is achieved, the state machine considers a bad code group received (cgbad) if one of the following two conditions are met:

- The incoming code group has a disparity error or a code group violation
- The incoming code group is a comma character and rx_even=true. This condition occurs when an even number of non-comma code groups are received between two comma code groups.

GIGE mode requires a special synchronization sequence that follows the IEEE 802.3 GMII PCS synchronization specification (Figure 2–129).

Figure 2–129. Synchronization State Machine Diagram

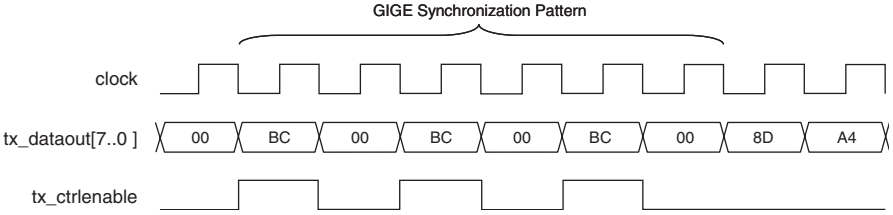


GIGE Transmitter Synchronization

The transmitter must send out the GIGE synchronization sequence to synchronize the target receiver. Stratix II GX devices do not contain a built-in macro that provides this function upon power-up or reset. You must implement this function in user logic to send out a /K28.5/, /Dx.y/, /K28.5/, /Dx.y/, /K28.5/, /Dx.y/ sequence.

Figure 2–130 shows an example of the GIGE synchronization pattern. Although the example shows one D0.0 (8'h00) as the /Dx.y/ code, any /Dx.y/ and any odd number of /Dx.y/ can be used.

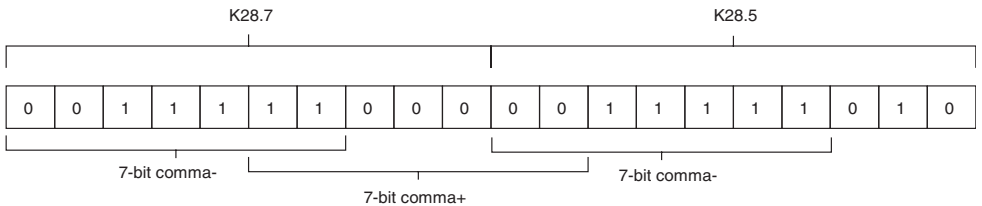
Figure 2–130. GIGE Synchronization Transmit Pattern



GIGE Receiver Synchronization

You must pre configure the receiver with a K28.5 (10'b0101111100 or 10'b1010000011) word align pattern (`ALIGN_PATTERN` = 0101111100 or `ALIGN_PATTERN` = 1010000011). The `ALIGN_PATTERN_LENGTH` must be set to 10 even though a 7-bit comma string (7'b0011111 as a comma- or 7'b1100000 as a comma+) is allowed as specified in IEEE 802.3. This 7-bit comma is located within the /K28.1/, /K28.5/, and /K28.7/ code groups. Using a 10-bit /K28.5/ helps prevent a 7-bit comma from being detected across boundaries when a /K28.7/ code is followed by a /K28.x/, /D3.x/, /D3.x/, /D11.x/, /D12.x/, /D19.x/, /D20.x/, or /D28.x/, where *x* is a value from 0 to 7 (Figure 2–131).

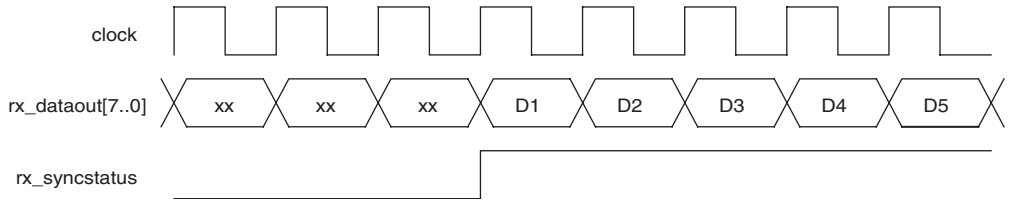
Figure 2–131. Cross Boundary 7-bit Comma When a /K28.7/ is Followed by a /K28.5/ Code Group



The receiver outputs a K28.4 (`8'h9c + rx_ctrldetect`) at the `rx_dataout` port and de-asserts the `rx_syncstatus` (1'b0) signal when the receiver is not synchronized. Once synchronized, the receiver asserts the `rx_syncstatus` signal (1'b1). This signal is aligned with the first valid data received from the `rx_dataout` port.

Figure 2–132 shows the receiver synchronization waveform. The `rx_syncstatus` port goes high when synchronization is complete, indicating that the data is valid. In Figure 2–132, D1 is the first valid data.

Figure 2–132. Synchronization Complete



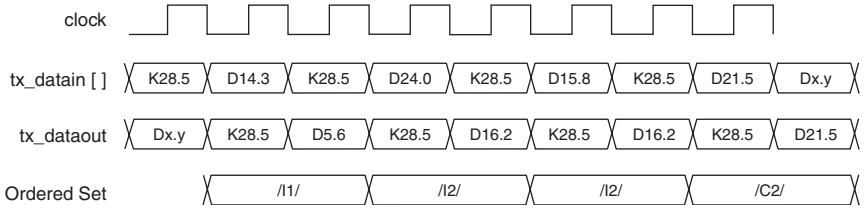
The receiver remains synchronized until it detects a string of bad code groups or is reset. A bad code group is defined by the IEEE 802.3 standard as four invalid code groups separated by less than three valid code groups. If the receiver detects a bad code group or is reset, the `rx_syncstatus` signal goes high, then low and a /K28.4/ appears on the `rx_outrx_dataout` port.

Idle Generation

In GIGE mode, any /Dx.y/ following a /K28.5/ comma is replaced by the transmitter with either a /D5.6/ (8'hc5) or a /D16.2/ (8'h50) depending on the current running disparity, except when the data following the /K28.5/ is /D21.5/ (8'hb5) or /D2.2/ (8'h42). This ensures the generation of the /I1/ (/K28.5/, /D5.6/) and /I2/ (/K28.5/, /D16.2/) ordered sets and to allows the configuration ordered sets /C1/ (/K28.5, /D21.5/) and /C2/ (/K28.5/, /D2.2/) to be received. If the running disparity before the idle ordered set is positive, a /I1/ is chosen. If the running disparity is negative, an /I2/ is chosen. The disparity at the end of an /I1/ is the opposite of that at the beginning of the /I1/. The disparity at the end of an /I2/ is the same as the beginning running disparity (right before the idle code). This ensures a negative running disparity at the end of an idle ordered set. A /Kx.y/ following a /K28.5/ is not replaced.

Figure 2–133 shows the input data codes versus the output data codes. /D14.3/, /D24.0/, and /D15.8/ were replaced by /D5.6/ or /D16.2/ (for /I1/, /I2/ ordered sets) and /D21.5/ (part of the /C2/ order set) was not replaced.

Figure 2–133. Example of the Input Data Codes Versus the Output Data Codes



Rate Matching (Rate Matcher)

GIGE can operate in a multi crystal environment, so rate matching is necessary to compensate for the frequency variations from different crystals. Stratix II GX devices contain a built-in rate matcher (12-word-deep FIFO buffer with control logic) that can tolerate up to, and compensate for, a ±100 PPM frequency variation.

In the GIGE mode, rate matching occurs automatically in the rate matcher. If the GMII protocol is followed, the /I/ sets (/I1/, /I2/) are sent during the inter-frame gap (IFG). (The GMII protocol specifies 96-bits long). The /I2/ ordered set (/K28.5/, /D16.2/) is added or deleted based on how full or empty the rate matcher FIFO buffer is and if the current running disparity is negative. The /I2/ order set contains two 10-bit code groups. Two 10-bit groups (20-bits total) are deleted or added at a time. If the number of words in the FIFO buffer (FIFO count) is greater than nine, the FIFO buffer stops writing when the /I2/ ordered set is detected (Figure 2–134). If the FIFO count is less than five, the FIFO buffer stops reading and inserts the /I2/ ordered sets in place of the next FIFO data (Figure 2–135).

The GIGE rate matcher does not have the capability of inserting or deleting /C1/or /C2/ ordered sets.

If the rate matching FIFO buffer is in an underflow or overflow condition (empty or full), the receiver outputs a /K28.4/ (`8'h9C + ctrl`). This might happen if the PPM (parts per million) difference in the read and write clock is too great, the IFG or IPG is too small (there are not enough /I2/ code groups to remove), and/or the frame or packet size is too big.

Figure 2–134. /I2/ Deleted When FIFO Count is Greater Than Nine

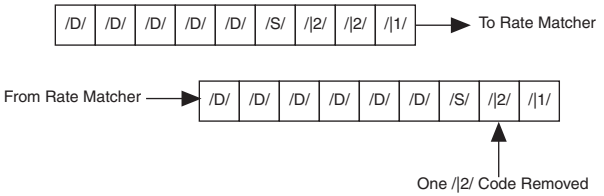
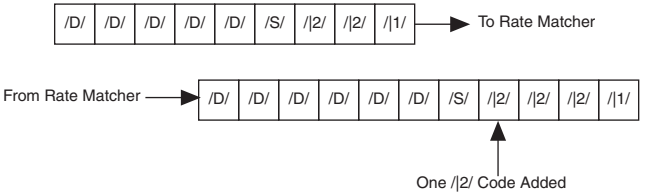


Figure 2–135. /I2/ Added When FIFO Count is Less Than Five



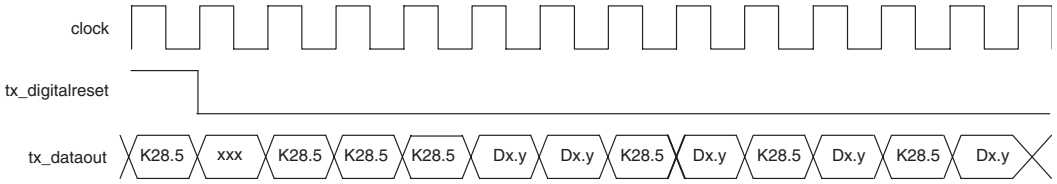
Reset Condition

After power up or reset, the GIGE transmitter outputs three /K28.5/ commas before user data can be sent. This affects the synchronization ordered set transmission.

After reset (`tx_digitalreset`), the 8B/10B encoder automatically sends three /K28.5/ commas (refer to "8B/10B Decoder" on page 2–103 for additional information). Depending on when you start outputting the synchronization sequence, there could be an even or odd number of /Dx.y/ sent as the transmitter before the synchronization sequence. The last of the three automatically sent /K28.5/and the first user-sent /Dx.y/ are treated as one idle ordered set. This can be a problem if there are an even number of /Dx.y/ transmitted before the start of the synchronization sequence.

Figure 2–136 shows an example of even numbers of /Dx.y/ between the last automatically sent /K28.5/ and the first user-sent /K28.5/. The first user-sent ordered set is ignored, so three additional ordered sets are required for proper synchronization. Figure 2–136 shows one don't care data between the `tx_digitalreset` signal going low and the first of three automatic K28.5, but there could be more.

Figure 2–136. GIGE Synchronization Ordered Set Considerations After Reset



SONET/SDH Mode

SONET/SDH is one of the most common serial-interconnect protocols used in backplanes deployed in communications and telecom applications. SONET/SDH defines various optical carrier (OC) subprotocols for carrying signals of different capacities through a synchronous optical hierarchy.

Stratix II GX transceivers can be employed as physical layer devices in a SONET/SDH system. These transceivers provide support for SONET/SDH protocol-specific functions and electrical features; for example, alignment to A1A2 or A1A1A2A2 pattern.

Stratix II GX transceivers are designed to support the following three SONET/SDH subprotocols:

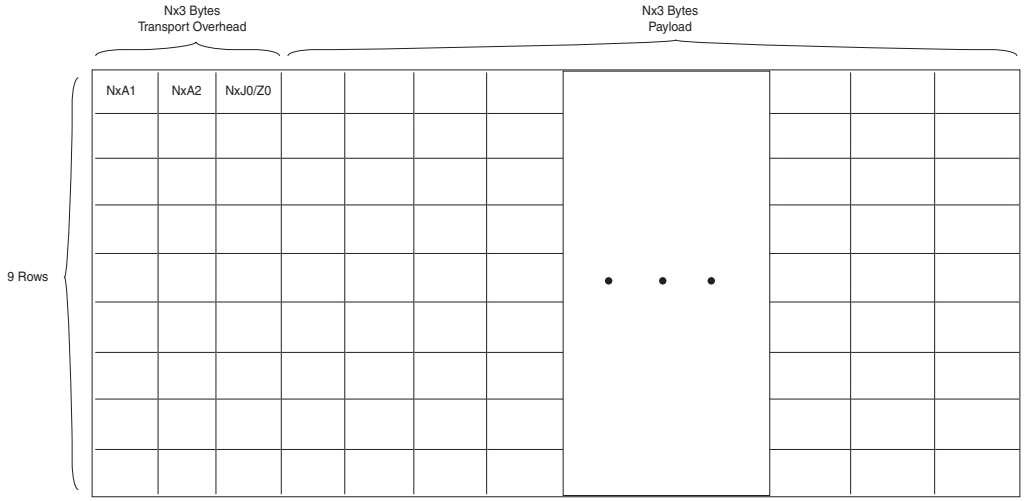
- OC-12 at 622 Mbps with 8-bit channel width
- OC-48 at 2488.32 Mbps with 16-bit channel width
- OC-96 at 4976 Mbps with 32-bit channel width

SONET/SDH Frame Structure

Base OC-1 frames are byte-interleaved to form SONET/SDH frames. For example, twelve OC-1 frames are byte-interleaved to form one OC-12 frame; forty-eight OC-1 frames are byte-interleaved to form one OC-48 frame and so on. SONET/SDH frame sizes are constant, with a frame transfer rate of 125 µs.

Figure 2–137 shows the SONET/SDH frame structure.

Figure 2–137. SONET/SDH Frame Structure



Note to Figure 2–137
(1) N=12 for OC-12, 48 for OC-48, and 96 for OC-96.

Transport overhead bytes A1 and A2 are used for recovering the frame boundary from the serial data stream. Frame sizes are fixed, so the A1 and A2 bytes appear within the serial data stream every 125 µs. In an OC-12 backplane system, twelve A1 bytes are followed by twelve A2 bytes. Similarly, in an OC-48 backplane system, forty-eight A1 bytes are followed by forty-eight A2 bytes.

In SONET/SDH systems, byte values of A1 and A2 are fixed as follows:

- A1 = “11110110” or 8'hF6
- A2 = “00101000” or 8'h28

OC-12 and OC-48 Data Paths

OC-12 and OC-48 configurations have similar data paths, as seen in Figures 2–138 and 2–139. The only difference is that OC-48 has a 16-bit PLD interface as compared to the 8-bit PLD interface in an OC-12 configuration. The OC-48 configuration employs the byte serializer and deserializer and a byte ordering block to translate 16-bit PLD interfaces into an 8-bit transceiver data path.

Figure 2–138. OC-12 Data Path

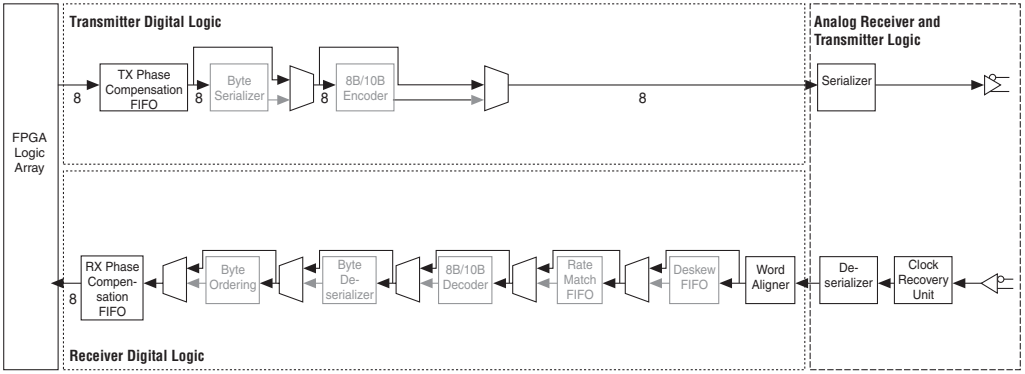
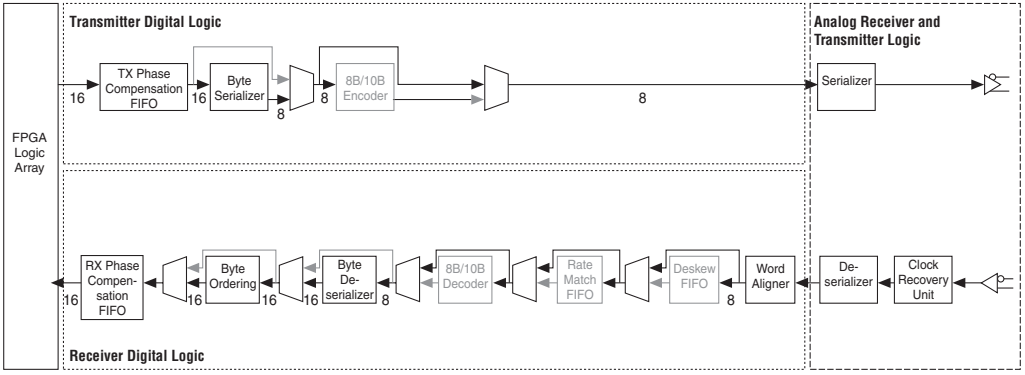


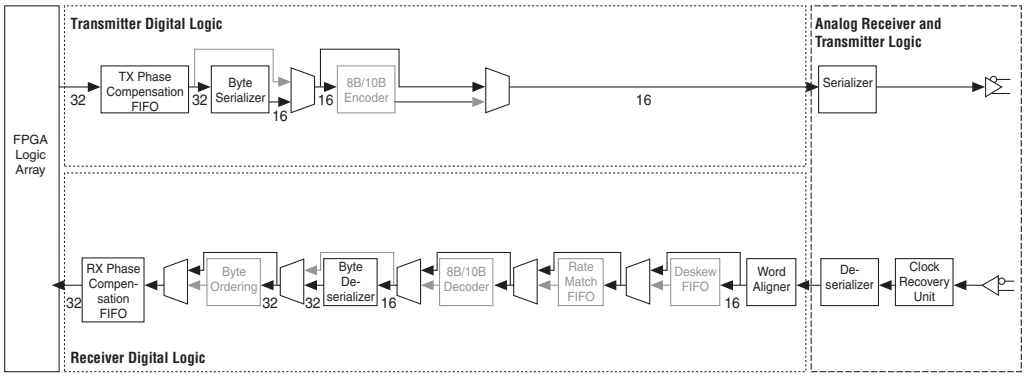
Figure 2–139. OC-48 Data Path



OC-96 Data Path

The OC-96 data path is wider than the OC-12 and OC-48 data paths (refer to Figure 2–140). It has a 32-bit wide PLD interface that is translated to a 16-bit wide transceiver data path by the byte serializer and deserializer. As a result, the OC-96 configuration has a bit serialization factor of 16, unlike OC-12 and OC-48 with bit serialization factors of 8. Also, the OC-96 configuration does not have the byte ordering block in the transceiver data path. If required, you should implement byte ordering logic in the PLD logic array in OC-96 configurations.

Figure 2–140. OC-96 Data Path



SONET/SDH Serial Data Transmission Bit Order

Unlike Ethernet where the least significant bit of the data byte is transferred first, SONET/SDH requires the most significant bit to be transferred first and the least significant bit to be transferred last. To facilitate MSBit to LSBit transfer, you must enable the following options in the MegaWizard:

- Flip Transmitter Input Data Bits (when used in transmit only or duplex mode)
- Flip Receiver Output Data Bits (when used in receive only or duplex mode)

Depending on whether data bytes are transferred MSBit to LSBit or LSBit to MSBit, you must select appropriate word aligner settings in the MegaWizard. Table 2–43 lists correct word aligner settings for each bit transmission order.

OC-12 and OC-48 Word Alignment

SONET/SDH mode uses manual word alignment as described in "Manual SONET/SDH Alignment Mode (Two Consecutive 8-bit Characters (A1A2) or Four Consecutive 8-bit Characters (A1A1A2A2))" on page 2–78.

In OC-12 and OC-48 configurations, you can configure the word aligner to either align to a 16-bit A1A2 pattern or a 32-bit A1A1A2A2 pattern. This is controlled by the `rx_a1a2size` input port to the transceiver. A

LOW level on the rx_a1a2size port configures the word aligner to align to a 16-bit A1A2 pattern and a high level configures it to align to a 32-bit A1A1A2A2 pattern.

You can configure the word aligner to flip the alignment pattern bits programmed in the MegaWizard and compare it them with the incoming data for alignment. This feature offers flexibility to the SONET/SDH system for either an MSBit to LSBit or LSBit to MSBit data transfer. Table 2–43 lists word alignment patterns that you must program in the MegaWizard based on the bit-transmission order and the word aligner bit-flip option.

Table 2–43. Word Aligner Settings

Serial Bit Transmission Order	Word Alignment Bit Flip	Word Alignment Pattern
MSBit to LSBit	On	1111011000101000 (16'hF628)
MSBit to LSBit	Off	0001010001101111 (16'h146F)
LSBit to MSBit	Off	0010100011110110 (16'h28F6)

The behavior of the SONET/SDH word aligner control and status signals along with an operational timing diagram are explained in "Manual SONET/SDH Alignment Mode (Two Consecutive 8-bit Characters (A1A2) or Four Consecutive 8-bit Characters (A1A1A2A2))" on page 2–78.

OC-96 Word Alignment

In OC-96 configuration, the word aligner is only allowed to align to a A1A1A2A2 pattern, so input port rx_a1a2size is unavailable. Barring this difference, the OC-96 word alignment operation is similar to that of the OC-12 and OC-48 configurations.

OC-48 Byte Serializer and Deserializer

The OC-48 transceiver data path includes the byte serializer and deserializer to allow the PLD interface to run at a lower speed. The OC-12 configuration does not use the byte serializer and deserializer blocks.

The byte serializer and deserializer blocks are explained in the sections "Byte Serializer" on page 2–32 and "Byte Deserializer" on page 2–112, respectively. The OC-48 byte serializer converts 16-bit data words from

the PLD logic array and translate the 16-bit data words into two 8-bit data bytes at twice the rate. The OC-48 byte deserializer takes in two consecutive 8-bit data bytes and translates them into a 16-bit data word to the PLD logic array at half the rate.

OC-96 Byte Serializer and Deserializer

The OC-96 byte serializer converts 32-bit data words from the PLD logic array and translates them into two 16-bit data bytes at twice the rate. The OC-48 byte deserializer takes in two consecutive 16-bit data bytes and translates them into a 32-bit data word to the PLD logic array at half the rate.

OC-48 Byte Ordering

Because of byte deserialization, the most significant byte of a word may appear at the `rx_dataout` port along with the least significant byte of the next word.

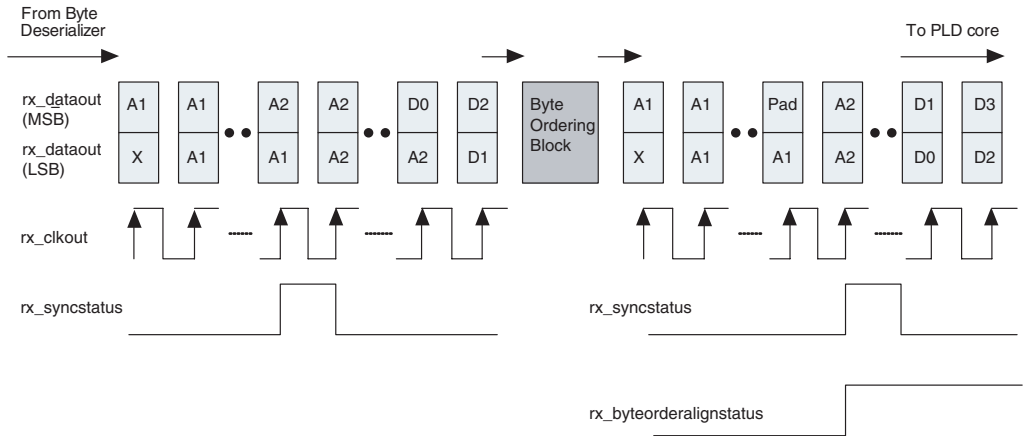
In an OC-48 configuration, the byte ordering block is built into the data path and can be leveraged to perform byte ordering. In an OC-96 configuration, the byte ordering block is unavailable and ordering must be performed in the PLD logic array.

The byte ordering in an OC-48 configuration is automatic as explained in "Word Alignment Based on Byte Ordering" on page 2–114. In automatic mode, the byte ordering block is triggered by the rising edge of the `rx_syncstatus` signal. As soon as the byte ordering block sees the rising edge of the `rx_syncstatus` signal, it compares the least significant byte coming out of the byte deserializer with the A2 byte of the A1A2 alignment pattern. If the least significant byte coming out of the byte deserializer does not match A2 byte set in the MegaWizard, the byte ordering block inserts a pad character as seen in Figure 2–141. Insertion of this pad character enables the byte ordering block to restore the correct byte order. Note that the pad character is defaulted to the A1 byte of the A1A2 alignment pattern.

Once the byte ordering is achieved, the `rx_byteorderalignstatus` signal remains asserted high until `rx_digitalreset` is asserted. The byte ordering within the transceiver is a one-time event after the receiver comes out of `rx_digitalreset`. So, if a byte ordering operation is required, the receiver must go through an `rx_digitalreset` cycle.

If successful byte ordering occurs without successful word alignment, you should assert receive digital reset (`rx_digitalreset`) so that the byte ordering block performs another round of byte ordering (one time after asserting `rx_digitalreset`). This is only required when the byte ordering block picks an incorrect byte order.

Figure 2–141. Byte Ordering Block Operation in OC-48



(OIF) CEI-PHY Interface Mode

The (OIF) CEI PHY Interface mode is intended to support two main protocols:

- Common Electrical I/O (CEI-6G) protocol defined by the Optical Internetworking Forum (OIF) at data rates between 4.976 Gbps and 6.375 Gbps
- Interlaken protocol at data rates between 3.135 Gbps and 6.375 Gbps

Stratix II GX transceivers support a data rate between 3.135 Gbps and 6.375 Gbps in (OIF) CEI PHY Interface Mode.

Figure 2–142 shows the ALT2GXB transceiver data path when configured in this mode.

Figure 2–142. (OIF) CEI-PHY Interface Data Path

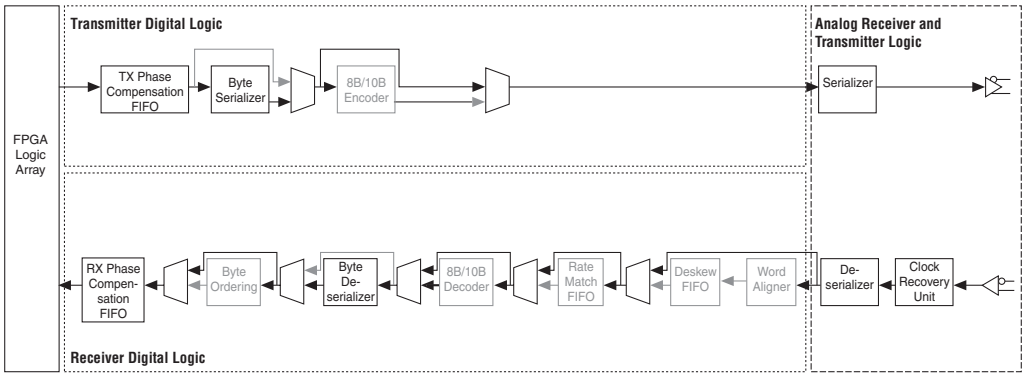


Table 2–44 shows ALT2GXB configurations supported by the Stratix II GX transceivers in (OIF) CEI PHY Interface mode.

Table 2–44. ALT2GXB Configurations in (OIF) CEI PHY Interface Mode

Data Rate (Mbps)	REFCLK Frequency (PLL Multiplication Factor)	Channel Width
3135 < Data Rate ≤5700	Data-Rate/10 (M = 10)*(1)* Data-Rate/20 (M = 10) Data-Rate/40 (M = 20)	32 bit
5800 < Data Rate ≤6375	Data-Rate/10 (M = 10) *(1)* Data-Rate/20 (M = 10)	32 bit

Note to Table 2–44:

(1) Selecting the REFCLK frequency of Data-Rate/10 requires the use of the /2 REFCLK pre-divider; for example, selecting 500 MHz REFCLK frequency for 5000 Mbps data rate.

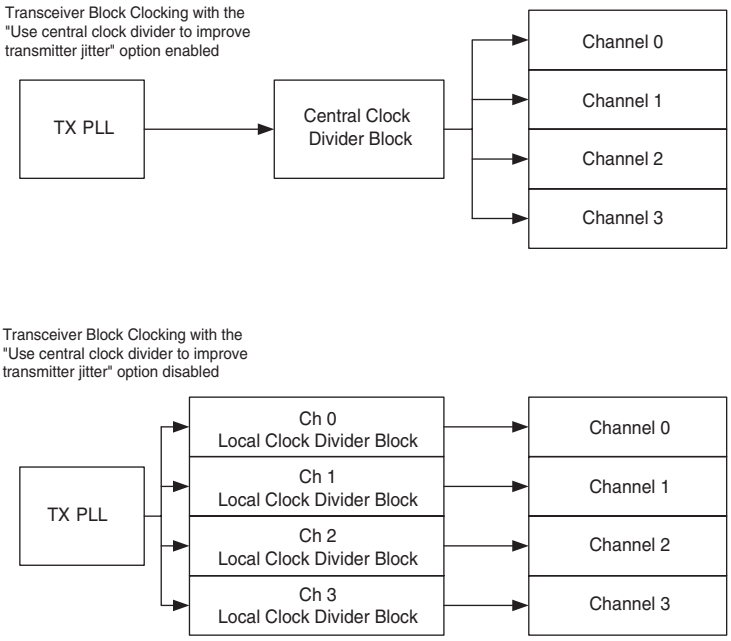
(OIF) CEI PHY Interface Mode Clocking

For improved transmitter jitter performance, the ALT2GXB MegaWizard Plug-In Manager provides an **Use central clock divider to improve transmitter jitter** option. If you select this option, clocks generated by the central clock divider clock all four transceiver channels within the same transceiver block. Otherwise, clocks generated by the local clock divider in each channel clock the respective channel.

Unlike PIPE ×4, XAUI or Basic x4 mode, the transmitter PCS is not bonded in the (OIF) CEI PHY Interface with the low-jitter option selected.

Figure 2–143 shows transceiver clocking in (OIF) CEI PHY Interface mode with and without the improved transmitter jitter option enabled.

Figure 2–143. (OIF) CEI PHY Interface Mode Clocking



Transceiver Placement Limitations with Improved Jitter Clocking Option

If one or more channels in a transceiver block are configured to (OIF) CEI PHY Interface mode with the improved jitter clocking option enabled, the remaining channels in that transceiver block must either be configured in (OIF) CEI PHY Interface mode with this option enabled or must be unused. All used channels within a transceiver block configured in (OIF) CEI PHY Interface mode with improved jitter clocking option enabled must also run at the same data rate.

Figures 2–144 and 2–145 show two examples each of legal and illegal transceiver placements with respect to the improved jitter clocking option in (OIF) CEI PHY Interface mode.

Figure 2–144. Examples of Legal Transceiver Placement in (OIF) CEI PHY Interface Mode

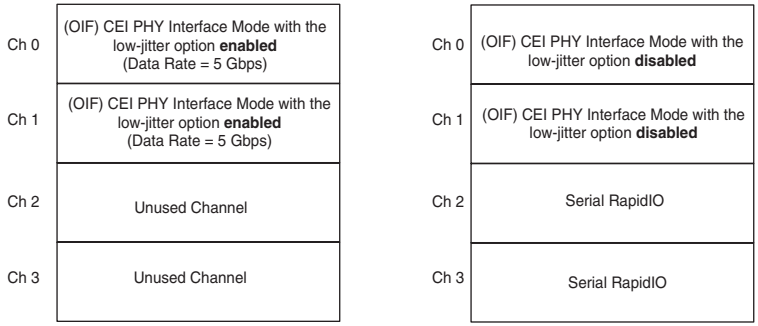
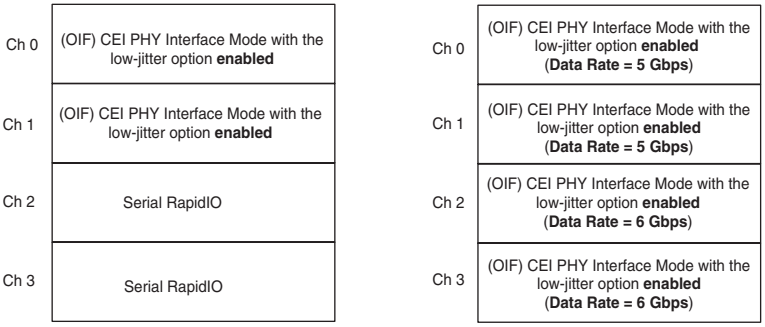


Figure 2–145. Examples of Illegal Transceiver Placement in (OIF) CEI PHY Interface Mode



Serial Digital Interface (SDI) Mode

The Society of Motion Picture and Television Engineers (SMPTE) defines various Serial Digital Interface (SDI) standards for transmission of uncompressed video.

The following three SMPTE standards are popular in video broadcasting applications:

- SMPTE 259M standard—more popularly known as the standard-definition (SD) SDI, is defined to carry video data at 270 Mbps.

- SMPTE 292M standard—more popularly known as the high-definition (HD) SDI, is defined to carry video data at either 1485 Mbps or 1483.5 Mbps.
- SMPTE 424M standard—more popularly known as the third-generation (3G) SDI, is defined to carry video data at either 2970 Mbps or 2967 Mbps.

You can configure Stratix II GX transceivers in HD-SDI or 3G-SDI configuration using the ALT2GXB MegaWizard Plug-In Manager.

Figure 2–146 shows the ALT2GXB transceiver data path in SDI mode.

Figure 2–146. SDI Mode Data Path

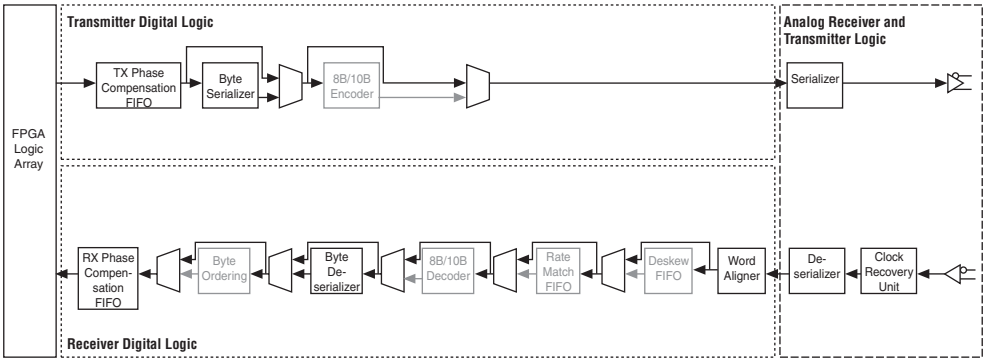


Table 2–45 shows ALT2GXB configurations supported by the Stratix II GX transceivers in SDI mode.

Table 2–45. ALT2GXB Configurations in SDI Mode

Configuration	Data Rate (Mbps)	REFCLK Frequencies (MHz)	Channel Width
HD	1485	74.25, 148.5	10 bit, 20 bit
	1483.5	74.175, 148.35	10 bit, 20 bit
3G	2970	148.5, 297	Only 20-bit interface allowed in 3G
	2967	148.35, 296.7	Only 20-bit interface allowed in 3G

Transmitter Data Path

In the 10-bit channel width SDI configuration, the transmitter data path is made up of the transmitter phase compensation FIFO and the 10:1 serializer. In the 20-bit channel width SDI configuration, the transmitter data path also includes the byte serializer.

In SDI mode, the transmitter is purely a parallel-to-serial converter. SDI transmitter functions, such as scrambling and cyclic redundancy check (CRC) code generation, must be implemented in the FPGA logic array.

Receiver Data Path

In the 10-bit channel width SDI configuration, the receiver data path comprises of the clock recovery unit (CRU), the 1:10 deserializer, the word aligner in bit-slip mode, and the receiver phase compensation FIFO. In the 20-bit channel width SDI configuration, the receiver data path also includes the byte deserializer.

SDI receiver functions, such as de-scrambling, framing, and CRC checker, must be implemented in the FPGA logic array.

Receiver Word Alignment/Framing

In SDI systems, since the word alignment and framing happens after de-scrambling, the word aligner in the receiver data path is not useful. Altera recommends driving the ALT2GXB `rx_bitslip` signal low to avoid the word aligner from inserting bits in the received data stream.

Altera offers SDI MegaCore function that can be configured at SD-SDI, HD-SDI, and 3G-SDI data rates. The SDI MegaCore function implements system-level functions like scrambling and de-scrambling and CRC generation and checking. It also offers the capability of configuring the three SDI data rates (SD, HD, and 3G) dynamically on the same transceiver channel. For more details, refer the *SDI MegaCore Function User Guide*.

Serial RapidIO Mode

The RapidIO™ Trade Association defines a high-performance, packet-switched interconnect standard to pass data and control information between microprocessors, digital signal, communications, and network processors, system memories, and peripheral devices.

Serial RapidIO physical layer specification defines three line rates:

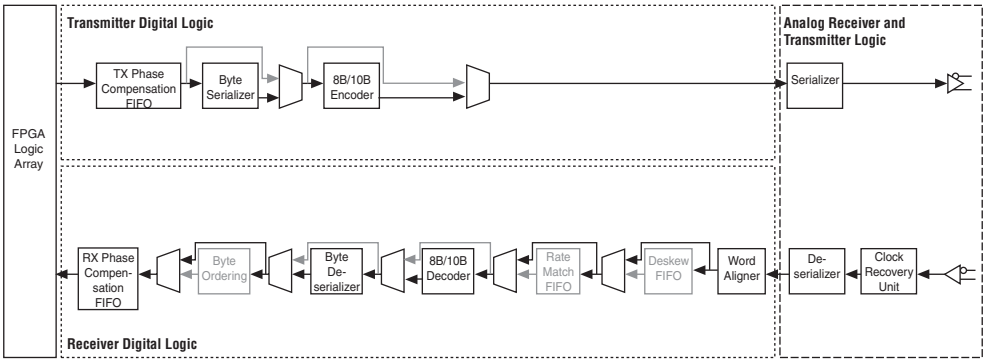
- 1.25 Gbps
- 2.5 Gbps
- 3.125 Gbps

It also defines two link widths—single-lane (1×) and bonded four-lane (4×) at each line rate.

Stratix II GX transceivers support only single-lane (1×) configuration at all three line rates. Four 1× channels configured in Serial RapidIO mode can be instantiated to achieve a 4× Serial RapidIO link. The four transmitter channels in this 4× Serial RapidIO link are not bonded. The four receiver channels in this 4× Serial RapidIO link do not have lane alignment or deskew capability.

Figure 2–147 shows the ALT2GXB transceiver data path when configured in Serial RapidIO mode.

Figure 2–147. Serial RapidIO Mode



Stratix II GX transceivers, when configured in Serial RapidIO functional mode, provide the following PCS and PMA functions:

- 8B/10B encoding/decoding
- Word alignment
- Lane Synchronization State Machine
- Clock recovery from the encoded data
- Serialization/deserialization

Stratix II GX transceivers do not have built-in support for other PCS functions; for example, clock frequency compensation between upstream transmitter clock and local receiver clock (rate matcher), pseudo-random idle sequence generation, and lane alignment in 4× mode. Depending on your system requirements, you must implement these functions in the logic array or external circuits.

Synchronization State Machine

In Serial RapidIO mode, the ALT2GXB MegaWizard Plug-In Manager defaults the word alignment pattern to K28.5. The word aligner has a Synchronization State Machine that handles the receiver lane synchronization.

The ALT2GXB MegaWizard Plug-In Manager automatically defaults the synchronization state machine to indicate synchronization when the receiver receives 127 K28.5 (10'b0101111100 or 10'b1010000011) synchronization code groups without receiving an intermediate invalid code group. Once synchronized, the state machine indicates loss of synchronization when it detects three invalid code groups separated by less than 255 valid code groups or when it is reset.

Receiver synchronization is indicated on the `rx_syncstatus` port of each channel. A high on the `rx_syncstatus` port indicates that the lane is synchronized and a low indicates that it has fallen out of synchronization.

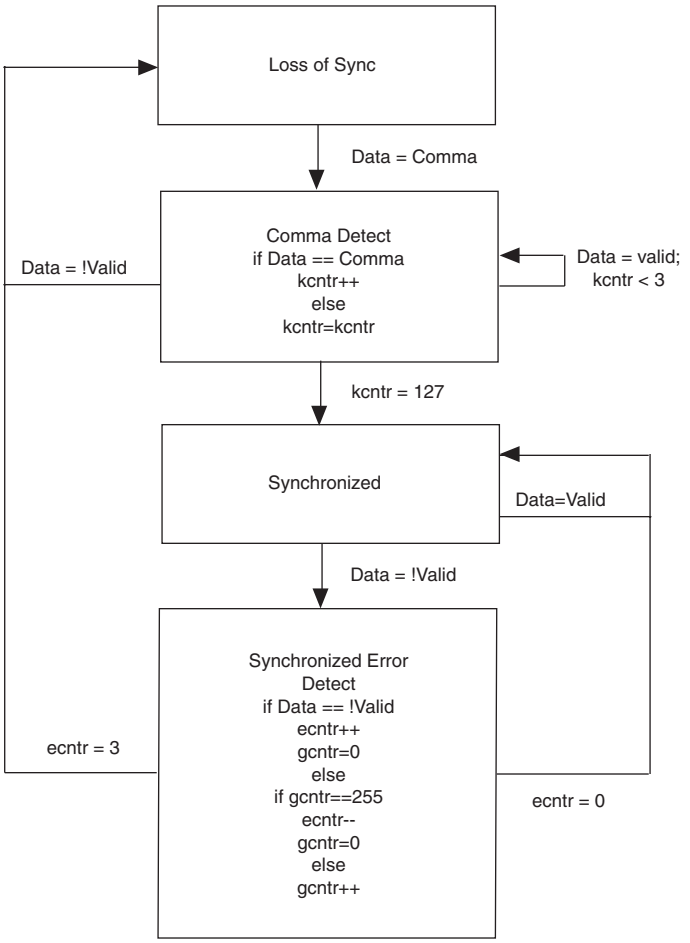
Table 2–46 lists the ALT2GXB synchronization state machine parameters when configured in Serial RapidIO mode.

Table 2–46. Synchronization State Machine Parameters in Serial RapidIO Mode

Parameters	Number
Number of valid K28.5 code groups received to achieve synchronization.	127
Number of errors received to lose synchronization.	3
Number of continuous good code groups received to reduce the error count by one.	255

Figure 2–148 gives a conceptual view of the synchronization state machine implemented in Serial RapidIO functional mode.

Figure 2–148. Synchronization State Machine in Serial RapidIO Mode



CPRI Mode

The common public radio interface (CPRI) specification defines a radio base station interface standard between the radio equipment control (REC) and the radio equipment (RE).

CPRI Specification V2.1 defines the following three line rates for deployment flexibility:

- CPRI line bit rate option 1: 614.4 Mbps
- CPRI line bit rate option 2: 1228.8 Mbps (2 x 614.4 Mbps)
- CPRI line bit rate option 3: 2457.6 Mbps (4 x 614.4 Mbps)

Stratix II GX transceivers support all three line bit rate options.

Figure 2–149 shows the ALT2GXB transceiver data path when configured in CPRI mode.

Figure 2–149. ALT2GXB Transceiver Data Path in CPRI Mode

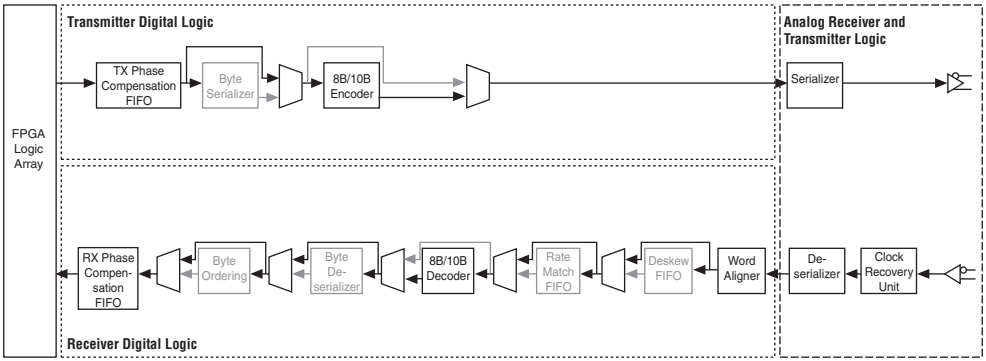


Table 2–47 shows ALT2GXB configurations supported by the Stratix II GX transceivers in CPRI mode.

Table 2–47. ALT2GXB Configurations in CPRI Mode

Data Rate (Mbps)	REFCLK Frequencies (MHz)	Channel Width
614	61.4, 76.675, 122.8, 153.5	8 bit
1228	61.4, 76.675, 122.8, 153.5, 245.6, 307	8 bit
2456	61.4, 76.675, 122.8, 153.5, 245.6, 307, 491.2, 614	8 bit

Transmitter Data Path

The CPRI transmitter data path includes the transmitter phase compensation FIFO, the 8B/10B encoder, and the 10:1 serializer. Layer 1 functions like Hyperframe framing that includes interleaving IQ data, sync data, L1 inband protocol data, and so forth, that must be performed in the FPGA logic array or external circuitry.

Receiver Data Path

The receiver data path includes the clock recovery unit (CRU), 1:10 deserializer, synchronization-state-machine-based word aligner, 8B/10B decoder, and receiver phase compensation FIFO.

The synchronization-state-machine-based word aligner is programmable. You can select the number of bad code groups detected to fall out of synchronization, the number of valid synchronization code groups to acquire synchronization, and the number of good code groups received to reduce the error count by 1.

You can use this programmability to implement the Loss of Signal (LOS), Loss of Frame Synchronization (LOF), and the Remote Alarm Indication (RAI) features in the FPGA logic array. The synchronization status is reported on the `rx_syncstatus` port from the ALT2GXB. The `rx_syncstatus` signal is driven high when the programmed conditions for synchronization state machine are met. Otherwise, it is driven low to indicate loss of synchronization.

Link Delay Accuracy

Requirement R-19 in CPRI Specification V2.1 requires the CPRI link delay accuracy (excluding the transmission medium delay) to be ± Tc/32, where Tc is the length of a basic frame (260.42 ns). This requirement mandates the total uncertainty in CPRI link latency to be less than ~16.3 ns.

To meet the strict CPRI link delay accuracy requirements, the Quartus II software automatically adjusts the routing delays on the transmitter and receiver phase compensation FIFO clocks. As a result of this delay adjustment, the transmitter and receiver phase compensation FIFO latency becomes constant.

The Quartus II software performs the delay adjustment to minimize the uncertainty in link latency only in CPRI mode.

Table 2–48 shows the uncertainty in the CPRI transceiver data path after delay adjustment.

Table 2–48. Uncertainty in CPRI Transceiver Datapath Latency With Delay Adjustment *Note (1)*

Data Rate (Mbps)	Receiver Deserializer Uncertainty (Parallel Clock Cycles)	Receiver Phase Comp FIFO Uncertainty (Parallel Clock Cycles)	Transmitter Phase Comp FIFO Uncertainty (Parallel Clock Cycles)	Parallel Clock Period (ns)	Total Uncertainty (ns)
614	0.9	0	0	16.3	14.67
1228	0.9	0	0	8.15	7.34

Note to Table 2–48:

(1) The delay adjustment is not made for CPRI 2456 Mbps line rate configuration since it meets the 16.3 ns link delay requirement without these adjustments.

Table 2–48 shows that the CPRI link delay accuracy requirements are met within the transceiver data path.

Transceiver Limitations in CPRI Mode

To meet the CPRI link delay accuracy requirements, the Quartus II software adjusts delays on the clock routes from the `tx_clkout` and `rx_clkout` ports to the write and read ports of the transmitter and receiver phase compensation FIFOs, respectively, for each transceiver channel. Due to this requirement, the Quartus II software only allows the `tx_clkout` signal from each channel to clock the write port of its transmitter phase compensation FIFO. Similarly, it allows the `rx_clkout` signal from each channel to clock the read port of its receiver phase compensation FIFO. If your design requires dynamic reconfiguration between CPRI mode and other modes, each channel's phase compensation FIFOs must be clocked by its own `tx_clkout` and `rx_clkout` for other modes as well. The default transceiver configuration used to create the programming file (**.sof** or **.pof**) must be CPRI for the delay algorithm to take effect.

In CPRI mode, you cannot group the `tx_coreclk` and/or `rx_coreclk` ports of multiple channels and drive them using a common clock driver using the shared clock or 0 PPM clock group assignments. Each CPRI channel will utilize at least two global and/or regional clock resources. Since a maximum of 32 global and/or regional clock resources are available for transceivers in the Stratix II GX device, the clock resource availability governs the maximum number of CPRI channels that you can instantiate per device.

The maximum number of CPRI channels per device also depends on the number of transceiver channels available in that device and the LRIO clock resource limitations.

CPRI mode is supported only in C3, C4, and I4 speed-grade devices.

PLD-Transceiver Interface Clock Duty Cycle

Due to delay adjustments made to the `tx_clkout` and `rx_clkout` routes in the FPGA clock network, the worst case duty cycle on these PLD-transceiver interface clocks can be 60-40%. If these clocks are used to clock FPGA logic array, you must set proper timing analyzer assignments to account for the 60-40% duty cycle on these clocks.

Figure 2–150 shows how to set 60-40% duty cycle constraints in TimeQuest Timing Analyzer for a CPRI 614.4 Mbps line rate configuration. In the TimeQuest Analyzer window, selecting the **Report Clocks** option lists all the `tx_clkout` and `rx_clkout` clocks in your design. Select all clocks and adjust the falling edge timing appropriately for 60-40% duty cycle.

Figure 2–150. Setting Duty Cycle in TimeQuest Timing Analyzer

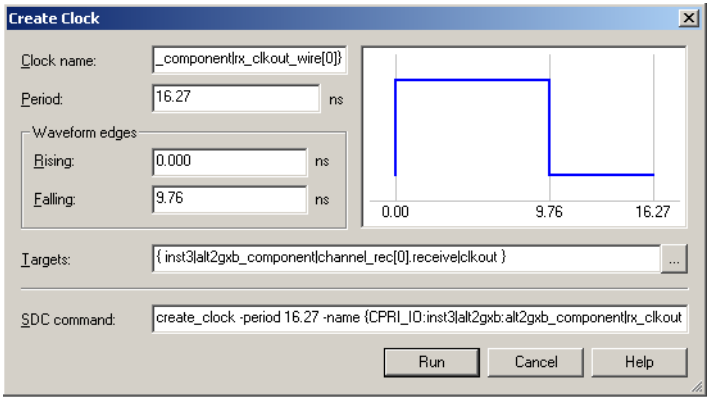
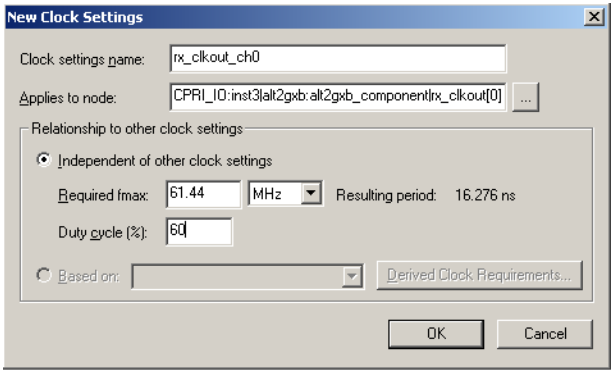


Figure 2–151 shows how to set 60-40% duty cycle constraints in Classic Timing Analyzer. To set this option, go to the Assignments menu and select **Settings**, then select **Timing Analysis Settings** and **Classic Timing Analyzer Settings**, and then click on the **Individual Clocks** tab. In the **Individual Clocks** window, select **New**. In the **New Clock Settings** window, browse to all `rx_clkout` and `tx_clkout` nodes and assign **60%** in the **Duty Cycle** option for each of these phase compensation FIFO clocks.

Figure 2–151. Setting Duty Cycle in Classic Timing Analyzer



If the PLD-transceiver interface clocks are fed to an off-chip PLL (for example, VCXO-based PLL for de-jittering purposes), you must make sure that the PLL can tolerate the 60-40% duty cycle on its input reference clock.

Loopback Modes

There are several loopback modes available on the Stratix II GX transceiver block that allow you to isolate portions of the circuit. All paths are designed to run up to full speed. The available loopback paths are:

- Serial loopback available in all functional modes except PCI Express (PIPE)
- Reverse serial loopback available in Basic mode with 8B/10B
- PCI Express PIPE reverse parallel loopback available in PCI Express protocol
- Reverse serial pre-CDR loopback available in Basic mode with 8B/10BReverse serial loopback available in Basic mode with 8B/10B
- Parallel loopback available in Basic mode for BIST testing only

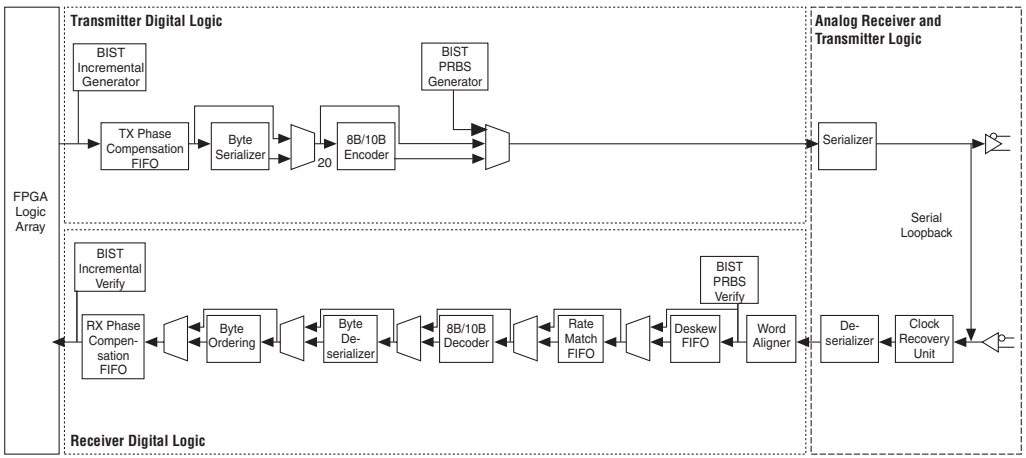
Serial Loopback

Figure 2–152 shows the data path for serial loopback. A data stream is fed to the transmitter from the FPGA logic array and has the option of utilizing all the blocks in the transmitter. The data, in serial form, then traverses from the transmitter to the receiver. The serial data is the data that is transmitted from the Stratix II GX device. Once the data enters the receiver in serial form, it can utilize any of the receiver blocks and is then fed into the FPGA logic array.

Use the rx_seriallpbken port to dynamically enable serial loopback on a channel by channel basis. When rx_seriallpbken is high, all blocks that are active when the signal is low are still active. When the serial loopback is enabled, the tx_dataout port is still active and drives out the output pins.

Serial loopback is often used to check the entire path of the transceiver. The data is retimed through different clock domains and an alignment pattern is still necessary for the word aligner.

Figure 2–152. Stratix II GX Block in Serial Loopback Mode

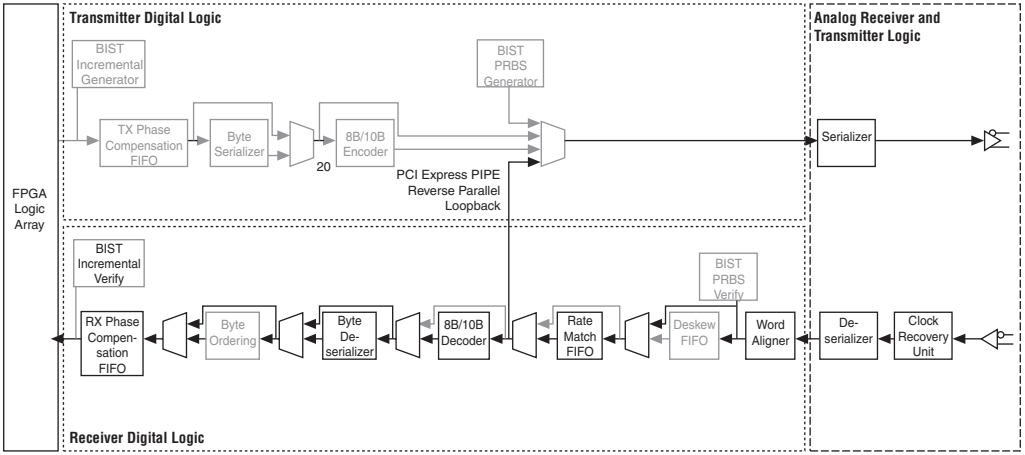


PCI Express PIPE Reverse Parallel Loopback

Figure 2–153 shows the data path for the PCI Express PIPE reverse parallel loopback. This data path is not flexible because it must be compliant with the PCI Express PIPE specification. The data comes in from the rx_datain ports. The receiver uses the CRU, deserializer, word aligner, and rate matching FIFO buffer, loops back to the transmitter serializer, and then goes out the transmitter tx_dataout ports. The data also goes to the PLD fabric on the receiver side to the tx_dataout port. The deskew FIFO buffer is not enabled in this loopback mode. This loopback mode is optionally controlled dynamically through the tx_detectrxloopback port.

This is the only loopback allowed in the PIPE mode.

Figure 2–153. Stratix II GX Block in PCI Express PIPE Reverse Parallel Loopback Mode



Reverse Serial Loopback

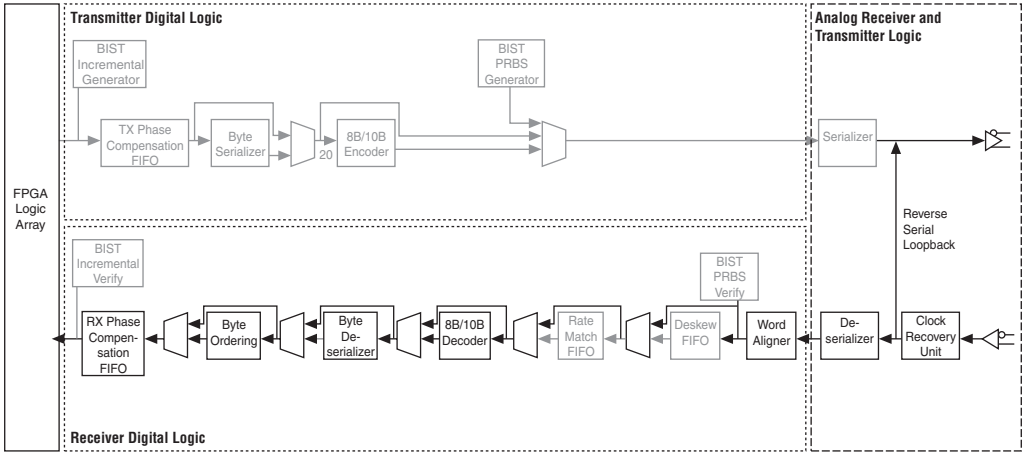
Reverse serial loopback is a subprotocol in Basic mode. It requires 8B/10B, and the word aligner pattern of K28.5. No dynamic pin control is available to select or deselect reverse serial loopback. The active block of the transmitter is only the buffer. The data sent to the receiver is retimed with the recovered clock and sent out to the transmitter.

The data path for reverse serial loopback is shown in Figure 2–154. Data comes in from the `rx_datain` ports in the receiver. The data is then fed through the CDR block in serial form directly to the `tx_dataout` ports in the transmitter block.

You can enable reverse serial loopback for all channels through the MegaWizard. Any pre-emphasis setting on the transmitter buffer is ignored in reverse serial loopback. The data flows through the active blocks of the receiver and into the logic array.

Reverse serial loopback is often implemented when using a bit error rate tester (BERT).

Figure 2–154. Stratix II GX Block in Reverse Serial Loopback Mode

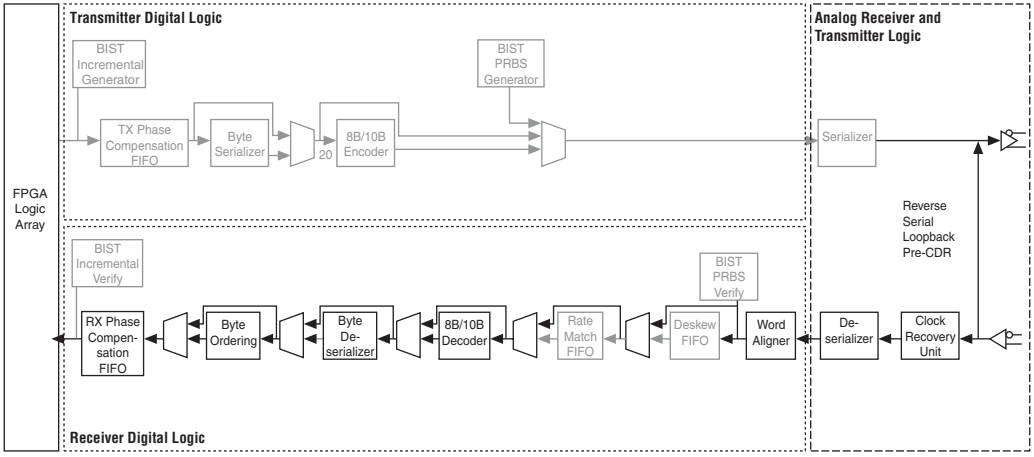


Reverse Serial Pre-CDR Loopback

The reverse serial pre-CDR loopback uses the analog portion of the transceiver. An external source (pattern generator or transceiver) generates the source data. The high-speed serial source data arrives at the high-speed differential receiver input buffer, loops back before the CRU unit, and is transmitted though the high-speed differential transmitter output buffer. It is for test or verification use only to verify the signal being received after the gain and equalization improvements of the input buffer. The signal at the output is not exactly what is received since the signal goes through the output buffer and the VOD is changed to the VOD setting level. The pre-emphasis settings have no effect.

Figure 2–155 show the Stratix II GX block in reverse serial pre-CDR loopback mode.

Figure 2–155. Stratix II GX Block in Reverse Serial Pre-CDR Loopback Mode



Parallel Loopback

The data path for parallel loopback is shown in Figure 2–156. The forward parallel loopback allows a test flow check of the PCS using either the built-in test incremental pattern. This is available only as a subprotocol in Basic Double-Width mode.

When using the BIST incremental parallel loopback, the deskew and the rate matching FIFO buffer are not available. The 8B/10B encoder and decoder are used. No dynamic control pin is available to enable or disable the loopback. Test result pins, `rx_bistdone` and `rx_bisterr`, are available in this loopback mode.

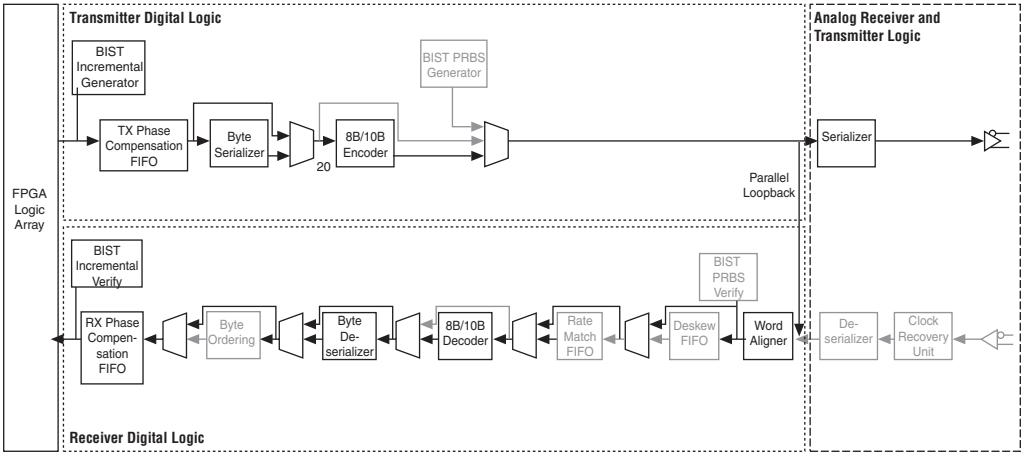
When using parallel loopback, the `tx_dataout` ports are active and the differential output voltage on the `tx_dataout` ports is based on the V_{OD} settings.

Table 2–49 shows the available BIST patterns in double-width mode.

Table 2–49. Available BIST Patterns in Double-Width Mode

PATTERN	Word Aligner Alignment Pattern	Byte Order Align Pattern	Description	Double-Width Mode	
				16 Bit	20 Bit
Incremental with 8B/10B	20'h16E83	N/A	All 8B/10B valid code groups		✓

Figure 2–156. Stratix II GX Block in Parallel Loopback Mode



Incremental Pattern Generator

The incremental data generator sweeps through all the valid 8B/10B data and control characters. This mode is only available in Basic mode with the BIST/parallel loopback subprotocol in the Quartus II software. You can also enable the incremental BIST verifier to perform a quick verification of the 8B/10B encoder/decoder paths.

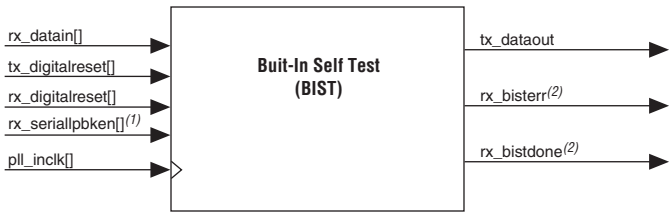
In incremental mode, the BIST generator sends out the data pattern in the following sequence: K28.5 (comma), K27.7 (start of frame, SOF), Data (00-FF incremental), K28.0, K28.1, K28.2, K28.3, K28.4, K28.6, K28.7, K23.7, K30.7, K29.7 (end of frame, EOF), and then repeats. You must enable the 8B/10B encoder for proper operation. No dynamic control pin is available to enable or disable the loopback. Test result pins are

rx_bistdone and rx_bisterr. The rx_bistdone signal goes high at the end of the sequence. If the verifier detects an error before it is finished, rx_bisterr pulses high as long as the data is in error.

Built-In Self-Test Modes

Besides the regular data flow blocks, each transceiver channel contains an embedded built-in self test (BIST) generator and corresponding verifier block that you can use for quick device and setup verification (refer to Figure 2–157). The generators reside in the transmitter block and the verifier in the receiver block. The generators can generate PRBS and incremental patterns. The incremental pattern is available only in Parallel loopback mode. The verifiers are only available for these data patterns. The BIST blocks operate differently when in the single-width mode and the double-width mode. The BIST modes are only available as subprotocols under Basic mode.

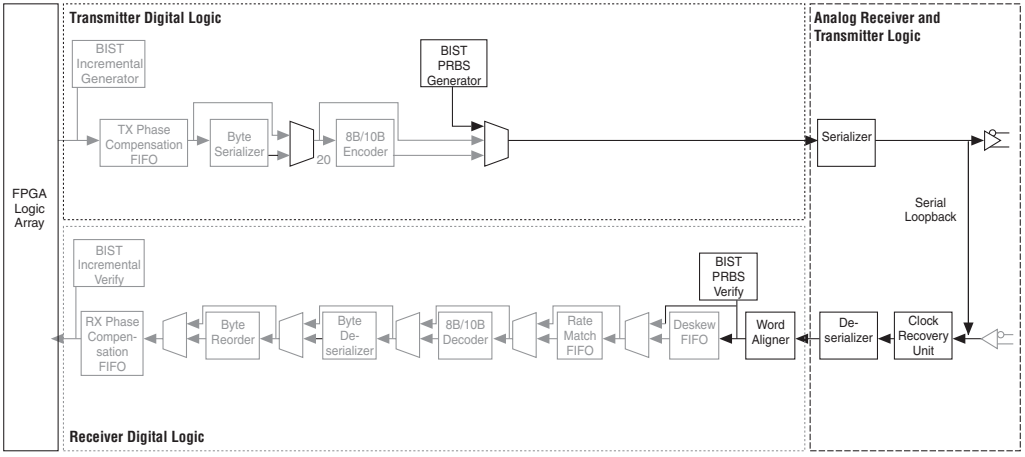
Figure 2–157. Built-In Self Test Mode



Notes to Figure 2–157:
(1) rx_seriallpbken[] is required in PRBS.
(2) rx_bisterr[] and rx_bistdone[] are only available in PRBS and BIST modes.

Figure 2–158 shows the PRBS blocks with loopback used in the transceiver channel.

Figure 2–158. PRBS Blocks With Loopback in Transceiver Channel



BIST in Single-Width Mode

Single-width mode supports PRBS10 pattern generation and verification. PRBS10 in Basic mode is supported with or without serial loopback

The PRBS10 pattern is only available when the SERDES factor is 10 bits.

Table 2–50 shows the BIST patterns for single-width mode.

Table 2–50. Available BIST Patterns in Single-Width Mode

Pattern	Word Aligner Alignment Pattern	Byte Order Align Pattern	Description	Single-Width Mode	
				8 Bit	10 Bit
PRBS10	10'h3FF	N/A	$X^{10} + X^7 + 1$		✓

PRBS10

Pseudo-Random Bit Sequences (PRBS) are commonly used in systems to verify the integrity and robustness of the data transmission paths. When the SERDES factor is 10, use the PRBS10 pattern. The PRBS generator yields 2^10-1 unique patterns. You can use PRBS with or without serial loopback. In PRBS/ serial loopback mode, the rx_seriallpbken signal is available. In the PRBS/no loopback mode, this control signal is not available.

You enable PRBS mode in the Quartus II ALT2GXB MegaWizard Plug-In Manager. PRBS10 does not use the 8B/10B encoder and decoder. The 8B/10B encoder and decoder are bypassed automatically in the PRBS mode.

The advantage of using a PRBS data stream is that the randomness yields an environment that stresses the transmission medium. In the data stream, you can observe both random jitter and deterministic jitter using a time interval analyzer, bit error rate tester, or oscilloscope.

The PRBS verifier can provide a quick check through the non-8B/10B path of the transceiver block. The PRBS verifier is active once the receiver channel is synchronized. Set the alignment pattern to 10'h3FF for the 10-bit SERDES modes.

The verifier stops checking the patterns after receiving all the PRBS patterns (1023 patterns for 10-bit mode). The `rx_bistdone` signal goes high, indicating that the verifier has completed. If the verifier detects an error before it is finished, `rx_bisterr` pulses high for the time the data is incorrect. Use the `rx_digitalreset` signal to re-start the PRBS verification.

The 8B/10B encoder is enabled, so the data stream is DC balanced. 8B/10B encoding guarantees a run length of less than 5 UI, which yields a less stressful pattern versus the PRBS data. However, since the PRBS generator bypasses the 8B/10B paths, the incremental BIST can test this path.

BIST in Double-Width Mode

Double-width mode supports only PRBS7 pattern generation and verification.

The PRBS7 pattern is only available when the SERDES factor is 20 bits.

Table 2–51 shows the BIST patterns for double-width mode.

Table 2–51. Available BIST Patterns in Double-Width Mode

PATTERN	Word Aligner Alignment Pattern	Byte Order Align Pattern	Description	Double-Width Mode	
				16 Bit	20 Bit
PRBS7	20'h43040	N/A	$X^7 + X^6 + 1$		✓

PRBS7

Pseudo-Random Bit Sequences (PRBS) are commonly used in systems to verify the integrity and robustness of the data transmission paths. The PRBS7 generator generates 2^7-1 unique patterns. PRBS can be used with or without serial loopback. In PRBS/ serial loopback mode, the `rx_seriallpbken` signal is available. In the PRBS/no loopback mode, this control signal is not available.

You enable PRBS mode in the MegaWizard. PRBS7 does not use the 8B/10B encoder and decoder. The 8B/10B encoder and decoder are bypassed automatically in the PRBS mode.

The advantage of using a PRBS data stream is that the randomness yields an environment that stresses the transmission medium. In the data stream you can observe both random jitter and deterministic jitter using a time interval analyzer, bit error rate tester, or oscilloscope.

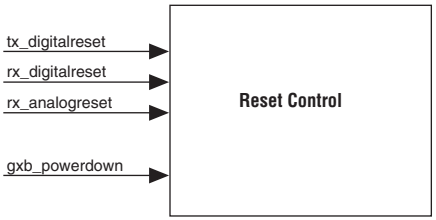
The PRBS verifier provides a quick check through the non-8B/10B path of the transceiver block. The PRBS verifier is active once the receiver channel is synchronized. Set the alignment pattern to 20'h43040 for the 20-bit SERDES modes. The PRBS verifier prevents the word aligner from aligning to a new pattern after the first five successfully verified words.

The verifier stops checking the patterns after receiving all the PRBS patterns (127 patterns for PRBS7). The `rx_bistdone` signal goes high, indicating that the verifier has completed. If the verifier detects an error before it is finished, `rx_bisterr` pulses high for as long as the data is in error. Use the `rx_digitalreset` signal to re-start the PRBS verification.

Reset Control and Power Down

Stratix II GX transceivers offer multiple reset signals to control separate ports of the transceiver channels and blocks (Figure 2–159). You can set each unused channel to a power-down mode to reduce power consumption.

Figure 2–159. Reset Control and Power Down



User Reset and Enable Signals

Each transceiver block and each channel in the transceiver block of the Stratix II GX device has individual reset signals to reset the digital and analog portions of the channel. The analog resets are power-down signals, which require a longer pulse width for the circuits to power down. The `tx_digitalreset`, `rx_digitalreset`, and `rx_analogreset` signals affect the channels individually. The `gxb_powerdown` signal affects the entire transceiver block.

All the reset and enable signals are not required. If not used, the signals are defaulted to not reset for all reset signals and enabled for the PLL enable signal. All reset and enable signals are asynchronous.

- `tx_digitalreset`. The `tx_digitalreset` signal resets all digital logic in the transmitter, including the XAUI transmit state machine, the BIST-PRBS generator, and the BIST pattern generator. This signal operates independently from the other reset signals. The minimum pulse width is two parallel cycles.
- `rx_digitalreset`. The `rx_digitalreset` signal resets all digital logic in the receiver, including the XAUI and GIGE receiver state machine, the XAUI channel alignment state machine, the BIST-PRBS verifier, and the BIST-incremental verifier. This signal operates independently from the other reset signals. The minimum pulse width is two parallel cycles.
- `rx_analogreset`. The `rx_analogreset` signal resets part of the analog portion of the receiver CDR. This signal operates independently from the other reset signals. The minimum pulse width is two parallel clock cycles.

- gxb_powerdown. The gxb_powerdown signal powers down the entire transceiver block. All digital and analog circuits are also reset. This signal operates independently from the other reset signals. The minimum pulse width for gxb_powerdown signal is 100 ns

Synchronization is performed after any reset condition. You must determine when the data is valid after reset (for example, by using the rx_syncstatus signal). Table 2–52 shows the blocks affected by each reset and power-down signal.

Table 2–52. Blocks Affected by Reset and Power-Down Signals (Part 1 of 2)

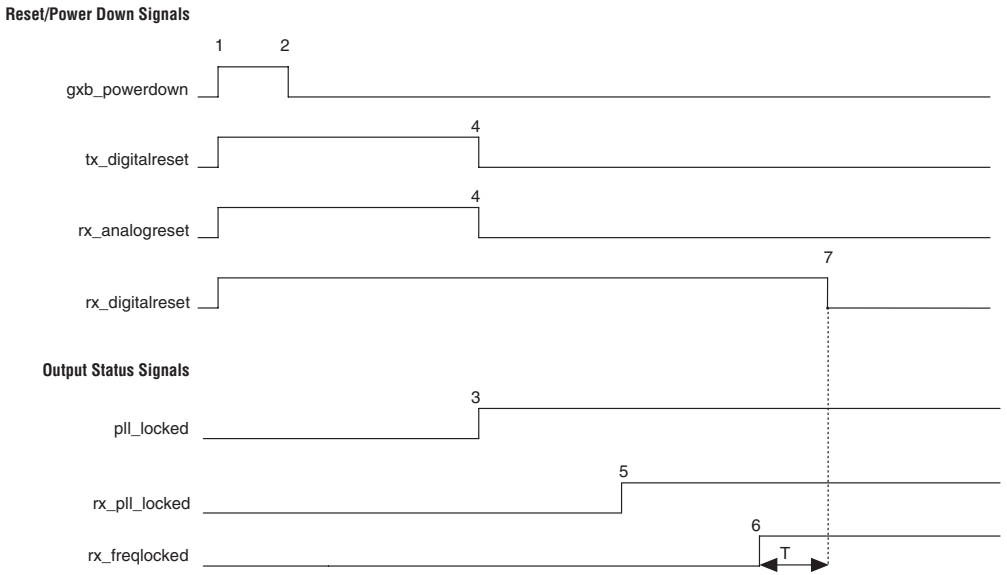
Transceiver Blocks	rx_digitalreset	rx_analogreset	tx_digitalreset	gxb_powerdown
Transmitter phase compensation FIFO buffer and byte serializer			✓	✓
Transmitter 8B/10B encoder			✓	✓
Transmitter serializer				✓
Transmitter analog circuits				✓
Transmitter PLLs				✓
Transmitter XAUI state machine			✓	✓
Transmitter analog circuits				✓
BIST generators			✓	✓
Receiver deserializer				✓
Receiver word aligner	✓			✓
Receiver deskew FIFO buffer	✓			✓
Receiver rate matcher	✓			✓
Receiver 8B/10B encoder	✓			✓
Receiver phase compensation FIFO buffer and byte deserializer	✓			✓
Receiver PLL and CRU		✓		✓

Table 2–52. Blocks Affected by Reset and Power-Down Signals (Part 2 of 2)

Transceiver Blocks	rx_digitalreset	rx_analogreset	tx_digitalreset	gxb_powerdown
Receiver XAUI state machine	✓			✓
Receiver byte ordering block	✓			✓
BIST verifiers	✓			✓
Receiver analog circuits				✓

Figure 2–160 shows a sample reset cycle.

Figure 2–160. Reset Power Signal Timing Waveform



Notes to Figure 2–160:
(1) `tx_digitalreset` is valid in transmitter only and duplex configuration.
(2) `rx_analogreset` and `rx_digitalreset` are valid in receiver only and duplex configuration.

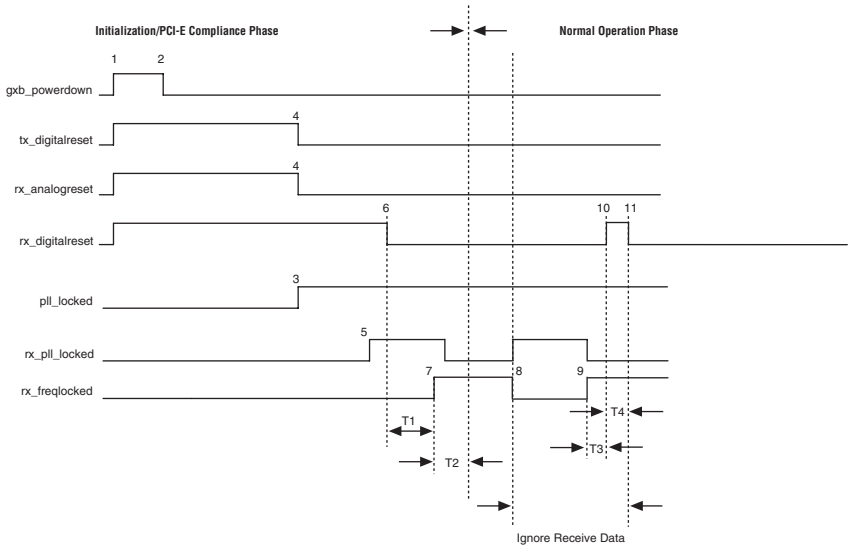
The minimum pulse width for the `gxb_powerdown` port (between time marker 1 and 2) is 100 ns. The `tx_digitalreset` and `rx_analogreset` signals can be deasserted after the driving PLL asserts its associated `pll_locked` signal. The `rx_digitalreset` signal can be de-asserted 4us after the `rx_freqlocklocked` signal goes high (time between markers 6 and 7).

In a transmitter only configuration, only the pll_locked, gxb_powerdown, and tx_digitalreset signals are used. In a receiver only configuration, only the rx_analogreset, rx_digitalreset, and rx_freqlocked signals are used.

Reset Sequence for PIPE Mode

The reset sequence used for the other modes looks for rx_freqlocked signal to deassert rx_digitalreset. In PIPE mode, the rx_freqlocked signal does not go high during the PCI-E compliance testing phase because of receiving Electrical Idle. Figure 2–161 shows the reset sequence for PIPE mode.

Figure 2–161. PIPE Mode Reset Sequence



Initialization and PCI-E Compliance Phase

After the device is powered up, any PCI-E compliant device performs compliance testing. During this phase, all the transceiver reset signals (gxb_powerdown, tx_digitalreset, rx_analogreset, and rx_digitalreset) are asserted.

The minimum time period between markers 1 and 2 for the gxb_powerdown signal is 100 ns (Figure 2–161).

The tx_digitalreset and rx_analogreset signals can be deasserted after the pll_locked signal goes high. The reset controller should deassert the rx_digitalreset when the rx_pll_locked signal goes high.

The parallel data sent to the PLD logic array in the receive side may not be valid until 4 us (T2) after rx_freqlocked goes high.

Normal Operation Phase

During normal operations, the receive data is valid and the rx_freqlocked signal is high. In this situation, when rx_freqlocked is deasserted, (marker 8 in Figure 2–161), the reset controller should wait for the rx_freqlocked to go high again and assert rx_digitalreset (marker 10 in Figure 2–161) for two parallel receive clock cycles.

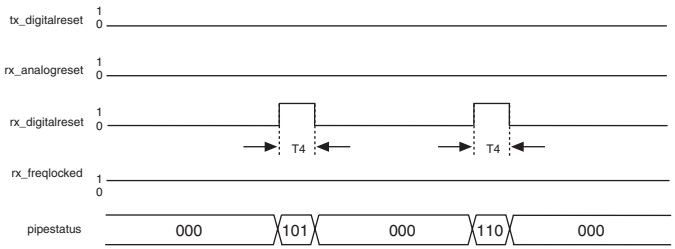
The data from the gigabit transceiver block is not valid between the time when rx_freqlocked goes low until rx_digitalreset is deasserted. The PLD logic should ignore the data during this time period (the time period between markers 8 and 11 in Figure 2–161).

Minimum T1 period is 100 ns. Minimum T2 and T3 periods are 4 us. T4 indicates two parallel receive clock cycles.

Rate Matcher FIFO Buffer Overflow and Underflow Condition

During the normal operating phase, the reset controller monitors the overflow and underflow status of the rate matcher FIFO buffer. If there is overflow and underflow on the rate matcher FIFO buffer, the reset controller asserts rx_digitalreset for two receive parallel clock cycles. You can monitor the rate matcher FIFO buffer status through the pipestatus[2:0] signal from the PIPE interface. This condition is shown in Figure 2–162.

Figure 2–162. PIPE Mode Reset During Rate Matcher FIFO Buffer Overflow and Underflow Condition



Notes to Figure 2–162:

(1) Pipestatus = 101 represents elastic overflow.

(2) Pipestatus = 110 represents elastic overflow.

Power Down

The Quartus II software automatically selects the power-down channel feature, which takes affect when you configure the Stratix II GX device. All unused transceiver and blocks in a design are powered down to reduce the overall power consumption. You cannot use the power-down feature on the fly to turn the transceiver channels and transceiver blocks on and off without reconfiguration.

You can set the transceiver block to power down automatically in the Quartus II software or to power down dynamically in the PLD fabric through the gxb_powerdown port. Assertion of this port does not power down the refclk reference clock buffer.

The gxb_powerdown port is optional. In simulation, if the gxb_powerdown port is not instantiated, you must assert tx_digitalreset, rx_digitalreset and rx_analogreset signals appropriately for correct simulation behavior. If the gxb_powerdown port is instantiated and other reset signals are not used, you must assert the gxb_powerdown signal for at least one parallel clock cycle for correct simulation behavior. In simulation, you can de-assert the rx_digitalreset immediately after rx_freqlocked signal goes high to reduce the simulation run time. It is not necessary to wait for 4 us as suggested in the actual reset sequence.

In PIPE mode simulation, you must assert the tx_forceelecidle signal for at least one parallel clock cycle before transmitting normal data for correct simulation behavior.

Table 2–53 lists the I/O pin states during power down for normal operation, power down, and PMA lookback.

Table 2–53. I/O Pin States During Power-Down (Part 1 of 2)

Operation	Transmitter Pins	Receiver Pins	REFCLK Pins	Rref Pins
Normal operation	Transmitter	Receiver	Clk input	Ext. reference R
Power down	Tri-state *(1)*	Tri-state *(1)*	Tri-state *(2)*	Low *(3)*

Table 2–53. I/O Pin States During Power-Down (Part 2 of 2)

Operation		Transmitter Pins	Receiver Pins	REFCLK Pins	Rref Pins
PMA loopback	Serial loopback	Tri-state *(4)* toggle	Receiver *(5)*	—	—
	Reverse serial loopback	Transmitter *(4)*	Receiver	—	—

Notes to Table 2–53:
(1) Either leave these pins floating or connect `n_leg` to `GND` through a 10-kΩ resistor and connect `p_leg` to `GXB_VCC` through a 10-KΩ resistor to improve the device's immunity to noise.
(2) Either leave these pins floating or connect `refclk(n)` to `GND` through a 10-kΩ resistor and connect `refclk(p)` to `GXB_VCC` through a 10-kΩ resistor to improve the device's immunity to noise.
(3) Altera recommends driving the reference resistor pin low for the powered down transceiver block.
(4) All supported VODs.
(5) It must be left floating or driven to a constant value.

TimeQuest Timing Analyzer

For Stratix II GX designs, you can either use the Classic Timing analyzer or TimeQuest for static timing analysis. TimeQuest does not automatically constrain the transceiver reset ports and asynchronous input/output ports. As a result, TimeQuest does not perform timing analysis on these paths.

TimeQuest reports these unconstrained paths in RED in the Timing Analyzer report. You must manually add the constraints in the Synopsys Design Constraints (**.sdc**) file for TimeQuest to analyze these paths.

Unconstrained Reset Ports

In the Quartus II software versions 7.1 and 7.1 sp1, TimeQuest does not constrain the following transceiver reset ports:

- `gxb_powerdown`
- `tx_digitalreset`
- `rx_digitalreset`
- `rx_analogreset`

Identifying Unconstrained Reset Ports
To identify the unconstrained reset/powerdown ports, follow these steps:

1. After compiling your design, select the **TimeQuest Timing Analyzer** in the **Tools** drop-down menu. This opens up the Quartus II TimeQuest Timing Analyzer window.

2. In the **Tasks** pane, execute **Report Unconstrained Paths**. This will report all unconstrained paths in RED in the **Report** pane.

3. Expand the **Unconstrained Paths** option in the **Report** pane and further expand the **Setup Analysis** or **Hold Analysis** option.

4. Under **Setup Analysis** or **Hold Analysis**, you will see **Unconstrained Input Port Paths**, **Unconstrained Output Port Paths**, or both, depending on how the reset/powerdown ports are driven.

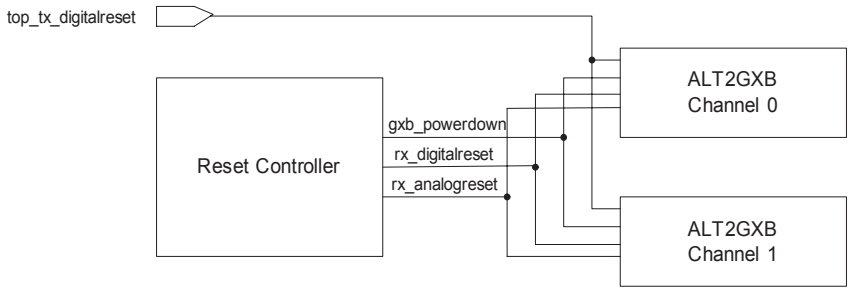
    a. If a reset/powerdown port is driven by an input pin, it will be listed in the **Unconstrained Input Port Paths** report.

    b. If a reset/powerdown port is driven by synchronous logic, it will be listed in the **Unconstrained Output Port Paths** report.

5. In the **Unconstrained Input Port Paths** and **Unconstrained Output Port Paths** reports, the unconstrained reset/powerdown ports of your ALT2GXB instances are listed under the **To** column.

Consider the design example in Figure 2–163.

Figure 2–163. Example Design for TimeQuest Timing Analyzer Constraints



In the design example in Figure 2–163, all reset/powerdown ports except the `tx_digitalreset` port for the two channels are driven by the reset controller. The `tx_digitalreset` port is driven from an input pin.

Figures 2–164 and 2–165 show the TimeQuest Timing Analyzer Report for **Unconstrained Input Port Paths** and **Unconstrained Output Port Paths**, respectively.

Figure 2–164. Unconstrained Input Port Paths

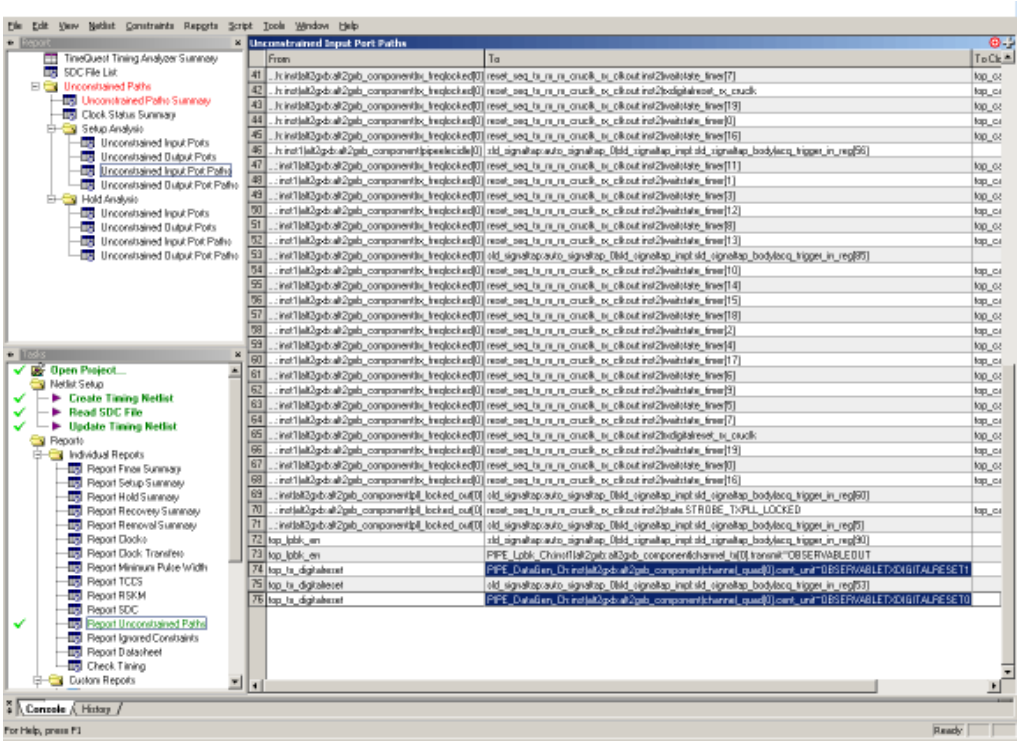
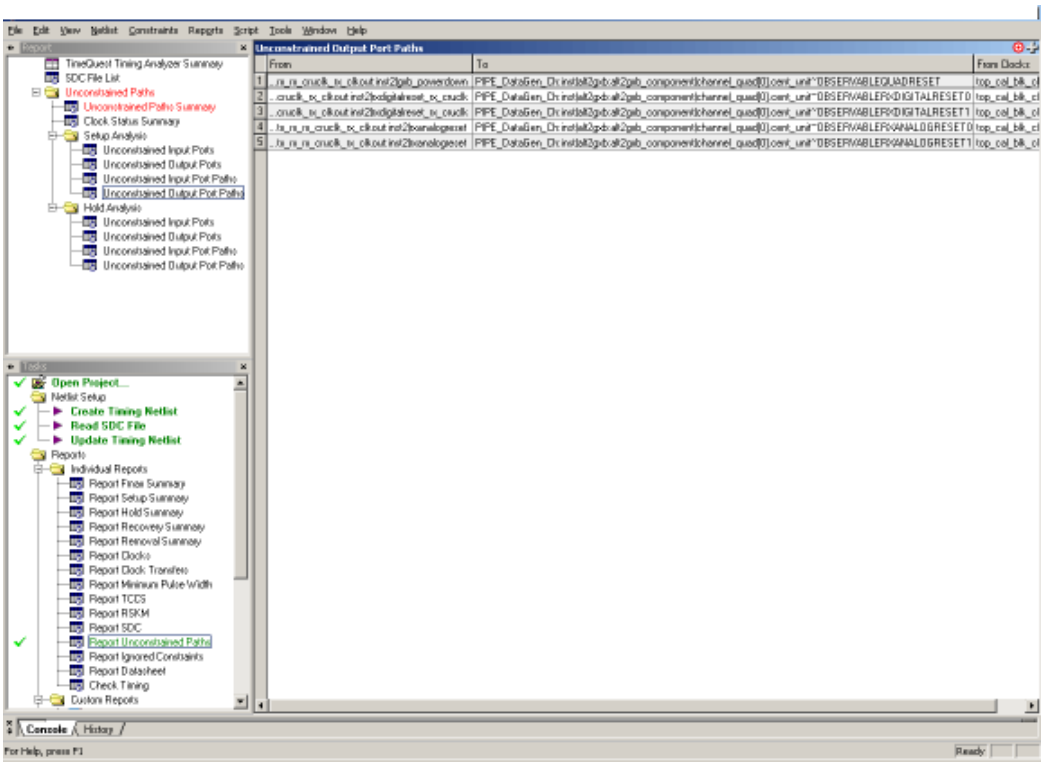


Figure 2–165. Unconstrained Output Port Paths



Having identified the unconstrained reset/powerdown ports in the design, the next step is to constrain these ports.

Setting Reset/Powerdown Port Timing Constraints

You must add the reset/powerdown port timing constraints either directly in the SDC file or through the TimeQuest Timing Analyzer GUI.

To add the timing constraints using the TimeQuest GUI, follow these steps:

1. Locate the reset/powerdown ports in either the **Unconstrained Input Port Paths** or **Unconstrained Output Port Paths** report.

2. Right click on the reset/powerdown port in the **To** column and select **Set Max Delay**. On the resulting window, enter an initial **Delay Value** of **4** ns.

3. Right click on the reset/powerdown port in the **To** column again and select **Set Min Delay**. On the resulting window, enter an initial **Delay Value** of **1.2** ns.

The difference between the maximum delay and minimum delay is set to **2.8** ns which is the maximum skew allowed on reset/powerdown ports.

4. Similarly, set the maximum and minimum delay for all transceiver reset/powerdown ports in your design.

5. Execute **Update Timing Netlist** and **Write SDC File** by double-clicking these options in the **Tasks** pane of the TimeQuest Timing Analyzer window. Confirm that the above timing constraints were added to the SDC file linked with your design.

6. Run the Quartus II Fitter.

7. After the Quartus II Fitter operation completes, execute **Update Timing Netlist** by double-clicking this option in the **Tasks** pane of TimeQuest Timing Analyzer window.

8. Execute **Report Top Failing Paths** by double-clicking this option in the **Tasks** pane of the TimeQuest Timing Analyzer window.

9. Assuming all other paths in your design meet timing, one or more of the paths involving reset/powerdown ports might report timing violations. This is because the design is not able to meet the preliminary timing constraints of 4 ns (maximum delay) and 1.2 ns (minimum delay).

10. Note the slack in the timing report for all failing paths and adjust the maximum delay and the minimum delay values in the SDC file. Maintain a difference of 2.8 ns between the maximum delay and the minimum delay for each reset/powerdown port.

11. After adjusting the delay values, execute **Update Timing Netlist** and run the Quartus II Fitter again.

12. After the Quartus II Fitter operation completes, execute **Update Timing Netlist**.

13. Execute **Report Top Failing Paths** once again. If there are any failing paths involving the reset/powerdown ports, adjust the delay values in the SDC file and repeat the procedure until no failing paths are reported.

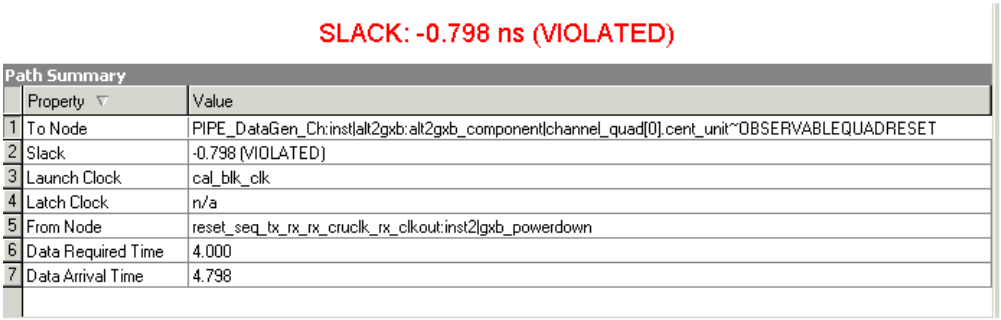
Consider the previous design example in which all unconstrained ports were identified. The following example shows how to set the constraints for the gxb_powerdown port. The same procedure must be followed for all other reset ports.

After setting the maximum and minimum delay for the gxb_powerdown port, the SDC file should have the following constraints:

```
#****************************************************
# Set Maximum Delay
#****************************************************

set_max_delay -from  [get_keepers
{reset_seq_tx_rx_rx_cruclk_rx_clkout:inst2|gxb_powerd
own}]  -to  [get_ports
{PIPE_DataGen_Ch:inst|alt2gxb:alt2gxb_component|chann
el_quad[0].cent_unit~OBSERVABLEQUADRESET}] 4.000

#****************************************************
# Set Minimum Delay
#****************************************************

set_min_delay -from  [get_keepers
{reset_seq_tx_rx_rx_cruclk_rx_clkout:inst2|gxb_powerd
own}]  -to  [get_ports
{PIPE_DataGen_Ch:inst|alt2gxb:alt2gxb_component|chann
el_quad[0].cent_unit~OBSERVABLEQUADRESET}] 1.200
```

After running the Quartus II fitter with the above timing constraints for the gxb_powerdown port, the following slack is reported on this path after executing **Report Top Failing Paths** (Figure 2–166).

Figure 2–166. Slack Reported for the gxb_powerdown Port



Since the data arrival time is later than the data required time by 0.798 ns, the maximum delay and minimum delay should both be incremented by 0.8 ns in the SDC file. The new SDC file should have the following modified constraints for the gxb_powerdown port.

```
#**************************************************
# Set Maximum Delay
#**************************************************

set_max_delay -from  [get_keepers
{reset_seq_tx_rx_rx_cruclk_rx_clkout:inst2|gxb_powerd
own}]  -to  [get_ports
{PIPE_DataGen_Ch:inst|alt2gxb:alt2gxb_component|chann
el_quad[0].cent_unit~OBSERVABLEQUADRESET}] 4.8

#**************************************************
# Set Minimum Delay
#**************************************************

set_min_delay -from  [get_keepers
{reset_seq_tx_rx_rx_cruclk_rx_clkout:inst2|gxb_powerd
own}]  -to  [get_ports
{PIPE_DataGen_Ch:inst|alt2gxb:alt2gxb_component|chann
el_quad[0].cent_unit~OBSERVABLEQUADRESET}] 2.000
```

After modifying the SDC file and running the Quartus II Fitter, the **Update Timing Netlist** option should be executed, followed by **Report Top Failing Paths**. If the gxb_powerdown port still shows in the failing paths, modify the slack appropriately in the SDC file and repeat the procedure until timing is met on this path.

Follow the same procedure to set timing constraints on all transceiver reset/powerdown ports in your design.

You should set constraints and meet timing for both fast and slow timing models. The same maximum and minimum delay constraints might not be able to meet timing for both timing models. This is acceptable as long as the skew is within the specified period (2.8 ns) for each path in the SDC file for each timing model.

Unconstrained Asynchronous ALT2GXB Ports

In the Quartus II software versions 7.1 and 7.1 sp1, TimeQuest does not automatically constrain transceiver asynchronous input/output ports. These ports are listed in Table 2–54.

Table 2–54. TImeQuest Port Names Versus ALT2GXB Port Names

TimeQuest Port Name	ALT2GXB Port Name
ala2size	rx_ala2size
enapatternalign	rx_enapatternalign
bitslip	rx_bitslip
rlv	rx_rlv
invpol	rx_invpolarity
enabyteord	rx_enabyteord
pipe8b10binvpolarity	pipe8b10binvpolarity
revbitorderwa	rx_revbitorderwa
bisterr	rx_bisterr
bistdone	rx_bitstdone
phaselockloss	rx_pll_locked
freqlock	rx_freqlocked
seriallpbkben	rx_seriallpbken

You must manually add the timing constraints in the SDC file for TimeQuest to analyze these paths. For these asynchronous ports, you only need to set a maximum delay constraint of **10** ns in the SDC file.

To identify all unconstrained ALT2GXB asynchronous ports, execute **Report Unconstrained Paths** in TimeQuest Timing Analyzer after running the Quartus II Fitter. Set a maximum delay of **10** ns for all such ports in the SDC file.

For example, if the `rx_invpolarity` signal is driven by the signal `top_rx_invpolarity` on an input pin, the SDC file constraint for this port should be set as:

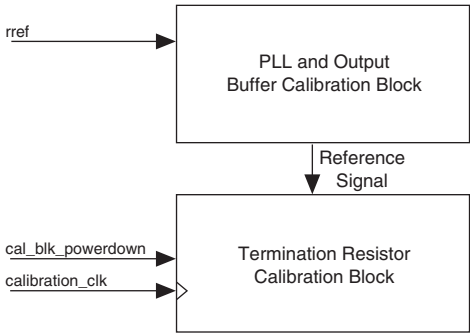
```
set_max_delay -from  [get_ports {top_rx_invpolarity}]
-to  [get_keepers
{xcvr_inst.receive~OBSERVABLEINVPOL}] 10.000
```

Follow the same procedure to constrain all asynchronous ALT2GXB ports in your design before closing timing analysis for your design.

Calibration Blocks

The Stratix II GX gigabit transceiver block contains calibration circuits to calibrate the on-chip termination, the PLLs, and the output buffers. The calibration circuits are divided into two main blocks: the PLL and output buffer calibration block and the termination resistor calibration block (refer to Figure 2–167). Each transceiver block contains a PLL and output buffer calibration block that calibrates the PLLs and output buffers within that particular transceiver block. Each device contains one termination resistor calibration block that calibrates all the termination resistors in the transceiver channels of the entire device.

Figure 2–167. Calibration Block



PLL and Output Buffer Calibration Block

Each Stratix II GX transceiver block contains a PLL and output buffer calibration circuit to counter the effects of PVT (process, voltage, and temperature) on the PLL and output buffer. Each transceiver block's calibration circuit uses a voltage reference derived from an external reference resistor. There is one reference resistor required for each active transceiver block in Stratix II GX devices. Unused transceiver block's (except the transceiver blocks feeding the termination resistor calibration block) can be left unconnected or be tied to the 3.3-V transceiver analog VCC (if the transceiver block's 3.3-V analog supply is connected to 3.3 V).

Termination Resistor Calibration Block

The Stratix II GX transceiver's on-chip termination resistors in the transceiver channels of the entire device are calibrated by a single calibration block. This block ensures that process, voltage, and temperature variations do not have an impact on the termination resistor value. There is only one termination resistor calibration block per device.

The calibration block uses the reference resistor of transceiver block 0 or transceiver block 1, depending on the device. The calibration block uses the reference resistor in transceiver block 0 for EP2SGX30 and EP2SGX60 devices and the reference resistor in transceiver block 1 for EP2SGX90 and EP2SGX130 devices. A reference resistor must be connected to either transceiver block 0 or transceiver block 1 to ensure proper operation of the calibration block, whether or not the transceiver block is in use. Failing to connect the reference resistor of the transceiver block feeding the calibration block results in incorrect termination values for all the termination resistors in the transceivers of the entire device.

The termination resistor calibration circuit requires a calibration clock. You can use a global clock line if the `REFCLK` pins are used for the reference clock. You can instantiate a calibration clock port in the MegaWizard to supply your own clock through the `cal_blk_clk` port.

The frequency range of the `cal_blk_clk` is 10 MHz to 125 MHz. If there are no slow-speed clocks available, use a divide-down circuit (for example, a ripple counter) to divide the available clock to a frequency in that range. The quality of the calibration clock is not an issue, so PLD local routing is sufficient to route the calibration clock.

For multiple ALT2GXB instances in the same device, if all the instances are the same, the calibration block must be active and the `cal_blk_clk` port of all instances must be tied to a common clock. Physically, there is one `cal_blk_clk` port per device. The Quartus II software provides an error message if the `cal_blk_clk` port is tied to different clock sources, because this would be impossible to fit into a device. If there are different configurations of the ALT2GXB instance, only one must have the calibration block instantiated. If multiple instances of the ALT2GXB custom megafunction variation have the calibration block instantiated, then all the `cal_blk_clk` ports must be tied to the same clock source.

The calibration block can be powered down through the optional `cal_blk_powerdown` port (this is an active low input). Powering down the calibration block during operations may yield transmit and receive data errors. Only use this port to reset the calibration block to initiate a recalibration of the termination resistors to account for variations in

temperature or voltage. The minimum pulse duration for this port is to be determined by characterization. If external termination is used on all signals, the calibration block in ALT2GXB need not be used.

Referenced Documents

This chapter references the following documents:

- *ALT2GXB Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *SDI MegaCore Function User Guide*
- *Specifications & Additional Information* chapter in volume 2 of the *Stratix II GX Handbook*.
- *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*

Document Revision History

Table 2–55 shows the revision history for this chapter.

Table 2–55. Document Revision History (Part 1 of 6)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v4.2	Updated: • Figure 2–1 • Figure 2–9 • Figure 2–12 • Figure 2–13 • Figure 2–14 • Figure 2–15 • Figure 2–38 • Figure 2–39 • Figure 2–44 • Figure 2–51 • Figure 2–70 • Figure 2–118	—
	Updated: • Table 2–1 • Table 2–3 • Table 2–6 • Table 2–8 • Table 2–18 • Table 2–19 • Table 2–24	—

Table 2–55. Document Revision History (Part 2 of 6)

Date and Document Version	Changes Made	Summary of Changes
	Updated: • "Reverse Serial Pre-CDR Loopback" • "(OIF) CEI-PHY Interface Mode" • "Clock Synthesis" • "Clock Multiplier Unit" • "Receiver PLL" • "Rate Matcher" • "Transmitter Bit Reversal" • "Receiver Common Mode" • "Native Modes" • "Basic Single-Width Mode" • "Basic Double-Width Mode" • "SONET/SDH Mode" • "Receiver Bit Reversal" • "Pattern Detector Module" • "Channel Clock Distribution" • "Low-Latency PIPE mode" • "Synchronization (Word Aligner)" • "TimeQuest Timing Analyzer"	—
	Added: • "Referenced Documents" • "Serial Digital Interface (SDI) Mode" • "Serial RapidIO Mode" • "CPRI Mode" • "DC Coupling" • "Basic Single-Width Mode with x4 Clocking"	—
	Added Table 2–2.	—
	Minor text edits.	—

Table 2–55. Document Revision History (Part 3 of 6)

Date and Document Version	Changes Made	Summary of Changes
August 2007, v4.1	Moved the Dynamic Reconfiguration section.	The Dynamic Reconfiguration section was moved to the *Stratix II GX Dynamic Reconfiguration* chapter in this handbook.
	Added Table 2–45.	—
	Added note to "Serializer" and "Deserializer".	—
	Updated the "NTFS Fast Recovery IP (NFRI)" section.	—
	Updated: ● "Double-Width General Rate Matching" and Figure 2–73 in the same section ● "PCI Express Receiver Detect" ● "Receiver Detect" ● "Native Modes" (introduction) ● "XAUI Mode" ● "Manual 7-bit Alignment Mode" ● "Manual 8-bit Alignment Mode" ● "Manual 10-Bit Alignment Mode" ● "Manual 16-Bit Alignment Mode" ● "Manual 20-bit Alignment Mode" ● "Manual 32-Bit Alignment Mode" ● Figure 2–72 ● Figure 2–73	—
	Updated Table 2–6 and Figure 2–62.	—
	Added the "TimeQuest Timing Analyzer" section.	—
	Added "Reverse Serial Pre-CDR Loopback" section.	—
February 2007, v4.0	Replaced old Dynamic Reconfiguration section with a new "Dynamic Reconfiguration" section.	—
	Added the "Document Revision History" section to this chapter.	—
	Updated Figures 2–1, 2–2, 2–4, 2–8, 2–9, 2–11, 2–12, 2–20, 2–118, 2–119, 2–120	—

Table 2–55. Document Revision History (Part 4 of 6)

Date and Document Version	Changes Made	Summary of Changes
	Added the following sections: • "Transmitter Phase Compensation FIFO Error Flag" • "Transmitter Force Disparity" • "Transmitter Polarity Inversion" • "Transmitter Bit Reversal" • "Generic Receiver Polarity Inversion" • "Receiver Bit Reversal" • "Receiver Byte Reversal" • "PLD-Controlled Byte Ordering" • "Receiver Phase Compensation FIFO Error Flag" • "Multiple Protocols and Data Rates in a Transceiver Block" • "Low-Latency PIPE mode" • "Design Flow"	—
	Added PLD Interface Clock Resources section, including: • Tables 2–21 through 2–26.	—
	Updated Tables 2–1, 2–9, 2–45.	—
	Updated content from "Transmitter Modules" through "Byte Serializer".	—
	Updated and moved PLD transceiver information to "Receiver Phase Compensation FIFO Error Flag".	—
	Sections modified: • "Receiver Detect" • "Reverse Serial Loopback" • "Signal Threshold Detection Circuit" • "Parallel Loopback" • "Clock Synthesis" • "Double-Width Mode" • "Inter-Transceiver Line Routing" • "Serial Loopback" • "Loopback Modes" • "BIST in Single-Width Mode" • "BIST in Double-Width Mode" • "Transmitter Buffer" • "Transmitter PLL Block" • "Transmitter PLL Bandwidth Setting" • "Transmitter Polarity Inversion"	—

Table 2–55. Document Revision History (Part 5 of 6)

Date and Document Version	Changes Made	Summary of Changes
	622 Mbps was changed to 600 Mbps in: • "Transmitter PLL Block" • "Transmitter PLLs" • "Serializer" • "Receiver Buffer" • "Receiver Common Mode" • "Deserializer"	—
	3.125 to 6.375 Gbps was changed to 1 to 6.375 Gbps in: • "Serializer" • "Deserializer" • "Native Modes"	—
	Updated note in "Normal Operation Phase" section.	—
	The Automatic Mode section was updated and changed to "Word Alignment Based on Byte Ordering".	—
	Changed "bits[15..8]" to "bits[31..24" in the "32-Bit Pattern Mode" section	—
	Changed "rx_outrx_dataout" to "rx_dataout" in the "GIGE Receiver Synchronization" section.	—
	Added new note to the "Dynamic Transmit Rate Switch" and "Dedicated Reference Clock Pin Specifications" sections.	—
	• Changed V_{CCHTX} to V_{CCH} throughout the chapter. • Changed TX V_{CM} to V_{CM} throughout the chapter. • Changed VCC_H to VCCH throughout the chapter.	—
June 2006, v3.2	• Minor change to Figure 2–1. • Updated Table 2–1. • Updated Figures 2–90 and 2–96. • Added "NTFS Fast Recovery IP (NFRI)" section.	• Updated descriptions for `rx_errdetect` and `cal_blk_powerdown` in Table 2–1.

Table 2–55. Document Revision History (Part 6 of 6)

Date and Document Version	Changes Made	Summary of Changes
April 2006, v3.1	• Added "Dedicated Reference Clock Pin Specifications" section, including Table 2–3 and Figure 2–3. • Updated Figures 2–9, 2–16, 2–21, 2–22, 2–30, 2–35, 2–52, 2–53, 2–57, 2–68, 2–69, 2–73, 2–83, 2–84, 2–109, 2–110, and 2–118. • Updated data rate in "Deserializer" section. • Added "7-bit Alignment Mode" section. • Removed references to the `rx_runningdisp` port. • Updated "Manual SONET Alignment Mode (Two Consecutive 8-bit Characters (A1A2) or Four Consecutive 8-bit Characters (A1A1A2A2))" section. • Updated "Manual Alignment Modes" section. • Updated "Code Error Detect" section. • Updated "Disparity Error Detector" section. • Added "Reset Sequence for PIPE Mode" section, including Figures 2–119 and 2–120. • Updated "Dynamic Reconfiguration Setup for alt2gxb Instance" section. • Updated Table 2–34.	• Updated `tx_preemp_2t` port description in Table 2–34 .
February 2006, v3.0	• Updated technical content throughout chapter. • Added "Dynamic Reconfiguration" section.	—
December 2005, v2.0	Updated technical content throughout chapter.	—
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	—

3. Stratix II GX Dynamic Reconfiguration

SIIGX52007-1.1

Introduction

The Stratix® II GX gigabit transceiver block gives you a simplified means to dynamically reconfigure:

- Transmit and receive analog settings
- Transmit data rate in the multiples of 1, 2, and 4
- One channel at a time
- Channel and clock multiplier unit (CMU) PLL
- CMU PLL only

Typically, to achieve the intended bit error rate (BER) for a system, you will take advantage of the multiple analog settings provided in the Stratix II GX device. Being able to change the analog settings is a powerful tool that you can use during link and system debug.

The following analog settings can be dynamically changed:

- Pre-emphasis settings
- Equalization settings
- DC gain settings
- Voltage output differential (V_{OD}) settings

In addition to allowing you to change the equalization settings during runtime, the dynamic reconfiguration controller provides an option to dynamically control the adaptive equalization (AEQ) hardware present in each of the transceiver channels. The AEQ hardware continuously tunes the receiver equalization settings based on the frequency content of the incoming signal.

The dynamic data rate switch feature on the transmitter is enabled through a PLD signal. Depending on the setting of this signal, the transmitter data rate can be divided in steps of 1, 2, or 4 per channel.

Another important feature is the ability to dynamically reconfigure from one mode to another mode. This mode reconfiguration may involve reconfiguring the transceiver data path or data rate or both. You can reconfigure the transceiver data rate either by switching to the other CMU PLL or by dynamically reconfiguring the CMU PLL. The former is enabled in the Quartus® II software version 6.1 and later, while the latter is enabled in the Quartus II software version 7.1 and later.

The dynamic reconfiguration feature facilitates mode transitions involving:

- Protocol functional mode (×1 only) to and from Basic functional mode
- Protocol functional mode (×1 only) to Protocol functional mode (×1 only)
- One Basic functional mode to other Basic functional modes

This is a very useful and powerful feature for transceiver system applications because it enables channels in a system to adapt to multiple serial data rates and system protocols.

Table 3–1 shows dynamic reconfiguration features supported in various Quartus II software versions.

Table 3–1. Software Support for Dynamic Reconfiguration

Version	Transmitter and Receiver Analog Settings (PMA Controls)	Transmitter Data Rate Switch (×1, ×2, ×4)	Channel Reconfiguration	Channel and CMU PLL Reconfiguration	CMU PLL-Only Reconfiguration
Quartus II 6.0	✓	—	—	—	—
Quartus II 6.1	✓	✓	✓	—	—
Quartus II 7.1	✓	✓	✓	✓	✓

Dynamic Reconfiguration Controller Architecture

The Stratix II GX device offers a simplified dynamic reconfiguration controller in the Quartus II ALT2GXB_RECONFIG module to control the configurable settings of the transceiver. The dynamic reconfiguration controller is a soft IP which utilizes Stratix II GX device PLD resources. It is optimized for minimal PLD resource usage. Only one controller is allowed per transceiver block. The dynamic reconfiguration controller does not have the capability to control multiple Stratix II GX devices or any off-chip interface.

The dynamic reconfiguration capability is only intended for Stratix II GX devices, having no backward compatibility to Stratix GX devices.

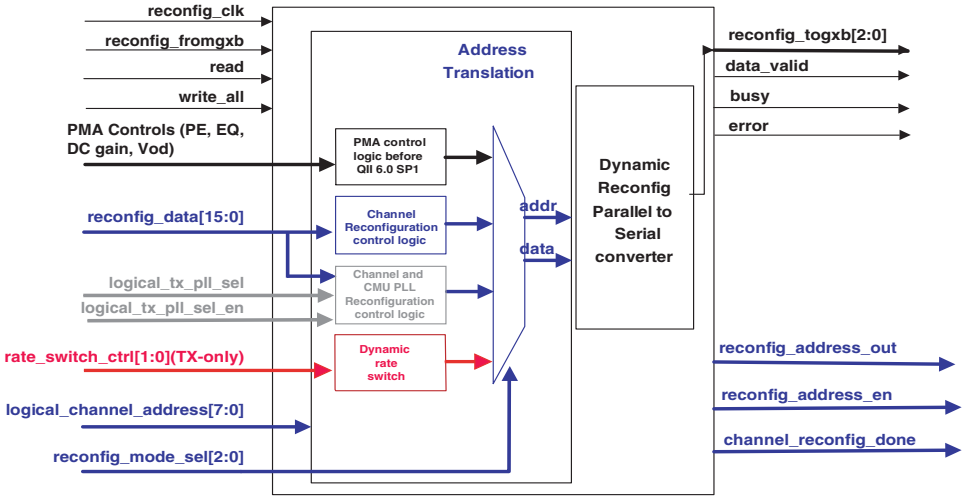
Stratix II GX dynamic reconfiguration is very flexible because of the following features:

- Two transmit PLLs enabled—This allows you to achieve multiple data rates and protocols in a single transceiver block.

- Basic double-width modes—The minimum data rate is lowered to 1 Gbps. This helps if you want to only switch data rates without changing the data path width.
- More optional features in Basic mode.
- PLD interface clocking of the transceiver is enhanced by introducing "Core Clocking Options". These core clocking options help you optimize clock resource usage and allows you to set up the proper PLD interface clocking on transmit and receive paths.

Figure 3–1 shows a conceptual view of these features.

Figure 3–1. Block Diagram of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG)



The following items are not supported as part of the dynamic reconfiguration feature:

- Mode switch to and from any ×4 and ×8 configurations
- Not backward compatible with Stratix GX devices
- To and from PCI Express (PIPE) mode with NFRI IP
- Testability features (pseudo-random binary sequence [PRBS] and built-in self test [BIST])

Dynamic Reconfiguration Setup in the MegaWizard Plug-In Manager

The optional dynamic reconfiguration interface must be enabled through the MegaWizard® Plug-In Manager (dynamic reconfiguration is turned OFF by default).

The dynamic reconfiguration interface has the following signals:

- `reconfig_togxb[2:0]` as an input signal bus
- `reconfig_fromgxb` as an output signal from ALT2GXB instance. `reconfig_fromgxb` is a transceiver block-based signal; for example, if the number of the channels selected in ALT2GXB are:
  - 0 < Channels < 4, then signal `reconfig_fromgxb` = 1 bit
  - 4 < Channels < 8, then signal `reconfig_fromgxb` = 2 bits
  - 8 < Channels < 12, then signal `reconfig_fromgxb` = 3 bits
  - 12 < Channels < 16, then signal `reconfig_fromgxb` = 4 bits
  - 16 < Channels < 20, then signal `reconfig_fromgxb` = 5 bits

After the dynamic reconfiguration option is enabled in the ALT2GXB MegaWizard, you must set one more setting—the **What is the dynamic reconfig starting channel number?** option. The dynamic reconfiguration starting channel number setting range is from 0 - 156 in multiples of 4 (because the dynamic reconfiguration interface is per transceiver block). This range of 0 - 156 is the logical channel address based purely on the number of possible ALT2GXB instances.

To better understand how logical addressing works, consider the scenario of 20 separate transmit and receive instances of the ALT2GXB megafunction in a design and how to set the address of the starting channel of each instance.

The first instance of a transmit and receive channel has the starting channel number setting of **0**. The second instance of a transmit and receive channel has the starting channel number setting of **4**. And so on. The twentieth instance of the same configuration has the starting channel number of **76**.

Extending the same logic to the maximum possible instances case of 20 transmit-only and 20 receive-only configurations, targeted for a five transceiver block Stratix II GX device, the maximum starting channel number of the dynamic reconfiguration option is **156** (40 instances * 4).

Configure the ALT2GXB_RECONFIG and the ALT2GXB modules, depending on the number of transceiver channels that are controlled by the dynamic reconfig controller (ALT2GXB_RECONFIG). Use the logical channel views with the above mentioned logical addressing in the

ALT2GXB instance. The Quartus II fitter errors out if the dynamic reconfiguration option is enabled in the ALT2GXB megafunction, but the `reconfig_fromgxb` and `reconfig_togxb` ports are NOT connected to the ALT2GXB_RECONFIG instance.

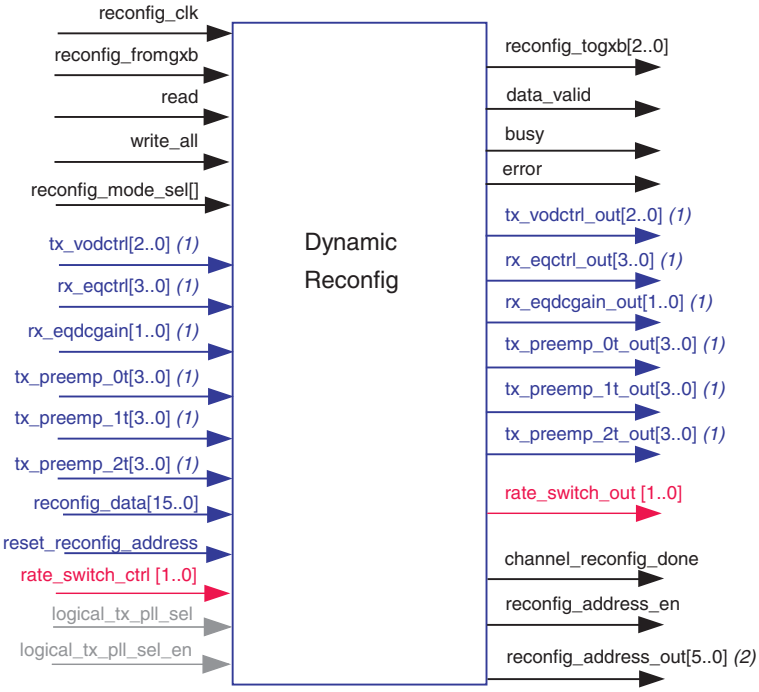
The megafunction and pre-fitter automatically map the logical channel into the physical placements. This physical placement includes merging (automatically done by the Quartus II software). The software performs merging (packing channels into the same transceiver block) only when multiple channels of the same data rate and data path configuration are controlled by one dynamic reconfiguration (ALT2GXB_RECONFIG) controller instance. Channels connected to multiple ALT2GXB_RECONFIG controllers will not be merged.

Dynamic Reconfiguration Controller Interface

The dynamic reconfiguration controller supports write and read transactions. Figure 3–2 shows the dynamic reconfiguration interface list. The following transactions are allowed, based on the dynamic reconfiguration features:

- Analog Settings Reconfiguration—Write and Read (read is optional)
- Channel Reconfiguration—Write Transaction Only
- Dynamic Transmit Rate Switch—Write and Read (read is optional)
- Channel and CMU PLL Reconfiguration
- CMU PLL-Only Reconfiguration

Figure 3–2. Dynamic Reconfiguration Interface



Notes to Figure 3–2:

(1) Optional control and status signals. At least one control signal must be enabled if only analog settings reconfiguration is enabled.

(2) If the channel reconfiguration feature is selected in the ALT2GXB_RECONFIG MegaWizard, the `reconfig_address_out` is 5-bits wide [4..0]. If the Channel and TXPLL select/reconfig feature is selected, the `reconfig_address_out` is 6-bits wide [5..0].

The `reconfig_mode_sel` signal determines the reconfiguration mode. This control signal is 3-bits wide if the **Adaptive Equalization control** option is **not** selected. If this option is selected, the `reconfig_mode_sel` signal is 4-bits wide. Encoding of the `reconfig_mode_sel` signal (when the **Adaptive Equalization control** option is **not** selected) is as follows:

- `reconfig_mode_sel [2:0]`:
  - 000 – Reconfiguration of Analog controls. The Analog controls feature has been enabled in the Quartus II software version 6.0 and later
  - 001 – Channel Reconfiguration
  - 011 – Dynamic Transmit rate switch
  - 100, 101, 110 – Channel and CMU PLL Reconfiguration

Refer to "Channel and PMA Controls Reconfiguration" on page 3–20 and "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" on page 3–87 for detailed `reconfig_mode_sel[2:0]` signal encoding.

As described in "Stratix II GX ALT2GXB Megafunction User Guide" on page 4–1, the signals `reconfig_togxb[2:0]` and `reconfig_fromgxb` are the interface signals between the ALT2GXB instance and the ALT2GXB_RECONFIG instance. The dynamic reconfiguration controller runs at a frequency determined by the clock `reconfig_clk` signal. The supported frequency range of the `reconfig_clk` is 2.5 MHz – 50 MHz.

Altera recommends the `reconfig_clk` signal be driven on a global clock resource.

You must set the following two settings in the ALT2GXB_RECONFIG MegaWizard:

1. What is the number of channels controlled by the controller?

You must provide the number of channels for the megafunction, depending on the design setup supported. There are two ways of using dynamic reconfiguration controllers. They are:

- Single Dynamic Reconfiguration Controller—one controller controlling all the instances of the ALT2GXB in a device. When multiple instances of the ALT2GXB megafunction are controlled by a single ALT2GXB_RECONFIG controller, the following rules should be followed for setting the "**What is the number of channels controlled by the controller**?" option:
  - Each instance of the megafunction must have a set of the consecutive channel numbers beginning with a unique number that is a multiple of four.
  - The number of channels controlled is the last channel number.

- Multiple Dynamic Reconfiguration Controllers—for multiple instances of the ALT2GXB, it is not possible to have two dynamic reconfiguration controllers controlling the same ALT2GXB instance. One controller is allowed to control multiple ALT2GXB instances or every channel will have its own dynamic reconfiguration controller. If every channel has its own dynamic reconfiguration controller, there may be problems with fitting.
  - The Quartus II software cannot merge multiple transceiver channel instances into a transceiver block if multiple dynamic reconfiguration controllers are used, even if the channels are configured to the same protocol functional mode and data rate.

For example, ALT2GXB instance1 has five channels of the same data rate and functional mode; ALT2GXB instance2 has three channels of same data rate and functional mode. Both ALT2GXB instances have separate dynamic reconfiguration controllers controlling them. These two ALT2GXB instances (a total of eight channels) cannot be merged into two transceiver blocks. These two instances can be merged only if they are controlled by one dynamic reconfiguration controller. This merging will not change the behavior of the silicon compared to functional simulations.

2. **Use the same control signals for all channels**.

Check this option when you know that the same analog control signals are used for all the channels in the design. By checking this option, the Quartus II software uses one set of analog signals to control all channels used in all transceiver blocks that are controlled by this reconfiguration controller.

Table 3–2 describes the ports for the dynamic reconfiguration controller.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 1 of 6)

Port Name	Input/Output	Description
reconfig_clk	Input	Input reference clock for the dynamic reconfiguration controller. The frequency range of this clock is 2.5 MHz to 50 MHz. The assigned clock uses global resources by default. This same clock should be connected to ALT2GXB.
ALT2GXB - ALT2GXB_RECONFIG Interface Signals		
reconfig_fromgxb	Input	Interface bus signal from ALT2GXB to ALT2GXB_RECONFIG instance. The width of the signal in ALT2GXB_RECONFIG is determined by the number of channels controlled by the controller.
reconfig_togxb[2..0]	Output	Fixed bus interface between ALT2GXB_RECONFIG and ALT2GXB. This signal is independent of the number of channels.
PLD Interface Signals		
write_all	Input	Control signal to initiate a write transaction. This signal is active high. When the analog settings (V_{OD}, equalization, etc.) are reconfigured, the reconfiguration controller writes to all the transceiver channels connected to the controller.
busy	Output	Status signal to indicate that the reconfiguration controller has not completed the read or write transaction.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 2 of 6)

Port Name	Input/Output	Description
read	Input	Control signal to initiate a read transaction. This signal is active high. When the analog settings (V_{OD}, equalization, etc.) are read, the reconfiguration controller reads the analog setting values from all the transceiver channels connected to the controller. When you select this signal, at least one of the output control ports (for example, tx_vodctrl_out) should be selected. Otherwise, when you initiate a read transaction, the reconfiguration controller may get into a deadlock state (since it cannot send data to any output).
data_valid	Output	Status signal for the read transaction. If data_valid is high, the read back data is valid. That is, the current data on the output control signals after data_valid is asserted high is the valid data read out. This signal is only enabled when at least one read control port is enabled. When a read control port is enabled and a write transaction is finished, the data_valid signal goes high and the busy signal goes low.
error	output	Optional status signal to indicate that an unsupported operation is attempted. The error port can be enabled by selecting the options in the **Error checks/data rate switch** tab. The dynamic reconfiguration controller de-asserts the busy signal and asserts the error signal for two reconfig_clk cycles when you attempt an unsupported operation.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 3 of 6)

Port Name	Input/Output	Description
Analog Settings Control/Status Signals		
tx_vodctrl	Input	Optional transmit buffer voltage output differential (V_{OD}) control signal. It is 3-bits per channel. The number of settings varies based on the transmit buffer supply setting and the termination resistor setting in ALT2GXB instance. The following shows the V_{OD} values corresponding to the tx_vodctrl settings for 100-Ω termination. For V_{OD} values corresponding to other termination settings, refer to Table 2–8. **tx_vodctrl** — **V_{OD} (mV) for 1.5V V_{CCH}** — **V_{OD} (mV) for 1.2V V_{CCH}** 000 — N/A — N/A 001 — 400 — 320 010 — 600 — 480 011 — 800 — 640 100 — 1000 — 800 101 — 1200 — 960 110 — 1400 — N/A 111 — N/A — N/A
tx_preemp_0t *(1)*	Input	Optional pre-emphasis control for pre-tap for the transmit buffer. It is 4-bits per channel. This signal controls both pre-emphasis positive and its inversion. 0 represents 0 1–7 represents -7 to -1 9–15 represents 1 to 7 8 maps to 0
tx_preemp_1t *(1)*	Input	Optional pre-emphasis control for first post tap for the transmit buffer. It is 4-bits per channel.
tx_preemp_2t *(1)*	Input	Optional pre-emphasis control for second post-tap for the transmit buffer. It is 4-bits per channel. This signal controls both pre-emphasis positive and its inversion. 0 represents 0 1–7 represents -7 to -1 9–15 represents 1 to 7 8 maps to 0
rx_eqctrl	Input	Optional equalization control signal on the receive side of the PMA. It is a 4-bit bus per each channel.

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 4 of 6)

Port Name	Input/Output	Description
rx_eqdcgain *(2)*	Input	Optional equalizer DC gain control. It supports three legal settings and is 2-bits wide per channel. 00 corresponds to 0 dB 01 and 10 correspond to 3 dB 11 corresponds to 6 dB
tx_vodctrl_out	Output	Optional transmit V_{OD} output signal. This signal reads out the value written into the V_{OD} control register. The signal width of this output signal is the same as its corresponding input signal.
tx_preemp_0t_out	Output	Optional pre-tap, pre-emphasis output signal. This signal reads out the value written by its input control signal. The signal width of this output signal is the same as its corresponding input control signal.
tx_preemp_1t_out	Output	Optional first post-tap, pre-emphasis output signal. This signal reads out the value written by its input control signal. The signal width of this output signal is the same as its corresponding input control signal.
tx_preemp_2t_out	Output	Optional second post-tap pre-emphasis output signal. This signal reads out the value written by its input control signal. The signal width of this output signal is the same as its corresponding input control signal.
rx_eqctrl_out	Output	Output signal to read the setting of equalization setting of the ALT2GXB instance. The signal width of this output signal is the same as its corresponding input signal.
rx_eqdcgain_out	Output	Equalizer DC gain output signal. This signal reads out the settings of the ALT2GXB instance DC gain. The signal width of this output signal is the same as its corresponding input signal.
Channel Reconfiguration Signals		
reset_reconfig_address	Input	Synchronous reset signal to the ALT2GXB_RECONFIG to reset the reconfig_address_out port to **0**. Use this signal when you want to restart the reconfiguration of a channel by initiating writing the memory initialization file (MIF) word 0.
reconfig_data[15:0]	Input	Sixteen bits input data word. You input it from the location that has the MIF to reconfigure the registers. This input port is only used in the Channel Reconfiguration or Channel and CMU PLL Reconfiguration feature (discussed in "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" on page 3–87).

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 5 of 6)

Port Name	Input/Output	Description
reconfig_mode_sel[2:0]	Input	Select the reconfiguration mode for the ALT2GXB_RECONFIG megafunction. The signal encoding is as follows: 000 - Reconfiguration for analog controls. This feature has been enabled in the Quartus II software version 6.0 and later versions. 001 - Channel Reconfiguration 010 - Not supported (do not attempt to read or write with this value) 011 - Dynamic Transmit data rate switch *100 - TXPLL *101 - Channel and TXPLL reconfiguration *110 - Channel reconfiguration with TXPLL select 111 - Not supported (do not attempt to read or write with this value) *The features corresponding to these values are discussed in "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" on page 3–87.
logical_channel_address [CHANNEL_ADDRESS_WIDTH-1:0]	Input	Specify the logical channel address for the channel that needs to be reconfigured. The CHANNEL_ADDRESS_WIDTH parameter is determined through the NUMBER_OF_CHANNELS parameter.
reconfig_address_out[5:0]	Output	This signal indicates the address out and that the address read out is the current address to be reconfigured by the ALT2GXB_RECONFIG megafunction during channel reconfiguration. This signal is 5-bit wide in channel reconfiguration mode and 6-bit wide in channel and CMU PLL reconfiguration mode.
reconfig_address_en	Output	This port indicates the current address to be reconfigured for the ALT2GXB_RECONFIG megafunction had already changed during channel reconfiguration.
channel_reconfig_done	Output	This port indicates that the ALT2GXB_RECONFIG megafunction has finished writing all the words of a MIF. This is only applicable for channel reconfiguration mode.
Dynamic Transmit Rate Switch Signals		
rate_switch_ctrl[1:0]	Input	This input is the control signal to write the desired division factors on a per-channel transmitter basis. This port is only applicable when reconf_mode_sel is set to **011**. The output value is listed below: 00 - Divide by 1 01 - Divide by 2 10 - Divide by 4 11 - Not supported (do not attempt to read or write with this value)

Table 3–2. Port List of the Dynamic Reconfiguration Controller (ALT2GXB_RECONFIG) (Part 6 of 6)

Port Name	Input/Output	Description
rate_switch_out[1:0]	Output	This signal reads out the value that has written in for the rate switch of specified transmitter outputs. This output port is only applicable when reconf_mode_sel is set to **011**. The output value is listed below: 00 - Divide by 1 01 - Divide by 2 10 - Divide by 4
Channel and CMU PLL Reconfiguration		
logical_tx_pll_sel	Input	This control signal allows you to select the CMU PLL that you wish to reconfigure. It also allows you to select the CMU PLL to which the channel is listening in Channel Reconfiguration with TX PLL Select mode. Refer to "Logical TX PLL Select" on page 3–105 for more information.
logical_tx_pll_sel_en	Input	This signal validates the logical_tx_pll_sel signal. Refer to "Logical TX PLL Select" on page 3–105 for more information.

Notes to Table 3–2:
(1) Not all combinations of bits are legal values.
(2) In PIPE mode, this input should be tied to **01** to be PCI E-compliant.

Dynamic Configuration Controller (ALT2GXB_RECONFIG), ALT2GXB Design Examples

The following design examples illustrate the various possible topologies of the dynamic reconfiguration controller with ALT2GXB instances. The first two design examples specifically discuss a single controller controlling multiple instances of an ALT2GXB and a single controller controlling one instance of an ALT2GXB. Design example three discusses the HDL construct needs if you are stamping the ALT2GXB instances. Each instance of an ALT2GXB in turn can have more than one transceiver channel. Also, in all the design examples, it is assumed that only the Analog (PMA) settings reconfiguration is enabled, to simplify the illustration. In the real system, you can enable other supported features along with the analog setting reconfiguration.

Example 1

Consider a design with two instances of an ALT2XGB configuration, Instance1 with five transceiver channels and Instance2 with three transceiver channels.

Assume the following for this example:

- Instance1 and Instance2 cannot be merged due to their configurations.
- One dynamic reconfiguration controller controls all eight channels.
- Only the transmit V_{OD} and receiver equalization controls are enabled.

The following are the typical steps that help setup the configuration:

Five Channel Transceiver Instance:

- In the ALT2GXB MegaWizard, set the **What is the number of channels?** option to **5** along with other options in the ALT2GXB MegaWizard.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver-block based, so the number of bits for this instance is two since the number of channels is five. The input signal `reconfig_togxb` is a fixed bus width of three bits.
- Set the **What is the starting channel number?** option to **0**.

Three Channel Transceiver Instance:

- In the ALT2GXB MegaWizard, set the **What is the number of channels?** option to **3**.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver block based, so the number of bits for this instance is one since the number of channels is three. The input signal `reconfig_togxb` is a fixed bus width of three bits.
- Set the **What is the starting channel number?** option to **8**. This address of eight is warranted because the previous ALT2GXB instance has five channels which logically fits into two transceiver blocks (transceiver blocks with starting channel numbers 0 and 4), and since this case has multiple instances of the ALT2GXB controlled by one dynamic controller, the numbering is consecutive channel numbers in multiples of four.

ALT2GXB_RECONFIG Setup for PMA Controls Reconfiguration:

- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **12**. The setting for this option has a number that is more than the total number of channels needed to be controlled

(eight channels) by dynamic reconfiguration. This is needed because based on this setting, the Quartus II software chooses the bus width of the signal `reconfig_fromgxb` in addition to the width of the analog control signals. In this case, the design needs 3-bits wide signals so the controller can control a total of three transceiver blocks (five channels in two transceiver blocks and three channels into one transceiver block).

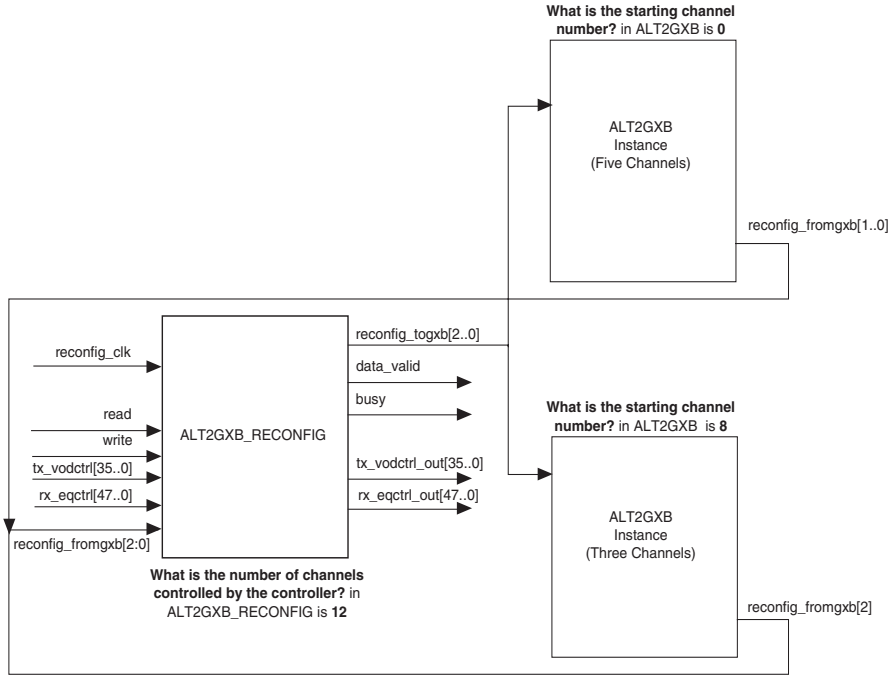
To make it simple, choose the channel number based on a rounded-up channel number to the nearest transceiver block multiple. In this case, it is eight channels required and since no merging is allowed, eight channels require three transceiver blocks. The three transceiver blocks round up to a transceiver block multiple channel number of 12 (3 × 4 = 12). Refer to the "**1. What is the number of channels controlled by the controller?**" option in "Dynamic Reconfiguration Controller Interface" on page 3–5 for more information about this setting.

- Select the necessary analog control signals to write in and read out for V_{OD} and equalization from all the options available in the MegaWizard. Also note the analog control signal widths are for 12 channels since the above channel setting is 12. Control signals for unused channels 5 to 7 and channel 11 can be tied to logic low (zero/ground). In this design scenario, the V_{OD} signal (`tx_vodctrl`) width is 36 bits (12 channel × `tx_vodctrl[2:0]` = `tx_vodctrl[35:0]`). Tie `tx_vodctrl[35:33]` and `tx_vodctrl[23:15]` to ground. Use similar methods for the equalization setting.

ALT2GXB Instances and ALT2GXB_RECONFIG Instance Connections:

- Connect the `reconfig_fromgxb` signal from the ALT2GXB instance to the same signal in the ALT2GXB_RECONFIG instance. The lowest starting channel number transceiver block is connected to the lowest significant bit and so on. In this case, the configuration instance with five channels of the ALT2GXB instance has a starting channel of zero, which has the signal `reconfig_fromgxb[1:0]` which should be connected to `reconfig_fromgxb[1:0]` of the ALT2GXB_RECONFIG instance. The other three channel instances of ALT2GXB, with a starting channel of eight, has the signal `reconfig_fromgxb` which should be connected to ALT2GXB_RECONFIG `reconfig_fromgxb[2]`. Refer to Figure 3–3 for more information.
- Connect the `reconfig_togxb` signal from the ALT2GXB_RECONFIG instance to the same signal on the ALT2GXB instance.

Figure 3–3. ALT2GXB Modules with One ALT2GXB_RECONFIG Module



Example 2

This design example has two instances of distinct configurations: Instance1 with five transceiver channels and Instance2 with three channels. This configuration requires separate dynamic reconfiguration controllers for the two instances. This scenario covers the case of multiple dynamic reconfiguration controllers controlling multiple instances of the ALT2GXB. Assume that the analog settings (transmit V_{OD} and receive equalization controls) for both instances are enabled. The following are the typical steps to setup the configuration:

Five Channel Transceiver Instance1:

- In the ALT2GXB MegaWizard, set the **What is the number of channels?** option to **5** along with other options in the ALT2GXB MegaWizard.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).

- The output signal `reconfig_fromgxb` is transceiver-block based so the number of bits for this instance is two since the number of channels is five. The input signal `reconfig_togxb` is a fixed width of three bits.
- Set the **What is the starting channel number?** option to **0**.

Dynamic Reconfiguration Controller Instance1:

- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **5**. This option helps the Quartus II software choose the bus width of the signal `reconfig_fromgxb` in addition to the width of the analog control signals. In this case, the design needs 2-bits wide signals so the controller can control a total of two transceiver blocks (five channels in two transceiver blocks). Refer to "Introduction" on page 3–1 for more information about this setting.
- Select the necessary analog control signals to write in and read out from the V_{OD}, pre-emphasis, equalization, and DC gain options for this setup.

Three Channel Transceiver Instance2:

- Set the **What is the number of channels?** option to **3**.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver block based so the number of bits for this instance is one since the number of channels is three. The input signal `reconfig_togxb` is a fixed width of three bits.
- Set the **What is the starting channel number?** option to **0**. This address number of 0 is the same as the previous five channel ALT2GXB instance setting. You do not need to have a consecutive channel starting number (multiples of four) since these two ALT2GXB instances are controlled by different dynamic reconfiguration controllers.

Dynamic Reconfiguration Controller Instance2:

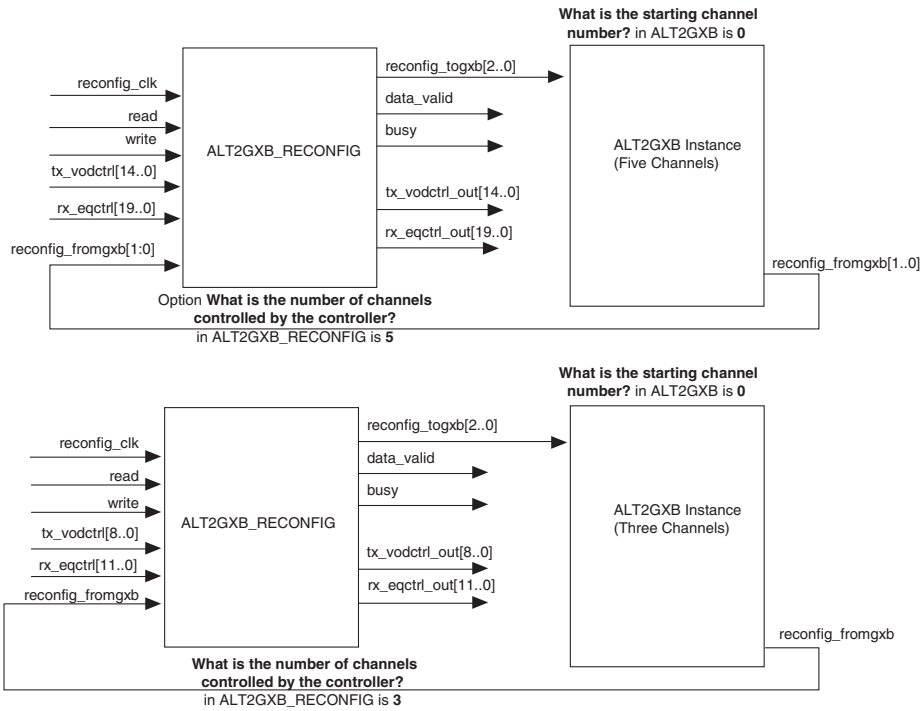
- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **3**. This option helps the Quartus II software choose the bus width of the signal `reconfig_fromgxb`, in addition to the width of the analog control signals. In this case, the design needs a 1-bit wide signal so the controller can control a total of one transceiver block (three channels into one transceiver block) and have the option set to at least three so that the Quartus II software enables three channels of the analog control signals in the options sections.

- Select the necessary analog control signals to write in and read out from the V_{OD}, pre-emphasis, equalization, and DC gain options.

ALT2GXB Instances and ALT2GXB_RECONFIG Instance Connections:

- Connect the `reconfig_fromgxb` signal from the ALT2GXB instance to the same signal of the corresponding ALT2GXB_RECONFIG instance. Refer to Figure 3–4 for more information.
- Connect the `reconfig_togxb` signal from the ALT2GXB_RECONFIG instance to the same signal of the corresponding ALT2GXB instance.

Figure 3–4. ALT2GXB_RECONFIG Modules with Two ALT2GXB Modules



Example 3

This design example consists of five channels of transceivers with the same data rate and functional mode. This configuration has one dynamic reconfiguration controller to control five channels. This scenario covers the case stamping five instantiations of one channel ALT2GXB instance configuration.

One Channel ALT2GXB Configuration:

- Set the **What is the number of channels?** option to **1** along with other options in the ALT2GXB MegaWizard.
- Enable the **Analog controls** option under the dynamic reconfiguration settings (to dynamically change equalization values, also enable the **Enable equalizer settings** option).
- The output signal `reconfig_fromgxb` is transceiver-block based so the number of bits for this instance is one since the number of channels is one. The input signal `reconfig_togxb` is a fixed width of three bits.
- Set the option **What is the starting channel number**? to **0**.

Instantiating Five Times Using the Above 1-Channel ALT2GXB:

- Instantiate the *ALT2GXB.v* file or the symbol file five times.
- Note that after instantiating five times, add the **starting channel number** parameter to the symbol file. Change the parameter option to **4**, **8**, **12**, and **16** for the instances 2, 3, 4, and 5 just created.
- If the instantiations are done in a verilog file, use the following command to force the parameter option to **4**, **8**, **12**, and **16** for the instances 2, 3, 4, and 5:

```
defparam inst2. starting_channel_number= 4;

defparam inst3. starting_channel_number= 8;
```

Dynamic Reconfiguration Controller Instance:

- Launch the ALT2GXB_RECONFIG MegaWizard.
- Set the **What is the number of channels controlled by the controller?** option to **20** so that five interface signals are enabled (`reconfig_fromgxb[4:0]`).
- Select the necessary analog control signals to write in and read out from the V_{OD}, pre-emphasis, equalization, and DC gain options.

ALT2GXB Instances and ALT2GXB_RECONFIG Instance Connections:

- Connect the `reconfig_fromgxb` signal from the ALT2GXB instance to the same signal in the `ALT2GXB_RECONFIG` instance.
- Connect the `reconfig_togxb` signal from the ALT2GXB_RECONFIG instance to the same signal in the ALT2GXB instance.

Channel and PMA Controls Reconfiguration

The write transaction of the controller is initiated on the assertion of the `write_all` signal. In PMA reconfiguration mode, the `write_all` signal writes the current state of all the selected input signals into the ALT2GXB instance channels. The write transaction involves the following sequence:

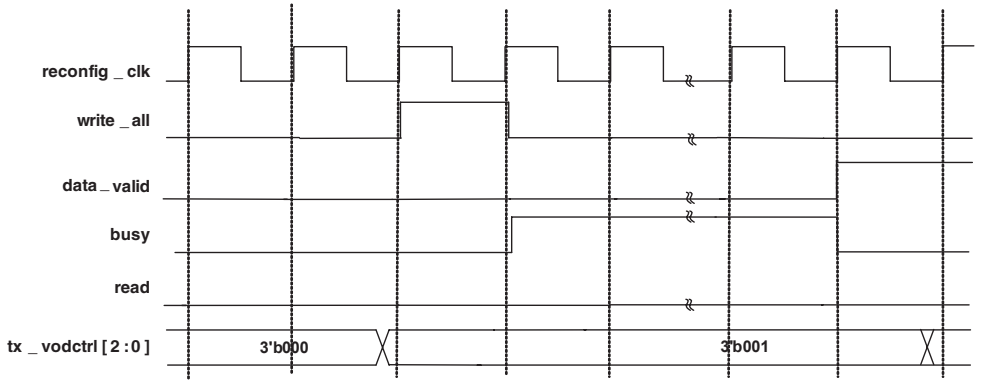
1. Read the control analog registers (read before write).
2. Write the current state of input signals of all channels into control registers.
3. Update the output control signals (optional read control ports if any of the read control ports are enabled).

If you select the read control port, the `data_valid` signal is enabled. Reading and updating all the output control signals is part of the write transaction. Therefore, the `data_valid` signal is asserted only when the write transaction is finished (`busy` signal is low) and all the output control ports are updated with the new data. When a write transaction is initiated and a set of values for the selected analog settings is being written, you cannot change the input values of the control ports until the transaction is completed. Otherwise, the results are unpredictable. The dynamic reconfiguration controller asserts the `busy` signal when you initiate a read or write transaction and is deasserted after the operation is complete.

Simultaneous write and read transactions are not allowed.

Figure 3–5 illustrates a write transaction for a transmit analog setting V_{OD} (`tx_vod`). The waveform shows a typical write transaction initiated by the pulsing of the `write_all` signal and also shows the behavior of the status signals `busy` and `data_valid`. Set the `reconfig_mode_sel` signal to **000** to reconfigure the analog settings of a transceiver channel.

Figure 3–5. Write Transaction Waveform - V_{OD}, Analog Settings Reconfiguration



In channel reconfiguration, only a write transaction can occur—no read transactions are allowed. Set the `reconfig_mode_sel` control signal to **001** to use the channel reconfiguration feature. When you use this feature, the dynamic reconfiguration controller requires that you provide a 16-bit word (`reconfig_data[15:0]`) on every write transaction, using the `write_all` signal. This 16-bit word is part of a Memory Initialization File (**.mif**, also known as MIF) that is generated by the Quartus II software when an ALT2GXB instance is compiled. Refer to "Channel Reconfiguration" on page 3–30 for more information about the MIF.

The dynamic reconfiguration controller ignores a new 16-bit word if the previously initiated write transaction is not complete. As explained above, an on-going or active write transaction is signified by the `busy` signal. You can only input a new word of 16-bits when the `busy` signal is de-asserted.

To properly initiate and complete a write transaction during channel reconfiguration, the dynamic reconfiguration controller provides additional signals. These signals are listed below and are classified into control and status signals.

The following are control signals (other than the write_all and reconfig_mode_sel signals):

- logical_channel_address[7:0]: Use this control signal to set the logical channel number of the channel that is being reconfigured by the dynamic reconfiguration controller. This signal gets enabled when the number of channels controlled by the dynamic reconfiguration controller is more than one. Since the channel reconfiguration is done on a per-channel basis, you have to use this signal and provide the necessary logical channel address to write the MIF words so that a successful channel reconfiguration is achieved for that channel.
- reset_reconfig_address: Use this optional control signal to reset the reconfig_address_out value to **0**. This reset control signal is only applicable in channel reconfiguration.

The following are status signals (other than the busy signal):

- reconfig_address_en: This is an optional output signal. The ALT2GXB_RECONFIG asserts this signal to indicate the change in value on the reconfig_address_out port. This signal only gets asserted after the dynamic reconfiguration controller completes writing the 16-bit data.
- reconfig_address_out[4:0]: This is an optional output signal. It provides the address value that you can use to read the appropriate word from the MIF. Use the value in this port in combination with the reconfig_address_en signal to decide when to initiate a new write transaction.
- channel_reconfig_done: This signal is available when you select the **Channel Reconfiguration** option in the dynamic reconfiguration controller. This port indicates that the ALT2GXB_RECONFIG megafunction has finished writing all the words of a MIF in a sequence. This signal is very useful for user logic to implement reset recommendations during and after dynamic reconfiguration. Refer to "Reset Recommendations" on page 3–66 for more information about using this signal.

- Error: The ALT2GXB_RECONFIG provides this status signal when you select the **Enable illegal mode checking** option or the **Enable self recovery** option in the **Error checks/data rate switch** tab. The conditions under which the error signal is asserted, when the above two options are enabled, are:
  - **Enable illegal mode checking** option—when you select this option, the dynamic reconfiguration controller checks whether an attempted operation falls under one of the seven conditions listed below. The dynamic reconfiguration controller detects these conditions within two reconfig_clk cycles, de-asserts the busy signal, and asserts the error signal for two reconfig_clk cycles.

    1. PMA controls - read operation:

    - None of the analog PMA read output ports (rx_eqctrl_out, rx_eqdcgain_out, tx_vodctrl_out, tx_preemp_0t_out, tx_preemp_1t_out, and tx_preemp_2t_out) are selected in the ALT2GXB_RECONFIG MegaWizard
    - reconfig_mode_sel is set to **0**
    - read signal is asserted

    2. PMA controls - write operation:

    - None of the analog PMA control write input ports (rx_eqctrl, rx_eqdcgain, tx_vodctrl, tx_preemp_0t, tx_preemp_1t, and tx_preemp_2t) are selected
    - reconfig_mode_sel is set to **0**
    - write_all signal is asserted

    3. Channel and/or TX PLL reconfiguration - read operation:

    - reconfig_mode_sel input port is set to **1**, **4**, **5**, or **6**
    - read signal is asserted

    4. Data rate switch - write operation with unsupported value:

    - The rate_switch_ctrl[1:0] input port is set to **11**
    - reconfig_mode_sel input port is set to **4** (if other reconfiguration mode options are selected in the **Reconfiguration settings** tab)
    - write_all is asserted

5. Data rate switch - write operation without input port:

- The `rate_switch_ctrl` input port is not used
- `reconfig_mode_sel` port is set to **4** (if other reconfiguration mode options are selected in the **Reconfiguration settings** tab)
- `write_all` is asserted

6. Data rate switch - read operation without output port:

- The `rate_switch_out` output port is not used
- `reconfig_mode_sel` port is set to **4** (if other reconfiguration mode options are selected in the **Reconfiguration settings** tab)
- `read` is asserted

7. Adaptive Equalization - read operation:

- `reconfig_mode_sel` input port is set to **7**, **8**, **9**, or **10**
- `read` signal is asserted

- **Enable self recovery** option—When this option is selected, the dynamic reconfiguration controller waits for a pre-defined number of `reconfig_clk` cycles based on the operation selected. If the `busy` signal does not go low within the pre-defined number of clock cycles, it asserts the `error` signal for two `reconfig_clk` cycles.

Example for Using Logical Channel Address to Perform Channel Reconfiguration

The dynamic reconfiguration controller provides an output port called `logical_channel_address`. This port is required for the channel reconfiguration and Channel and CMU PLL reconfiguration features to specify the logical transceiver channel that is to be reconfigured. The `logical_channel_address` value depends on how the ALT2GXB is instantiated in the design. In this section, the different ways of setting up the ALT2GXB instantiation and the corresponding `logical_channel_address` values for these transceiver channels are shown.

Example 1:

Consider a design example in which the ALT2GXB instantiation has six transceiver channels:

- In the ALT2GXB MegaWizard (in the **RECONFIG** tab) set the **starting channel number** option to **0**.
- In the ALT2GXB_RECONFIG MegaWizard, set the **Number of channels controlled by the reconfig controller** option to **6**.
- The `logical_channel_address` value for channel 0 is **0** (Channel 0 is the one that is assigned to `tx_dataout[0]`). Similarly, the `logical_channel_address` values for channels 1 through 5 are **1** through **5**, respectively.

Example 2:

Consider a design example with ALT2GXB instance an that has one transceiver channel (assume the instantiation name is **instantiation0**). The **starting channel number** option value for this channel is **0**. If you use this instantiation to create five additional transceiver channels, you will need the following **defparam** parameter settings (for Verilog designs) to change the starting channel number for the stamped instantiations:

defparam <instantiation1>. `starting_channel_number` = **4**

defparam <instantiation2>. `starting_channel_number` = **8**

defparam <instantiation3>. `starting_channel_number` = **12**

defparam <instantiation4>. `starting_channel_number` = **16**

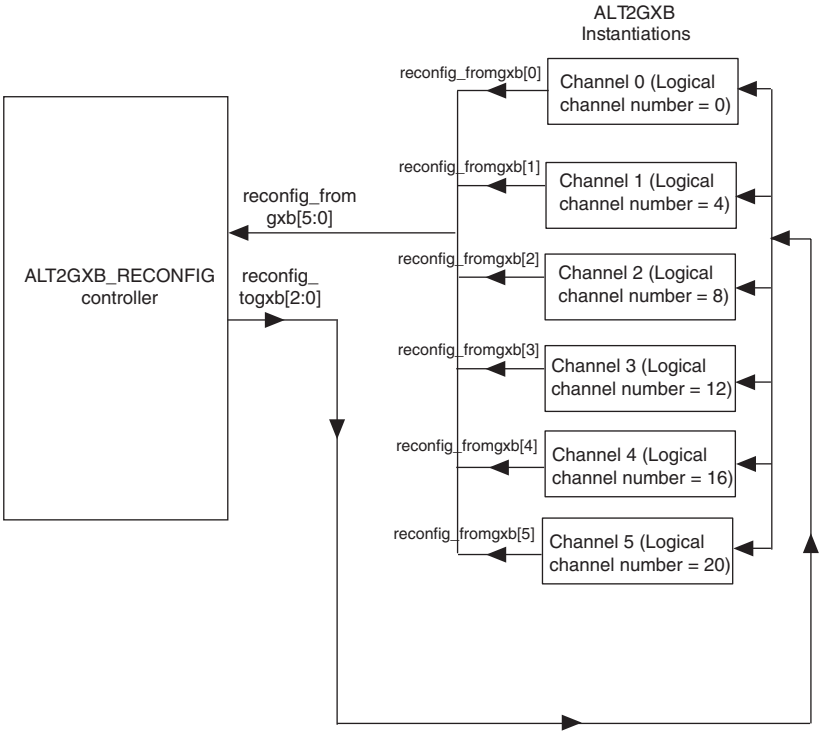
defparam <instantiation5>. `starting_channel_number` = **20**

Therefore, the **starting channel number** option values for channels 0 through 5 are **0**, **4**, **8**, **12**, **16**, and **20**, respectively.

- In the ALT2GXB_RECONFIG MegaWizard, set the **Number of channels controlled by the reconfig controller** option to **24**. By setting this option, you get the `reconfig_fromgxb` port with a bus width of 6.
- Connect the `reconfig_fromgxb[0 to 5]` port of the ALT2GXB_RECONFIG instantiation to the `reconfig_fromgxb` ports of transceiver channels 0 to 5, respectively (as shown in Figure 3–6).
- The `logical_channel_address` values for transceiver channels 0 through 5 (`tx_dataout[0]` to `tx_dataout[5]`) are **0**, **4**, **8**, **12**, **16**, and **20**, respectively.

The `logical_channel_address` value depends on the **starting channel number** option value that you set in the ALT2GXB MegaWizard for the transceiver channel. However, it does not depend on the physical placements of the transceiver channel. For example, you can physically assign `tx_dataout[1]` (`tx_dataout` of instantiation1) in the same transceiver block or in the other transceiver block. For both these assignments, the `logical_channel_address` value is **4** for instantiation1.

Figure 3–6. Multiple Stampings of a Single Channel ALT2GXB Instantiation



Example 3:

Consider a design example with ALT2GXB instance an that has two transceiver channels (assume the example name is **instantiation0**). The **starting channel number** option for this instance is set to **0**. If you want to create six transceiver channels, stamp this instance three times. Modify the **starting channel number** option for the other two instances to **4** and **8** using the **defparam** setting (for verilog design):

defparam <instantiation1>. `starting_channel_number` = **4**

defparam <instantiation2>. `starting_channel_number` = **8**

- In the ALT2GXB_RECONFIG MegaWizard, set the **Number of channels controlled by the reconfig controller** option to 12.
- Connect the `reconfig_fromgxb[0 to 2]` port of the ALT2GXB_RECONFIG instantiation to the `reconfig_fromgxb` ports of instantiation0 to instantiation2, respectively (as shown in Figure 3–7).
- In this case, the `logical_channel_address` values for transceiver channels 0 and 1 (`tx_dataout[0]` and `tx_dataout[1]`) are **0** and **1**. Similarly, the `logical_channel_address` values for channels 2 to 5 are **4**, **5**, **8**, and **9**, respectively. (The **starting channel number** option value for instantiation1 is **4**. Therefore, the `logical_channel_address` value for channel 2 is **4**).

Figure 3–7. Multiple Stampings of a Two Channel ALT2GXB Instantiation

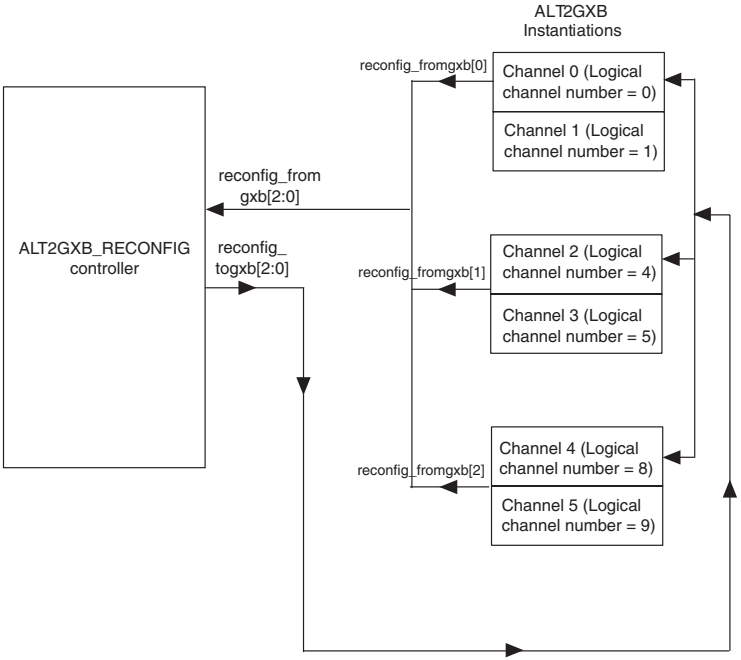
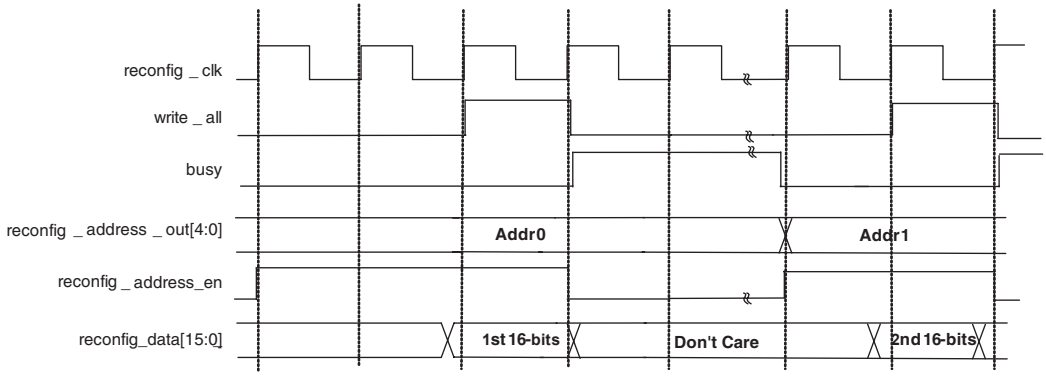


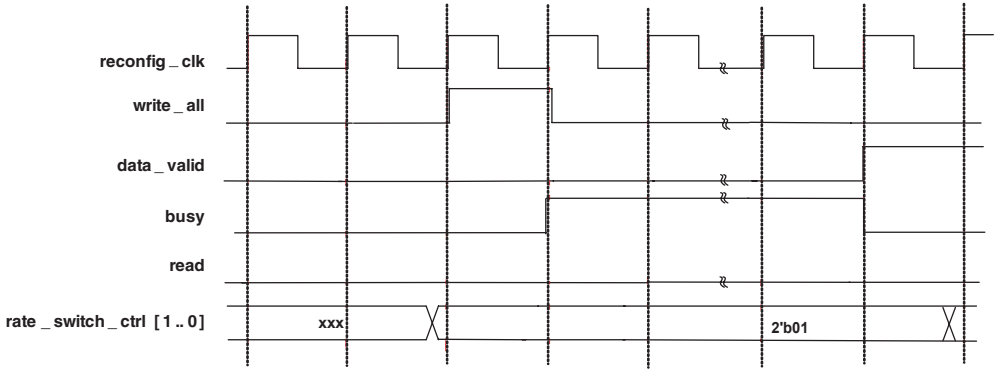
Figure 3–8 illustrates the write transaction for channel reconfiguration.

Figure 3–8. Write Transaction General Waveform – Channel Reconfiguration



A write is allowed in dynamic transmit rate switch mode. The control signal `rate_switch_ctrl[1:0]` determines which division factor is written into the ALT2GXB transmitter (Figure 3–9).

Figure 3–9. Write Transaction General Waveform – Dynamic Transmit Rate Switch Reconfiguration (Division 2)



To initiate a read transaction, assert the `read` signal. The data on the output control ports is not valid until the `data_valid` signal is high. The `data_valid` signal goes high when the entire selected output signals have valid read values. Both read and write transactions are based on the `reconfig_clk` and are edge triggered. Assert the `write_all` and `read` signal for one `reconfig_clk` cycle.

Figure 3–10. Read Transaction Waveform – V_{OD}, Analog Settings Reconfiguration

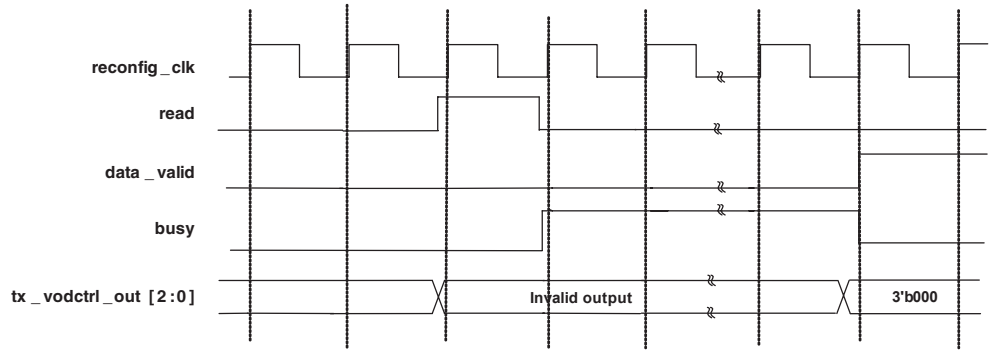
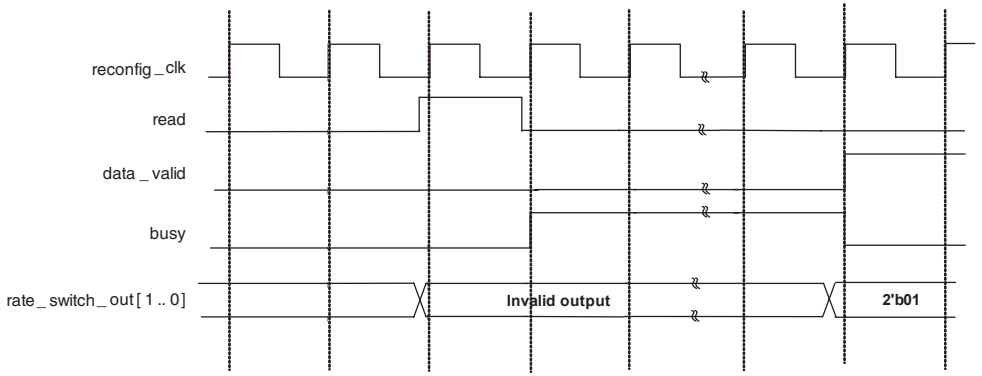


Figure 3–11. Read Transaction Waveform – Dynamic Transmit Rate Switch Reconfiguration (Division 2)



In addition to the PMA reconfiguration, the Quartus II software (version 6.1 and later) dynamic reconfiguration controller enables these two features:

- Channel Reconfiguration
- Dynamic Transmit Rate Switch

The following two sections explain these features.

Channel Reconfiguration

Introduction

Channel reconfiguration provides you the flexibility to reconfigure a channel by writing a new set of legal register bits into the ALT2GXB by the dynamic reconfiguration controller. With this feature you can either reconfigure the data rate of a channel or functional mode (including Basic mode with the custom mode enumeration [CME] features), or a mix of data rates and functional modes. The CME features are additional transceiver features introduced in Basic functional mode. Some of the CME features are controlled by PLD signals that allow you to dynamically control certain features in real time. However, some of the CME features are static and set through the Quartus II ALT2GXB configuration.

Channel reconfiguration only affects the channel involved in the reconfiguration; other channels are not affected.

Channel reconfiguration can be classified into two major areas—data rate reconfiguration and functional mode reconfiguration:

- Data Rate Reconfiguration—Data rate reconfiguration involves switching the data rate of a channel by switching between two TX PLLs and reconfiguring the RX PLLs. The two TX PLLs can be set to different base rates. With data rate reconfiguration, you can also switch the data rate using local clock dividers present in the transmit and receive sides of every transceiver channel. You can reconfigure these clock dividers to 1, 2, and 4. When you reconfigure the clock dividers, ensure that the functional mode supports the minimum and maximum data rate.
- Functional Mode Reconfiguration—this can be:
  - switched between one protocol functional mode to another protocol functional mode
  - switched between a protocol functional mode to a Basic functional mode
  - switched between a Basic mode to another Basic mode

There is no limit to the number of mode switches in channel reconfiguration, assuming transceiver and core clocking supports the transition.

Channel reconfiguration supports the following configurations of the physical transceiver channel:

- Duplex Channels (TX and RX)
- TX Only
- RX Only
- Independent TX/Independent RX in one physical channel

For the following discussion, the reference of a channel is a duplex channel, unless mentioned as TX-only or RX-only.

Design Flow

The Quartus II software provides a design flow called user memory initialization file (**.mif**, also known as MIF) flow to use the channel reconfiguration feature. This design flow involves writing the entire contents of the MIF for a channel. The Quartus II software generates the MIFs when you provide appropriate project settings (discussed below) and then compile an ALT2GXB instance. Each MIF has the settings for a full-duplex transceiver channel. The settings are all legal register settings of the transceiver channel. The ALT2GXB_RECONFIG instance reads the value in the MIF using the `reconfig_data[15..0]` port for every write transaction.

Each MIF contains twenty-eight 16-bit words if you enable the settings shown below.

For the Channel and CMU PLL Reconfiguration feature, the Quartus II software provides new settings that generates a MIF file with 38 words. This is discussed in "Quartus II Settings and Requirements" on page 3–111.

The Quartus II software creates the MIF under the **<Project_DIR>/reconfig_mif** folder. The file name is based on the design name and the rx_ and tx_ pin names. For example: **reconfig_datarate_1Gto2G_pin_af1_pin_af4.mif** (the Quartus II software automatically generates file name).You can change the MIF name. One design can have multiple MIFs (no limit) and one MIF can be used to reconfigure multiple channels. These MIFs can be stored in on-chip or off-chip memory.

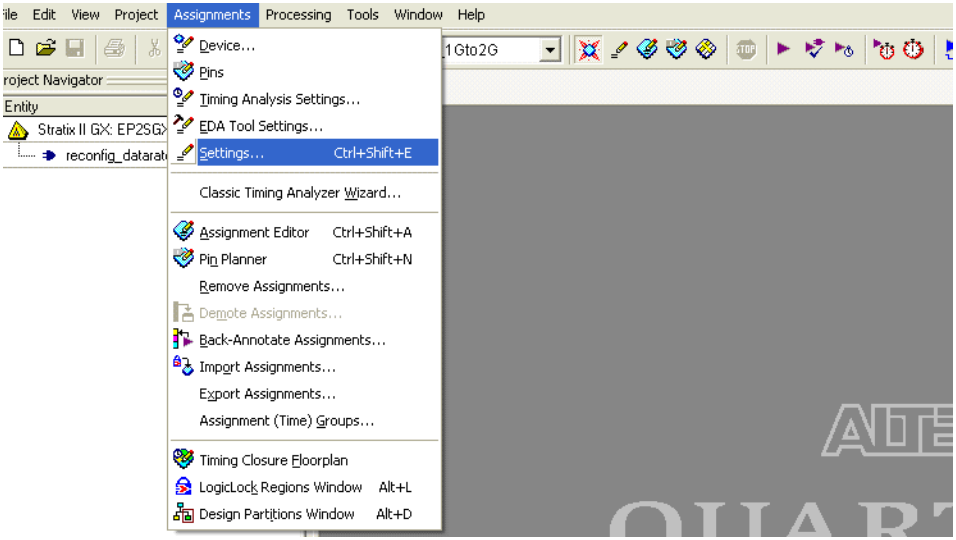
If you do not specify pins for the `tx_dataout` and `rx_datain` for the transceiver channel, the Quartus II software selects a channel and generates a MIF for that channel. However, the MIF can still be used for any transceiver channel.

MIF Generation in Quartus II Software

The MIF is not generated by default in a Quartus II compilation. There are three steps to enable MIF generation. Once the Quartus II software settings are enabled, a MIF is generated after you compile an ALT2GXB instance. The three steps to enable MIF generation are shown below.

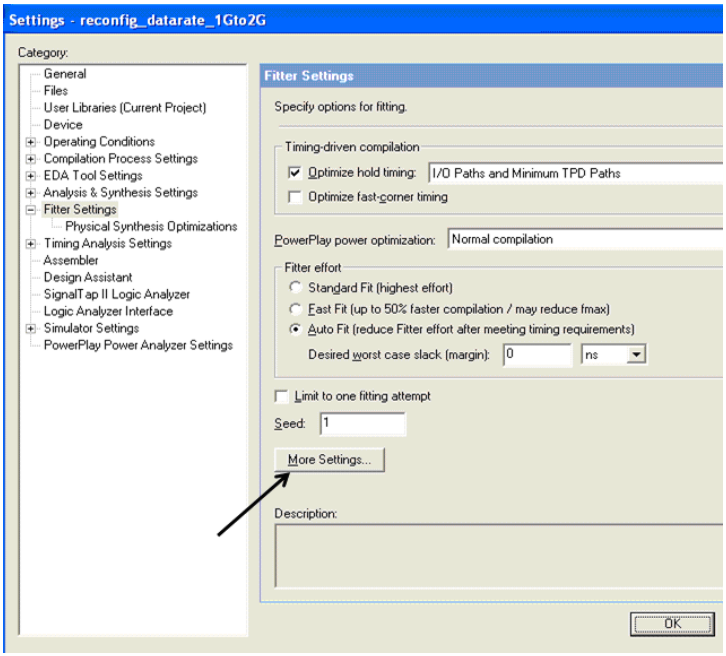
1. On the Assignments menu, select **Settings** (Figure 3–12).

Figure 3–12. MIF Generation, Step 1 (Settings Option)



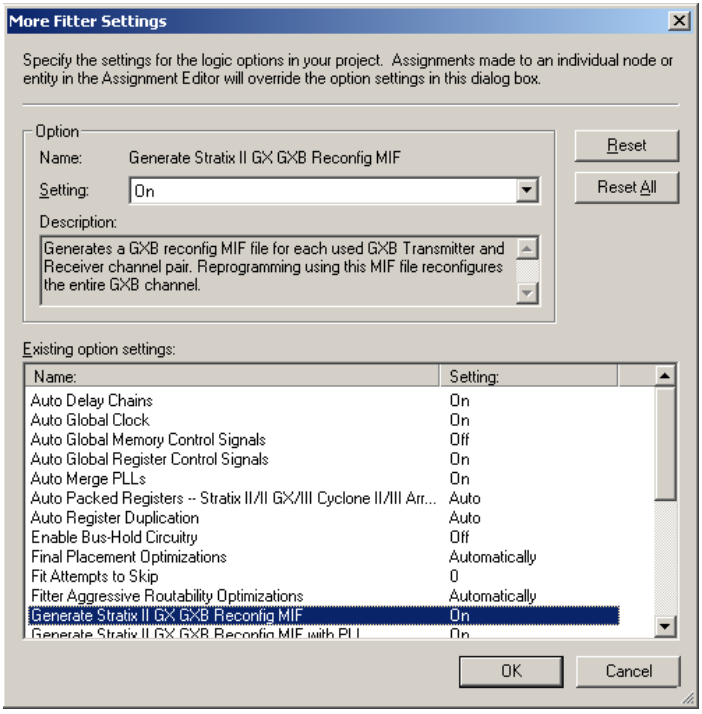
2. Select **Fitter settings**, then choose **More Settings** (Figure 3–13).

Figure 3–13. MIF Generation, Step 2 (Fitter Settings)



3. In the Option box of the **More Fitter Settings** page, set the **Generate Stratix II GX GXB reconfig MIF** option to **On** c (Figure 3–14).

Figure 3–14. MIF Generation, Step 3 (Enable Settings)



The MIF is generated in the Assembler stage of the compilation process. However, for any change in the design or the above settings, the Quartus II software runs through the fitter stage before starting the assembler stage.

As previously discussed, the channel reconfiguration can be a data rate reconfiguration using two TX PLLs and local clock dividers, or a functional mode reconfiguration, or both. To reconfigure a channel successfully, select the appropriate options in the ALT2GXB MegaWizard (discussed in the sections below).

ALT2GXB Configuration Related to Channel Reconfiguration

You must setup the following two system design aspects in a ALT2GXB MegaWizard instance:

- Transceiver and Core Clocking
- PLD Data Path Interface

Transceiver and Core Clocking

You must set up the core clocking and transceiver clocking options as part of channel reconfiguration for functional mode switchover or data rate transition. Transceiver clocking covers all the clock options you need to set up:

- Two TX PLLs for data rates and functional modes
- Input reference clocks for transmit and receive
- Internal clock MUX reference index setups

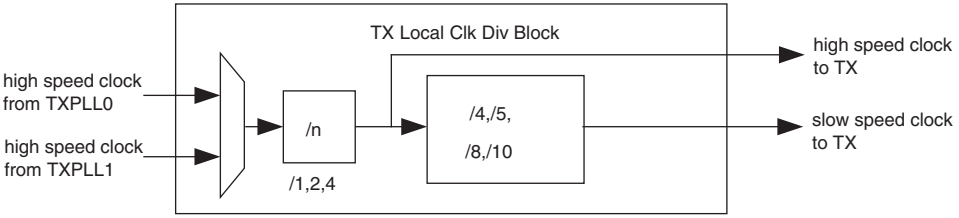
Core clocking covers the PLD interface clocking. PLD interface clocking is related to the parallel transmit and receive clocks (`tx_clkout` and `rx_clkout`). These clocks are used to parallel transmit data into and parallel receive data out of the transceiver. Core clocking is needed in any channel reconfiguration. Core clock assignments (clock grouping assignment and 0 PPM assignments) will override the core clocking set in the ALT2GXB instance. The details related to transceiver and core clocking are discussed in the following section. Transceiver and core clocking are classified as:

- Data rate switch using local clock block dividers
- Data rate switch based on clock frequencies of two PLLs in the transceiver block

Data Rate Switch Using Local Clock Block Dividers

If you intend to switch the data rate in multiples of 1, 2, and 4 of the base data rate, use the local clock dividers. Local clock dividers further divide the TX PLL base rate and are present in transmit and receive block of every transceiver channel (refer toFigure 3–15).

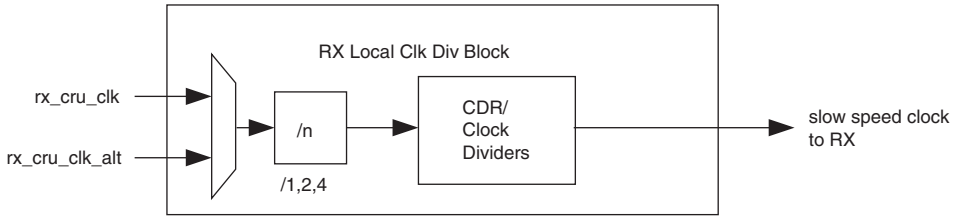
Figure 3–15. Transmit Local Clock Divider Block



Transmit local clock dividers are placed after the CMU PLLs and thus the TX PLLs are not affected during a data rate switch using local clock dividers.

Receive local clock dividers are placed before the RX PLL (CDR). Thus the RX PLL is affected every time the data rate switch using local clock dividers occurs. The Quartus II software data rate division factor chooses a combination of local clock dividers and feedback dividers present in the CDR that yields the best performance (refer to Figure 3–16).

Figure 3–16. Receive Local Clock Divider Block



To configure the local divider using the same TX PLL base setting, use the following steps:

1. Set the base setting on the CMU PLL (use the fastest data rate that is intended to be reconfigured to).
2. Set the local clock divider setting (use the effective data rate for that configuration).
3. Enable either the **Channel Internals** or **Channel Interface** option (refer to "Channel Internals" on page 3–53 and "Channel Interface" on page 3–53 for more information).

4. Setup core clocking (refer to "Transmitter Core Clocking" on page 3–45 and "Receiver Core Clocking" on page 3–48 for more information).

5. Finish the ALT2GXB configuration.

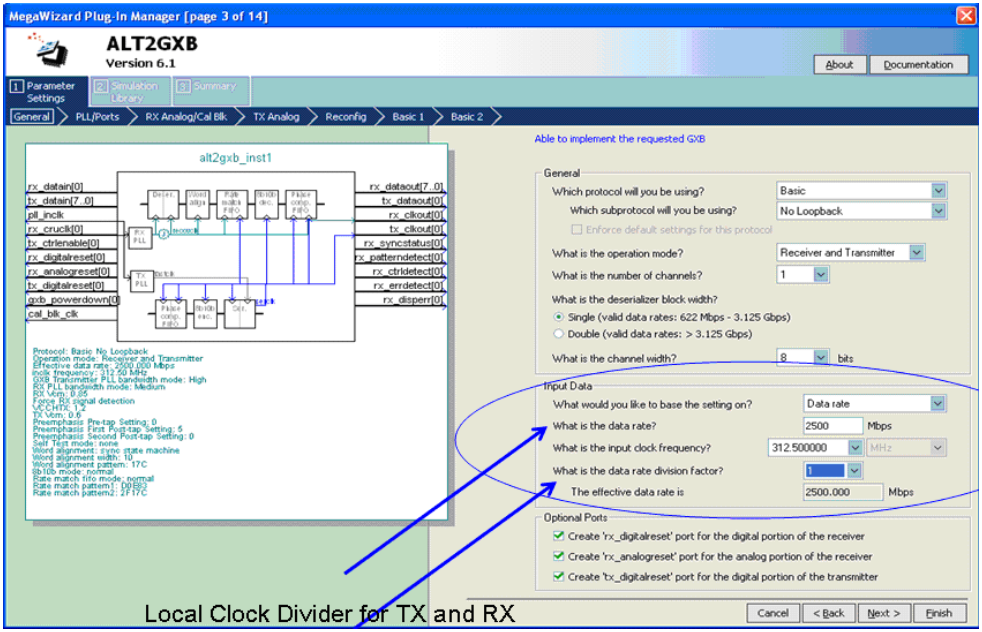
6. Repeat the previous 5 steps with the same TX PLL base setting and different local clock divider settings.

7. Group core clocking.

8. Lock down the pin assignments for the clocks and generate the MIFs for above instances.

Steps 1 and 2 are the only steps related to the local clock divider settings. Step 4 is a mandatory step and is an important part of clocking in every channel reconfiguration (refer to Figure 3–17).

Figure 3–17. ALT2GXB Instance—TX/RX Local Clock Divider



Data Rate Switch Based on Clock Frequencies of Two PLLs

If your application requires the transceiver to switch between multiple data rates, you can use channel reconfiguration to switch between the two TX PLLs in the transceiver block. The following sections explain how to setup two PLLs and achieve multiple data rates using channel reconfiguration:

1. Set the primary PLL (mode1) data rate setting.
2. Set the local clock divider (if needed).
3. Enable the **Channel Internals** option in the dynamic reconfiguration section of the ALT2GXB (refer to "Channel Internals" on page 3–53 for more information).
4. In **Channel Internals** option, enable the **use alternate reference clock (Mode 2)** option.
   - Set all the parameters related to alternate PLL protocols, data rates, bandwidth, and clock frequency.
5. Set the **what is the logical reference index?** option (refer to the Logical Reference Index).
6. Set the core clocking options—transmit and receive
   - This is a mandatory step for every channel reconfiguration that uses `tx_clkout` and `rx_clkout` (refer to "Transmitter Core Clocking" on page 3–45).
7. If there are no other settings to configure in the ALT2GXB, select **finish the ALT2GXB instantiation**.
8. Lock down the input reference clocks pin placements (refer to Pin Assignments).
9. Compile and generate a MIF for Mode1 as primary and Mode2 as alternate.
10. Similarly, generate a MIF for Mode2 as primary and Mode1 as alternate by going through steps 1 through 9 again (refer to "Example 1" on page 3–13).

Figure 3–18 illustrates steps 1 and 2.

Figure 3–18. Local Clock Divider Settings in the ALT2GXB MegaWizard

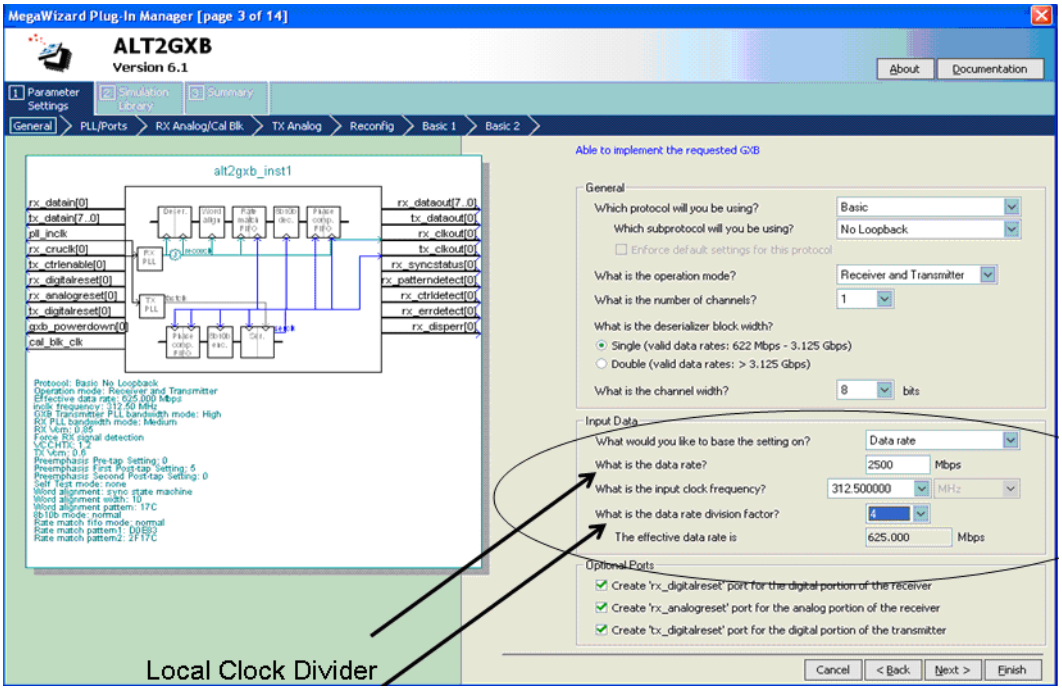
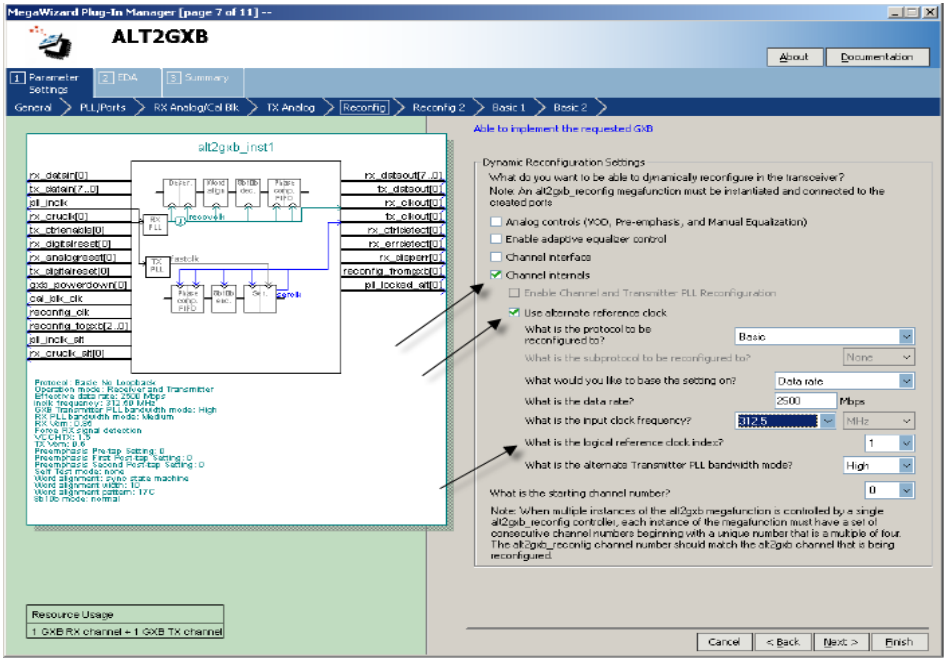


Figure 3–19 illustrates steps 3 and 4 using Basic mode at 2.5 Gbps. In Basic mode, the alternate PLL setup is the most flexible, you can choose and set from the supported bandwidth options and input reference clock frequencies. For example, if the alternate PLL happens to be a protocol functional mode like PCI-E or GIGE, the alternate PLL related options will be automatically populated by the Quartus II software. For more information about the **channel internal** option, refer to "Channel Interface" on page 3–53.

Figure 3–19. Channel Reconfiguration Settings for the ALT2GXB Megafunction



To enable dynamic reconfiguration of a transceiver channel, select either the **channel internals** or **channel interface** options. Selecting these fields creates the `reconfig_fromgxb` and `reconfig_togxb` ports in the ALT2GXB instance. The ALT2GXB_RECONFIG uses these ports to configure the transceiver channel.

In step 5, selecting the **What is the local reference clock index?** option controls the:

- MUX that selects the high-speed clocks from the two TX PLLs
- MUX that selects one of the two input reference clocks (`rx_cruclk` or `rx_cruclk_alt`) on the receive side

For example, consider a system switching from GIGE to SONET/SDH and vice versa. Since both protocols (GIGE with 125-MHz input reference clock and SONET/SDH OC48 with a 77.76-MHz input reference clock) cannot be achieved by one TX PLL, you need a two TX PLL setup. As part of the two TX PLL setup, you will set the logical reference index. To generate a MIF for the GIGE protocol, set the GIGE as the main configuration in the ALT2GXB instance and SONET/SDH mode as the

alternate protocol. This means that GIGE is achieved with the main PLL and the alternate PLL/input reference clock configuration is SONET/SDH OC48. Assume that you set the **Logical Reference Index** option value to **0** (in the **Reconfig** tab).

By setting the logical reference index to **0**, you provide the Quartus II software with the following information.

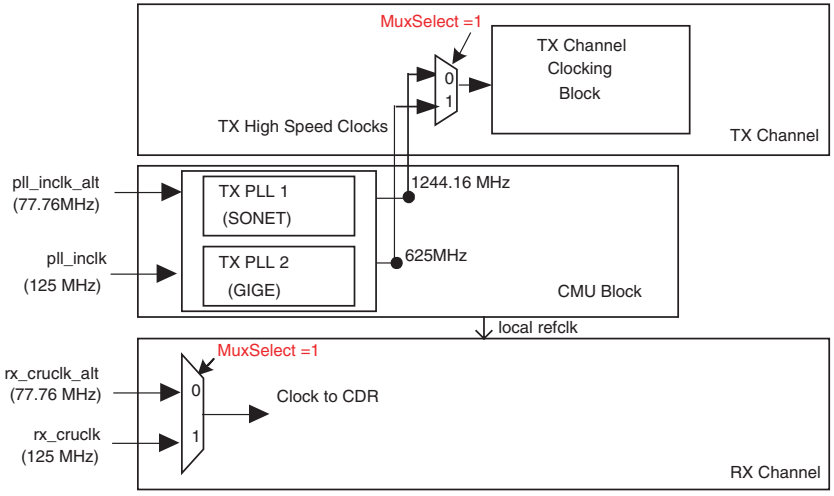
Selection values for the two MUXs mentioned above. The signal name `MuxSelect*` is an assumed name.

- Logical reference index = alternate input reference clock input leg
- MuxSelect* = ~(logical reference index)

In this case, since the logical reference index is set to **0** (represents the SONET/SDH), the TX PLL based on GIGE is routed to input1 of the clock MUX, and the alternate PLL configured for SONET/SDH is connected to input0 of the clock MUX. In the GIGE MIF, the clock MUX select value is set to **1** to choose the clock from the GIGE TX PLL.

Figures 3–20 and 3–21 show the clock MUX connections for GIGE and SONET/SDH, respectively.

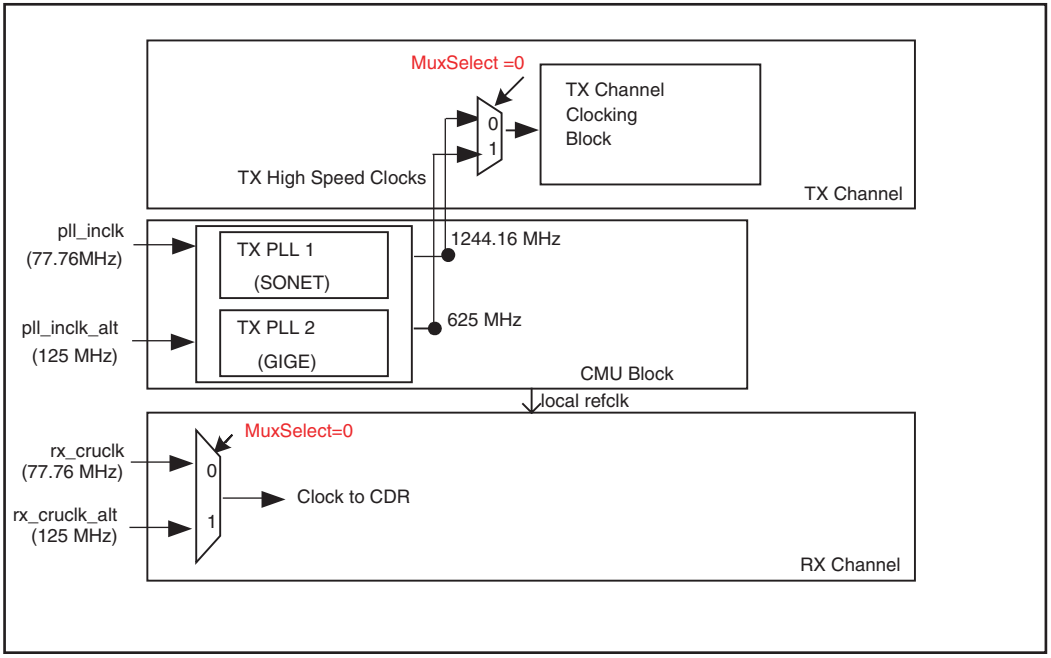
Figure 3–20. TX PLL for GIGE and SONET/SDH OC48 Mode Reconfiguration



To generate a MIF for the SONET/SDH protocol, set the SONET/SDH as the main configuration in the ALT2GXB instance and GIGE as the alternate protocol. This means SONET/SDH OC48 is achieved by the main PLL and the alternate PLL/input reference clock configuration is GIGE. Set the **Logical Reference index** option to **1** (since you have set the logical reference index to **0** for the GIGE instance).

In the SONET/SDH MIF, the clock MUX select value is set to **0** to choose the clock from the SONET/SDH TX PLL

Figure 3–21. MUX Setting - GIGE and SONET/SDH Mode, Logical Reference Clock Index = 1



When two modes are configured to switch from one to another using two TX PLLs, you have to carefully select the logical reference index. In this case, make sure the logical reference index that is set in one MIF is a complement in the second MIF.

Steps 6 is discussed in "Core Clocking" on page 3–45.

Channel Reconfiguration Supported Modes

Channel reconfiguration is supported in the following modes:

- Duplex channels (TX and RX)
- TX only
- RX only
- Independent TX/Independent RX in one physical channel

In the TX-only configuration, there is only one transmitter in a physical transceiver channel. The MIF for the TX-only file has the bits of the unused receiver, but these bits are disabled. The RX-only configuration is the same as the TX-only configuration except it pertains to the receiver.

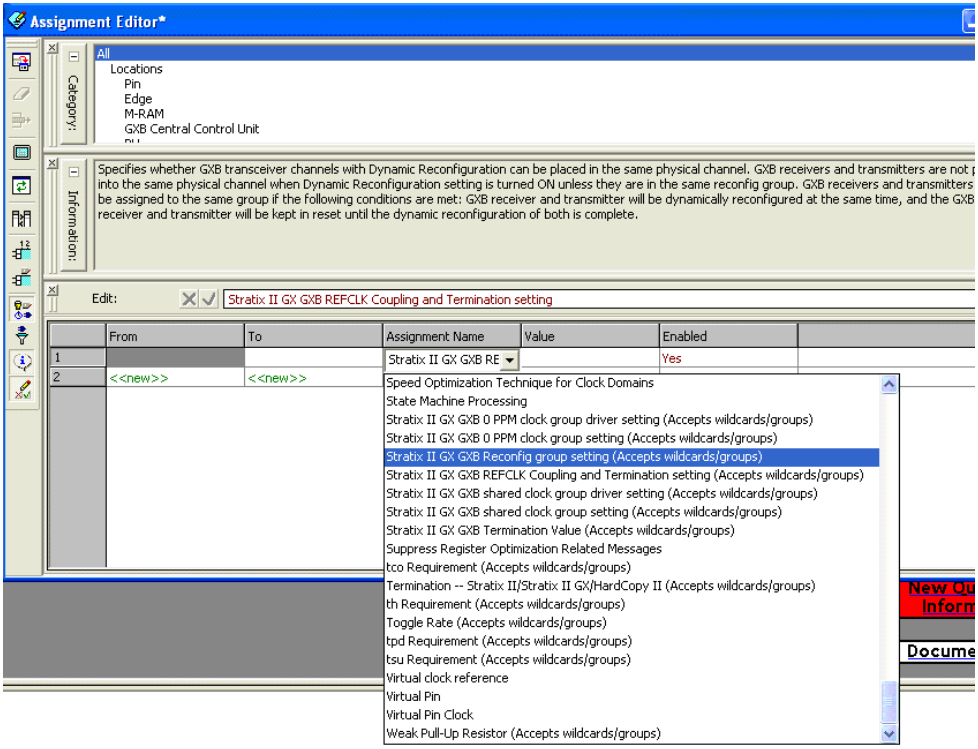
Channel reconfiguration from a TX-only mode to an RX-only mode and vice versa is not allowed.

The Quartus II software allows Independent TX-only configuration with another Independent RX-only configuration in one physical channel. To place an Independent TX configuration and an Independent RX configuration in one physical channel, follow the steps below:

- Perform the pin assignments accordingly
- Instruct the Quartus II software to merge or group the TX and RX register settings into one MIF

There are constraints with the Independent TX-only and Independent RX-only configurations. Both transmitter and receiver have to go through a reset sequence, even if the TX or RX is reconfigured. To merge or group the Independent TX-only and Independent RX-only configurations, place the RX and TX pins into one physical channel. You can accomplish this with the appropriate pin assignment and generation of a MIF through the Quartus II Assignment Editor by setting the **Stratix II GXB reconfig group setting** option to **ON** in the **Quartus Assignment Editor** (Figure 3–22).

Figure 3–22. Quartus II Assignment Editor – TX-Only/RX-Only Merge Option



Core Clocking

Core clocking configuration setup is a mandatory step in every channel reconfiguration. Core clocking is the write and read clock options for the Transmit Phase Comp FIFO and the Receive Phase Comp FIFO, respectively. Core clocking can be further classified to:

- Transmitter core clocking
- Receiver core clocking

Transmitter Core Clocking

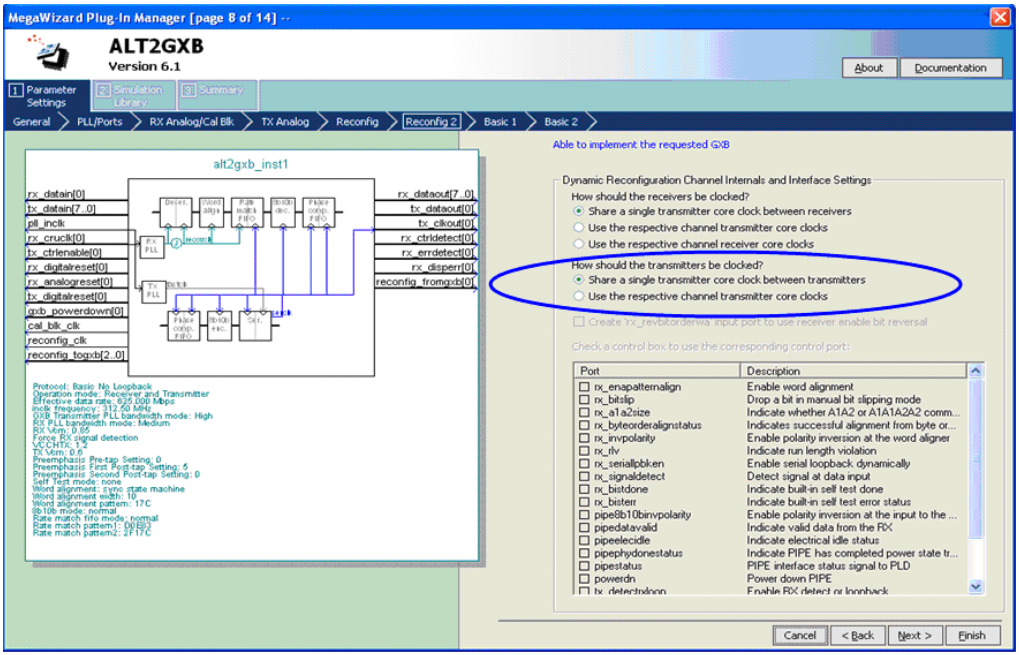
Transmitter core clocking is the write clocking options for the Transmit Phase Comp FIFO. The transmitter core clocking is used to write the parallel data into the Transmit Phase Comp FIFO from the PLD interface.

The possible transmit core clock options are:

- `tx_clkout` (the Quartus II software automatically routes to PLD and back into Transmitter Phase Comp FIFO)
- `tx_coreclk` (user-supplied input clock)

Dynamic reconfiguration allows both transmit clock options. The ALT2GXB MegaWizard provides two options only for the `tx_clkout` settings. When you select the `tx_clkout` options, ensure that the selected `tx_clkout` option is compatible for all the intended reconfiguration modes for the transceiver channel. The `tx_coreclk` selection and clock grouping assignments (Assignment editor) overrides the `tx_clkout` settings set in the ALT2GXB MegaWizard. Figure 3–23 shows the two options in transmit core clocking for `tx_clkout` routing.

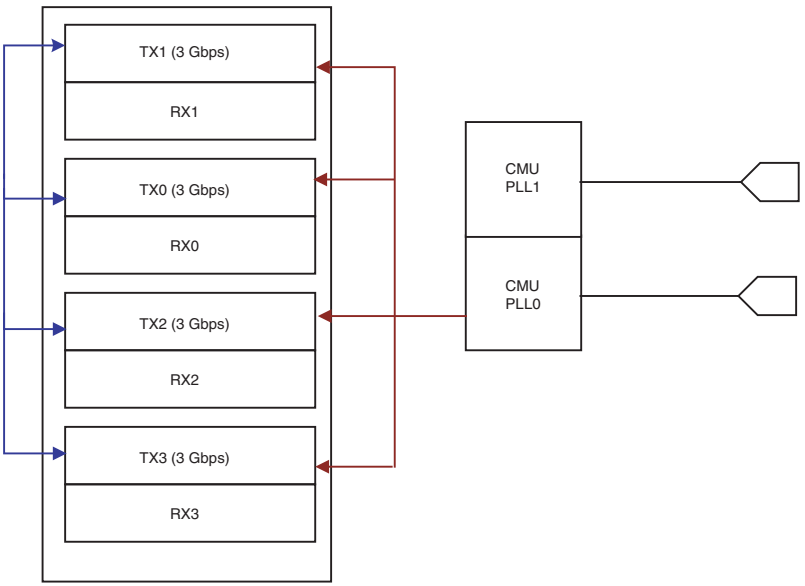
Figure 3–23. ALT2GXB MegaWizard Reconfiguration – Transmit Core Clocking Options



Option 1: Share a Single Transmitter Core Clock Between Transmitters
This option enables the Quartus II software to select channel 0 `tx_clkout` of a transceiver block and routes it to itself and three other channels. This is typically used when all four transmit channels are of the same mode (and also the same data rate) and switch to another mode.

For example, Figure 3–24 shows a setup which has all the transmits configured at 3 Gbps and in the same functional mode. With the dynamic reconfiguration controller and using the channel reconfiguration feature, all four channels switch to 1.5 Gbps and vice versa. Option 1 is applicable in this case and saves clock resources.

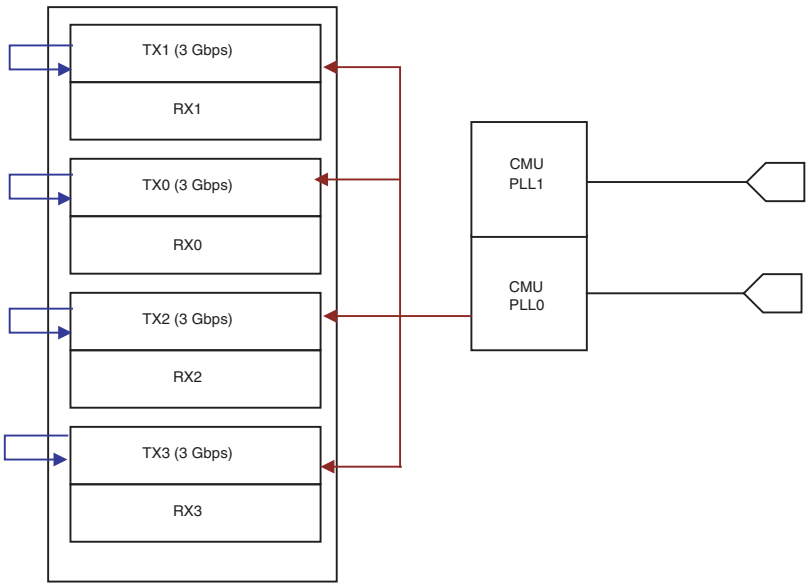
Figure 3–24. Option 1: Channel Reconfiguration—Transmit Core Clocking



Option 2: Use Respective Channel Transmitter Core Clocks
This option enables the Quartus II software to select the individual channel `tx_clkout` signals and route them back through PLD write clock resources to the TX Phase Comp FIFO. This type of core clocking configuration is needed when individual transmit channels can switch modes (basically, each channel switches to a different mode using channel reconfiguration).

Figure 3–24 shows a setup with all the transmitters configured at 3 Gbps and each one at a unique functional mode. Each channel can be switched to a different functional mode using the channel reconfiguration feature of the dynamic reconfiguration controller. In this case, option 2 is applicable.

Figure 3–25. Option 2: Channel Reconfiguration—Transmit Core Clocking



Receiver Core Clocking

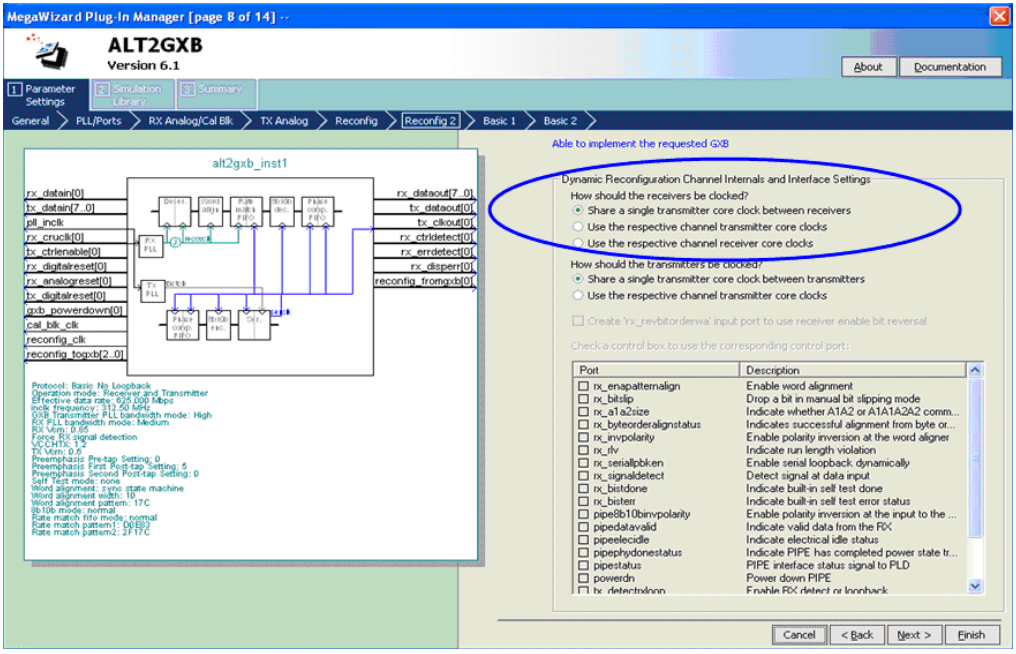
Receiver core clocking is the read clocking options for the Receive Phase Comp FIFO. The receiver core clocking is used to read the parallel data into the Receive Phase Comp FIFO from the PLD interface. The possible transmit core clock options are:

- `rx_clkout` (the Quartus II software automatically routes to PLD and back into Phase Comp)
- `rx_coreclk` (user-supplied input clock)

Dynamic reconfiguration supports both receive clock options. The ALT2GXB MegaWizard only asks for the `rx_clkout` settings. The Quartus II software automatically routes the clock paths based on a given mode setup. You must verify that clock routing is compatible with each

mode. The rx_coreclk selection and its grouping will override the rx_clkout settings set in the ALT2GXB MegaWizard. There are three options in the receiver core clocking for rx_clkout routing.

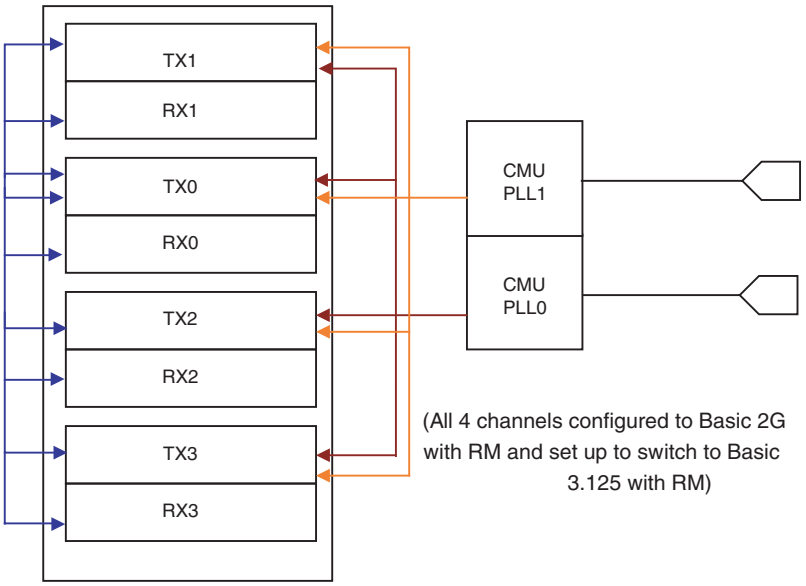
Figure 3–26. ALT2GXB MegaWizard Reconfiguration – Receive Core Clocking Options



Option 1: Share a Single Transmitter Core Clock Between Receivers
This option enables the Quartus II software to select channel 0 tx_clkout of a transceiver block and route it to all four receiver channels. This option is typically set when a transceiver block (all four channels) is in Basic or Protocol mode, with rate matching, switches to another Basic or Protocol mode with rate matching.

Figure 3–27 shows a setup with all four channels configured to a Basic 2 Gbps mode with rate matching, and then switches to a Basic 3.125 Gbps mode with rate matching. In this case, option 1 is applicable.

Figure 3–27. Option 1: Channel Reconfiguration—Receive Core Clocking



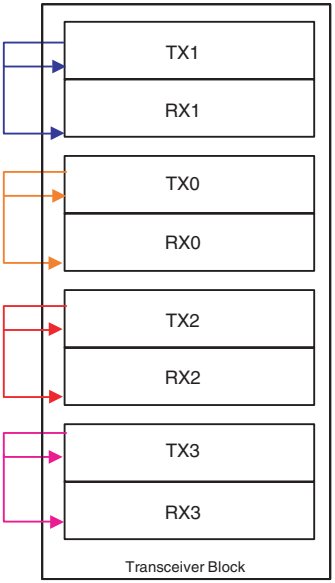
Option 2: Use Respective Channel Transmitter Core Clocks

This option enables the Quartus II software to select the individual channel `tx_clkout` signal and route it to the same channel's receiver PLD interface clock signal. Typically, this option is used when the individual channels in a transceiver block have rate matching with different data rates switched to another Basic or Protocol mode with rate matching.

Figure 3–28 illustrates a setup which has to switch between the following modes:

- TX1/RX1: Basic 1 Gbps with rate matching to Basic 2 Gbps with rate matching
- TX3/RX3: Basic 4 Gbps with rate matching to Basic 1 Gbps with rate matching
- TX0/RX0: Basic 3.125 Gbps with rate matching to 1 Gbps with rate matching and vice versa

Figure 3–28. Option 2: Separate Transmitter Core Clocks—Receive Core Clocking



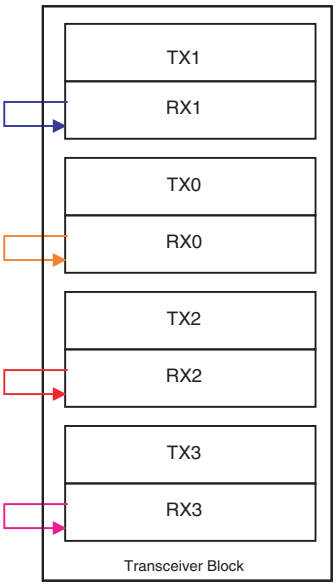
Option 3: Use Respective Channel Receiver Core Clocks
This option enables the Quartus II software to select the individual channel `rx_clkout` signal and route it to the same channel's receiver. Typically, this option is used when a channel is set up to switch from a Basic or Protocol mode with or without rate matching to another Basic or Protocol mode with or without rate matching.

Figure 3–29 illustrates a setup which intends to switch between the following modes:

- TX1/RX1: GIGE to SONET/SDH OC48
- TX2/RX2: Basic 2.5 Gbps no rate matching to Basic 1.244 G bps no rate matching

In this case, option 3 is applicable.

Figure 3–29. Option 3: rxclk_out—Receive Core Clocking



PLD Data Path Interface

For successful channel reconfiguration, you need to set up the following two system design aspects in the ALT2GXB MegaWizard:

- Transceiver and core clocking
- PLD data path interface

Transceiver and core clocking has been explained in detail in the preceding sections. This section discusses the PLD data path interface.

The PLD data path interface needs to be set up when dynamic reconfiguration involves the following:

- Mode switches involving PLD data width changes
- Mode switches involving enabling and disabling of PCS blocks or features (for example, CME features) in a transceiver channel

In the ALT2GXB instance's reconfiguration section, the PLD data path interface can be set up through two subsections:

- Channel Internals
- Channel Interface

Channel Internals

You should enable the **Channel internals** option if the modes that are switched to and from involve the following:

- Static PCS features (including CME features) are enabled or disabled.
- A data rate that needs another TX PLL to be set up (options related to alternate TXPLL need to be configured; for example, the **Use alternate reference clock** option and other sub-options).
- As long as the PLD data path width is not changed and no additional control and status signals are needed. In this case, you only need to enable the **Channel internals** option.

To reconfigure between two modes that differ only in the static features, generate the following MIFs:

- Generate a MIF with the **Channel internals** option enabled and set the appropriate PCS and analog features in the ALT2GXB megafunction.
- Generate a MIF with the **Channel internals** option enabled but with a different set of PCS features (same analog features) configured in the ALT2GXB megafunction.

In this case, the **Use alternate reference clock** option is not enabled, since the reconfiguration did not involve any changes to the data rate that would require another TX PLL.

You can use the **Channel internals** option in conjunction with the **Channel interface** option.

Channel Interface

The **Channel interface** option is enabled if the mode switches involve:

- PLD data path width changes
- PLD control and status flag changes

The **Channel interface** option involves the following:

- A new port called `tx_datainfull[43:0]` is enabled to the PLD interface port list on the transmit side (44-bits wide)
- A new port called `rx_dataoutfull[63:0]` is enabled to the PLD interface port list on the receive side (64-bits wide)
- Enabling the channel interface provides an option pane in the ALT2GXB megafunction where you can select the necessary ports for control and status signals that are needed for each of their channel reconfiguration.

The signals tx_datainfull[43:0] and rx_dataoutfull[63:0] replace the existing tx_datain and rx_dataout ports of a channel. The Quartus II fitter and mapper imposes fewer legal checks related to the connectivity of the signals in tx_datainfull, rx_dataoutfull, and other optional signals. For example, the PIPE mode signals pipestatus and powerdn can be potentially enabled through the ALT2GXB MegaWizard (enabled through the **Reconfig2** tab); the Quartus II software will not restrict this selection. In this case, the software assumes you are planning to switch to and from a PCI-E mode. Figures 3–30 and 3–31 show the MegaWizard pages you use to select the **channel internals** and **channel interface** options.

If the **Channel interface** option is enabled, the following signals are disabled:

- Receiver PLD interface:
  - rx_dataout[39:0]
  - rx_syncstatus[3:0]
  - rx_patterndetect[3:0]
  - rx_a1a2sizeout[3:0]
  - rx_ctrldetect[3:0]
  - rx_errdetect[3:0]
  - rx_disperr[3:0]
- Transmitter PLD interface:
  - tx_datain[39..0]
  - tx_ctrlenable[3:0]
  - tx_forcedisp[3:0]
  - tx_dispval[3:0]

Figure 3–30. ALT2GXB Reconfiguration – Channel Interface Enabled

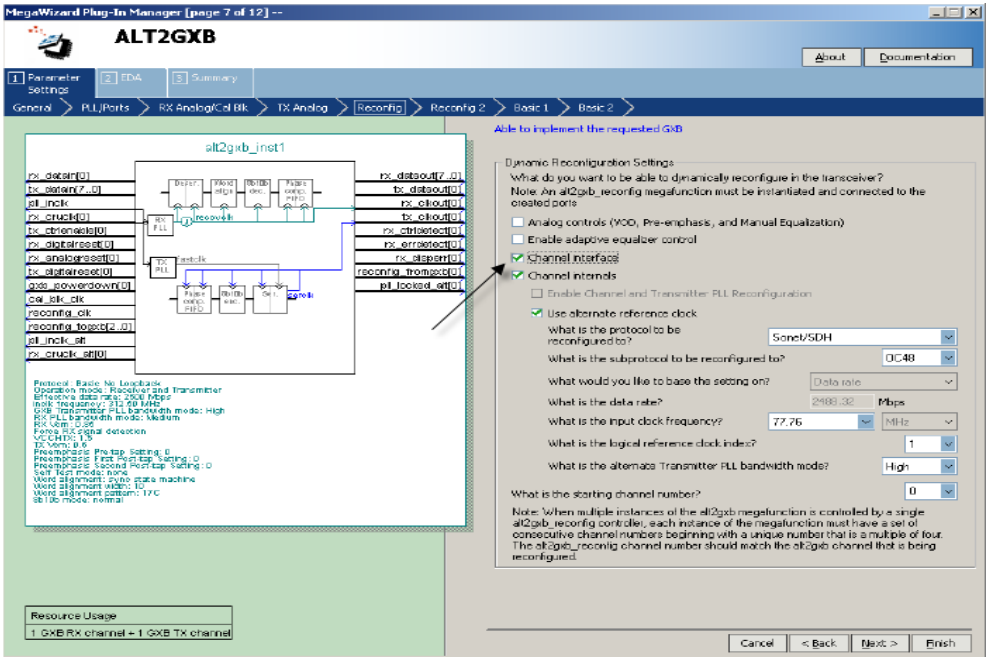
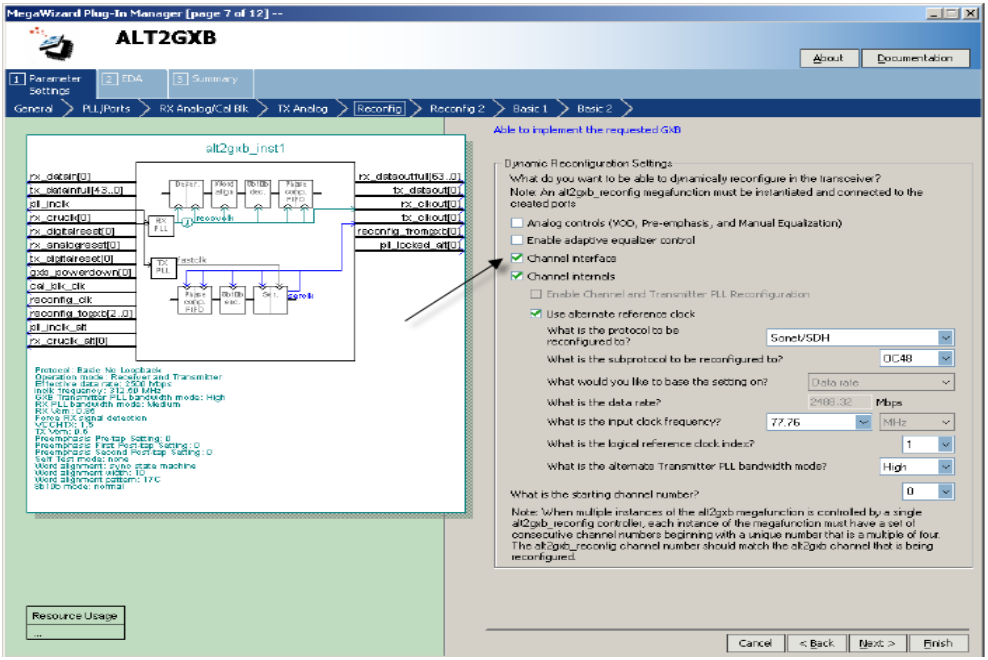


Figure 3–31. ALT2GXB Reconfiguration – Control/Status Signals, Channel Interface Enabled



Signal descriptions for `TX_datainfull[43:0]` and `RX_dataoutfull[63:0]` are shown in Tables 3–3 and 3–4.

Table 3–3. tx_datainfull[43:0] PLD Data Signal Descriptions (Part 1 of 3)

PLD Interface Description	Transmit Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
8-bit PLD Interface	`tx_datainfull[7:0]`: 8-bit data (`tx_datain`)
	The following signals are used only in 8B/10B modes:
	`tx_datainfull[8]`: Control bit (`tx_ctrlenable`)
	`tx_datainfull[9]`: Force disparity enable for `tx_datainfull[7:0]` (non PIPE mode). Transmitter force disparity Compliance (PIPE) (`tx_forcedisp`) in all modes except PIPE. For PIPE mode, (`tx_forcedispcompliance`) is used.
	`tx_datainfull[10]`: Forced disparity value for `tx_datainfull[7:0]` (`tx_dispval`)
10-bit PLD Interface	`tx_datainfull[9:0]`: 10-bit data (`tx_datain`)

Table 3–3. tx_datainfull[43:0] PLD Data Signal Descriptions (Part 2 of 3)

PLD Interface Description	Transmit Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 16/20 bits	Two 8-bit Data (tx_datain) tx_datainfull[7:0] - tx_datain (LSByte) and tx_datainfull[18:11] - tx_datain (MSByte)
	The following signals are used only in 8B/10B modes:
	Two Control Bits (tx_ctrlenable) tx_datainfull[8] - tx_ctrlenable (LSB) and tx_datainfull[19] - tx_ctrlenable (MSB)
	Force Disparity Enable tx_datainfull[9] - tx_forcedisp (LSB) and tx_datainfull[20] - tx_forcedisp (MSB)
	Force Disparity Value tx_datainfull[10] - tx_dispval (LSB) and tx_datainfull[21] - tx_dispval (MSB)
16-bit PLD interface with PCS-PMA set to 8/10 bits	Two 8-bit Data (tx_datain) tx_datainfull[7:0] - tx_datain (LSByte) and tx_datainfull[29:22] - tx_datain (MSByte)
	The following signals are used only in 8B/10B modes:
	Two Control Bits (tx_ctrlenable) tx_datainfull[8] - tx_ctrlenable (LSB) and tx_datainfull[30] - tx_ctrlenable (MSB)
	Force Disparity Enable For non-PIPE: tx_datainfull[9] - tx_forcedisp (LSB) and tx_datainfull[31] - tx_forcedisp (MSB) For PIPE: tx_datainfull[9] - tx_forcedispcompliance (LSB) and tx_datainfull[31] - tx_forcedispcompliance (MSB)
	Force Disparity Value tx_datainfull[10] - tx_dispval (LSB) and tx_datainfull[32] - tx_dispval (MSB)
20-bit PLD interface with PCS-PMA set to 20 bits	Two 10-bit Data (tx_datain) tx_datainfull[9:0] - tx_datain (LSByte) and tx_datainfull[20:11] - tx_datain (MSByte)
20-bit PLD interface with PCS-PMA set to 10 bits	Two 10-bit Data (tx_datain) tx_datainfull[9:0] - tx_datain (LSByte) and tx_datainfull[31:22] - tx_datain (MSByte)

Table 3–3. tx_datainfull[43:0] PLD Data Signal Descriptions (Part 3 of 3)

PLD Interface Description	Transmit Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
32-bit PLD interface with PCS-PMA set to 16/20 bits	Four 8-bit Data (tx_datain) tx_datainfull[7:0] - tx_datain (LSByte) and tx_datainfull[18:11] tx_datainfull[29:22] tx_datainfull[40:33] - tx_datain (MSByte)
	The following signals are used only in 8B/10B modes:
	Four Control Bits (tx_ctrlenable) tx_datainfull[8] - tx_ctrlenable (LSB) and tx_datainfull[19] tx_datainfull[30] tx_datainfull[41] - tx_ctrlenable (MSB)
	Force Disparity Enable (tx_forcedisp) tx_datainfull[9] - tx_forcedisp (LSB) and tx_datainfull[20] tx_datainfull[31] tx_datainfull[42] - tx_forcedisp (MSB)
	Force Disparity Value (tx_dispval) tx_datainfull[10] - tx_dispval (LSB) and tx_datainfull[21] tx_datainfull[32] tx_datainfull[43] - tx_dispval (MSB)
40-bit PLD interface with PCS-PMA set to 20 bits	Four 10-bit Data (tx_datain) tx_datainfull[9:0] - tx_datain (LSByte) and tx_datainfull[20:11] tx_datainfull[31:22] tx_datainfull[42:33] - tx_datain (MSByte)

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 1 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
8-bit PLD fabric interface	**The following signals are used in 8-bit 8B/10B modes:**
	rx_dataoutfull[7:0]: 8-bit decoded data (rx_dataout)
	rx_dataoutfull[8]: Control bit (rx_ctrldetect)
	rx_dataoutfull[9]: Code violation status signal. It indicates error detected in rx_dataoutfull[7:0], which is replaced by invalid code-group (invalid or running disp.error) in GIGE mode. In PCI Express, when code violation occurs, the EDB character is placed on the erroneous data byte (= K30.7) (rx_errdetect)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Disparity error status signal. It indicates disparity error detected in rx_dataoutfull[7:0] (rx_disperr)
	rx_dataoutfull[12]: Pattern detect status signal (rx_patterndetect)
	rx_dataoutfull[13]: Reserved
	rx_dataoutfull[14]: Reserved
	rx_dataoutfull[14:13]: PIPE/PCI-E mode: 2'b00: data OK; 2'b01: 1 SKP deletion; 2'b10: elastic buffer underflow if data is 0xFE, else 1 SKP insertion; 2b11: elastic buffer overflow (rx_pipestatus)
	rx_dataoutfull[15]: Reserved
	The following signals are used in 8-bit SONET/SDH mode:
	rx_dataoutfull[7:0]: 8-bit un-encoded data (rx_dataout)
	rx_dataoutfull[8]: rx_a1a2sizeout
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Reserved
	rx_dataoutfull[12]: rx_patterndetect
10-bit PLD fabric interface	rx_dataoutfull[9:0]: 10-bit un-encoded data (rx_dataout)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Reserved
	rx_dataoutfull[12]: rx_patterndetect
	rx_dataoutfull[13]: Reserved
	rx_dataoutfull[14]: Reserved
	rx_dataoutfull[15]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 2 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 16/20 bits	Two 8-bit un-encoded Data (rx_dataout) rx_dataoutfull[7:0] - rx_dataout (LSByte) and rx_dataoutfull[23:16]- rx_dataout (MSByte)
	The following signals are used in 16-bit 8B/10B modes:
	Two Control Bits rx_dataoutfull[8] - rx_ctrldetect (LSB) and rx_dataoutfull[24]- rx_ctrldetect (MSB)
	Two Receiver Error Detect Bits rx_dataoutfull[9] - rx_errdetect (LSB) and rx_dataoutfull[25]- rx_errdetect (MSB)
	Two Receiver Sync Status Bits rx_dataoutfull [10] - rx_syncstatus (LSB) and rx_dataoutfull[42] - rx_syncstatus (MSB)
	Two Receiver Disparity Error Bits rx_dataoutfull [11] - rx_disperr (LSB) and rx_dataoutfull[43] - rx_disperr (MSB)
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44]- rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[45]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[46]: Reserved
	Two 2-bit PIPE Status Bits rx_dataoutfull[14:13] - rx_pipestatus (LSB) and rx_dataoutfull[46:45] - rx_pipestatus (MSB) PIPE/PCI-E mode: 2'b00: data OK 2'b01: 1 SKP deletion 2'b10: elastic buffer underflow if data is hexFE, else 1 SKP insertion 2'b11: elastic buffer overflow
	rx_dataoutfull[15] and rx_dataoutfull[47]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 3 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 8/10 bits	Two 8-bit Data rx_dataoutfull[7:0] - rx_dataout (LSByte) and rx_dataoutfull[39:32] - rx_dataout (MSByte)
	The following signals are used in 16-bit 8B/10B mode:
	Two Control Bits rx_dataoutfull[8] - rx_ctrldetect (LSB) and rx_dataoutfull[40] - rx_ctrldetect (MSB)
	Two Receiver Error Detect Bits rx_dataoutfull[9] - rx_errdetect (LSB) and rx_dataoutfull[41]- rx_errdetect (MSB)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[42]- rx_syncstatus (MSB)
	Two Receiver Disparity Error Bits rx_dataoutfull[11] - rx_disperr (LSB) and rx_dataoutfull[43] - rx_disperr (MSB)
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44] - rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[45]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[46]: Reserved
	Two 2-bit PIPE Status Bits rx_dataoutfull[14:13] - rx_pipestatus (LSB) and rx_dataoutfull[46:45]- rx_pipestatus (MSB) PIPE/PCI-E mode: 2'b00: data OK 2'b01: 1 SKP deletion 2'b10: elastic buffer underflow if data is hexFE, else 1 SKP insertion 2'b11: elastic buffer overflow (rx_pipestatus)
	rx_dataoutfull[15] and rx_dataoutfull[47]: Reserved
	The following signals are used in 16-bit SONET/SDH mode:
	Two 8-bit Data rx_dataoutfull[7:0] - rx_dataout (LSByte) and rx_dataoutfull[39:32] - rx_dataout (MSByte)
	Two Receiver Alignment Pattern Length Bits rx_dataoutfull[8] - rx_a1a2sizeout (LSB) and rx_dataoutfull[40]- rx_a1a2sizeout (MSB)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[42] - rx_syncstatus (MSB)

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 4 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
16-bit PLD interface with PCS-PMA set to 8/10 bits (continued)	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44] - rx_patterndetect (MSB)
20-bit PLD interface with PCS-PMA set to 20 bits	Two 10-bit Data (rx_dataout) rx_dataoutfull[9:0] - rx_dataout (LSByte) and rx_dataoutfull[25:16] - rx_dataout (MSByte)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[26] - rx_syncstatus (MSB)
	rx_dataoutfull[11] and rx_dataoutfull[27]: Reserved
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[28] - rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[29]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[30]: Reserved
	rx_dataoutfull[15] and rx_dataoutfull[31]: Reserved
20-bit PLD interface with PCS-PMA set to 10 bits	Two 10-bit Data rx_dataoutfull[9:0] - rx_dataout (LSByte) and rx_dataoutfull[41:32] - rx_dataout (MSByte)
	Two Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[42] - rx_syncstatus (MSB)
	rx_dataoutfull[11] and rx_dataoutfull[43]: Reserved
	Two Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) and rx_dataoutfull[44] - rx_patterndetect (MSB)
	rx_dataoutfull[13] and rx_dataoutfull[45]: Reserved
	rx_dataoutfull[14] and rx_dataoutfull[46]: Reserved
	rx_dataoutfull[15] and rx_dataoutfull[47]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 5 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
32-bit mode	Four 8-bit un-encoded Data (rx_dataout) rx_dataoutfull[7:0] - rx_dataout (LSByte) rx_dataoutfull[23:16] rx_dataoutfull[39:32] rx_dataoutfull[55:48] - rx_dataout (MSByte)
	The following signals are used in 32-bit 8B/10B mode:
	Four Control Data Bits (rx_dataout) rx_dataoutfull[8] - rx_ctrldetect (LSB) rx_dataoutfull[24] rx_dataoutfull[40] rx_dataoutfull[56] - rx_ctrldetect (MSB)
	Four Receiver Error Detect Bits rx_dataoutfull[9] - rx_errdetect (LSB) rx_dataoutfull[25] rx_dataoutfull[41] rx_dataoutfull[57] - rx_errdetect (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[10] - rx_syncstatus (LSB) and rx_dataoutfull[26] rx_dataoutfull[42] rx_dataoutfull[58] rx_syncstatus (MSB)
	Four Receiver Disparity Error Bits rx_dataoutfull[11] - rx_disperr (LSB) rx_dataoutfull[27] rx_dataoutfull[43] rx_dataoutfull[59] - rx_disperr (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) rx_dataoutfull[28] rx_dataoutfull[44] rx_dataoutfull[60] - rx_patterndetect (MSB)
	rx_dataoutfull[13], rx_dataoutfull[29], rx_dataoutfull[45] and rx_dataoutfull[61]: Reserved
	rx_dataoutfull[14], rx_dataoutfull[30], rx_dataoutfull[46], and rx_dataoutfull[62]: Reserved
	rx_dataoutfull[15], rx_dataoutfull[31], rx_dataoutfull[47], and rx_dataoutfull[63]: Reserved

Table 3–4. rx_dataoutfull[63:0] PLD Data Signal Descriptions (Part 6 of 6)

PLD Interface Description	Receive Signal Description (Based on Stratix II GX Supported PLD Interface Widths)
	The following signals are used in 32-bit SONET/SDH scrambled backplane mode:
	Four Control Data Bits (rx_dataout) rx_dataoutfull[7:0] - rx_dataout (LSByte) rx_dataoutfull[23:16] rx_dataoutfull[39:32] rx_dataoutfull[55:48] - rx_dataout (MSByte)
	rx_dataoutfull[8], rx_dataoutfull[24], rx_dataoutfull[40], and rx_dataoutfull[56]: four Reserved
	Four Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) rx_dataoutfull[26] rx_dataoutfull[42] rx_dataoutfull[58] - rx_syncstatus (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) rx_dataoutfull[28] rx_dataoutfull[44] rx_dataoutfull[60] - rx_patterndetect (MSB)
40-bit mode	Four 10-bit Control Data Bits (rx_dataout) rx_dataoutfull[9:0] - rx_dataout (LSByte) rx_dataoutfull[25:16] rx_dataoutfull[41:32] rx_dataoutfull[57:48] - rx_dataout (MSByte)
	Four Receiver Sync Status Bits rx_dataoutfull[10] - rx_syncstatus (LSB) rx_dataoutfull[26] rx_dataoutfull[42] rx_dataoutfull[58] - rx_syncstatus (MSB)
	Four Receiver Pattern Detect Bits rx_dataoutfull[12] - rx_patterndetect (LSB) rx_dataoutfull[28] rx_dataoutfull[44] rx_dataoutfull[60] - rx_patterndetect (MSB)

ALT2GXB_RECONFIG Setup for Channel Reconfiguration

The ALT2GXB_RECONFIG (dynamic reconfiguration controller) instance must be set up for the channel reconfiguration feature. You can have one dynamic reconfiguration controller for one ALT2GXB instance (each ALT2GXB instance can have multiple transceiver channels) or one dynamic reconfiguration controller controlling more than one ALT2GXB instance. Select the feature **Channel Reconfiguration (Protocol Switch)**. Set the **What is the number of channels controlled by the reconfig controller?** option and select the optional signals in the channel reconfiguration section. Connect the ALT2GXB and ALT2GXB_RECONFIG instances.

Dynamic Transmit Rate Switch

Dynamic rate switch is ONLY available for the transmit side and not for the receive side. The control signal `rate_switch_ctrl[1:0]` sets up the division factor for the local divider inside the transmit side of the transceiver channel. The following is the encoding for the `rate_switch_ctrl` port:

- 00 - Divide by 1
- 01 - Divide by 2
- 10 - Divide by 4
- 11 - not supported, do not set this value

The above values are written and based in the ALT2GXB_RECONFIG instance initiating a write transaction by pulsing the `write_all` signal. This feature can be enabled through two other features—the analog settings and the Channel Reconfiguration. When two or more features are enabled, the `reconfig_mode_sel` signal needs to be set to the desired feature before a write transaction is initiated. A read transaction is allowed in this feature and the `rate_switch_out[1:0]` is required to read out the current data rate division factor through the ALT2GXB_RECONFIG instance. Do not perform a read transaction in this mode if the `rate_switch_out` is not selected in the ALT2GXB_RECONFIG MegaWizard.

The dynamic rate switch has no effect on the dividers on the receive side of the transceiver channel. It can be used only for the transmitter.

You must be aware of the device operating range before you enable and use this feature. There are no legal checks that are imposed by the Quartus II software, since it is an on-the-fly control feature. You also need to ensure that a specific functional mode supports the data rate range before dividing the clock when using this rate switch option.

Reset Recommendations

Altera recommends that you follow a proper reset sequence during and after for PMA controls reconfiguration, channel reconfiguration, and dynamic transmit rate switching.

PMA Controls Reconfiguration

During the first time the dynamic reconfiguration controller initiates a read or write, for example to change or read the PMA controls (VOD, pre-emphasis, equalization, or DC gain), the transceiver channel switches permanently from the registers that contain static transceiver settings to registers that are written by the dynamic reconfiguration controller. Due to this asynchronous switching, there may a few bit errors and transitions in the transceiver status signals.

Therefore, perform a **one time** PMA control read or write transaction from the dynamic reconfiguration controller during system bringup (initialization). This operation sets the transceiver to listen to the registers written by the dynamic reconfiguration controller during system bringup. By performing this read or write transaction during system bringup, you avoid errors during normal system operation.

Channel Reconfiguration

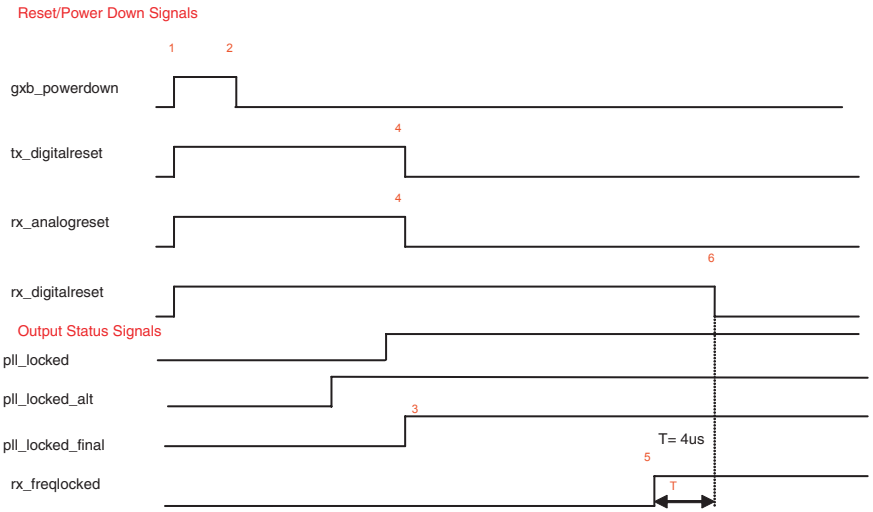
When you use the two TX PLLs in your design to reconfigure the channel, use the combination of the `pll_locked` and `pll_locked_alt` signals as part of your reset sequence.

In the two TX PLL designs, the updated TX PLL locked signal is:

`pll_locked_final* =` (`pll_locked` AND `pll_locked_alt`)

Figure 3–32 shows a waveform of the initialization and reset sequence of a design that uses main and alternate TX PLLs.

Figure 3–32. Reset Sequence (1 > 2 > 3 > 4 > 5 > 6)



The general reset sequence recommendations for bringing the device up are also valid.

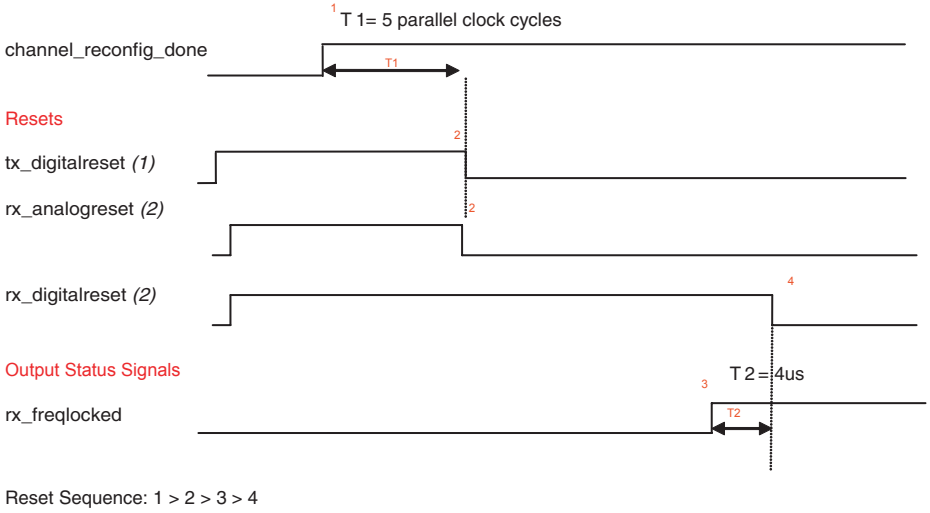
Refer to the Reset Control and Power Down section of the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Handbook* for more information.

In addition to the above recommendations, if you are using the Channel Reconfiguration feature, consider this additional recommendation:

If the channels are duplex and have individually switching modes using channel reconfiguration, consider the following:

- Reset controllers should be channel based
- The `channel_reconfig_done` signal can be used as a condition to reset the transmit and receive of the transceiver channel during and after channel reconfiguration
- Transmit digital resets (`tx_digitalreset`) are asserted during and after channel reconfiguration
- Assert the `rx_analogreset` signal and follow the reset sequence on the receiver side during and after the channel reconfiguration. Figure 3–33 shows the reset sequence for channel reconfiguration.

Figure 3–33. Reset Sequence (1 > 2 > 3 > 4) During and After Channel Reconfiguration



Notes to Figure 3–33:
(1) `tx_digitalreset` is valid in transmitter-only and duplex configurations.
(2) `rx_analogreset` and `rx_digitalreset` are valid in receiver-only and duplex configurations.

If the channel reconfiguration is done in a TX-only design, assert the transmit digital reset (`tx_digitalreset`) during and after channel reconfiguration, as shown in Figure 3–33. (`channel_reconfig_done` assertion signifies that the ALT2GXB_RECONFIG controller finished the channel reconfiguration by shifting an entire MIF into an ALT2GXB channel.)

If the channel reconfiguration is done in an RX-only design, assert the `rx_analogreset` signal and follow the reset sequence on the receiver side. Figure 3–33 shows the receiver sequence.

Dynamic Rate Switching

For a design using dynamic transmit rate switching, Altera recommends that you assert the `tx_digitalreset` when you initiate the rate switch operation until the `busy` signal goes low.

Overall Design Flow for Channel Reconfiguration

The following describes the design flow for Stratix II GX channel reconfiguration.

ALT2GXB Instantiation

1. Create an ALT2GXB MegaWizard instantiation. Select the protocol mode, single width, double width, data rate, and input reference clock frequency.
2. Select the required status signals.
3. In the **Reconfig** tab, select the channel internals.

If you intend to perform only rate division control, proceed to step 6.

4. If you would like to switch between two configurations that have different input clock frequencies, select the **Use alternate reference clock** option to configure the second TX PLL. Specify the **What is the logical reference clock index?** option value.
5. If the configuration requires different PLD interface widths or additional control signals provided in the **Reconfig2** tab, select the **Channel interface** option.
6. Select the appropriate clocking scheme in the **Reconfig2** tab.
7. Select the required additional control signals for the configuration in the **Reconfig2** tab (this is only enabled if the **Channel interface** option is selected).

MIF Generation:

8. Create a top-level design and connect the clock inputs in the RTL/schematic. Specifically, for the transceiver clock inputs, connect `pll_inclk` and `rx_cruclk` to the input pins that provide the clock for the protocol mode specified in the **General** tab of the ALT2GXB MegaWizard. Similarly, connect `pll_inclk_alt` and `rx_cruclk_alt` to the clock source that provides the clock for the protocol mode specified in the **Reconfig** tab of the ALT2GXB MegaWizard.

If you do not specify pins for `tx_dataout` and `rx_datain` for the transceiver channel, the Quartus II software selects a channel and generates a MIF for that channel. However, the MIF can still be used for any transceiver channel.

9. You can generate multiple MIF in the following two ways:

   Method 1:

- Compile the design created in step 8 and generate the first MIF.
- Update the ALT2GXB MegaWizard instance with the alternate configuration and connect the appropriate clock inputs, as mentioned in step 8.
- Compile the design to get the second MIF.

If you have to generate MIFs for many configurations, this method takes more time to complete.

Method 2:

- In the top-level design, instantiate all the different configurations of the ALT2GXB instantiation for which the MIF is required.
- Connect the appropriate clock inputs of all the ALT2GXB instantiations (see step 8).
- Generate the MIF. The MIFs are generated for all the ALT2GXB configurations.

This method requires attention when generating the MIF. Please check the following:

- The different ALT2GXB instantiations should have the appropriate **logical reference clock index** option values.
- The clock inputs for each instance should be connected to the appropriate clock source.
- When you generate the MIF, use proper naming for the files so you know the configuration supported by the MIF.

ALT2GXB_RECONFIG:

10. Create the ALT2GXB_RECONFIG instance. Select the **Channel reconfiguration** option to perform channel reconfiguration. Select `rate_switch_ctrl` for the transmit side data rate division.

11. Select the `reconfig_address_out` and `reset_reconfig_address` signals from the **Channel Reconfiguration** tab.

Control Logic for ALT2GXB_RECONFIG:

12. Implement logic to control the `ALT2GXB_RECONFIG` signals and to select the appropriate MIFs from memory and send the MIF data to ALT2GXB_RECONFIG.

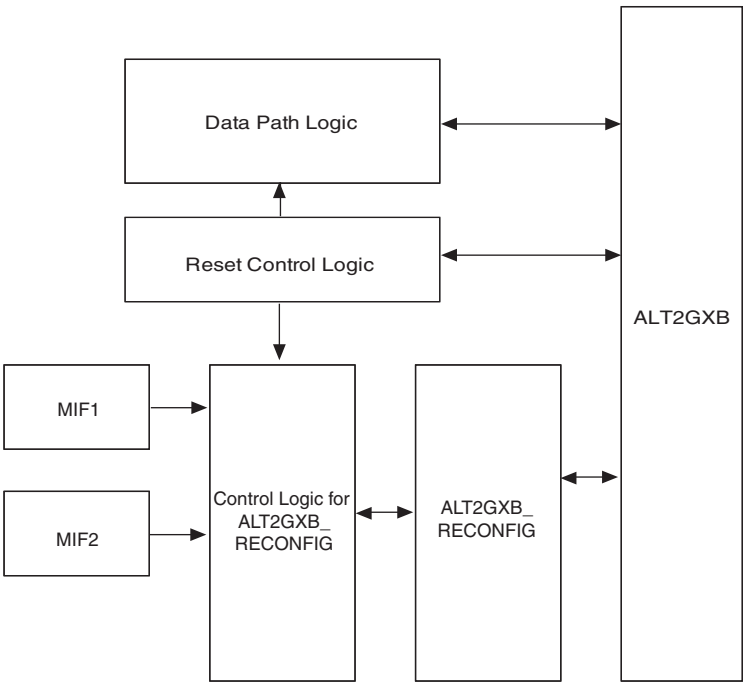
Data Path Logic:

13. Implement logic to handle the data from the transceiver. When you change the transceiver channel configuration, the data path, clocking, or PLD interface width may change. Therefore, implement logic in the PLD to transmit and receive data between the transceiver and the PLD logic, based on the transceiver configuration. Figure 3–34 shows the design functional blocks to perform channel reconfiguration.

Reset Control Logic:

14. Implement the reset control logic to handle the transceiver and the system resets. Refer to "Reset Recommendations" on page 3–66 for more information.

Figure 3–34. Functional Blocks for Channel Reconfiguration



Channel Reconfiguration Design Examples

This section provides three examples for performing dynamic reconfiguration on a transceiver channel.

Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode

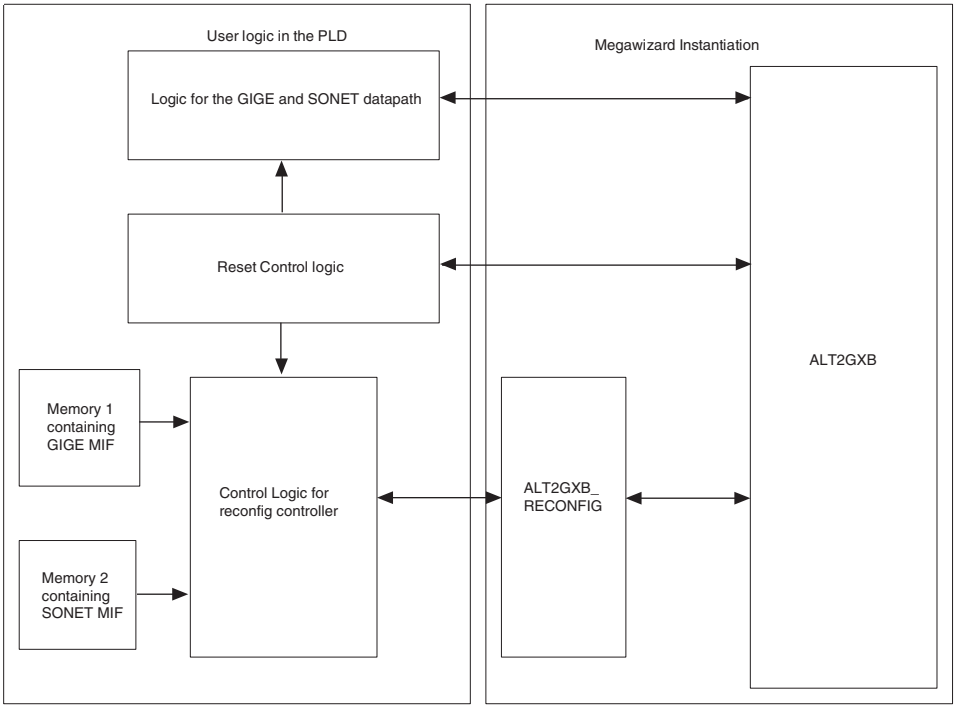
The GIGE mode in the ALT2GXB megafunction is different from the SONET/SDH OC48 mode in data path, clocking, and PLD interface width. The differences between the two modes are listed in Table 3–5.

Table 3–5. Differences Between GIGE and SONET/SDH OC48

Number	Functional Block	GIGE	SONET/SDH OC48
1	PLD width	8	16
2	8B/10B enabled	Yes	No
3	Rate matcher	Yes	No
4	Byte order block	No	Yes
5	Clock used for synchronizing the receive output data (`rx_dataout`)	`tx_clkout` (since rate matcher is used)	`rx_clkout`
6	Data rate	1.25 Gbps	2.488 Gbps
7	Allowed input reference clock	62.5 MHz 125 MHz	77.76 MHz 155.52 MHz 311.04 MHz 622.08 MHz
8	PCS-PMA interface width	10 (since data is 8B/10B encoded)	8

These differences determine the selection of parameters in the ALT2GXB MegaWizard and the required PLD logic to configure a transceiver channel between these two modes. Figure 3–35 shows the required functional blocks to perform channel reconfiguration.

Figure 3–35. Reconfiguring Between GIGE and SONET /SDH OC48 Modes



The discussion of the functional blocks is divided into four sections. The topics discussed in each section are as follows:

- Section I—Lists the steps to configure the ALT2GXB instance to generate the MIF for GIGE and SONET/SDH OC48 modes. Then lists the steps to create the ALT2GXB_RECONFIG instance.
- Section II—Sets up the control logic for the ALT2GXB_RECONFIG controller.
- Section III—Logic to process the GIGE and SONET/SDH data. This logic is required due to the differences in the data interface widths and the clocking between the two modes (shown in Table 3–5).
- Section IV—Resets the control logic to control the transceiver and system resets.

Section I
Use the following steps to generate a MIF for GIGE and SONET/SDH OC48 modes:

1. Generate the ALT2GXB instantiation for GIGE mode.
2. Generate the AL2GXB_RECONFIG instantiation.
3. Create a top-level design and generate the MIF for the GIGE protocol mode.
4. Modify the ALT2GXB instantiation for SONET /SDH OC48 mode.
5. Generate the MIF for SONET/SDH OC48 mode.
6. Initialize two memory elements with the MIF contents and write logic to select the MIF and to control the ALT2GXB_RECONFIG instance.

Step 1—Generate the ALT2GXB Instantiation for GIGE Mode:
This example shows the ALT2GXB instantiation for one-channel GIGE and SONET/SDH mode.

1. Set the protocol to **GIGE** mode in the first screen.

   Select the following control and status signals:

   - `rx_digitalreset`
   - `tx_digitalreset`
   - `rx_analogreset`
   - `rx_pll_locked`
   - `rx_freqlocked`

2. Add the other required status signals.

For a list of ALT2GXB signals and their functionality, refer to "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

3. In the **Reconfig** tab, set the following options:

   - If you need control of the transceiver PMA values, select the Analog PMA controls. For more information about PMA controls, refer to "Introduction" on page 3–1.

4. Select the **Channel internals** and **Use alternate reference clock** options. Selecting these options enable the second PLL for the SONET/SDH OC48 mode. A second PLL is needed because of the difference in the required input clock frequency and data rate between the GIGE and SONET/SDH OC48 modes (refer to rows 6 and 7 in Table 3–5).

5. Set the **What is the protocol to be reconfigured?** option to **SONET/SDH**.

6. Set the **sub protocol** option to **OC48** (see Figure 3–31).

7. Select the **input clock frequency** and **alternate transmitter PLL bandwidth mode** options based on the requirements. The allowed reference clock input frequencies for SONET/SDH OC48 are specified in row 7 of Table 3–5.

8. For the **What is the logical reference index?** option, select **1** or **0**. The Quartus II software uses the logical reference index to select the PLL clock outputs for the transmit and receive channels when configured to SONET/SDH OC48 protocol. The MUX values selected for the GIGE and SONET/SDH OC48 modes should be different.

For example, if you select **1** for the **What is the logical reference index?** option for the SONET/SDH OC48 mode, you should select **0** for GIGE mode. If you select the same values for the two modes, the transceiver behavior after reconfiguration becomes unpredictable.

9. Select the **Channel interface** option. Selecting **Channel interface** creates the data interface signals `tx_datainfull` and `rx_dataoutfull` that are comprised of control and data signals. This selection is required because of the differences in the PLD interface width between the GIGE and SONET/SDH modes (row 1 in Table 3–5). The description of individual bits of `tx_datainfull` and `rx_datainfull` are provided in "Channel Interface" on page 3–53.

10. In the **Reconfig2** tab, under the **How should the receivers be clocked?** option, check the **Use the respective channel core clocks** option. Selecting this option creates the `rx_clkout` port. Select this option because of the clocking differences between the two modes (row 5 of Table 3–5). Therefore, the PLD logic can clock the receive output of the ALT2GXB with `rx_clkout` for SONET/SDH mode and `tx_clkout` for the GIGE mode.

11. In the **How should the transmitters be clocked?** option, select any option. Since this example assumes a one-channel reconfiguration in the transceiver block, the above options will not make a difference. However, if the number of channels used in channel reconfiguration is more than one, Altera recommends you select the **share single transmitter core clock between transmitters** option to conserve clock routing resources.

12. Select signals in the **check a control box to use the corresponding control fields** option based on the requirements. The signals in this tab can be selected only if the **Channel interface** option is enabled in the **Reconfig** tab. For this example, select the signals `rx_byteorderalignstatus` and `rx_a1a2sizeout`, since these signals are required for SONET/SDH OC48 mode.

Some of the signals are meaningful only for the modes for which they are intended. For example, the `rx_byteorderalignstatus` signal is only meaningful in SONET/SDH OC48 mode. PLD logic should not use these signals for GIGE mode.

For more information about the protocol-specific ALT2GXB interface signals, refer to "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

13. In the subsequent tabs, select the required signals and complete the MegaWizard instantiation.

Step 2—Generate the ALT2GXB_RECONFIG Instantiation:

1. Set the **What is number of channels controlled by the reconfig controller?** option to **1**.

2. Select **Analog controls** to modify the PMA values, if desired.

3. Select **Channel reconfiguration**. This selection is required to perform a channel reconfiguration.

4. Select the required signals under the **write control** and **read control** options, if the **Analog controls** option in screen 1 is selected.

Refer to the *ALT2GXB_RECONFIG Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about write-control and read-control signals.

5. In the **channel reconfiguration** page, select the `reconfig_address_out`. This signal increments by 1, from 0 to 27, then starts at zero again. The other available control signals in the MegaWizard are `reconfig_address_en` and `reset_reconfig_address`. Selecting these signals is optional. The timing and description of these signals are provided in "Section II—Control Logic for the RECONFIG Controller:" on page 3–84.

6. Complete the ALT2GXB_RECONFIG MegaWizard instantiation.

Step 3—Create a Top-Level Design and Generate the MIF for GIGE Protocol Mode:

Clock input connections for the ALT2GXB megafunction are listed below. The clock source should feed the following clock inputs:

- GIGE mode—`pll_inclk` and `rx_cruclk` inputs.
- SONET/SDH OC48 mode—`pll_inclk_alt` and `rx_cruclk_alt` inputs.

1. Since GIGE is the protocol mode you selected in the first page of the ALT2GXB MegaWizard, the Quartus II software requires the GIGE clock source to be connected to `pll_inclk` and `rx_cruclk` inputs.

2. Connect the `cal_blk_clock` input of the ALT2GXB instance to a clock source.

Refer to the "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for the `cal_block_clk` signal requirements.

3. Connect the `tx_dataout` and `rx_datain` ports to the top-level module. This is required for the Quartus II software to compile successfully. To generate the MIF, connecting the other input and output ports of the ALT2GXB instance is not mandatory.

4. Assign pins for the clock ports (`pll_inclk`, `rx_cruclk`, `pll_inclk_alt`, and `rx_cruclk_alt`). If pin assignments are not made for the `tx_dataout` and `rx_datain` ports of the ALT2GXB instantiation, the Quartus II software automatically selects pins for these ports and names the MIF with the pin name extension. The MIF can still be used by any physical transceiver channel to perform reconfiguration.

After compilation of the design, the Quartus II software creates the MIF in the **reconfig_mif** folder under the project directory. Copy the MIF and save it in a separate folder. Otherwise, the new MIF that is generated for the SONET/SDH mode will overwrite the current MIF.

Step 4—Modify the ALT2GXB Instantiation for SONET/SDH OC48 Mode:

1. To create a MIF for the SONET/SDH OC48 mode, either modify the existing ALT2GXB instantiation created for GIGE mode or create a new instantiation for SONET mode. However, the first method is easier since it does not require major RTL or schematic changes.

2. Open the existing ALT2GXB instantiation. Select the **Which protocol you will be using?** option and set it to **SONET/SDH**. Set the **sub protocol** option to **OC48**. All the other signals selected for GIGE mode should not be changed.

3. In the **Reconfig** tab, select the **Channel internals** and **alternate reference clock** options. In the protocol section, select **GIGE**. Select the same input clock frequency selected in step 1.

4. For the **logical reference clock index** option, choose the complement of what you selected in step 10.

5. Select the channel interface and complete the instantiation.

Step 5—Generate the MIF for SONET/SDH OC48 Mode:

1. Before compiling the design, in the RTL or schematic, connect the `pll_inclk` and `rx_cruclk` to the clock source that provides the SONET/SDH OC48 clock. Similarly, connect `pll_inclk_alt` and `rx_cruclk_alt` to the clock source that provides the GIGE clock.

The Quartus II software generates the new MIF in the **/reconfig_mif** directory.

Step 6—Initialize Two Memory Elements with the MIF Contents and Write Logic to Select the MIF and to Control the ALT2GXB_RECONFIG:

1. Create two memory elements, each 16-wide and 28-bits deep. The memory elements can be a RAM/ROM inside or outside the Stratix II GX device. Assign the two MIFs to each of these memory elements.

Section II—Control Logic for the RECONFIG Controller:
The control logic block is required to perform the following functions:

- Select the memory to configure a channel to the GIGE or SONET/SDH mode.
- Control the reconfiguration mode (namely the PMA mode or the channel configuration mode).
- Control the read and write signals to the ALT2GXB_RECONFIG megafunction based on the `busy`, `data valid`, and `address_out` signals.

The following is an example flow of channel reconfiguration by writing the MIF contents of memory location 1 to the reconfig controller:

1. Set the `reconfig_mode_sel` to **-001**.
2. Select the data input from memory location 1.
3. Wait until the `busy` signal from the ALT2GXB_RECONFIG megafunction is low.
4. Check whether the `reconfig_address_out` is less than 28 decimals.
5. Wait for the data out from memory location 1 corresponding to the new `reconfig_address_out` becomes available at the `reconfig_data` port before asserting the `write_all` signal.

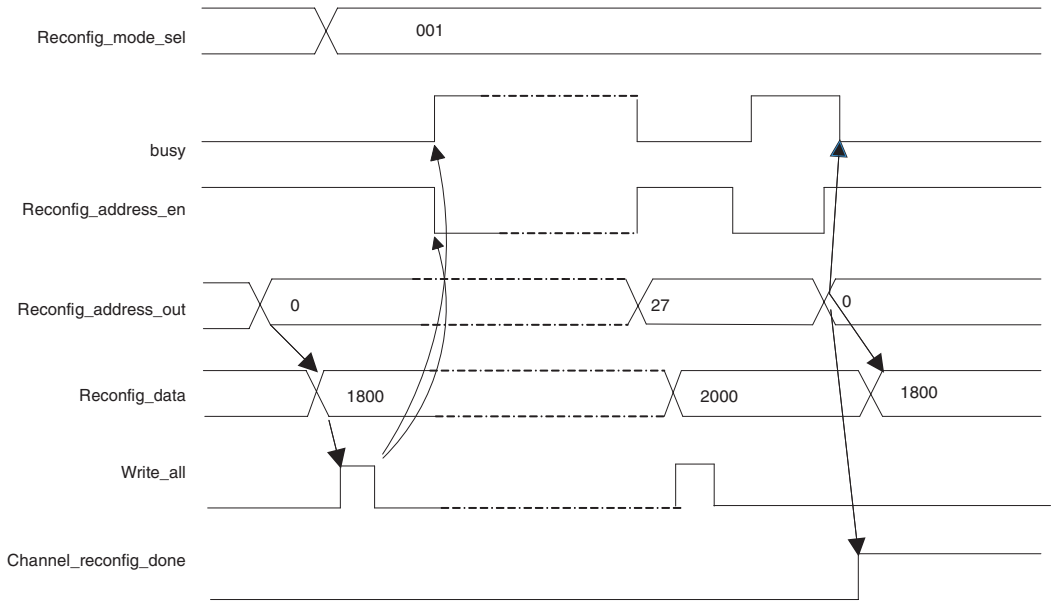
Figure 3–36 shows the various signal transitions during channel reconfiguration.

The ALT2GXB_RECONFIG megafunction provides these additional signals:

- `reconfig_address_en`
- `reset_reconfig_address`

The ALT2GXB_RECONFIG megafunction asserts the `reconfig_address_en` signal to indicate that the `reconfig_address_out` has changed. In test designs, the `reconfig_address_out` was monitored to determine the end of the write operation. You can also use the `channel_reconfig_done` signal to determine the end of the write operation. The `channel_reconfig_done` signal goes high one clock cycle after the `reconfig_address_out` signal changes from 27 back to 0.

Figure 3–36. Timing of the ALT2GXB_RECONFIG Signals



Section III—Logic and Clocking for the GIGE and SONET/SDH OC48 Datapath:

In the ALT2GXB MegaWizard, the channel interface that created `tx_datainfull` (44-bits wide) and `rx_dataoutfull` (64-bits wide) was selected. In addition, the `rx_byteorderalignstatus` and the `rx_a1a2size` signals were selected. The PLD logic should selectively use some of these signals based on whether the transceiver channel is configured in GIGE mode or SONET/SDH OC48 mode.

Table 3–6 provides descriptions for the tx_datainfull and rx_dataoutfull signals for GIGE and SONET/SDH OC48 modes.

Table 3–6. PLD Interface Signals—GIGE and SONET/SDH OC48 Modes

Signal Name	Description
GIGE MODE	
tx_datainfull[7:0]	8-bit unencoded data input to the transceiver channel
tx_datainfull[8]	tx_ctrlenable (control signal K/D)
rx_dataoutfull[7:0]	8-bit unencoded data output from the transceiver channel
rx_dataoutfull[8]	rx_ctrldetect (control signal K/D)
rx_dataoutfull[9]	rx_errdetect
rx_dataoutfull[10]	rx_syncstatus
rx_dataoutfull[11]	rx_disperr
rx_dataoutfull[12]	rx_patterndetect
SONET/SDH OC48 MODE	
tx_datainfull[7:0]	LSB data input to the transceiver channel
tx_datainfull[29:22]	MSB data input to the transceiver channel
rx_dataoutfull[7:0]	LSB data output from the transceiver channel
rx_dataoutfull[29:22	MSB data output from the transceiver channel
rx_dataoutfull[10], rx_dataoutfull[42]	rx_syncstatus[1:0]
rx_dataoutfull[12], rx_dataoutfull[44]	rx_patterndetect[1:0]

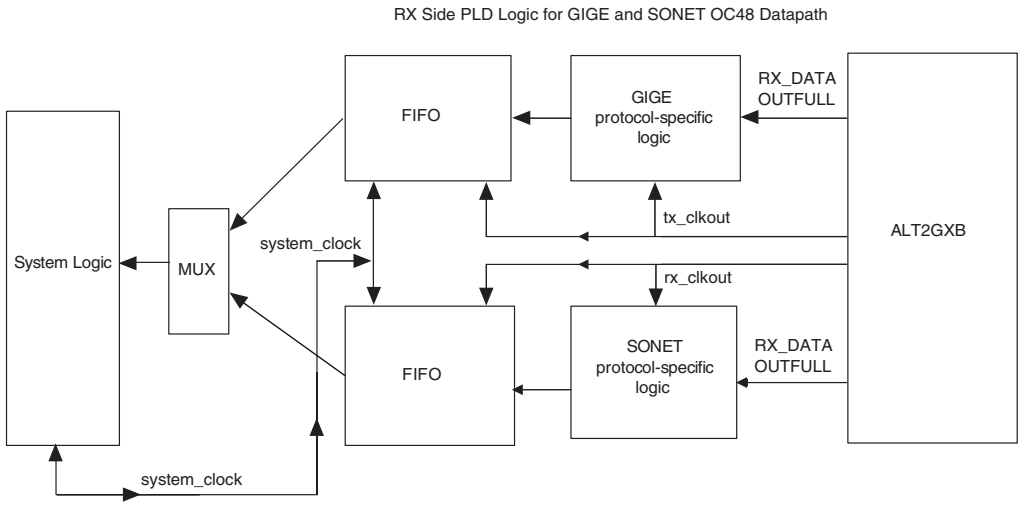
Clocking

For the transmit side, the PLD logic for the SONET/SDH OC48 and GIGE modes sends the data synchronized to the tx_clkout signal. Therefore, the clocking for the transmit side remains the same for the two modes.

For the receive side, the data and status signals from the ALT2GXB megafunction for the GIGE mode is synchronized to tx_clkout since rate matching is used. For the SONET/SDH OC48 mode, the signals are synchronized to rx_clkout. Therefore, the PLD logic has two functional protocol-specific logic blocks to handle data for the GIGE and SONET/SDH OC48 modes. Based on the configured protocol mode, the receive side logic selects the appropriate data path.

Figure 3–37 shows the block diagram for a receive side PLD logic to handle the GIGE and SONET/SDH OC48 datapath.

Figure 3–37. RX PLD Logic to Process GIGE and SONET/SDH OC 48 Data



Section IV—Reset Control Logic:
The reset control sequence for channel reconfiguration (explained in "Reset Recommendations" on page 3–66) must be followed during and after the channel configuration process. In addition, when resetting the transceiver channel, the reset control logic should reset the data path in the PLD logic to clear the error data received during the reconfiguration process.

For a PMA-only configuration (for example, changing the V_{OD}, equalization, DC gain, or pre-emphasis), the transceiver channel or the datapath in the PLD logic does not require a reset after reconfiguration. Reset is required only for channel reconfiguration or rate switch.

Simulation:
To simulate channel reconfiguration, some simulation tools only allow **.ram** or **.hex** files to initialize the memory. To convert the generated MIF to a **.hex** file, open the **.mif** in the Quartus II software and save it as a **.hex** file. Initialize the memory elements with the **.hex** file to simulate the design.

Example 2—Channel Configuration Between a Basic Mode Configured for 3.125 Gbps and a Basic Mode Configured for 2.000 Gbps

The PCS functional blocks and the PLD interface (16 bits) is the same for both modes. Given that the functional blocks are the same, to achieve the two data rates mentioned above, use two different input reference clock frequencies for the two modes. Table 3–7 shows the transceiver configuration for the two modes.

Table 3–7. Differences Between Two Basic Modes

Number	1 Channel Basic Mode 1	1 Channel Basic Mode 2
Data rate	3125	2000
Input reference clock frequency	156.25 MHz	125 MHz
PLD interface width	16	16
8B/10B enabled	Yes	Yes
Rate matcher	No	No
Clock used for synchronizing the receive output data (`rx_dataout`)	`rx_clkout`	`rx_clkout`

The description for this design example is divided into four sections:

- Section I—Steps to Create the MIF for the Two Transceiver Modes
- Section II—Steps to Create the ALT2GXB_RECONFIG Instantiation
- Section III—Sets Up Control Logic in the PLD for ALT2GXB_RECONFIG
- Section IV—Resets Control Logic

Section I—Steps to Create the MIF for the Two Transceiver Modes:

1. In the first page of the ALT2GXB MegaWizard, complete the following:
   - Set Mode to **Basic**
   - Select **single width**
   - Set the channel width to **16**
   - Set the number of channels to **1**
   - Set the data rate to **3.125 Gbps**.
2. Set the input reference clock to **156.25 MHz**.
3. Select all the resets, `pll_locked`, `rx_freqlocked`, and other required status signals.

4. In the **Reconfig** tab, select **Analog controls** if you wish to modify the PMA values dynamically (V_{OD}, pre-emphasis, DC gain, and equalization).

5. Select the **Channel internals** and **Use Alternate reference clock** options. These options must be selected since Basic mode 2 requires a different input clock frequency.

6. Set the **protocol** option to **Basic**, set the data rate to **2.000 Gbps**, and set the **input clock frequency** option to **125 MHz**. For this example, since both the configurations have the same PLD interface width and functional blocks (Table 3–7), you do not need the **Channel interface** option.

7. In the **Reconfig 2** tab, select the **Use respective receiver core clocks** and **Use respective transmitter core clocks** options (row 6, Table 3–7).

8. Complete the ALT2GXB MegaWizard instantiation.

Section II—Control Logic for the RECONFIG Controller:
Follow the same procedure as described in "Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode" on page 3–72.

Section III—Logic for the Basic Mode 1 and Basic Mode 2 Datapath:
Clock the data to the transceiver with the `tx_clkout` signal on the transmitter side for both configurations. Similarly, for the receive side, clock the data from the transceiver with the `rx_clkout` signal for both configurations. A single logic block can handle data processing when the transceiver channel is configured between these two modes.

Section IV—Reset and Control Logic:
Follow the same procedure as described in "Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode" on page 3–72.

Example 3: Dynamic Rate Switch for the Transmit Side Design Example

This design example explains the steps to dynamically divide the transmit data rate of a transceiver channel by 4, 2, or 1 without requiring MIF generation. The ALT2GXB_RECONFIG instance provides a `rate_switch_ctrl` signal for this purpose.

Use the rate_switch_ctrl signal only for dividing the data rate of the transmit side. To divide the data rate for both the transmit and receive sides, a MIF-based approach is required.

This example uses a Basic mode with 8B/10B enabled running at 4 Gbps data rate. You can configure the mode dynamically between 4.25 Gbps, 2.25 Gbps, and 1.125 Gbps.

The description for this design example is divided into the following sections:

- Section I—Create the ALT2GXB Instantiation for the Transceiver Channel
- Section II—Create the ALT2GXB_RECONFIG Instantiation
- Section III—Create the Top-Level Design

Section I—Create the ALT2GXB Instantiation for the Transceiver Channel:

1. Create a **Basic mode** by setting the operation mode to **Transmit and Receive**.

2. Select **double width** mode. This is required since the highest data rate in this example is 4.25 Gbps (**Single width** can be selected only up to 3.125 Gbps). Set the channel width to **32**. The lowest PLD frequency allowed in the Quartus II software is 25 MHz. Therefore, the transceiver runs at 1.125 Gbps with a 32-bit PLD interface. The PLD clock frequency in this case is 26.5 MHz (1125/40 = 26.5).

3. Set the input frequency to **106.25 MHz**.

4. In the **Reconfig** tab, check the **Channel internals** option. This is required to enable the ALT2GXB_RECONFIG instance to modify the channel local divider values dynamically. The alternate reference clock is not required since one clock source is used. Also, the data rates can be derived from the 106.25 MHz clock.

5. Complete the ALT2GXB MegaWizard instantiation.

Section II—Create the ALT2GXB_RECONFIG Instantiation:

1. Instantiate the ALT2GXB_RECONFIG megafunction as described in "Example 1—Configuring a Transceiver Channel Between GIGE Mode and SONET/SDH OC48 Mode" on page 3–72.

2. Select the **Modify the data rate using the local divider** option in the **Reconfiguration Settings** tab. This creates the rate_switch_ctrl and rate_switch_ctrl_out signals.

Table 3–8 shows the values for each of the rate_switch_ctrl settings.

Table 3–8. Rate Switch Control Signal Settings

rate_switch_ctrl[1:0] Settings	Local Divider Value in the Transmit Channel
00	1
01	2
10	4
11	Not applicable

3. Complete the ALT2GXB_RECONFIG MegaWizard instantiation.

Section III: Create the Top-Level Design
Create the ALT2GXB_RECONFIG instance control logic, reset control logic, and the PLD logic to handle the data path. Refer to "Reset Recommendations" on page 3–66 for information on transceiver resets.

4. Create the top-level design and connect the functional blocks.

Pseudo-Write Sequence for Simulating Channel Reconfiguration

If you are simulating channel reconfiguration, consider a case where you are using multiple transceiver channels in your design driven by a single dynamic reconfiguration controller. When you first perform channel reconfiguration on a transceiver channel, rx_freqlocked and rx_clkout of all channels that are connected to the reconfiguration controller go to 0 for a few clock cycles. This occurs because the receive PLLs in the simulation model require a relock when channel reconfiguration is enabled.

This issue happens only in simulation the first time you initiate channel reconfiguration.

To work around this issue, perform the following one-time write sequence as part of your system initialization when you assert the gxb_powerdown or rx_analogreset signals.

The signals that are referred to in the following write sequence correspond to the input and output ports of the ALT2GXB_RECONFIG instantiation in your design.

1. Set the reconfig_mode_sel signal to **001**. Write the default **.hex**/**.mif** file contents for two reconfig_address_out signal increments. That is, pulse the write_all signal for the reconfig_address_out 0 and 1 based on the busy and reconfig_address_en signals.

2. The **.hex**/**.mif** file selected for writing should correspond to the default configuration in the ALT2GXB MegaWizard. For example, if you have two **.hex**/**.mif** files that correspond to GIGE and SONET/SDH OC48 protocols, and if you have set GIGE as your default configuration (the protocol set in the **General** tab of the ALT2GXB MegaWizard), write the first two words of the **.hex**/**.mif** file generated for GIGE protocol.

3. After you complete writing the first two words, wait for the busy signal to go low and assert the reset_reconfig_address signal to initialize the reconfig_address_out to 0.

Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration

Introduction

The Stratix II GX transceiver can be dynamically reconfigured to various protocols and data rates. This section discusses the dynamic reconfiguration features introduced in Quartus II software version 7.1. Altera assumes you have prior knowledge about the dynamic reconfiguration controller architecture (refer to "Dynamic Reconfiguration Controller Architecture" on page 3–2), the Stratix II GX transceiver architecture, and the memory initialization file (**.mif**, also known as MIF) flow (refer to "MIF Generation in Quartus II Software" on page 3–32) for dynamic reconfiguration.

Synopsis of Existing Dynamic Reconfiguration Features

In the Quartus II software version 7.0 and earlier, the following dynamic reconfiguration features were available:

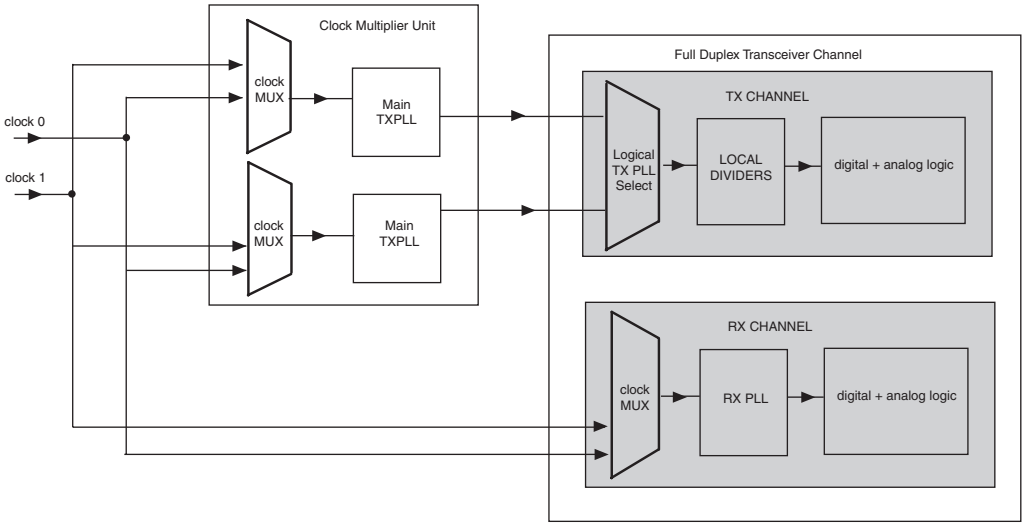
- PMA reconfiguration—to control voltage output differential (V_{OD}), pre-emphasis, equalization, and DC gain.
- Channel reconfiguration—to dynamically reconfigure the transceiver data rates, protocol modes, or a combination of these two options. This method requires a MIF to reconfigure a channel.
- Dynamic transmit rate switch—to dynamically reconfigure the transmit data rate by changing the local divider settings in the transmit side. The available local divider options are by /1, /2, or /4. This method does not require a MIF for reconfiguration.

This option can only change the transmitter data rate and not the receiver data rate.

Figure 3–38 shows the transceiver blocks (grayed out) that can be dynamically reconfigured using the channel reconfiguration feature.

Figure 3–38 shows that the TX PLLs and the clock multiplexer on the transmit side (inside the clock multiplier unit) could not be reconfigured using these features. It also shows that only two sources of input reference clocks were available for the TX PLLs and RX PLLs.

Figure 3–38. Reconfigured Functional Blocks with Channel Reconfiguration *Note (1)*



Note to Figure 3–38:
(1) Supported from the Quartus II software version 6.1.

Overview of Quartus II Software Version 7.1 Features for Dynamic Reconfiguration

The Quartus II software version 7.1 provides the following enhancements to support dynamic reconfiguration:

- Three additional features to dynamically reconfigure the transceiver channel and the TX PLLs:
  - TX PLL-only reconfiguration
  - Channel and TX PLL reconfiguration
  - Channel reconfiguration with TX PLL select
- The number of possible clock sources for the input reference clocks is increased from two to five.

- Settings to generate a MIF with 38 words (required for the features mentioned above). You will need this MIF just as you needed one in the channel reconfiguration feature (see "Channel Reconfiguration" on page 3–30).

To write the MIF, follow the same method used for channel reconfiguration (see "Channel Reconfiguration" on page 3–30 for more information). The functionality of all other signals, such as `write_all`, `channel_reconfig_done`, `reconfig_address_en`, `logical_channel_address`, `data_valid`, and `busy` is the same as that of the channel reconfiguration feature.

These new features provide flexibility to reconfigure a TX PLL to multiple data rates, dynamically switch the transmit channel to listen to any of the two TX PLLs, and to configure a transceiver channel. Using these enhancements, you can use the Stratix II GX transceiver to dynamically support multiple protocols and data rates. In the following sections, the new dynamic reconfiguration features and the different software settings required to implement these features are discussed in detail.

Conventions Used

Throughout this document, the following conventions are used:

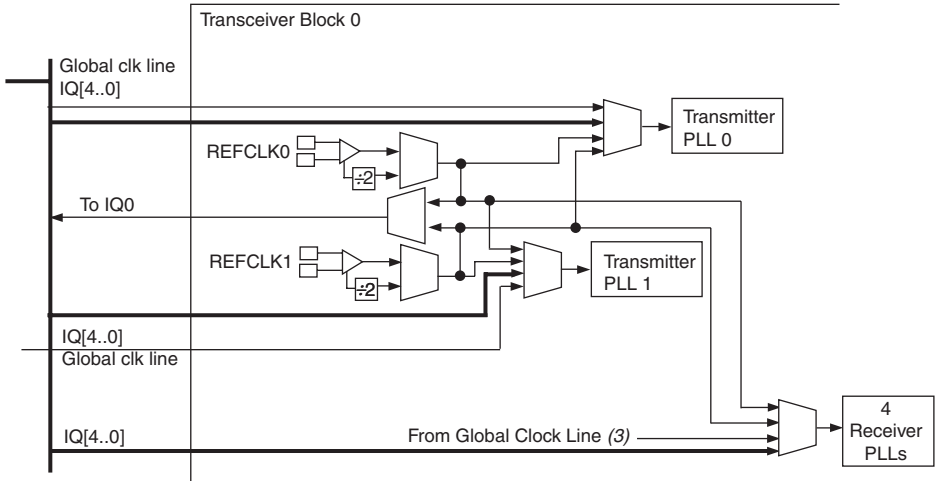
- Channel and CMU PLL reconfiguration—refers to the three dynamic reconfiguration features introduced in the Quartus II software version 7.1.
- Channel and TX PLL reconfiguration—refers to one of the features in Channel and CMU PLL reconfiguration.
- Channel—refers to the transceiver channel with digital and analog functional blocks.
- Main TXPLL— refers to the TX PLL that is configured in the **General** tab of the ALT2GXB MegaWizard.
- Alternate TXPLL—refers to the TX PLL that is configured in the **Reconfig Alt PLL** tab of the ALT2GXB MegaWizard.
- logical tx pll—refers to the logical identification value 0 or 1, assigned to the main and alternate TXPLLs. You set this value in the **Reconfig Clks 1** and **Reconfig Alt PLL** tabs of the ALT2GXB MegaWizard.
- Reconfig controller—refers to the dynamic reconfiguration controller that you instantiate using the ALT2GXB_RECONFIG MegaWizard. In this document, reconfig controller and ALT2GXB_RECONFIG are used interchangeably.

Clocking Enhancements and Requirements

To configure the TX PLLs and RX PLLs for multiple data rates, it is important to understand the input reference clock requirements. This helps you to efficiently create the clocking scheme for reconfiguration and to reuse the MIFs across all channels in the device. The new clocking enhancements and the implications of using input clocks from various clock sources are reviewed in this section.

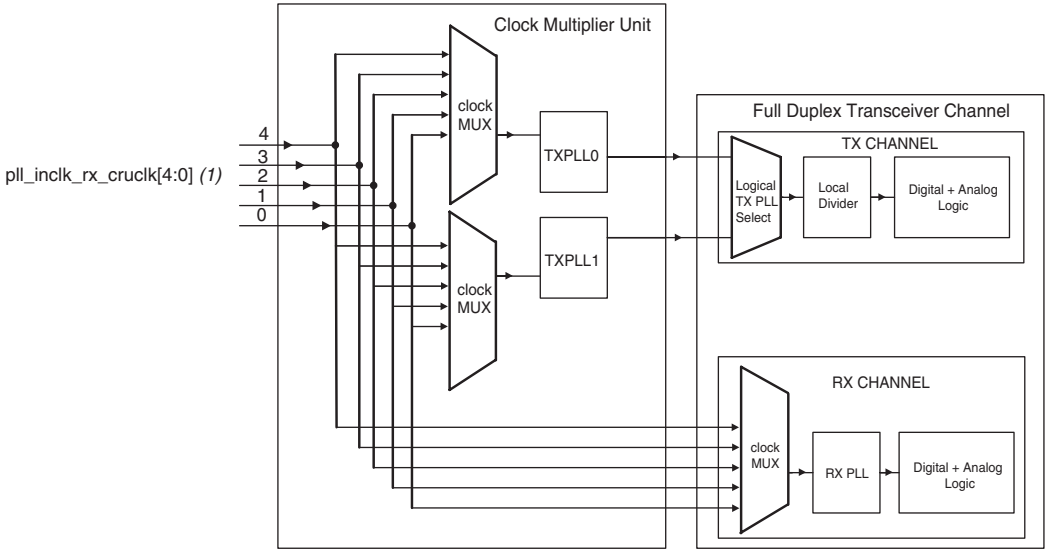
When you enable the **Channel and CMU PLL Reconfiguration** option in the ALT2GXB MegaWizard (by selecting the **Enable Channel and Transmitter PLL Reconfiguration** option in the **Reconfig** tab), the Quartus II software version 7.1 allows a maximum of five possible sources available for input reference clocks. Figure 3–39 shows the different clock sources that connect to the transceiver block.

Figure 3–39. Transceiver Block with Global Clock Line Connections



These five clock inputs appear as a `pll_inclk_rx_cruclk[]` port and can be provided from the inter transceiver block lines, also referred as Inter Quad (IQ) lines, or from the global clock networks that are driven by an input pin. Figure 3–40 shows the reference clock connections to TX PLLs and RX PLLs in a transceiver channel.

Figure 3–40. Input Reference Clock Connections to the Transceiver Channel



Note to Figure 3–40:
(1) These clocks can be provided from IQ lines, global clock networks, or dedicated local `refclks`.

Figure 3–40 shows the same input reference clocks connected to both the TX PLLs and RX PLLs. If you enable the **Channel and CMU PLL Reconfiguration** option for a full-duplex configuration, you cannot provide separate reference clocks to the TX PLLs and RX PLLs.

When you use the global clock line to provide input reference clocks, be aware of the following restrictions and implications:

- The hardware allows only one global clock input for the two TX PLLs in a transceiver block (refer to Figure 3–39).
- In a receiver-only channel configuration, the RX PLL of each channel in a transceiver block can be clocked by an independent global clock line. But, if you connect different clock input pins to the RX PLL in each channel, you cannot reuse the MIFs between these two channels. This constraint is explained further in "Input Reference Clock Requirements for Reusing MIFs" on page 3–94.
- Each global clock line consumes a local route input output (LRIO) resource. Since each transceiver block has a fixed LRIO resource, using the global clock line may restrict the number of clocks you can provide to the transceiver channels in your design.

For more information on LRIO resource limitation, refer to the PLD Clock Resource section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Using Dedicated refclks

When you use dedicated `refclks` as input reference clocks, the `refclk` pre-divider is required if **one** of the following conditions is satisfied:

1. If the input clock frequency is greater than 325 MHz.
2. For functional modes with a data rate less than 3.125 Gbps (the data rate is specified in the **what is the data rate?** option in the **General** tab of the ALT2GXB MegaWizard). The TXPLL and RXPLL is configured with the data rate that you set in this option.
   - If the input clock frequency is greater than or equal to 100 MHz AND
   - If the ratio of data rate to input clock frequency is 4, 5, or 25
3. For functional modes with an data rate greater than 3.125 Gbps:
   - If the input clock frequency is greater than or equal to 100 MHz AND
   - If the ratio of data rate to input clock frequency is 8, 10, or 25

When you use the channel and CMU PLL reconfiguration feature, you can dynamically reconfigure the TX PLLs and the RX PLL in a transceiver block from any of the five available input reference clocks. The Quartus II software automatically instantiates the `refclk` pre-divider (if one of the above mentioned conditions is satisfied) for the clock sources that drive the main and alternate TX PLL.

You specify the information for main and alternate TX PLL in the **General** and **Reconfig Alt PLL** tabs, respectively.

For the other clock inputs, the ALT2GXB MegaWizard provides optional options in the **Reconfig Clks 1** and **Reconfig Clks 2** tabs to specify whether a `refclk` pre-divider should be instantiated by the Quartus II software. The available options are:

a. **what is the reconfig protocol driven by clock x?** ("x" can be 0, 1, 2, 3, 4)

b. **what is clock x input frequency?**

c. **use clock x reference clock divider?**

For specific option values in **a** and **b** (mentioned above), the Quartus II software automatically instantiates the `refclk` pre-divider (field **c** is automatically selected in the ALT2GXB MegaWizard). For example, when you select the **PCI Express (PIPE)** option in the **what is the reconfig protocol driven by clock0** field, the Quartus II software automatically instantiates the `refclk` pre-divider for clock source 0.

When you select values in **a** and **b** (mentioned above) for a clock source and if field **c** is enabled, determine whether the data rate or the input clock frequency for the clock input meets one of the three conditions mentioned above. If one of the conditions is met, select field **c**.

For example, if you intend to use clock `source1` to reconfigure the channel to Basic mode with a 100 MHz input reference clock and a data rate of 2500 MHz, clock `source1` satisfies condition 2 mentioned above. In the **Reconfig Clk 1** tab, set the following values for clock `source1`:

- set the **what is the reconfig protocol driven by clock1?** option to **Basic**
- set the **what is clock 1 input frequency?** option to **100 MHz**
- select the **use clock1 reference clock divider?** option (this enables the Quartus II software to instantiate the `refclk` pre-divider for clock `source1`).

You can also use a dedicated `refclk` input pin from an unused transceiver block. If the Quartus II software creates a pre-divider for this clock input, it automatically feeds the output of the pre-divider to the TX PLLs and RX PLLs of other transceiver blocks. The `refclks` do not use the LRIO resource.

Refer to Figure 2–4 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information.

The `refclk` pre-divider is not part of the information stored in the MIF. It is a static setting created during the ALT2GXB MegaWizard configuration.

The above mentioned clocking scheme also determines whether you can reuse the MIF across all transceiver channels in your device. In the following section, the clocking requirements to reuse MIFs are discussed.

Input Reference Clock Requirements for Reusing MIFs

The MIF contains information about the input clock multiplexer and the functional blocks that you selected during the ALT2GXB MegaWizard instantiation. The Quartus II software generates a MIF for each channel. This MIF can be used in any of the other channels in the device if you satisfy the following two requirements for the input reference clocks:

- The order of the clock inputs must be consistent. For instance, assume that a MIF is generated for a transceiver channel in bank 13 and the clock source is connected to the `pll_inclk_rx_cruclk[0]` port. When the generated MIF is used in a channel in other transceiver blocks (for example, bank 14), the same clock source needs to be connected to the `pll_inclk_rx_cruclk[0]` port. Figures 3–41 and 3–42 show the incorrect and correct order of input reference clocks, respectively.
  - In Figure 3–41, the clocking is incorrect to reuse the MIF because the input reference clock is not connected to the corresponding `pll_inclk_rx_cruclk[]` ports in the two instances.

Figure 3–41. Incorrect Input Reference Clock Connection to Reuse the MIF

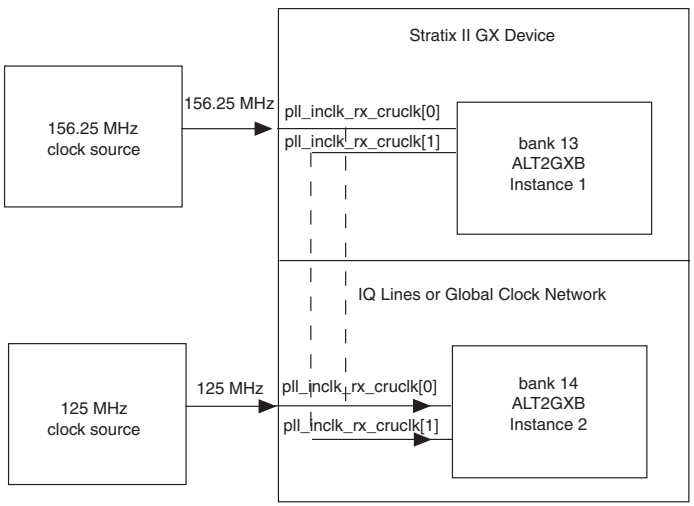
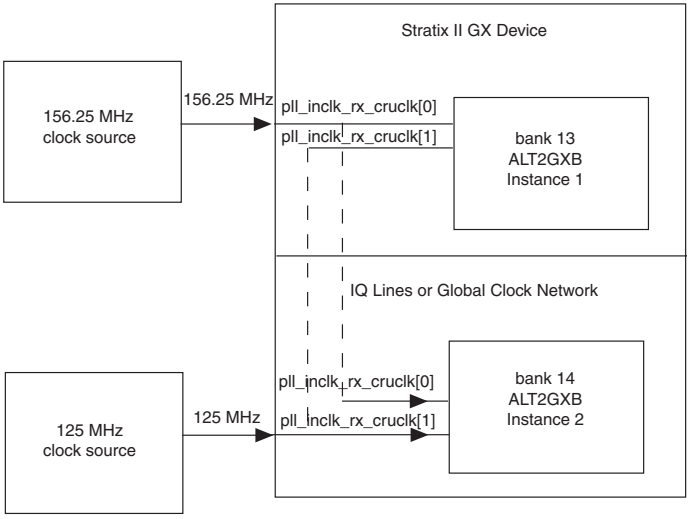


Figure 3–42. Correct Input Reference Clock Connection to Reuse the MIF



- If you connect the input reference clock ports of the ALT2GXB instances through different input pins, you cannot reuse the MIF generated between these two instances, even if you provide the same clock frequency on these two pins. For example, in Figure 3–43 the clock source provides 156.25 MHz clock to instance1 and instance2 through two different pins. In this case, if you generate a MIF for instance1, you cannot reuse it in instance2.
  - When you try to reconfigure using the MIF for instance1 in a transceiver block (for example, bank 13) on instance2 in another transceiver block (for example, bank 14), the reconfig controller remaps the clock input multiplexer information in the MIF (generated for instance1) to correspond to instance2. During this translation process, it assumes that the same clock input is connected to the `pll_inclk_rx_cruclk[]` port. Therefore, the reconfig controller selects the clock multiplexer value for the IQ line or global clock network that connects to the clock input of instance1.

If you want to reuse the MIF, connect the clock source to only one clock pin in the device. In your design, connect the clock input port of your transceiver instances to that clock pin. The Quartus II software automatically routes the clock input to all the transceiver blocks through IQ lines or global clock routing resources, depending on whether you

selected a dedicated refclk pin or a clock I/O pin. Figure 3–43 shows the incorrect clocking scheme. Figure 3–44 shows the correct clocking scheme.

Figure 3–43. Incorrect Clocking Scheme to Reuse MIF

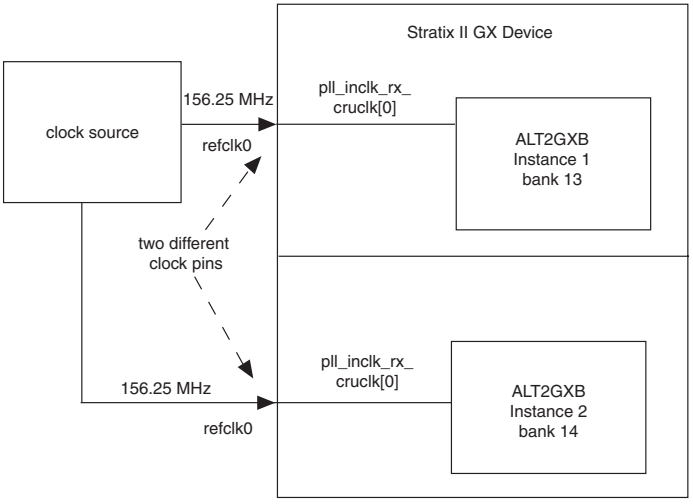
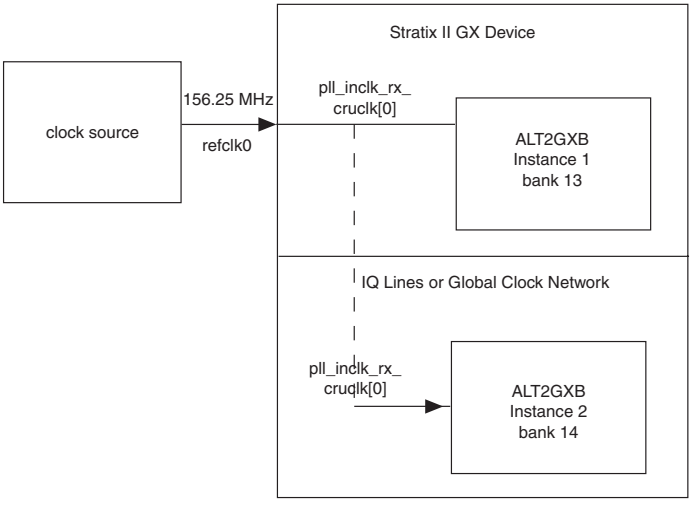


Figure 3–44. Correct Clocking Scheme for Reusing MIF



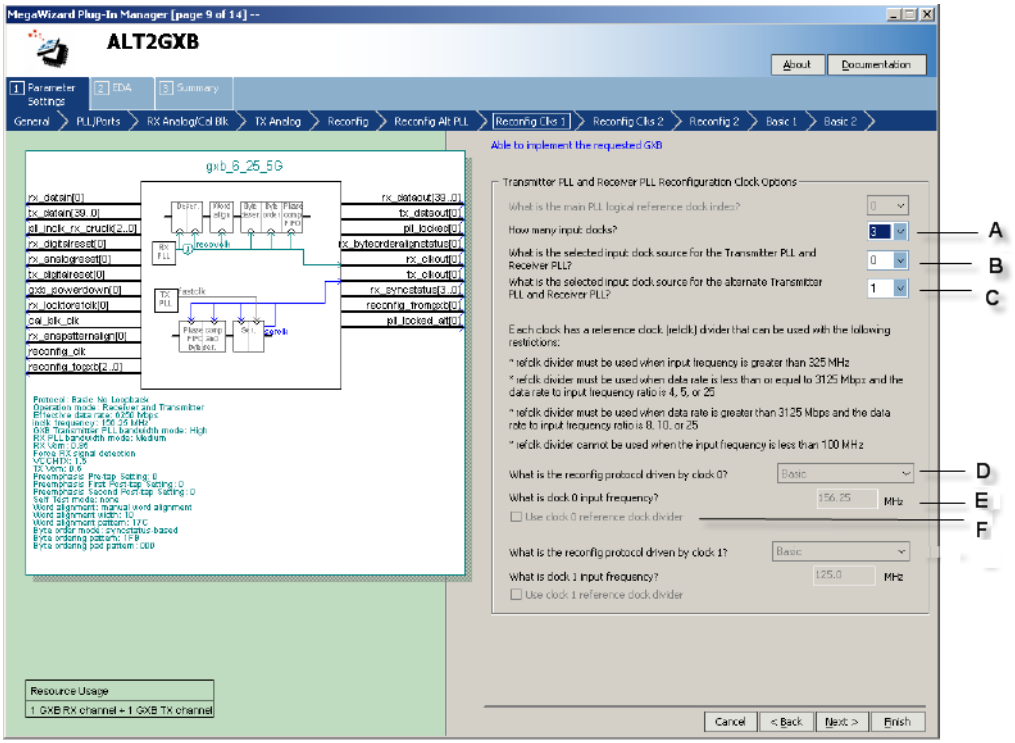
General Guidelines for Specifying the Input Reference Clocks

The following are general guidelines for your input reference clocks:

- Assign the identification numbers to all input reference clocks that are used in the design (0, 1, 2, 3, and 4). The identification numbers are indicated by "A" in Figure 3–45.
- Keep the identification numbering consistent for all the subsequent MIF configurations. Provide the identification numbers indicated by "B" and "C" in Figure 3–45.
- Maintain a consistent protocol, input reference clock frequencies, and reference clock pre-divider settings for all the MIF. Set these options indicated by "D", "E", and "F" in Figure 3–45.

These fields are explained in detail in "ALT2GXB MegaWizard Settings" on page 3–113.

Figure 3–45. Input Clock Settings of the Reconfig Clks 1 Tab



Logical TX PLL

In the channel reconfiguration feature ("Channel Reconfiguration" on page 3–30), you cannot reconfigure the TX PLLs dynamically. It was not necessary to know which of the two TX PLLs was configured for the specified data rate. You were required to set only the **logical reference index** option in the ALT2GXB MegaWizard to select the output clocks of the main and alternate TXPLLs and the **clock mux** in the RX PLL side (refer to Figure 3–40 for the location of these multiplexers).

When you use the **Channel and CMU PLL Reconfiguration** option, you can dynamically reconfigure the channel and the TX PLL. Therefore, to reconfigure the TX PLL during run time, you need the flexibility to select the TX PLL. When you enable this option, the ALT2GXB MegaWizard provides a logical identification for the main and alternate TXPLLs. This identification is referred to as the "logical tx pll" value. This value provides a logical identification to the TX PLL that is associated with a transceiver channel, without requiring the knowledge of its physical location.

In the ALT2GXB MegaWizard, when you provide the main TXPLL with a logical tx pll value, for example **0**, the alternate TXPLL automatically takes the complement value **1**. The logical tx pll value for the main TXPLL is stored along with the other transceiver channel information in the generated MIF. You can reuse the MIF generated for one TX PLL to reconfigure the other TX PLL in the same or in other transceiver blocks. The dynamic reconfig controller provides you an optional `logical_tx_pll_sel` port for this purpose. The method to use this port and reconfigure a TX PLL is explained in "Logical TX PLL" on page 3–98.

Using the Channel and CMU PLL Reconfiguration Feature

The channel and CMU PLL reconfiguration feature is divided into three categories based on the functionality:

- Channel and TX PLL reconfiguration
- TX PLL reconfiguration
- Channel reconfiguration with TX PLL select

You can select these features by setting the appropriate values in the reconfig_mode_sel port of the **ALT2GXB RECONFIG** tab. Table 3–9 shows the reconfig_mode_sel values for all the dynamic reconfiguration features.

Table 3–9. reconfig_mode_sel Values for All Dynamic Reconfiguration Modes

reconfig_mode_sel[2:0]	Description
000	PMA controls
001	Channel reconfiguration
010	Not supported (do not attempt to read or write with this value)
011	Dynamic Transmit rate switch
100	TX PLL
101	Channel and TX PLL reconfiguration
110	Channel Reconfiguration with TX PLL select
111	Not supported (do not attempt to read or write with this value)

The **read** operation is valid only for the reconfig_mode_sel value **000**. Do not use **read** for any other modes. Ensure that the reconfig_mode_sel port is set to the above mentioned supported values only. Setting the reconfig_mode_sel port to non-supported values may yield unpredictable transceiver behavior. When the **Enable Adaptive Equalization control** option is enabled, the reconfig_mode_sel port is 4-bits wide. In this case, set the most significant bit (MSB) of the reconfig_mode_sel to **0** when you use any of the above mentioned values.

As with the channel reconfiguration feature, the reconfig controller automatically increments the reconfig_address_out values to read the appropriate words from the MIF memory. Table 3–10 shows the address values incremented by the reconfig controller for the different features.

Table 3–10. Address Incremented by the Reconfig Controller (Part 1 of 2)

reconfig_mode_sel[2:0]	Incremented Address
001 (channel reconfiguration)	0-27
100 (TX PLL only)	0, 28-37

Table 3–10. Address Incremented by the Reconfig Controller (Part 2 of 2)

reconfig_mode_sel[2:0]	Incremented Address
101 (Channel and TX PLL reconfiguration)	0-37
110 (Channel reconfiguration with TX PLL select)	0-27

The procedure to write the MIF contents to the transceiver is the same as in with the channel reconfiguration feature (refer to Figure 3–8 on page 3–28).

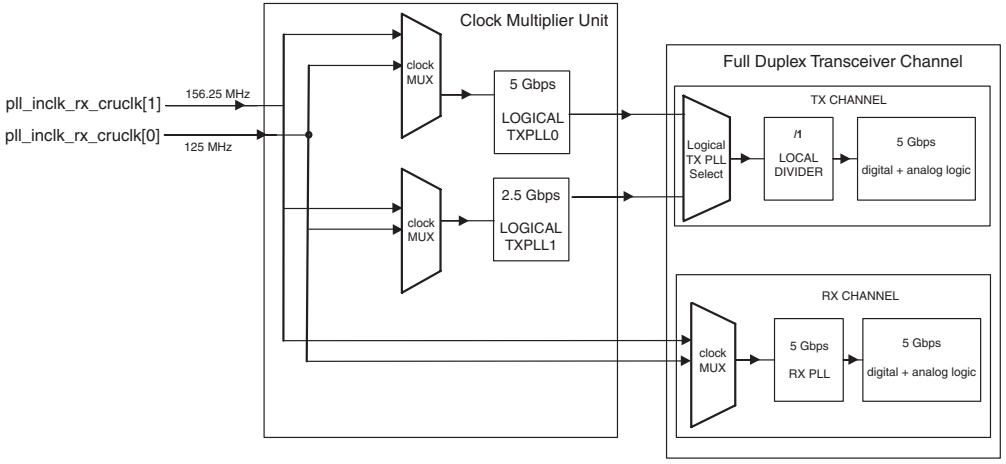
Channel and TX PLL Reconfiguration

The channel and TX PLL reconfiguration mode reconfigures the transceiver channel and the TX PLL that provide high-speed clocks to the transceiver channel. This mode helps to switch across multiple protocols that require different data rates and functional blocks. Since the TX PLL is also reconfigured when you use this feature, all the channels that are listening to the TX PLL are affected. To perform channel and TX PLL configuration, set the `reconfig_mode_sel` to **101** and write the MIF contents. During reconfiguration, the reconfig controller powers down the selected logical TX PLL until the new values are updated. The power down feature is explained in "TX PLL Powerdown" on page 3–109. To illustrate the functional blocks that are reconfigured, the following example is used. This same example is used for the three reconfiguration features.

Consider that you have an ALT2GXB instantiation with the following default configuration:

- The full-duplex channel with the main TXPLL configured to 5 Gbps data using a 156.25 MHz reference clock. The alternate TXPLL is configured to 2.5 Gbps using a 125 MHz reference clock (Figure 3–46 shows the default configuration).
- Assume that the logical tx pll value is set to **0** for the main TXPLL. (The settings to select the logical tx pll value for the TX PLLs are discussed in "ALT2GXB MegaWizard Settings" on page 3–113).

Figure 3–46. Transceiver Channel Default Configuration



Consider that you intend to switch to the following two modes:

mode1:

- Full-duplex channel with the main TXPLL configured to 6.25 Gbps data using a 156.25 MHz reference clock. Assume that the logical tx pll is set to **0** for the main TXPLL.
- Rate matcher is not enabled in the ALT2GXB megafunction.
- The alternate TXPLL is configured to 2.5 Gbps using a 125 MHz reference clock.

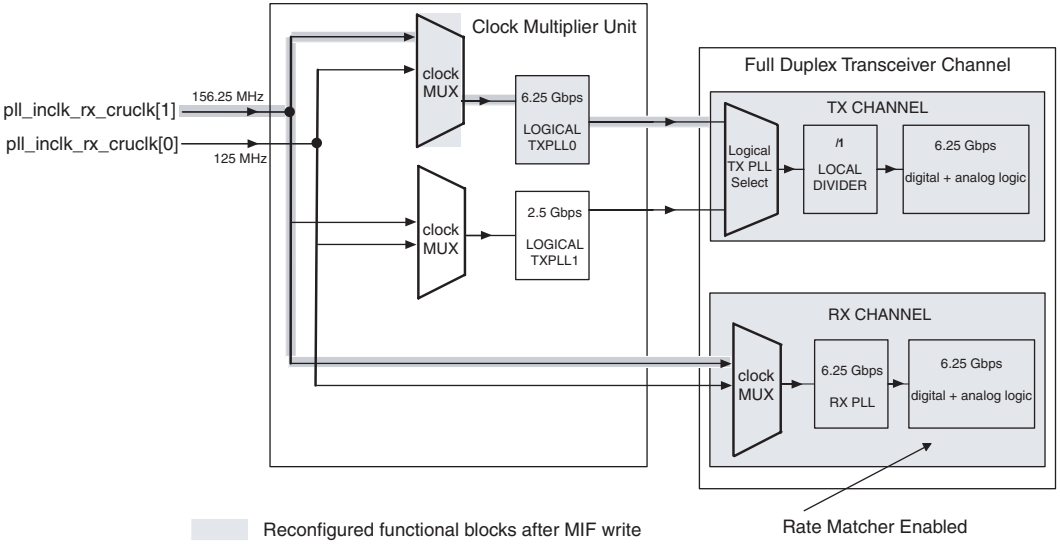
mode2:

- Full-duplex channel with the main TXPLL configured to 5 Gbps data using a 156.25 MHz reference clock. Assume that the logical tx pll is set to **0** for the main TXPLL.
- Rate matcher is enabled in the ALT2GXB megafunction.
- The alternate TXPLL is configured to 2.5 Gbps using a 125 MHz reference clock.

Consider that the MIF is generated for mode1 and mode2. (Details on the steps to generate the MIF are covered later in this section. The intent of this example is to show how the functional blocks are reconfigured based on the feature used).

Figure 3–47 shows the functional blocks that are reconfigured after the dynamic reconfig controller writes the mode2 MIF. Note that on the receive side, the rate matcher gets enabled after reconfiguration since the mode2 MIF contains settings to enable the rate matcher block.

Figure 3–47. Reconfigured Functional Blocks After the Channel and TX PLL Reconfiguration



TX PLL Reconfiguration

Using the TX PLL reconfiguration mode, you can reconfigure the TX PLLs. This mode is very useful in saving reconfiguration time in certain applications. Consider that you have four transmit-only instances in the same transceiver block that switch to different data rates together (assuming that the functional blocks are the same across the data rates). In this case, all these channels can listen to the same TX PLL. Instead of using the **Channel and TX PLL Reconfiguration** option (write 38 words) for individual channels, you can configure the TX PLL (write 10 words) once, to a different data rate. This, in turn, changes the transmit data rate of all the channels listening to this TX PLL.

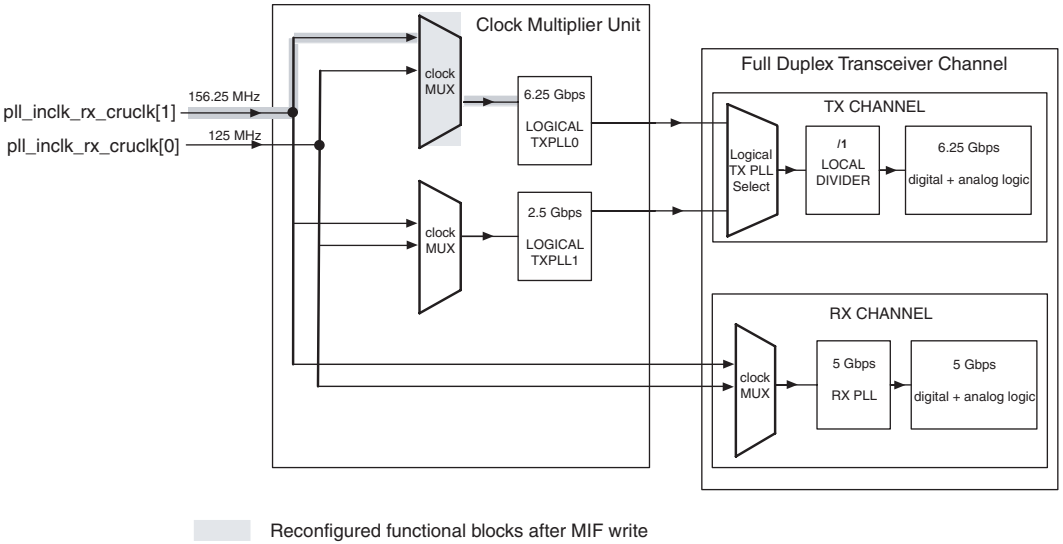
This mode is also useful when combined with the channel reconfiguration with TX PLL select mode. When the channel is listening to one TX PLL, you can reconfigure the other TX PLL, and later switch the transmit channel to listen to the configured TX PLL (explained in "Channel Reconfiguration with TX PLL Select" on page 3–103).

To perform TX PLL reconfiguration, set the `reconfig_mode_sel` value to **100** and write the MIF contents. The dynamic reconfig controller automatically increments the values on the `reconfig_address_out` port to word **0** and **28** through **37** from the specified MIF. The words 28-37 contain information to reconfigure the TX PLL. During reconfiguration, the reconfig controller powers down the TX PLL until the new values are written.

To understand the functional blocks that get reconfigured by this mode, consider the same example mentioned in "Channel and TX PLL Reconfiguration" on page 3–100. Figures 3–46 and 3–48 show the conditions before and after reconfiguration (using the mode1 MIF), respectively.

All the channels listening to the configured TX PLL are affected due to this reconfiguration.

Figure 3–48. Reconfigured Functional Blocks After TX PLL-Only Reconfiguration



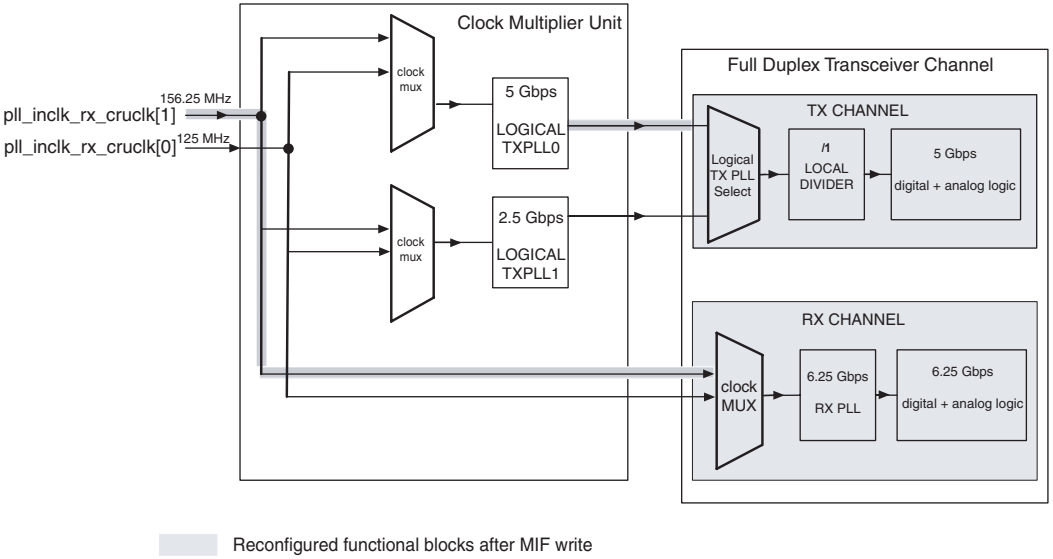
Channel Reconfiguration with TX PLL Select

This option reconfigures the channel and the logical TX PLL select multiplexer (shown in Figure 3–49) that selects the clock output from one of the TX PLLs. To use this feature, set the `reconfig_mode_sel` value to **110** and write the MIF contents. The **Channel Reconfiguration with TX PLL Select** option, in combination with the **TX PLL Reconfiguration**

option, is useful to switch the TX channel to multiple data rates. When the transmit side is using one TX PLL, reconfigure the second TX PLL using the TX PLL Reconfiguration feature. Then, use the channel reconfiguration with TX PLL select feature to switch the logical TX PLL select multiplexer to listen to the reconfigured TX PLL. This feature may require you to use the optional `logical_tx_pll_sel` port available in the **ALT2GXB RECONFIG** tab. The function of this port is explained in "Logical TX PLL" on page 3–98.

Consider the same example mentioned in "Channel and TX PLL Reconfiguration" on page 3–100 (refer to Figure 3–46 for conditions before reconfiguration). Figure 3–49 shows the reconfigured functional blocks after the mode1 MIF write is completed. Note that the TX PLLs are not reconfigured. Since the new MIF has the same functional blocks as the original configuration, there is no change in the functional blocks or the data rate in the transmit side after reconfiguration. Note that the MIF configures the RX PLL to 6.25 Gbps.

Figure 3–49. Reconfigured Functional Blocks After Channel and TX PLL Select Reconfiguration



Logical TX PLL Select

You can reuse the MIF created for one TX PLL on the other TX PLL using the optional `logical_tx_pll_sel` port in the ALT2GXB_RECONFIG MegaWizard. If the `logical_tx_pll_sel` port is enabled, the reconfig controller uses the value on this port irrespective of the logical tx pll value contained in the MIF. By using this port, you specify the identity of the TX PLL that you intend to reconfigure.

If you want to use the `logical_tx_pll_sel` only under some conditions and use the logical tx pll value stored in the MIF otherwise, enable an additional optional `logical_tx_pll_sel_en` port. If this port is enabled, the dynamic reconfig controller uses the value on the `logical_tx_pll_sel` port ONLY if the `logical_tx_pll_sel_en` port is set to **1** (refer to Figure 3–50). The values on these two ports should be held at a constant logic level until reconfiguration is completed. Table 3–11 shows the selected `logical_tx_pll` value under all the combinations of these two signals.

Figure 3–50. Effect of Using logical_tx_pll_sel and logical_tx_pll_sel_en Ports

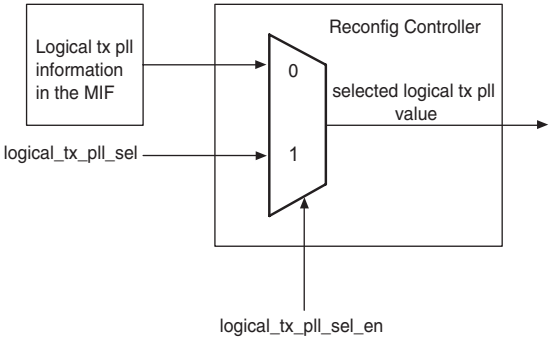


Table 3–11. logical_tx_pll_sel and logical_tx_pll_en Combinations (Part 1 of 2)

logical_tx_pll_sel Port	logical_tx_pll_se_en Port	Selected logical tx pll Value by the Reconfig Controller
enabled	enabled - value high	Value on the `logical_tx_pll_sel` port.
enabled	enabled - value zero	`logical tx pll` value stored in the MIF.

Table 3–11. logical_tx_pll_sel and logical_tx_pll_en Combinations (Part 2 of 2)

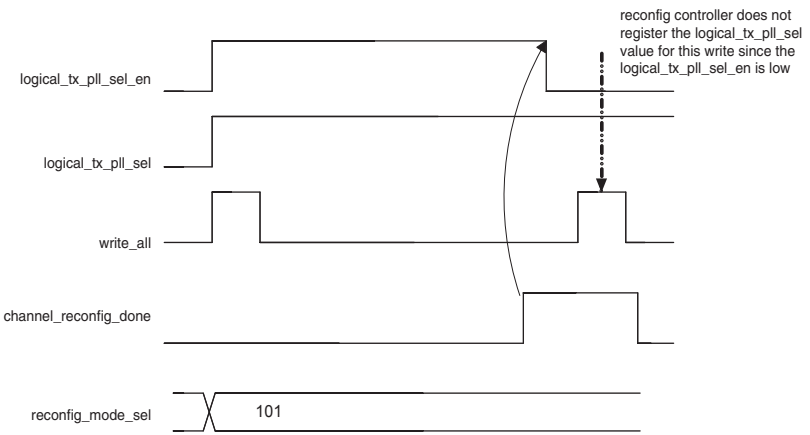
logical_tx_pll_sel Port	logical_tx_pll_se_en Port	Selected logical tx pll Value by the Reconfig Controller
enabled	not enabled	value on the `logical_tx_pll_sel` port.
not enabled	not enabled	logical `tx pll value` stored in the MIF.

When you configure a transceiver channel in the ALT2GXB MegaWizard, Altera recommends that you keep track of the TX PLL that drives the channel. You may require this information when you want to reconfigure the TX PLLs dynamically. This is illustrated in "Design Examples" on page 3–121.

The `logical_tx_pll_sel` port does not modify any transceiver setting on the RX side.

Figure 3–51 shows the required signal transitions to reconfigure the TX PLL with a `logical_tx_pll` value of **1**. Keep the `logical_tx_pll_sel` and `logical_tx_pll_sel_en` signals at a constant logic level until the reconfig controller asserts the `channel_reconfig_done` signal.

Figure 3–51. Signal Transitions of the logical_tx_pll_sel and logical_tx_pll_sel_en Ports

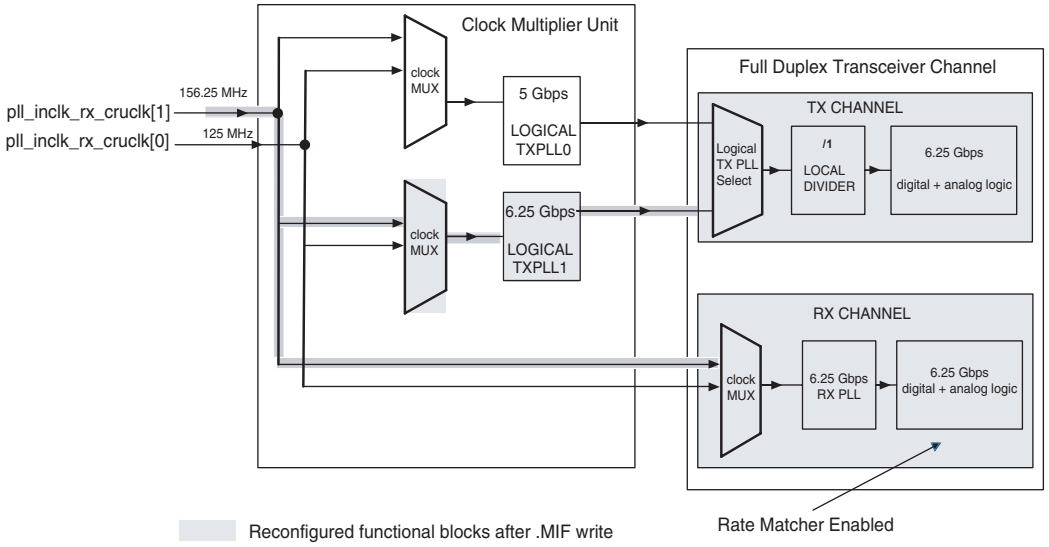


For illustration, the same example and MIF specified in the "Channel and TX PLL Reconfiguration" on page 3–100 is used here. The results of the reconfiguration under all the channel and CMU PLL reconfiguration modes are shown below. These results were achieved during run time, with the `logical_tx_pll_sel` input of the dynamic reconfig controller set to **1** (assuming the `logical_tx_pll_sel_en` is tied to **1**) and using the mode1 or mode2 MIF.

Channel and TX PLL Reconfiguration

Refer to Figure 3–46 for the channel configuration before the MIF write. Figure 3–52 shows the conditions after the channel is reconfigured using the mode2 MIF and setting the `logical_tx_pll_sel` to **1** during reconfiguration.

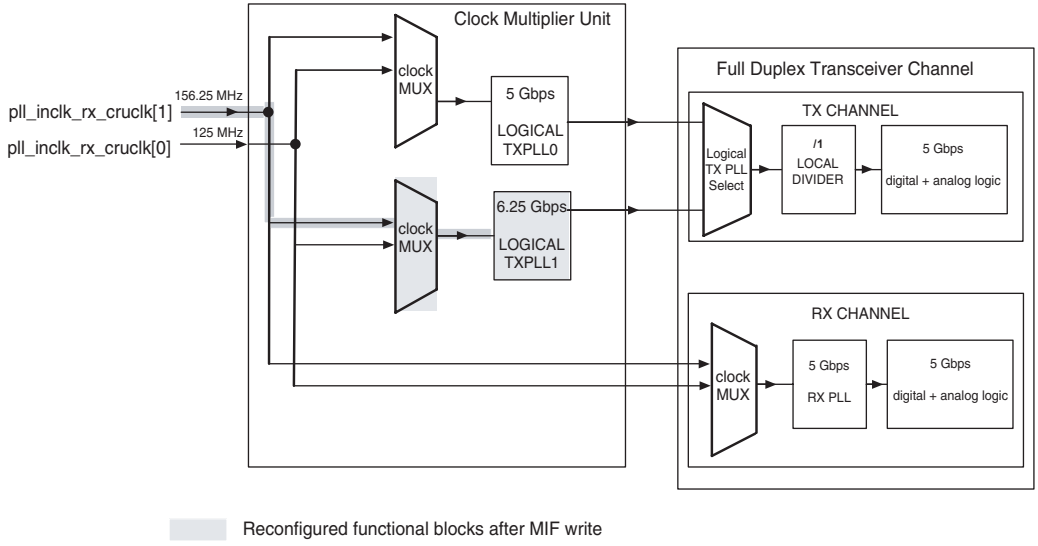
Figure 3–52. Reconfigured Functional Blocks Using logical_tx_pll_sel in Channel and TX PLL Mode



TX PLL Reconfiguration

Refer to Figure 3–46 for the channel configuration before the mode1 MIF is written. Figure 3–53 shows that the logical TXPLL1 is configured to 6.25 Gbps. The transmit channel still listens to the logical TXPLL0 and therefore runs at 5 Gbps (since in this mode, the logical tx pll select MUX is not reconfigured). The receive side is not configured with this feature.

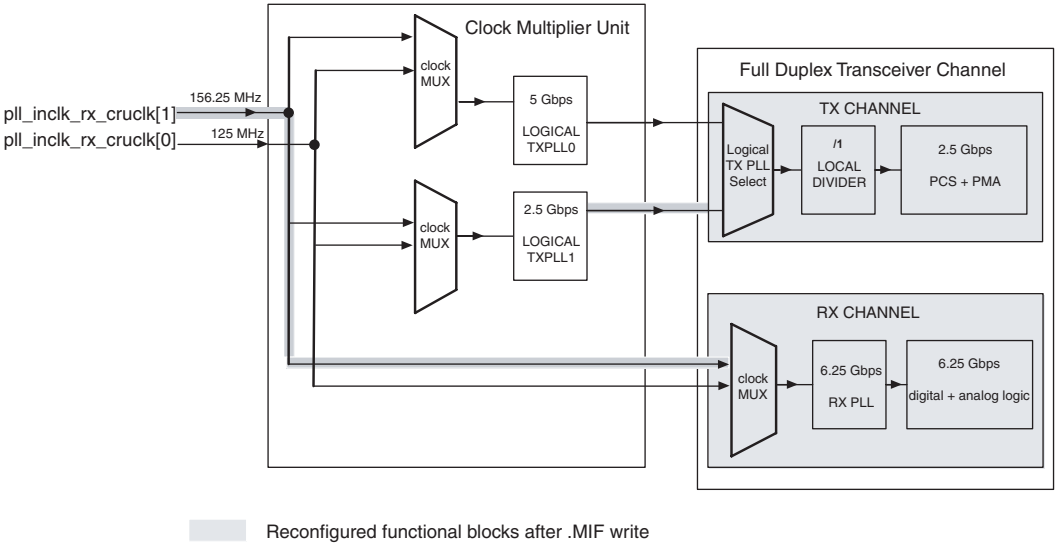
Figure 3–53. Reconfiguration Functional Blocks Using logical_tx_pll_sel in TX PLL Mode



Channel Reconfiguration with TX PLL Select

Refer to Figure 3–46 for the channel configuration before the mode1 MIF is written. Figure 3–54 shows the blocks that are reconfigured by the mode1 MIF and the `logical_tx_pll_sel` set to **1**. Note that in this case, the TX PLL is not configured. After the MIF is written, the logical TX PLL multiplexer gets configured to select the logical TXPLL1.

Figure 3–54. Reconfigured Functional Blocks using logical_tx_pll_sel in Channel Reconfiguration with TX PLL Select



You should keep the transceiver channel under reset during reconfiguration. Therefore, the channel may not be able to receive or transmit user data during reconfiguration.

You can use a MIF generated for one channel to all the other channels in the device if you meet the clocking requirements mentioned in "Clocking Enhancements and Requirements" on page 3–90.

TX PLL Powerdown

During channel and TX PLL reconfiguration or TX PLL Reconfiguration, the dynamic reconfig controller automatically powers down the selected TX PLL until it completes reconfiguring the selected TX PLL. The ALT2GXB_RECONFIG megafunction does not provide any external ports to control the TX PLL power down. If you reconfigure the main TXPLL, the `pll_locked` signal goes low. If you reconfigure the alternate TXPLL, the `pll_locked_alt` signal gets deasserted. Therefore, after reconfiguring the transceiver, wait for the `pll_locked` or `pll_locked_alt` signal from the ALT2GXB megafunction (depending on the TX PLL that is reconfigured) before continuing normal operation.

The dynamic reconfig controller powers down ONLY the selected TX PLL. The other TX PLL is not affected.

The main TXPLL corresponds to the TX PLL configuration set in the **General** tab of the ALT2GXB MegaWizard and the alternate TXPLL corresponds to the **Reconfig Alt PLL** tab.

Channel and CMU PLL Reconfiguration Duration

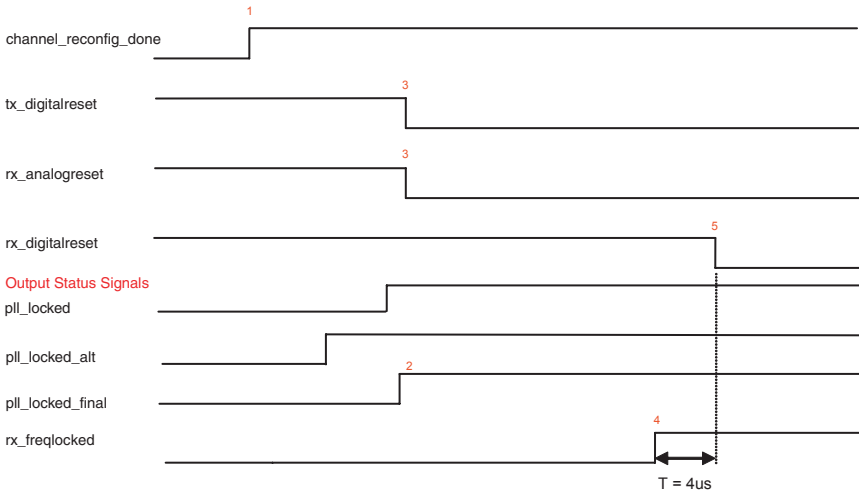
The dynamic reconfig controller takes the following number of reconfig clock cycles to write the contents in the MIF:

- Words 0 to 27 - approximately 260 clock cycles per word
- Words 28 to 37 - approximately 521 clock cycles per word

Reset Recommendations

Altera recommends that you follow a proper reset sequence during and after CMU PLL reconfiguration. Figure 3–55 shows the recommended reset sequence.

Figure 3–55. Reset Sequence During and After CMU PLL Reconfiguration (1 > 2 > 3 > 4 > 5)



As shown in Figure 3–55, assert the `tx_digitalreset`, `rx_digitalreset` and `rx_analogreset` when you initiate the CMU PLL reconfiguration MIF writes. After the dynamic reconfiguration control completes the CMU PLL reconfiguration, it asserts the `channel_reconfig_done` signal. After the `channel_reconfig_done` signal goes high, wait for `pll_locked` and `pll_locked_alt` (if you are using the alternate PLL) to go high (as

represented by pll_locked_final) and then de-assert the tx_digitalreset and rx_analogreset signals. Wait for a minimum of 4 µs after the rx_freqlocked signal goes high, then de-assert the rx_digitalreset signal.

Quartus II Settings and Requirements

The Quartus II software version 7.1 provides new assignments and settings to support the above mentioned channel and CMU PLL reconfiguration features.

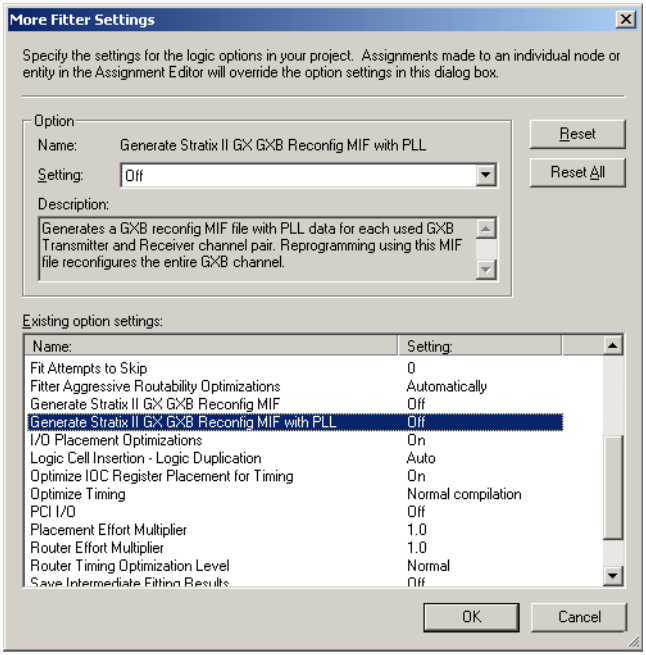
MIF Generation for Channel and CMU PLL Reconfiguration

To enable the Quartus II software version 7.1 to generate a MIF with 38 words, complete the following steps:

1. Go to the Assignments menu and select **Settings**, then **Fitter settings**.

2. Click the **more settings** button and set the **Generate Stratix II GX GXB Reconfig MIF with PLL** option to **ON** using the **Settings** option (as shown in Figure 3–56).

Figure 3–56. Quartus II Setting for MIF Generation



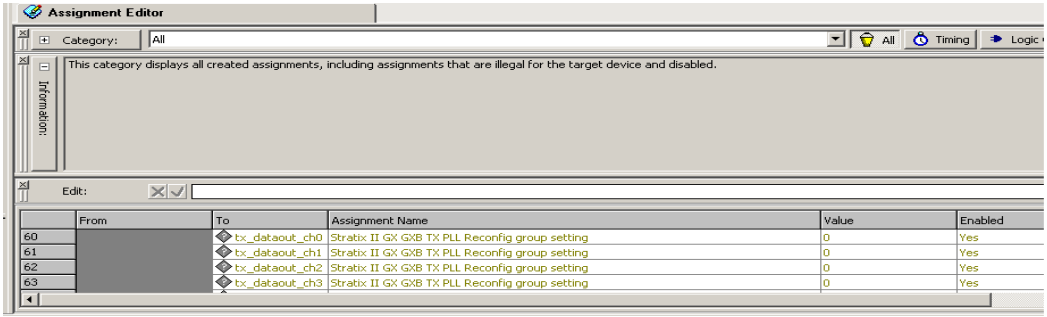
Grouping Transceiver Channels

The Quartus II software version 7.1 requires the following assignment editor setting for all channels assigned to the same transceiver bank, when you enable the **Channel and CMU PLL reconfiguration** option.

- Assignment setting: Assignment Name - **Stratix II GX GXB TX PLL Reconfig group setting** (as shown in Figure 3–57).

If you have more than one channel with the **Channel and CMU PLL Reconfiguration** feature enabled, and if you assign them to different reconfig groups without pin assignments for the `tx_dataout` pins, the Quartus II software automatically assigns these channels to different transceiver blocks. If you use a Stratix II GX device with one transceiver block, you cannot compile the design if you assign different TX PLL reconfig group values for the channels in your design.

Figure 3–57. Reconfig Group Setting Required for Channel and CMU PLL Reconfiguration



To understand the usage of this assignment setting, assume that you have two transmit channels in the same transceiver bank with the **Channel and CMU PLL Reconfiguration** option enabled. If the transmit output pins are `tx_dataout_ch0` and `tx_dataout_ch1`, set the following assignment setting to compile the design:

To : **tx_dataout_ch0**
Assignment Name: **Stratix II GX GXB TX PLL Reconfig group setting**
Value : **0**

To : **tx_dataout_ch1**
Assignment Name: **Stratix II GX GXB TX PLL Reconfig group setting**
Value : **0**

ALT2GXB MegaWizard Settings

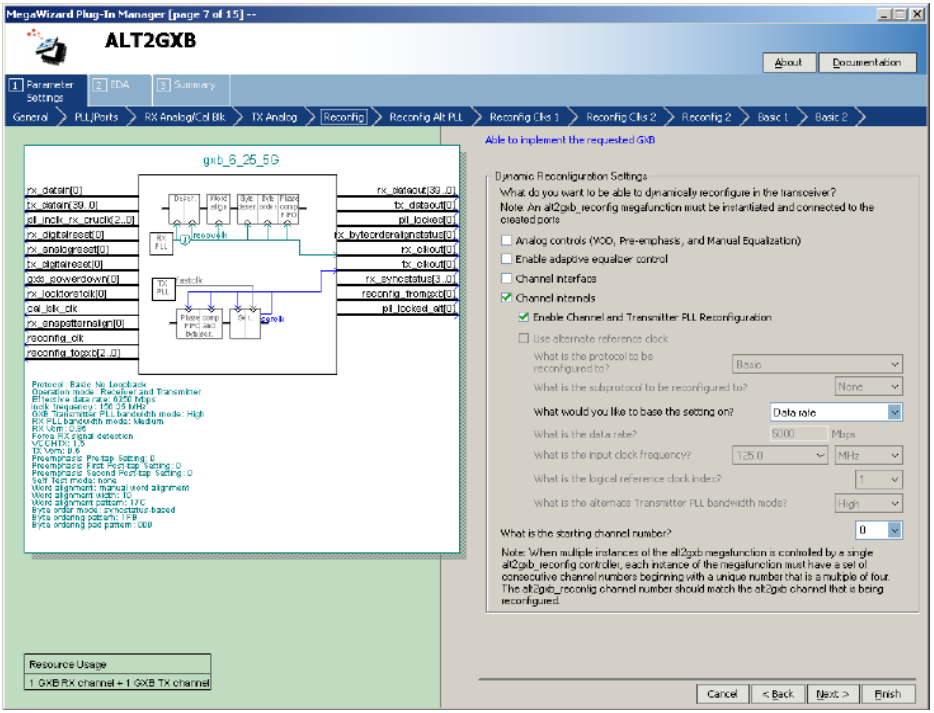
This section discusses the enhancements in the ALT2GXB MegaWizard to support the channel and CMU PLL reconfiguration feature.

Reconfig Tab Settings

When you enable the **channel internals** field, you can select the **Enable channel and Transmitter PLL** option (shown in Figure 3–58). This option allows you to use the channel and CMU PLL reconfiguration feature. When you select this option, the ALT2GXB MegaWizard enables new tabs—**Reconfig_Alt_PLL**, **Reconfig Clks 1,** and **Reconfig Clks 2**—to differentiate the settings for this feature from those of the channel reconfiguration feature (introduced in the Quartus II software version 6.1). This helps to maintain backward compatibility with the channel reconfiguration feature.

When you select the **Enable Channel and Transmitter PLL Reconfiguration** option, you cannot select the **Use alternate reference clock** option (used in channel reconfiguration feature). These two fields are mutually exclusive.

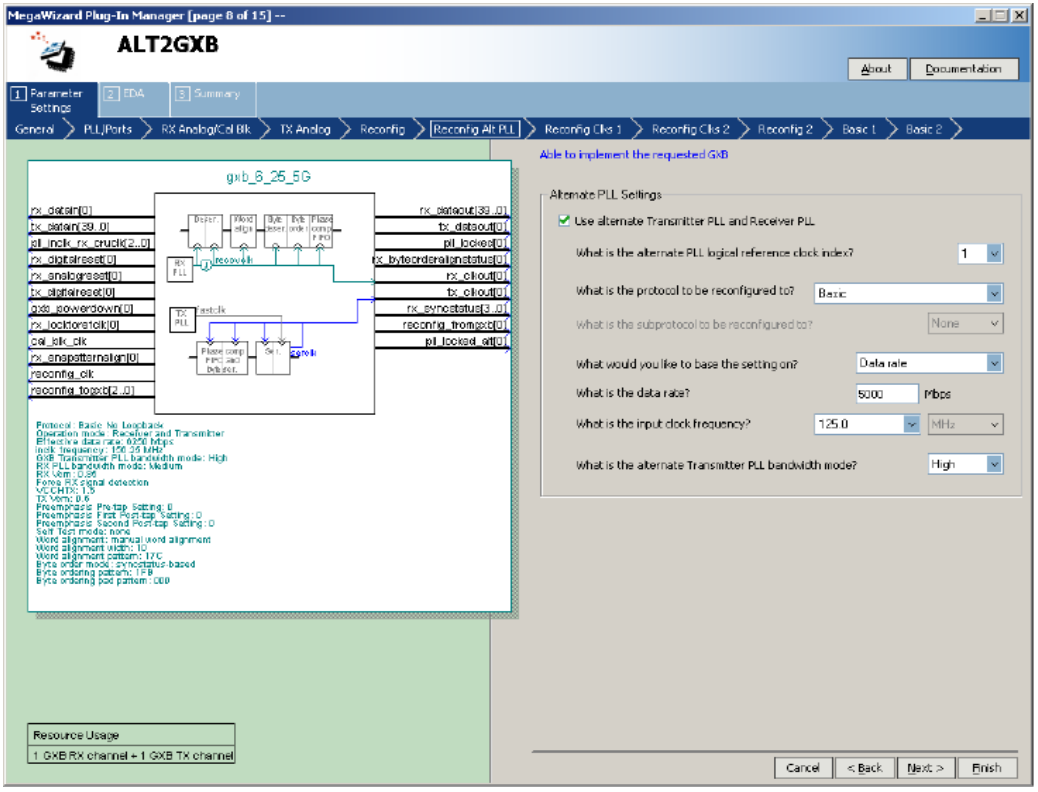
Figure 3–58. Reconfig Tab



Reconfig Alt PLL Tab Settings

When you select the **Use Alternate Transmitter and Receiver PLL** option, you can set the logical tx pll value to **0** or **1** for the alternate TXPLL from the **What is the alternate PLL logical reference index?** option (Figure 3–59).

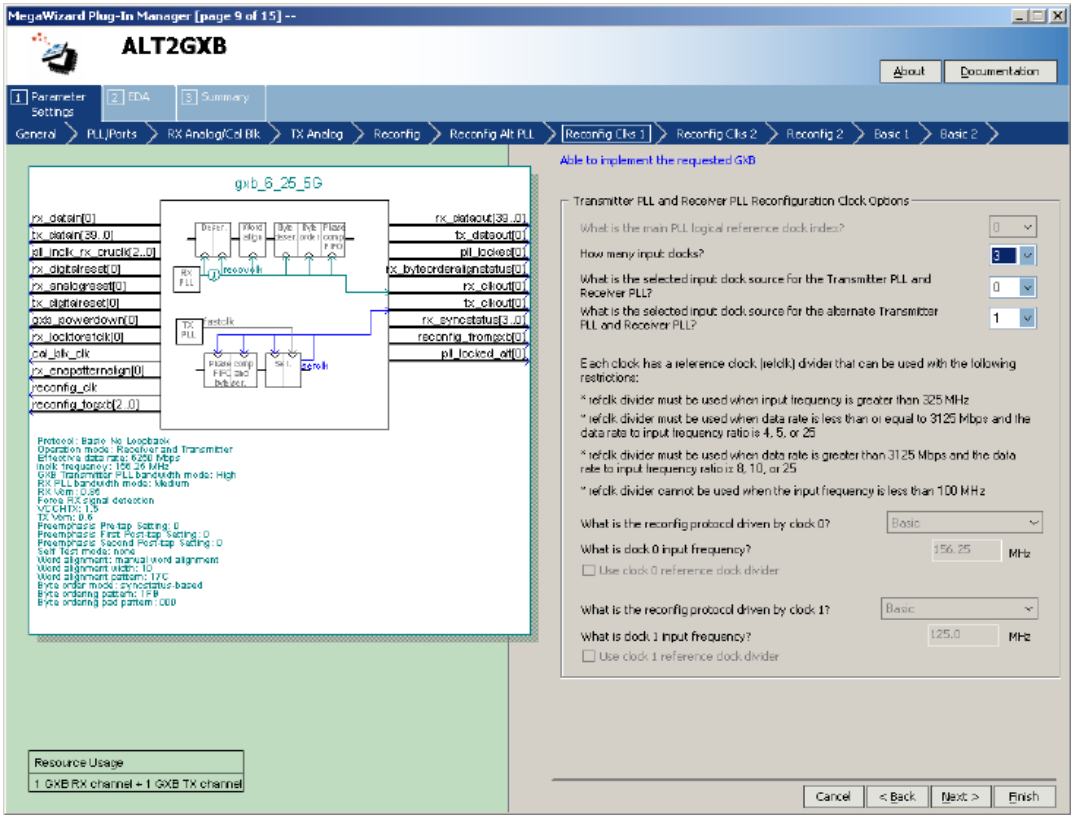
Figure 3–59. Reconfig Alt PLL Tab



Reconfig Clks 1 Tab

This tab provides options for the input reference clocks. The first option, **what is the main PLL logical reference clock index**, provides the logical tx pll value for the main TXPLL. If you have enabled the alternate TXPLL in the **Reconfig Alt PLL** tab, the ALT2GXB MegaWizard automatically selects the logical tx pll value of the main TXPLL as the complement of the alternate TXPLL. Otherwise, you can select the logical tx pll value for the main TXPLL in this tab (at the **What is the main PLL logical reference clock index?** option in Figure 3–60).

Figure 3–60. Reconfig Clks 1 Tab



The **How many input clocks?** option in Figure 3–60 shows the number of input reference clocks. When you set this field to **5**, the ALT2GXB Megawizard provides a **Reconfig Clks 2** tab to specify information about additional clock inputs. The **What is the selected input clock source for the Transmitter PLL and Receiver PLL?** and **What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?** options are used to select the input clocks for the main and alternate TXPLLs as well as the RX PLLs. The **What is the reconfig protocol driven by clock 0?** and **What is clock 0 input frequency?** options provide the protocol and clock frequency options for other clock sources that you anticipate you will use in your design.

For additional information on the input clock requirements, refer to "Clocking Enhancements and Requirements" on page 3–90.

Based on your settings in these fields, the ALT2GXB MegaWizard determines whether the `refclk` pre-divider should be enabled. For example, if you select the **SONET/SDH OC-12** protocol in the **what is the reconfig protocol driven by clock0** option, the ALT2GXB megafunction automatically enables the `refclk` pre-divider and connects the output of the pre-divider to the input reference clock port of the TX PLLs and RX PLLs. Similarly, if you select the input clock frequency greater than 325 MHz, the `refclk` pre-divider is enabled.

When you select the **Use clock 0 reference clock divide**r option, the Quartus II software instantiates the `refclk` pre-divider for the clock input.

If the information provided in the **General** and **Reconfig Alt Pll** tabs meet one of the conditions specified in "Using Dedicated refclks" on page 3–92, the Quartus II software automatically instantiates the `refclk` pre-divider for the corresponding clock input. For other clock inputs, you should determine whether the clock input frequency and the data rate meets one of the conditions specified in "Using Dedicated refclks" on page 3–92.

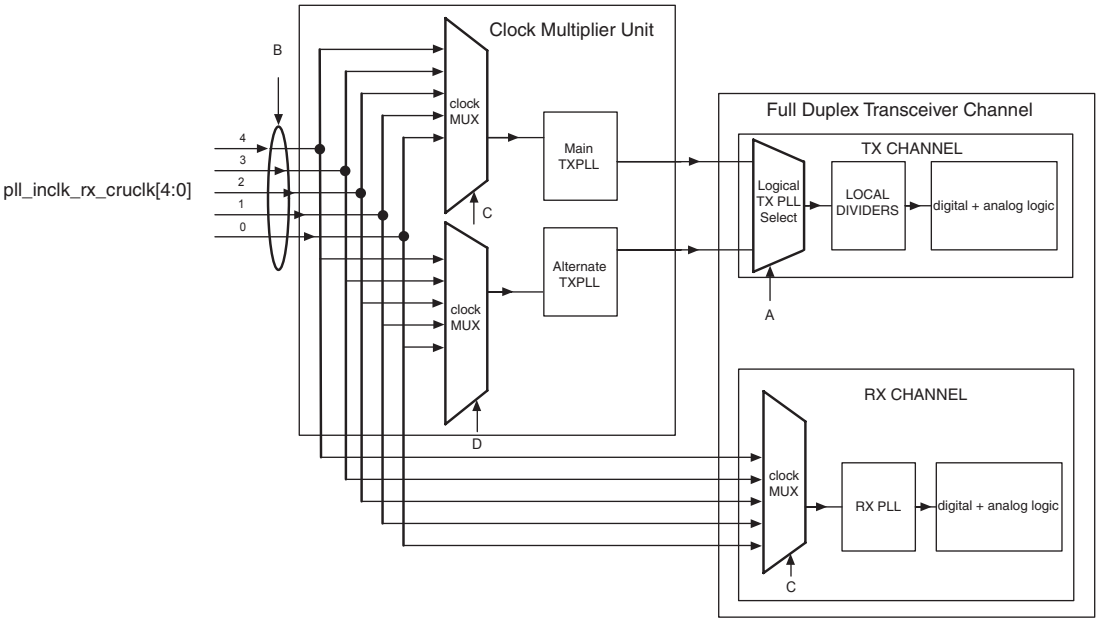
Example of a Condition to Select this Option:
Assume that you are using a clock input with a 125 MHz to configure the TX PLL to run the channel at the 3.125 Gbps data rate. In this case, the ratio of TX PLL data rate to input clock frequency is 25. This meets condition 2 specified in "Using Dedicated refclks" on page 3–92. Therefore, select this option so that the Quartus II software instantiates the `refclk` pre-divider for this clock source.

When the Quartus II software creates a pre-divider for a dedicated input reference clock (`refclk`), only the output of the pre-divider is available to clock the TX PLL and /or RX PLL.

If you would like to reuse the MIF across transceiver channels, you must have the same order of clock inputs across all the ALT2GXB instantiations that are using this MIF.

Figure 3–61 shows the mapping of the MegaWizard settings marked by "A", "B", "C", and "D" with the actual settings in the hardware (refer to Figure 3–60 for the **Reconfig Clks 1** tab).

Figure 3–61. Mapping Between the MegaWizard Settings and Hardware Settings



The other settings in the ALT2GXB MegaWizard are not specific to the channel and CMU PLL reconfiguration feature. Therefore, the other tabs are not discussed in this section.

Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about other ALT2GXB MegaWizard settings.

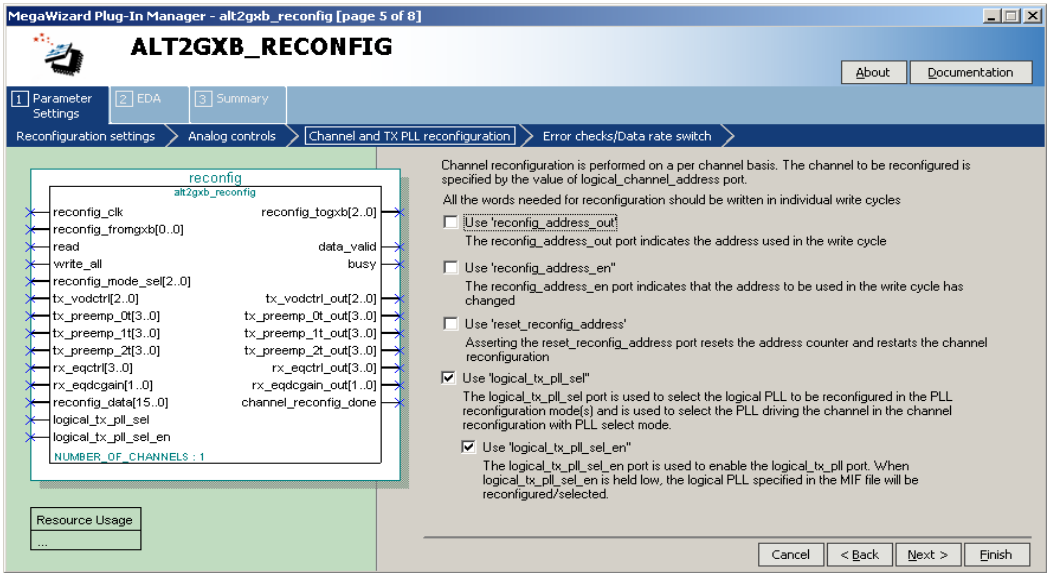
ALT2GXB RECONFIG Tab Settings

The Quartus II software version 7.1 has the following enhancements in the **ports/values** option in the ALT2GXB_RECONFIG MegaWizard.

- `reconfig_mode_sel`: The ALT2GXB_RECONFIG MegaWizard has new `reconfig_mode_sel` values to support the **channel and CMU PLL reconfiguration** option. For a complete list of `reconfig_mode_sel` values, refer to Table 3–9.

- `logical_tx_pll_sel`: The `logical_tx_pll_sel` port is optional. You can select this port in the **Channel and TX PLL Reconfiguration** tab. The value set in this port during reconfiguration overrides the logical tx pll value stored in the MIF. Refer to "Logical TX PLL Select" on page 3–105 for more information.
- `logical_tx_pll_sel_en`: The `logical_tx_pll_sel_en` port is optional. You can select this port in the **Channel and TX PLL Reconfiguration** tab. If this port is selected, the ALT2GXB_RECONFIG block registers the value on the `logical_tx_pll_sel` only if the `logical_tx_pll_sel_en` is asserted. Figure 3–62 shows the **Channel and TX PLL reconfiguration** tab in the ALT2GXB_RECONFIG MegaWizard in the Quartus II software version 7.1.

Figure 3–62. ALT2GXB RECONFIG Tab



The functionality of all other signals, such as `write_all`, `channel_reconfig_done`, `reconfig_address_en`, `logical_channel_address`, `data_valid`, and `busy`, have not changed since Quartus II software version 6.1. To write the MIF, follow the method used for the channel reconfiguration feature. Refer to Figure 3–8 on page 3–28 for more information.

Merging Transceiver Channels with Dynamic Reconfiguration Enabled

The following are the Quartus II software version 7.1 requirements for merging multiple transceiver channels that have the **Channel and CMU PLL Reconfiguration** option selected in the same transceiver bank:

- Assign all the channels to the same reconfig group. Refer to "Quartus II Settings and Requirements" on page 3–111 for more information on the reconfig group setting.
- All the channels should have the same reconfig options. That is, if you select **PMA controls**, **channel interface**, or **channel internals** in one channel, all the other channels should have the same selection. Some of the other scenarios for merging channels in the same transceiver bank are discussed below.

Case 1: Merging Transceiver Channels Listening to Two TX PLLs

Consider that you create an ALT2GXB instantiation for a full-duplex or TX-only configuration that has a main and alternate TXPLL. If you want to place other channels in the same transceiver bank, the other channels should also have a main and alternate TXPLL option to merge successfully. For example, consider that you create the following instantiation:

- **Instantiation1**—one full-duplex channel with the main TXPLL (assume a `logical_tx_pll` value of **0**), configured to 6.25 Gbps data rate and the alternate TXPLL configured to 2.500 Gbps.

Assume that you create another instantiation with the following configuration:

- **Instantiation2**—one full-duplex channel with only one TX PLL (assume a `logical_tx_pll` value of **0**), configured to 6.25 Gbps.

In this case you cannot merge instantiation1 and instantiation2 in the same transceiver bank since instantiation2 listens to only one TX PLL. To successfully merge the two instances, create instantiation2 with an alternate TXPLL configured to 2.500 Gbps.

Case II: Merging Transceiver Channels Listening to One TX PLL

Consider that you create an ALT2XB instantiation (full-duplex or TX-only configuration) that has only one TX PLL. If you would like to create another ALT2GXB instantiation configured at a different data rate in the

same transceiver bank, provide different logical tx pll values for the two instantiations. For example, to merge the following instantiations in the same transceiver bank:

- **Instantiation1**—full-duplex channel configured at 3.125 Gbps.
- **Instantiation 2**—full-duplex channel configured at 2.500 Gbps.

If you set the **what is the main PLL logical reference clock index** (in the **Reconfig Clks 1** tab) for instantiation1 to **0**, set this option to **1** for instantiation2. Since the Quartus II software requires separate TX PLLs for these two channels, the two instantiations should have different logical tx pll values.

Case III: Merging Separate Transmit-Only and Receive-Only Instantiation

In a full-duplex configuration with the **Channel and CMU PLL Reconfiguration** option enabled, the software automatically connects the same reference clock input to the TX PLL and RX PLL (explained in "Clocking Enhancements and Requirements" on page 3–90). If you merge a **transmit only** and a **receive only** configuration, the Quartus II software allows you to provide separate clock inputs for the TX PLL and RX PLL (you can connect the `pll_inclk_rx_cruclk[]` port of the two instances to two different clock source).

When you merge the **transmit only** and **receive only** configurations, you should add the **Stratix II GX Reconfig group setting** in the assignment editor for the `tx_dataout` and `rx_datain` pins and assign the same value to these two pins (**0** or **1**). This setting enables the Quartus II software to create a single (combined) MIF for the TX only and RX only instance.

Using this merging method, you can provide separate clock inputs to the TX PLL and RX PLL.

If you set the starting channel numbers in the ALT2GXB MegaWizard for the TX instance to **0** and RX instance to **4**, you can use `logical_channel_address` in the reconfig controller set to **0** or **4** to perform **Channel and CMU PLL Reconfiguration** on this transceiver channel.

Design Examples

This section covers the steps used in creating a design with the **Channel and CMU PLL reconfiguration** feature enabled.

Case I: Configuring Transceiver Channels to Switch Together Between GIGE, SONE-OC48, and Fibre Channel (FC)-4G Protocols

The GIGE, SONET/SDH OC48, and FC-4G have different input reference clocks, data path, and clocking requirements. For this example, assume the following Stratix II GX device configuration:

- Three transceiver banks
- Six full-duplex channels with two channels in each transceiver bank for a total of six channels (CH0, CH1, CH2, CH3, CH4, CH5).
- Each channel can independently switch between GIGE, SONET/SDH-OC48, and FC- 4G protocols.
- Assume that all channels are configured to FC-4G protocol at system power up.
- FC-4G uses Basic mode.

FC-4G and FC-2G refer to the fibre channel protocol at 4.25 Gbps and 2.125 Gbps data rate, respectively.

Table 3–12 shows the different parameters and ALT2GXB functional blocks for these three protocols.

Table 3–12. Differences in Functional Blocks Between GIGE, Fibre Channel, and SONET/SDH-OC48 *Note (1)*

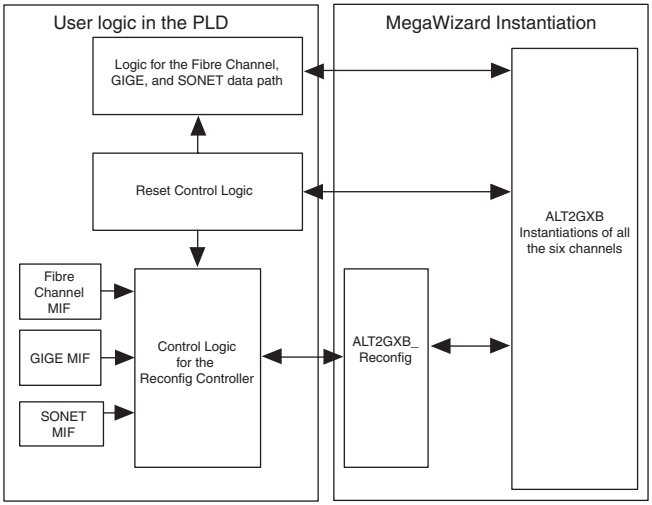
Parameters	Fibre Channel (Basic Mode) 4.25 Gbps	GIGE 1.25 Gbps	SONET/SDH OC48 2.488 Gbps
Selected input reference clock	106.25 MHz	125 MHz	77.76 MHz
PLD width	40	8	16
Byte Serializer/ Byte deserializer	yes	no	yes
8B/10B	no	yes	no
Rate Matcher	no	yes	no
Byte order block	yes	no	yes
Clock used for the receive side parallel interface	`rx_clkout`	`tx_clkout` (since rate matcher is used)	`rx_clkout`

Note to Table 3–12:

(1) The ALT2GXB MegaWizard allows more options for the input reference clock. For this example, we have selected the values shown in the table.

The differences between the three protocols determine the ALT2GXB MegaWizard settings. Figure 3–63 shows the top-level block diagram of the example design.

Figure 3–63. Top-Level Block Diagram of the Example Design

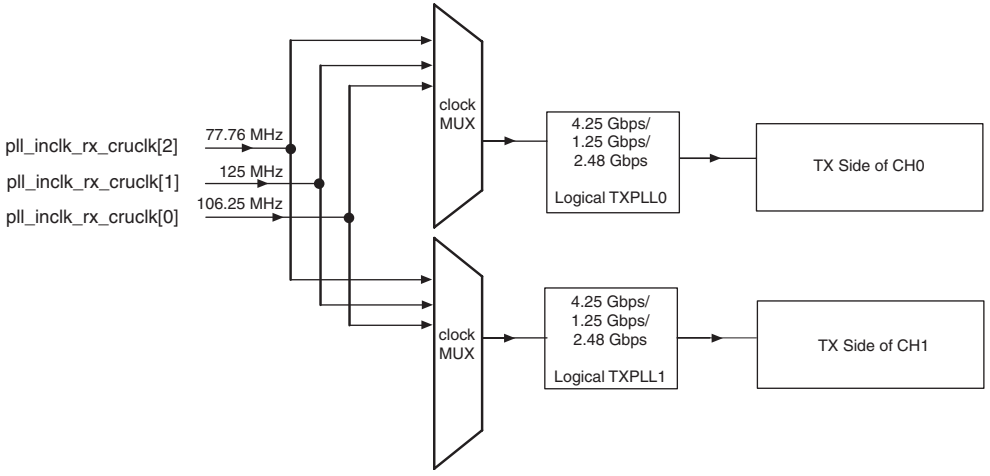


How Many MIFs do I Require?

For this example design, you can use one TX PLL per channel since you require only two full-duplex channels in the transceiver bank. To switch between three protocols, you need three MIFs. If you have consistent clocking across the three transceiver banks, you can reuse the same MIF across all the channels in the device. Figure 3–64 shows the clocking and TX PLL connections only for CH0 and CH1 (CH2-CH3, CH4-CH5 have the same configuration). To simplify the illustration, only the transmit channels of CH0 and CH1 and the TX PLL connections are shown.

If you create three MIFs (for FC-4G, GIGE, and SONET/SDH OC48) for one TX PLL, you can reuse the MIF in the other TX PLL using the `logical_tx_pll_sel` port in the dynamic reconfig controller. This means that you do not need a separate MIF for CH1 (since the other TX PLL is connected to CH1). Similarly, the same method can be applied for all the other channels in this example design. In total, you only need three MIFs for this example design.

Figure 3–64. TX PLL Connections to CH0 and CH1 for the Example Design



The 4.25/1.25/2.48 Gbps shown in Figure 3–64 indicates the possible switched data rates for the TX PLL to implement this example design.

The following discussion of the design is divided into five sections:

- Section I—ALT2GXB MegaWizard Settings for the Three Protocols
- Section II—ALT2GXB_RECONFIG MegaWizard Instantiation
- Section III—Steps to Create the MIF
- Section IV—Reset Control Logic and User Logic
- Section V—Top-Level Design and SRAM Object File (**.sof**) Generation

Section I— ALT2GXB MegaWizard Settings for the Three Protocols

Tables 3–13, 3–14, and 3–15 list the MegaWizard settings for each of the three protocols.

Table 3–13. FC-4G Protocol Settings (Part 1 of 3)

Tab Page and Option	Setting
General Tab Settings	
which protocol you will be using	basic
which sub protocol you will be using	serial loopback
operation mode	receiver and transmitter

Table 3–13. FC-4G Protocol Settings (Part 2 of 3)

Tab Page and Option	Setting
what is deserializer block width	double
what is channel width	40 (8b/10b encoder/decoder in the ALT2GXB is not used)
what is the data rate	4250 Mbps
what is the input clock frequency	106.25 MHz
what is the data rate division factor	1
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
PLL/Ports Tab Settings	
select the `gxb_powerdown`, `rx_freqlocked`, `pll_locked` in the screen	
RX Analog/Cal BLK Tab Settings	
select the calibration block	
select the `cal_blk_powerdown` if required	
TX Analog Tab Setting	
select the appropriate settings based on your requirements	
Reconfig Tab Settings	
select channel interface	This is required since the three protocols require different PLD widths (refer to Table 3–12).
select channel internals and enable channel and transmitter PLL reconfiguration	
Reconfig Alt PLL Tab Setting	
In this example design, you are using only two channels in the transceiver block. Since there are two TX PLLs per transceiver block, use one TX PLL for each channel and reconfigure the same TX PLL to switch across protocols. Therefore, you do not need an alternate TXPLL for this instance.	
Reconfig Clks 1 Tab Settings	
what is the main PLL logical reference clock index	0

Table 3–13. FC-4G Protocol Settings (Part 3 of 3)

Tab Page and Option	Setting
how many input clocks	3 (77.76 MHz, 125 MHz, and 106.25 MHz). Assume: clock2 = 77.76 MHz clock1 = 125 MHz clock0 = 106.25 MHz
what is the select input clock source for transmitter and receiver PLL	0
what is the reconfig protocol driven by clock1	GIGE
what is clock1 input frequency	125 MHz
use clock 1 reference clock divider	do not check this option
what is the reconfig protocol driven by clock2	SONET/SDH
what is clock2 input frequency	77.76 MHz
use clock 2 reference clock divider	do not check this option
Reconfig2 Tab Settings	
how should the receivers be clocked	select **use respective core clocks** since you clock the receive parallel date with `tx_clkout` for the GIGE protocol and `rx_clkout` for the other two protocols. Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information about these options.
how should the transmitters be clocked	use the respective channel transmitter core clocks.
check the control box to use the corresponding control port	select the protocol-specific signals. For SONET/SDH, you need `rx_byteorderalignstatus`, `rx_ala2sizeout`, etc. Refer to the *Stratix II GX ALT2GXB Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information.
Basic1 and Basic2 Tab Setting	
select the word alignment and other ports based on your requirements and complete the MegaWizard	

.

Table 3–14. GIGE Protocol Settings (Part 1 of 2)

Tab Page and Option	Setting
General Tab Settings	
which protocol you will be using	GIGE
operation mode	receiver and transmitter
what is the input clock frequency	125 MHz
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
PLL/Ports, RX Analog, Cal Blk, TX Analog, Reconfig Tab Settings	
Set the same settings as the FC-4G ALT2GXB instance mentioned in Tables 3–13.	
Reconfig Alt PLL Tab Setting	
no selection required.	
Reconfig Clks 1 Tab Settings	
what is the main PLL logical reference clock index	0 Note: Use this setting because you intend to generate the MIF with a logical tx pll value of **0**. Refer to "How Many MIFs do I Require?" on page 3–123
how many input clocks	3 (77.76 MHz, 125 MHz, and 106.25 MHz).
what is the select input clock source for transmitter and receiver PLL	1
what is the reconfig protocol driven by clock0	BASIC
what is clock0 input frequency	106.25 MHz
use clock 0 reference clock divider	do not check this option
what is the reconfig protocol driven by clock2	SONET/SDH
what is clock2 input frequency	77.76 MHz Note: The order of the clock inputs is the same as of the FC-4G instantiation shown in Table 3–13
use clock 2 reference clock divider	do not check this option
Reconfig2 Tab Settings	
same as of the FC-4G instantiation shown in Table 3–13	

Table 3–14. GIGE Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
Basic1 and Basic2 Tab Setting	
select the word alignment and other ports based on your requirements and complete the MegaWizard	

.

Table 3–15. SONET/SDH OC48 Protocol Settings (Part 1 of 2)	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	SONET/SDH
which sub protocol	OC48
operation mode	receiver and transmitter
what is the input clock frequency	77.76 MHz
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
PLL/Ports, RX Analog, Cal Blk, TX Analog, Reconfig Tab Settings	
set the same settings as the FC-4G ALT2GXB instance mentioned in Tables 3–13	
Reconfig Alt PLL Tab Setting	
no selection required.	
Reconfig Clks 1 Tab Settings	
what is the main PLL logical reference clock index	0
how many input clocks	3 (77.76 MHz, 125 MHz, and 106.25 MHz)
what is the select input clock source for transmitter and receiver PLL	2
what is the reconfig protocol driven by clock0	BASIC
what is clock0 input frequency	106.25 MHz
use clock 0 reference clock divider	do not check this option
what is the reconfig protocol driven by clock1	GIGE
what is clock1 input frequency	125 MHz

Table 3–15. SONET/SDH OC48 Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
use clock 1 reference clock divider	do not check this option
Reconfig2 Tab Settings	
same as of the FC-4G instantiation shown in Table 3–13	
Basic1 and Basic2 Tab Setting	
select the word alignment and other ports based on your requirements and complete the MegaWizard	

The ALT2GXB MegaWizard instantiation for the three protocols is complete.

The following are the settings for channel 1:

- Assume that the default configuration of CH1 is FC-4G. You can copy the FC-4G instantiation created for CH0. The only change required for this new instantiation is in the **Reconfig Clks 1** tab.
- Set the **What is the main PLL logical reference clock index?** option to **1** (this is the logical tx pll value) and complete the MegaWizard. For CH0, set the value in this field to **0**. Since the design goal is to reconfigure these two channels independently, set different logical tx pll values for these two channels and complete the MegaWizard. For CH2 and CH4, reuse the CH0 instance. Similarly, for CH3 and CH5, reuse the CH1 instance.

Section II — ALT2GXB_RECONFIG MegaWizard Instantiation

The following are settings for the ALT2GXB_RECONFIG MegaWizard:

- Set the **What is the number of channels controlled by the reconfig controller?** option to **24**. In this design, you have six instantiations (for six channels). The starting channel numbers for each of these instantiations should be a multiple of four. Each of these ALT2GXB instantiation has a `reconfig_fromgxb` output port.
- The reconfig controller provides one `reconfig_fromgxb` input port for a multiple of 4 channels. Therefore, set the above field to **24** (rounded to the nearest transceiver block). For additional information on starting channel numbers and logical channel addressing, refer to "Introduction" on page 3–1.

- Select the **Channel and TX PLL select/reconfig** option.
- In the **Channel and TX PLL reconfiguration** tab, select `reconfig_address_out` and `reconfig_address_en`. Select `logical_tx_pll_sel` and `logical_tx_pll_sel_en` ports so that you can reuse the MIF. (Refer to "How Many MIFs do I Require?" on page 3–123.

Section III — Steps to Create MIFs

This section explains the steps to create all the MIFs at one time:

1. Go to the Assignments menu and select **Settings**, then **Fitter settings**.

2. Click the **more settings** button and set the **Generate Stratix II GX GXB Reconfig MIF with PLL** option to **ON** using the settings option (as shown in Figure 3–56).

3. Create a top-level design file and include the three instantiations created for CH0 (FC-4G, GIGE, and SONET/SDH OC48 protocol). Connect the `pll_inclk_rx_cruclk[]` ports to the following clock source:

   a. `pll_inclk_rx_cruclk[0]` - **106.25** MHz. Assume `refclk0` of transceiver bank 13.

   b. `pll_inclk_rx_cruclk[1]` - **125** MHz. Assume `refclk0` of transceiver bank 14.

   c. `pll_inclk_rx_cruclk[2]` - **77.76** MHz. Assume `refclk0` of transceiver bank 15.

4. For this example design, assume that the three clock inputs are provided from the dedicated `refclk` pins. If you provide input reference clocks through the global clock networks, refer to "Clocking Enhancements and Requirements" on page 3–90 for usage limitations.

5. Assign the `tx_dataout` and `rx_datain` pins of the FC-4G, GIGE, and SONET/SDH OC48 instantiations to transceiver banks 13, 14, and 15, respectively. Since you have assigned logical tx pll value to **0** to all these instantiations, place these channels in three different transceiver banks to compile successfully (refer to "Case II: Merging Transceiver Channels Listening to One TX PLL" on page 3–120). The intent of placing these instantiations in different banks is to generate all the MIFs at one time.

6. Rename the generated MIFs to indicate the protocol for which the MIF is configured.

Section IV — Reset Control Logic and User Logic

The reset control logic takes care of resetting the transceiver during system initialization and during reconfiguration (Altera recommends a specific reset sequence, refer to "Reset Recommendations" on page 3–66 for more information).

For the user logic, use different clocks (`tx_clkout` for GIGE protocol and `rx_clkout` for FC-4G and SONET/SDH OC48) for the parallel data in the receive interface of the ALT2GXB. The user logic is not discussed in this section. Refer to Figure 3–37 on page 3–82 for more information regarding user logic in a similar configuration.

Section V — Top-Level Design and SRAM Object File (.sof) Generation

Follow these steps to generate a SRAM object file:

1. Instantiate the six ALT2GXB channels in the top-level design. That is, stamp the FC-4G instance created for `logical_tx_pll` value **0** three times for CH0, CH2, and CH4.

2. Similarly, stamp the instance created for `logical_tx_pll` value **1** for CH1, CH3, and CH5.

3. Add the reset and user logic and connect the signals. In the assignment editor, use the **Stratix II GX GXB TX PLL Reconfig group setting** option and assign the `tx_dataout` of CH0 and CH1 to the same reconfig group (this is required to assign CH0 and CH1 to the same transceiver bank).

4. Similarly, assign the same reconfig groups for CH2-CH3 and CH4-CH5.

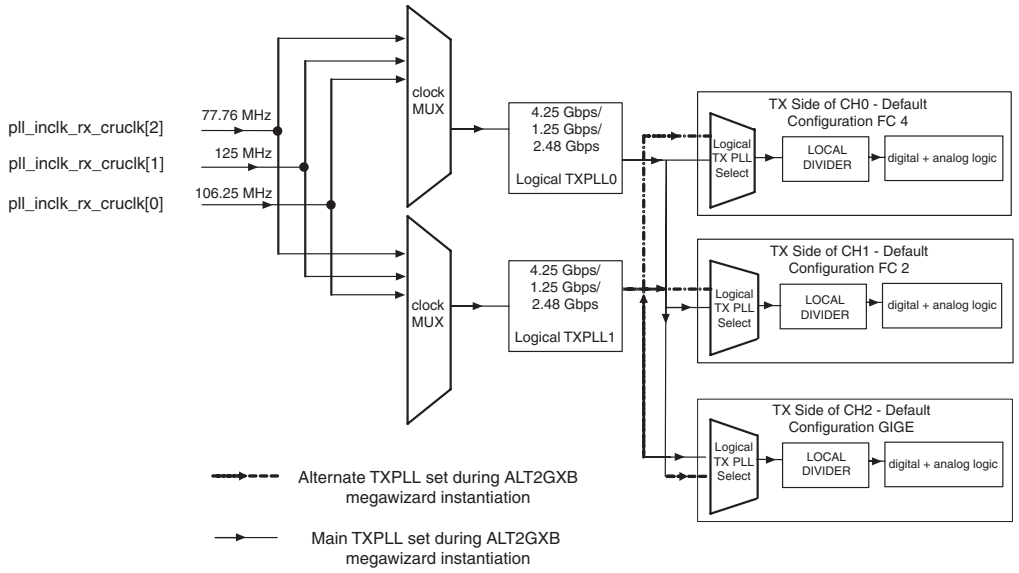
Case II: Configuring Transceiver Channels to Switch Independently Between Three Different Protocols

This example discusses the steps to reconfigure the three full-duplex channels (CH0, CH1, and CH2) in a transceiver bank between the FC-4G, FC-2G, GIGE, SONET/SDH OC48 protocols. In the previous example, the design used only two channels in a transceiver bank. Therefore, each channel could use a dedicated TX PLL.

In this example, the three channels are reconfigured in a transceiver bank. This means that two TX PLLs are shared between three channels. Therefore, if you use a main and alternate TXPLL for each channel, you can reconfigure the channel to any two of the four protocols by switching between the two TX PLLs.

Figure 3–65 shows the TX PLL connections of the TX side. (To simplify the illustration, only the TX side is shown). The figure shows that the default configurations of CH0, CH1, and CH2 are FC-4G, FC-2G, and GIGE, respectively.

Figure 3–65. Logical TX PLL Connections with the Transceiver Channel



How Many MIFs Do I Require?

You will need four MIFs for this design. You can generate the MIFs for one TX PLL and use the `logical_tx_pll_sel` in the reconfig controller to write the MIF contents into the second TX PLL. Assume that the TX PLL configured for FC- 4G data rate is assigned a logical tx pll value of **0**. This means that the other TX PLL configured for GIGE and SONET/SDH

OC48 protocol is assigned a logical tx pll value of **1**. Table 3–15 shows the number of MIFs required and the logical tx pll value assigned for the different configurations.

Table 3–16. MIF File for the Four Configurations

Number	MIF	logical tx pll Value
1	FC-4G	0
2	FC-2G	0
3	GIGE	1
4	SONET/SDH OC48	1

Use the same clocking connections and data path for these protocols that were provided in the previous example.

The following discussion of the design is divided into three sections:

- Section I—ALT2GXB MegaWizard Settings for the Three Protocols
- Section II—ALT2GXB_RECONFIG MegaWizard Instantiation
- Section III—Using the logical_tx_pll_sel During Reconfiguration

Section I — ALT2GXB MegaWizard Settings for the Three Protocols
In this section, only the ALT2GXB MegaWizard settings relevant to the channel and CMU PLL reconfiguration feature are discussed in Tables 3–17 through 3–20.

Refer to Tables 3–13, 3–14, and 3–15 for the tab settings that are not specified in this section.

Table 3–17. FC-4G Protocol Settings (Part 1 of 2)

Tab Page and Option	Setting
General Tab Settings	
which protocol you will be using	basic
which sub protocol you will be using	serial loopback
operation mode	receiver and transmitter
what is deserializer block width	double
what is channel width	40 (8b/10b encoder/decoder in the ALT2GXB is not used)
what is the data rate	4250 Mbps
what is the input clock frequency	106.25 MHz

Table 3–17. FC-4G Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
what is the data rate division factor	1
select the `rxdigitalreset`, `txdigitalreset`, and `rxanalogreset` ports	
Reconfig Tab Settings	
select channel interface	this is required since the three protocols require different PLD widths (refer to Table 3–12)
select channel internals and enable channel and transmitter PLL reconfiguration	
Reconfig Alt PLL Tab Setting	
what is the alternate PLL reference clock index	1 you used a logical tx pll value of **0** for FC-4G. Therefore, this alternate index (for example, GIGE) should be set to **1**
protocol	GIGE
data rate	1.25 Gbps
clock frequency	125 MHz
Reconfig Clks 1 and Reconfig2 Tab Settings	
use the same clock order as the previous example. Refer to Table 3–13	

For the FC-2G configuration, use the TX PLL that provides FC-4G clock frequency and divide by two with the local divider in the TX channel.

Copy the instance created for the FC-4G instance. The only change required is in the **General** tab.

Table 3–18. FC-2G Protocol Settings (Part 1 of 2)	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	basic
which sub protocol you will be using	serial loopback
operation mode	receiver and Transmitter
what is deserializer block width	double
what is channel width	(8b/10b in the ALT2GXB)

Table 3–18. FC-2G Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
what is the data rate	4250 Mbps
what is the input clock frequency	106.25 MHz
what is the data rate division factor	2

.

Table 3–19. GIGE Protocol Settings	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	GIGE
what is the input clock frequency	125 MHz
Reconfig Alt PLL Tab Setting	
what is the alternate PLL reference clock index	0 You used a logical tx pll value of **1** for GIGE. Therefore, the alternate index (for FC-4G) should be set to **0.**
protocol	BASIC
data rate	4.25 Gbps
clock frequency	106.25 MHz
Reconfig Clks 1 and Reconfig2 Tab Settings	
use the same clock order as the previous example. Refer to Table 3–13	

Copy the instance created for FC-4G instance. The only change required is in the **General** tab.

Table 3–20. FC-2G Protocol Settings (Part 1 of 2)	
Tab Page and Option	**Setting**
General Tab Settings	
which protocol you will be using	SONET/SDH
which sub protocol you will be using	OC48
what is the input clock frequency	77.76 MHz
Reconfig Alt PLL Tab Settings	

Table 3–20. FC-2G Protocol Settings (Part 2 of 2)	
Tab Page and Option	**Setting**
what is the alternate PLL reference clock index	0 You used a logical tx pll value of **1** for SONET/SDH OC48. Therefore, the alternate index (for FC-4G) should be set to **0**.
protocol	BASIC
data rate	4.25 Gbps
clock frequency	106.25 MHz

Section II — ALT2GXB_RECONFIG MegaWizard Instantiation
The following are the settings for the ALT2GXB_RECONFIG MegaWizard:

- Set the **What is the number of channels controlled by the reconfig controller** option to **12** (This will provide separate `reconfig_from_gxb` ports for each instance).
- Select the **Channel and TX PLL select/reconfig** option.
- In the **Channel and TX PLL Reconfiguration** tab, select the `reconfig_address_out`, `reconfig_address_en`, `logical_tx_pll_sel`, and `logical_tx_pll_sel_en` ports so that you can reuse the MIF. (Refer to "How Many MIFs Do I Require?" on page 3–132).

Section III — Using the logical_tx_pll_sel During Reconfiguration
Follow the same procedure as mentioned in "MIF Generation for Channel and CMU PLL Reconfiguration" on page 3–111 to create your MIFs. In the top-level design, assign the **Stratix II GX GXB TX PLL Reconfig group setting** and assign the same reconfig group to the three channels.

If you would like to reconfigure CH0 to SONET/SDH OC48 mode, use the following steps:

1. In the MegaWizard instantiation, set the logical tx pll value of the main TXPLL for CH0 to **0**. The SONET/SDH OC48 MIF contains the logical tx pll value of **1**. To use the SONET/SDH OC48 MIF for CH0, set both the `logical_tx_pll_sel` and `logical_tx_pll_sel_en` ports to **1** in the reconfig controller.

2. Set the `reconfig_mode_sel` value to **101** (channel and TX PLL reconfiguration) and write the SONET/SDH OC48 MIF.

Since CH0 and CH1 share the same TX PLL, configuring CH0 affects CH1. You can either configure CH1 to go to SONET/SDH OC48 mode or switch CH1 to listen to the GIGE mode by switching it to listen to the alternate TXPLL.

Adaptive Equalization (AEQ)

High-speed interface systems are used at different data rates with multiple backplane environments. These systems require different equalization settings to compensate for changing data rates and backplane characteristics. Manually selecting optimal equalization settings is cumbersome under these changing system characteristics. The adaptive equalization feature solves this problem by enabling the Stratix II GX device to continuously tune the receiver equalization settings based on the frequency content of the incoming signal. Five equalizer filters are tuned during this adaptive equalization process. The user logic can dynamically control the AEQ hardware through the dynamic reconfiguration controller.

This section explains the method to enable different options to control the AEQ hardware. Altera assumes that you have prior knowledge about the dynamic reconfiguration controller. For basic information, refer to "Dynamic Reconfiguration Controller Architecture" on page 3–2.

Conventions Used

The following conventions are used in this section:

- ALT2GXB_RECONFIG—Refers to the dynamic reconfiguration controller logic generated by the Quartus II AL2GXB_RECONFIG MegaWizard Plug-In Manager. ALT2GXB_RECONFIG and dynamic reconfiguration controller are used interchangeably in this section.
- Active channels—channels that have the **Enable adaptive equalizer control** option selected in the ALT2GXB MegaWizard. Selecting this option enables the adaptive equalization hardware.

AEQ Feature Requirements

The following are device requirements for the AEQ feature:

- Different device families require different silicon revisions, as shown in Table 3–21:

Table 3–21. Silicon Revision Requirements for the AEQ Feature

Device Family	Silicon Revision *(1)*
2SGX30	Revision A
2SGX60	Revision A
2SGX90	Revision C
2SGX130	Revision B

Note to Table 3–21:

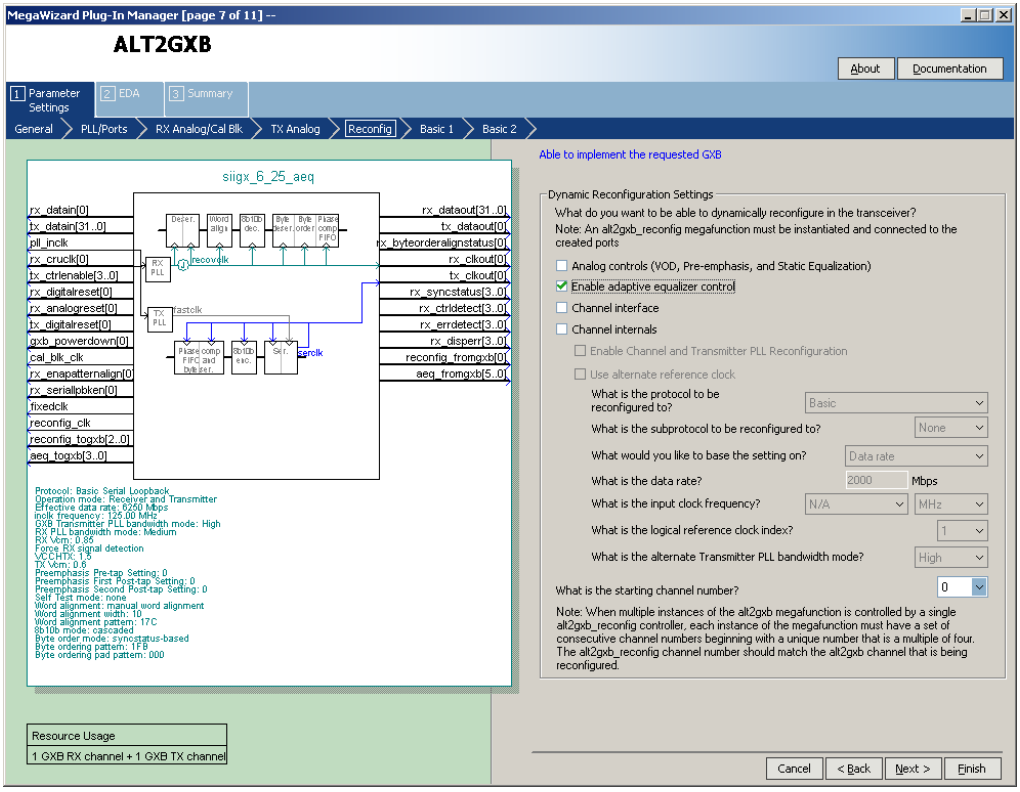
(1) You can identify the silicon revision by looking at the print below the device name in the device package. The third letter from the left indicates the silicon revision. For example, the print **AAC9X0607A** (third letter from left, **"C**") indicates a REV C silicon.

- The transceiver data rate needs to be > 2.5 Gbps
- The receive data needs to be 8B/10B encoded.
- Not available in PCI-Express (PIPE) functional mode (since the adaptive equalization hardware cannot perform the equalization process when the receive link is under the **electrical idle** condition)
- The receiver input signal should have a minimum envelope of 400 mv (differential peak-to-peak). The Quartus II software does not check for this requirement.
- AEQ is supported only in device speed grades C3, C4, or I4.

Enabling the AEQ Hardware

The AEQ hardware is available for each transceiver channel in the Stratix II GX device. To enable the AEQ hardware, select the **Enable adaptive equalizer control** option in the **Reconfig** page of the ALT2GXB MegaWizard plug-in Manager (Figure 3–66).

Figure 3–66. Enable the Adaptive Equalization Hardware



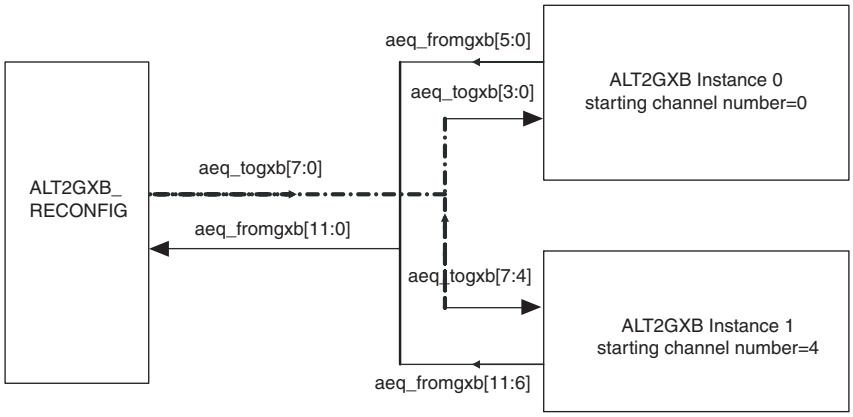
When you select this option, the ALT2GXB MegaWizard provides the following additional ports:

- `aeq_fromgxb[]`
- `aeq_togxb[]`
- `fixedclk`

The `aeq_fromgxb[]` and `aeq_togxb[]` ports provide the interface between the transceiver channel (ALT2GXB) and the dynamic reconfiguration controller (ALT2GXB_RECONFIG). For each channel, the width of the `aeq_fromgxb[]` and `aeq_togxb[]` ports are 6 bits and 4 bits, respectively. If you have multiple transceiver instances, connect the least significant byte of the `aeq_togxb[3:0]` and `aeq_fromgxb[5:0]` ports between the ALT2GXB_RECONFIG and the transceiver channel

with `logical_channel_address` value of **0**. Figure 3–67 shows the connections between multiple ALT2GXB instances and the dynamic reconfiguration controller.

Figure 3–67. Interface Connection Between the ALT2GXB and the ALT2GXB_RECONFIG Instance

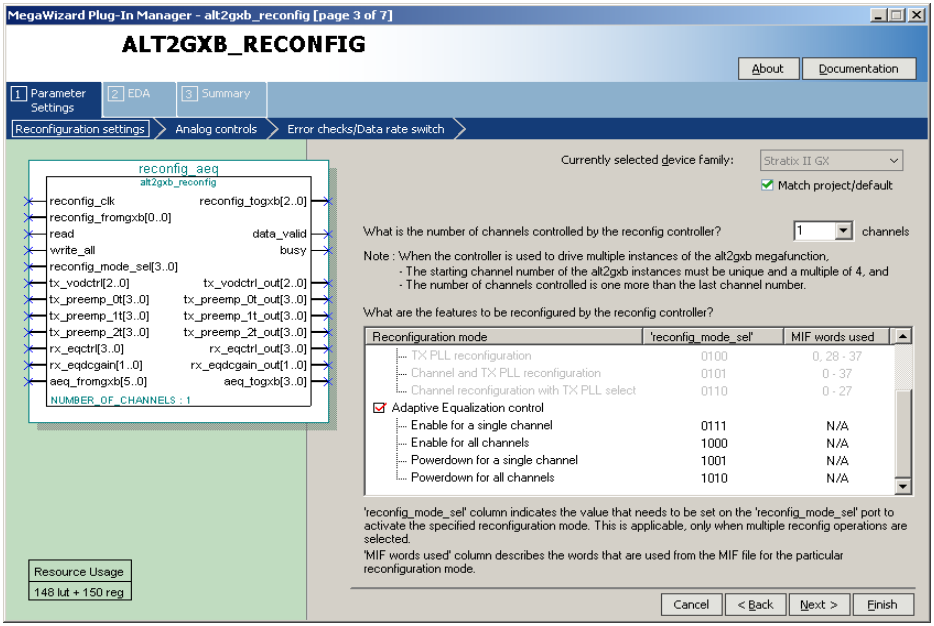


The `fixedclk` port provides the clock input to run the AEQ hardware. The range of the input clock frequency to the `fixedclk` port should be between 2.5 MHz and 125 MHz. To save clock routing resources, you can use the same clock pin to provide input clocks for the `fixedclk` and `reconfig_clk` ports.

When the transceiver channel is configured for PCI-Express (PIPE) protocol, the `fixedclk` is used to operate the receiver detect circuitry. In this protocol mode, `fixedclk` requires a fixed 125 MHz input clock frequency. The AEQ feature is not available in PCI-Express (PIPE) protocol mode.

The ALT2GXB_RECONFIG block provides a simple interface between the user logic and the transceiver channel to control the AEQ hardware. Figure 3–68 shows the ALT2GXB_RECONFIG MegaWizard page with different AEQ options.

Figure 3–68. ALT2GXB_RECONFIG with Different Adaptive Equalization Control Options



Controlling the AEQ Hardware

The ALT2GXB_RECONFIG provides different options to start and power down the adaptive equalization hardware. You can select these options by setting different values in the `reconfig_mode_sel[]` port. To use these options, set the `reconfig_mode_sel[]` port to the corresponding value shown in Table 3–22.

The dynamic reconfiguration controller provides the options shown in Table 3–22 to control the adaptive equalization operation.

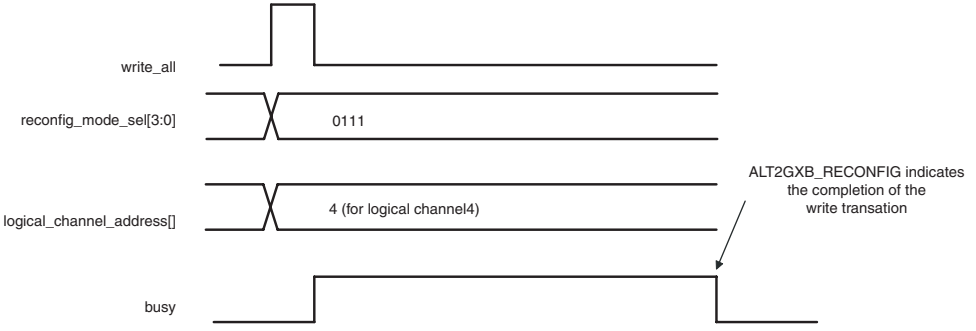
Table 3–22. reconfig_mode_sel Port Options for the Adaptive Equalization Feature

reconfig_mode_sel[3:0]	Adaptive Equalization Options
0111	Enable for a single channel
1000	Enable for all channels
1001	Power down for a single channel
1010	Power down for all channels

Enable for a Single Channel

This option, shown in Figure 3–68, provides the flexibility to start the adaptive equalization operation in a specific transceiver channel. To initiate the AEQ operation for a single channel, set the logical address of the channel in the `logical_channel_address[]` port. (For more information, refer to "Example for Using Logical Channel Address to Perform Channel Reconfiguration" on page 3–24). Set the `reconfig_mode_sel[3:0]` port to **0111** and assert the `write_all` signal for one `reconfig_clk` cycle, as shown in Figure 3–69.

Figure 3–69. AEQ Write Timing Diagram on a Single Channel



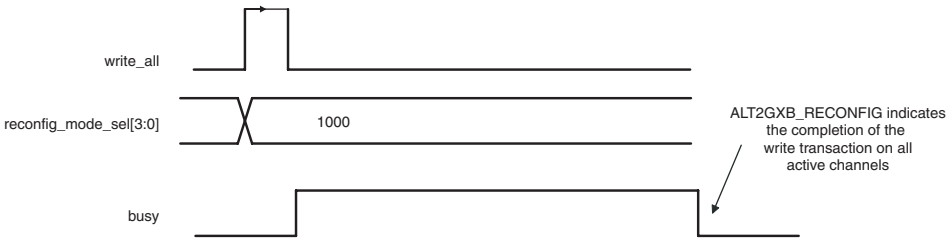
When the `write_all` signal is asserted, the dynamic reconfiguration controller writes the initialization and control values into the transceiver registers and initiates the adaptive equalization process. During this initialization process, the dynamic reconfiguration controller powers down the receiver buffer. This results in transient bit errors on the parallel interface on the receive side. The ALT2GXB_RECONFIG de-asserts the `busy` signal after the write transaction is completed.

Figure 3–69 shows the value **4** in the `logical_channel_address[]` signal as an example. The ALT2GXB_RECONFIG takes a maximum of approximately 7,000 `reconfig_clk` cycles to complete the AEQ write operation.

Enable for All Channels

This option allows you to initiate the adaptive equalization operation on all the active channels connected to the same dynamic reconfiguration controller. The method to initiate the AEQ operation for all channels is similar to "Enable for a Single Channel" on page 3–142. Set the `reconfig_mode_sel[3:0]` to **1000** and assert the `write_all` signal for one `reconfig_clk` cycle, as shown in Figure 3–70.

Figure 3–70. AEQ Write Timing Diagram on All Active Channels



The dynamic reconfiguration controller initiates the write transaction for all the active channels starting with the lowest logical channel that has the AEQ feature enabled. For example, assume that you have three channels connected to a dynamic reconfiguration controller with logical address values of **0**, **4**, and **8**, respectively. If the logical channels **4** and **8** have the AEQ feature enabled, when you use this option, the ALT2GXB_RECONFIG starts the AEQ write operation from logical address value of **8**.

The ALT2GXB_RECONFIG de-asserts the `busy` signal after the write transaction is completed for all active channels. The number of `reconfig_clk` cycles required to complete the AEQ write operation in this mode is approximately 7,000 multiplied by the number of active channels.

Power Down Options

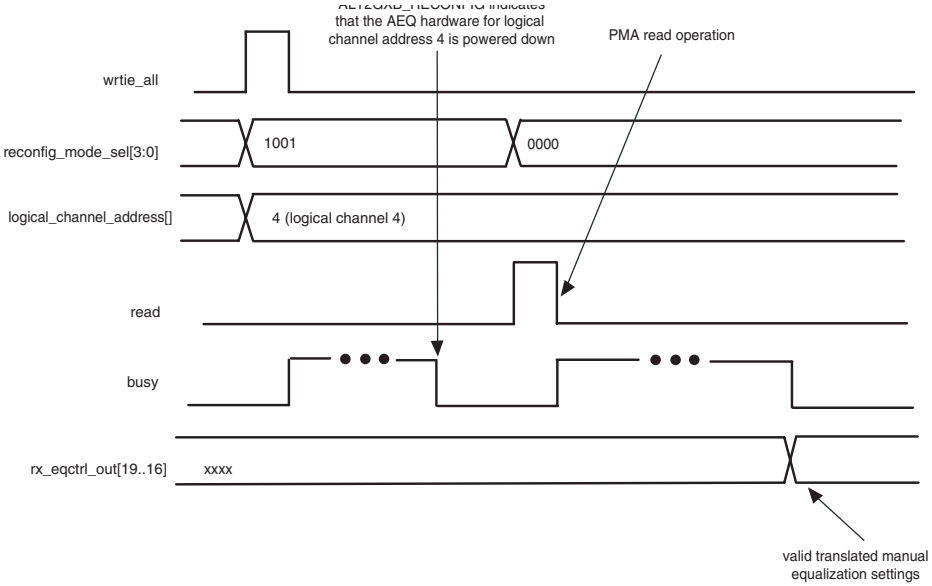
The AEQ hardware consumes approximately 80 mW power (typical) per channel. Therefore, the dynamic reconfiguration controller provides options to dynamically power down the AEQ hardware. The options are:

- Power down for a single channel
- Power down for all channels

Power Down for a Single Channel

When you use this option, the ALT2GXB_RECONFIG controller powers down the AEQ hardware in the selected channel specified by the `logical_channel_address` value. Before powering down the AEQ hardware, the ALT2GXB_RECONFIG reads the adaptive equalization settings, translates them to the nearest available manual equalization setting, and automatically writes the translated manual equalization settings into the transceiver channel. To read the translated manual equalization values, perform a read operation using the **PMA controls** option. The translated manual equalization values are available in the corresponding byte positions of the `rx_eqctrl_out` port. For example, assume that you perform an AEQ write using this option to power down the AEQ hardware in logical channel 4. When you perform a read operation using the **PMA controls** option, the translated manual equalization values are available in `rx_eqctrl_out` port in bits 19 down to 16, as shown in Figure 3–71.

Figure 3–71. AEQ Write Timing Diagram for Power Down for a Single Channel (Logical Channel 4)



For more information about using the **PMA controls** option, refer to "Channel and PMA Controls Reconfiguration" on page 3–20. The dynamic reconfiguration controller takes approximately 700 `reconfig_clk` cycles to complete the write transaction with this option.

During the power down process, there may be bit errors on the receiver output data for a few receive parallel clock cycles.

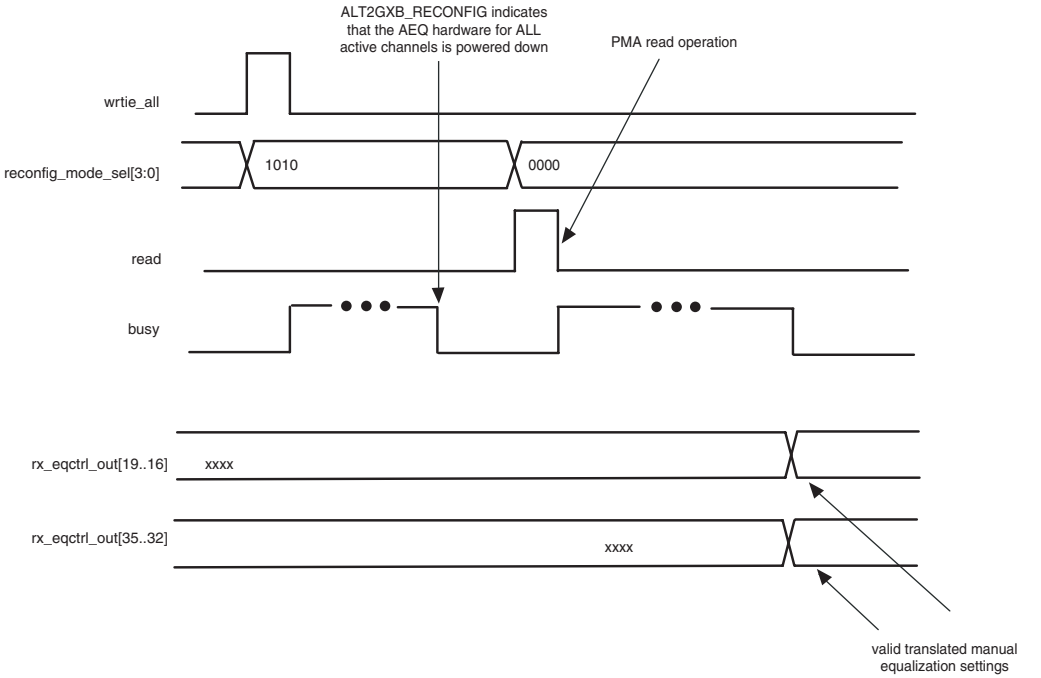
The ALT2GXB_RECONFIG translates the equalization values converged by the AEQ hardware and performs a rounding to the nearest manual equalization setting.

Power Down for All Channels

This option provides the flexibility to power down the AEQ hardware in all the active channels connected to the same dynamic reconfiguration controller. The ALT2GXB_RECONFIG performs the translation as explained in "Power Down Options" on page 3–144. The dynamic reconfiguration controller powers down the AEQ hardware on all the active channels starting with the lowest logical channel. For example, assume that you have three channels with logical address values of **0**, **4**, and **8**, respectively. If only logical channels **4** and **8** have the AEQ feature enabled, when you use this option, the ALT2GXB_RECONFIG starts the

power down operation from logical address value of **4**. To read the translated manual equalization values, perform a read operation using the **PMA controls** option. The translated manual equalization values are available in the corresponding byte positions of the `rx_eqctrl_out` port, as shown in Figure 3–72.

Figure 3–72. AEQ Write Timing Diagram for Power Down for All Active Channels



For more information on the byte positions of the PMA control input and output ports, refer to "Design Examples" on page 3–121. The number of `reconfig_clk` cycles that the dynamic reconfiguration controller takes is approximately 700 times the number of active channels connected to the dynamic reconfiguration controller. During the power down process, there may be bit errors on the receiver output data for few receive parallel clock cycles.

In addition to controlling the AEQ hardware, ALT2GXB_RECONFIG supports multiple features; for example, PMA controls, channel reconfiguration, etc. Therefore, only one operation (selected by `reconfig_mode_sel[]`) can be performed at any given time.

If the AEQ hardware is enabled for a channel, you can reconfigure the channel with the manual equalization values (using `reconfig_mode_sel` - **0000**) only after the AEQ hardware is powered down (using `reconfig_mode_sel` - **1001** or **1010**).

If you perform an unsupported AEQ operation, the dynamic reconfiguration controller waits for a pre-defined number of clock cycles for the AEQ operation to complete. If the `busy` signal does not get deasserted within the pre-defined number of `reconfig_clk` cycles, the dynamic reconfiguration controller de-asserts the `busy` signal.

Quartus II Software Merging Requirements

The Quartus II software has certain requirements for merging multiple transceiver channel instances in the same transceiver block, as discussed in "Merging Transceiver Channels with Dynamic Reconfiguration Enabled" on page 3–120.

In addition to the above requirements, when you enable the **adaptive equalization** option in the ALT2GXB MegaWizard for one transceiver instance, the Quartus II software requires that you enable this option in the other channels to merge them in the same transceiver block.

Summary

Using the dynamic reconfiguration feature, you can reconfigure the analog controls, data rates, and protocols of the transceiver without requiring a system power down. These features provide a flexible and effective solution for various line card and backplane applications.

Referenced Documents

This chapter references the following documents:

- *ALT2GXB_RECONFIG Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Stratix II GX ALT2GXB Megafunction User Guide* chapter in volume 2 of the *Stratix II GX Device Handbook.*
- "Stratix II GX ALT2GXB Ports List" on page 2–2 in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Stratix II GX Transceiver Architecture Overview* chapter (the Reset Control and Power Down section) in volume 2 of the *Stratix II GX Handbook.*

Document Revision History

Table 3–23 shows the revision history for this chapter.

Table 3–23. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007, v1.1	Added: • "Adaptive Equalization (AEQ)" • "Using Dedicated refclks"	—
	Updated: • "Reconfig Clks 1 Tab" • "Channel and PMA Controls Reconfiguration" • "Example 3"	—
	Updated: • Figure 3–1 • Figure 3–2 • Figure 3–19 • Figure 3–30 • Figure 3–31 • Figure 3–45 • Figure 3–58 • Figure 3–59 • Figure 3–60 • Figure 3–62	—
	Updated Table 3–2.	—
	Added the "Referenced Documents" section.	—
	Minor text edits.	—
August 2007, v1.0	Moved the "Introduction" section from the *Stratix II GX Architecture Overview* chapter to this chapter.	—
	Updated the "Introduction" and "Channel and PMA Controls Reconfiguration" sections.	—
	Initial release of the "Channel and Clock Multiplier Unit (CMU) PLL Reconfiguration" section.	—

4. Stratix II GX ALT2GXB Megafunction User Guide

SIIGX52003-4.2

Introduction

The MegaWizard® Plug-In Manager in the Quartus® II software creates or modifies design files that contain custom megafunction variations that can then be instantiated in a design file. The MegaWizard Plug-In Manager provides a MegaWizard that allows you to specify options for the ALT2GXB megafunction. You can use the MegaWizard to set the ALT2GXB megafunction features in the design.

Start the MegaWizard Plug-In Manager using one of the following methods:

- Choose the **MegaWizard Plug-In Manager** command (Tools menu).
- When working in the Block Editor, click **MegaWizard Plug-In Manager** in the **Symbol** dialog box (Edit menu).
- Start the stand-alone version of the MegaWizard Plug-In Manager by typing the following command at the command prompt: `qmegawiz`.

The ALT2GXB MegaWizard Plug-In Manager allows you to configure one or more transceiver channels. It also allows you to enable the dynamic reconfiguration feature for these channels, depending on your system requirements.

Figure 4–1 shows the first page of the MegaWizard Plug-In Manager. To generate an ALT2GXB custom megafunction variation, select **Create a new custom megafunction variation**.

Figure 4–1. MegaWizard Plug-In Manager (Page 1)

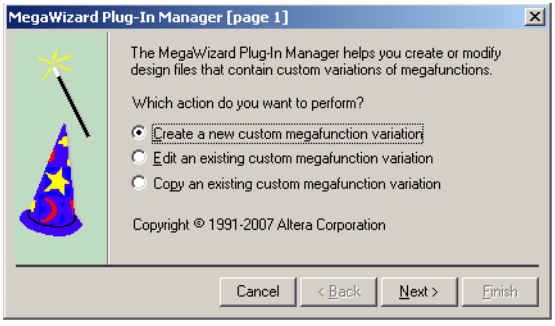
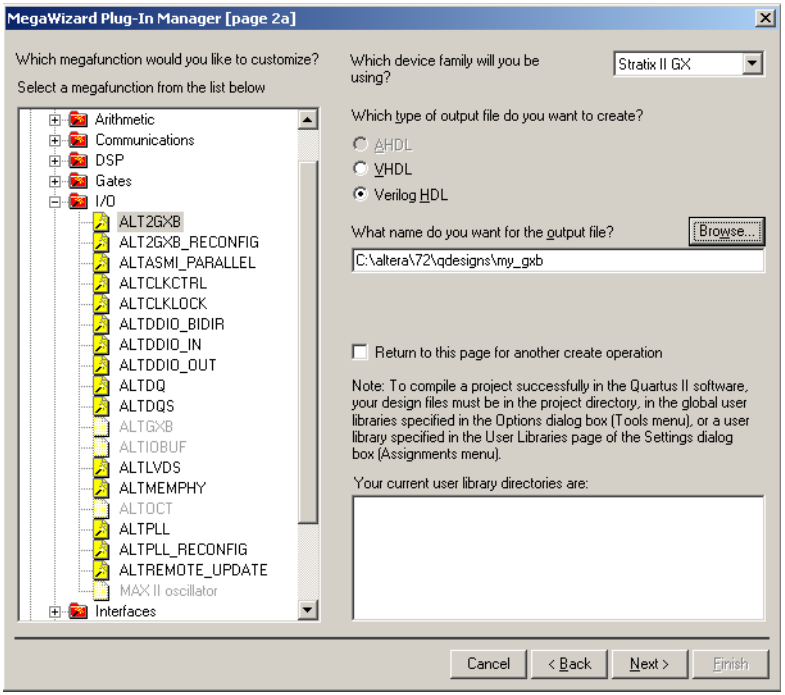


Figure 4–2 shows the second page of the MegaWizard Plug-In Manager. Select the **Stratix II GX** device as the device family.

Figure 4–2. MegaWizard Plug-In Manager (Page 2)



Basic Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for Basic mode.

The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

Figure 4–3 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager in Basic mode.

Figure 4–3. MegaWizard Plug-In Manager - ALT2GXB (General)

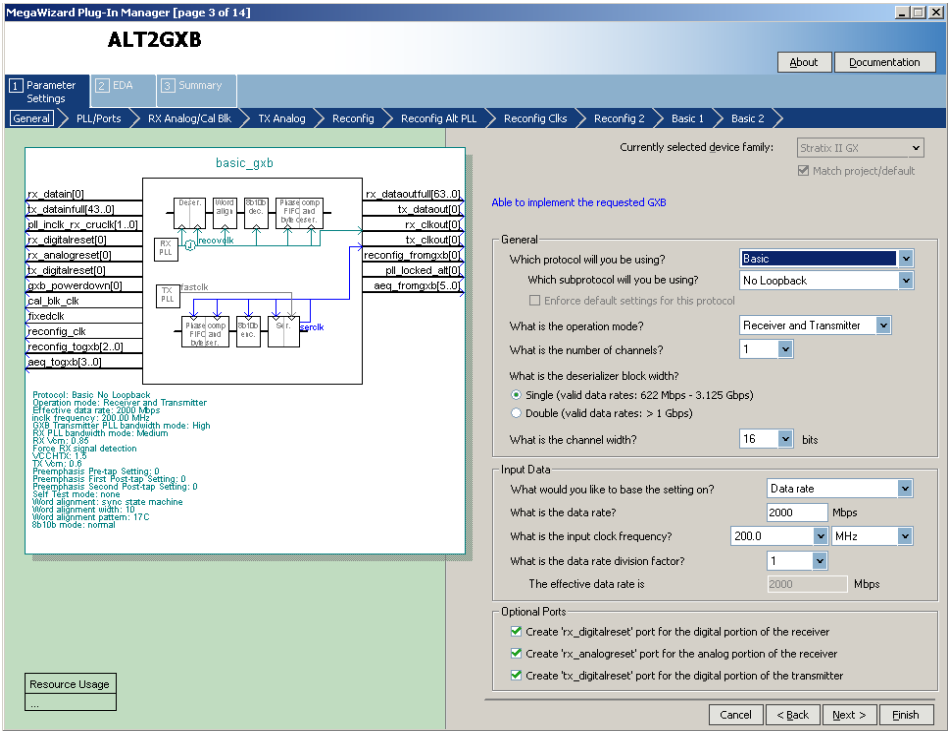


Table 4–1 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–1. MegaWizard Plug-In Manager Options (Page 3 for Basic Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Determines the specific protocol or modes under which the transceiver operates. For the Basic mode, you must select the **Basic** protocol.	
Which subprotocol will you be using?	In Basic mode, the subprotocols are the diagnostic modes. The available options are as follows: • No loopback – This is the normal operation of the transceiver. • Serial loopback – This mode loops the user data from the transmitter path back to the receiver path right before the buffers. The serial loopback can be controlled dynamically. • Parallel loopback/BIST – This mode loops the parallel data from the BIST (non PRBS) back to the BIST verifier in the receiver path. Parallel Loopback is allowed only in Basic Double-Width mode. • Reverse serial loopback – This is a loopback after the receiver's CDR block to the transmitter buffer. The RX path in the PCS is active but the TX side is not. • Reverse serial loopback (pre-CDR) – This is the loopback before the receiver's CDR block to the transmitter buffer. The RX path in the PCS is active but the TX side is not. • PRBS/Serial loopback – This is another serial loopback mode, but with the PRBS BIST block active. The PRBS pattern depends on the SERDES factor. • X4 – This mode can be used to implement SFI-5 interface. In this mode, all four channels within the transceiver block are clocked from its central clock divider block to minimize the transmitter channel-to-channel skew.	Loopback Modes and Built-In Self-Test Modes sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enforce default settings for this protocol	This selection is not active in Basic mode because there is no pre-defined protocol.	
What is the operation mode?	The available operation modes are receiver only, transmitter only, and receiver and transmitter.	
What is the number of channels?	This option determines how many duplicate channels this ALT2GXB instance contains.	

Table 4–1. MegaWizard Plug-In Manager Options (Page 3 for Basic Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
What is the deserializer block width?	This option sets the transceiver data path width. • Single width – This operates from 600 Mbps to 3.125 Gbps. The features of each block may differ from the double-width mode. • Double width – This mode operates from 1 Gbps to 6.375 Gbps. The features of each block in this mode may differ from the single-width mode.	
What is the channel width?	This option determines the transceiver to PLD interface width. • In single-width mode, selecting 8 or 10 bits bypasses the byte serializer/deserializer. If you select 16 or 20 bits, the byte serializer/deserializer is used. • In double-width mode, selecting 16 or 20 bits bypasses the byte serializer/deserializer. Any width greater uses the byte serializer/deserializer.	Byte Serializer and Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option allows you to do one of following: • Enter a data rate and select an input clock frequency through a pull-down menu (with the data rate selection). • Enter your input clock frequency through a pull-down menu (with the data rate selection) or enter your input clock frequency and select from the available data rates for a clock frequency.	
What is the data rate?	Determines the TX and RX PLL VCO frequency.	
What is the input clock frequency?	Determines the input clock frequency you want as a reference clock for the transceiver.	
What is the data rate division factor?	This setting, in conjunction with the selected data rate, determines the effective data rate for the transceiver channel. Division factors of 1, 2, and 4 are available. For example, a data rate setting of 3000 Mbps and data rate division factor of 2 yields an effective data rate of 1500 Mbps.	
Create `rx_digitalreset` port	Receiver digital reset port. Resets the PCS portion of the receiver. Altera® recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–1. MegaWizard Plug-In Manager Options (Page 3 for Basic Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_analogreset` port	Receiver analog reset port.	Reset Control and Power Down" section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_digitalreset` port	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–4 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for Basic mode.

Figure 4–4. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

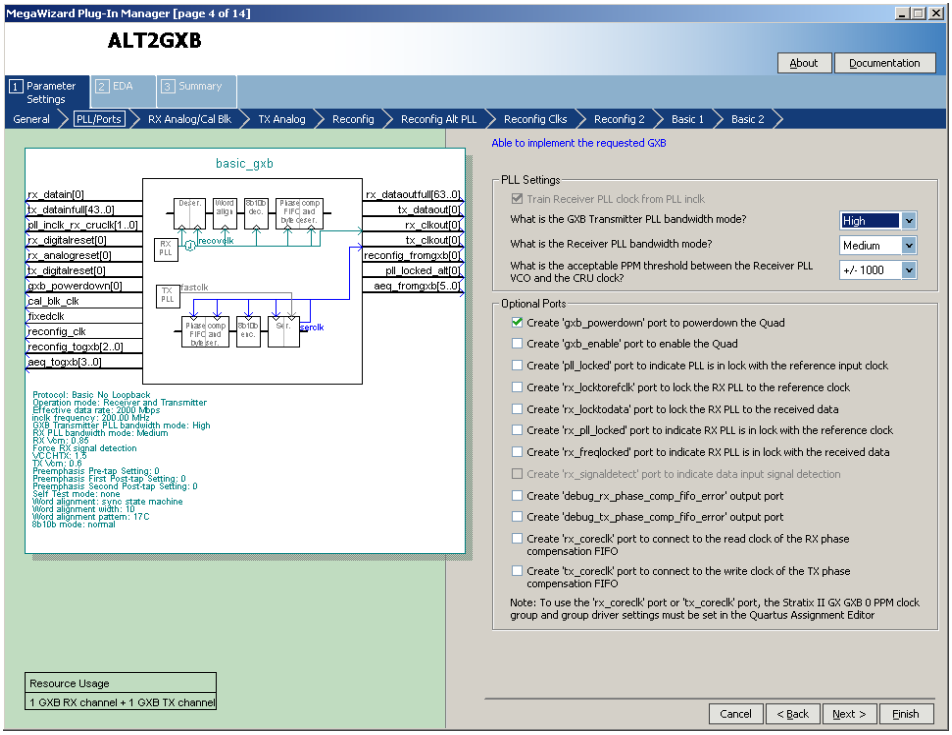


Table 4–2 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–2. MegaWizard Plug-In Manager Options (Page 4 for Basic Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If **CMU PLL reconfiguration** is enabled, this option is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	Transmitter PLL bandwidth selection of low, medium, and high. The recommendations will be determined by characterization.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver PLL bandwidth mode?	Receiver PLL bandwidth selection of low, medium, and high. The recommendations will be determined by characterization.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver clock recovery unit (CRU) switchover between lock-to-data and lock-to-reference. (There are additional factors that affect the CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–2. MegaWizard Plug-In Manager Options (Page 4 for Basic Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicated data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–2. MegaWizard Plug-In Manager Options (Page 4 for Basic Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–5 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for Basic mode.

Figure 4–5. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

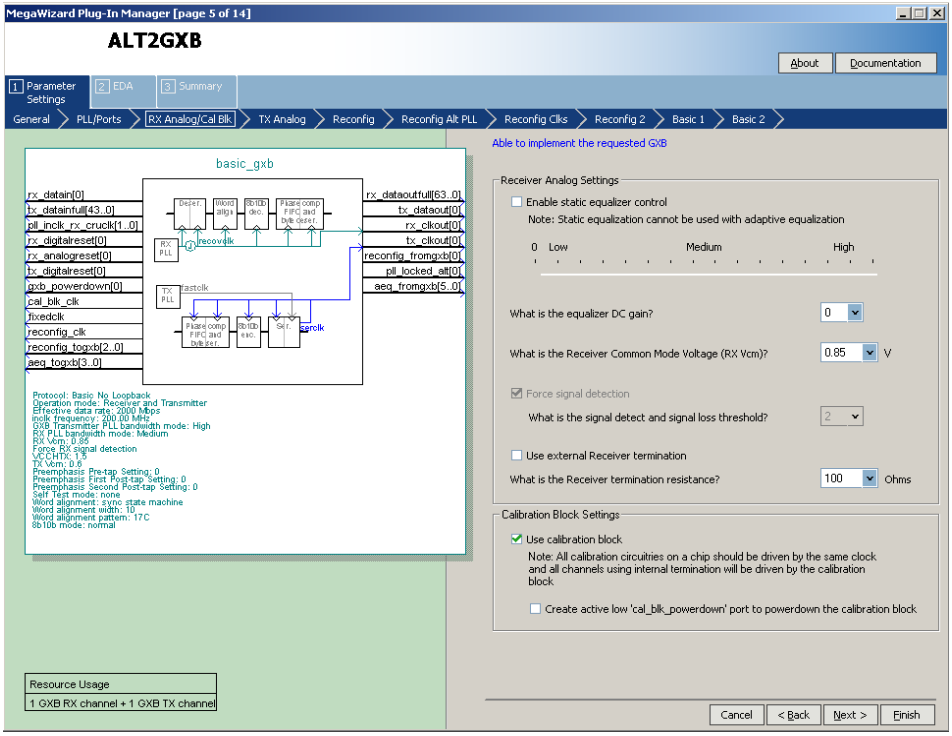


Table 4–3 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–3. MegaWizard Plug-In Manager Options (Page 5 for Basic Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is programmable between 0.85 V and 1.2 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Force signal detection	This option disables the signal detect circuit. This removes the signal detect criterion for the receiver CRU lock-to-reference and lock-to-data switchover. This option is available only in PIPE mode.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	Use this option when the forced signal detection option is off and to set the trip point of the signal detect circuit. This option is available only in PIPE mode.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–3. MegaWizard Plug-In Manager Options (Page 5 for Basic Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create active low `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–6 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for Basic mode.

Figure 4–6. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

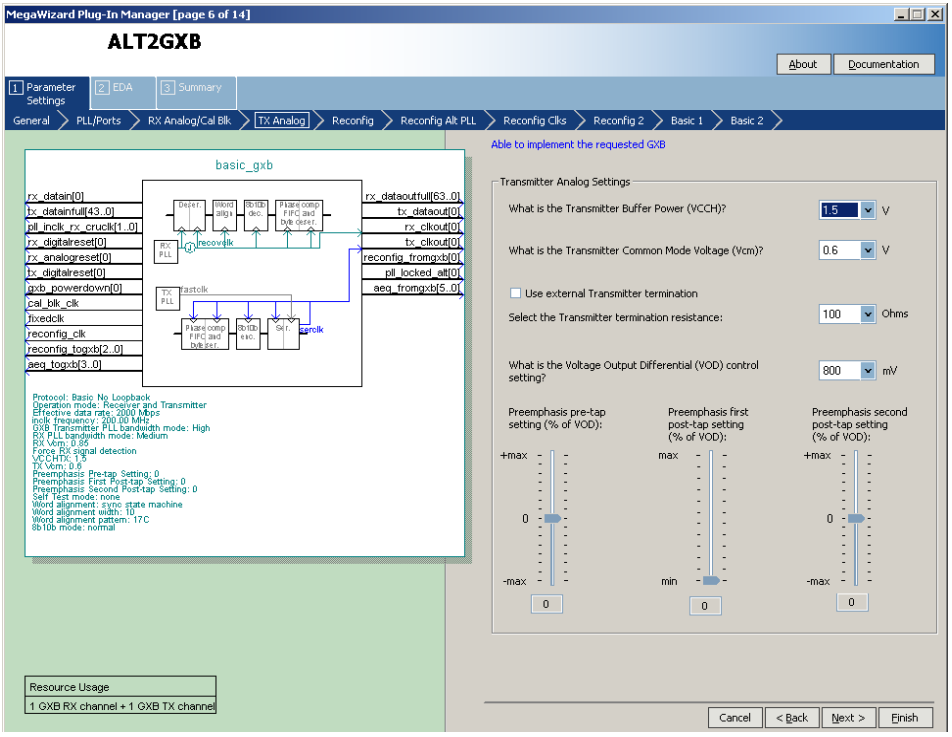


Table 4–4 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–4. MegaWizard Plug-In Manager Options (Page 6 for Basic Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. The selections available are 1.2 V and 1.5 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Restrictions apply based on the V_{CCH} setting.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–4. MegaWizard Plug-In Manager Options (Page 6 for Basic Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap.	

Figure 4–7 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for Basic mode.

Figure 4–7. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

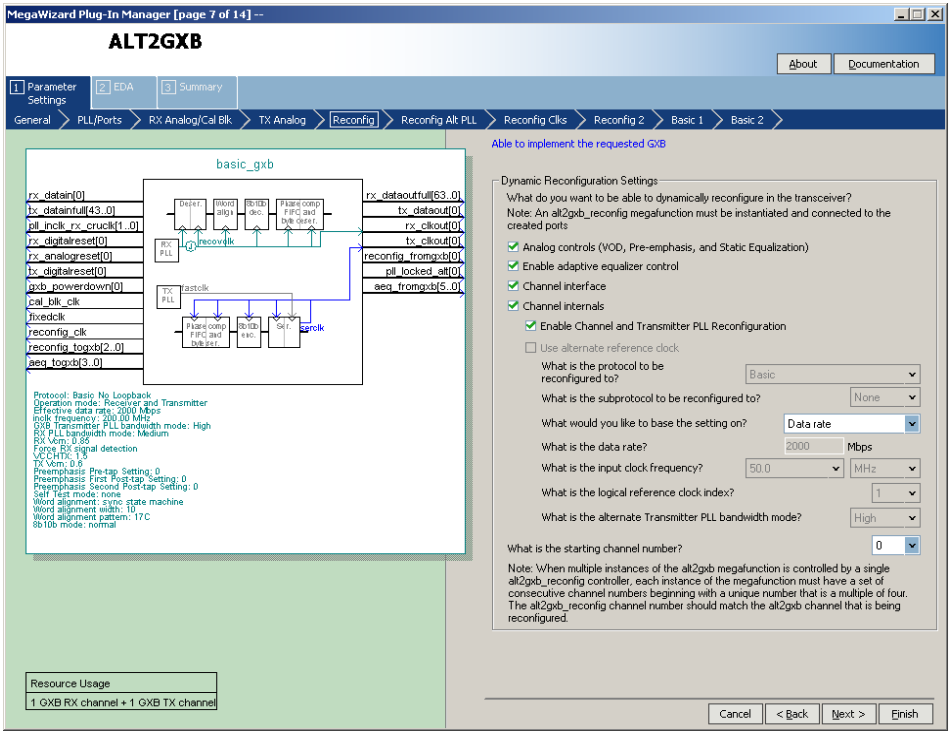


Table 4–5 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–5. MegaWizard Plug-In Manager Options (Page 7 for Basic Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are: • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals. • Channel internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but have the same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–8 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for Basic mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–8. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

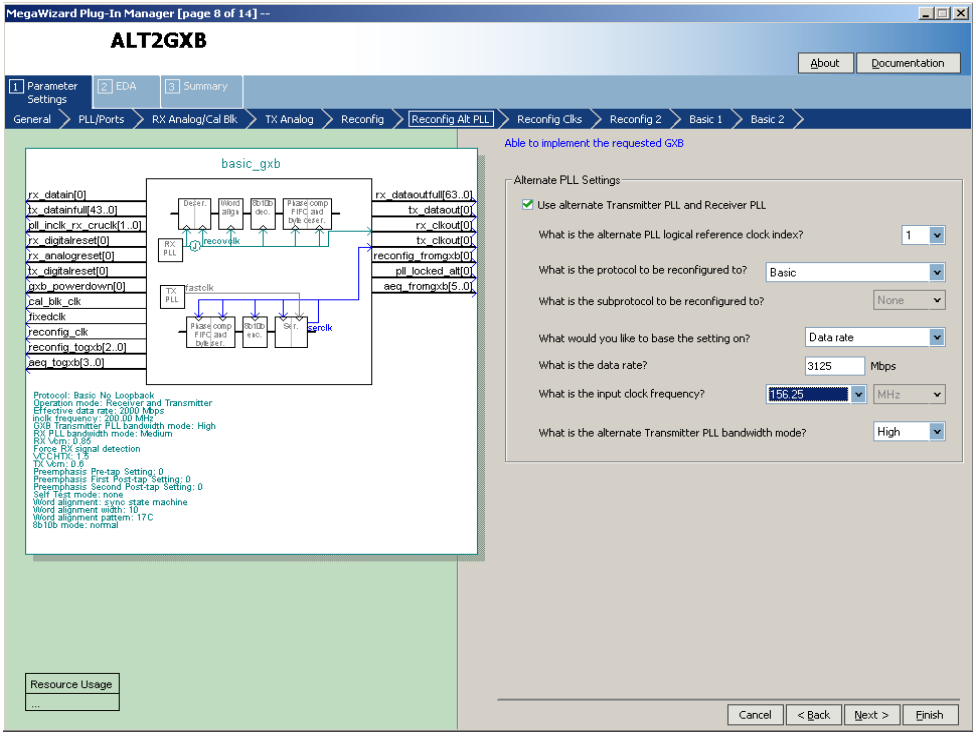


Table 4–8 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–6. MegaWizard Plug-In Manager Options (Page 8 for Basic Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–9 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for Basic mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–9. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

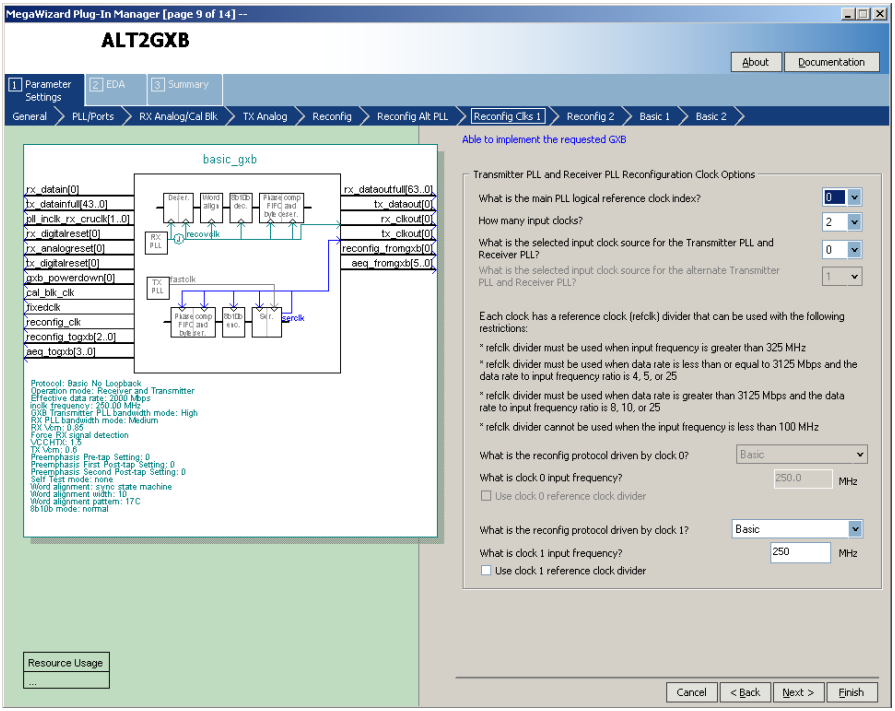


Table 4–7 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–7. MegaWizard Plug-In Manager Options (Page 9 for Basic Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–10 shows page 10 of the ALT2GXB MegaWizard Plug-In Manager for Basic mode. This page appears only if the **Channel Internals** or the **Channel Interface** option is selected in the **Reconfig** page (Page 7).

Figure 4–10. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

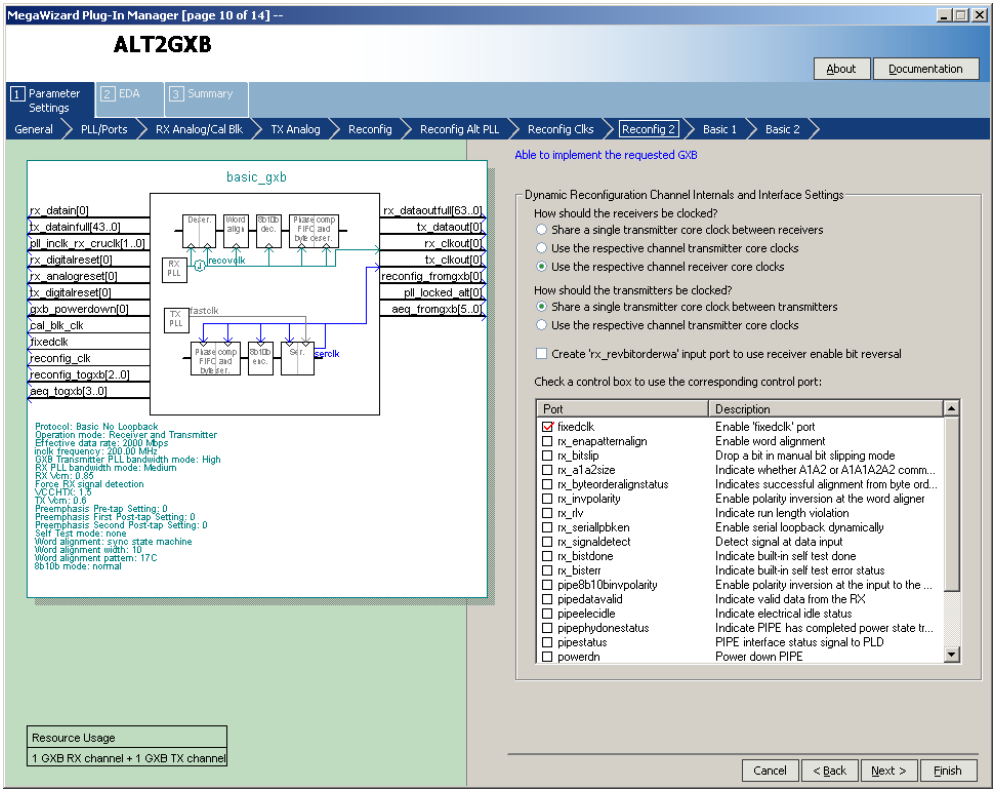


Table 4–8 describes the available options on page 10 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–8. MegaWizard Plug-In Manager Options (Page 10 for Basic Mode)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–11 shows page 11 of the MegaWizard Plug-In Manager for the Basic protocol mode set up.

Figure 4–11. MegaWizard Plug-In Manager - ALT2GXB (Basic 1)

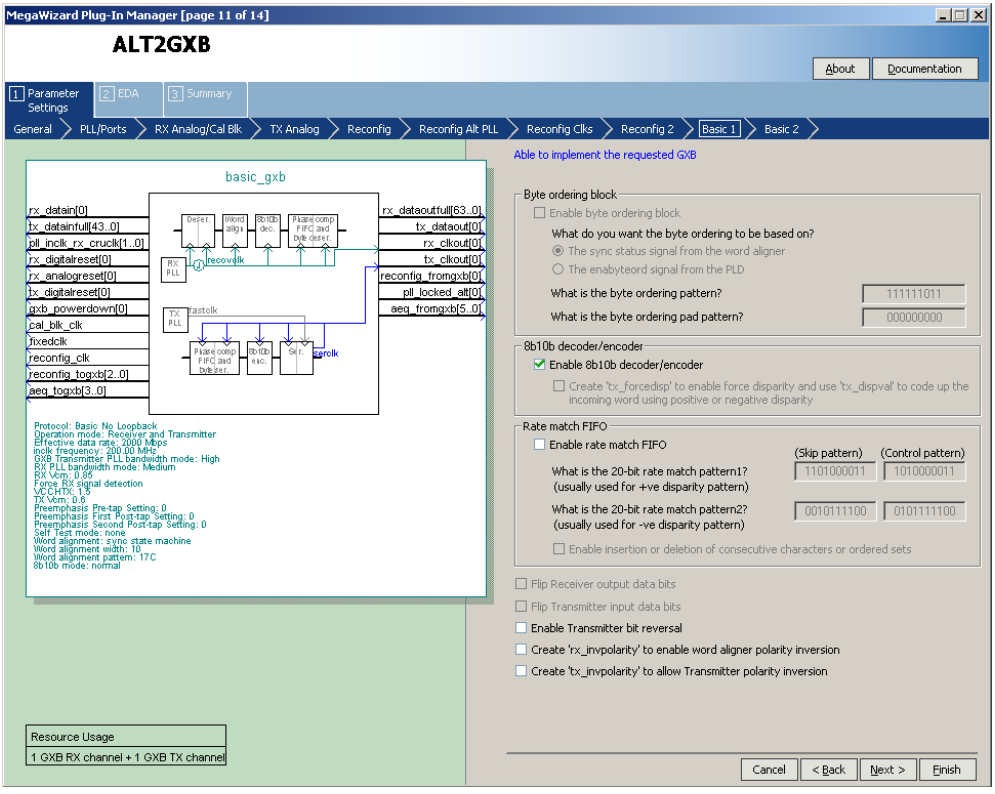


Table 4–9 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–9. MegaWizard Plug-In Manager Options (Page 11 for Basic Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable byte ordering block	This option enables the byte ordering block. For the Basic protocol mode, this is only available in double-width mode with the 8B/10B decoder.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What do you want the byte ordering to be based on?	This option allows you to trigger the byte ordering block either on the rising edge of `rx_syncstatus` signal or user-controlled `rx_enabyteord` signal from the PLD.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the byte ordering pattern?	Enter the 10-bit pattern that the byte ordering block must place in the LSByte position of the receiver-PLD interface bus.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the byte ordering pad pattern?	Enter the pad pattern that the byte ordering block inserts until the byte ordering pattern can be placed in the LSByte position.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Enable 8B/10B decoder/encoder	This option enables the 8B/10B encoder and decoder. This option is only available if the channel width is a multiple of 8 bits.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_forcedisp` to enable Force disparity and use `tx_dispval` to code up the incoming word using positive or negative disparity	This option allows you to force positive or negative disparity on transmitted data in 8B/10B configurations.	8B/10B Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Enable rate match FIFO	This option enables the rate matcher and is only available with the 8B/10B decoder.	Rate Matcher section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–9. MegaWizard Plug-In Manager Options (Page 11 for Basic Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the 20-bit rate match pattern1? (usually used for +ve disparity pattern)	Enter the positive disparity rate matcher pattern and control code pattern. The skip pattern is used for insertion or deletion, and the control pattern identifies which group of skip patterns to use for rate matching. If only one disparity is needed for rate matching, you can enter the same pattern for both rate matching patterns (pattern1 and pattern2).	Rate Matcher section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the 20-bit rate match pattern2? (usually used for -ve disparity pattern)	Enter the negative disparity rate matcher pattern and control code pattern. The skip pattern is used for insertion or deletion, and the control pattern identifies which group of skip patterns to use for rate matching. If only one disparity is needed for rate matching, you can enter the same pattern for both rate matching patterns (pattern1 and pattern2).	Rate Matcher section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Flip Receiver output data bits	This option reverses the bit order of the data at the receiver-PLD interface at a byte level to support MSBit to LSBit transmission protocols. The default transmission order is LSBit to MSBit.	
Flip Transmitter input data bits	This option reverses the bit order of the data bits at the input of the transmitter at a byte level to support MSBit to LSBit transmission protocols. The default transmission order is LSBit to MSBit.	
Enable Transmitter bit reversal	This option inverts (flips) the bit order of the data bits at the transmitter PCS-PMA interface at a byte level to support MSBit to LSBit transmission protocols. The default transmission is LSBit to MSBit.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–12 shows page 12 of the MegaWizard Plug-In Manager for the Basic protocol mode set up.

Figure 4–12. MegaWizard Plug-In Manager - ALT2GXB (Basic 2)

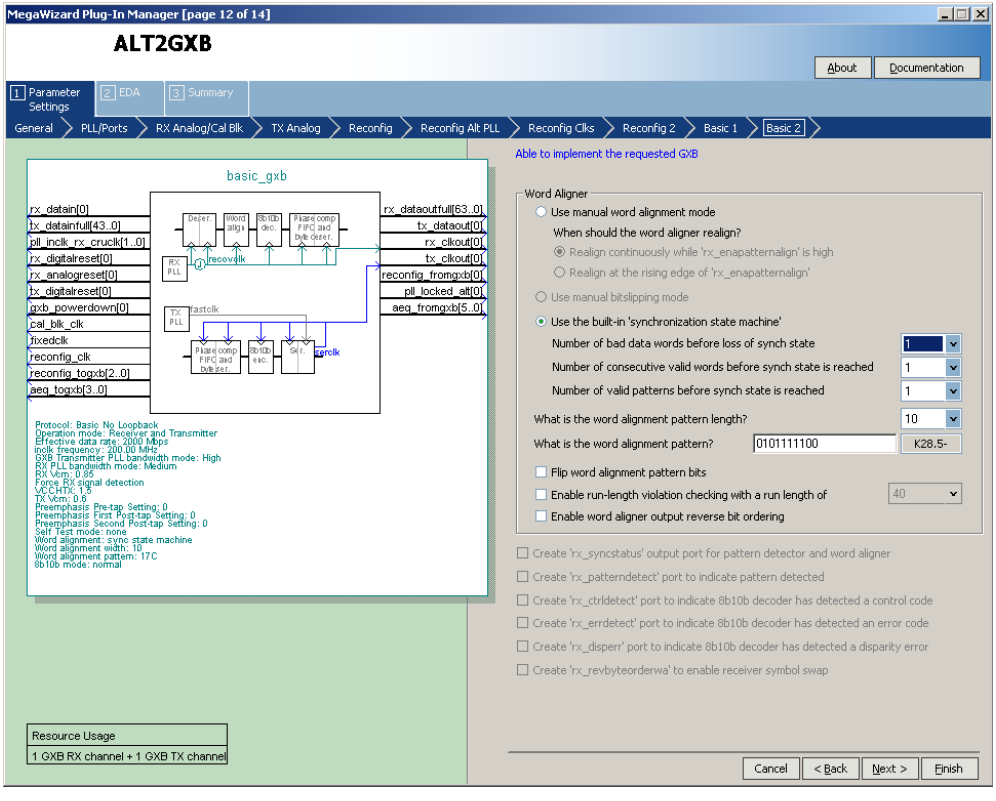


Table 4–10 describes the available options on page 12 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–10. MegaWizard Plug-In Manager Options (Page 12 for Basic Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Use manual word alignment mode	This option sets the word aligner in manual alignment mode. (Manual alignment, bit-slipping, and the built-in state machine are mutually exclusive options.)	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
When should the word aligner realign?	This option sets the behavior of the `rx_enapatternalign` signal to either edge or level sensitive. Altera recommends using edge sensitive for scrambled data (non-8B/10B) traffic and level sensitive for 8B/10B traffic.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use manual bit slipping mode	This option sets the word aligner to use the bit-slip port to alter the byte boundary one bit at a time.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use the built-in 'synchronization state machine'	This option sets the word aligner to use the built-in synchronization state machine. The behavior is similar to the PIPE synchronization state machine with adjustable synchronization thresholds.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Number of bad data words before loss of synch state	Use this option with the built-in state machine to transition from a synchronized state to an unsynchronized state.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Number of consecutive valid words before synch state is reached	This option sets the word aligner to check for a given number of good code groups. Use this option with the built-in state machine in conjunction with the **Number of valid patterns before synchronization state is reached** option to achieve synchronization.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Number of valid patterns before synch state is reached	This option checks for the number of valid alignment patterns seen. Use this option with the built-in state machine in conjunction with the **Number of consecutive valid words before synch state is reached** option to achieve synchronization.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–10. MegaWizard Plug-In Manager Options (Page 12 for Basic Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
What is the word alignment pattern length?	This option sets the word alignment length. The available choices depend on whether 8B/10B is used and which mode (single or double width) is used.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the word alignment pattern?	Enter the word alignment pattern here. The length of the alignment pattern is based on the word alignment pattern length. In bit-slip mode, this option triggers the `rx_patterndetect`.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Flip word alignment pattern bits	This option reverses the bit order of the alignment pattern at a byte level to support MSB to LSB transmission protocols. The default transmission order is LSB to MSB.	
Enable run-length violation checking with a run length of	This option activates the run-length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Enable word aligner output reverse bit ordering	In manual bit-slip mode, this option creates an input port `rx_revbitorderwa` to dynamically reverse the bit order at the output of the receiver word aligner. In other Basic modes, this option statically configures the receiver to always reverse the bit order of the data at the output of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_patterndetect` port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–10. MegaWizard Plug-In Manager Options (Page 12 for Basic Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_ctrldetect` port to indicate 8B/10B decoder has detected a control code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_errdetect` port to indicate 8B/10B decoder has detected an error code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_disperr` port to indicate 8B/10B decoder has detected a disparity code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbyteorderwa` to enable receiver symbol swap	This option is available only in Basic double-width mode. It creates an `rx_revbyteorderwa` port to dynamically swap the MSByte and LSByte of the data at the output of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–13 shows page 13 of the MegaWizard Plug-In Manager for the Basic protocol mode set up. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for the third party EDA synthesis tool to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–13. MegaWizard Plug-In Manager - ALT2GXB (EDA)

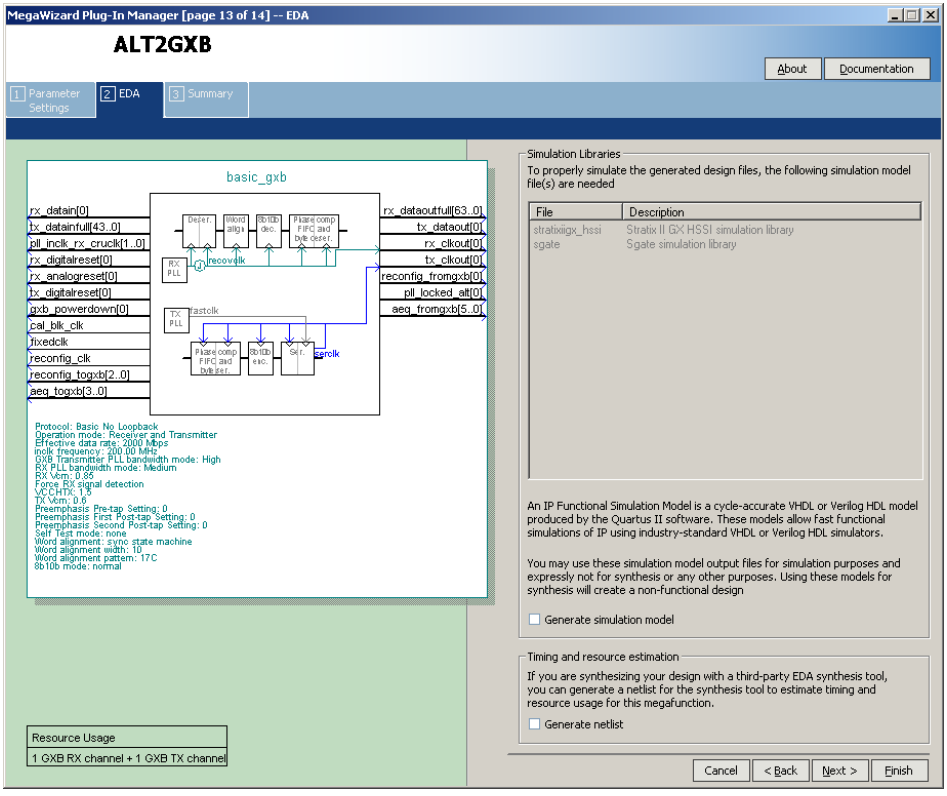
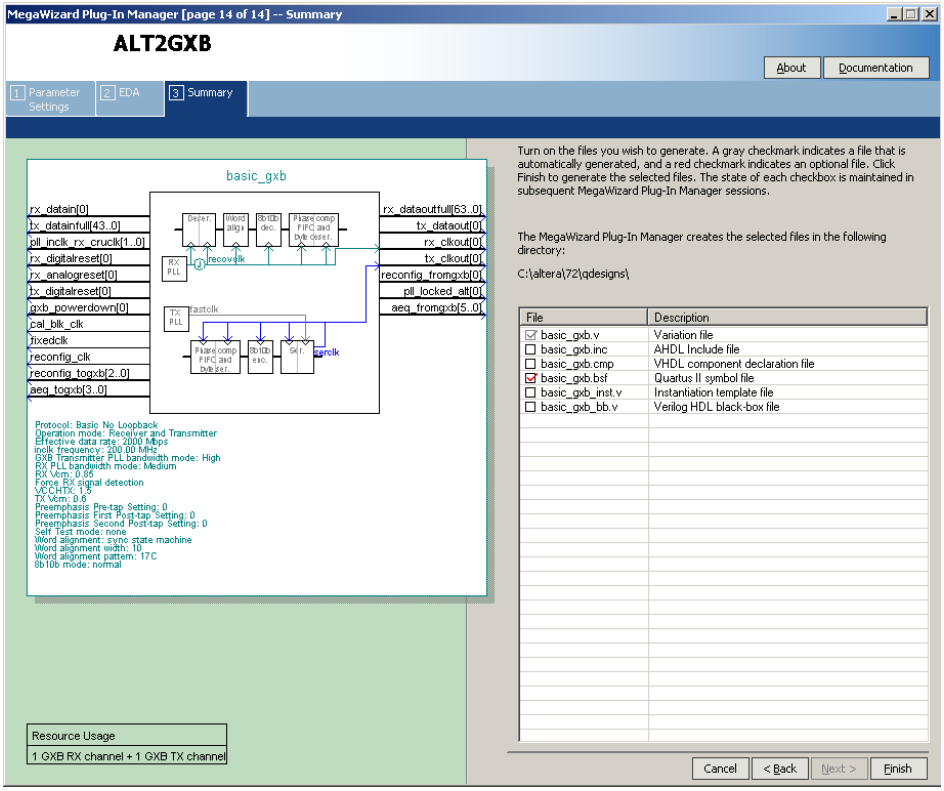


Figure 4–14 shows page 14 (last page) of the MegaWizard Plug-In Manager for the Basic protocol mode set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–14. MegaWizard Plug-In Manager - ALT2GXB (Summary)



Physical Interface for PCI-Express (PIPE) Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for the PIPE mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

The word aligner and rate matcher operations and patterns are pre-configured for the PIPE mode and cannot be altered.

Figure 4–15 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager for PIPE mode.

Figure 4–15. MegaWizard Plug-In Manager - ALT2GXB (General)

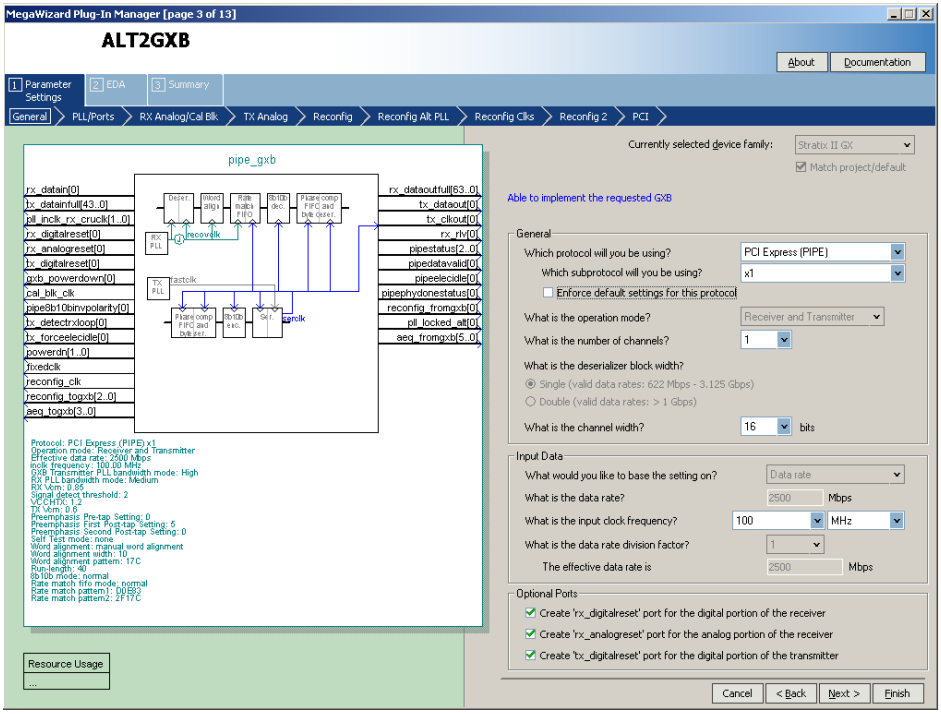


Table 4–11 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–11. MegaWizard Plug-In Manager Options (Page 3 for PIPE Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Determines the specific protocol or modes under which the transceiver operates. For the PIPE mode, you must select the **PCI Express (PIPE)** protocol.	
Which subprotocol will you be using?	In PIPE mode, the subprotocols are the lane configurations: ×1, ×4, or ×8 modes.	PIPE Mode and Clock Multiplier Unit sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enforce default settings for this protocol	Selecting this option skips the **PCI** page in the PIPE MegaWizard Plug-In Manager. The **PCI** page allows you to select the PIPE-specific ports for your design. If you select this option, all PIPE-specific ports are used.	
What is the operation mode?	Only the receiver and transmitter (full duplex) mode is allowed in the PIPE mode. Receiver only and transmitter only modes are not allowed.	
What is the number of channels?	This determines how many duplicate channels this ALT2GXB instance contains. In a ×4 subprotocol, the number of channels increments by 4. In a ×8 subprotocol, the number of channels increment by 8.	
What is the deserializer block width?	PIPE mode only operates in a single-width mode. Double-width mode is not allowed.	
What is the channel width?	This option determines the transceiver to PLD interface width. In PIPE mode, 8 bits and 16 bits are allowed.	
What would you like to base the setting on?	This option is not used because the data rate is fixed at 2.5 Gbps for PIPE mode.	
What is the data rate?	This option is not used because the data rate is fixed at 2.5 Gbps for PIPE mode.	
What is the input clock frequency?	Determines the input reference clock frequency for the transceiver. In PIPE mode, only 100 MHz is allowed.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–11. MegaWizard Plug-In Manager Options (Page 3 for PIPE Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the data rate division factor?	This option is not used because the data rate is fixed at 2.5 Gbps for PIPE mode.	
Create `rx_digitalreset` port for the digital portion of the receiver	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_analogreset` port for the analog portion of the receiver	Receiver analog reset port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_digitalreset` port for the digital portion of the transmitter	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–16 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for PIPE mode.

Figure 4–16. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

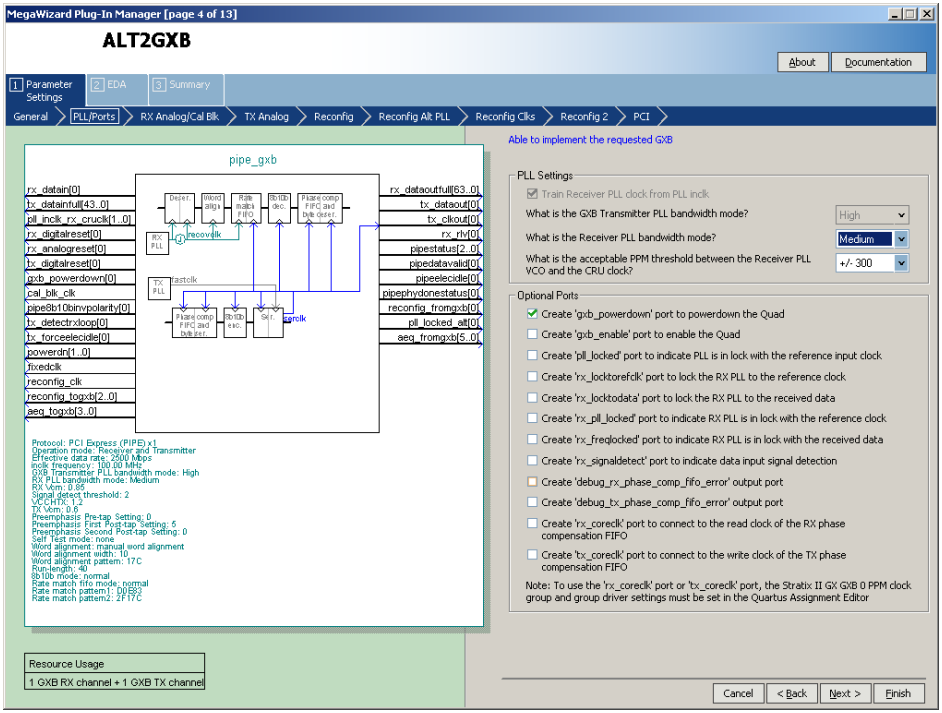


Table 4–12 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–12. MegaWizard Plug-In Manager Options (Page 4 for PIPE Mode) (Part 1 of 4)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If **CMU PLL reconfiguration** is enabled, this option is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	This option is not available in PIPE mode because the transmitter PLL bandwidth is fixed at high.	

Table 4–12. MegaWizard Plug-In Manager Options (Page 4 for PIPE Mode) (Part 2 of 4)

ALT2GXB Setting	Description	Reference
What is the Receiver PLL bandwidth mode?	This option is not available in PIPE mode because the receiver PLL bandwidth is fixed at medium.	
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver CRU switchover between lock-to-data and lock-to-reference. (There are additional factors that affect the CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–12. MegaWizard Plug-In Manager Options (Page 4 for PIPE Mode) (Part 3 of 4)

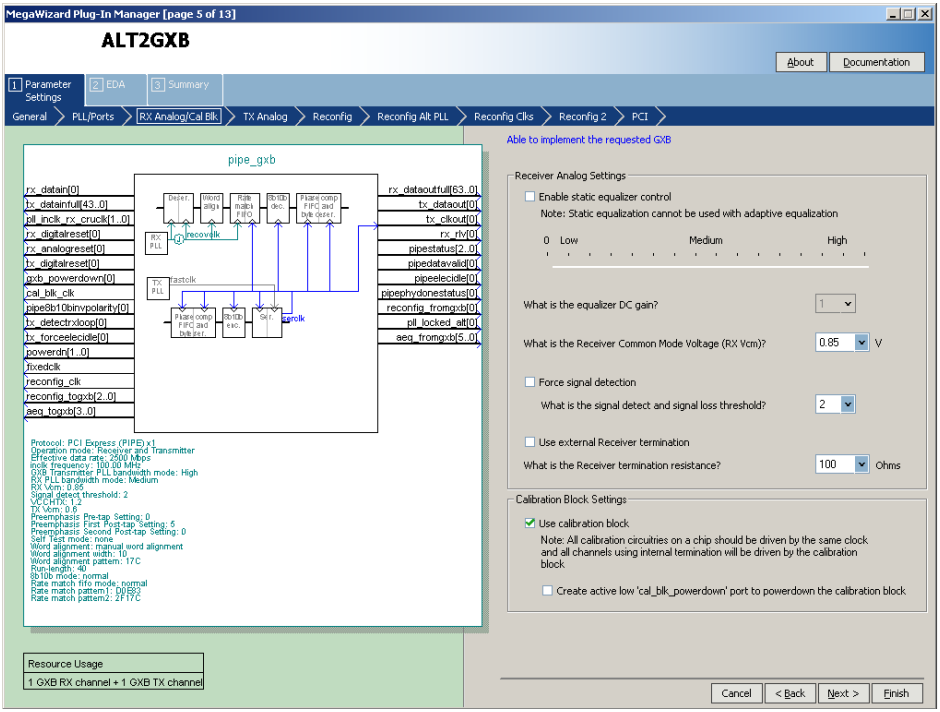
ALT2GXB Setting	Description	Reference
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_signaldetect` port to indicate data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–12. MegaWizard Plug-In Manager Options (Page 4 for PIPE Mode) (Part 4 of 4)

ALT2GXB Setting	Description	Reference
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–17 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for PIPE mode.

Figure 4–17. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk) *Note (1)*



Note to Figure 4–17:

(1) If the equalizer DC gain is controlled by the `ALT2GXB_RECONFIG` controller, the `rx_eqdcgain` input to the `ALT2GXB_RECONFIG` controller should be tied to "01" to be PCI E-compliant.

Table 4–13 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–13. MegaWizard Plug-In Manager Options (Page 5 for PIPE Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. In PIPE mode, a DC gain setting of 1 is forced.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is set to 0.85 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Force signal detection	This option disables the signal detect circuit. This removes the signal detect criterion for the receiver CRU lock-to-reference and lock-to-data switchover.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	Use this option when the forced signal detection option is off and to set the trip point of the signal detect circuit. The levels are to be determined after characterization.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–13. MegaWizard Plug-In Manager Options (Page 5 for PIPE Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create active low `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–18 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for PIPE mode.

Figure 4–18. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

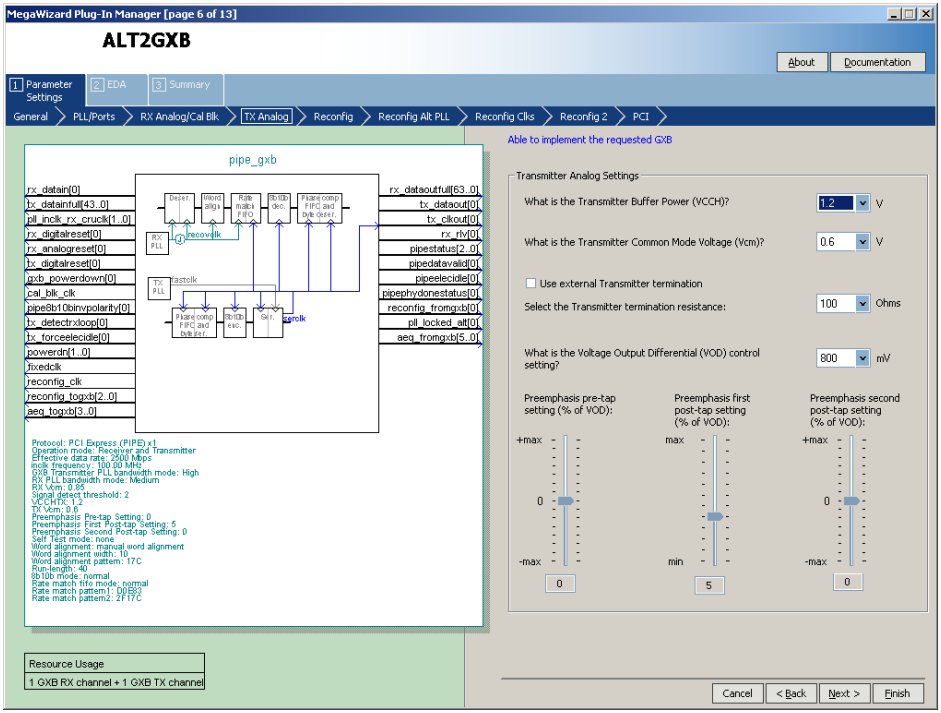


Table 4–14 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–14. MegaWizard Plug-In Manager Options (Page 6 for PIPE Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. In PIPE mode, this value is fixed at 1.2 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Transmitter Common Mode Voltage (V_{CM})?	In PIPE mode, the transmitter common mode voltage is fixed at 0.6 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap. The amount of pre-emphasis is to be determined by characterization.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–14. MegaWizard Plug-In Manager Options (Page 6 for PIPE Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap. The amount of pre-emphasis is to be determined by characterization.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap. The amount of pre-emphasis is to be determined by characterization.	

Figure 4–19 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for PIPE mode.

Figure 4–19. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

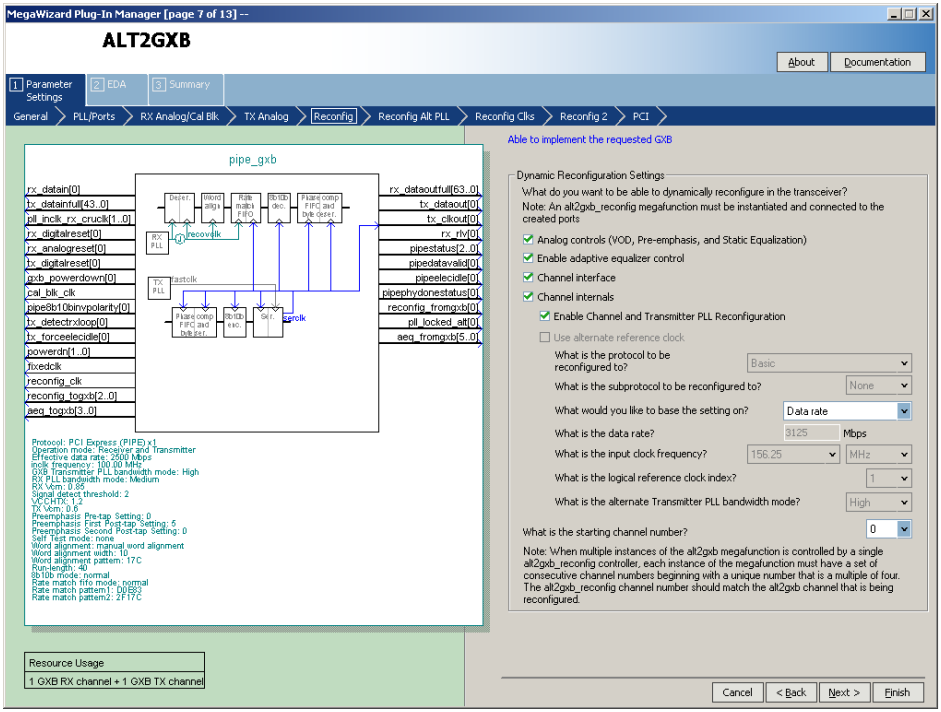


Table 4–15 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–15. MegaWizard Plug-In Manager Options (Page 7 for PIPE Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	In ×1 mode, the available options are Analog Controls, Enable adaptive equalizer control, Channel Internals, and Channel Interface. In ×4 and ×8 modes, only Analog Controls and Enable adaptive equalizer control are dynamically reconfigurable. • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel • Channel Interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals • Channel Internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but the same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–20 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for PCI Express (PIPE) mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–20. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

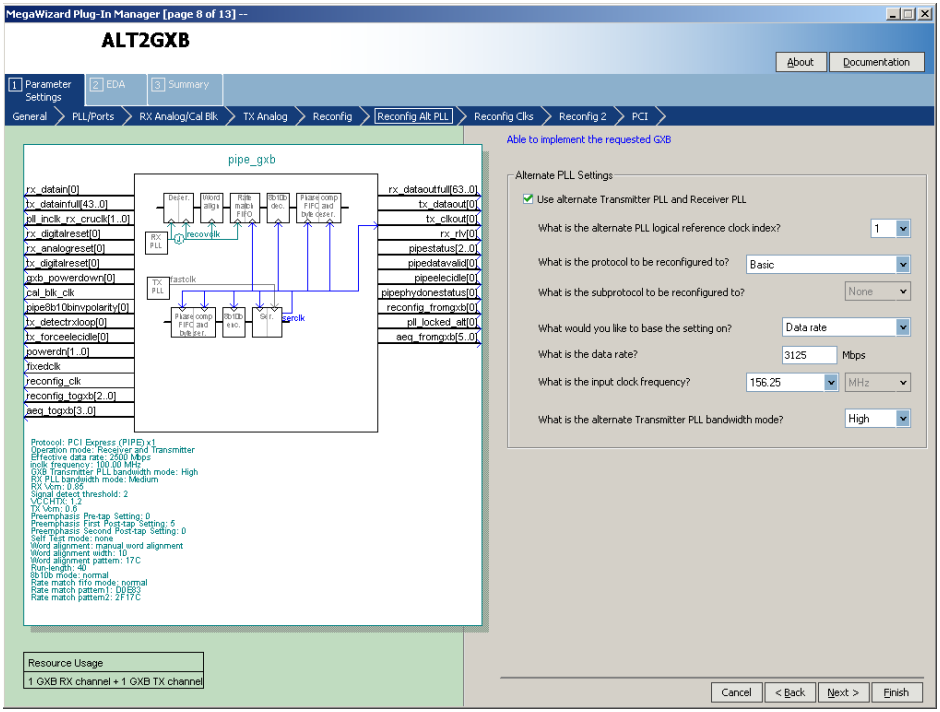


Table 4–16 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–16. MegaWizard Plug-In Manager Options (Page 8 for PIPE Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–21 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for PCI Express (PIPE) mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–21. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

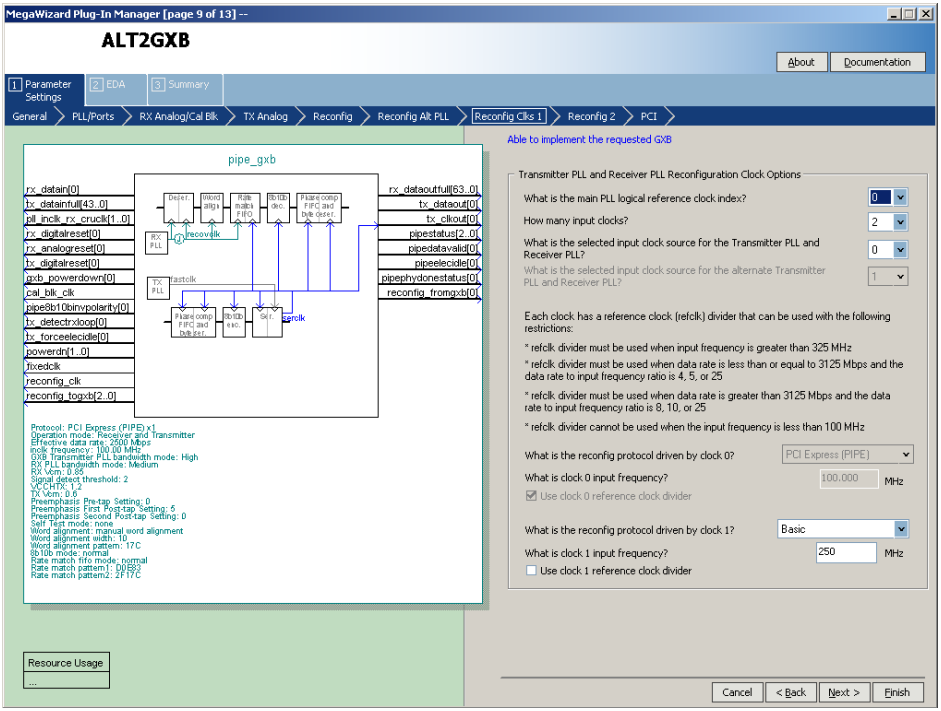


Table 4–17 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–17. MegaWizard Plug-In Manager Options (Page 9 for PIPE Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–22 shows page 10 of the MegaWizard Plug-In Manager for the PIPE protocol selection. This page appears only when the **Channel Internals** or **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–22. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

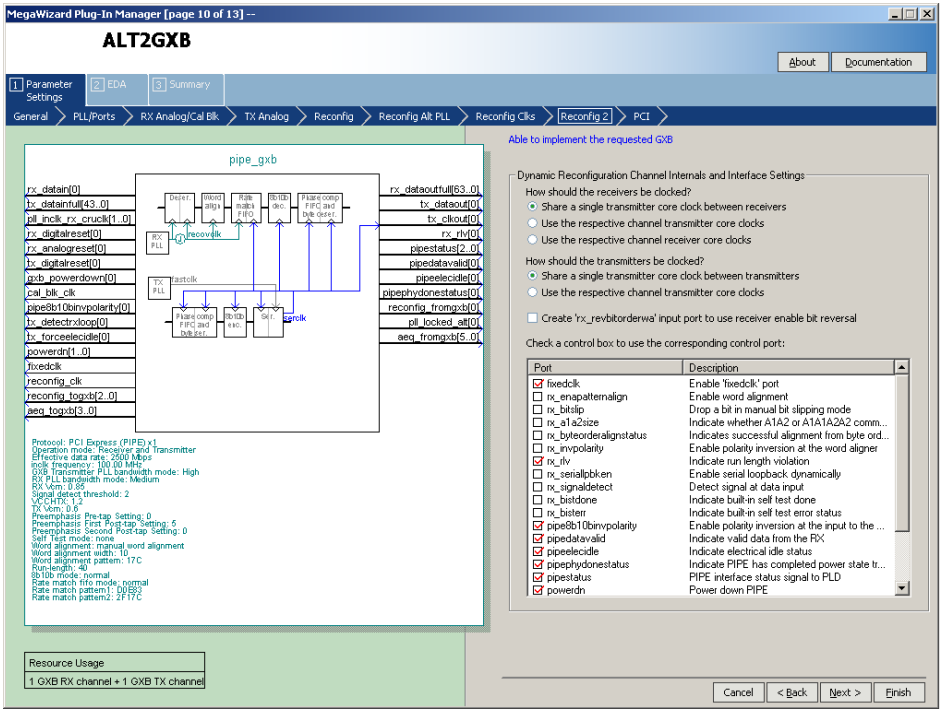


Table 4–18 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–18. MegaWizard Plug-In Manager Options (Page 10 for PIPE Mode)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section n the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–23 shows page 11 of the MegaWizard Plug-In Manager for the PIPE protocol selection. If the **Enforce default settings for this protocol** option is selected, this page does not appear in the MegaWizard.

Figure 4–23. MegaWizard Plug-In Manager - ALT2GXB (PCI)

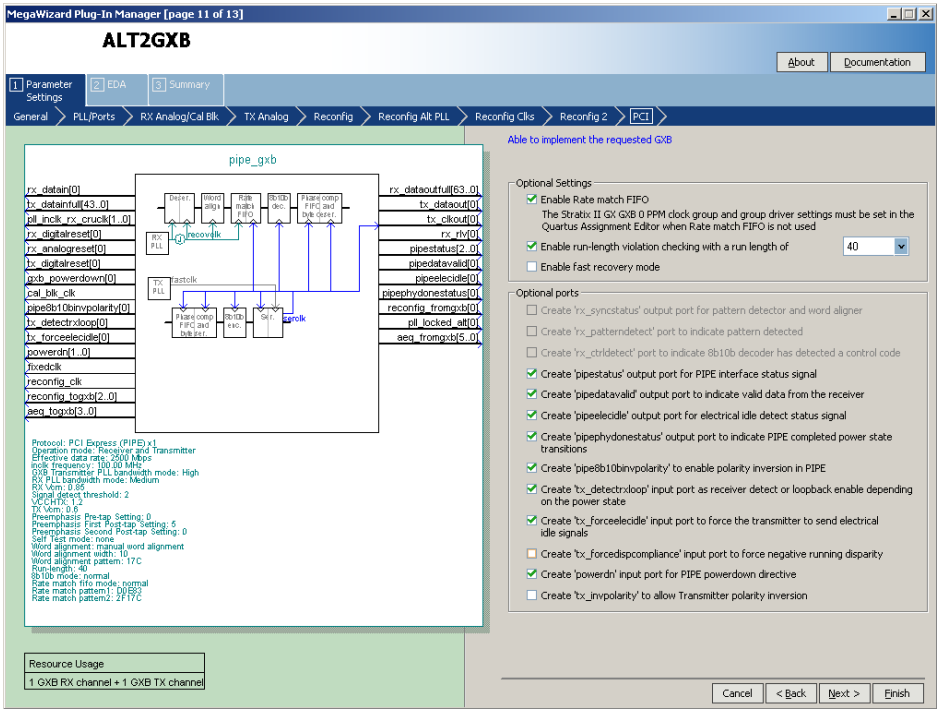


Table 4–19 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–19. MegaWizard Plug-In Manager Options (Page 11 for PIPE Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Enable Rate match FIFO	This option enables bypassing of the rate match FIFO in the receiver data path.	Low-latency PIPE mode n the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable run-length violation checking with a run length of	This option activates the run-length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable fast recovery mode	This option creates the NTFS fast recovery IP required to meet the PCI-E specification in the PLD logic array.	PCI Express (PIPE) Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_patterndetect` output port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_ctrldetect` output port to indicate 8B/10B decoder has detected a control code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–19. MegaWizard Plug-In Manager Options (Page 11 for PIPE Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pipestatus` output port for PIPE interface status signal	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `pipedatavalid` output port to indicate valid data from the receiver	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `pipeelecidle` output port for Electrical Idle detect status signal	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `pipephydonestatus` output port to indicate PIPE completed power state transitions	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `pipe8b/10binvpolarity` to enable polarity inversion in PIPE	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_detectrxloop` input port as receiver detect or loopback enable, depending on the power state	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_forceelecidle` input port to force the transmitter to send Electrical Idle signals	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–19. MegaWizard Plug-In Manager Options (Page 11 for PIPE Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `tx_forcedispcompliance` input port to force negative running disparity	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `powerdn` input port for PIPE powerdown directive	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	PIPE Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner. This feature must not be enabled when `pipe8b/10binvpolarity` is enabled.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–24 shows page 12 of the MegaWizard Plug-In Manager for the PIPE protocol selection. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for the third party EDA synthesis tool to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–24. MegaWizard Plug-In Manager - ALT2GXB (EDA)

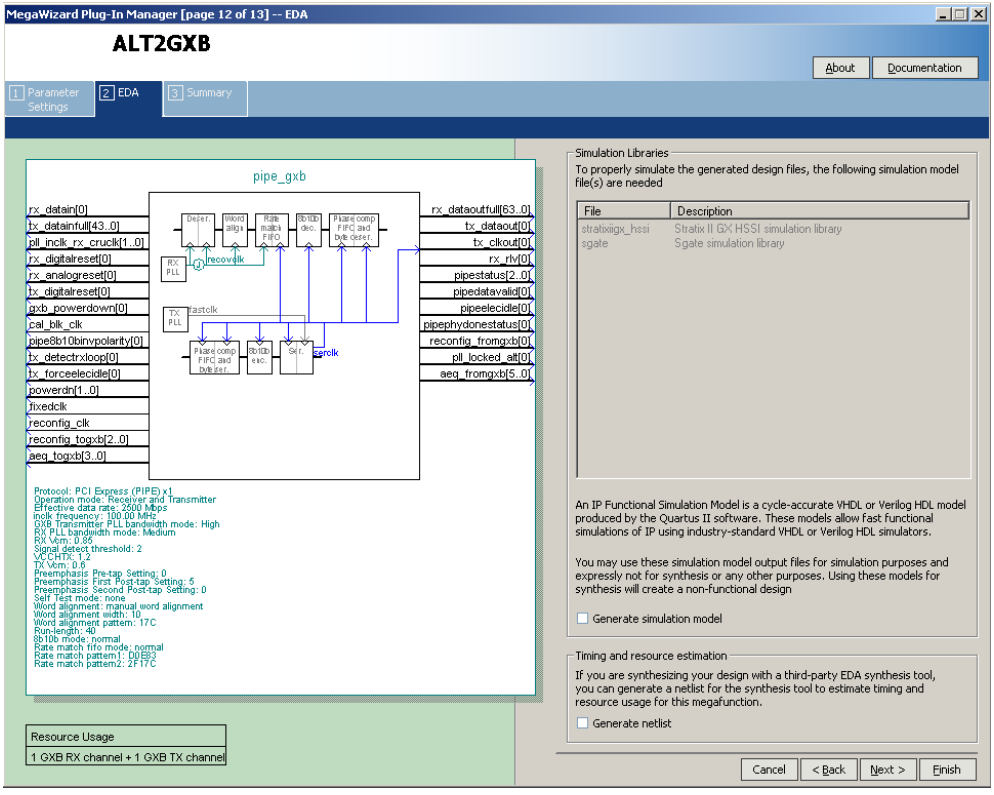
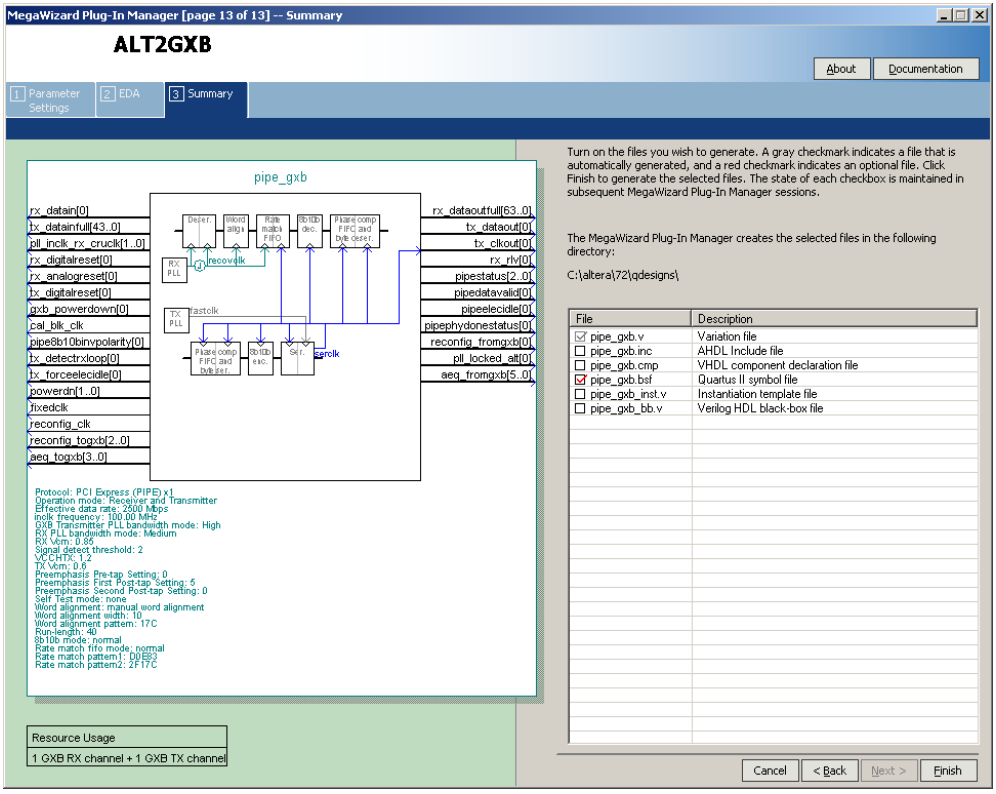


Figure 4–25 shows page 13 (the last page) of the MegaWizard Plug-In Manager for the PIPE protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–25. MegaWizard Plug-In Manager - ALT2GXB (Summary)



XAUI Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for the XAUI mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

The word aligner and rate matcher operations and patterns are pre-configured for the XAUI mode and cannot be altered.

Figure 4–26 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager for XAUI mode.

Figure 4–26. MegaWizard Plug-In Manager - ALT2GXB (General)

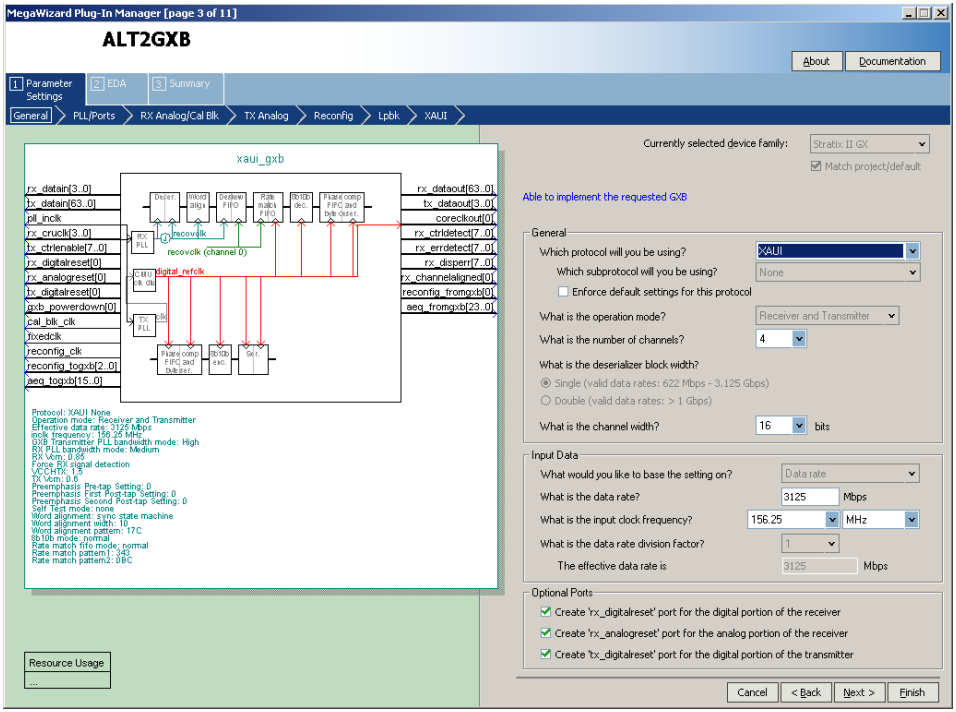


Table 4–20 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–20. MegaWizard Plug-In Manager Options (Page 3 for XAUI Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Selects the specific protocol or modes under which the transceiver operates. For the XAUI or HiGig, you must select the **XAUI** protocol.	
Which subprotocol will you be using?	Not applicable to XAUI mode.	

Table 4–20. MegaWizard Plug-In Manager Options (Page 3 for XAUI Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Enforce default settings for this protocol	Selecting this option skips the **XAUI** page of the XAUI MegaWizard Plug-In Manager. The **XAUI** page allows you to select the XAUI-specific ports for your design. If you select this option, all XAUI-specific ports are used.	
What is the operation mode?	Only receiver and transmitter (full duplex) is allowed in the XAUI protocol. Receiver only and transmitter only modes are not allowed.	
What is the number of channels?	This selects how many duplicate channels this ALT2GXB instance contains. In XAUI mode, the number of channels increments by 4.	
What is the deserializer block width?	XAUI mode only operates in a single-width mode. Double-width mode is not allowed.	
What is the channel width?	This option determines the transceiver to PLD interface width. Only 16-bit channel width is allowed in XAUI mode.	Byte Serializer and Byte Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option is not available.	
What is the data rate? *(1)*	Enter a data rate from 3.125 Gbps to 3.75 Gbps.	
What is the input clock frequency?	Determines the input reference clock frequency for the transceiver. The Quartus II software automatically selects the input reference clock frequency based on the entered data rate.	
What is the data rate division factor?	This option is not available.	
Create `rx_digitalreset` port for the digital portion of the receiver	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_analogreset` port for the analog portion of the receiver	Receiver analog reset port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–20. MegaWizard Plug-In Manager Options (Page 3 for XAUI Mode) (Part 3 of 3)		
ALT2GXB Setting	**Description**	**Reference**
Create `tx_digitalreset` port for the digital portion of the transmitter	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Note to Table 4–20:

(1) A data rate higher than 3.125 Gbps requires a -3 speed grade device. The higher speed also requires a recompile of the design for different device settings.

Figure 4–27 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for XAUI mode.

Figure 4–27. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

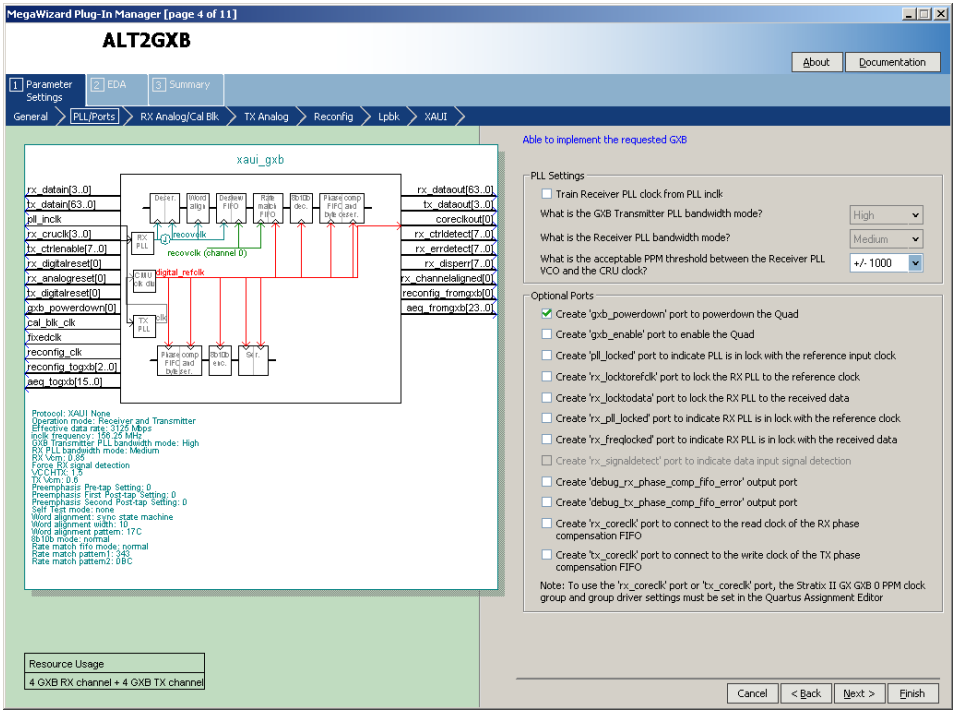


Table 4–21 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–21. MegaWizard Plug-In Manager Options (Page 4 for XAUI Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the GXB Transmitter PLL bandwidth mode?	In XAUI mode, only high bandwidth is supported for the transmitter PLL.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Receiver PLL bandwidth mode?	In XAUI mode, only medium bandwidth is supported for the receiver PLL and VCO.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver CRU switchover between lock-to-data and lock-to-reference. (There are additional factors that affect CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–21. MegaWizard Plug-In Manager Options (Page 4 for XAUI Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicate data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–21. MegaWizard Plug-In Manager Options (Page 4 for XAUI Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–28 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for XAUI mode.

Figure 4–28. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

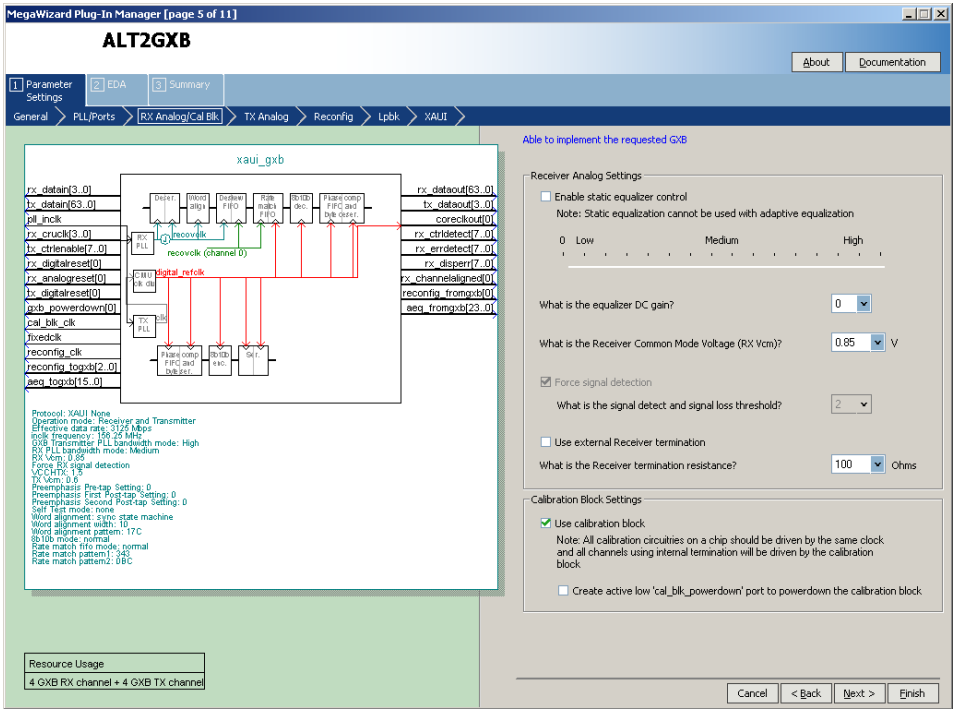


Table 4–22 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–22. MegaWizard Plug-In Manager Options (Page 5 for XAUI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is set to 0.85 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Force signal detection	This option is available only in PIPE mode.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	This option is available only in PIPE mode.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–22. MegaWizard Plug-In Manager Options (Page 5 for XAUI Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–29 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for XAUI mode.

Figure 4–29. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

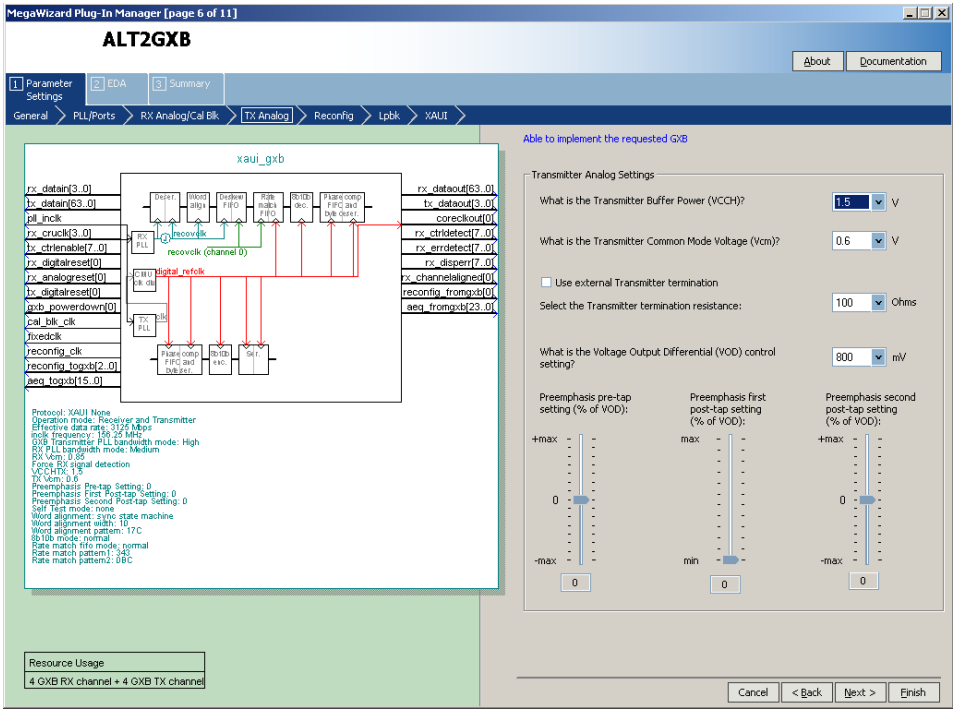


Table 4–23 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–23. MegaWizard Plug-In Manager Options (Page 6 for XAUI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. The selections available are 1.2 V and 1.5 V	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Restrictions apply based on the V_{CCH} setting.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap. The amount of pre-emphasis is to be determined by characterization.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–23. MegaWizard Plug-In Manager Options (Page 6 for XAUI Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap. The amount of pre-emphasis is to be determined by characterization.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap. The amount of pre-emphasis is to be determined by characterization.	

Figure 4–30 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for XAUI mode.

Figure 4–30. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

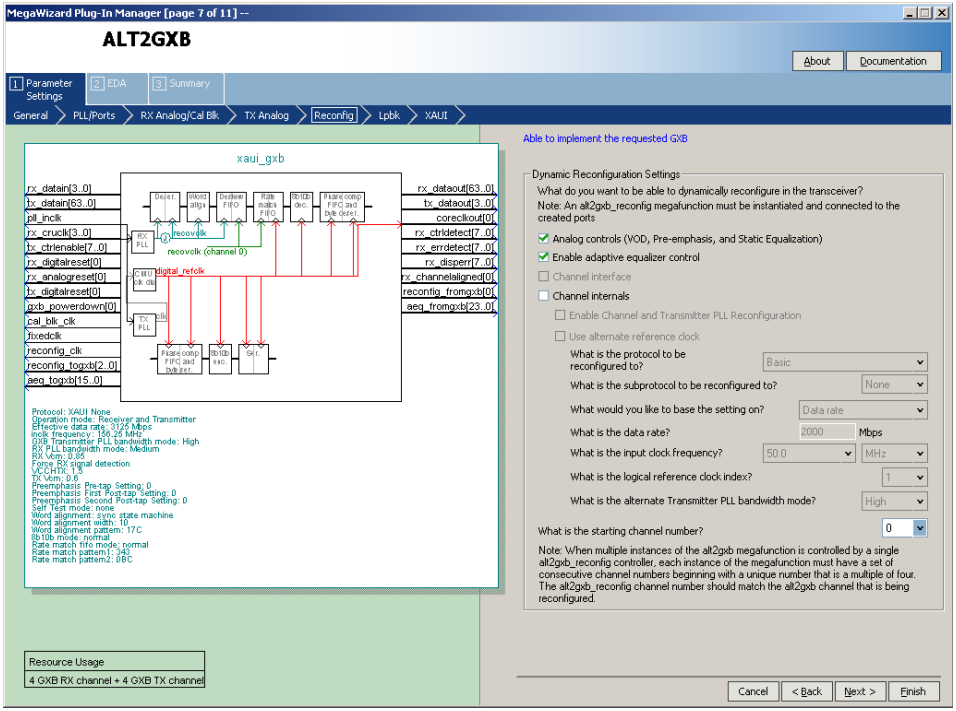


Table 4–24 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–24. MegaWizard Plug-In Manager Options (Page 7 for XAUI Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are Analog Controls, Enable adaptive equalizer control, and Channel internals. • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel Internals: Enables MIF-based reconfiguration to a XAUI mode with different internal parameters. Note that the **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock** options are not available in XAUI mode.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–31 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for XAUI mode.

Figure 4–31. MegaWizard Plug-In Manager - ALT2GXB (Loopback)

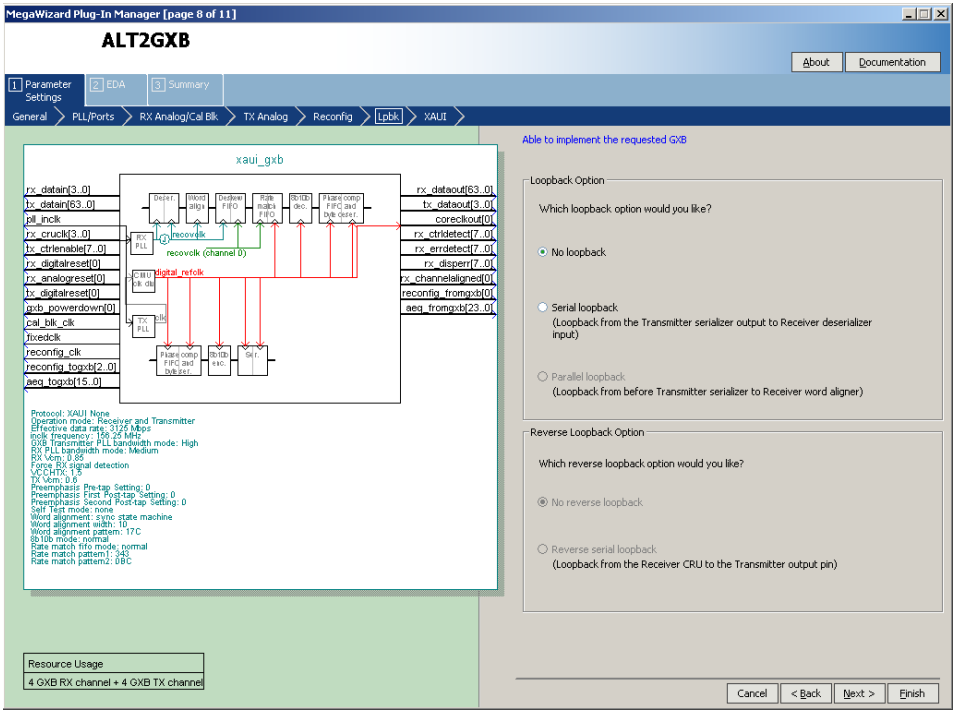


Table 4–25 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–25. MegaWizard Plug-In Manager Options (Page 8 for XAUI Mode)

ALT2GXB Setting	Description	Reference
Which loopback option would you like?	There are two option available in XAUI mode: no loopback and serial loopback. • No loopback - this is the default mode. • Serial loopback - if you select serial loopback, the `rx_seriallpbken` port is available to control the serial loopback feature dynamically. A 1'b1 enables serial loopback and a 1'b0 disables loopback on a channel-by-channel basis. Altera recommends controlling all four channels simultaneously. A digital reset must be asserted for the transceiver.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Reverse Loopback option	This option is not available in XAUI mode.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–32 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for XAUI mode. If the **Enforce default settings for this protocol** option is selected, this page does not appear in the MegaWizard.

Figure 4–32. MegaWizard Plug-In Manager - ALT2GXB (XAUI)

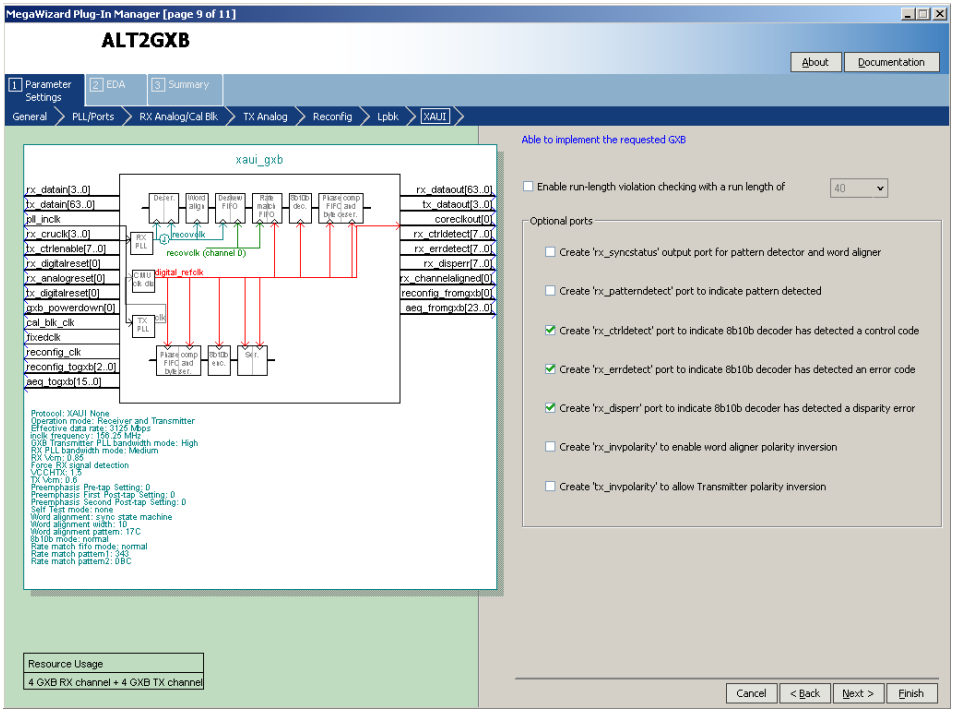


Table 4–26 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–26. MegaWizard Plug-In Manager Options (Page 9 for XAUI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable run-length violation checking with a run length of	This option activates the run-length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_patterndetect` port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_ctrldetect` port to indicate 8B/10B decoder has detected a control code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_errdetect` port to indicate 8B/10B decoder has detected an error code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_disperr` port to indicate 8B/10B decoder has detected a disparity error	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–26. MegaWizard Plug-In Manager Options (Page 9 for XAUI Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–33 shows page 10 of the MegaWizard Plug-In Manager for the XAUI protocol selection. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for the third party EDA synthesis tool to be able to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–33. MegaWizard Plug-In Manager - ALT2GXB (EDA)

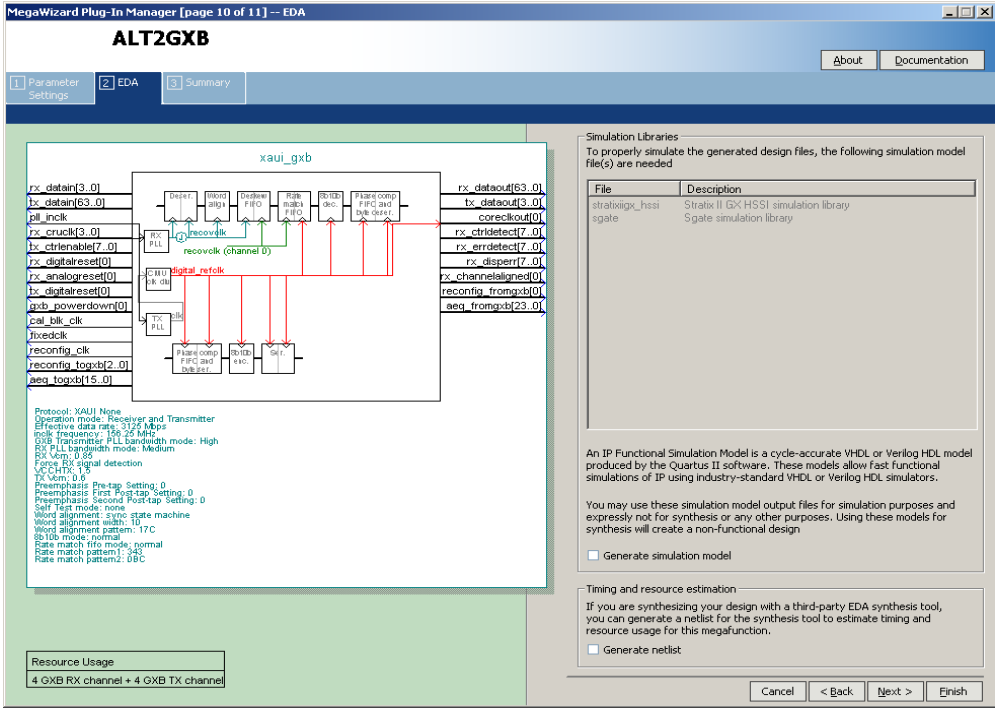
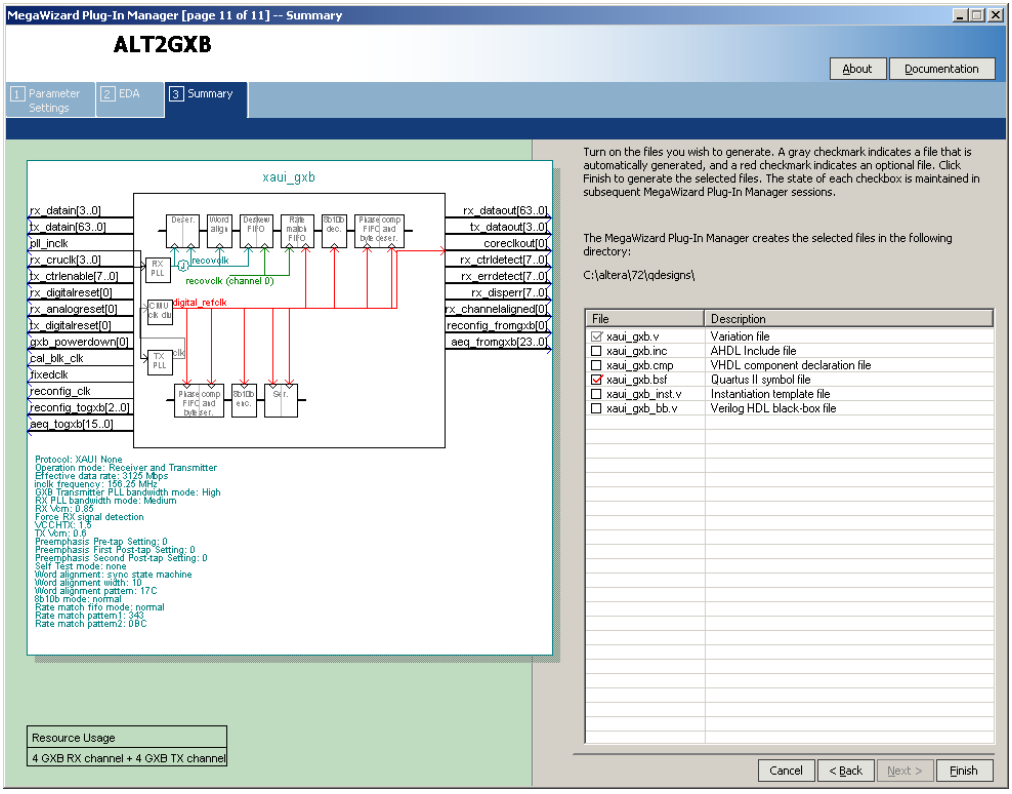


Figure 4–34 shows page 11 (the last page) of the MegaWizard Plug-In Manager for the XAUI protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–34. MegaWizard Plug-In Manager - ALT2GXB (Summary)



GIGE Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for the GIGE mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

The word aligner and rate matcher operations and patterns are pre-configured for the GIGE mode and cannot be altered.

Figure 4–35 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode.

Figure 4–35. MegaWizard Plug-In Manager - ALT2GXB (General)

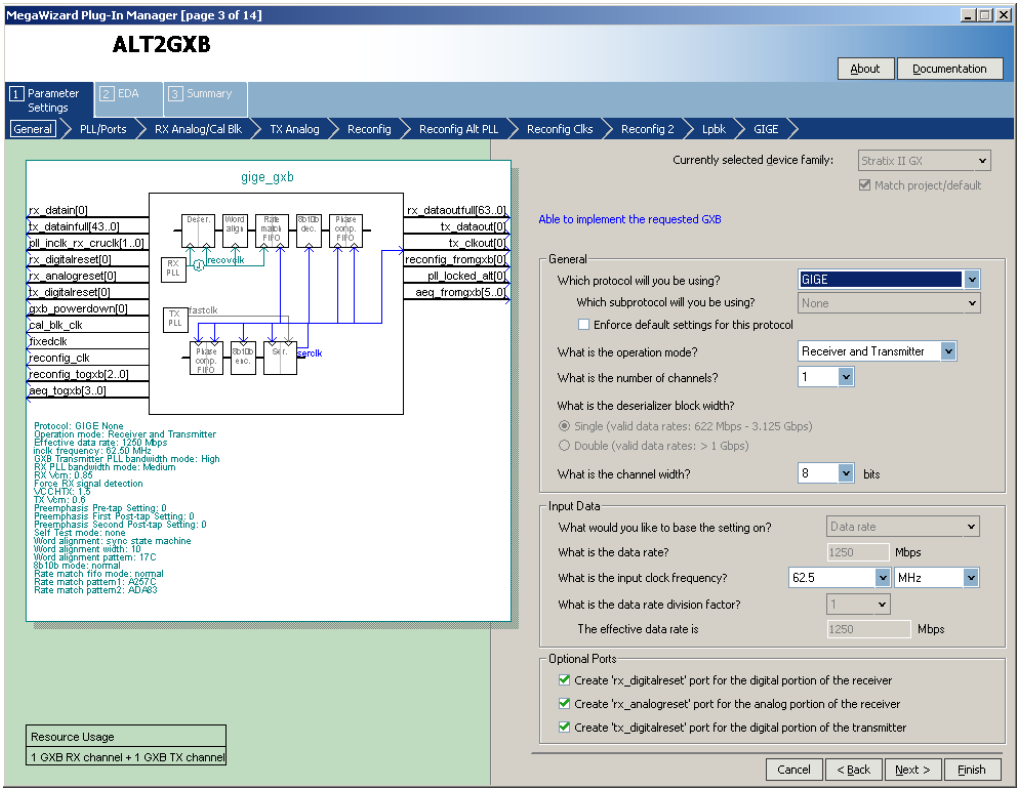


Table 4–27 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–27. MegaWizard Plug-In Manager Options (Page 3 for GIGE Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Selects the specific protocol or modes under which the transceiver operates. For the GIGE mode, you must select the **GIGE** protocol.	
Which subprotocol will you be using?	Not applicable to GIGE mode.	

Table 4–27. MegaWizard Plug-In Manager Options (Page 3 for GIGE Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Enforce default settings for this protocol	Selecting this option skips the **GIGE** page of the GIGE MegaWizard Plug-In Manager. The **GIGE** page allows you to select the GIGE-specific ports for your design. If you select this option, all GIGE-specific ports are used.	
What is the operation mode?	The transmitter only and receiver and transmitter (full duplex) modes are allowed in GIGE protocol. The receiver only mode is not available.	
What is the number of channels?	This selects how many duplicate channels this ALT2GXB instance contains. In GIGE mode, the number of channels increments by 1.	
What is the deserializer block width?	GIGE mode only operates in a single-width mode. Double-width mode is not allowed.	
What is the channel width?	This option determines the transceiver to PLD interface width. In GIGE mode, 8 bits are allowed.	Byte Serializer and Byte Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option is not used because the data rate is fixed at 1.25 Gbps for GIGE mode.	
What is the data rate?	This option is not used because the data rate is fixed at 1.25 Gbps for GIGE mode.	
What is the input clock frequency?	Determines the input clock frequency or period you want as a reference clock for the transceiver. In GIGE mode, only 62.5 MHz and 125 MHz are allowed.	
What is the data rate division factor?	This option is not used because the data rate is fixed at 1.25 Gbps for GIGE mode.	
Create `rx_digitalreset` port for the digital portion of the receiver	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–27. MegaWizard Plug-In Manager Options (Page 3 for GIGE Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_analogreset` port for the analog portion of the receiver	Receiver analog reset port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_digitalreset` port for the digital portion of the receiver	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–36 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode.

Figure 4–36. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

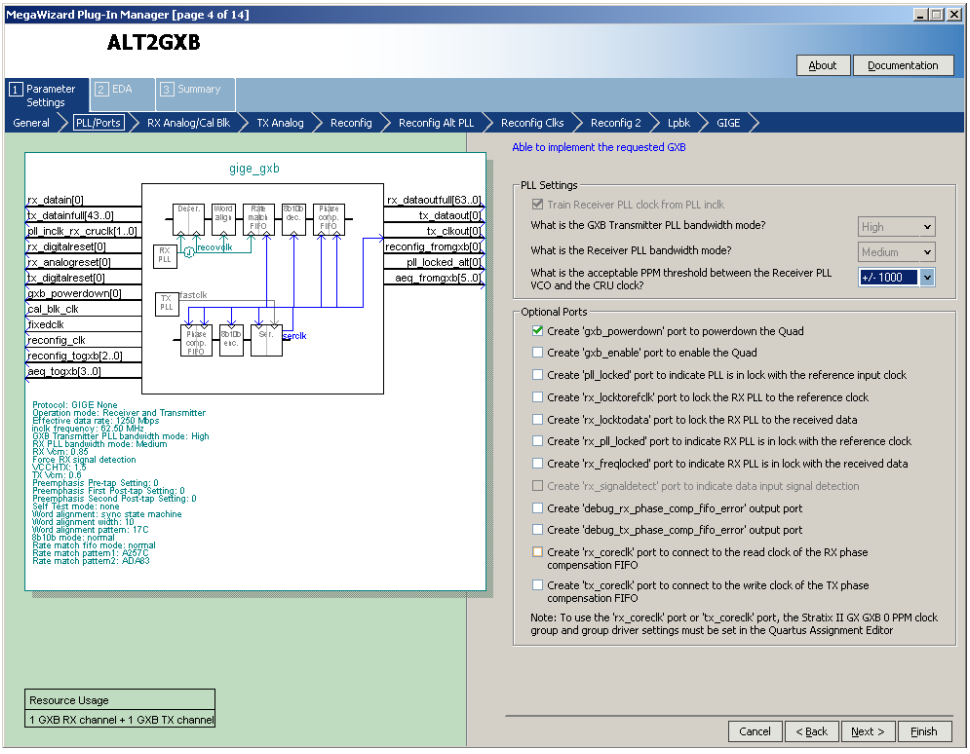


Table 4–28 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–28. MegaWizard Plug-In Manager Options (Page 4 for GIGE Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If **CMU PLL reconfiguration** is enabled, this option is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	In GIGE mode, only high bandwidth is supported for the transmitter PLL.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver PLL bandwidth mode?	In GIGE mode, only medium bandwidth is supported for the receiver PLL and V_{CO}.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver CRU switchover between lock-to-data and lock-to-reference. (There are additional factors that affect CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–28. MegaWizard Plug-In Manager Options (Page 4 for GIGE Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicate data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–28. MegaWizard Plug-In Manager Options (Page 4 for GIGE Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–37 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode.

Figure 4–37. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

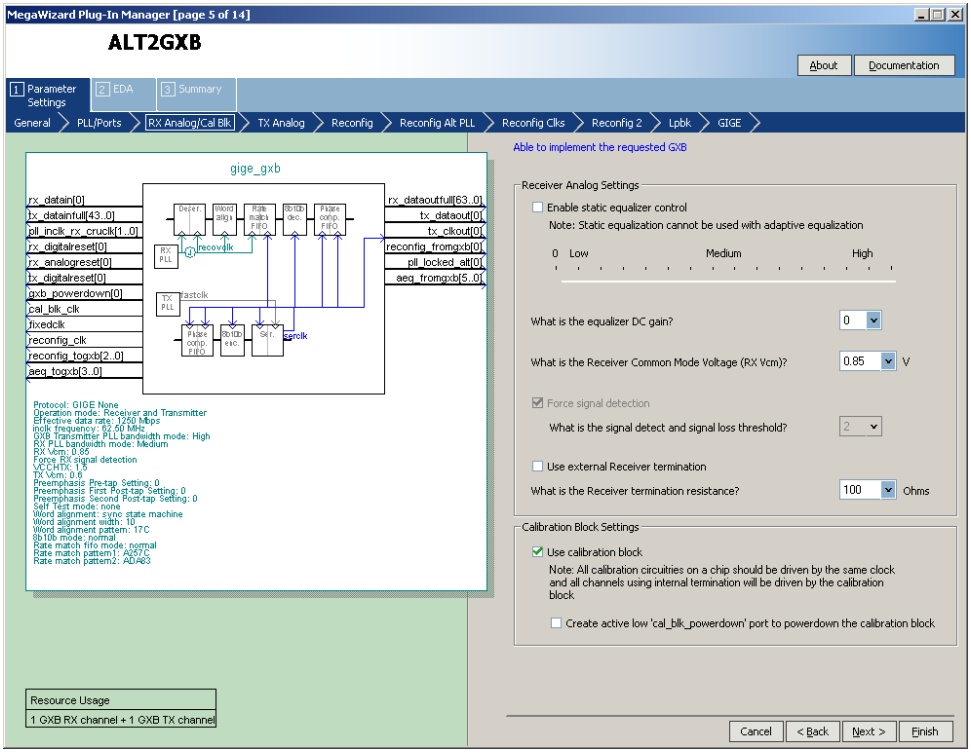


Table 4–29 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–29. MegaWizard Plug-In Manager Options (Page 5 for GIGE Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is set to 0.85 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Forced signal detection	This option disables the signal detect circuit. This removes the signal detect criterion for the receiver CRU lock-to-reference and lock-to-data switchover.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	Use this option when the forced signal detection option is off and to set the trip point of the signal detect circuit. The levels are to be determined after characterization.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–29. MegaWizard Plug-In Manager Options (Page 5 for GIGE Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–38 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode.

Figure 4–38. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

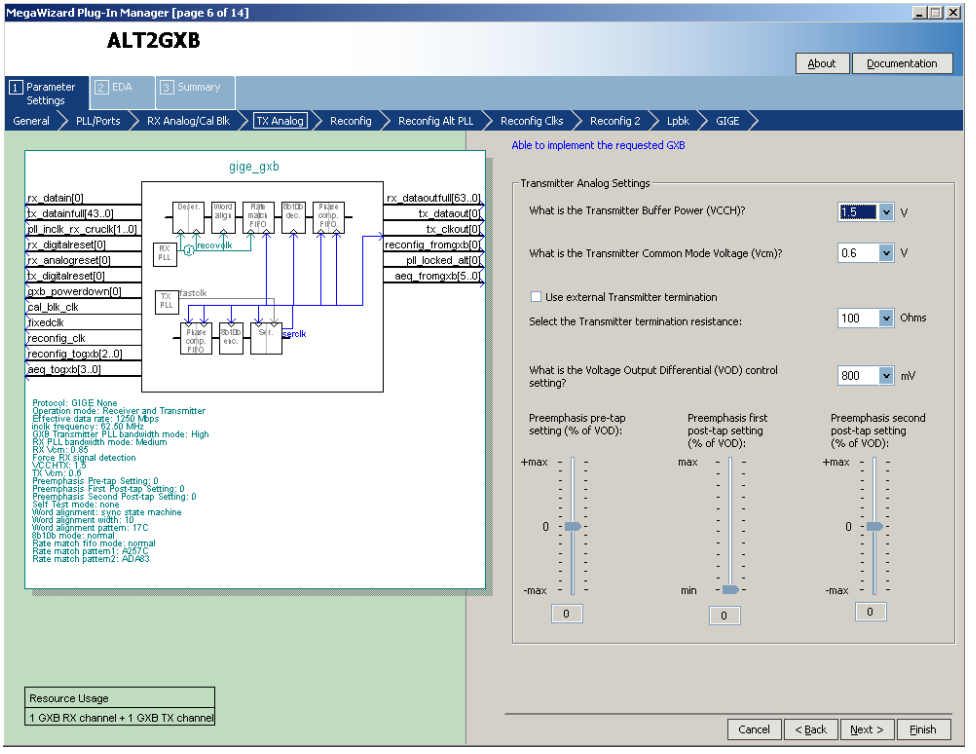


Table 4–30 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–30. MegaWizard Plug-In Manager Options (Page 6 for GIGE Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. The selections available are 1.2 V and 1.5 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Restrictions apply based on the V_{CCH} setting.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap. The amount of pre-emphasis is to be determined by characterization.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–30. MegaWizard Plug-In Manager Options (Page 6 for GIGE Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap. The amount of pre-emphasis is to be determined by characterization.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap. The amount of pre-emphasis is to be determined by characterization.	

Figure 4–39 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode.

Figure 4–39. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

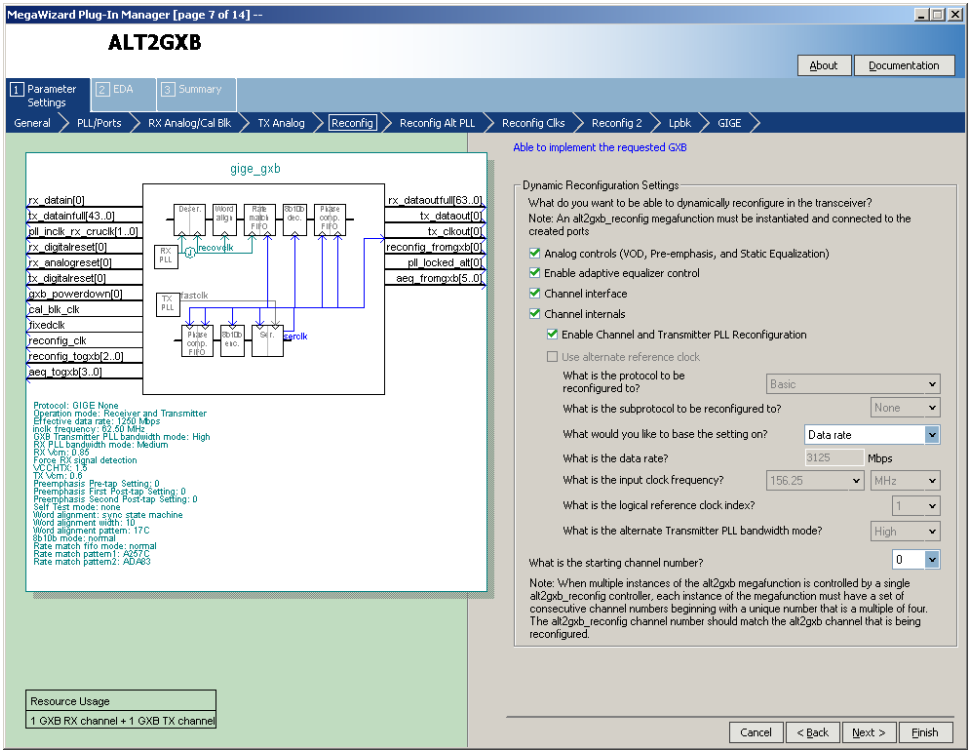


Table 4–31 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–31. MegaWizard Plug-In Manager Options (Page 7 for GIGE Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are: • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel Interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals. • Channel Internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–40 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode. This page appears only when the **Channel Internals** or **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–40. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

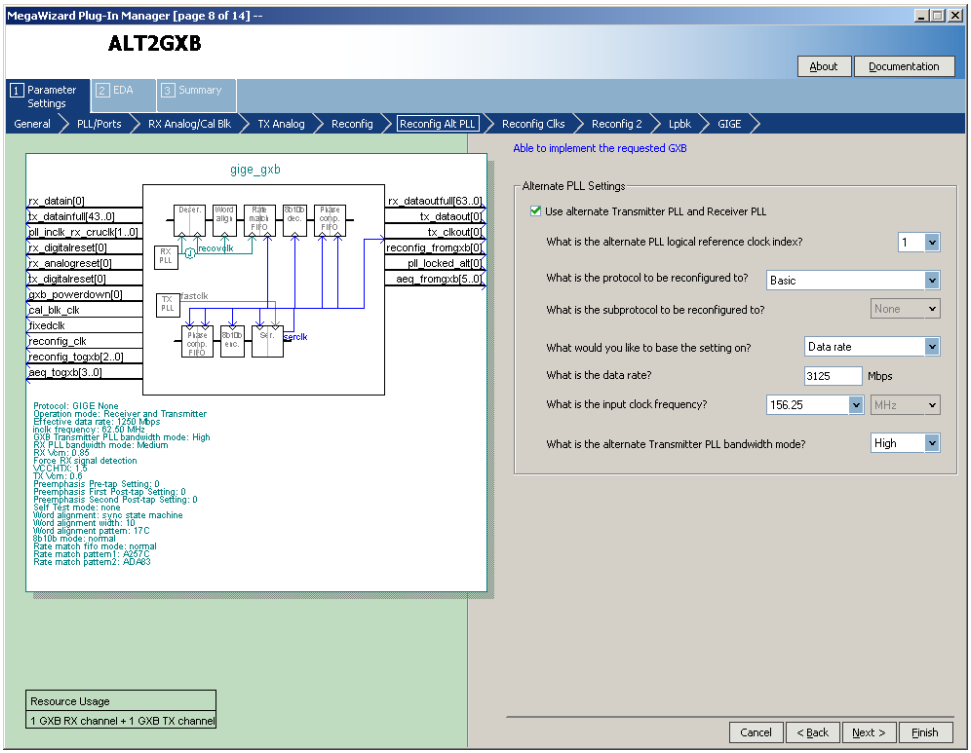


Table 4–32 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–32. MegaWizard Plug-In Manager Options (Page 8 for GIGE Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–41 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected on the **Reconfig** page (Page 7).

Figure 4–41. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

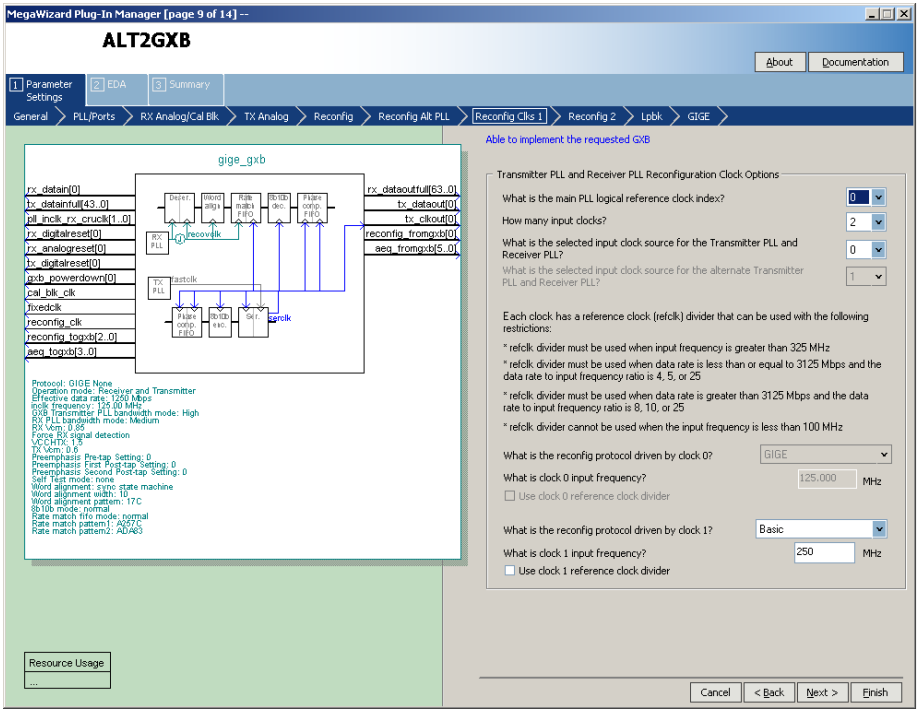


Table 4–33 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–33. MegaWizard Plug-In Manager Options (Page 9 for GIGE Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–42 shows page 10 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode. This page appears only when the **Channel Internals** and **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–42. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

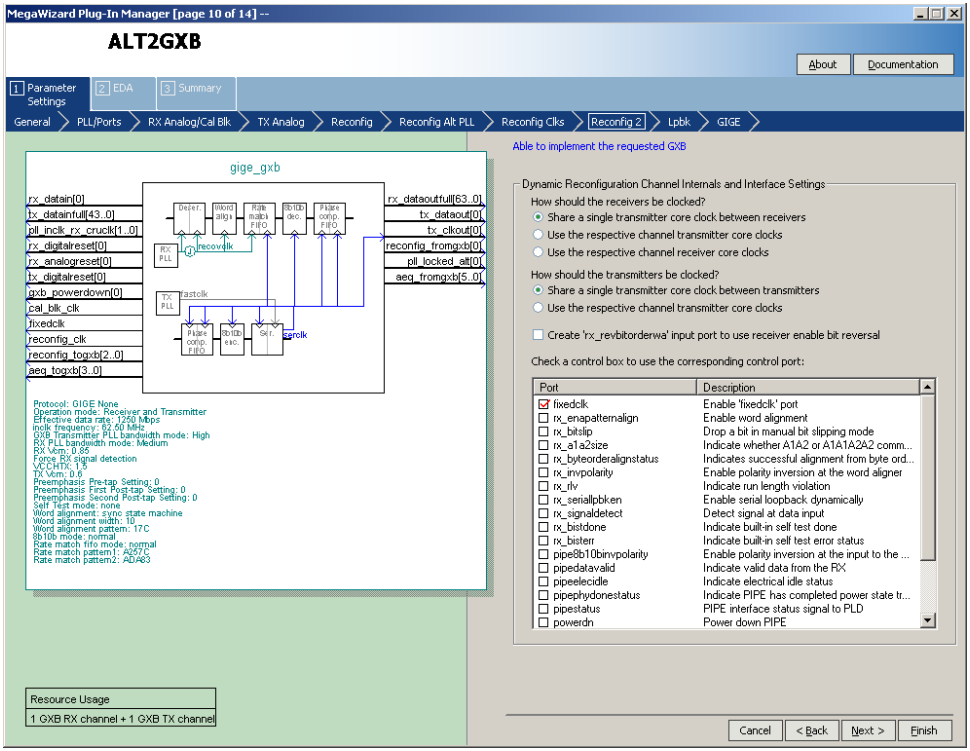


Table 4–34 describes the available options on page 10 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–34. MegaWizard Plug-In Manager Options (Page 10 for GIGE Mode)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–43 shows page 11 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode.

Figure 4–43. MegaWizard Plug-In Manager - ALT2GXB (Loopback)

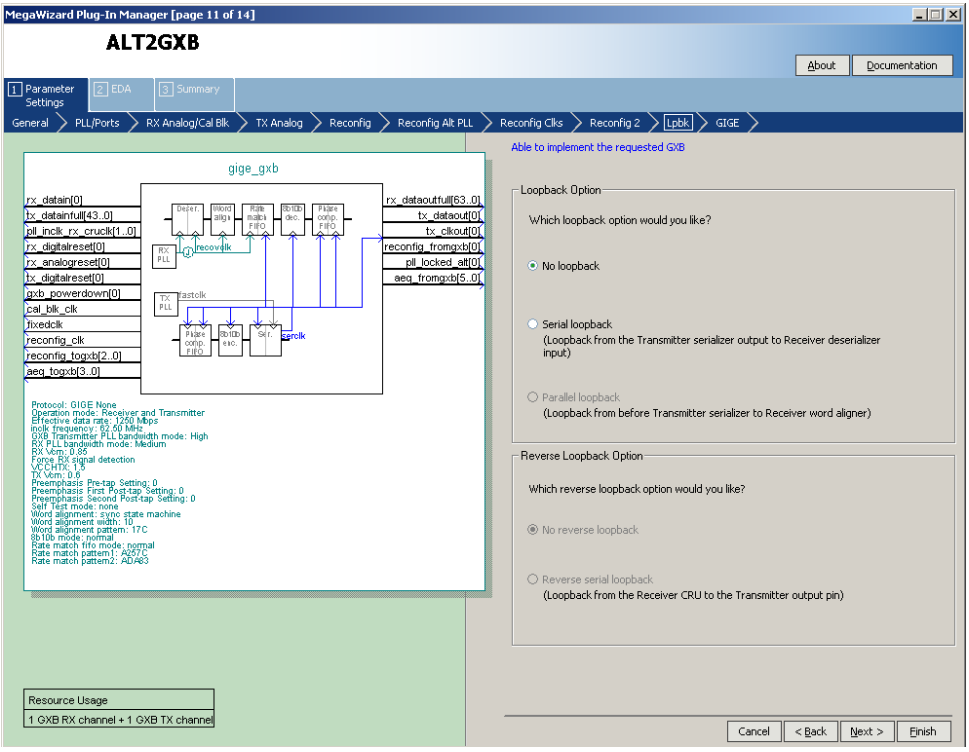


Table 4–35 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–35. MegaWizard Plug-In Manager Options (Page 11 for GIGE Mode)

ALT2GXB Setting	Description	Reference
Which loopback option would you like?	There are two options in GIGE mode: no loopback and serial loopback. • No loopback - this is the default mode. • Serial loopback - if you select serial loopback, the `rx_seriallpbken` port is available to control the serial loopback feature dynamically. A 1'b1 enables serial loopback and a 1'b0 disables loopback on a channel-by-channel basis. Altera recommends controlling all four channels simultaneously. A digital reset must be asserted for the transceiver.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Reverse Loopback option	This option is not available in GIGE mode.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–44 shows page 12 of the ALT2GXB MegaWizard Plug-In Manager for GIGE mode. If the **Enforce default settings for this protocol** option is selected, this page does not appear in the MegaWizard.

Figure 4–44. MegaWizard Plug-In Manager - ALT2GXB (GIGE)

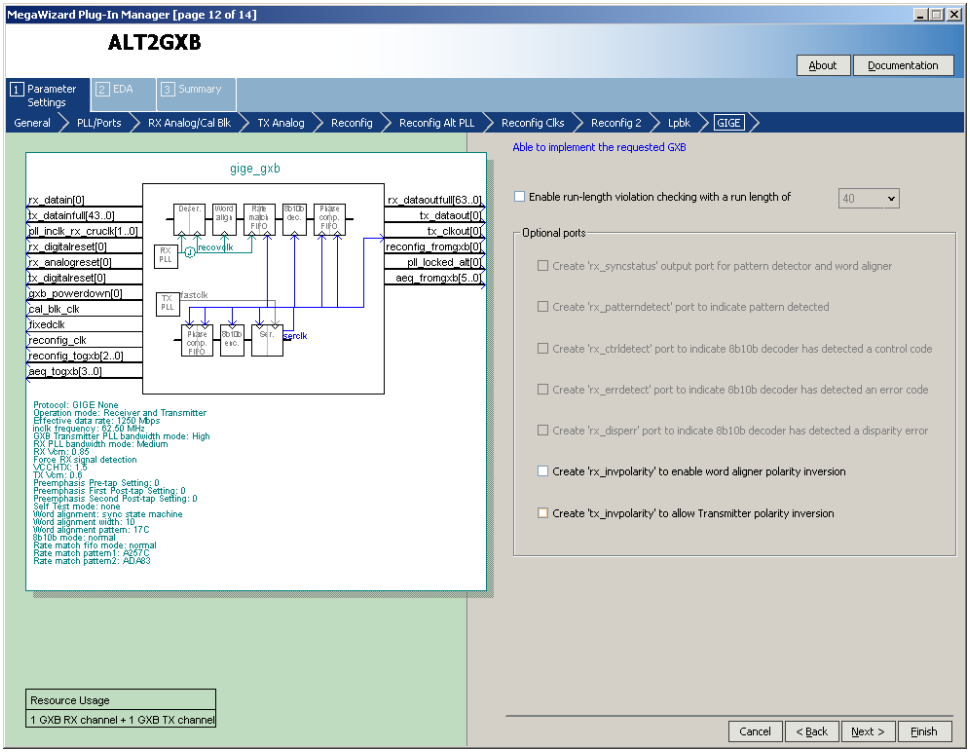


Table 4–36 describes the available options on page 12 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–36. MegaWizard Plug-In Manager Options (Page 12 for GIGE Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable run-length violation checking with a run length of	This option activates the run-length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_patterndetect` port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_ctrldetect` port to indicate 8B/10B decoder has detected a control code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_errdetect` port to indicate 8B/10B decoder has detected an error code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_disperr` port to indicate 8B/10B decoder has detected a disparity error	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–36. MegaWizard Plug-In Manager Options (Page 12 for GIGE Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–45 shows page 13 of the MegaWizard Plug-In Manager for the GIGE protocol selection. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for third party EDA synthesis tool to be able to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–45. MegaWizard Plug-In Manager - ALT2GXB (EDA)

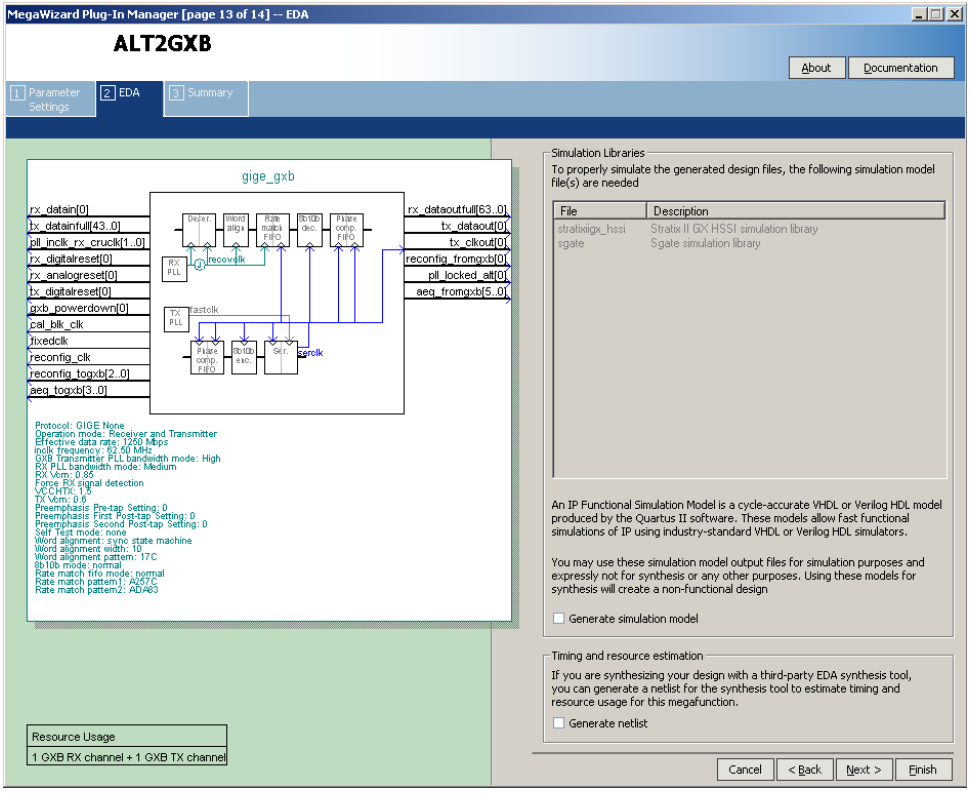
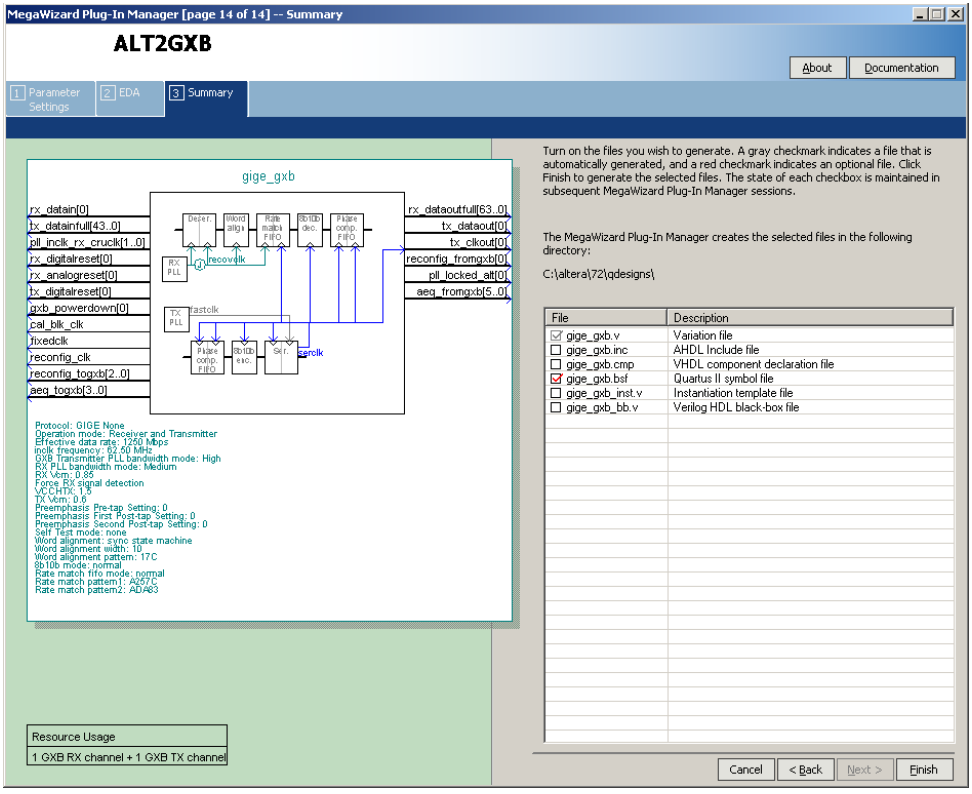


Figure 4–46 shows page 13 (the last page) of the MegaWizard Plug-In Manager for the GIGE protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–46. MegaWizard Plug-In Manager - ALT2GXB (Summary)



SONET/SDH Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for the SONET/SDH mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

Figure 4–47 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode.

Figure 4–47. MegaWizard Plug-In Manager - ALT2GXB (General)

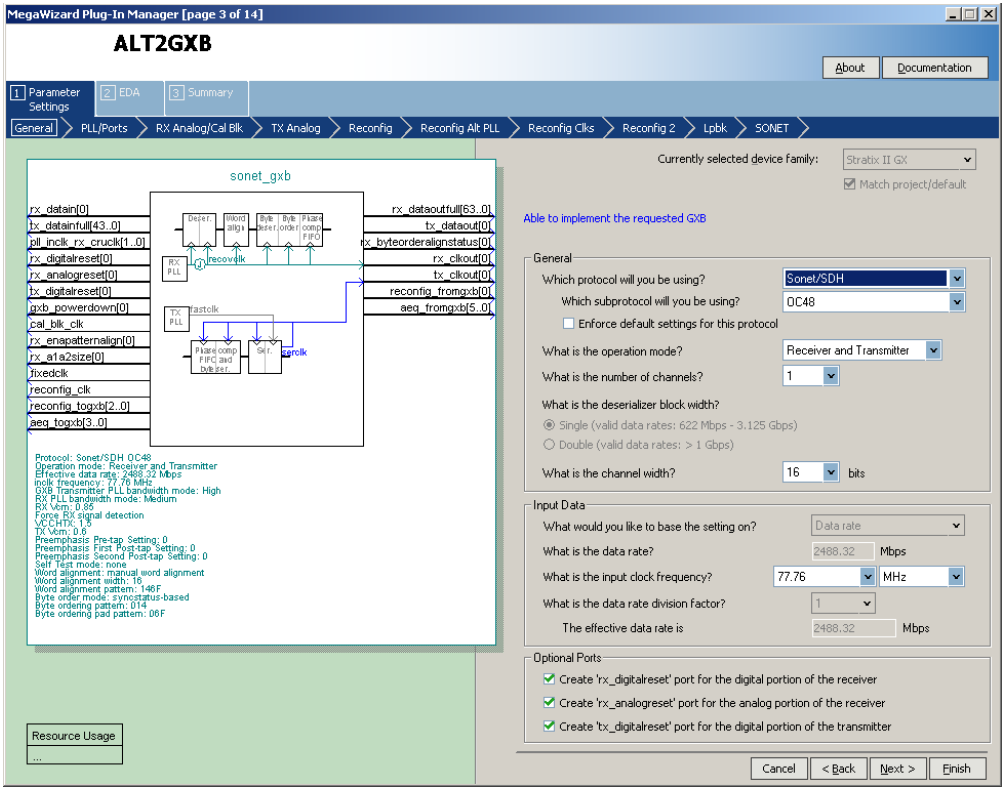


Table 4–37 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–37. MegaWizard Plug-In Manager Options (Page 3 for SONET/SDH Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Selects the specific protocol or modes that the transceiver operates under. For the SONET/SDH mode, you must select the **SONET/SDH** protocol.	
Which subprotocol will you be using?	There are three subprotocols allowed in SONET/SDH mode: OC-48, OC-12, and OC-96. Supported data rates are as follows: • OC-48 – 2488.32 Mbps • OC-12 – 622 Mbps • OC-96 – 4976 Mbps	
Enforce default settings for this protocol	Selecting this option skips the **SONET** page of the SONET/SDH MegaWizard Plug-In Manager. The **SONET** page allows you to select which SONET/SDH-specific port and word alignment pattern you want to use. If you select this option, all SONET/SDH-specific ports are used and the defaulted alignment pattern is locked at 16'hF628.	
What is the operation mode?	The transmitter only, receiver only, and receiver and transmitter (full duplex) modes are allowed in the SONET/SDH protocol.	
What is the number of channels?	This selects how many duplicate channels this ALT2GXB instance contains. In SONET/SDH mode, the number of channels increments by 1.	
What is the deserializer block width?	This option sets the transceiver data path width. • Single width – Selected automatically in OC-12 and OC-48 configurations. The transceiver data path width is 8 bits. • Double width – Selected automatically in OC-96 configurations. The transceiver data path width is 16 bits.	
What is the channel width?	This option selects the transceiver to PLD interface width. Depending on the subprotocol selection, choose one of the following: • 8 bits for OC-12 • 16 bits for OC-48 • 32 bits for OC-96	Byte Serializer and Byte Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option is not used because the data rate is fixed at: • 622 Mbps for OC-12 • 2488.32 Mbps for OC-48 • 4976 Mbps for OC-96	

Table 4–37. MegaWizard Plug-In Manager Options (Page 3 for SONET/SDH Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the data rate?	This option is not used because the data rate is fixed at: ● 622 Mbps for OC-12 ● 2488.32 Mbps for OC-48 ● 4976 Mbps for OC-96	
What is the input clock frequency?	Indicates the input reference clock frequencies for the transceiver. ● OC-48 – 77.76 MHz, 155.52 MHz, 311.04 MHz, and 622.08 MHz are allowed. ● OC-12 – 62.2 MHz, 77.76 MHz, 155.52 MHz, 311 MHz, and 622 MHz are allowed ● OC-96 – 124.4 MHz is allowed	
What is the data rate division factor?	This option is not used because the data rate is fixed at: ● 622 Mbps for OC-12 ● 2488.32 Mbps for OC-48 ● 4976 Mbps for OC-96	
Create `rx_digitalreset` port for the digital portion of the receiver	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_analogreset` port for the analog portion of the receiver	Receiver analog reset port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_digitalreset` port for the digital portion of the transmitter	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–48 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode.

Figure 4–48. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

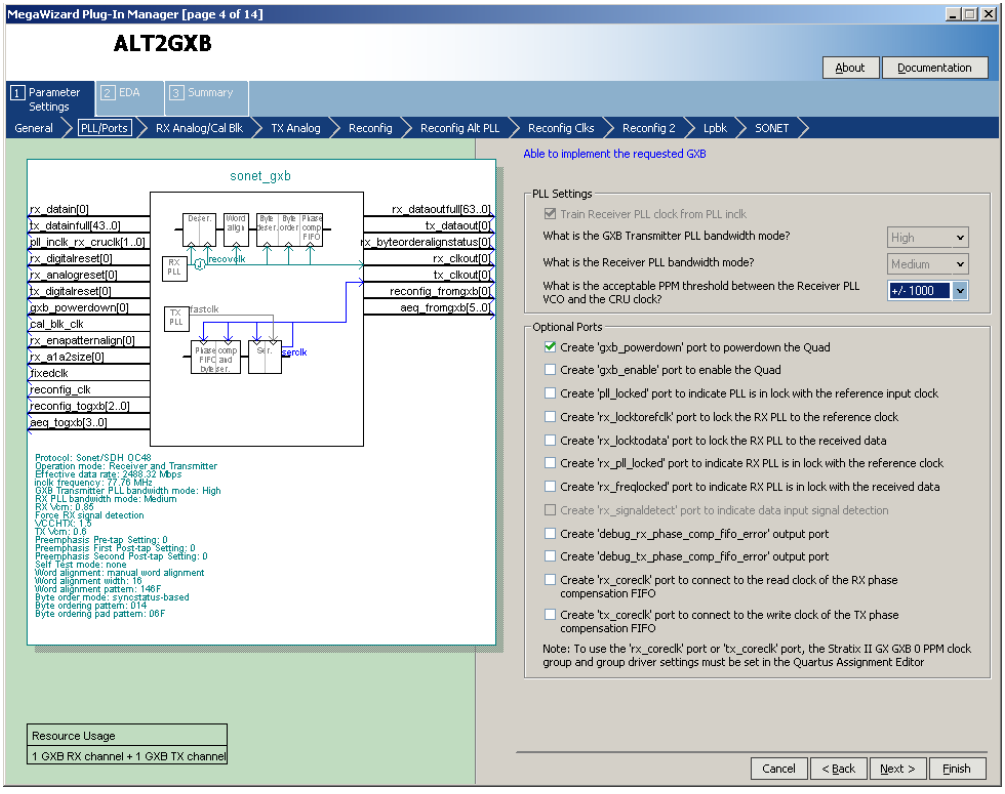


Table 4–38 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–38. MegaWizard Plug-In Manager Options (Page 4 for SONET/SDH Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If the **CMU PLL reconfiguration** option is enabled, it is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	This option is not available in SONET/SDH mode because the transmitter PLL bandwidth is fixed at high.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver PLL bandwidth mode?	This option is not available in SONET/SDH mode because the receiver PLL bandwidth is fixed at medium.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver CRU switchover between lock-to-data and lock-to-reference. (There are additional factors that affect CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–38. MegaWizard Plug-In Manager Options (Page 4 for SONET/SDH Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicate data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–38. MegaWizard Plug-In Manager Options (Page 4 for SONET/SDH Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–49 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode.

Figure 4–49. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

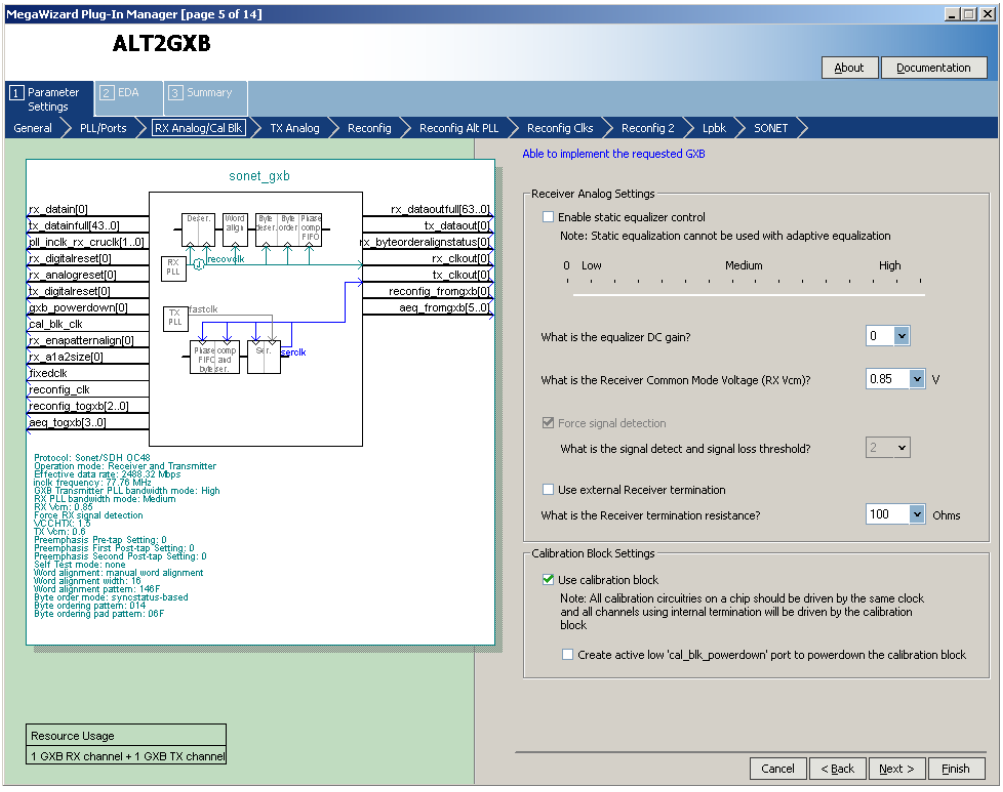


Table 4–39 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–39. MegaWizard Plug-In Manager Options (Page 5 for SONET/SDH Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is set to 0.85 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Force signal detection	This option disables the signal detect circuit. This removes the signal detect criterion for the receiver CRU lock-to-reference and lock-to-data switchover.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	Use this option when the forced signal detection option is off and to set the trip point of the signal detect circuit. The levels are to be determined after characterization.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–39. MegaWizard Plug-In Manager Options (Page 5 for SONET/SDH Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create active low `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–50 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode.

Figure 4–50. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

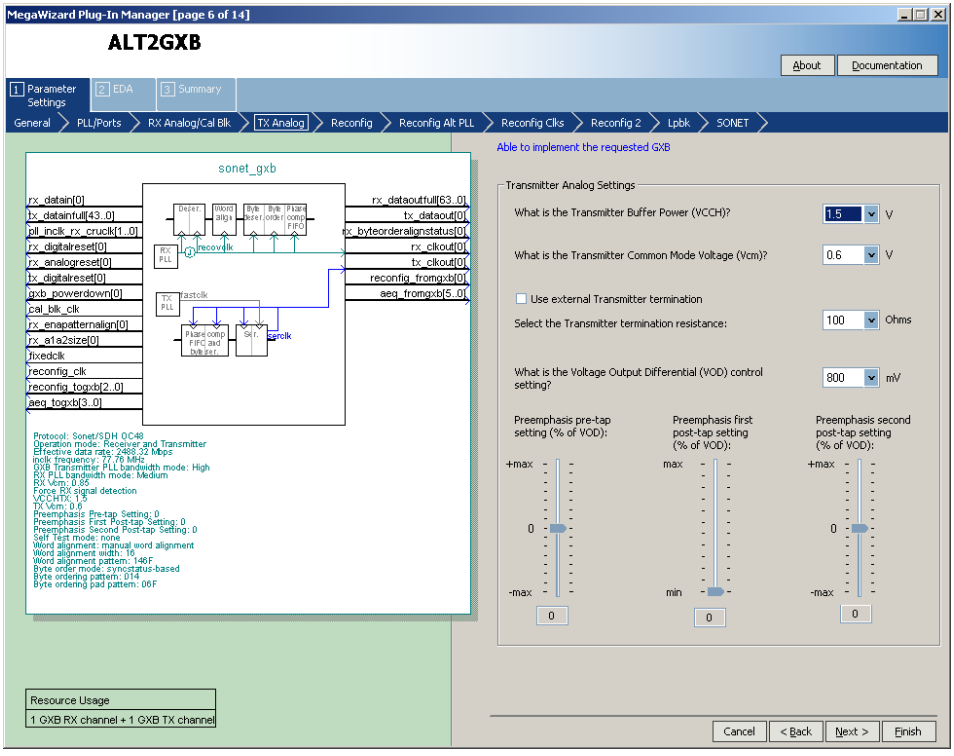


Table 4–40 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–40. MegaWizard Plug-In Manager Options (Page 6 for SONET/SDH Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. The selections available are 1.2 V and 1.5 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Restrictions apply based on the V_{CCH} setting.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap. The amount of pre-emphasis is to be determined by characterization.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–40. MegaWizard Plug-In Manager Options (Page 6 for SONET/SDH Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap. The amount of pre-emphasis is to be determined by characterization.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap. The amount of pre-emphasis is to be determined by characterization.	

Figure 4–51 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode.

Figure 4–51. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

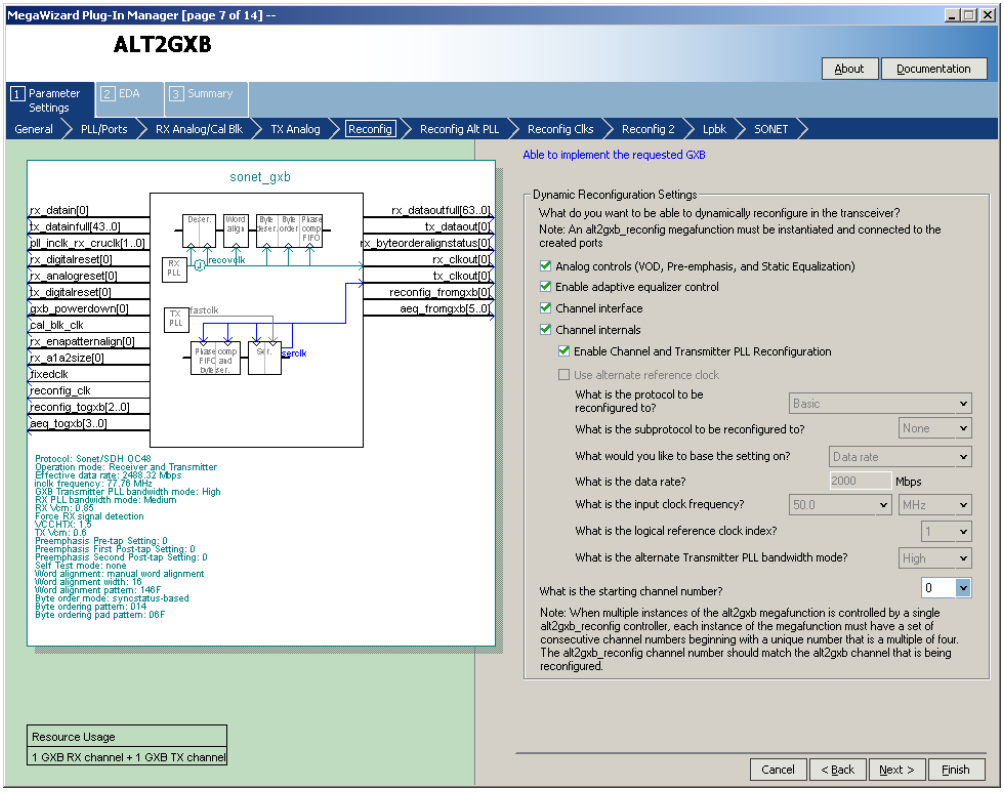


Table 4–41 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–41. MegaWizard Plug-In Manager Options (Page 7 for SONET/SDH Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are: • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel Interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals. • Channel Internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–52 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–52. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

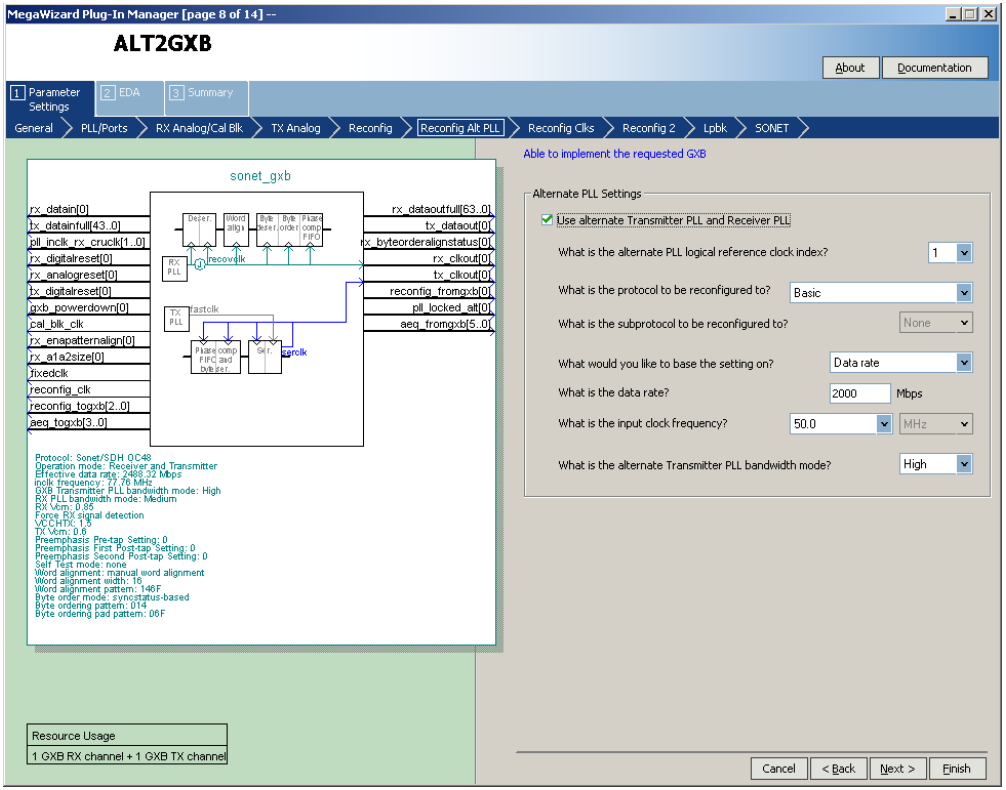


Table 4–42 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–42. MegaWizard Plug-In Manager Options (Page 8 for SONET/SDH Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–53 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–53. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

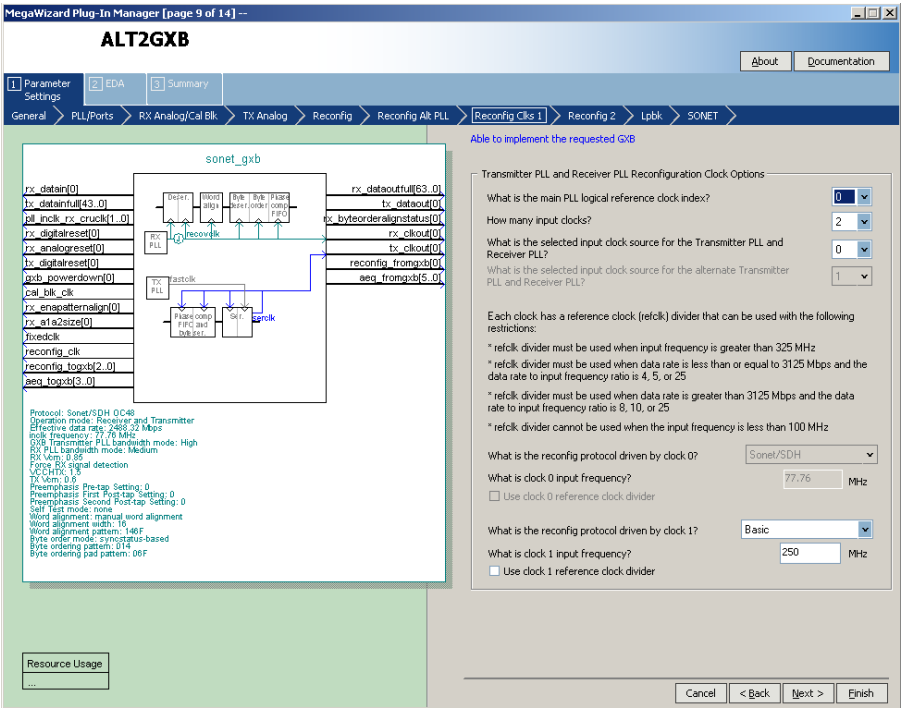


Table 4–43 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–43. MegaWizard Plug-In Manager Options (Page 9 for SONET/SDH Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–54 shows page 10 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode. This page appears only when the **Channel Internals** or **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–54. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

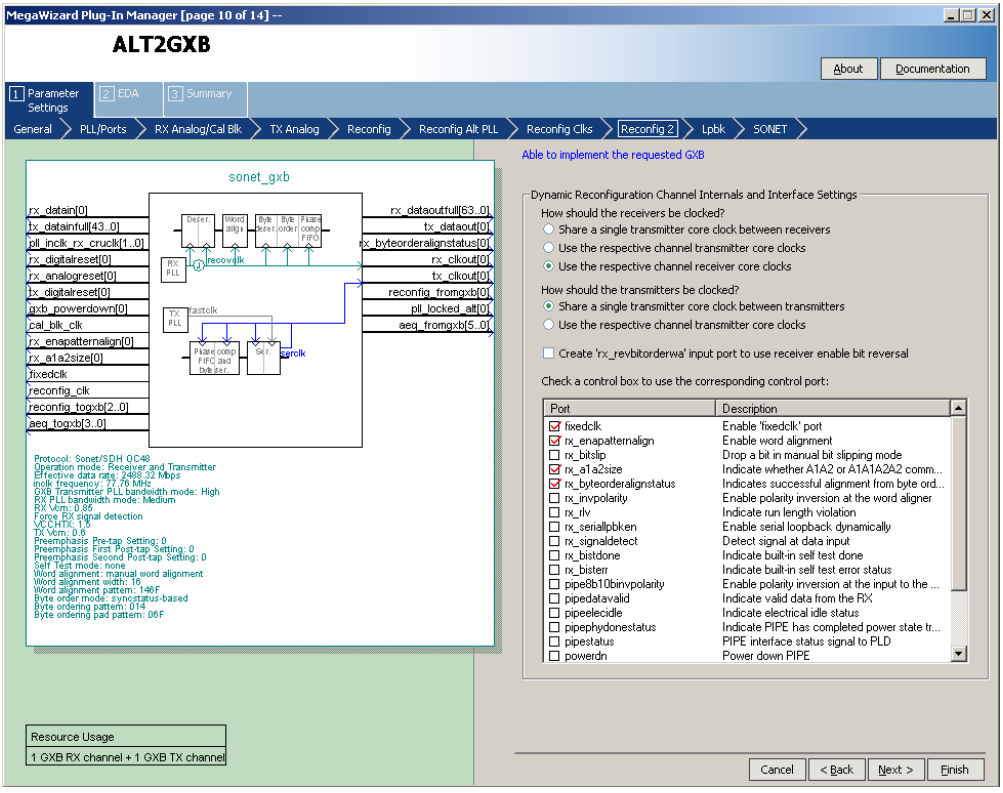


Table 4–44 describes the available options on page 10 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–44. MegaWizard Plug-In Manager Options (Page 10 for SONET/SDH Mode)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–55 shows page 11 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode.

Figure 4–55. MegaWizard Plug-In Manager - ALT2GXB (Loopback)

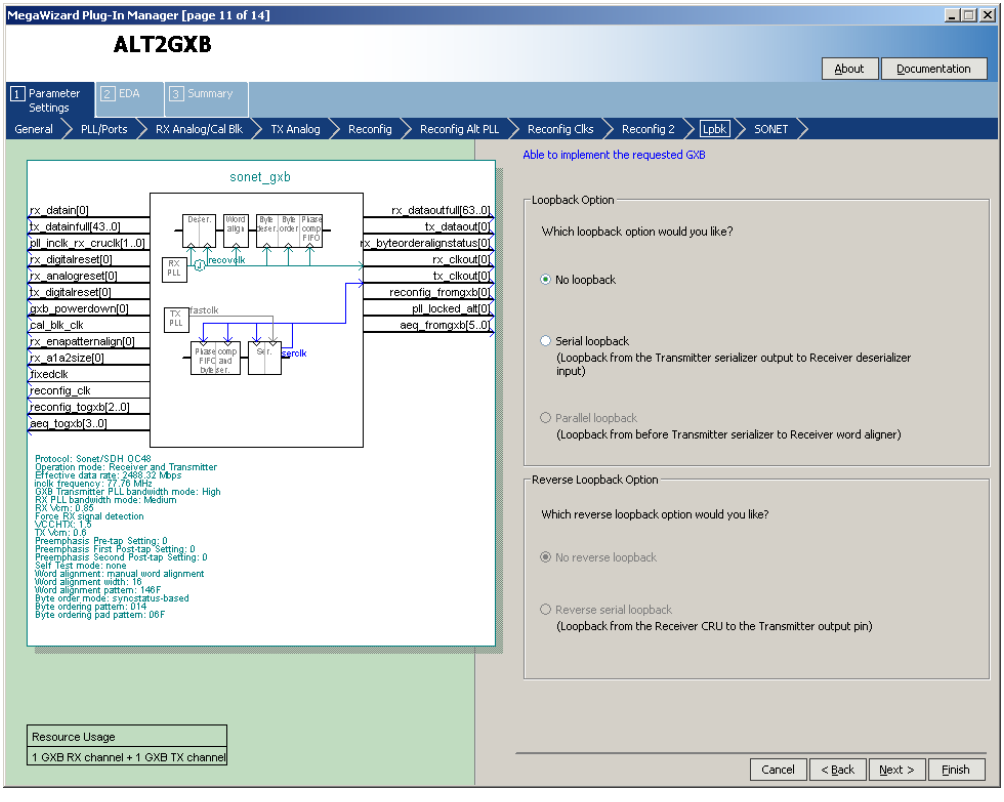


Table 4–45 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–45. MegaWizard Plug-In Manager Options (Page 11 for SONET/SDH Mode)

ALT2GXB Setting	Description	Reference
Which loopback option would you like?	There are two options available in SONET/SDH mode: no loopback and serial loopback. • No loopback - this is the default mode. • Serial loopback - if you select serial loopback, the `rx_seriallpbken` port is available to control the serial loopback feature dynamically. A 1'b1 enables serial loopback and a 1'b0 disables loopback on a channel-by-channel basis. Altera recommends controlling all four channels simultaneously. A digital reset must be asserted for the transceiver.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Reverse Loopback option	This option is not available in SONET/SDH mode.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–56 shows page 12 of the ALT2GXB MegaWizard Plug-In Manager for SONET/SDH mode. If the **Enforce default settings for this protocol** option is selected, this page does not appear in the MegaWizard.

Figure 4–56. MegaWizard Plug-In Manager - ALT2GXB (SONET)

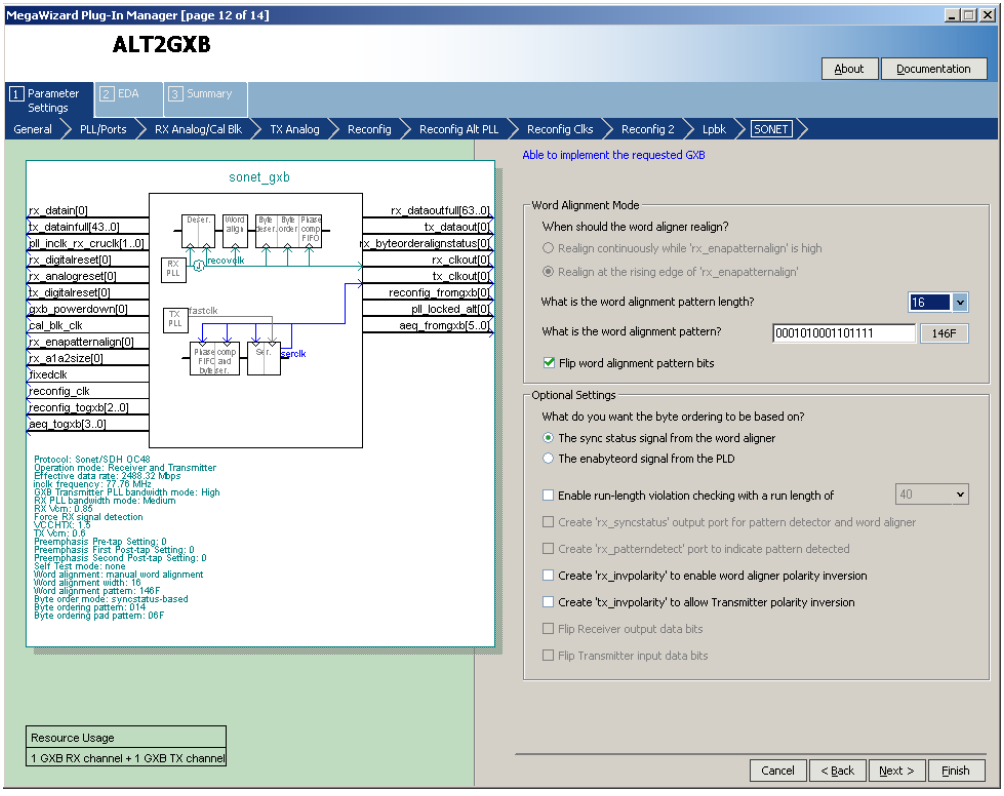


Table 4–46 describes the available options on page 12 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–46. MegaWizard Plug-In Manager Options (Page 12 for SONET/SDH Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
What should the word aligner realign?	In SONET/SDH mode, the word aligner is defaulted to manual alignment. The re-alignment is also defaulted to occur following a rising edge of the `rx_enapatternalign` input signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the word alignment pattern length?	This option sets the length of the word alignment. The following options are available: • OC-12 – only 16-bit pattern is allowed • OC-48 – only 16-bit pattern is allowed • OC-96 – 16-bit and 32-bit patterns are allowed	SONET/SDH Mode (OC-12, OC-48, and OC-96) section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the word alignment pattern?	Enter the word alignment pattern here. By default, the pattern that appears in the MegaWizard Plug-In Manager is 0001010001101111 (16'h146F).	SONET/SDH Mode (OC-12, OC-48, and OC-96) section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Flip word alignment pattern bits	This option reverses the order of the alignment pattern at a byte level to support MSB to LSB transmission protocols such as SONET/SDH. (This option is used in conjunction with the Flip receiver output data bits and Flip transmitter input data bits options.) By default, this option is selected in the MegaWizard Plug-In Manager.	SONET/SDH Mode (OC-12, OC-48, and OC-96) section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What do you want the byte ordering to be based on?	This option allows you to trigger the byte ordering block either on the rising edge of `rx_syncstatus` signal or user-controlled `rx_enabyteord` signal from the PLD.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–46. MegaWizard Plug-In Manager Options (Page 12 for SONET/SDH Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Enable run-length violation checking with a run-length of	When enabled, this option activates the run length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_patterndetect` output port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–46. MegaWizard Plug-In Manager Options (Page 12 for SONET/SDH Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Flip Receiver output data bits	This option reverses the bit order of the receiver output data (`rx_dataout`) at a byte level to support MSB to LSB transmission protocols such as SONET/SDH.	SONET/SDH Mode (OC-12, OC-48, and OC-96) section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Flip Transmitter input data bits	This option reverse the bit order of the transmitter input data (`tx_datain`) at a byte level to support MSB to LSB transmission protocols such as SONET/SDH.	SONET/SDH Mode (OC-12, OC-48, and OC-96) section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–57 shows page 13 of the MegaWizard Plug-In Manager for the SONET/SDH protocol selection. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for third party EDA synthesis tool to be able to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–57. MegaWizard Plug-In Manager - ALT2GXB (EDA)

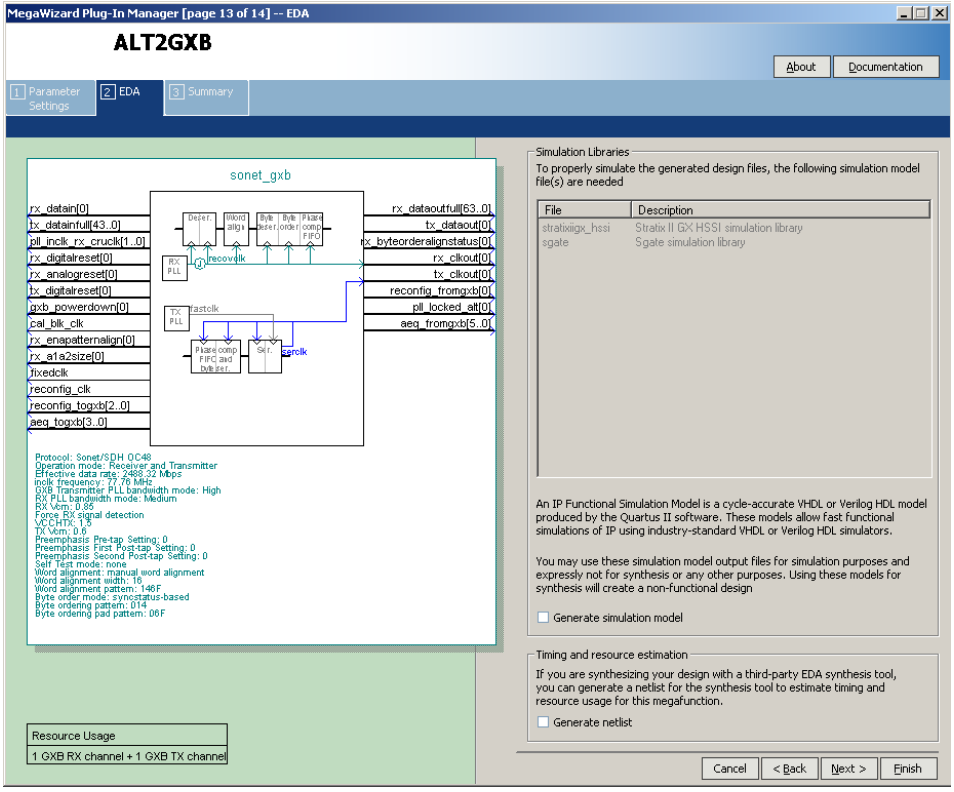
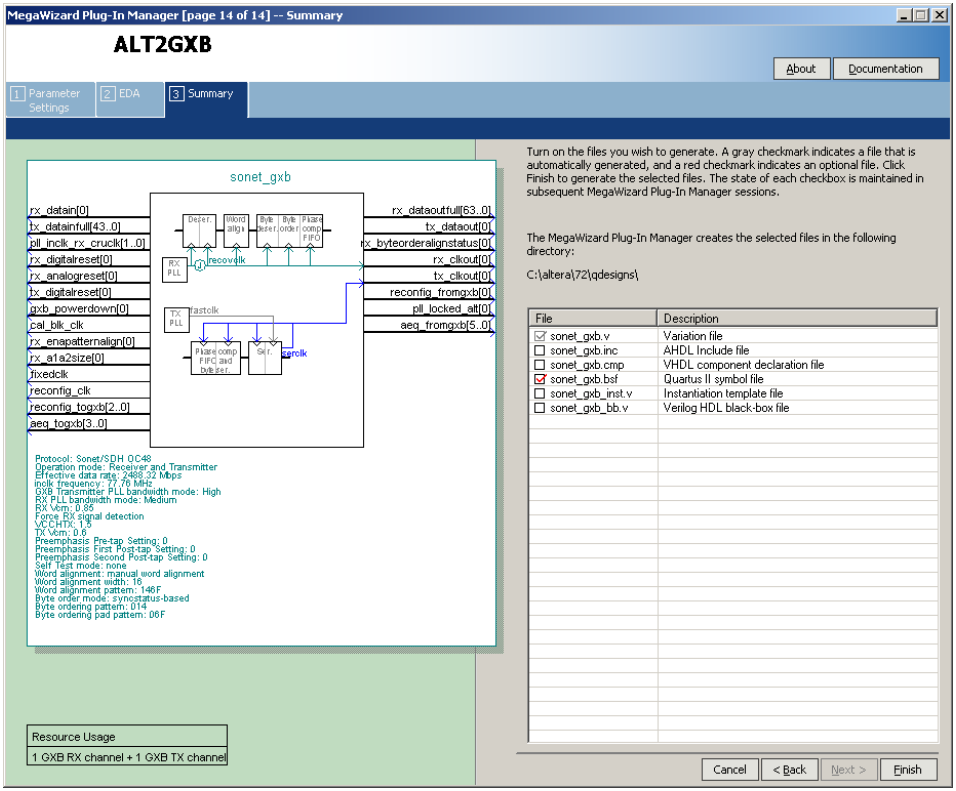


Figure 4–58 shows page 14 (the last page) of the MegaWizard Plug-In Manager for the SONET/SDH protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–58. MegaWizard Plug-In Manager - ALT2GXB (Summary)



(OIF) CEI PHY Interface Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for the (OIF) CEI PHY Interface mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

Figure 4–59 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager for (OIF) CEI PHY Interface mode.

Figure 4–59. MegaWizard Plug-In Manager - ALT2GXB (General)

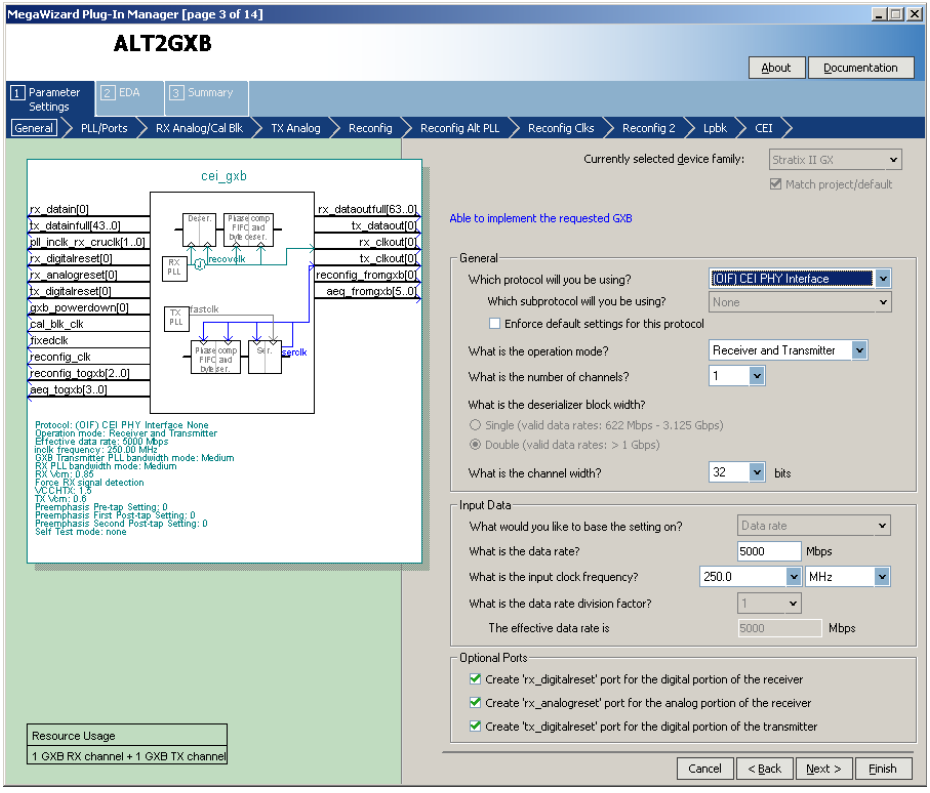


Table 4–47 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–47. MegaWizard Plug-In Manager Options (Page 3 for [OIF] CEI PHY Interface Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Selects the specific protocol or modes that the transceiver operates under. For the (OIF) CEI PHY Interface mode, you must select the **(OIF) CEI PHY Interface** protocol.	
Which subprotocol will you be using?	This option does not apply to the (OIF) CEI PHY Interface mode.	

Table 4–47. MegaWizard Plug-In Manager Options (Page 3 for [OIF] CEI PHY Interface Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Enforce default settings for this protocol	If this option is checked, all (OIF) CEI PHY Interface-specific ports are used.	
What is the operation mode?	The transmitter only, receiver only, and receiver and transmitter (full duplex) modes are allowed in the (OIF) CEI PHY Interface mode.	
What is the number of channels?	This selects how many duplicate channels this ALT2GXB instance contains. In (OIF) CEI PHY Interface mode, the number of channels increments by 1.	
What is the deserializer block width?	The (OIF) CEI PHY Interface operates in double-width mode only. Single-width mode is not allowed.	
What is the channel width?	This option selects the transceiver to PLD interface width. Only 32 bits are allowed in (OIF) CEI PHY Interface mode.	Byte Serializer and Byte Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option is fixed to **Data rate** in (OIF) CEI PHY Interface mode.	
What is the data rate?	This field allows you to enter the data rate. In (OIF) CEI PHY mode, you can enter a data rate between 3125 Gbps and 6375 Gbps.	(OIF) CEI PHY Interface Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the input clock frequency?	This field allows you to select the available input reference clock frequencies in (OIF) CEI PHY Interface mode.	(OIF) CEI PHY Interface Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the data rate division factor?	This option is not available in (OIF) CEI PHY Interface mode.	
Create `rx_digitalreset` port for the digital portion of the receiver	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–47. MegaWizard Plug-In Manager Options (Page 3 for [OIF] CEI PHY Interface Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_analogreset` port for the analog portion of the receiver	Receiver analog reset port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_digitalreset` port for the digital portion of the transmitter	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–60 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for (OIF) CEI PHY Interface mode.

Figure 4–60. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

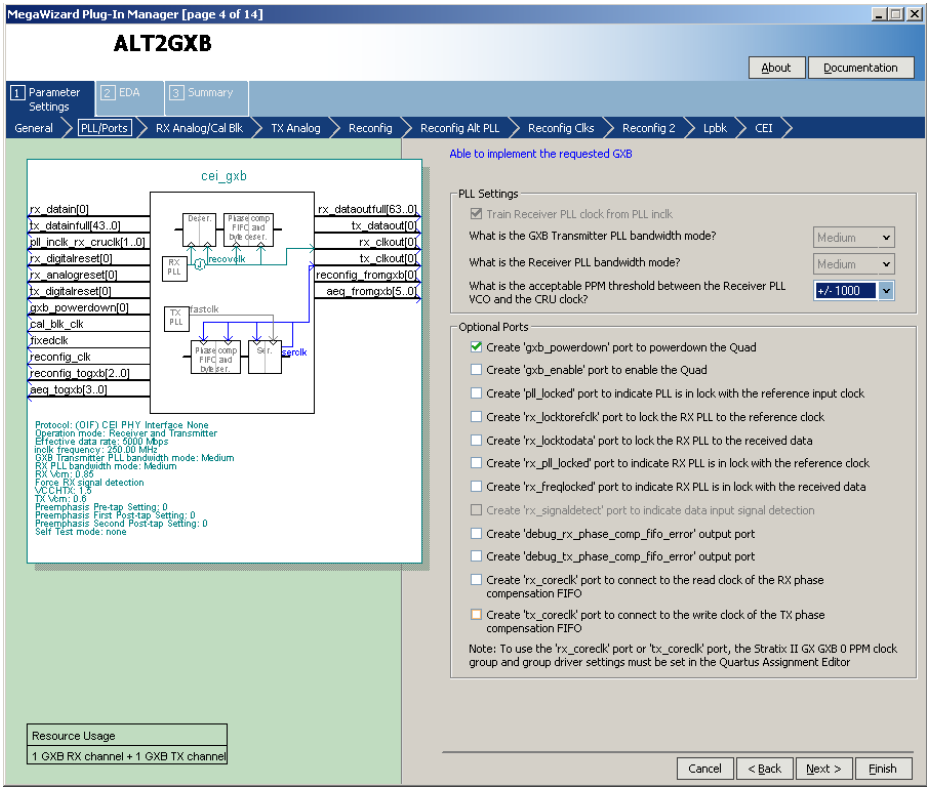


Table 4–48 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–48. MegaWizard Plug-In Manager Options (Page 4 for [OIF] CEI PHY Interface Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If **CMU PLL reconfiguration** is enabled, this option is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	In (OIF) CEI PHY Interface mode, only medium bandwidth is supported for the transmitter PLL.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver PLL bandwidth mode?	In (OIF) CEI PHY Interface mode, only medium bandwidth is supported for the receiver PLL and VCO.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver CRU switchover between lock-to-data and lock-to-reference. (There are additional factors that affect CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–48. MegaWizard Plug-In Manager Options (Page 4 for [OIF] CEI PHY Interface Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicate data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–48. MegaWizard Plug-In Manager Options (Page 4 for [OIF] CEI PHY Interface Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–61 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for (OIF) CEI PHY Interface mode.

Figure 4–61. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

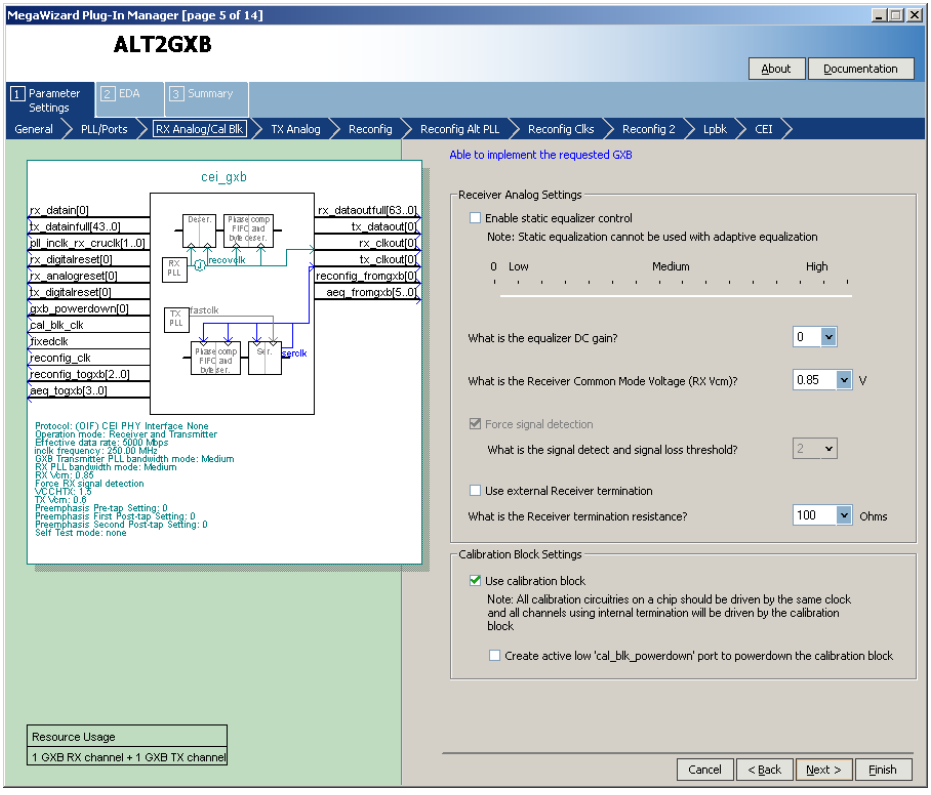


Table 4–49 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–49. MegaWizard Plug-In Manager Options (Page 5 for [OIF] CEI PHY Interface Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is set to 0.85 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Force signal detection	This option disables the signal detect circuit. This removes the signal detect criterion for the receiver CRU lock-to-reference and lock-to-data switchover.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	Use this option when the forced signal detection option is off and to set the trip point of the signal detect circuit. The levels are to be determined after characterization.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–49. MegaWizard Plug-In Manager Options (Page 5 for [OIF] CEI PHY Interface Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–62 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for (OIF) CEI PHY Interface mode.

Figure 4–62. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

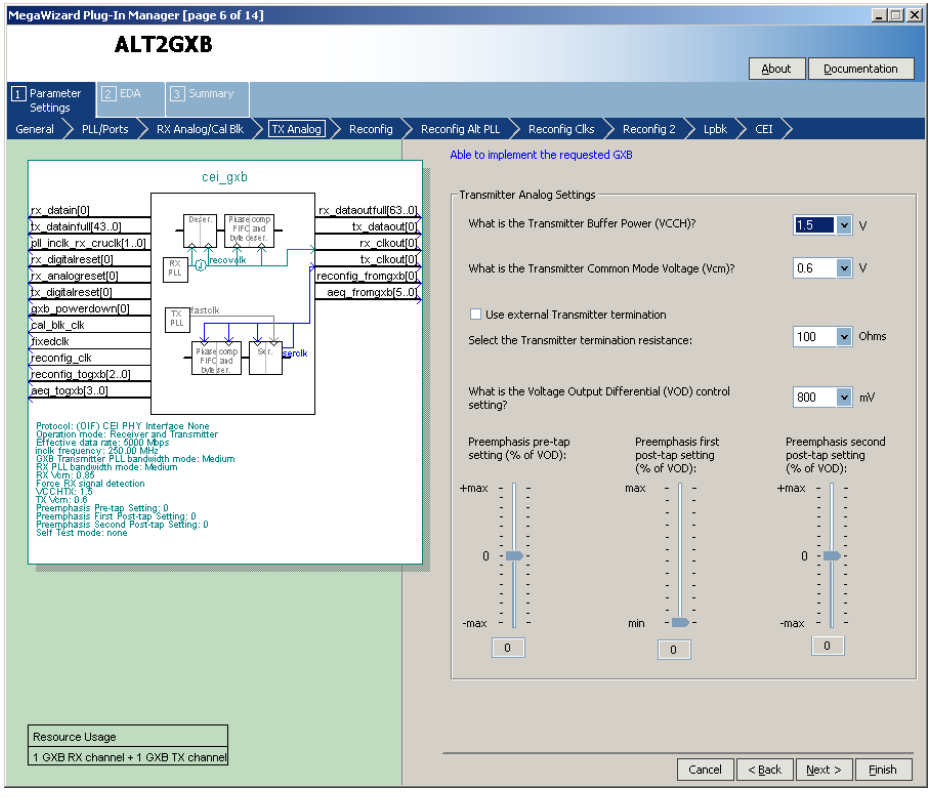


Table 4–50 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–50. MegaWizard Plug-In Manager Options (Page 6 for [OIF] CEI PHY Interface Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. Only 1.5 .V is allowed in (OIF) CEI PHY Interface mode.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Only 0.6 V is allowed in (OIF) CEI PHY Interface mode.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap. The amount of pre-emphasis is to be determined by characterization.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–50. MegaWizard Plug-In Manager Options (Page 6 for [OIF] CEI PHY Interface Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap. The amount of pre-emphasis is to be determined by characterization.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap. The amount of pre-emphasis is to be determined by characterization.	

Figure 4–63 shows page 7 of theALT2GXB MegaWizard Plug-In Manager for (OIF) CEI PHY Interface mode.

Figure 4–63. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

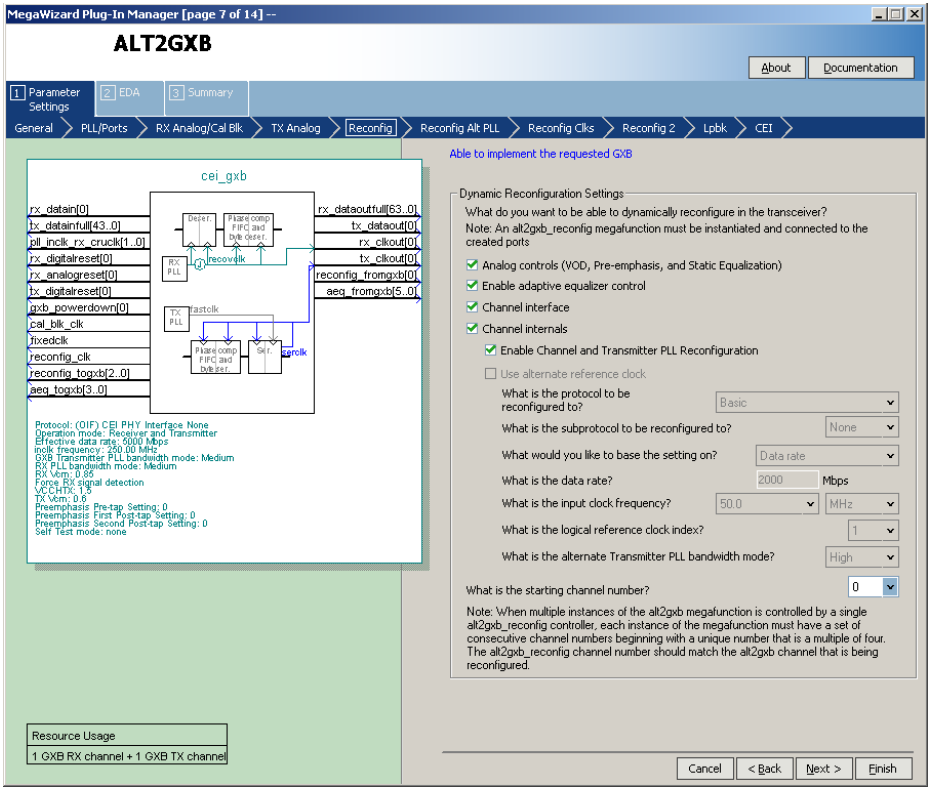


Table 4–51 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–51. MegaWizard Plug-In Manager Options (Page 7 for [OIF] CEI PHY Interface Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are: • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel Interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals. • Channel Internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–64 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for (OIF) CEI PHY Interface mode. This page appears only if the **Channel Internals** and **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–64. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

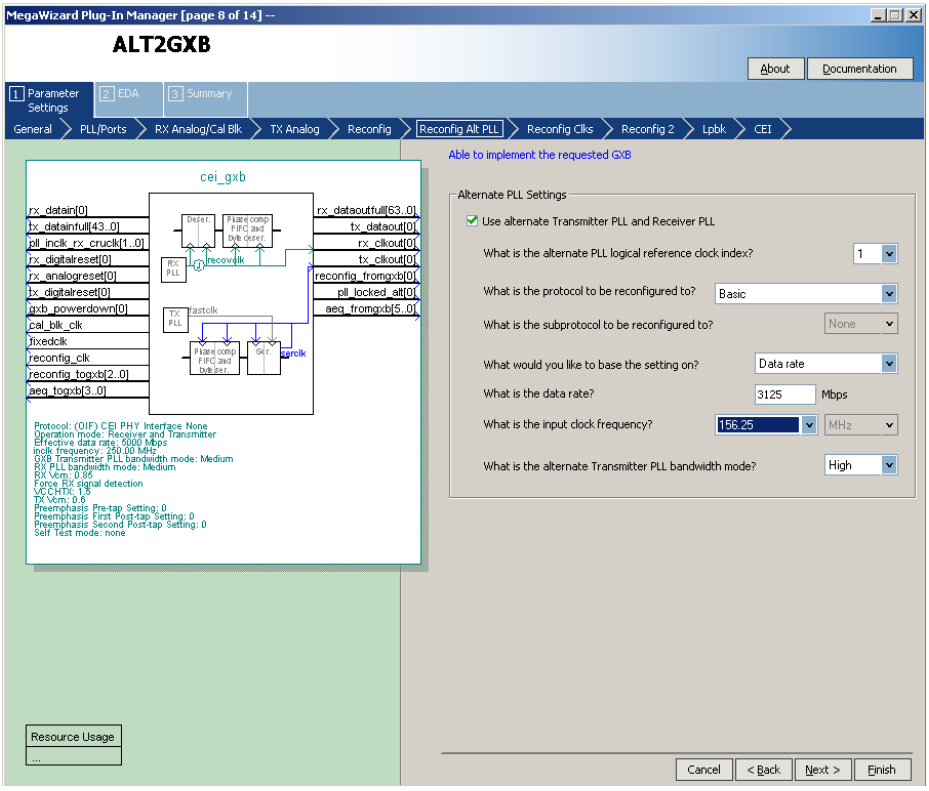


Table 4–52 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–52. MegaWizard Plug-In Manager Options (Page 8 for [OIF] CEI PHY Interface Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–65 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for (OIF) CEI PHY Interface mode. This page appears only if the **Channel Internals** and **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–65. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

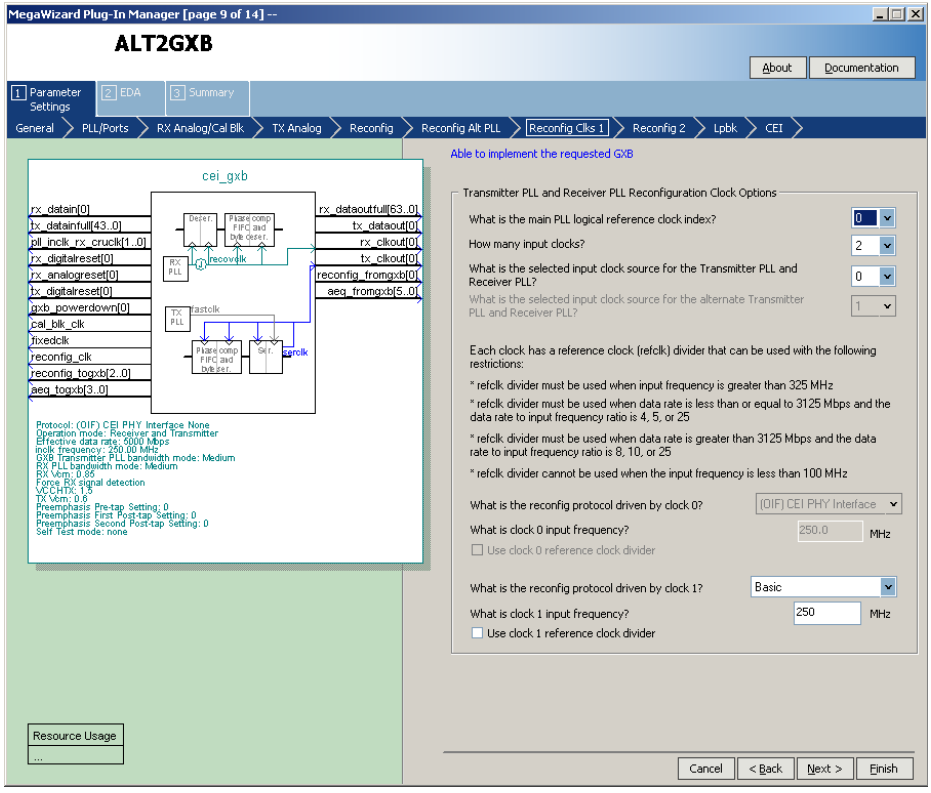


Table 4–53 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–53. MegaWizard Plug-In Manager Options (Page 9 for [OIF] CEI PHY Interface Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–66 shows page 10 of the MegaWizard Plug-In Manager for the (OIF) CEI PHY Interface protocol selection. This page appears only when the **Channel Internals** or **Channel Interface** option is selected in the **Reconfig** page (Page 7).

Figure 4–66. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

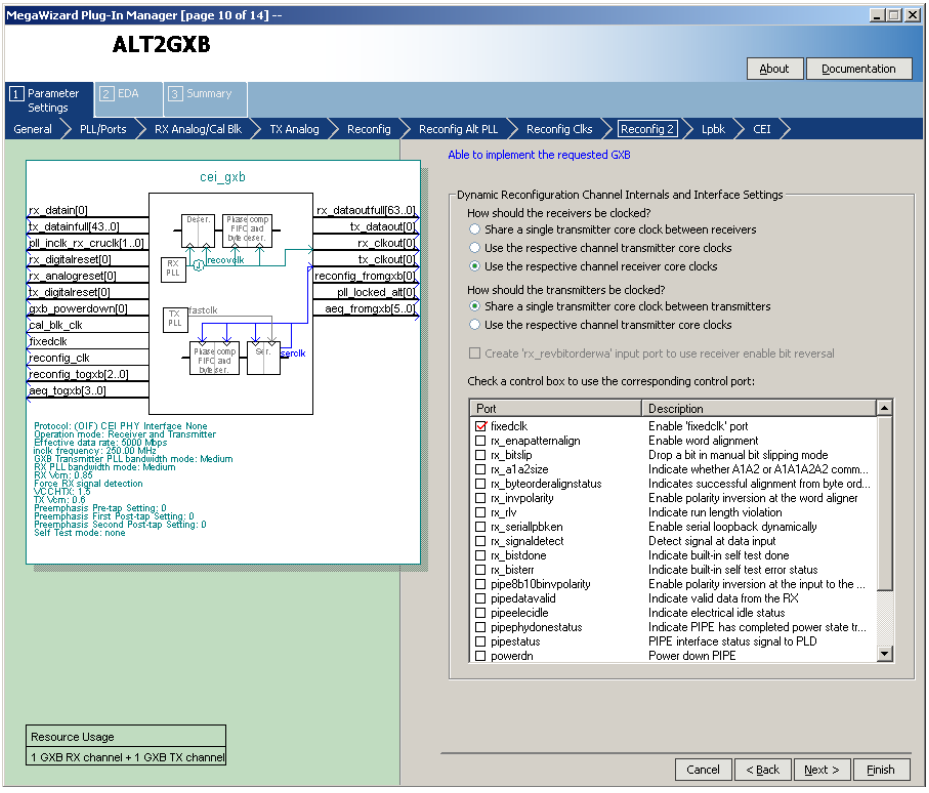


Table 4–54 describes the available options on page 10 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–54. MegaWizard Plug-In Manager Options (Page 10 for [OIF] CEI PHY Interface Mode)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–67 shows page 11 of the MegaWizard Plug-In Manager for the (OIF) CEI PHY Interface protocol selection.

Figure 4–67. MegaWizard Plug-In Manager - ALT2GXB (Loopback)

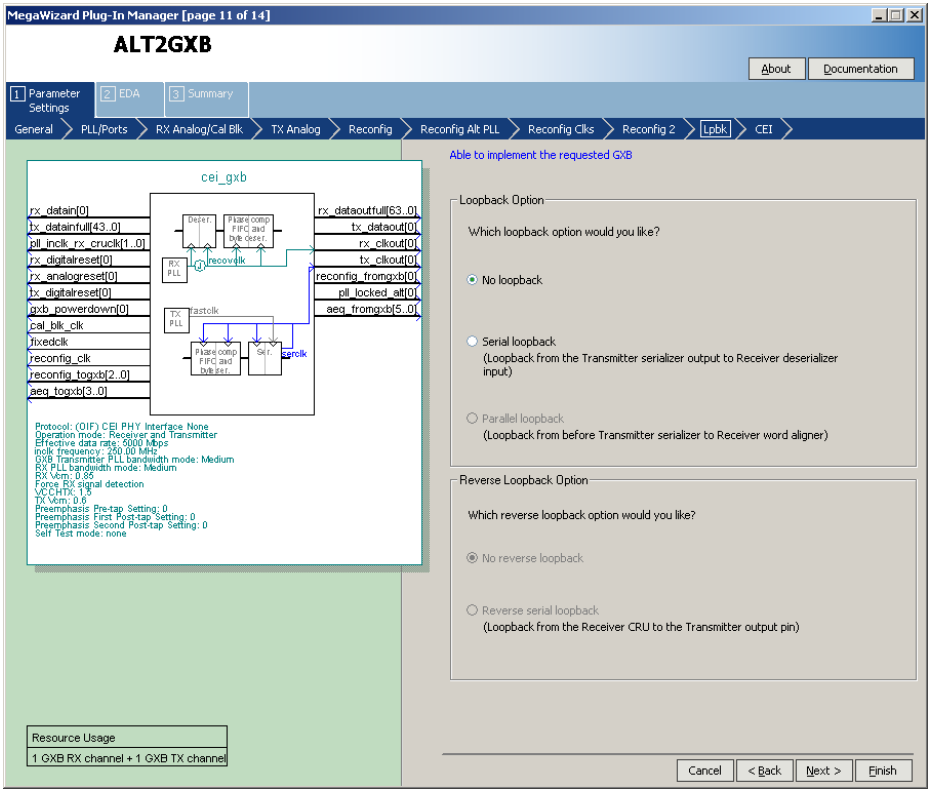


Table 4–55 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–55. MegaWizard Plug-In Manager Options (Page 11 for [OIF] CEI PHY Interface Mode)

ALT2GXB Setting	Description	Reference
Which loopback option would you like?	There are two options available in (OIF) CEI PHY Interface mode: no loopback and serial loopback. • No loopback - this is the default mode. • Serial loopback - if you select serial loopback, the `rx_seriallpbken` port is available to control the serial loopback feature dynamically. A 1'b1 enables serial loopback and a 1'b0 disables loopback on a channel-by-channel basis. Altera recommends controlling all four channels simultaneously. A digital reset must be asserted for the transceiver.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Reverse Loopback option	This option is not available in (OIF) CEI PHY Interface mode.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–68 shows the **CEI** page of the MegaWizard Plug In Manager for the (OIF) CEI PHY Interface protocol selection. If the **Enforce** default settings for this protocol option is selected, this page does not appear in the MegaWizard. The **Use central clock divider to improve transmitter jitter** option allows you to set-up bundled clocking for channels within the same transceiver block to improve transmitter jitter performance. This option is not available if the **Channel Interface** or **Channel Internals** options with **alternate PLL** or **CMU PLL reconfiguration** are selected.

Figure 4–68. MegaWizard Plug-In Manager - ALT2GXB (CEI)

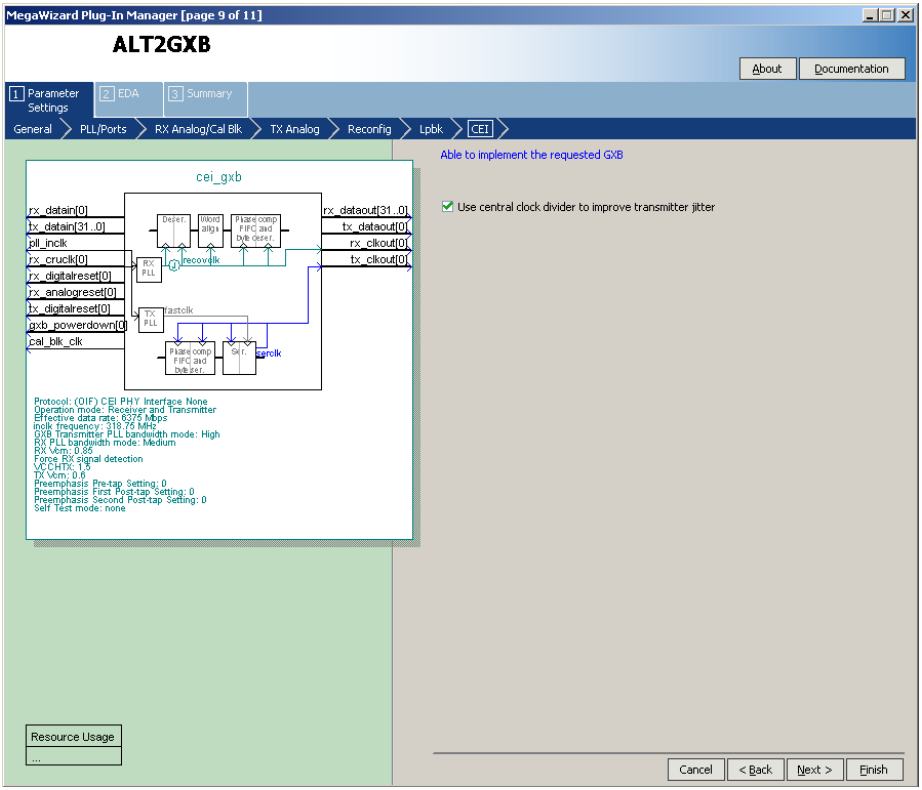


Figure 4–69 shows page 13 of the MegaWizard Plug-In Manager for the (OIF) CEI PHY Interface protocol selection. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for third party EDA synthesis tool to be able to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–69. MegaWizard Plug-In Manager - ALT2GXB (EDA)

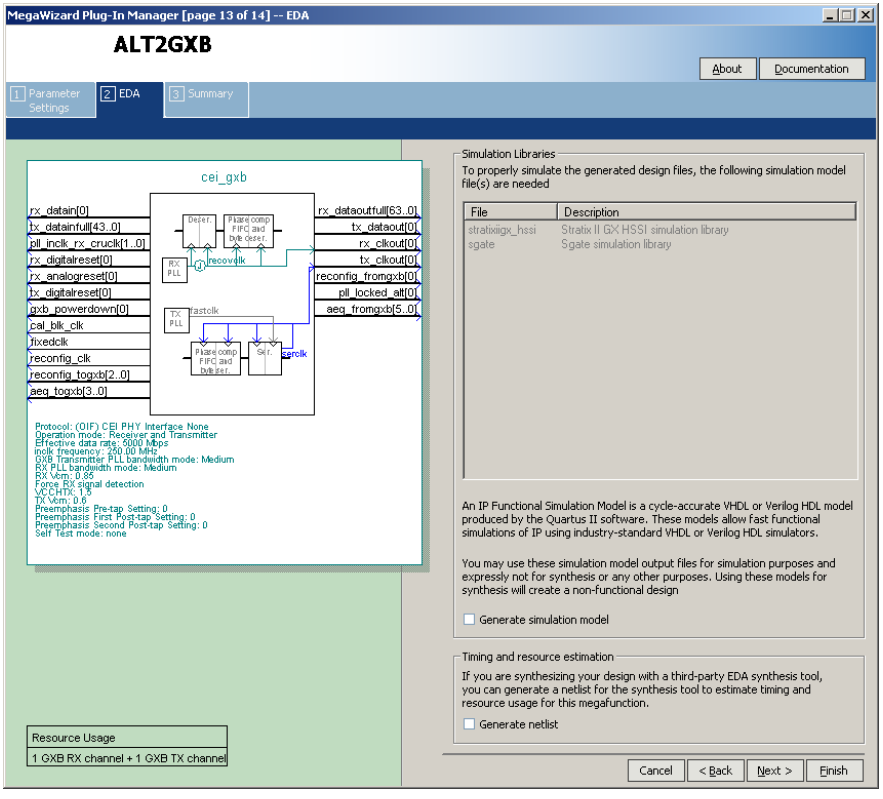
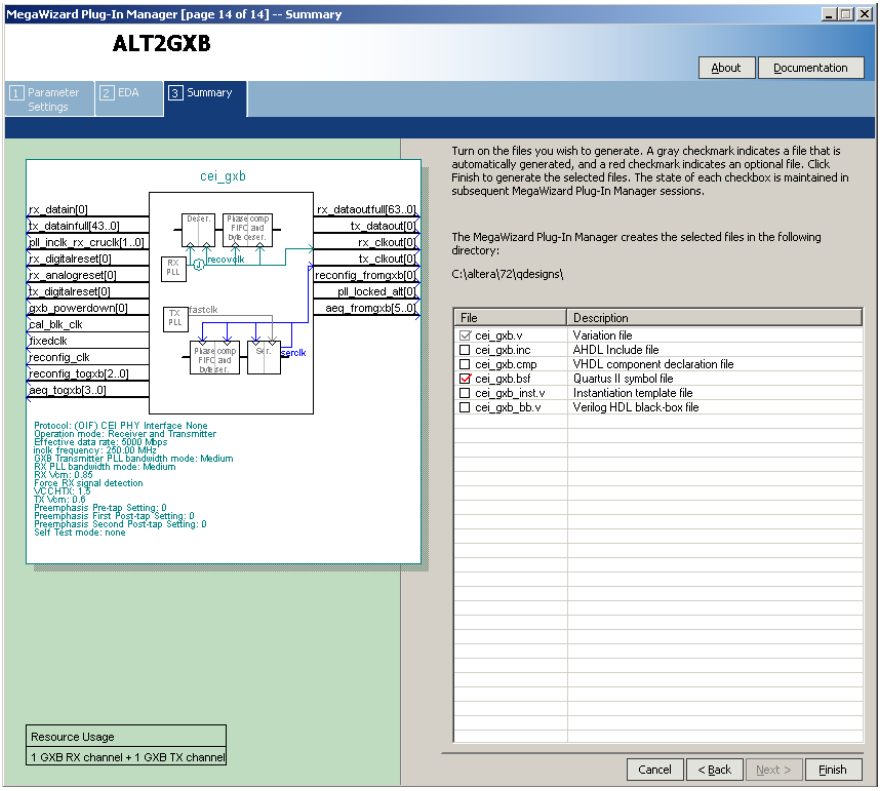


Figure 4–70 shows page 14 (the last page) of the MegaWizard Plug-In Manager for the (OIF) CEI PHY Interface protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–70. MegaWizard Plug-In Manager - ALT2GXB (Summary)



CPRI Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for the CPRI mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

Figure 4–71 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode.

Figure 4–71. MegaWizard Plug-In Manager - ALT2GXB (General)

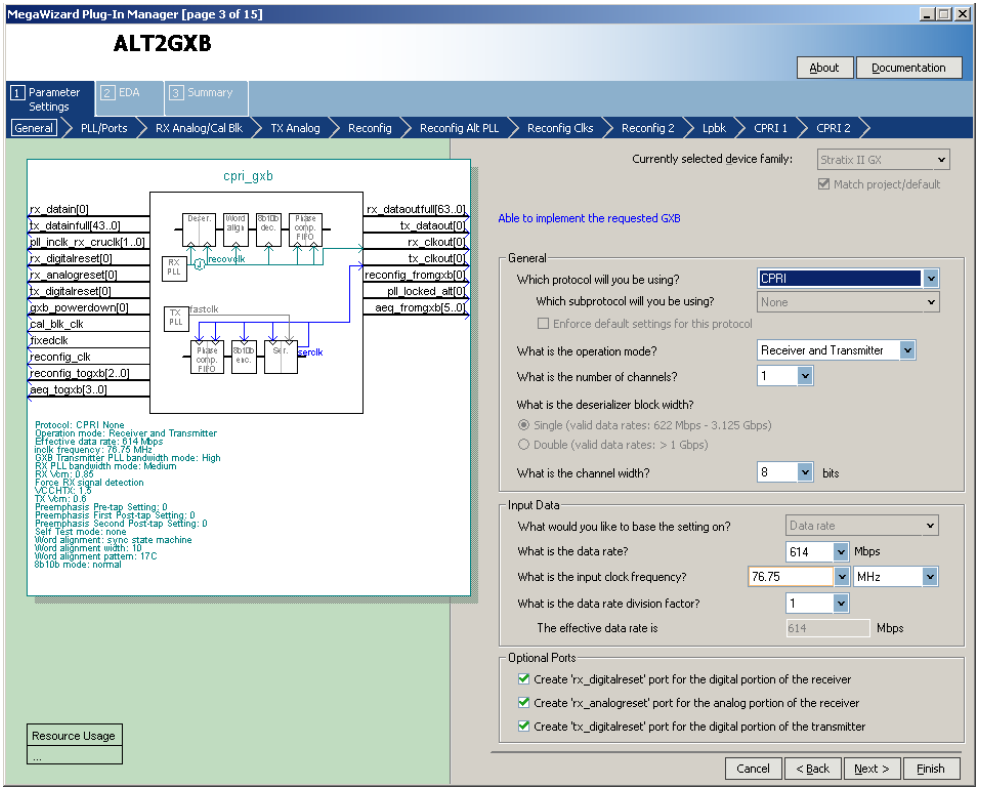


Table 4–56 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–56. MegaWizard Plug-In Manager Options (Page 3 for CPRI Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Selects the specific protocol or modes under which the transceiver operates. For the CPRI mode, you must select the **CPRI** protocol.	
Which subprotocol will you be using?	This option is not available in CPRI mode.	

Table 4–56. MegaWizard Plug-In Manager Options (Page 3 for CPRI Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Enforce default settings for this protocol	This option is not available in CPRI mode.	
What is the operation mode?	The transmitter only, receiver only, and receiver and transmitter (full duplex) modes are allowed in CPRI protocol.	
What is the number of channels?	This selects how many duplicate channels this ALT2GXB instance contains. In CPRI mode, the number of channels increments by 1. A maximum of 16 CPRI channels can be instantiated in the largest Stratix II GX device (EP2SGX130G) due to clocking constraints.	CPRI mode section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the deserializer block width?	CPRI mode only operates in a single-width mode. Double-width mode is not allowed.	
What is the channel width?	This option determines the transceiver to PLD interface width. In CPRI mode, 8 bits are allowed.	Byte Serializer and Byte Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option is forced to **Data rate**.	
What is the data rate?	Three CPRI data rates are supported: • 614 Mbps • 1228 Mbps • 2456 Mbps.	
What is the input clock frequency?	Determines the input clock frequency you want as a reference clock for the transceiver.	
What is the data rate division factor?	This setting, in conjunction with the selected data rate, determines the effective data rate for the transceiver channel.	
Create `rx_digitalreset` port for the digital portion of the receiver	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–56. MegaWizard Plug-In Manager Options (Page 3 for CPRI Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_analogreset` port for the analog portion of the receiver	Receiver analog reset port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_digitalreset` port for the digital portion of the transmitter	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–72 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode.

Figure 4–72. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

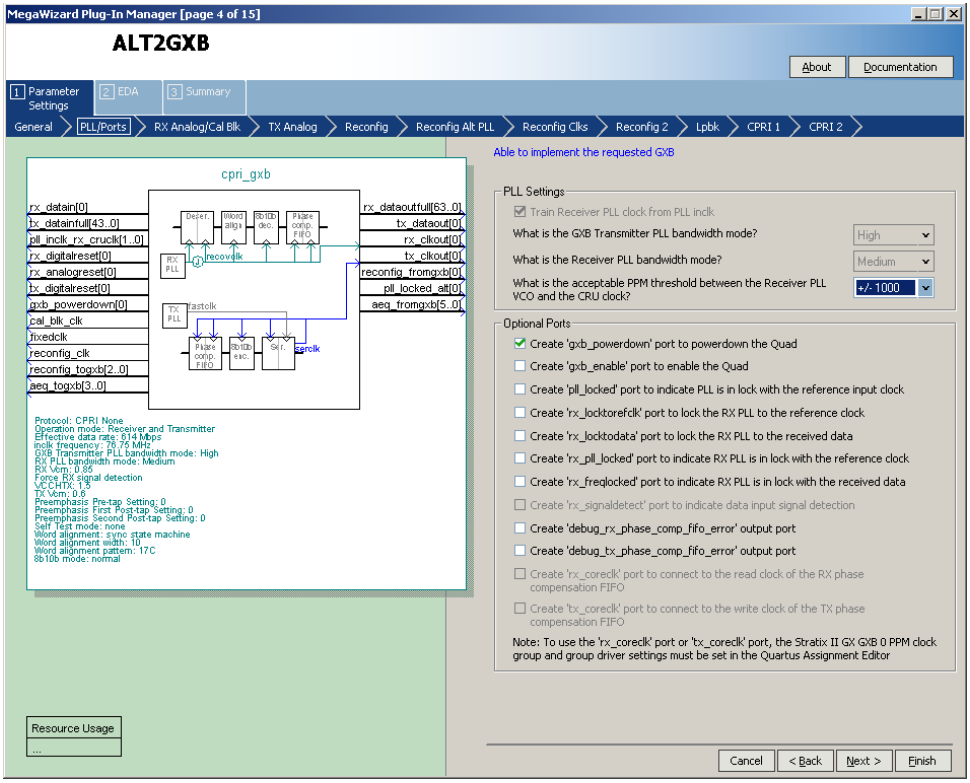


Table 4–57 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–57. MegaWizard Plug-In Manager Options (Page 4 for CPRI Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If **CMU PLL reconfiguration** is enabled, this option is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	In CPRI mode, only high bandwidth is supported for the transmitter PLL.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver PLL bandwidth mode?	In CPRI mode, only medium bandwidth is supported for the receiver PLL and V_{CO}.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver CRU switchover between lock-to-data and lock-to-reference. (There are additional factors that affect CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–57. MegaWizard Plug-In Manager Options (Page 4 for CPRI Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicate data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–57. MegaWizard Plug-In Manager Options (Page 4 for CPRI Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–73 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode.

Figure 4–73. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

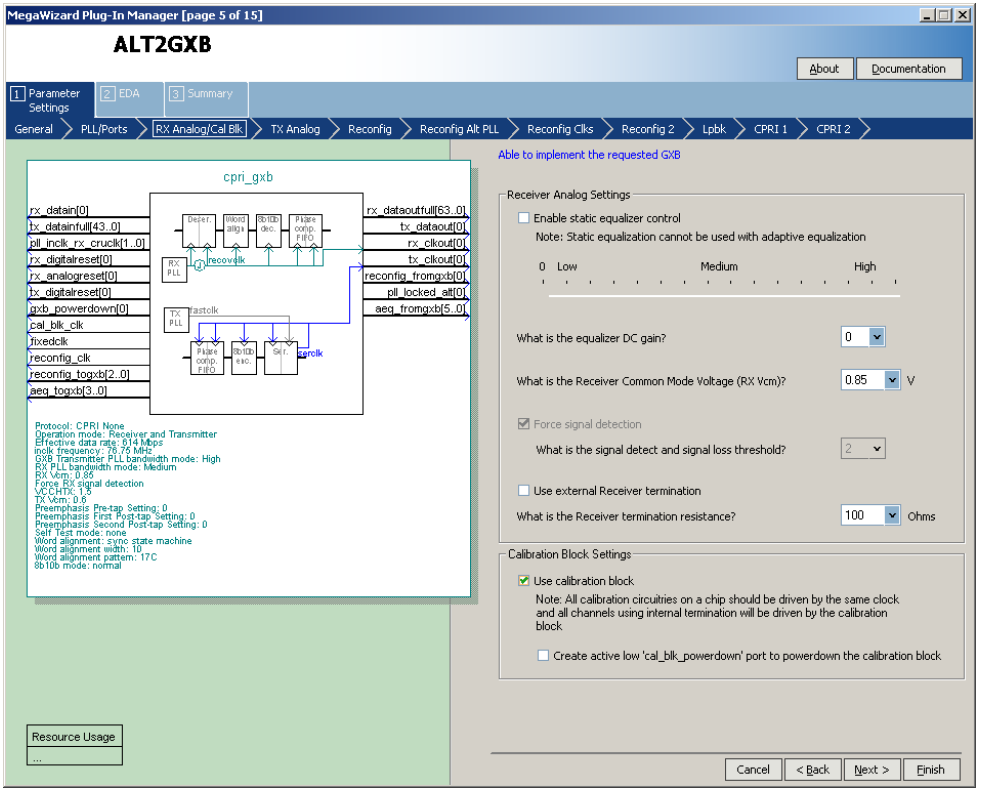


Table 4–58 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–58. MegaWizard Plug-In Manager Options (Page 5 for CPRI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is set to 0.85 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Forced signal detection	This option disables the signal detect circuit. This removes the signal detect criterion for the receiver CRU lock-to-reference and lock-to-data switchover.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	Use this option when the forced signal detection option is off and to set the trip point of the signal detect circuit. The levels are to be determined after characterization.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–58. MegaWizard Plug-In Manager Options (Page 5 for CPRI Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–74 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode.

Figure 4–74. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

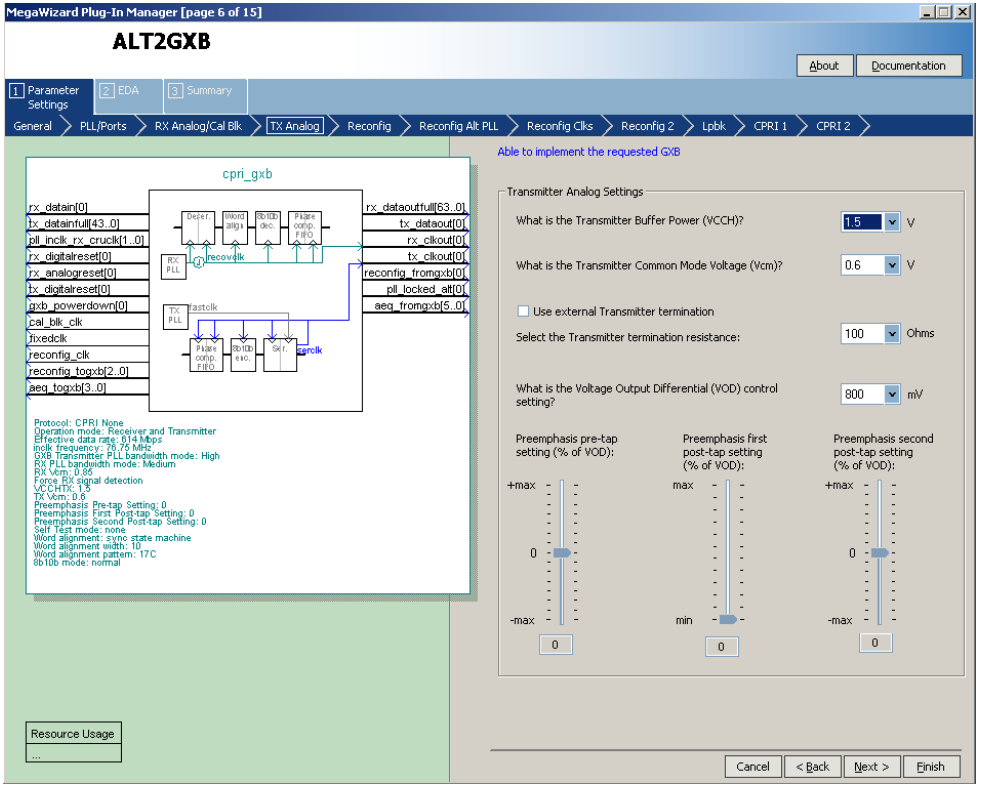


Table 4–59 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–59. MegaWizard Plug-In Manager Options (Page 6 for CPRI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. The selections available are 1.2 V and 1.5 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Restrictions apply based on the V_{CCH} setting.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap. The amount of pre-emphasis is to be determined by characterization.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–59. MegaWizard Plug-In Manager Options (Page 6 for CPRI Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap. The amount of pre-emphasis is to be determined by characterization.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap. The amount of pre-emphasis is to be determined by characterization.	

Figure 4–75 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode.

Figure 4–75. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

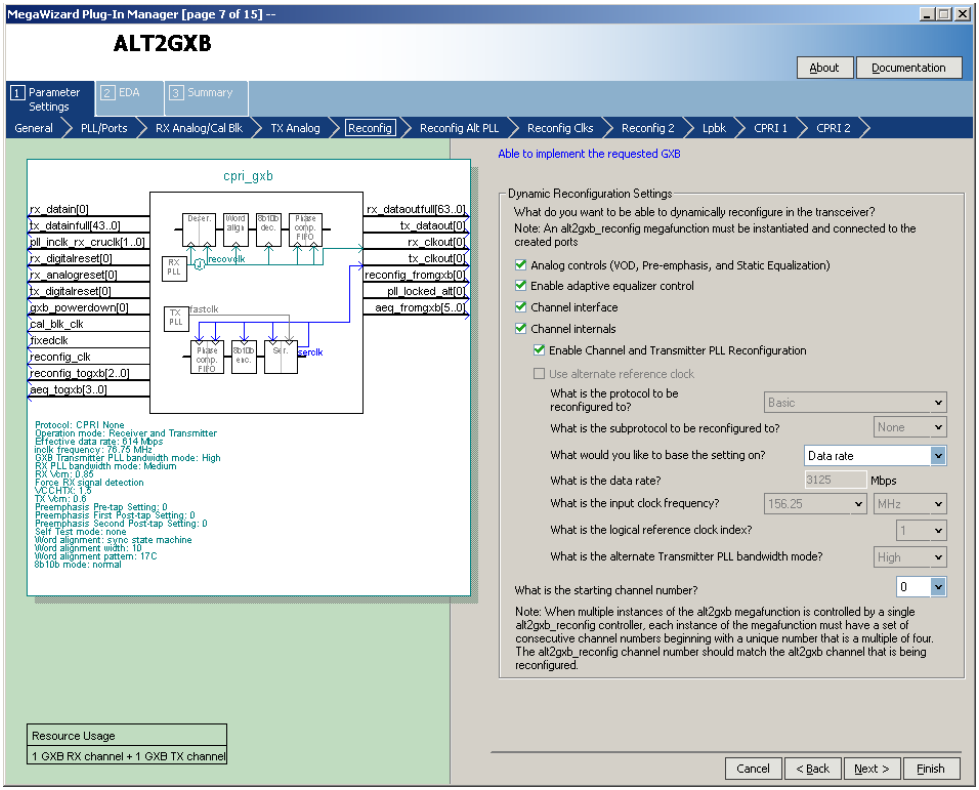


Table 4–60 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–60. MegaWizard Plug-In Manager Options (Page 7 for CPRI Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are: • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel Interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals. • Channel Internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–40 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode. This page appears only when the **Channel Internals** or **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–76. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

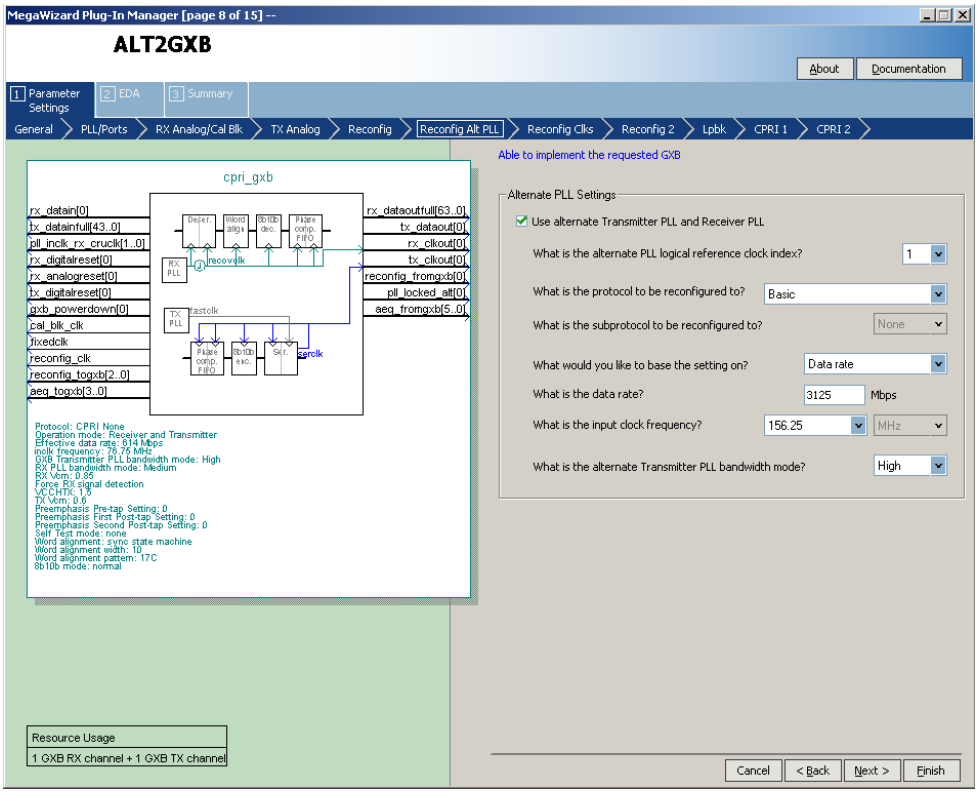


Table 4–61 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–61. MegaWizard Plug-In Manager Options (Page 8 for CPRI Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–77 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected on the **Reconfig** page (Page 7).

Figure 4–77. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

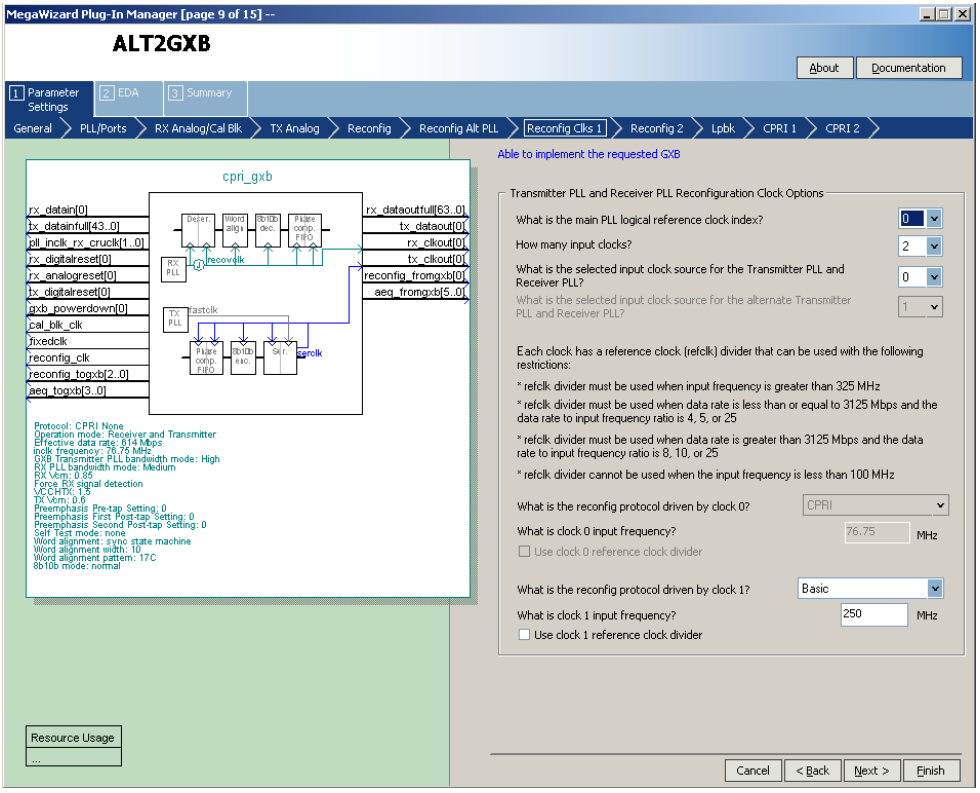


Table 4–62 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–62. MegaWizard Plug-In Manager Options (Page 9 for CPRI Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–78 shows page 10 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode. This page appears only when the **Channel Internals** or **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–78. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

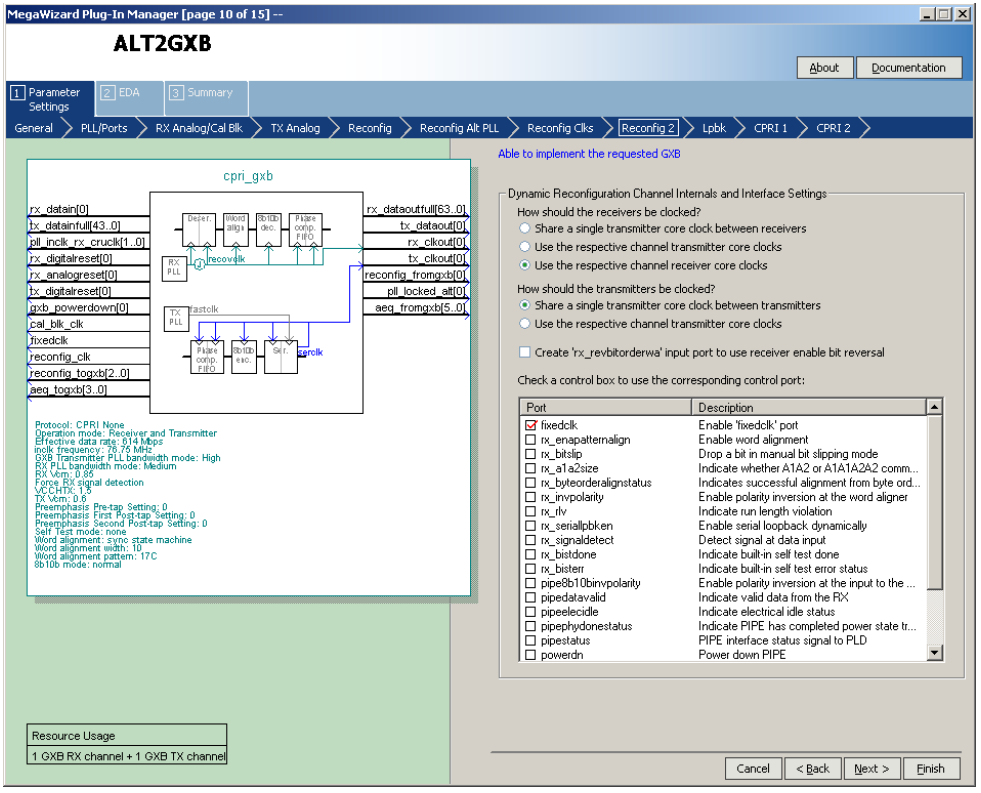


Table 4–63 describes the available options on page 10 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–63. MegaWizard Plug-In Manager Options (Page 10 for CPRI Mode)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–79 shows page 11 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode.

Figure 4–79. MegaWizard Plug-In Manager - ALT2GXB (Loopback)

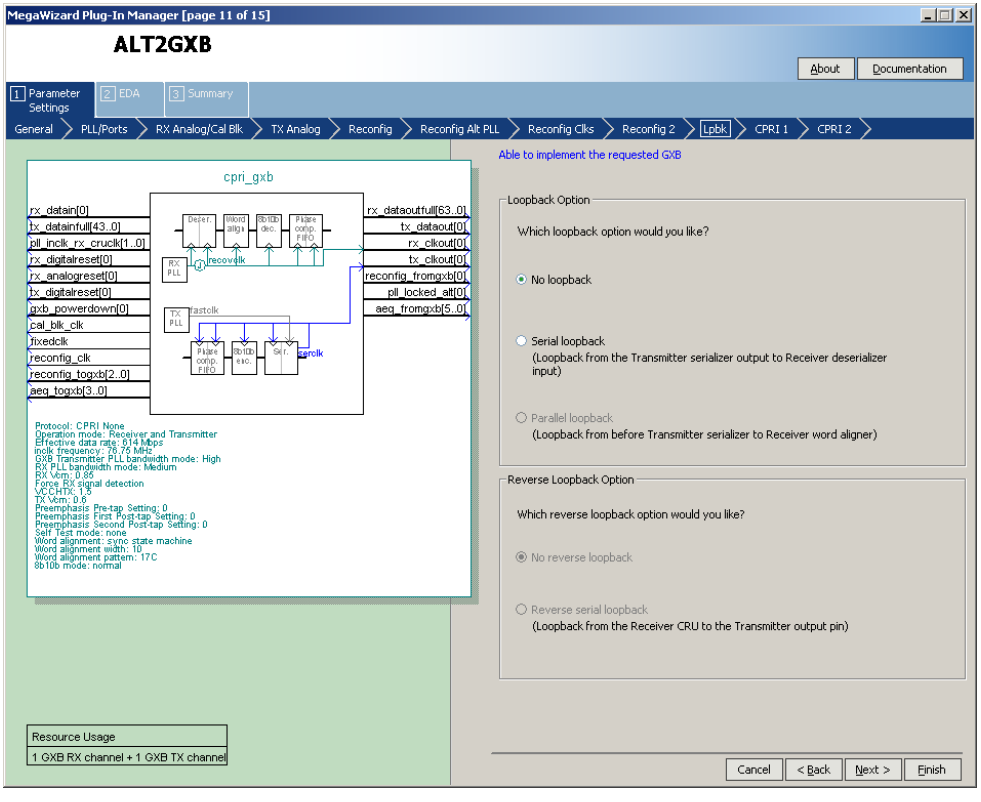


Table 4–64 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–64. MegaWizard Plug-In Manager Options (Page 11 for CPRI Mode)

ALT2GXB Setting	Description	Reference
Which loopback option would you like?	There are two options available in CPRI mode: no loopback and serial loopback. • No loopback - this is the default mode. • Serial loopback - if you select serial loopback, the `rx_seriallpbken` port is available to control the serial loopback feature dynamically. A 1'b1 enables serial loopback and a 1'b0 disables loopback on a channel-by-channel basis. Altera recommends controlling all four channels simultaneously. A digital reset must be asserted for the transceiver.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Reverse Loopback option	This option is not available in CPRI mode.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–80 shows page 12 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode. If the **Enforce default settings for this protocol** option is selected, this page does not appear in the MegaWizard.

Figure 4–80. MegaWizard Plug-In Manager - ALT2GXB (CPRI 1)

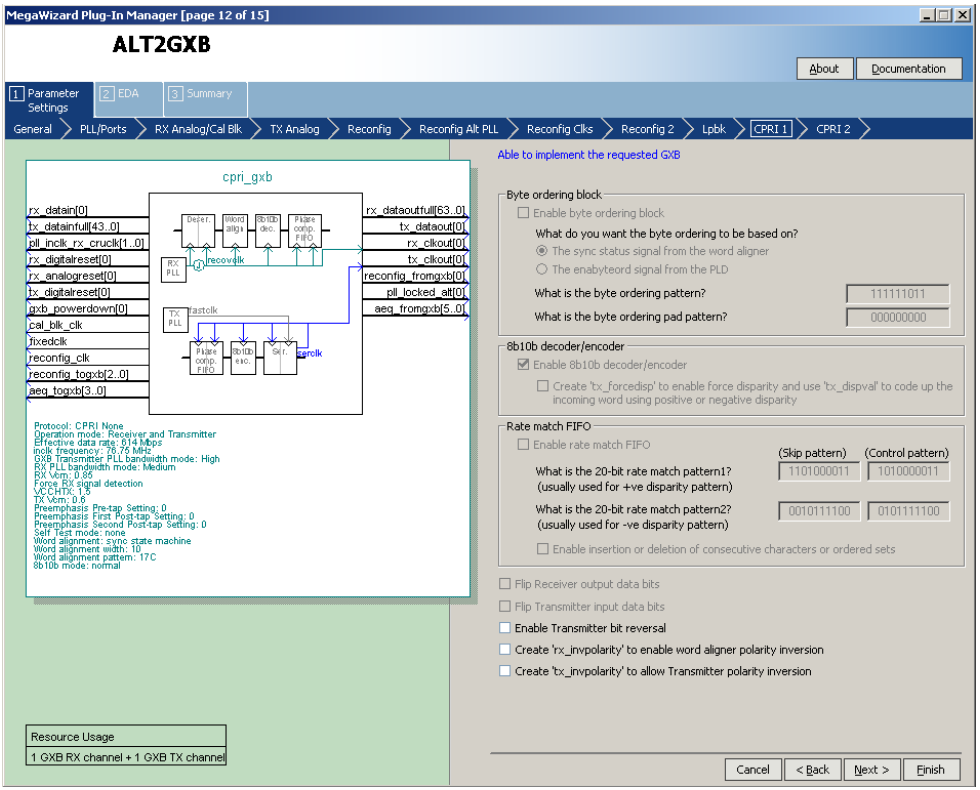


Table 4–65 describes the available options on page 12 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–65. MegaWizard Plug-In Manager Options (Page 12 for CPRI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable byte ordering block	This option is not available in CPRI mode.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable 8B/10B decoder/encoder	This option is forced selected in CPRI mode since 8B/10B decoder/encoder is always used.	8B/10 Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_forcedisp` to enable Force disparity and use `tx_dispval` to code up the incoming word using positive or negative disparity	This option allows you to force positive or negative disparity on transmitted data in 8B/10B configurations.	8B/10 Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable rate match FIFO	This option is not available in CPRI mode since the rate match FIFO is always bypassed.	Rate Matcher section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Flip Receiver output data bits	This option reverses the bit order of the data at the receiver-PLD interface at a byte level to support MSBit to LSBit transmission protocols. The default transmission order is LSBit to MSBit.	
Flip Transmitter input data bits	This option reverses the bit order of the data bits at the input of the transmitter at a byte level to support MSBit to LSBit transmission protocols. The default transmission order is LSBit to MSBit.	

Table 4–65. MegaWizard Plug-In Manager Options (Page 12 for CPRI Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Enable Transmitter bit reversal	This option inverts (flips) the bit order of the data bits at the transmitter PCS-PMA interface at a byte level to support MSBit to LSBit transmission protocols. The default transmission is LSBit to MSBit.	8B/10B Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–81 shows page 13 of the ALT2GXB MegaWizard Plug-In Manager for CPRI mode.

Figure 4–81. MegaWizard Plug-In Manager - ALT2GXB (CPRI 2)

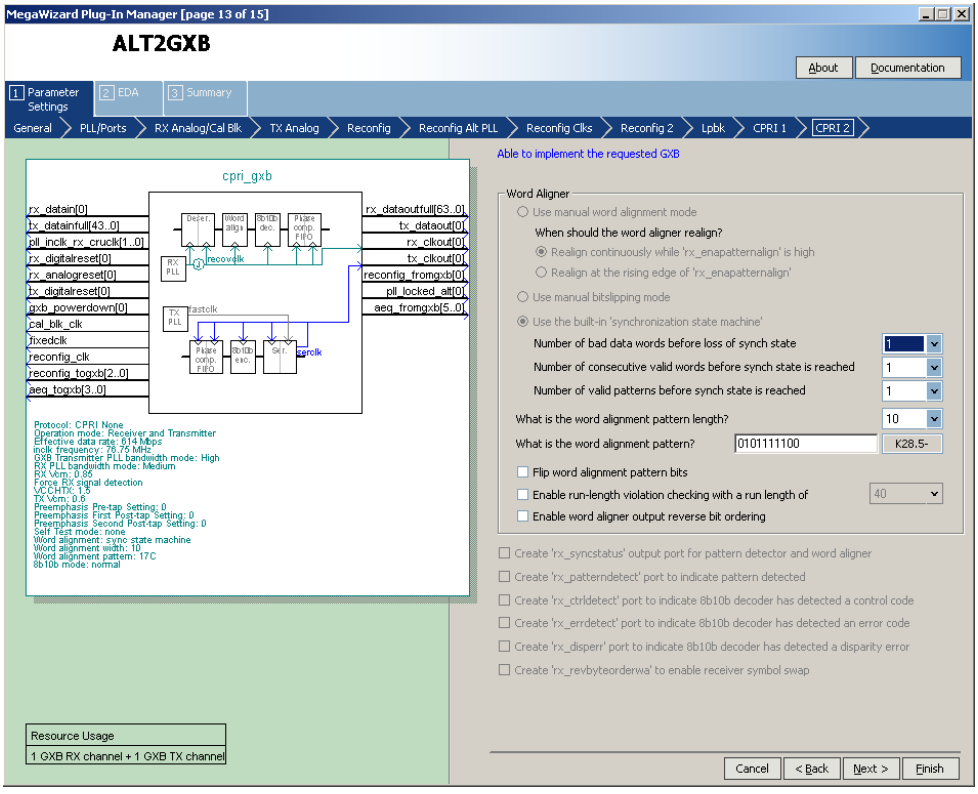


Table 4–66 describes the available options on page 13 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–66. MegaWizard Plug-In Manager Options (Page 13 for CPRI Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Use manual word alignment mode	This option is not available in CPRI mode as the word aligner is synchronization state machine based.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use manual bit slipping mode	This option is not available in CPRI mode as the word aligner is synchronization state machine based.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use the built-in 'synchronization state machine'	This option is forced selected in CPRI mode. This option sets the word aligner to use the built-in synchronization state machine.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Number of bad data words before loss of synch state	Use this option with the built-in state machine to transition from a synchronized state to an unsynchronized state.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Number of consecutive valid words before synch state is reached	This option sets the word aligner to check for a given number of good code groups. Use this option with the built-in synchronization state machine in conjunction with the **Number of valid patterns before synchronization state is reached** option to achieve synchronization.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*
Number of valid patterns before synch state is reached	This option checks for the number of valid alignment patterns seen. Use this option with the built-in synchronization state machine in conjunction with the **Number of consecutive valid words before synch state is reached** option to achieve synchronization.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*

Table 4–66. MegaWizard Plug-In Manager Options (Page 13 for CPRI Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
What is the word alignment pattern length?	This option sets the word alignment length. The available choices are 7 bit and 10 bit.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the word alignment pattern?	Enter the word alignment pattern here. The length of the alignment pattern is based on the word alignment pattern length.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Flip word alignment pattern bits	This option reverses the bit order of the alignment pattern at a byte level to support MSB to LSB transmission protocols. The default transmission order is LSB to MSB.	
Enable run-length violation checking with a run length of	This option activates the run-length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable word aligner output reverse bit ordering	This option statically configures the receiver to reverse the bit order of the data at the output of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–66. MegaWizard Plug-In Manager Options (Page 13 for CPRI Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_patterndetect` port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_ctrldetect` port to indicate 8B/10B decoder has detected a control code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_errdetect` port to indicate 8B/10B decoder has detected an error code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_disperr` port to indicate 8B/10B decoder has detected a disparity code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbyteorderwa` to enable receiver symbol swap	This option is not available in CPRI mode.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–82 shows page 14 of the MegaWizard Plug-In Manager for the CPRI protocol selection. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for third party EDA synthesis tool to be able to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–82. MegaWizard Plug-In Manager - ALT2GXB (EDA)

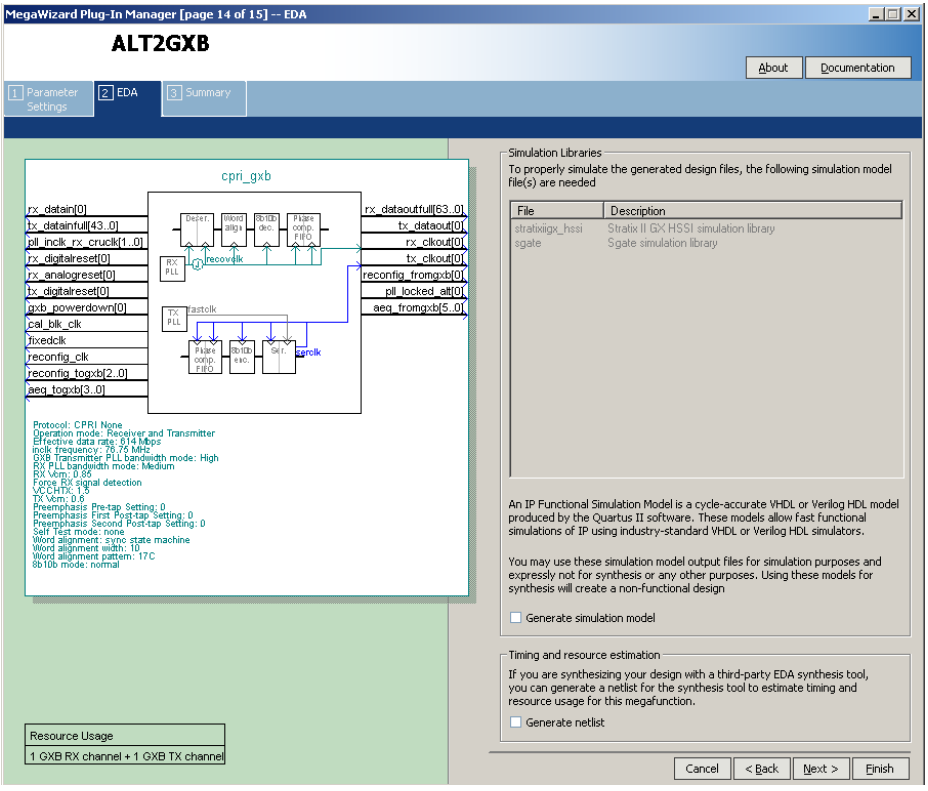
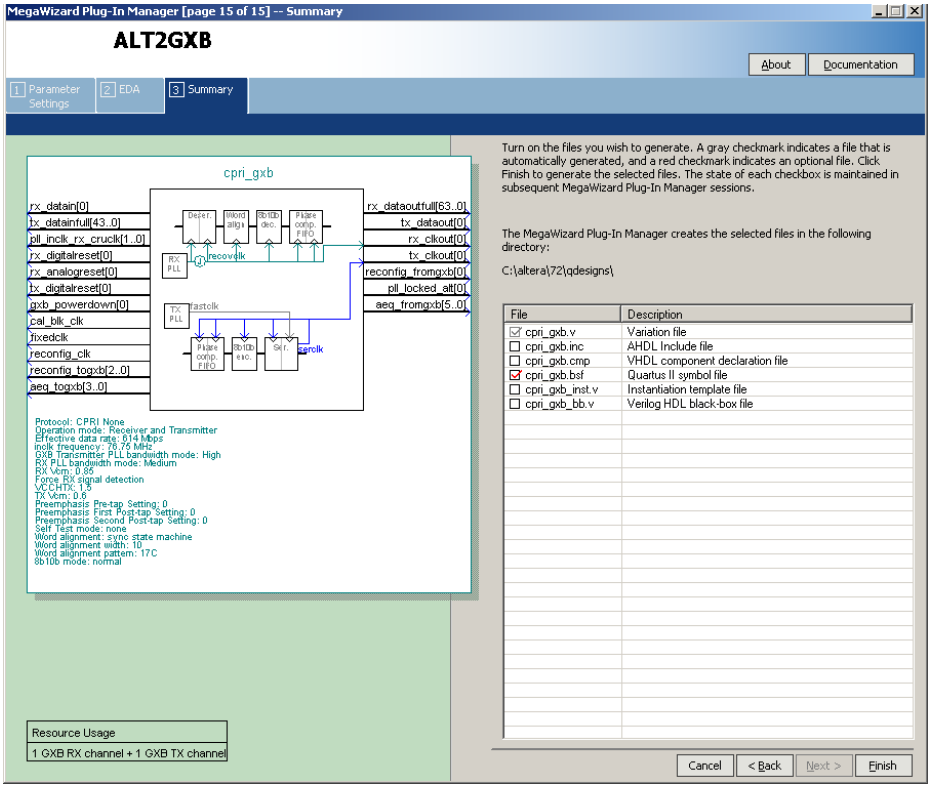


Figure 4–83 shows page 15 (the last page) of the MegaWizard Plug-In Manager for the CPRI protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–83. MegaWizard Plug-In Manager - ALT2GXB (Summary)



SDI Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for the SDI mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

Figure 4–84 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode.

Figure 4–84. MegaWizard Plug-In Manager - ALT2GXB (General)

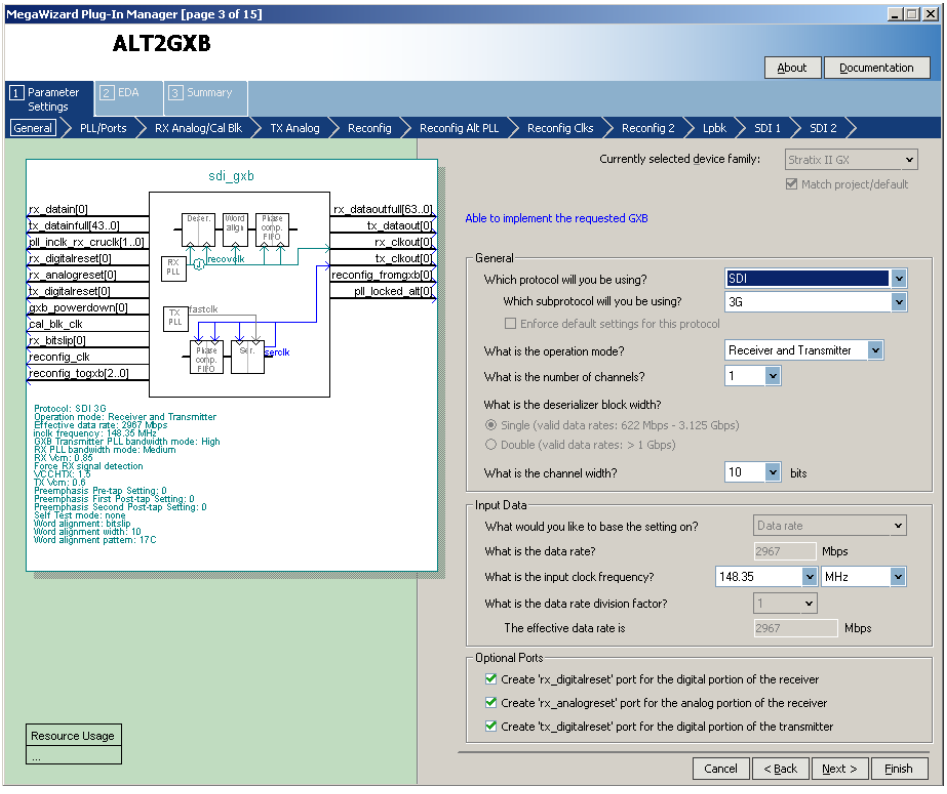


Table 4–67 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–67. MegaWizard Plug-In Manager Options (Page 3 for SDI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Selects the specific protocol or modes under which the transceiver operates. For the SDI mode, you must select the **SDI** protocol.	
Which subprotocol will you be using?	In SDI mode, the two available subprotocols are: • 3G: third-generation (3 Gbps) SDI at 2970 Mbps or 2967 Mbps • HD: high-definition SDI at 1485 Mbps or 1483.5 Mbps	
Enforce default settings for this protocol	This option is not available in SDI mode.	
What is the operation mode?	The transmitter only, receiver only, and receiver and transmitter (full duplex) modes are allowed in SDI protocol.	
What is the number of channels?	This selects how many duplicate channels this ALT2GXB instance contains.	
What is the deserializer block width?	SDI mode only operates in a single-width mode. Double-width mode is not allowed.	
What is the channel width?	This option determines the transceiver to PLD interface width. In SDI mode, 10-bit and 20-bit channel widths are allowed. In 10-bit configuration, the byte serializer is not used. In 20-bit configuration, the byte serializer is used.	Byte Serializer and Byte Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option not available in SDI mode.	
What is the data rate?	This field is automatically set based on the subprotocol (3G or HD) and the input clock frequency selection.	
What is the input clock frequency?	Four input reference clock options are available, depending on the subprotocol (3G or HD). • For 3G subprotocol the available options are: 148.5 MHz and 297 MHz for 2970 Mbps data rate and 148.35 MHz and 296.7 MHz for 2967 Mbps data rate • For HD subprotocol the available option are: 74.25 MHz and 148.5 MHz for 1485 Mbps data rate and 74.175 MHz and 148.35 MHz for 1483.5 Mbps data rate	

Table 4–67. MegaWizard Plug-In Manager Options (Page 3 for SDI Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the data rate division factor?	This option is not available in SDI Mode.	
Create `rx_digitalreset` port for the digital portion of the receiver	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_analogreset` port for the analog portion of the receiver	Receiver analog reset port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_digitalreset` port for the digital portion of the receiver	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–85 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode.

Figure 4–85. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

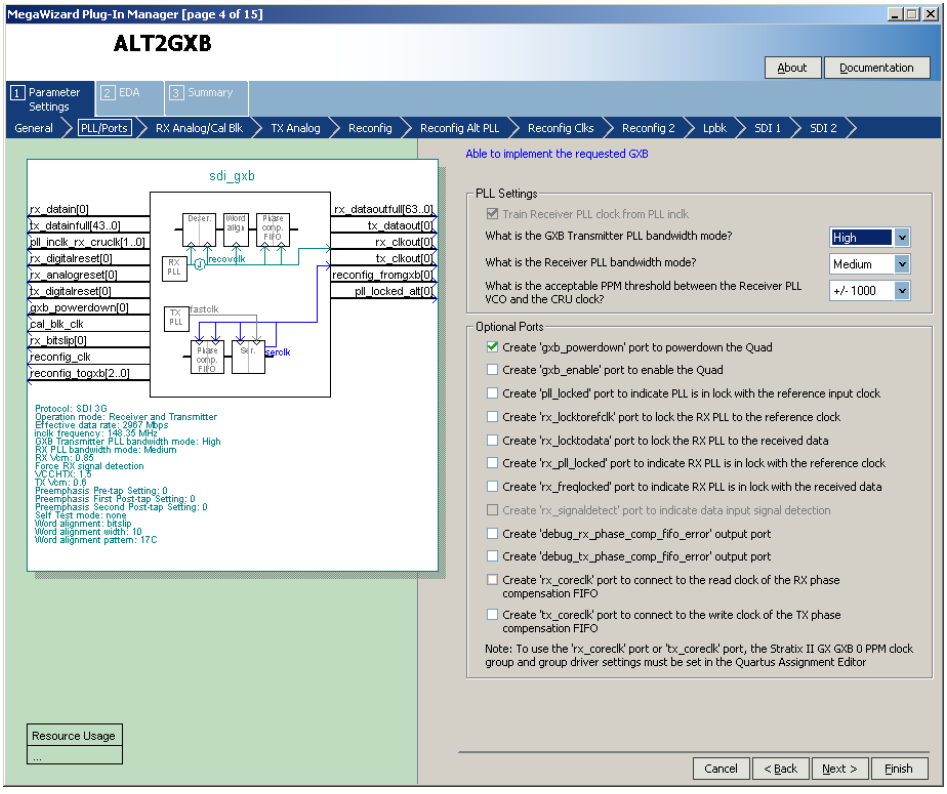


Table 4–68 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–68. MegaWizard Plug-In Manager Options (Page 4 for SDI Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If **CMU PLL reconfiguration** is enabled, this option is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	Three available bandwidth options are high, medium and low. The default transmitter PLL bandwidth is high.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver PLL bandwidth mode?	Three available bandwidth options are high, medium and low. The default receiver PLL bandwidth is medium.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver CRU switchover between lock-to-data and lock-to-reference. (There are additional factors that affect CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–68. MegaWizard Plug-In Manager Options (Page 4 for SDI Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicate data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–68. MegaWizard Plug-In Manager Options (Page 4 for SDI Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–86 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode.

Figure 4–86. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

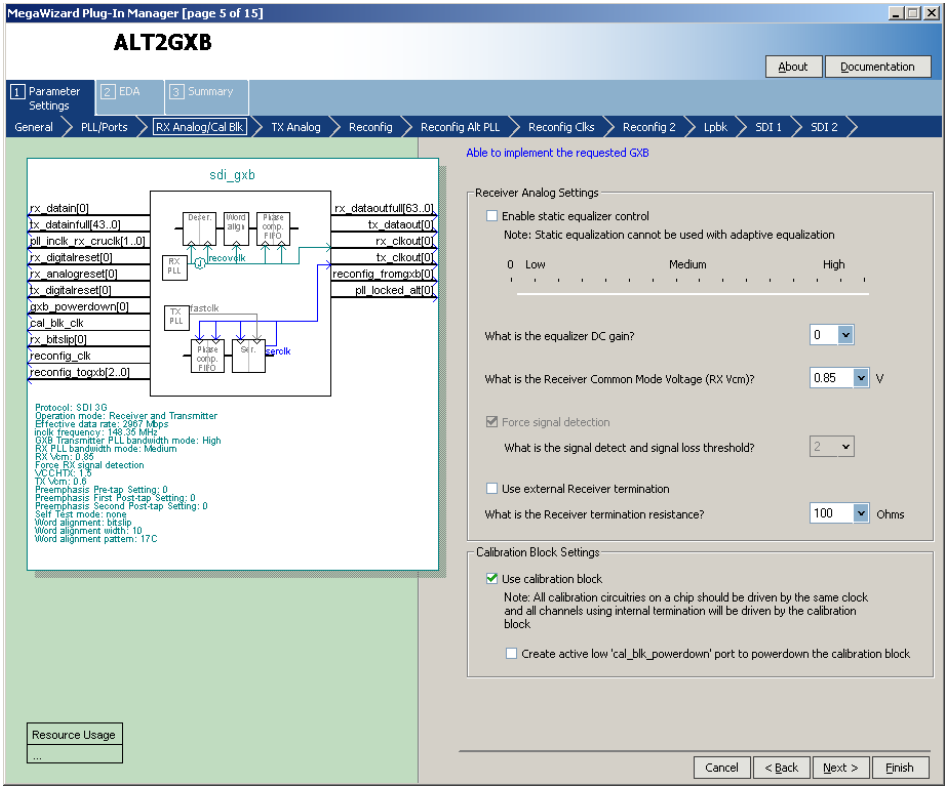


Table 4–69 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–69. MegaWizard Plug-In Manager Options (Page 5 for SDI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is set to 0.85 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Force signal detection	This option if force-selected in SDI mode.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	This option is not available in SDI mode.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–69. MegaWizard Plug-In Manager Options (Page 5 for SDI Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–87 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode.

Figure 4–87. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

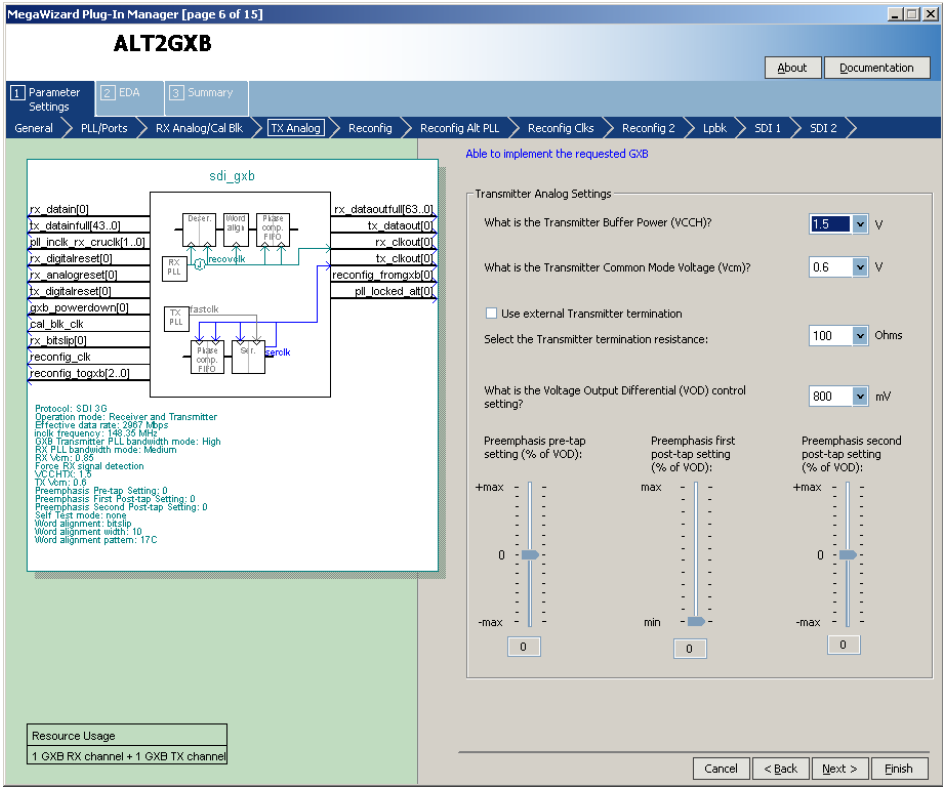


Table 4–70 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–70. MegaWizard Plug-In Manager Options (Page 6 for SDI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. The selections available are 1.2 V and 1.5 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Restrictions apply based on the V_{CCH} setting.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap. The amount of pre-emphasis is to be determined by characterization.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–70. MegaWizard Plug-In Manager Options (Page 6 for SDI Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap. The amount of pre-emphasis is to be determined by characterization.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap. The amount of pre-emphasis is to be determined by characterization.	

Figure 4–75 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode.

Figure 4–88. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

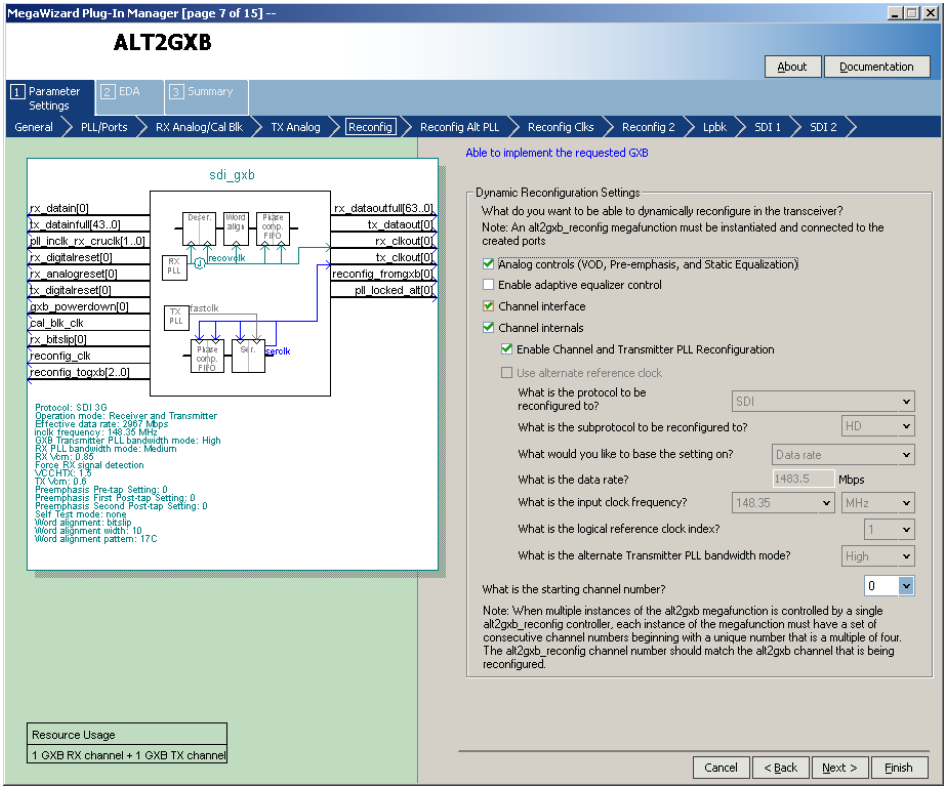


Table 4–71 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–71. MegaWizard Plug-In Manager Options (Page 7 for SDI Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are: • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel Interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals. • Channel Internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–89 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode. This page appears only when the **Channel Internals** or **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–89. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

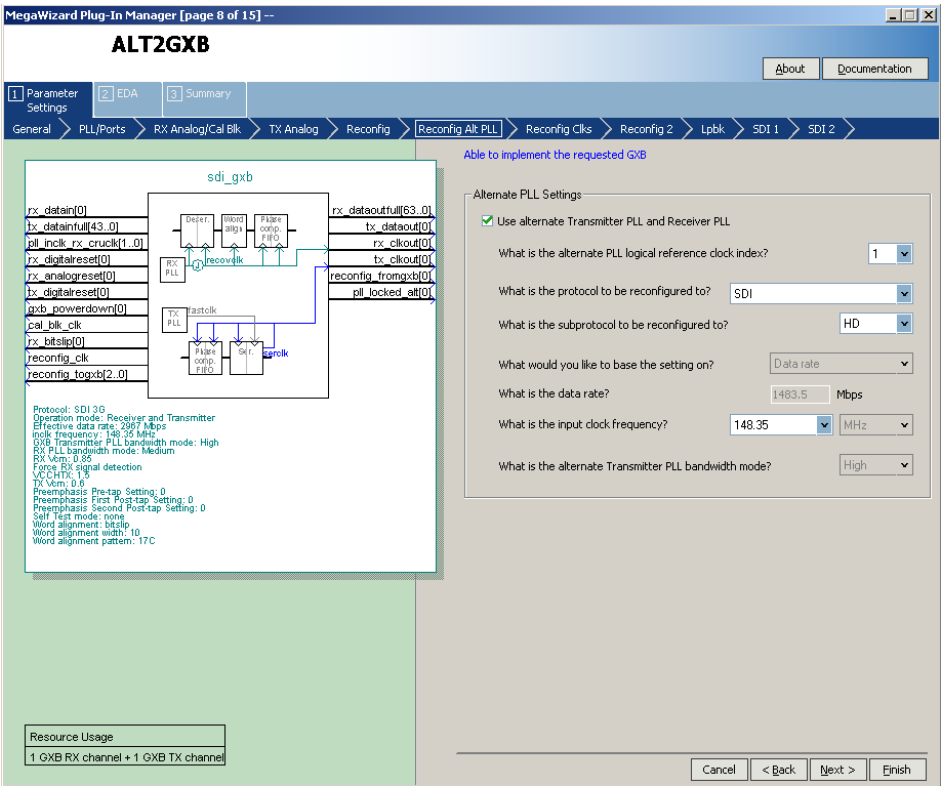


Table 4–72 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–72. MegaWizard Plug-In Manager Options (Page 8 for SDI Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–90 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode. This page appears only if the **Channel Internals** and **Enable Channel and Transmitter PLL Reconfiguration** options are selected on the **Reconfig** page (Page 7).

Figure 4–90. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

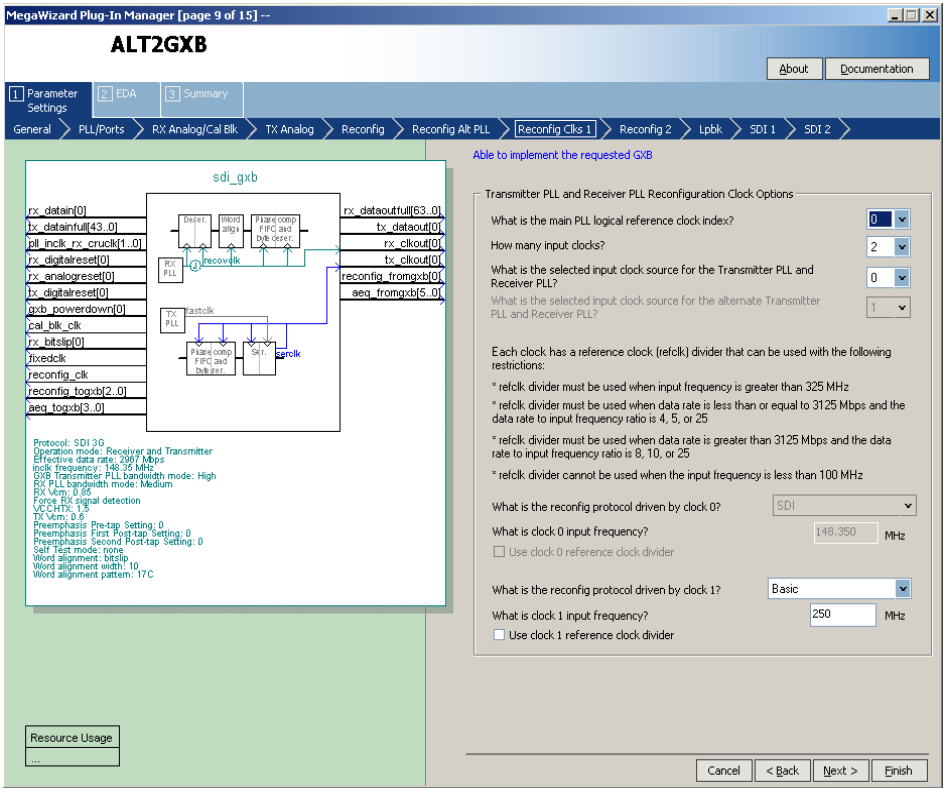


Table 4–73 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–73. MegaWizard Plug-In Manager Options (Page 9 for SDI Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–91 shows page 10 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode. This page appears only when the **Channel Internals** or **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–91. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

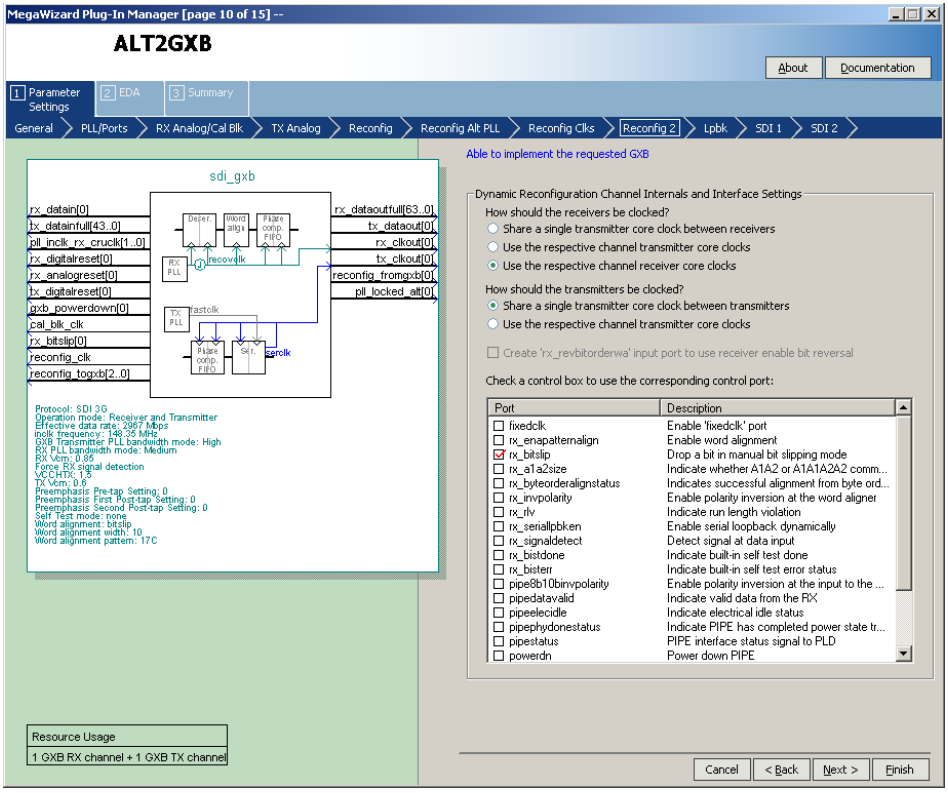


Table 4–74 describes the available options on page 10 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–74. MegaWizard Plug-In Manager Options (Page 10 for SDI Mode)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–92 shows page 11 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode.

Figure 4–92. MegaWizard Plug-In Manager - ALT2GXB (Loopback)

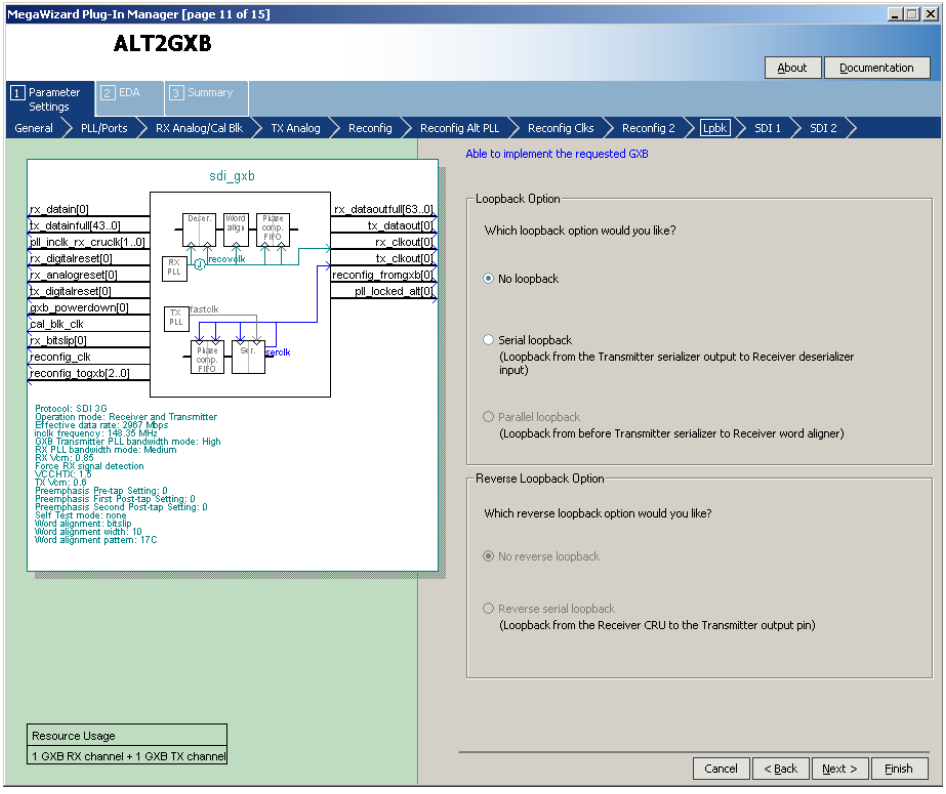


Table 4–75 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–75. MegaWizard Plug-In Manager Options (Page 11 for SDI Mode)

ALT2GXB Setting	Description	Reference
Which loopback option would you like?	There are two options available in SDI mode: no loopback and serial loopback. • No loopback - this is the default mode. • Serial loopback - if you select serial loopback, the `rx_seriallpbken` port is available to control the serial loopback feature dynamically. A 1'b1 enables serial loopback and a 1'b0 disables loopback on a channel-by-channel basis. Altera recommends controlling all four channels simultaneously. A digital reset must be asserted for the transceiver.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Reverse Loopback option	This option is not available in SDI mode.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–93 shows page 12 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode. If the **Enforce default settings for this protocol** option is selected, this page does not appear in the MegaWizard.

Figure 4–93. MegaWizard Plug-In Manager - ALT2GXB (SDI 1)

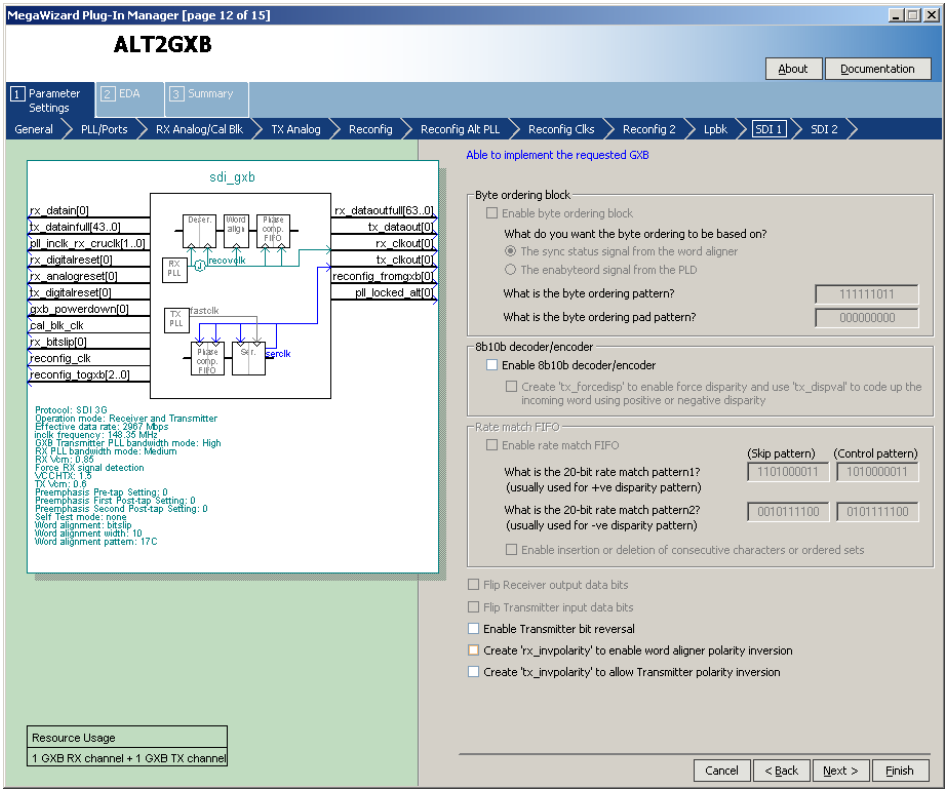


Table 4–74 describes the available options on page 12 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–76. MegaWizard Plug-In Manager Options (Page 12 for SDI Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable byte ordering block	This option is not available in SDI mode.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable 8B/10B decoder/encoder	This option is force-selected in SDI mode since 8B/10B decoder/encoder is always used.	8B/10 Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_forcedisp` to enable Force disparity and use `tx_dispval` to code up the incoming word using positive or negative disparity	This option allows you to force positive or negative disparity on transmitted data in 8B/10B configurations.	8B/10 Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable rate match FIFO	This option is not available in SDI mode since the rate match FIFO is always bypassed.	Rate Matcher section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Flip Receiver output data bits	This option reverses the bit order of the data at the receiver-PLD interface at a byte level to support MSBit to LSBit transmission protocols. The default transmission order is LSBit to MSBit.	
Flip Transmitter input data bits	This option reverses the bit order of the data bits at the input of the transmitter at a byte level to support MSBit to LSBit transmission protocols. The default transmission order is LSBit to MSBit.	

Table 4–76. MegaWizard Plug-In Manager Options (Page 12 for SDI Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Enable Transmitter bit reversal	This option inverts (flips) the bit order of the data bits at the transmitter PCS-PMA interface at a byte level to support MSBit to LSBit transmission protocols. The default transmission is LSBit to MSBit.	8B/10B Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–94 shows page 13 of the ALT2GXB MegaWizard Plug-In Manager for SDI mode.

Figure 4–94. MegaWizard Plug-In Manager - ALT2GXB (SDI 2)

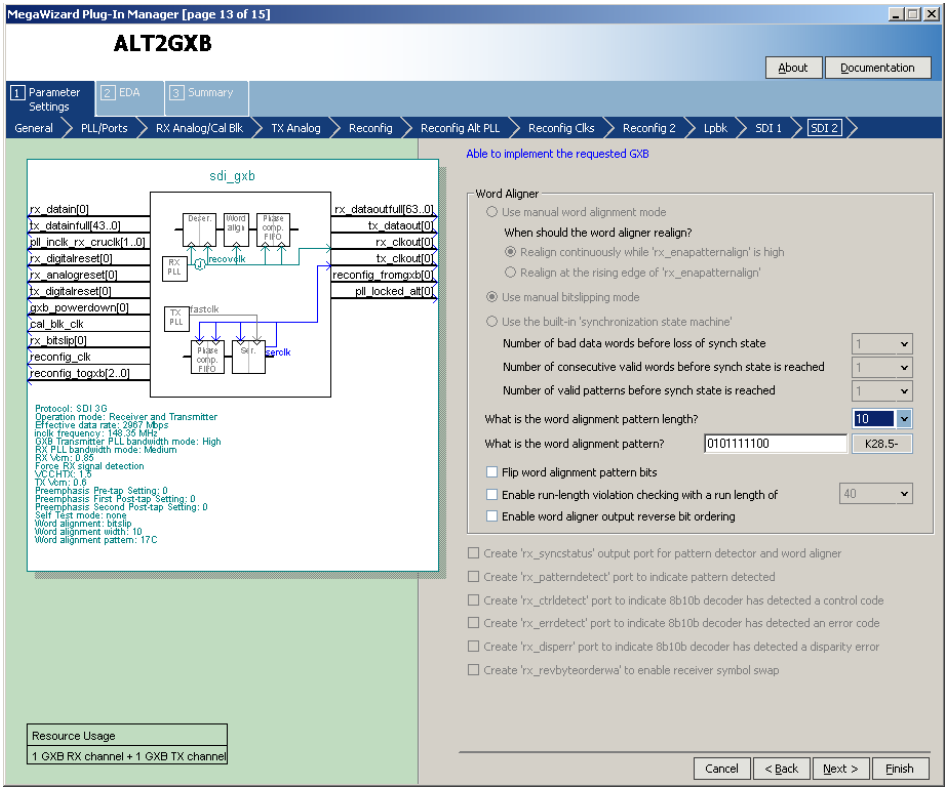


Table 4–77 describes the available options on page 13 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–77. MegaWizard Plug-In Manager Options (Page 13 for SDI Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Use manual word alignment mode	This option is not available in SDI mode as the word aligner is synchronization state machine based.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use manual bit slipping mode	This option is not available in SDI mode as the word aligner is synchronization state machine based.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use the built-in 'synchronization state machine'	This option is force-selected in SDI mode. This option sets the word aligner to use the built-in synchronization state machine.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Number of bad data words before loss of synch state	Use this option with the built-in synchronization state machine to transition from a synchronized state to an unsynchronized state.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Number of consecutive valid words before synch state is reached	This option sets the word aligner to check for a given number of good code groups. Use this option with the built-in synchronization state machine in conjunction with the **Number of valid patterns before synchronization state is reached** option to achieve synchronization.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*
Number of valid patterns before synch state is reached	This option checks for the number of valid alignment patterns seen. Use this option with the built-in synchronization state machine in conjunction with the **Number of consecutive valid words before synch state is reached** option to achieve synchronization.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*

Table 4–77. MegaWizard Plug-In Manager Options (Page 13 for SDI Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
What is the word alignment pattern length?	This option sets the word alignment length. The available choices are 7 bit and 10 bit.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the word alignment pattern?	Enter the word alignment pattern here. The length of the alignment pattern is based on the word alignment pattern length.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Flip word alignment pattern bits	This option reverses the bit order of the alignment pattern at a byte level to support MSB to LSB transmission protocols. The default transmission order is LSB to MSB.	
Enable run-length violation checking with a run length of	This option activates the run-length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Enable word aligner output reverse bit ordering	This option statically configures the receiver to reverse the bit order of the data at the output of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–77. MegaWizard Plug-In Manager Options (Page 13 for SDI Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_patterndetect` port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_ctrldetect` port to indicate 8B/10B decoder has detected a control code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_errdetect` port to indicate 8B/10B decoder has detected an error code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_disperr` port to indicate 8B/10B decoder has detected a disparity code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the Stratix II GX Device Handbook for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_revbyteorderwa` to enable receiver symbol swap	This option is not available in SDI mode.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–82 shows page 14 of the MegaWizard Plug-In Manager for the SDI protocol selection. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for third party EDA synthesis tool to be able to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–95. MegaWizard Plug-In Manager - ALT2GXB (EDA)

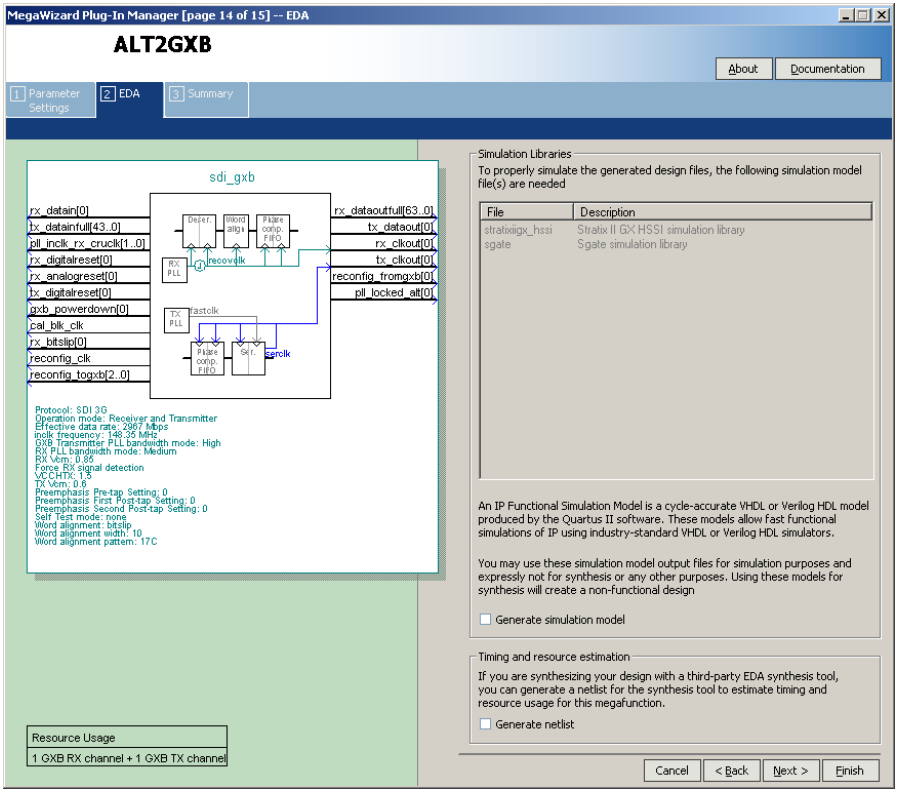
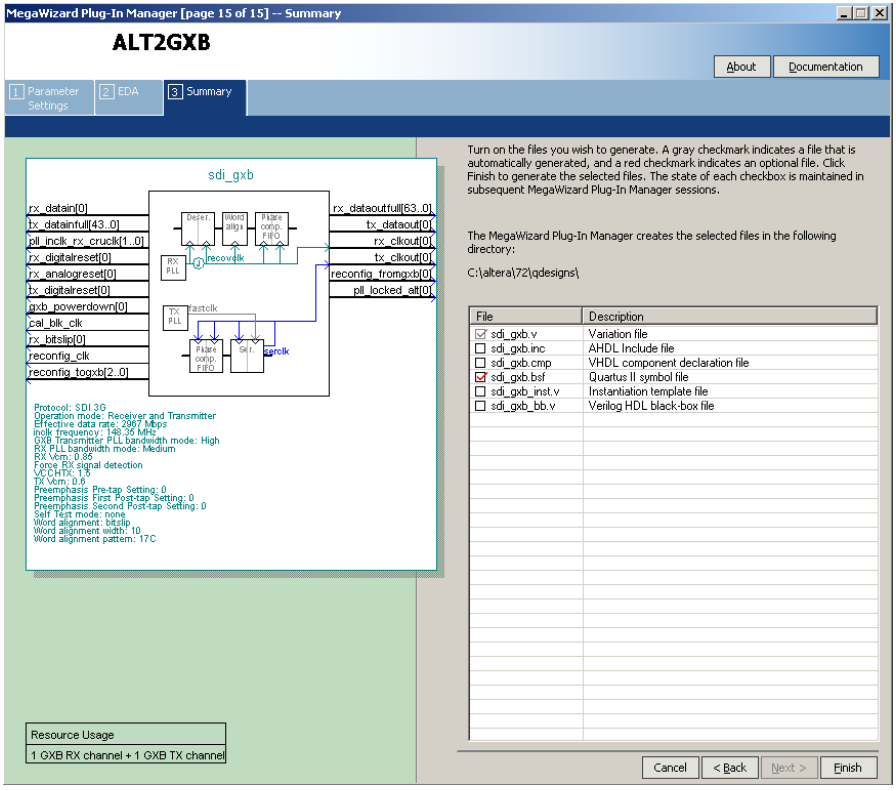


Figure 4–83 shows page 15 (the last page) of the MegaWizard Plug-In Manager for the SDI protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–96. MegaWizard Plug-In Manager - ALT2GXB (Summary)



Serial RapidIO Mode

This section provides descriptions of the options available on the individual pages of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode. The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

Figure 4–97 shows page 3 of the ALT2GXB MegaWizard Plug-In Manager in Serial RapidIO mode.

Figure 4–97. MegaWizard Plug-In Manager - ALT2GXB (General)

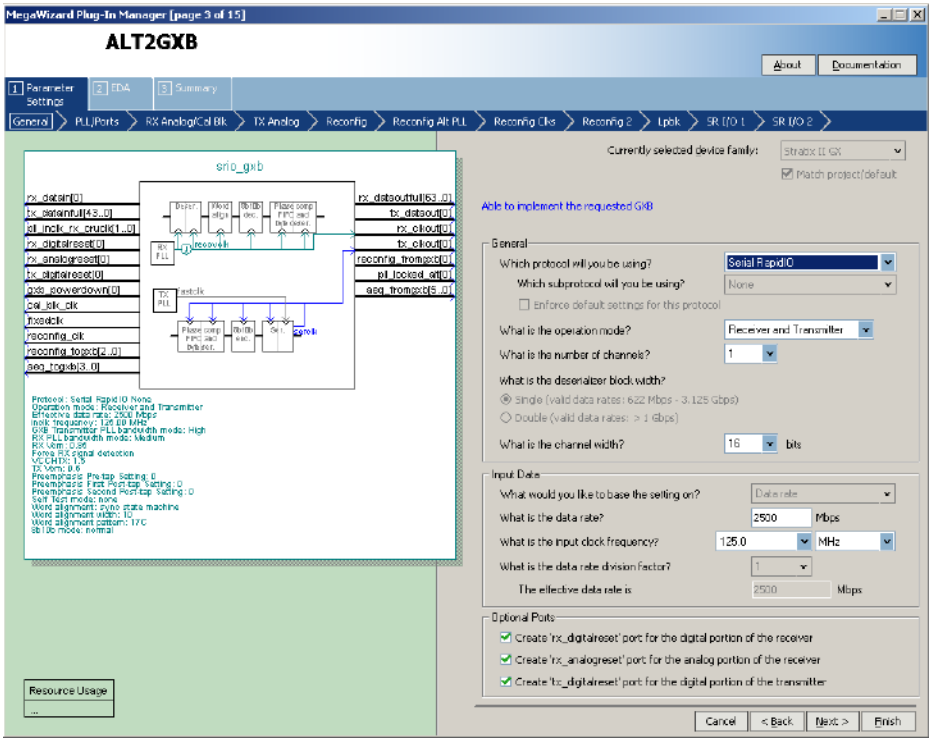


Table 4–78 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–78. MegaWizard Plug-In Manager Options (Page 3 for Serial RapidIO Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Which protocol will you be using?	Determines the specific protocol or modes under which the transceiver operates. For Serial RapidIO mode, you must select the **Serial RapidIO** protocol.	
Which subprotocol will you be using?	This option is not available in Serial RapidIO mode.	
Enforce default settings for this protocol	This option is not available in Serial RapidIO mode.	

Table 4–78. MegaWizard Plug-In Manager Options (Page 3 for Serial RapidIO Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
What is the operation mode?	The available operation modes are receiver only, transmitter only, and receiver and transmitter.	
What is the number of channels?	This option determines how many duplicate channels this ALT2GXB instance contains.	
What is the deserializer block width?	Serial RapidIO mode only operates in a single-width mode. Double-width mode is not allowed.	
What is the channel width?	The channel width is fixed to16 in Serial RapidIO mode.	Byte Serializer and Deserializer sections in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What would you like to base the setting on?	This option is not available in Serial RapidIO mode.	
What is the data rate?	Enter **1250** Mbps, **2500** Mbps, or **3125** Mbps depending on your design requirements.	
What is the input clock frequency?	Determines the input clock frequency you want as a reference clock for the transceiver.	
What is the data rate division factor?	This option is not available in Serial RapidIO mode.	
Create `rx_digitalreset` port	Receiver digital reset port. Resets the PCS portion of the receiver. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_analogreset` port	Receiver analog reset port.	Reset Control and Power Down" section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `tx_digitalreset` port	Transmitter digital reset port. Resets the PCS portion of the transmitter. Altera recommends using this port along with logic to implement the recommended reset sequence.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–98 shows page 4 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode.

Figure 4–98. MegaWizard Plug-In Manager - ALT2GXB (PLL/Ports)

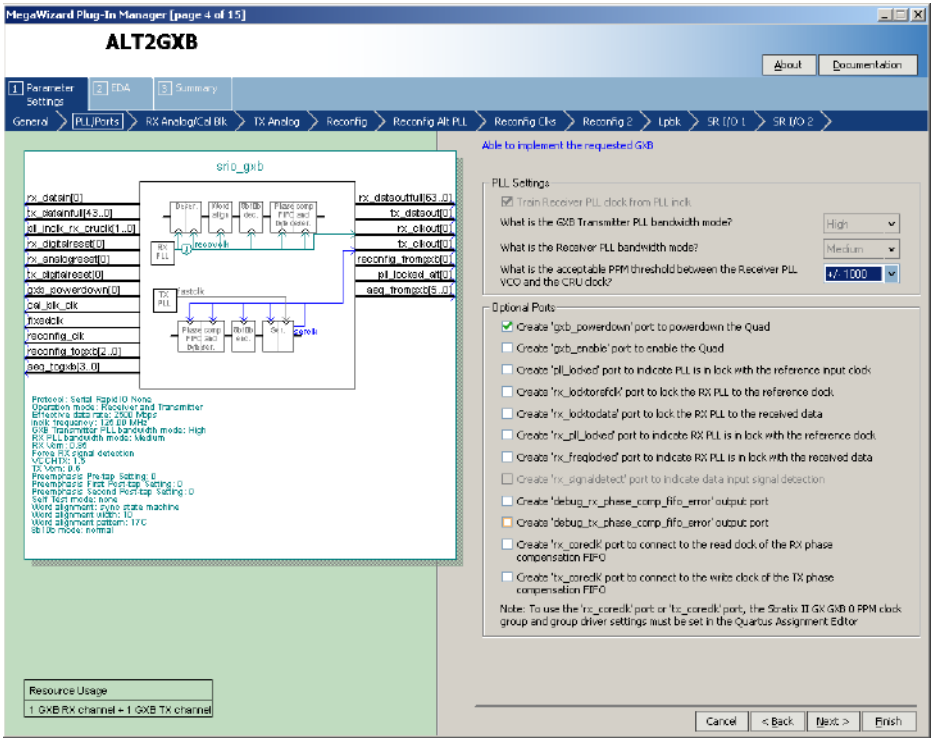


Table 4–79 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–79. MegaWizard Plug-In Manager Options (Page 4 for Serial RapidIO Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Train Receiver PLL clock from PLL inclk	If you turn this option on, your design uses the input reference clock to the transmitter PLL to train the receiver PLL. This reduces the need to supply a separate receiver PLL reference clock. If **CMU PLL reconfiguration** is enabled, this option is automatically enabled by the Megawizard Plug-In Manager.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the GXB Transmitter PLL bandwidth mode?	The Quartus II software automatically selects the high bandwidth setting in Serial RapidIO mode.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver PLL bandwidth mode?	The Quartus II software automatically selects the medium bandwidth setting in Serial RapidIO mode.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock?	This option determines the PPM difference that affects the automatic receiver clock recovery unit (CRU) switchover between lock-to-data and lock-to-reference. (There are additional factors that affect the CRU's transition.)	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_powerdown` port to power down the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `gxb_enable` port to enable the Quad	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Reset Control and Power Down section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–79. MegaWizard Plug-In Manager Options (Page 4 for Serial RapidIO Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
Create `pll_locked` port to indicate PLL is in lock with the reference input clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Multiplier Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktorefclk` port to lock the RX PLL to the reference clock	Refer to the *Stratix II GX Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_locktodata` port to lock the RX PLL to the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_pll_locked` port to indicate RX PLL is in lock with the reference clock	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_freqlocked` port to indicate RX PLL is in lock with the received data	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Clock Recovery Unit section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Create `rx_signaldetect` port to indicated data input signal detection	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–79. MegaWizard Plug-In Manager Options (Page 4 for Serial RapidIO Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `debug_rx_phase_comp_fifo_error` output port	This optional output port indicates Receiver Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Receiver Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `debug_tx_phase_comp_fifo_error` output port	This optional output port indicates Transmitter Phase Compensation FIFO overflow/underrun condition. Note that no PPM difference is allowed between FIFO read and write clocks. This port should be used for debug purpose only.	Transmitter Phase Compensation FIFO section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_coreclk` port to connect to the read clock of the RX phase compensation FIFO	This optional input port allows you to clock the read side of the Receiver Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_coreclk` port to connect to the write clock of the TX phase compensation FIFO	This optional input port allows you to clock the write side of the Transmitter Phase Compensation FIFO with a non-transceiver PLD clock.	Transceiver Clocking section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–99 shows page 5 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode.

Figure 4–99. MegaWizard Plug-In Manager - ALT2GXB (RX Analog/Cal Blk)

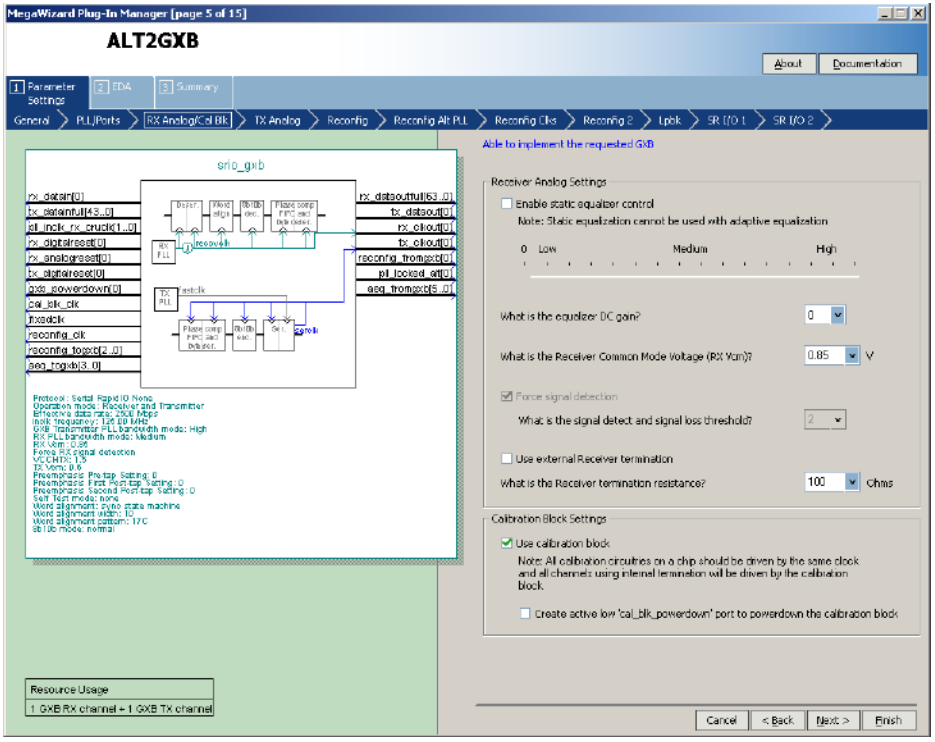


Table 4–80 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–80. MegaWizard Plug-In Manager Options (Page 5 for Serial RapidIO Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
Enable static equalizer control	This option enables the static equalizer settings. If the equalizer settings are not changed through the dynamic reconfiguration controller, the equalizer remains configured to these static settings. Enabling the equalizer control enables the equalizer DC gain option. This DC gain option can be used in conjunction with equalizer controls and has three legal settings.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the Receiver Common Mode Voltage (RX V_{CM})?	The receiver common mode voltage is programmable between 0.85 V and 1.2 V.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Force signal detection	This option is available only in PIPE mode.	Receiver Buffer Section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the signal detect and signal loss threshold?	Use this option when the forced signal detection option is off and to set the trip point of the signal detect circuit. This option is available only in PIPE mode.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Use external receiver termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). If checked, this option turns off the receiver OCT.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
What is the receiver termination resistance?	This option selects the receiver termination value. The settings allowed are 100 Ω, 120 Ω, and 150 Ω.	Receiver Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Table 4–80. MegaWizard Plug-In Manager Options (Page 5 for Serial RapidIO Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Use calibration block	This option allows you to select which instance of ALT2GXB instantiates the calibration block. Only one instance of ALT2GXB is required to instantiate the calibration block.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create active low `cal_blk_powerdown` to power down the calibration block	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Calibration Blocks section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–100 shows page 6 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode.

Figure 4–100. MegaWizard Plug-In Manager - ALT2GXB (TX Analog)

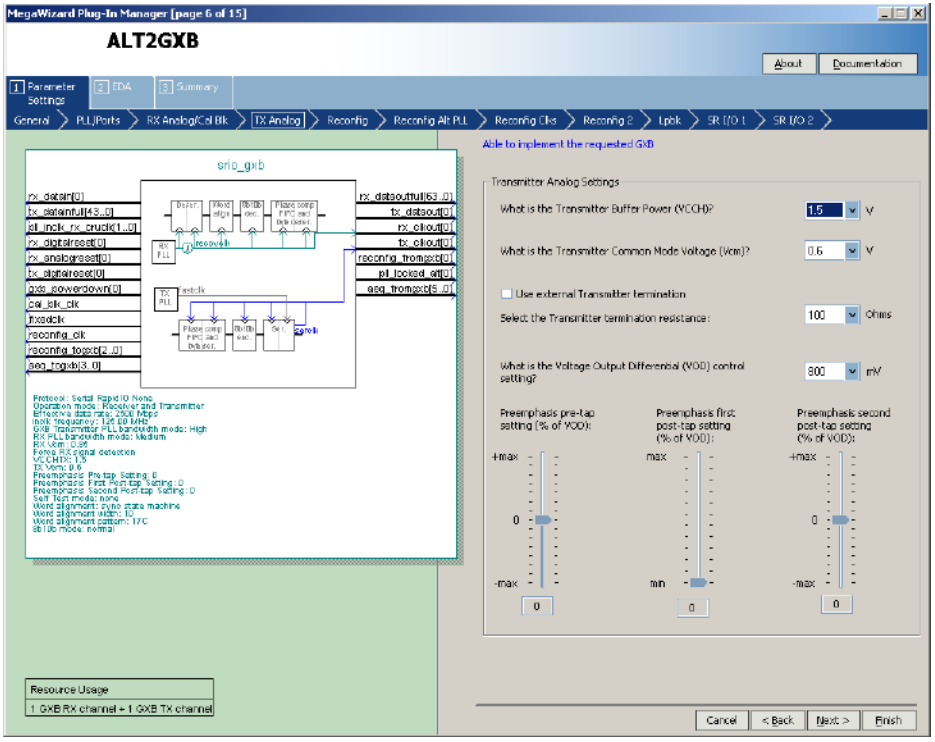


Table 4–81 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–81. MegaWizard Plug-In Manager Options (Page 6 for Serial RapidIO Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
What is the Transmitter Buffer Power (V_{CCH})?	This setting is for information only and is used to calculate the V_{OD} from the buffer power supply (V_{CCH}) and the transmitter termination to derive the proper V_{OD} range. The selections available are 1.2 V and 1.5 V.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Transmitter Common Mode Voltage (V_{CM})?	The transmitter common mode voltage setting is between 0.7 V and 0.6 V. Restrictions apply based on the V_{CCH} setting.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use external Transmitter termination	This option is available if you want to use an external termination resistor instead of the on-chip termination (OCT). Checking this option turns off the transmitter OCT.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Select the Transmitter termination resistance	This option selects the transmitter termination value. This option is also used in the calculation of the available V_{OD}.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the Voltage Output Differential (V_{OD}) control setting?	This option selects the V_{OD} of the transmitter buffer. The available V_{OD} settings change based on V_{CCH} and the transmitter termination resistance value.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Pre-emphasis pre-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using pre-tap.	Transmitter Buffer section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–81. MegaWizard Plug-In Manager Options (Page 6 for Serial RapidIO Mode) (Part 2 of 2)		
ALT2GXB Setting	**Description**	**Reference**
Pre-emphasis first post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using first post-tap.	
Pre-emphasis second post-tap setting (% of V_{OD})	This option sets the amount of pre-emphasis on the transmitter buffer using second post-tap.	

Figure 4–101 shows page 7 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode.

Figure 4–101. MegaWizard Plug-In Manager - ALT2GXB (Reconfig)

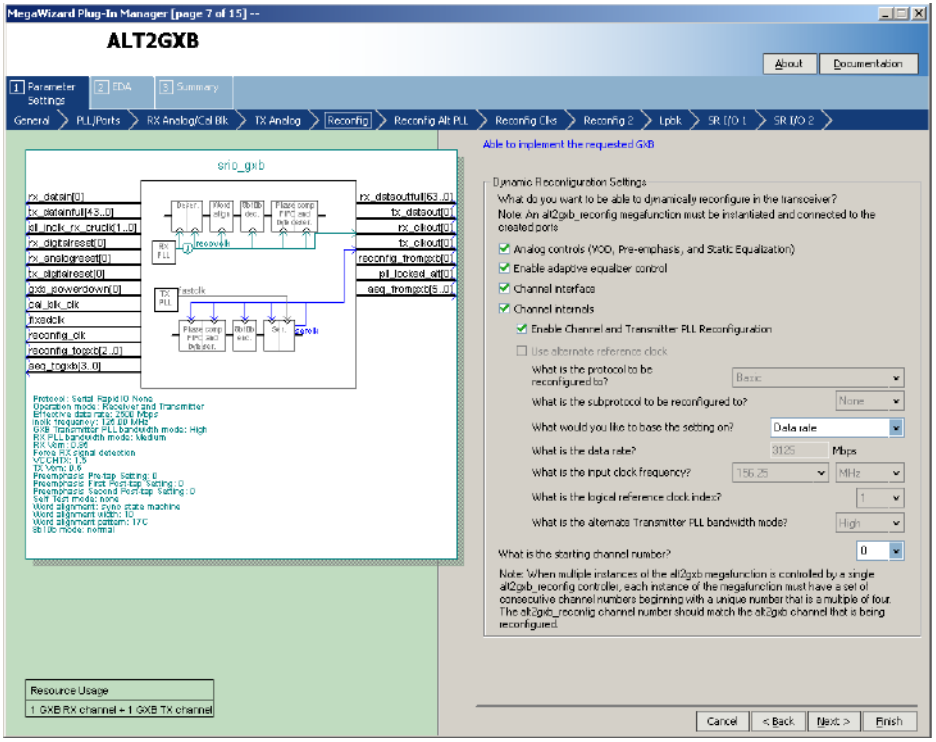


Table 4–82 describes the available options on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–82. MegaWizard Plug-In Manager Options (Page 7 for Serial RapidIO Mode)

ALT2GXB Setting	Description	Reference
What do you want to be able to dynamically reconfigure in the transceiver?	Available options are: • Analog controls: Dynamically reconfigures the PMA control settings like Vod, Pre-emphasis, Equalization, etc. • Enable adaptive equalizer control: Dynamically enables adaptive equalization for the selected receiver channel. • Channel interface: Enables MIF-based reconfiguration among modes that have different PLD interface signals. • Channel internals: Enables MIF-based reconfiguration among modes that have different data paths within the channel but have the same PLD interface signals. When this option is enabled, two mutually exclusive options, **Enable Channel and Transmitter PLL Reconfiguration** and **Use alternate reference clock**, are available.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the starting channel number?	The range for the dynamic reconfiguration starting channel number setting is 0—156, in multiples of 4. It is in multiples of 4 because the dynamic reconfiguration interface is per transceiver block. The range of 0—156 is the logical channel address, based purely on the number of possible ALT2GXB instances.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–102 shows page 8 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** option is selected in the **Reconfig** page (Page 7).

Figure 4–102. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Alt PLL)

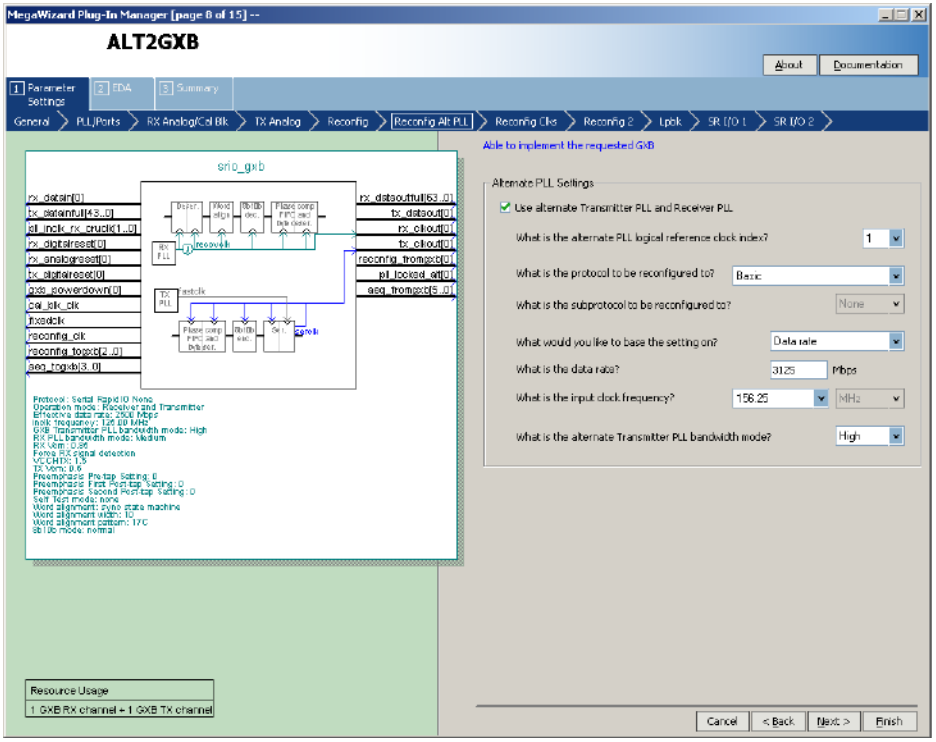


Table 4–83 describes the available options on page 8 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–83. MegaWizard Plug-In Manager Options (Page 8 for Serial RapidIO Mode)

ALT2GXB Setting	Description	Reference
Use alternate Transmitter PLL and Receiver PLL	Selecting this option sets up the transmitter channel to listen to one of the two PLLs in its transceiver block. The information regarding which PLL it listens to is stored in the MIF.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–103 shows page 9 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode. This page appears only if the **Channel Internals** and the **Enable Channel and Transmitter PLL Reconfiguration** options are selected in the **Reconfig** page (Page 7).

Figure 4–103. MegaWizard Plug-In Manager - ALT2GXB (Reconfig Clks 1)

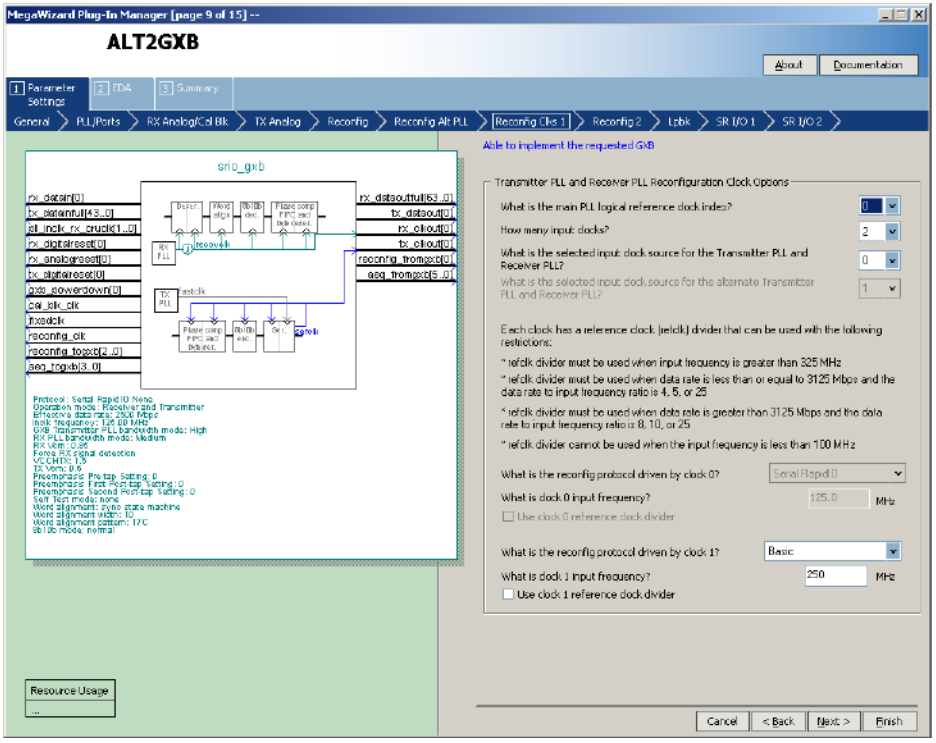


Table 4–84 describes the available options on page 9 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–84. MegaWizard Plug-In Manager Options (Page 9 for Serial RapidIO Mode)

ALT2GXB Setting	Description	Reference
What is the main PLL logical reference clock index?	This option allows you to select the logical index for the PLL that you intend to use with the current configuration. This option is meaningful only if you select the **Use alternate Transmitter PLL and Receiver PLL** option on the **Reconfig Alt PLL** page.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How many input clocks?	This field allows you to select the number of reference clock inputs needed to meet your **CMU PLL reconfiguration** design goals. A maximum of five input reference clocks are allowed.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the Transmitter PLL and Receiver PLL?	If you select more than one input reference clock sources for the transmitter and/or receiver PLL, this option allows you to select the clock source for the current configuration.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the selected input clock source for the alternate Transmitter PLL and Receiver PLL?	If you select the **Use alternate Transmitter PLL and Receiver PLL** option, you can select the clock source for the alternate Transmitter PLL and the Receiver PLL.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the reconfig protocol driven by clock 0….4?	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to select the functional mode for the respective reference clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the clock 0….4 input frequency?	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to select the reference clock frequencies for each clock source.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use clock 1....4 reference clock divider	If you select more than one input reference clock source for the transmitter and/or receiver PLL, these options allow you to instruct the MegaWizard about the `REFCLK` pre-divider on input reference clocks.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–104 shows page 10 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode. This page appears only if the **Channel Internals** or the **Channel Interface** options are selected in the **Reconfig** page (Page 7).

Figure 4–104. MegaWizard Plug-In Manager - ALT2GXB (Reconfig 2)

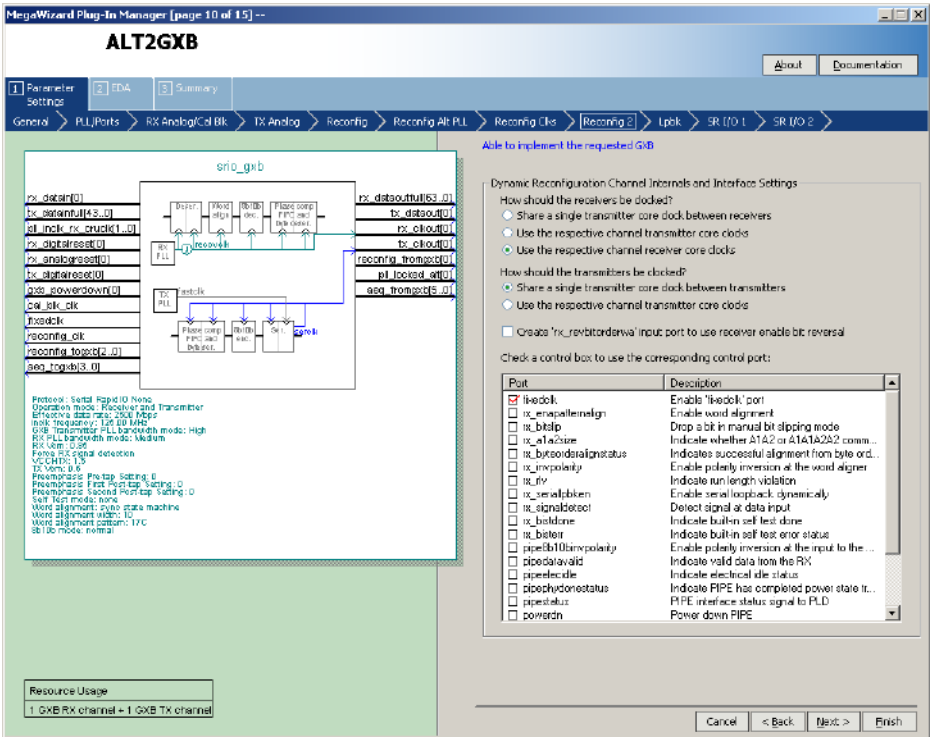


Table 4–85 describes the available options on page 10 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–85. MegaWizard Plug-In Manager Options (Page 10 for Serial RapidIO Mode) (Part 1 of 2)

ALT2GXB Setting	Description	Reference
How should the receivers be clocked?	Three options are available: • Share a single transmitter core clock between receivers • Use the respective channel transmitter core clock • Use the respective channel receiver core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
How should the transmitters be clocked?	Two options are available: • Share a single transmitter core clock between transmitters • Use the respective channel transmitter core clocks	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–85. MegaWizard Plug-In Manager Options (Page 10 for Serial RapidIO Mode) (Part 2 of 2)

ALT2GXB Setting	Description	Reference
Create `rx_revbitorderwa` input port to use receiver enable bit reversal	This optional input port allows you to dynamically reverse the bit order at the output of the receiver word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Check a control box to use the corresponding control port	You can select various control and status signals depending on what protocol(s) you intend to dynamically reconfigure the transceiver to.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–105 shows page 11 of the ALT2GXB MegaWizard Plug-In Manager for Serial RapidIO mode.

Figure 4–105. MegaWizard Plug-In Manager - ALT2GXB (Loopback)

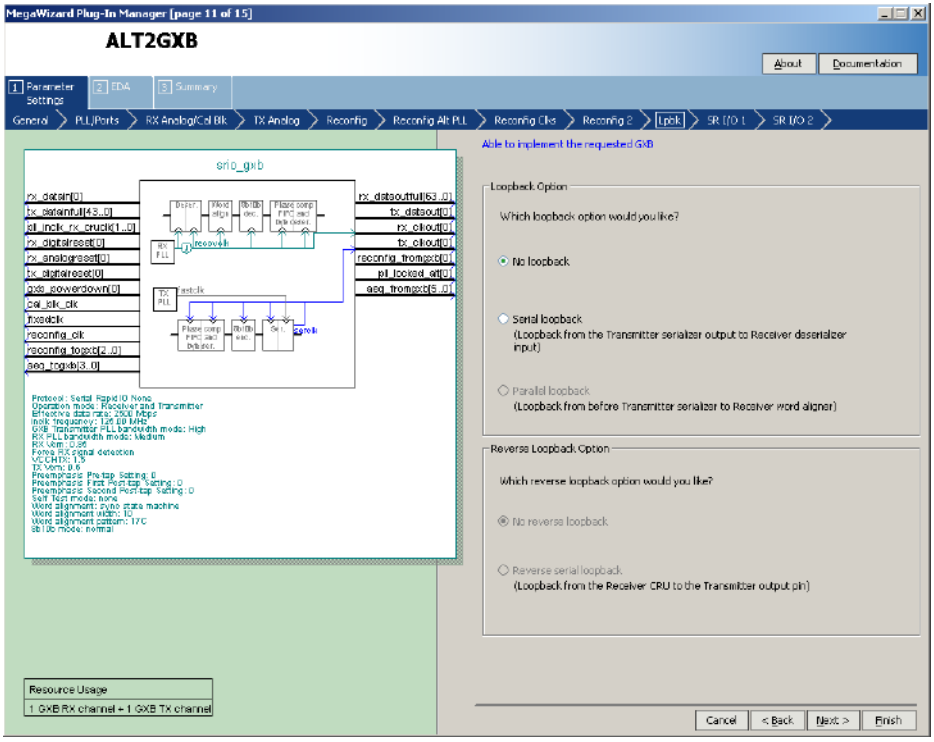


Table 4–86 describes the available options on page 11 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–86. MegaWizard Plug-In Manager Options (Page 11 for Serial RapidIO Mode)

ALT2GXB Setting	Description	Reference
Which loopback option would you like?	There are two options available in RapidIO mode: no loopback and serial loopback. • No loopback - this is the default mode. • Serial loopback - if you select serial loopback, the `rx_seriallpbken` port is available to control the serial loopback feature dynamically. A 1'b1 enables serial loopback and a 1'b0 disables loopback on a channel-by-channel basis. A digital reset must be asserted for the transceiver.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*
Reverse Loopback option	This option is not available in Serial RapidIO mode.	Loopback Modes section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Figure 4–106 shows page 12 of the MegaWizard Plug-In Manager for the Serial RapidIO protocol set up.

Figure 4–106. MegaWizard Plug-In Manager - ALT2GXB (SR I/O 1)

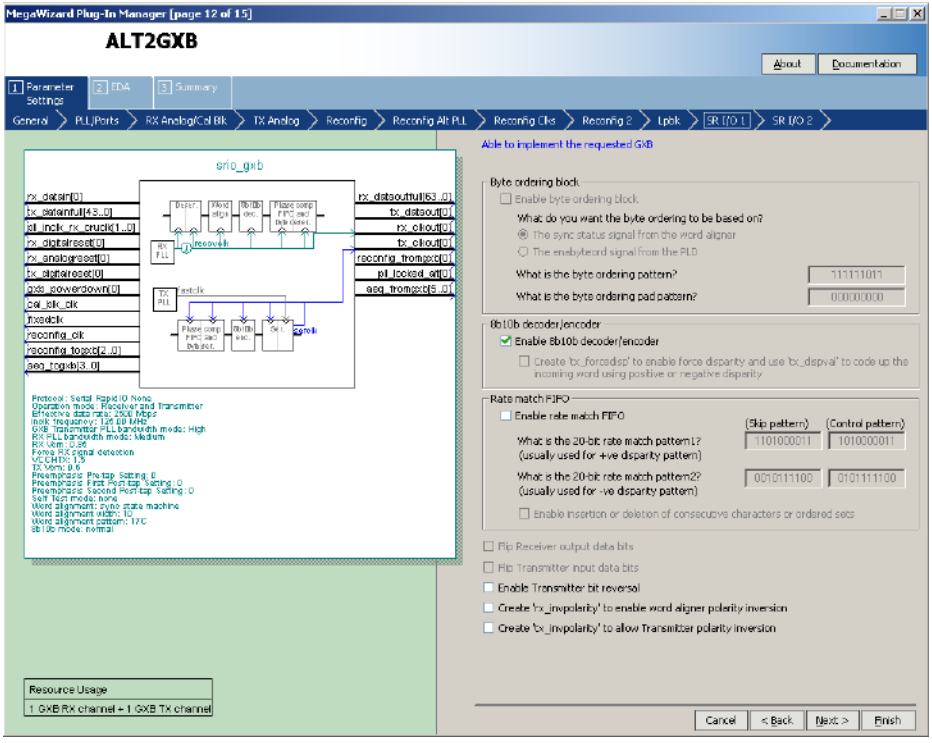


Table 4–87 describes the available options on page 12 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–87. MegaWizard Plug-In Manager Options (Page 12 for Serial RapidIO Mode)

ALT2GXB Setting	Description	Reference
Enable byte ordering block	This option is not available in Serial RapidIO mode.	Byte Ordering Block section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Enable 8B/10B decoder/encoder	The 8B/10B decoder/encoder is always enabled in Serial RapidIO mode.	8B/10B Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_forcedisp` to enable Force disparity and use `tx_dispval` to code up the incoming word using positive or negative disparity	This option is not available in Serial RapidIO mode.	8B/10B Encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Enable rate match FIFO	This option is not available in Serial RapidIO mode.	Rate Matcher section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Flip Receiver output data bits	This option is not available in Serial RapidIO mode.	
Flip Transmitter input data bits	This option is not available in Serial RapidIO mode.	
Enable Transmitter bit reversal	This option inverts (flips) the bit order of the data bits at the transmitter PCS-PMA interface at a byte level to support MSBit to LSBit transmission protocols. The default transmission is LSBit to MSBit.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_invpolarity` to enable word aligner polarity inversion	This optional port allows you to dynamically reverse the polarity of the received data at the input of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `tx_invpolarity` to allow Transmitter polarity inversion	This optional port allows you to dynamically reverse the polarity of the data to be transmitted at the transmitter PCS-PMA interface.	8B/10B encoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–107 shows page 13 of the MegaWizard Plug-In Manager for the Serial RapidIO protocol set up.

Figure 4–107. MegaWizard Plug-In Manager - ALT2GXB (SR I/O 2)

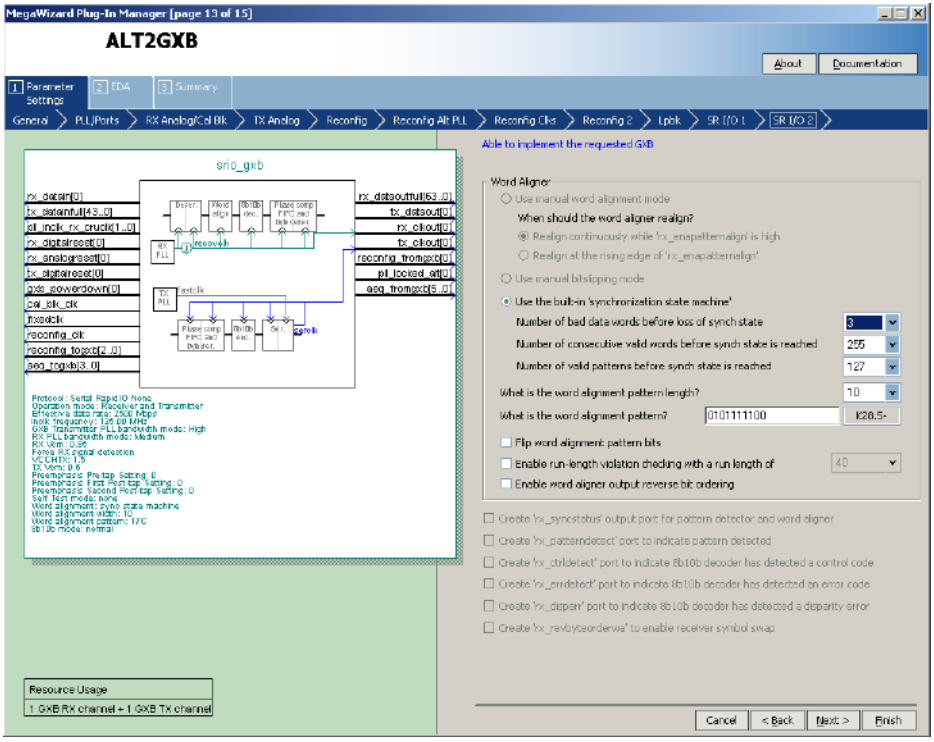


Table 4–88 describes the available options on page 13 of the MegaWizard Plug-In Manager for your ALT2GXB custom megafunction variation.

Table 4–88. MegaWizard Plug-In Manager Options (Page 13 for Serial RapidIO Mode) (Part 1 of 3)

ALT2GXB Setting	Description	Reference
Use manual word alignment mode	This option is not available in Serial RapidIO mode as the word aligner is synchronization state machine based.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use manual bit slipping mode	This option is not available in Serial RapidIO mode as the word aligner is synchronization state machine based.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use the built-in 'synchronization state machine'	This option is always enabled in Serial RapidIO mode as the word aligner is synchronization state machine based.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Number of bad data words before loss of synch state	The Quartus II software forces this field to **3** to comply with the Serial RapidIO specification.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Number of consecutive valid words before synch state is reached	The Quartus II software forces this field to **255** to comply with the Serial RapidIO specification.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Number of valid patterns before synch state is reached	The Quartus II software forces this field to **127** to comply with the Serial RapidIO specification.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
What is the word alignment pattern length?	The Quartus II software only allows a 10-bit wide word alignment pattern.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–88. MegaWizard Plug-In Manager Options (Page 13 for Serial RapidIO Mode) (Part 2 of 3)

ALT2GXB Setting	Description	Reference
What is the word alignment pattern?	The Quartus II software defaults the word alignment pattern to K28.5- (10'b0101111100).	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Flip word alignment pattern bits	This option reverses the bit order of the alignment pattern at a byte level to support MSB to LSB transmission protocols. The default transmission order is LSB to MSB.	
Enable run-length violation checking with a run length of	This option activates the run-length violation circuit. You can program the run length at which the circuit triggers the `rx_rlv` signal.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Enable word aligner output reverse bit ordering	In manual bit-slip mode, this option creates an input port `rx_revbitorderwa` to dynamically reverse the bit order at the output of the receiver word aligner. In other modes, this option statically configures the receiver to always reverse the bit order of the data at the output of the word aligner.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_syncstatus` output port for pattern detector and word aligner	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_patterndetect` port to indicate pattern detected	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_ctrldetect` port to indicate 8B/10B decoder has detected a control code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 4–88. MegaWizard Plug-In Manager Options (Page 13 for Serial RapidIO Mode) (Part 3 of 3)

ALT2GXB Setting	Description	Reference
Create `rx_errdetect` port to indicate 8B/10B decoder has detected an error code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_disperr` port to indicate 8B/10B decoder has detected a disparity code	Refer to the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook* for information about this port.	8B/10B Decoder section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Create `rx_revbyteorderwa` to enable receiver symbol swap	This option is not available in Serial RapidIO mode.	Word Aligner section in the *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 4–108 shows page 14 of the MegaWizard Plug-In Manager for the Serial RapidIO protocol set up. The Generate simulation model creates a behavioral model (*.vo* or *.vho*) of the transceiver instance for third-party simulators. The **Generate Netlist** option generates a netlist for the third party EDA synthesis tool to estimate timing and resource utilization for the ALT2GXB instance.

Figure 4–108. MegaWizard Plug-In Manager - ALT2GXB (EDA)

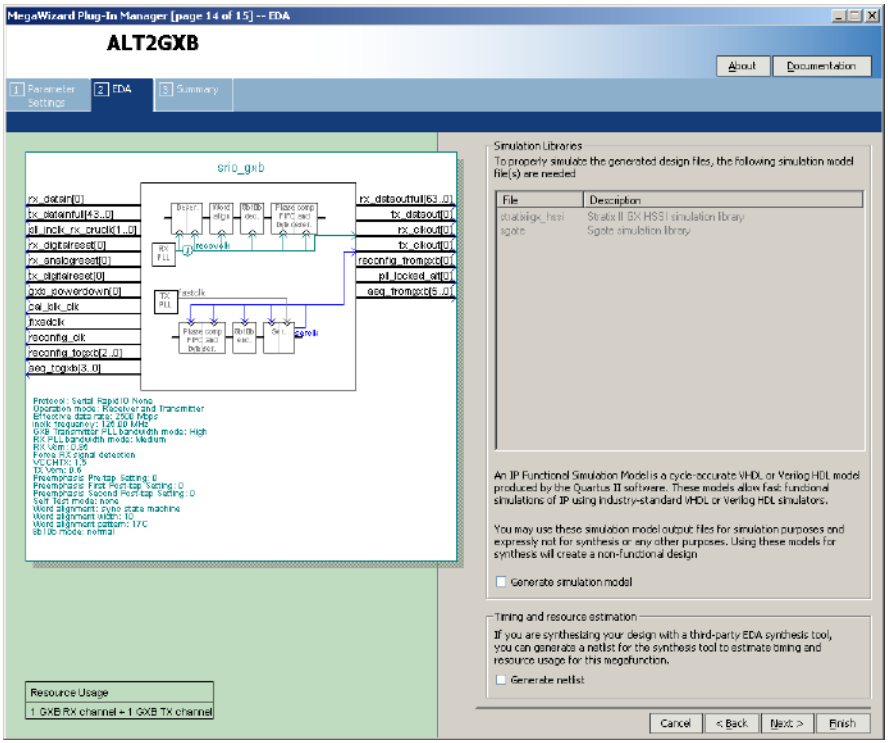
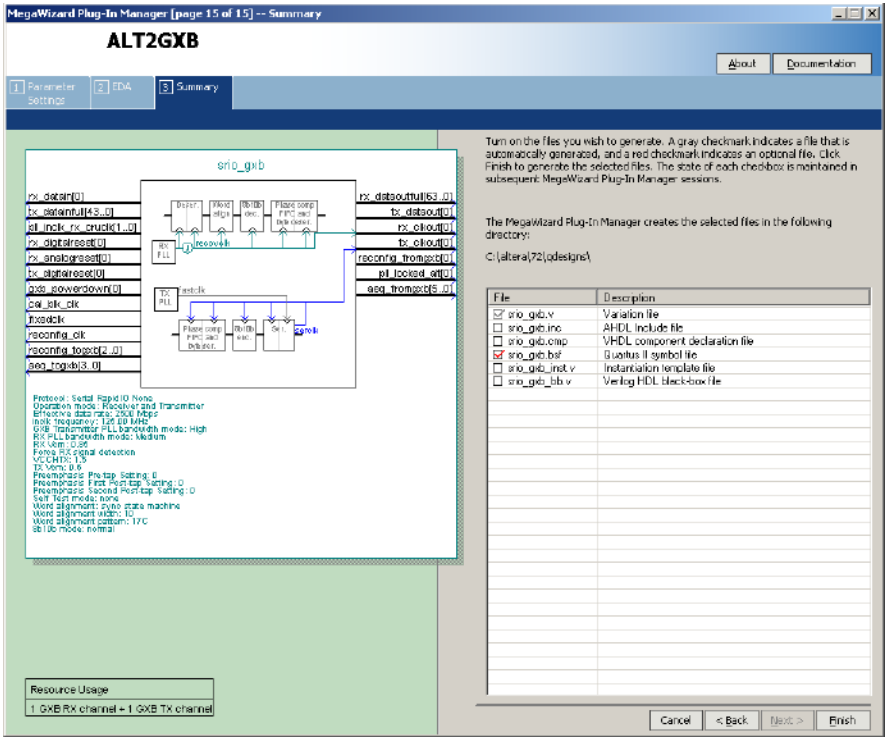


Figure 4–109 shows page 15 (last page) of the MegaWizard Plug-In Manager for the Serial RapidIO protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 4–109. MegaWizard Plug-In Manager - ALT2GXB (Summary)



Referenced Documents

This chapter references the following documents:

- *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
- *Stratix II GX Transceiver Architecture Overview* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Document Revision History

Table 4–89 shows the revision history for this chapter.

Table 4–89. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v4.2	Updated the entire chapter. • All tables updated. • All figures updated.	—
	New sections added: • "CPRI Mode" • "SDI Mode" • "Serial RapidIO Mode"	—
	Added the "Referenced Documents" section.	—
August 2007, v4.1	• Updated the "Which subprotocol will you be using?" section in Table 4–1. • Updated the "What is the input clock frequency?" section in Table 4–32. • Updated Table 4–16. • Updated "Reference" column in Table 4–2.	—
	Updated Figure 4–22.	—
	Formerly chapter 3. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter.	—
February 2007 v4.0	Added the "Document Revision History" section to this chapter.	—
	This entire chapter was updated. All the tables were updated and all new graphics were included.	—
June 2006, v3.2	• Updated MegaWizard Plug-In Manager figures for page 6 for all modes and page 9 for PIPE mode. • Updated Table 3–11 to include **Enable fast recovery mode** option.	—
April 2006, v3.1	• Updated all the MegaWizard Plug-In Manager figures to match the Quartus II software GUI. • Updated Tables 3–3, 3–8, 3–9, 3–13, 3–20, 3–25, 3–26, 3–29, 3–31, and 3–32.	—
February 2006, v3.0	• Updated technical content throughout chapter. • Added "(OIF) CEI PHY Interface Mode" section.	—

Table 4–89. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
December 2005, v2.0	Added XAUI, GIGE, and SONET sections.	—
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	—

5. Stratix II GX ALT2GXB_RECONFIG Megafunction User Guide

SIIGX52006-1.4

Introduction

The MegaWizard® Plug-In Manager in the Quartus® II software creates or modifies design files that contain custom megafunction variations. These auto-generated MegaWizard files can then be instantiated in a design file. The MegaWizard Plug-In Manager provides a MegaWizard that allows you to specify options for the ALT2GXB_RECONFIG megafunction.

Start the MegaWizard Plug-In Manager using one of the following methods:

- Choose the **MegaWizard Plug-In Manager** command (Tools menu).
- When working in the Block Editor (schematic symbol), click **MegaWizard Plug-In Manager** in the **Symbol** dialog box (Edit menu > Insert Symbol).
- Start the stand-alone version of the MegaWizard Plug-In Manager by typing the following command at the command prompt: `qmegawiz`.

Dynamic Reconfiguration

This section provides descriptions of the options available on the individual pages of the ALT2GXB_RECONFIG MegaWizard Plug-In Manager.

The MegaWizard Plug-In Manager provides a warning if any of the settings you choose are illegal.

Figure 5–1 shows the first page of the MegaWizard Plug-In Manager. To generate an ALT2GXB_RECONFIG custom megafunction variation, select **Create a new custom megafunction variation**. Click **Next**.

Figure 5–1. MegaWizard Plug-In Manager (Page 1)

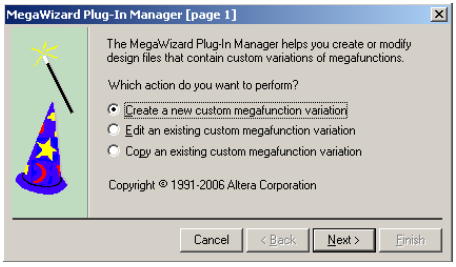


Figure 5–2 shows the second page of the MegaWizard Plug-In Manager. Select the following options (click **Next** when you are done):

- **ALT2GXB_RECONFIG megafunction** option, under the I/O folder.
- **Stratix II GX** as the device family.
- Your desired type of output file format (**Verilog**, **VHDL**, or **AHDL**).
- Your desired file name.

For the design to compile successfully, you must enable the dynamic reconfiguration controller in the `alt2gxb` instance.

Figure 5–2. MegaWizard Plug-In Manager - ALT2GXB_RECONFIG (Page 2)

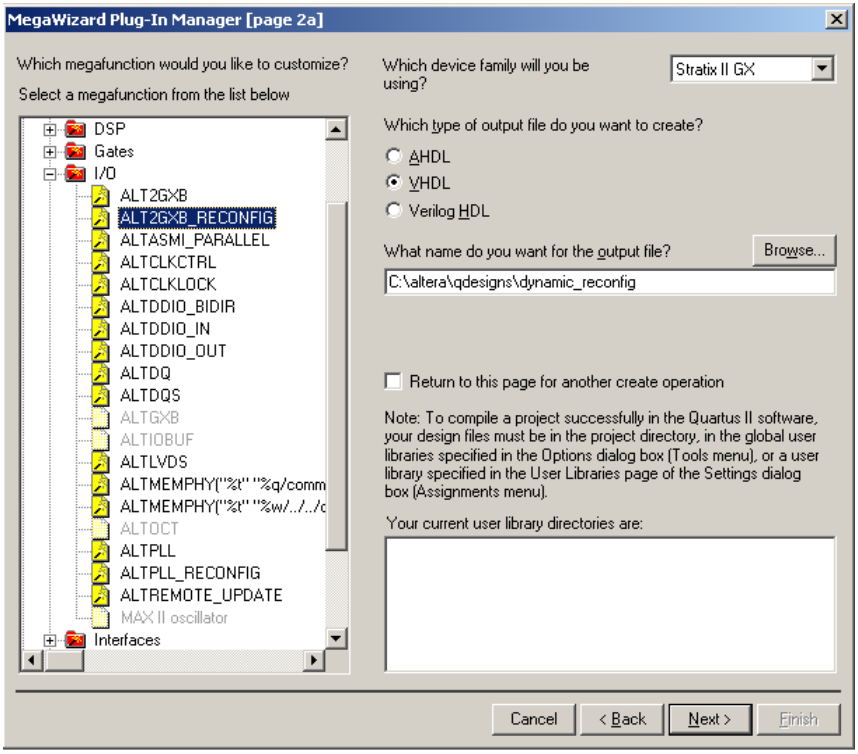


Figure 5–3 shows page 3 of the ALT2GXB_RECONFIG MegaWizard Plug-In Manager. From the drop-down menu, select the number of channels controlled by the reconfig controller. Check off the reconfig controller features that you would like to activate; for example, Analog controls, Channel Reconfiguration, change the local divider values of the transmitter or Channel and TX PLL reconfiguration.

Figure 5–3. MegaWizard Plug-In Manager - ALT2GXB_RECONFIG (Reconfiguration Settings)

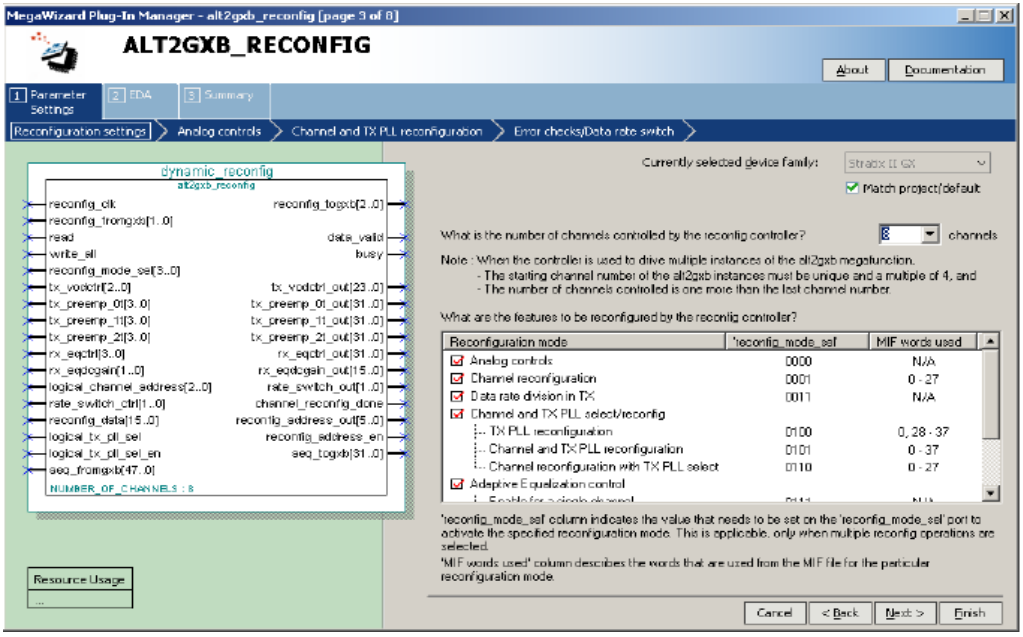


Table 5–1 describes the available options on page 3 of the MegaWizard Plug-In Manager for your ALT2GXB_RECONFIG custom megafunction variation. Select the **Match project/default** option if you want to change the device **Currently selected device family** options.

Make your selections on Page 3 and click **Next**.

Table 5–1. MegaWizard Plug-In Manager Options (Page 3) (Part 1 of 2)

ALT2GXB_RECONFIG Setting	Description	Reference
What is the number of channels controlled by the reconfig controller?	Depending on this setting, ALT2GXB_RECONFIG generates the required signal width for the interface signal (RECONFIG_FROMGXB) to `alt2gxb` and also gives the necessary bus width for all the selected physical media attachment (PMA) signals. For this setting, Altera® recommends that if there are multiple controllers for multiple instances of `alt2gxb`, then the setting is same as the number of channels set in the `alt2gxb` instance. If a single controller controls multiple instances of `alt2gxb`, the setting is rounded up to the multiple of the nearest transceiver (for the number of transceivers needed to fit the channels selected for that instance). Depending on the number of channels set, the resource estimate changes because this is a soft implementation that uses fabric logic resources. The resource estimate is shown in the bottom left of page 3 of the MegaWizard.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 5–1. MegaWizard Plug-In Manager Options (Page 3) (Part 2 of 2)

ALT2GXB_RECONFIG Setting	Description	Reference
What are the features to be reconfigured by the reconfig controller?	Three available selections are: **Analog Controls** – Allows dynamic reconfiguration of PMA settings like Equalization, Pre-emphasis, DC Gain, and V_{OD} **Channel Reconfiguration** – Allows dynamic reconfiguration of the transceiver channel from one pre-configured functional mode to another pre-configured functional mode. This includes switching from one protocol to another, as well as a data rate switch within BASIC mode. For example, dynamic reconfiguration from SONET/SDH to GIGE mode. **Data rate division in TX** – Allows dynamic switch of the CMU local clock divider. Division factors of 1, 2, and 4 are supported. For example, dynamic rate switching of the transmitter from 4 Gbps to 2 Gbps to 1 Gbps. **Channel and TX PLL select/reconfig** – The following three features are available under this option: ● TX PLL reconfiguration – Allows dynamic reconfiguration of the TX PLL only. ● Channel and TX PLL reconfiguration – Allows dynamic reconfiguration of the transceiver channel from one protocol mode to another and allows reconfiguration of the TX PLL. ● Channel reconfiguration with TX PLL select – Allows dynamic reconfiguration of the transceiver channel and allows selecting one of the two TX PLLs that the channel can listen to. ● Enable Adaptive Equalization Control – This feature is currently not supported.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 5–4 shows page 4 of the ALT2GXB_RECONFIG MegaWizard Plug-In Manager. Page 4 appears only if **Analog Controls** is selected in the **"What are the features to be reconfigured by the reconfig controller?"** setting on page 3.

Figure 5–4. MegaWizard Plug-In Manager - ALT2GXB_RECONFIG (Analog Controls)

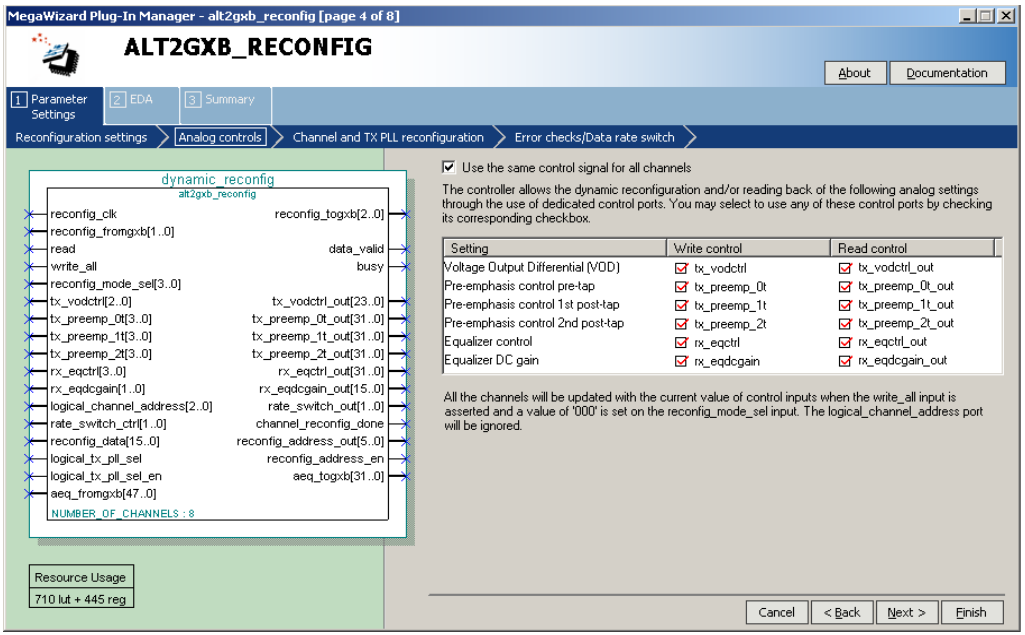


Table 5–2 describes the available options on page 4 of the MegaWizard Plug-In Manager for your ALT2GXB_RECONFIG custom megafunction variation.

Make your selections on page 4 and click **Next**.

Table 5–2. MegaWizard Plug-In Manager Options (Page 4) (Part 1 of 2)

ALT2GXB_RECONFIG Setting	Description	Reference
Use the same control signal for all channels (grayed out in Figure 5–4)	In Figure 5–4, this option is grayed out because it is not applicable for a one-channel instance. If the number of channels controlled by the controller is more than one, this setting is enabled. The setting is checked if the design needs the same control signal written into all channels simultaneously. If the design requires the control signal to write in and read out of individual channels, then the setting is not checked.	Dynamic Reconfiguration Setup in the MegaWizard Plug-In Manager section in the *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 5–2. MegaWizard Plug-In Manager Options (Page 4) (Part 2 of 2)

ALT2GXB_RECONFIG Setting	Description	Reference
Write Control	PMA write control signals are as follows: • Voltage Output Differential (V_{OD}) – 3 bits per channel • Pre-emphasis control pre-tap – 4 bits per channel • Pre-emphasis control 1st post-tap – 4 bits per channel • Pre-emphasis control 2nd post-tap – 4 bits per channel • Equalizer control – 4 bits per channel • Equalizer DC gain – 2 bits per channel These are optional signals. The signal widths are based on the setting you entered for the "What is the number of channels controlled by the controller?" option. At least one write signal must be enabled to configure and use the dynamic reconfiguration controller.	Channels and PMA Controls Reconfiguration *section of the Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Read Control	PMA read control signals are: • Voltage Output Differential (V_{OD}) • Pre-emphasis control pre-tap – 4 bits per channel • Pre-emphasis control 1st post-tap – 4 bits per channel • Pre-emphasis control 2nd post-tap – 4 bits per channel • Equalizer control – 4 bits per channel • Equalizer DC gain – 2 bits per channel These are optional signals. The signal widths are based on the setting you entered for the "What is the number of channels controlled by the controller?" option. The read out option is enabled for selection if the corresponding write control is selected. The read out option enable is not independent of write control. Read and write cannot be performed simultaneously into these PMA read control signals and PMA write control signals.	Channels and PMA Controls Reconfiguration *section of the Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 5–5 shows page 5 of the ALT2GXB_RECONFIG MegaWizard Plug-In Manager. Page 5 appears only if **Channel Reconfiguration** is selected in the **"What are the features to be reconfigured by the reconfig controller?"** setting on page 3.

Figure 5–5. MegaWizard Plug-In Manager - ALT2GXB_RECONFIG (Channel and TX PLL Reconfiguration)

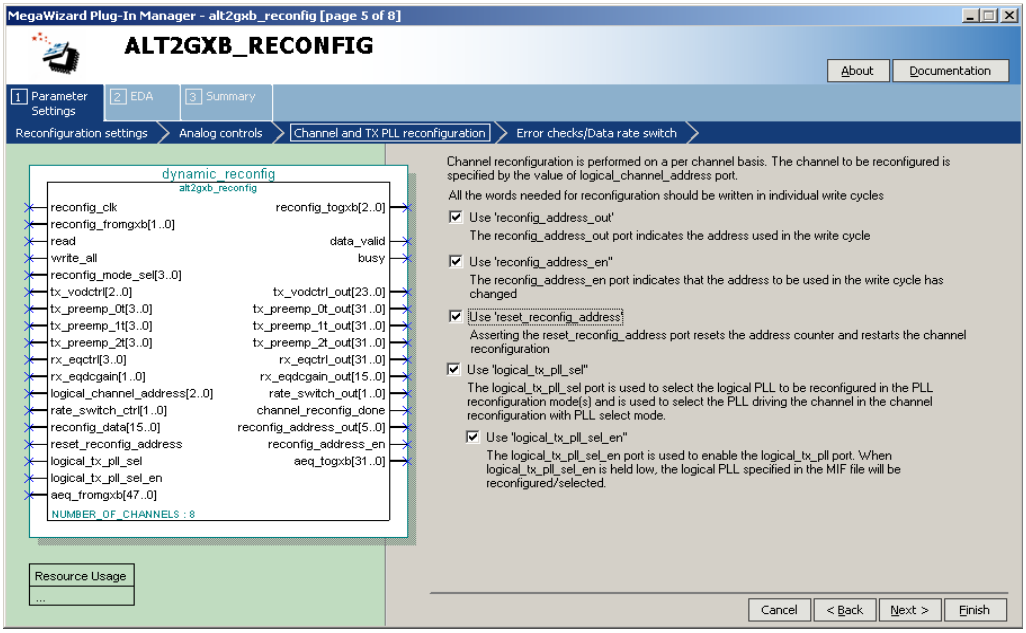


Table 5–3 describes the available options on page 5 of the MegaWizard Plug-In Manager for your ALT2GXB_RECONFIG custom megafunction variation.

Make your signal selection on page 5 and click **Next**.

Table 5–3. MegaWizard Plug-In Manager Options (Page 5) (Part 1 of 2)

ALT2GXB_RECONFIG Setting	Description	Reference
Use `reconfig_address_out`	The value on this optional port indicates the address associated with the words (reconfig instructions) in the .mif. Each dynamic configuration feature requires a maximum of 28 or 38 addresses. For example, if the Channel Reconfiguration feature is selected, the dynamic reconfiguration controller automatically increments the address from 0 to 27. If the Channel and TX PLL Reconfiguration feature is selected, the address is incremented from 0 to 37. Therefore, the width of the `reconfig_address_out` is set to either **5-bits** or **6-bits** wide, depending on the feature selected. The dynamic reconfiguration controller automatically increments the address at the end of each write cycle.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use `reconfig_address_en`	When high, this optional output status signal indicates that the address to be used in the write cycle has changed. This signal gets asserted when the write transaction is completed (`busy` signal de-asserted).	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use `reset_reconfig_address`	When asserted, this optional control signal resets the current reconfiguration address to 0.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 5–3. MegaWizard Plug-In Manager Options (Page 5) (Part 2 of 2)

ALT2GXB_RECONFIG Setting	Description	Reference
Use `logical_tx_pll_sel`	This is an optional control signal. The functionality of the signal depends on the feature selected, as shown below: • TX PLL reconfiguration – The corresponding TX PLL is reconfigured based on the value on this signal. • Channel and TX PLL reconfiguration – The corresponding TX PLL is reconfigured based on the value on this signal. The transceiver channel listens to the TX PLL selected by this signal. • Channel reconfiguration with TX PLL select - The transceiver channel listens to the TX PLL selected by this signal.	Channel and CMU PLL Reconfiguration section in the *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use `logical_tx_pll_sel_en`	This is an optional control signal. When this signal is enabled in the ALT2GXB_Reconfig Megawizard, the value set on the `logical_tx_pll_sel` signal is valid only if the `logical_tx_pll_sel_en` is set to **1**. For more information, refer to the "Logical TX PLL Select" section in the *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.	Channel and CMU PLL Reconfiguration section in the *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 5–6 shows page 6 of the ALT2GXB_RECONFIG MegaWizard Plug-In Manager. Page 6 appears only if the **Data rate division in TX** is selected in the **"What are the features to be reconfigured by the reconfig controller?"** setting on page 3.

Figure 5–6. MegaWizard Plug-In Manager - ALT2GXB_RECONFIG (Error Checks/Data Rate Switch)

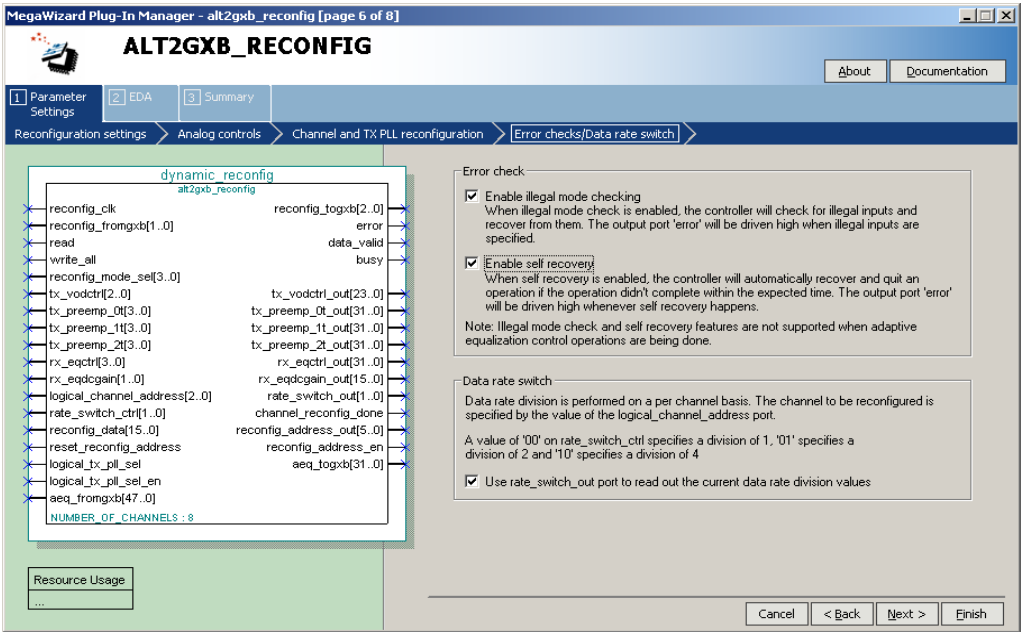


Table 5–4 describes the available options on page 6 of the MegaWizard Plug-In Manager for your ALT2GXB_RECONFIG custom megafunction variation.

Make your selections on page 6 and click **Next**.

Table 5–4. MegaWizard Plug-In Manager Options (Page 6)

ALT2GXB_RECONFIG Setting	Description	Reference
Enable illegal mode checking	When this option is selected, the ALT2GXB_RECONFIG MegaWizard provides the `error` output port. The dynamic reconfiguration controller checks for specific unsupported options within 2 `reconfig_clk` cycles, de-asserts the `busy` signal and asserts the `error` output port for 2 `reconfig_clk` cycles. The dynamic reconfiguration controller does not execute the unsupported operation.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Enable self recovery	When this option is selected, the ALT2GXB_RECONFIG MegaWizard provides the `error` output port. The dynamic reconfiguration controller quits an operation if it did not complete within the expected number of clock cycles. After recovering from the illegal operation, the dynamic reconfiguration controller de-asserts the `busy` signal and asserts the `error` output port for 2 `reconfig_clk` cycles.	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.
Use the `rate_switch_out`	Port to read out the current data rate division values. This optional output status port reads out the current setting on the CMU local divider. 00 — Division of 1 01 — Division of 2 10 — Division of 4 11 — illegal value (do not use this value)	*Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 5–7 shows page 7 (the Simulation Libraries page) of the MegaWizard Plug-In Manager for the Dynamic Reconfiguration selection.

Click **Next**.

Figure 5–7. MegaWizard Plug-In Manager - ALT2GXB_RECONFIG (Simulation Libraries)

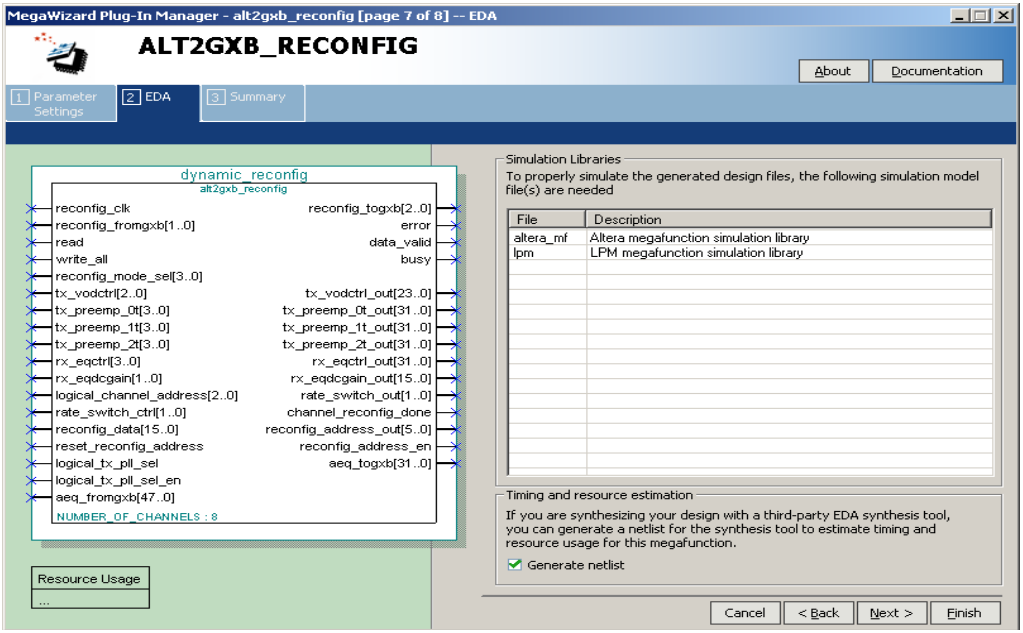


Table 5–5 describes the available option on page 7 of the MegaWizard Plug-In Manager for your ALT2GXB_RECONFIG custom megafunction variation.

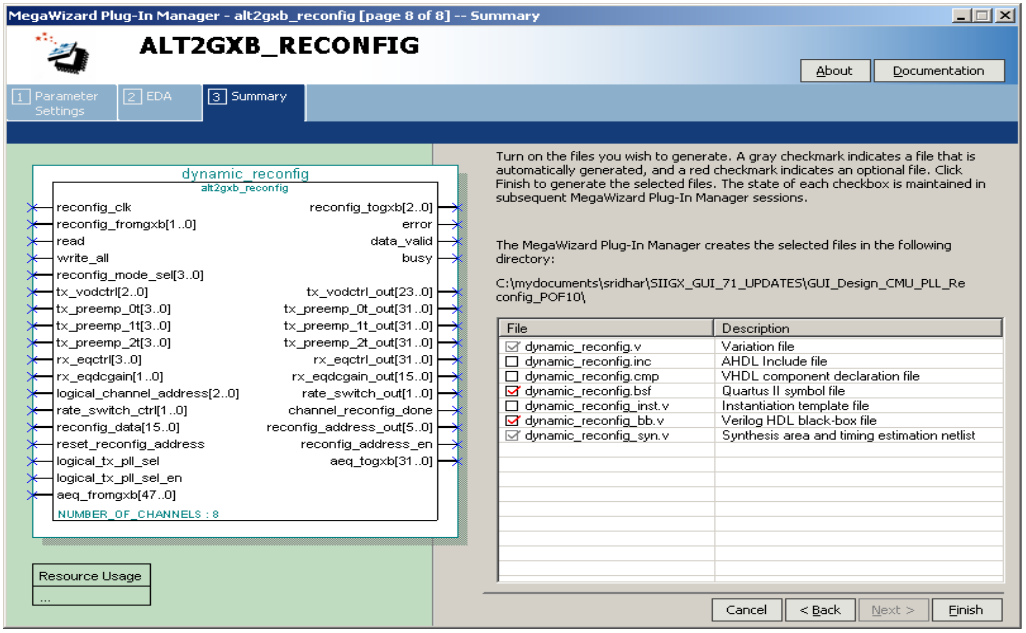
Make your selections on page 7 and click **Next**

Table 5–5. MegaWizard Plug-In Manager Options (Page 7)

ALT2GXB_RECONFIG Setting	Description
Generate a netlist for synthesis area and timing estimation	Selecting this option generates a netlist file that third party synthesis tools can use to estimate the timing and resource usage

Figure 5–8 shows page 8 (the last page) of the MegaWizard Plug-In Manager for the Dynamic Reconfiguration protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 5–8. MegaWizard Plug-In Manager - ALT2GXB_RECONFIG (Summary)



Referenced Document

This chapter references the following document:

- *Stratix II GX Dynamic Reconfiguration* chapter in volume 2 of the *Stratix II GX Device Handbook*

Document Revision History

Table 5–6 shows the revision history for this chapter.

Table 5–6. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007 v1.4	Updated Table 5–4.	—
	Updated all the figures in this chapter.	—
	Added "Referenced Document" section.	—
August 2007 v1.3	Updated Figures 5–2 through 5–8.	—
	Updated Tables 5–1 through 5–4.	—
	Added Table 5–5.	—
	Formerly chapter 4. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter.	—
February 2007 v1.2	Modified "Introduction".	Removed one sentence.
	Added the "Document Revision History" section to this chapter.	—
	Changed alt2gxb_reconfig to ALT2GXB_RECONFIG throughout.	Per new style guide convention.
	Added more information describing Figures 5–2 and 5–3.	—
	Updated "Use `reconfig_address_out`" section of Table 5–3.	—
	Updated "Use the `rate_switch_out` port to read out the current data rate division value" section of Table 5–4.	—
	Added "Click **Next**" instructions after each step.	—
April 2006, v1.1	Updated all the MegaWizard Plug-In Manager figures to match the Quartus II software GUI.	—
February 2006, v1.0	Added chapter to the *Stratix II GX Device Handbook*.	—

6. Specifications & Additional Information

SIIGX52004-3.1

Transceiver Blocks

Table 6–1 shows the transceiver blocks for Stratix® II GX and Stratix GX devices and compares their features.

Table 6–1. Stratix II GX Features Versus Stratix GX Features (Part 1 of 2)

Blocks	Features	Stratix GX	Stratix II GX
	Data rate	500 Mbps to 3.1875 Gbps	600 Mbps to 6.375 Gbps
	Data path	Single width	Single or double width
	Native protocol support	Basic, XAUI, GIGE, SONET/SDH	Basic, XAUI, GIGE, SONET/SDH (OC-12, OC-48, OC-96), PIPE, (OIF) CEI PHY Interface, CPRI, Serial RapidIO, SDI
CMU	Transmitter PLL	Single transmitter PLL for entire transceiver block	Multiple transmitter PLLs
	REFCLK	One reference clock per transceiver block	Two reference clocks per transceiver block
8B/10B	Polarity inversion		✓
Transmitter buffer	Low-power mode		✓
	Tri-state		✓
	Receiver detect		✓
	V_{OD} and PE	✓ - Dynamic signals	✓ - Dynamic signals
Receiver buffer	EQ	✓ - Dynamic signals	✓ - Dynamic signals

Table 6–1. Stratix II GX Features Versus Stratix GX Features (Part 2 of 2)

Blocks	Features	Stratix GX	Stratix II GX
Word aligner	Synchronization SM	GIGE and XAUI only	Available in Basic single-width, PIPE, XAUI, and GIGE modes
	32-bit pattern		✓
	20-bit pattern		✓
	16-bit pattern	✓	✓
	10-bit pattern	✓	✓
	8-bit pattern		✓
	7-bit pattern	✓	✓
	Bit slip	✓	✓
Rate matcher		GIGE and XAUI only	Available in Basic, PIPE, XAUI, and GIGE modes
Byte ordering			✓
Loopback	Serial	✓	✓ - All modes except PCI Express (PIPE) mode
	Parallel	✓	✓ - Only in Basic mode
	Reverse serial	✓	✓ - no longer dynamic
	PIPE reverse parallel		✓ - PIPE mode only
	Post 8B/10B	✓	
BIST	PRBS 7		✓- Only in Basic double-width mode
	PRBS 8 *(1)*	✓	
	PRBS 10	✓	✓- Only in Basic double-width mode
	Low frequency *(2)*	✓	
	High frequency *(2)*	✓	
	Mixed frequency *(2)*	✓	
	Incremental *(2)*	✓	✓- Only in parallel loopback mode

Notes to Table 6–1:
(1) Non-8B/10B.
(2) With 8B/10B encoding.

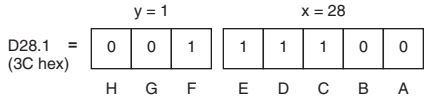
8B/10B Code

This section provides information about the data and control codes for Stratix II GX devices.

Code Notation

The 8B/10B data and control codes are referred to as Dx.y and Kx.y, respectively. The 8-bit byte (H G F E D C B A, where H is the MSB and A is the LSB) is broken up into two groups, x and y, where x is the five lower bits (E D C B A) and y is the upper three bits (H G F). Figure 6–1 shows the designation for 3C hex.

Figure 6–1. Sample Notation for 3C hex



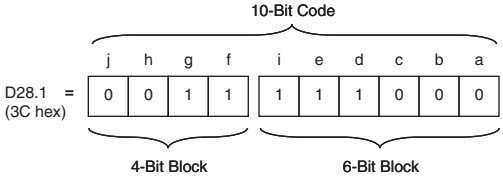
There are 256 Dx.y and 12 Kx.y valid 8-bit codes. These codes have two 10-bit equivalent codes associated with each 8-bit code. The 10-bit codes have either a neutral disparity or a non-neutral disparity. With neutral disparity, two neutral disparity 10-bit codes are associated with an 8-bit code. With non-neutral disparity 10-bit code, a positive and a negative disparity code are associated with the 8-bit code.

The positive disparity 10-bit code is associated in the RD- column. The negative disparity 10-bit code is associated in the RD+ column.

Disparity Calculation

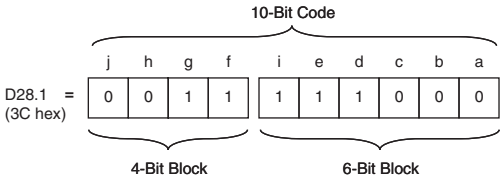
The running disparity is calculated based on the sub-blocks of the 10-bit code. The 10-bit code is divided into two sub blocks, a 6-bit sub-block (abcdei) and a 4-bit sub-block (fghj), as shown in Figure 6–2.

Figure 6–2. 10-Bit Grouping of 6-bit and 4-Bit Sub-Blocks



The running disparity at the beginning of the 6-bit sub-block is running disparity at the end of the previous 10-bit code. The running disparity of the 4-bit sub-block is the running disparity of the end of the 6-bit sub-block. The running disparity of the end of the 4-bit sub-block is the running disparity of the 10-bit code (refer to Figure 6–3).

Figure 6–3. Running Disparity Between Sub-Blocks



The running disparity calculation rules are as follows:

- The current running disparity at the end of a sub-block is positive if any of the following is true:
  - The sub-block contains more ones than zeros
  - The 6-bit sub-block is 6'b000111
  - The 4-bit sub-block is 4'b0011
- The current running disparity at the end of a sub-block is negative if any of the following is true:
  - The sub-block contains more zeros than ones
  - The 6-bit sub-block is 6'b111000
  - The 4-bit sub-block is 4'b1100

If those conditions are not met, the running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

Supported Codes

The 8B/10B scheme defines the 12 control codes listed in Table 6–2 for synchronization, alignment, and general application purposes.

Table 6–2. Supported K Codes (Part 1 of 2)

K Code	Octal Value	8-Bit Code HGF_EDCBA	10-Bit Code RD- abcdei_fghj	10-Bit Code RD+ abcdei_fghj
K28.0	1C	8'b000_11100	10'b001111_0100	10'b110000_1011
K28.1	3C	8'b001_11100	10'b001111_1001	10'b110000_0110
K28.2	5C	8'b010_11100	10'b001111_0101	10'b110000_1010

Table 6–2. Supported K Codes (Part 2 of 2)

K Code	Octal Value	8-Bit Code HGF_EDCBA	10-Bit Code RD- abcdei_fghj	10-Bit Code RD+ abcdei_fghj
K28.3	7C	8'b011_11100	10'b001111_0011	10'b110000_1100
K28.4	9C	8'b100_11100	10'b001111_0010	10'b110000_1101
K28.5 *(1)*	BC	8'b101_11100	10'b001111_1010	10'b110000_0101
K28.6	DC	8'b110_11100	10'b001111_0110	10'b110000_1001
K28.7	FC	8'b111_11100	10'b001111_1000	10'b110000_0111
K23.7	F7	8'b111_10111	10'b111010_1000	10'b000101_0111
K27.7	FB	8'b111_11011	10'b110110_1000	10'b001001_0111
K29.7	FD	8'b111_11101	10'b101110_1000	10'b010001_0111
K30.7	FE	8'b111_11110	10'b011110_1000	10'b100001_0111

Note to Table 6–2:

(1) K28.5 is a comma code used for word alignment and indicates an IDLE state.

Table 6–3 shows the valid data code-groups.

Table 6–3. Valid Data Code-Groups (Part 1 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD-	Current RD+
			abcdei fghj	abcdei fghj
D0.0	00	000 00000	100111 0100	011000 1011
D1.0	01	000 00001	011101 0100	100010 1011
D2.0	02	000 00010	101101 0100	010010 1011
D3.0	03	000 00011	110001 1011	110001 0100
D4.0	04	000 00100	110101 0100	001010 1011
D5.0	05	000 00101	101001 1011	101001 0100
D6.0	06	000 00110	011001 1011	011001 0100
D7.0	07	000 00111	111000 1011	000111 0100
D8.0	08	000 01000	111001 0100	000110 1011
D9.0	09	000 01001	100101 1011	100101 0100
D10.0	0A	000 01010	010101 1011	010101 0100
D11.0	0B	000 01011	110100 1011	110100 0100
D12.0	0C	000 01100	001101 1011	001101 0100
D13.0	0D	000 01101	101100 1011	101100 0100
D14.0	0E	000 01110	011100 1011	011100 0100
D15.0	0F	000 01111	010111 0100	101000 1011

Table 6–3. Valid Data Code-Groups (Part 2 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD-	Current RD+
			abcdei fghj	abcdei fghj
D16.0	10	000 10000	011011 0100	100100 1011
D17.0	11	000 10001	100011 1011	100011 0100
D18.0	12	000 10010	010011 1011	010011 0100
D19.0	13	000 10011	110010 1011	110010 0100
D20.0	14	000 10100	001011 1011	001011 0100
D21.0	15	000 10101	101010 1011	101010 0100
D22.0	16	000 10110	011010 1011	011010 0100
D23.0	17	000 10111	111010 0100	000101 1011
D24.0	18	000 11000	110011 0100	001100 1011
D25.0	19	000 11001	100110 1011	100110 0100
D26.0	1A	000 11010	010110 1011	010110 0100
D27.0	1B	000 11011	110110 0100	001001 1011
D28.0	1C	000 11100	001110 1011	001110 0100
D29.0	1D	000 11101	101110 0100	010001 1011
D30.0	1E	000 11110	011110 0100	100001 1011
D31.0	1F	000 11111	101011 0100	010100 1011
D0.1	20	001 00000	100111 1001	011000 1001
D1.1	21	001 00001	011101 1001	100010 1001
D2.1	22	001 00010	101101 1001	010010 1001
D3.1	23	001 00011	110001 1001	110001 1001
D4.1	24	001 00100	110101 1001	001010 1001
D5.1	25	001 00101	101001 1001	101001 1001
D6.1	26	001 00110	011001 1001	011001 1001
D7.1	27	001 00111	111000 1001	000111 1001
D8.1	28	001 01000	111001 1001	000110 1001
D9.1	29	001 01001	100101 1001	100101 1001
D10.1	2A	001 01010	010101 1001	010101 1001
D11.1	2B	001 01011	110100 1001	110100 1001
D12.1	2C	001 01100	001101 1001	001101 1001
D13.1	2D	001 01101	101100 1001	101100 1001
D14.1	2E	001 01110	011100 1001	011100 1001
D15.1	2F	001 01111	010111 1001	101000 1001
D16.1	30	001 10000	011011 1001	100100 1001

Table 6–3. Valid Data Code-Groups (Part 3 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD- abcdei fghj	Current RD+ abcdei fghj
D17.1	31	001 10001	100011 1001	100011 1001
D18.1	32	001 10010	010011 1001	010011 1001
D19.1	33	001 10011	110010 1001	110010 1001
D20.1	34	001 10100	001011 1001	001011 1001
D21.1	35	001 10101	101010 1001	101010 1001
D22.1	36	001 10110	011010 1001	011010 1001
D23.1	37	001 10111	111010 1001	000101 1001
D24.1	38	001 11000	110011 1001	001100 1001
D25.1	39	001 11001	100110 1001	100110 1001
D26.1	3A	001 11010	010110 1001	010110 1001
D27.1	3B	001 11011	110110 1001	001001 1001
D28.1	3C	001 11100	001110 1001	001110 1001
D29.1	3D	001 11101	101110 1001	010001 1001
D30.1	3E	001 11110	011110 1001	100001 1001
D31.1	3F	001 11111	101011 1001	010100 1001
D0.2	40	010 00000	100111 0101	011000 0101
D1.2	41	010 00001	011101 0101	100010 0101
D2.2	42	010 00010	101101 0101	010010 0101
D3.2	43	010 00011	110001 0101	110001 0101
D4.2	44	010 00100	110101 0101	001010 0101
D5.2	45	010 00101	101001 0101	101001 0101
D6.2	46	010 00110	011001 0101	011001 0101
D7.2	47	010 00111	111000 0101	000111 0101
D8.2	48	010 01000	111001 0101	000110 0101
D9.2	49	010 01001	100101 0101	100101 0101
D10.2	4A	010 01010	010101 0101	010101 0101
D11.2	4B	010 01011	110100 0101	110100 0101
D12.2	4C	010 01100	001101 0101	001101 0101
D13.2	4D	010 01101	101100 0101	101100 0101
D14.2	4E	010 01110	011100 0101	011100 0101
D15.2	4F	010 01111	010111 0101	101000 0101
D16.2	50	010 10000	011011 0101	100100 0101
D17.2	51	010 10001	100011 0101	100011 0101

Table 6–3. Valid Data Code-Groups (Part 4 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD-	Current RD+
			abcdei fghj	abcdei fghj
D18.2	52	010 10010	010011 0101	010011 0101
D19.2	53	010 10011	110010 0101	110010 0101
D20.2	54	010 10100	001011 0101	001011 0101
D21.2	55	010 10101	101010 0101	101010 0101
D22.2	56	010 10110	011010 0101	011010 0101
D23.2	57	010 10111	111010 0101	000101 0101
D24.2	58	010 11000	110011 0101	001100 0101
D25.2	59	010 11001	100110 0101	100110 0101
D26.2	5A	010 11010	010110 0101	010110 0101
D27.2	5B	010 11011	110110 0101	001001 0101
D28.2	5C	010 11100	001110 0101	001110 0101
D29.2	5D	010 11101	101110 0101	010001 0101
D30.2	5E	010 11110	011110 0101	100001 0101
D31.2	5F	010 11111	101011 0101	010100 0101
D0.3	60	011 00000	100111 0011	011000 1100
D1.3	61	011 00001	011101 0011	100010 1100
D2.3	62	011 00010	101101 0011	010010 1100
D3.3	63	011 00011	110001 1100	110001 0011
D4.3	64	011 00100	110101 0011	001010 1100
D5.3	65	011 00101	101001 1100	101001 0011
D6.3	66	011 00110	011001 1100	011001 0011
D7.3	67	011 00111	111000 1100	000111 0011
D8.3	68	011 01000	111001 0011	000110 1100
D9.3	69	011 01001	100101 1100	100101 0011
D10.3	6A	011 01010	010101 1100	010101 0011
D11.3	6B	011 01011	110100 1100	110100 0011
D12.3	6C	011 01100	001101 1100	001101 0011
D13.3	6D	011 01101	101100 1100	101100 0011
D14.3	6E	011 01110	011100 1100	011100 0011
D15.3	6F	011 01111	010111 0011	101000 1100
D16.3	70	011 10000	011011 0011	100100 1100
D17.3	71	011 10001	100011 1100	100011 0011
D18.3	72	011 10010	010011 1100	010011 0011

Table 6–3. Valid Data Code-Groups (Part 5 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD-	Current RD+
			abcdei fghj	abcdei fghj
D19.3	73	011 10011	110010 1100	110010 0011
D20.3	74	011 10100	001011 1100	001011 0011
D21.3	75	011 10101	101010 1100	101010 0011
D22.3	76	011 10110	011010 1100	011010 0011
D23.3	77	011 10111	111010 0011	000101 1100
D24.3	78	011 11000	110011 0011	001100 1100
D25.3	79	011 11001	100110 1100	100110 0011
D26.3	7A	011 11010	010110 1100	010110 0011
D27.3	7B	011 11011	110110 0011	001001 1100
D28.3	7C	011 11100	001110 1100	001110 0011
D29.3	7D	011 11101	101110 0011	010001 1100
D30.3	7E	011 11110	011110 0011	100001 1100
D31.3	7F	011 11111	101011 0011	010100 1100
D0.4	80	100 00000	100111 0010	011000 1101
D1.4	81	100 00001	011101 0010	100010 1101
D2.4	82	100 00010	101101 0010	010010 1101
D3.4	83	100 00011	110001 1101	110001 0010
D4.4	84	100 00100	110101 0010	001010 1101
D5.4	85	100 00101	101001 1101	101001 0010
D6.4	86	100 00110	011001 1101	011001 0010
D7.4	87	100 00111	111000 1101	000111 0010
D8.4	88	100 01000	111001 0010	000110 1101
D9.4	89	100 01001	100101 1101	100101 0010
D10.4	8A	100 01010	010101 1101	010101 0010
D11.4	8B	100 01011	110100 1101	110100 0010
D12.4	8C	100 01100	001101 1101	001101 0010
D13.4	8D	100 01101	101100 1101	101100 0010
D14.4	8E	100 01110	011100 1101	011100 0010
D15.4	8F	100 01111	010111 0010	101000 1101
D16.4	90	100 10000	011011 0010	100100 1101
D17.4	91	100 10001	100011 1101	100011 0010
D18.4	92	100 10010	010011 1101	010011 0010
D19.4	93	100 10011	110010 1101	110010 0010

Table 6–3. Valid Data Code-Groups (Part 6 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD-	Current RD+
			abcdei fghj	abcdei fghj
D20.4	94	100 10100	001011 1101	001011 0010
D21.4	95	100 10101	101010 1101	101010 0010
D22.4	96	100 10110	011010 1101	011010 0010
D23.4	97	100 10111	111010 0010	000101 1101
D24.4	98	100 11000	110011 0010	001100 1101
D25.4	99	100 11001	100110 1101	100110 0010
D26.4	9A	100 11010	010110 1101	010110 0010
D27.4	9B	100 11011	110110 0010	001001 1101
D28.4	9C	100 11100	001110 1101	001110 0010
D29.4	9D	100 11101	101110 0010	010001 1101
D30.4	9E	100 11110	011110 0010	100001 1101
D31.4	9F	100 11111	101011 0010	010100 1101
D0.5	A0	101 00000	100111 1010	011000 1010
D1.5	A1	101 00001	011101 1010	100010 1010
D2.5	A2	101 00010	101101 1010	010010 1010
D3.5	A3	101 00011	110001 1010	110001 1010
D4.5	A4	101 00100	110101 1010	001010 1010
D5.5	A5	101 00101	101001 1010	101001 1010
D6.5	A6	101 00110	011001 1010	011001 1010
D7.5	A7	101 00111	111000 1010	000111 1010
D8.5	A8	101 01000	111001 1010	000110 1010
D9.5	A9	101 01001	100101 1010	100101 1010
D10.5	AA	101 01010	010101 1010	010101 1010
D11.5	AB	101 01011	110100 1010	110100 1010
D12.5	AC	101 01100	001101 1010	001101 1010
D13.5	AD	101 01101	101100 1010	101100 1010
D14.5	AE	101 01110	011100 1010	011100 1010
D15.5	AF	101 01111	010111 1010	101000 1010
D16.5	B0	101 10000	011011 1010	100100 1010
D17.5	B1	101 10001	100011 1010	100011 1010
D18.5	B2	101 10010	010011 1010	010011 1010
D19.5	B3	101 10011	110010 1010	110010 1010
D20.5	B4	101 10100	001011 1010	001011 1010

Table 6–3. Valid Data Code-Groups (Part 7 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD-	Current RD+
			abcdei fghj	abcdei fghj
D21.5	B5	101 10101	101010 1010	101010 1010
D22.5	B6	101 10110	011010 1010	011010 1010
D23.5	B7	101 10111	111010 1010	000101 1010
D24.5	B8	101 11000	110011 1010	001100 1010
D25.5	B9	101 11001	100110 1010	100110 1010
D26.5	BA	101 11010	010110 1010	010110 1010
D27.5	BB	101 11011	110110 1010	001001 1010
D28.5	BC	101 11100	001110 1010	001110 1010
D29.5	BD	101 11101	101110 1010	010001 1010
D30.5	BE	101 11110	011110 1010	100001 1010
D31.5	BF	101 11111	101011 1010	010100 1010
D0.6	C0	110 00000	100111 0110	011000 0110
D1.6	C1	110 00001	011101 0110	100010 0110
D2.6	C2	110 00010	101101 0110	010010 0110
D3.6	C3	110 00011	110001 0110	110001 0110
D4.6	C4	110 00100	110101 0110	001010 0110
D5.6	C5	110 00101	101001 0110	101001 0110
D6.6	C6	110 00110	011001 0110	011001 0110
D7.6	C7	110 00111	111000 0110	000111 0110
D8.6	C8	110 01000	111001 0110	000110 0110
D9.6	C9	110 01001	100101 0110	100101 0110
D10.6	CA	110 01010	010101 0110	010101 0110
D11.6	CB	110 01011	110100 0110	110100 0110
D12.6	CC	110 01100	001101 0110	001101 0110
D13.6	CD	110 01101	101100 0110	101100 0110
D14.6	CE	110 01110	011100 0110	011100 0110
D15.6	CF	110 01111	010111 0110	101000 0110
D16.6	D0	110 10000	011011 0110	100100 0110
D17.6	D1	110 10001	100011 0110	100011 0110
D18.6	D2	110 10010	010011 0110	010011 0110
D19.6	D3	110 10011	110010 0110	110010 0110
D20.6	D4	110 10100	001011 0110	001011 0110
D21.6	D5	110 10101	101010 0110	101010 0110

Table 6–3. Valid Data Code-Groups (Part 8 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD-	Current RD+
			abcdei fghj	abcdei fghj
D22.6	D6	110 10110	011010 0110	011010 0110
D23.6	D7	110 10111	111010 0110	000101 0110
D24.6	D8	110 11000	110011 0110	001100 0110
D25.6	D9	110 11001	100110 0110	100110 0110
D26.6	DA	110 11010	010110 0110	010110 0110
D27.6	DB	110 11011	110110 0110	001001 0110
D28.6	DC	110 11100	001110 0110	001110 0110
D29.6	DD	110 11101	101110 0110	010001 0110
D30.6	DE	110 11110	011110 0110	100001 0110
D31.6	DF	110 11111	101011 0110	010100 0110
D0.7	E0	111 00000	100111 0001	011000 1110
D1.7	E1	111 00001	011101 0001	100010 1110
D2.7	E2	111 00010	101101 0001	010010 1110
D3.7	E3	111 00011	110001 1110	110001 0001
D4.7	E4	111 00100	110101 0001	001010 1110
D5.7	E5	111 00101	101001 1110	101001 0001
D6.7	E6	111 00110	011001 1110	011001 0001
D7.7	E7	111 00111	111000 1110	000111 0001
D8.7	E8	111 01000	111001 0001	000110 1110
D9.7	E9	111 01001	100101 1110	100101 0001
D10.7	EA	111 01010	010101 1110	010101 0001
D11.7	EB	111 01011	110100 1110	110100 1000
D12.7	EC	111 01100	001101 1110	001101 0001
D13.7	ED	111 01101	101100 1110	101100 1000
D14.7	EE	111 01110	011100 1110	011100 1000
D15.7	EF	111 01111	010111 0001	101000 1110
D16.7	F0	111 10000	011011 0001	100100 1110
D17.7	F1	111 10001	100011 0111	100011 0001
D18.7	F2	111 10010	010011 0111	010011 0001
D19.7	F3	111 10011	110010 1110	110010 0001
D20.7	F4	111 10100	001011 0111	001011 0001
D21.7	F5	111 10101	101010 1110	101010 0001
D22.7	F6	111 10110	011010 1110	011010 0001

Table 6–3. Valid Data Code-Groups (Part 9 of 9)

Code-Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD- abcdei fghj	Current RD+ abcdei fghj
D23.7	F7	111 10111	111010 0001	000101 1110
D24.7	F8	111 11000	110011 0001	001100 1110
D25.7	F9	111 11001	100110 1110	100110 0001
D26.7	FA	111 11010	010110 1110	010110 0001
D27.7	FB	111 11011	110110 0001	001001 1110
D28.7	FC	111 11100	001110 1110	001110 0001
D29.7	FD	111 11101	101110 0001	010001 1110
D30.7	FE	111 11110	011110 0001	100001 1110
D31.7	FF	111 11111	101011 0001	010100 1110

Document Revision History

Table 6–4 shows the revision history for this chapter.

Table 6–4. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007 v3.1	Updated Table 6–1.	—
No change	Formerly chapter 5. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
February 2007 v3.0	Removed the Stratix II GX Feature Versus Protocol Mode Matrix table.	—
	Added the "Document Revision History" section to this chapter.	—
	Updated Table 6–1.	—
February 2006, v2.1	• Changed to chapter 5. • Updated Tables 5–1 and 5–2.	—
December 2005, v2.0	• Removed "Appendix" from chapter title. • Updated Tables 5–1 and 5–2.	—
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	—

Section II. Clock Management

This section provides information on clock management in Stratix® II GX devices. It describes the enhanced and fast phase-locked loops (PLLs) that support clock management and synthesis for on-chip clock management, external system clock management, and high-speed I/O interfaces.

This section includes the following chapter:

- Chapter 7, PLLs in Stratix II and Stratix II GX Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

7. PLLs in Stratix II and Stratix II GX Devices

SII52001-4.5

Introduction

Stratix® II and Stratix II GX device phase-locked loops (PLLs) provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces. Stratix II devices have up to 12 PLLs, and Stratix II GX devices have up to 8 PLLs. Stratix II and Stratix II GX PLLs are highly versatile and can be used as a zero delay buffer, a jitter attenuator, low skew fan out buffer, or a frequency synthesizer.

Stratix II and Stratix II GX devices feature both enhanced PLLs and fast PLLs. Stratix II and Stratix II GX devices have up to four enhanced PLLs. Stratix II devices have up to eight fast PLLs and Stratix II GX devices have up to four PLLs. Both enhanced and fast PLLs are feature rich, supporting advanced capabilities such as clock switchover, reconfigurable phase shift, PLL reconfiguration, and reconfigurable bandwidth. PLLs can be used for general-purpose clock management, supporting multiplication, phase shifting, and programmable duty cycle. In addition, enhanced PLLs support external clock feedback mode, spread-spectrum clocking, and counter cascading. Fast PLLs offer high speed outputs to manage the high-speed differential I/O interfaces.

Stratix II and Stratix II GX devices also support a power-down mode where clock networks that are not being used can easily be turned off, reducing the overall power consumption of the device. In addition, Stratix II and Stratix II GX PLLs support dynamic selection of the PLL input clock from up to five possible sources, giving you the flexibility to choose from multiple (up to four) clock sources to feed the primary and secondary clock input ports.

The Altera® Quartus® II software enables the PLLs and their features without requiring any external devices.

Tables 7–1 and 7–2 show the PLLs available for each Stratix II and Stratix II GX device, respectively.

Table 7–1. Stratix II Device PLL Availability *Note (1)*

Device	Fast PLLs								Enhanced PLLs			
	1	2	3	4	7	8	9	10	5	6	11	12
EP2S15	✓	✓	✓	✓					✓	✓		
EP2S30	✓	✓	✓	✓					✓	✓		
EP2S60	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP2S90 *(2)*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP2S130 *(3)*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP2S180	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notes for Table 7–1:

(1) The EP2S60 device in the 1,020-pin package contains 12 PLLs. EP2S60 devices in the 484-pin and 672-pin packages contain fast PLLs 1–4 and enhanced PLLs 5 and 6.

(2) EP2S90 devices in the 1020-pin and 1508-pin packages contain 12 PLLs. EP2S90 devices in the 484-pin and 780-pin packages contain fast PLLs 1–4 and enhanced PLLs 5 and 6.

(3) EP2S130 devices in the 1020-pin and 1508-pin packages contain 12 PLLs. The EP2S130 device in the 780-pin package contains fast PLLs 1–4 and enhanced PLLs 5 and 6.

Table 7–2. Stratix II GX Device PLL Availability *Note (1)*

Device	Fast PLLs								Enhanced PLLs			
	1	2	3 *(3)*	4 *(3)*	7	8	9 *(3)*	10 *(3)*	5	6	11	12
EP2SGX30 *(2)*	✓	✓							✓	✓		
EP2SGX60 *(2)*	✓	✓			✓	✓			✓	✓	✓	✓
EP2SGX90	✓	✓			✓	✓			✓	✓	✓	✓
EP2SGX130	✓	✓			✓	✓			✓	✓	✓	✓

Notes for Table 7–2:

(1) The global or regional clocks in a fast PLL's transceiver block can drive the fast PLL input. A pin or other PLL must drive the global or regional source. The source cannot be driven by internally generated logic before driving the fast PLL.

(2) EP2SGX30C and EP2SGX60C devices only have two fast PLLs (PLLs 1 and 2), but the connectivity from these two PLLs to the global and regional clock networks remains the same as shown in this table.

(3) PLLs 3, 4, 9, and 10 are not available in Stratix II GX devices. however, these PLLs are listed in Table 7–2 because the Stratix II GX PLL numbering scheme is consistent with Stratix and Stratix II devices.

Table 7–3 shows the enhanced PLL and fast PLL features in Stratix II and Stratix II GX devices.

Table 7–3. Stratix II and Stratix II GX PLL Features

Feature	Enhanced PLL	Fast PLL
Clock multiplication and division	*m*/(*n* × post-scale counter) *(1)*	*m*/(*n* × post-scale counter) *(2)*
Phase shift	Down to 125-ps increments *(3)*	Down to 125-ps increments *(3)*
Clock switchover	✓	✓ *(4)*
PLL reconfiguration	✓	✓
Reconfigurable bandwidth	✓	✓
Spread-spectrum clocking	✓	
Programmable duty cycle	✓	✓
Number of clock outputs per PLL *(5)*	6	4
Number of dedicated external clock outputs per PLL	Three differential or six single-ended	*(6)*
Number of feedback clock inputs per PLL	1 *(7)*	

Notes to Table 7–3:

(1) For enhanced PLLs, *m* and *n* range from 1 to 512 with 50% duty cycle. Post-scale counters range from 1 to 512 with 50% duty cycle. For non-50% duty-cycle clock outputs, post-scale counters range from 1 to 256.

(2) For fast PLLs, *n* can range from 1 to 4. The post-scale and *m* counters range from 1 to 32. For non-50% duty-cycle clock outputs, post-scale counters range from 1 to 16.

(3) The smallest phase shift is determined by the voltage controlled oscillator (VCO) period divided by eight. The supported phase-shift range is from 125 to 250 ps. Stratix II and Stratix II GX devices can shift all output frequencies in increments of at least 45° . Smaller degree increments are possible depending on the frequency and divide parameters. For non-50% duty cycle clock outputs post-scale counters range from 1 to 256.

(4) Stratix II and Stratix II GX fast PLLs only support manual clock switchover.

(5) The clock outputs can be driven to internal clock networks or to a pin.

(6) The PLL clock outputs of the fast PLLs can drive to any I/O pin to be used as an external clock output. For high-speed differential I/O pins, the device uses a data channel to generate the transmitter output clock (`txclkout`).

(7) If the design uses external feedback input pins, you will lose one (or two, if f_{BIN} is differential) dedicated output clock pin.

Figure 7–1 shows a top-level diagram of Stratix II device and PLL locations. Figure 7–2 shows a top-level diagram of Stratix II device and PLL locations. See "Clock Control Block" on page 7–86 for more detail on PLL connections to global and regional clocks networks.

Figure 7–1. Stratix II PLL Locations

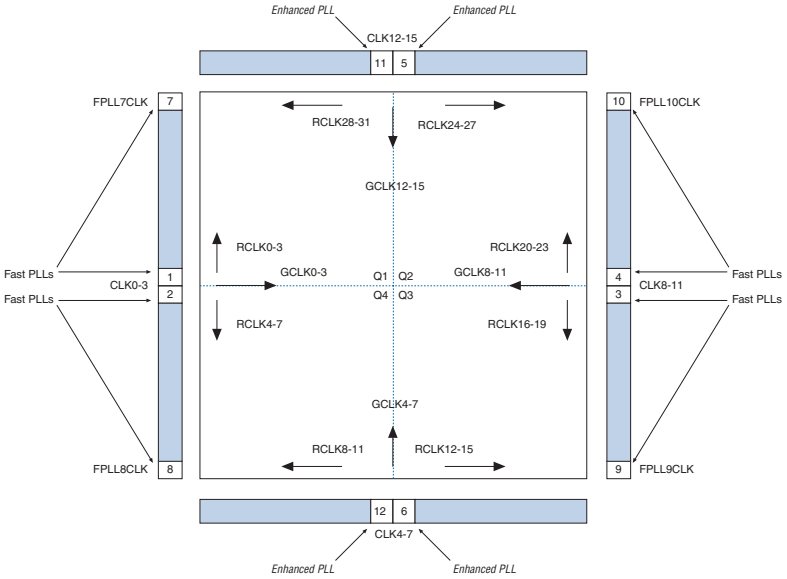
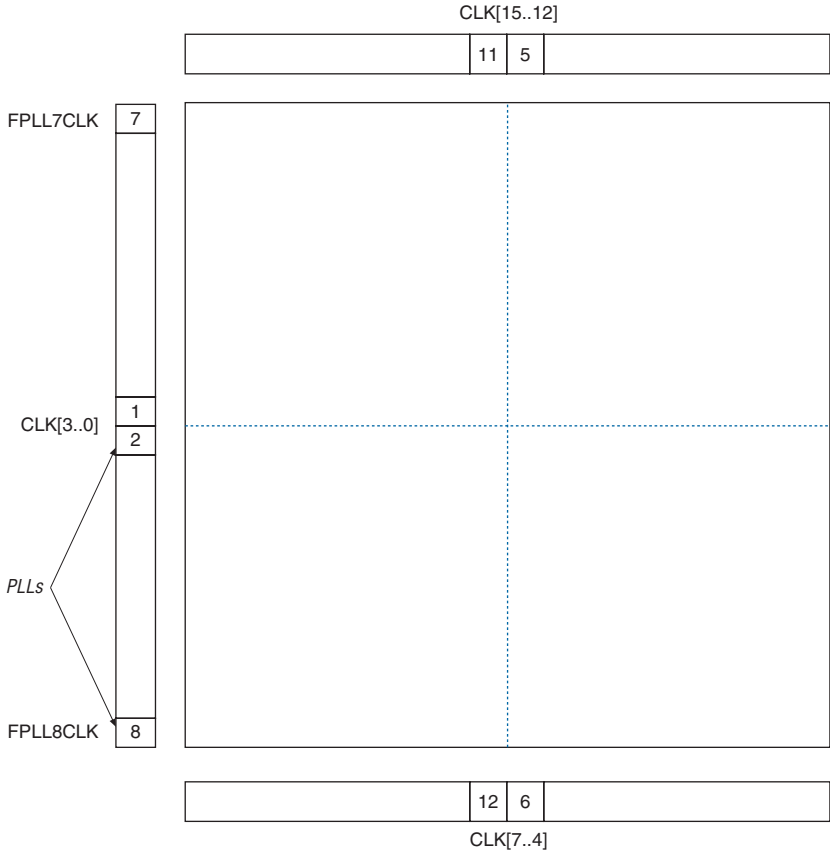


Figure 7–2. Stratix II GX PLL Locations



Enhanced PLLs

Stratix II and Stratix II GX devices contain up to four enhanced PLLs with advanced clock management features. The main goal of a PLL is to synchronize the phase and frequency of an internal and external clock to an input reference clock. There are a number of components that comprise a PLL to achieve this phase alignment.

Enhanced PLL Hardware Overview

Stratix II and Stratix II GX PLLs align the rising edge of the reference input clock to a feedback clock using the phase-frequency detector (PFD). The falling edges are determined by the duty-cycle specifications. The PFD produces an up or down signal that determines whether the VCO needs to operate at a higher or lower frequency.

The PFD output is applied to the charge pump and loop filter, which produces a control voltage for setting the VCO frequency. If the PFD produces an up signal, then the VCO frequency increases. A down signal decreases the VCO frequency. The PFD outputs these up and down signals to a charge pump. If the charge pump receives an up signal, current is driven into the loop filter. Conversely, if it receives a down signal, current is drawn from the loop filter.

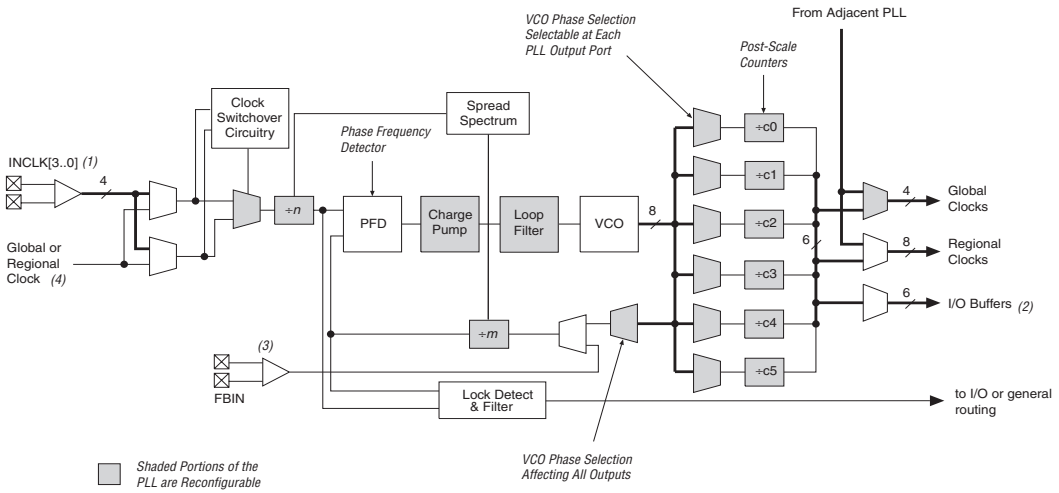
The loop filter converts these up and down signals to a voltage that is used to bias the VCO. The loop filter also removes glitches from the charge pump and prevents voltage over-shoot, which filters the jitter on the VCO.

The voltage from the loop filter determines how fast the VCO operates. The VCO is implemented as a four-stage differential ring oscillator. A divide counter (*m*) is inserted in the feedback loop to increase the VCO frequency above the input reference frequency. VCO frequency (f_{VCO}) is equal to (*m*) times the input reference clock (f_{REF}). The input reference clock (f_{REF}) to the PFD is equal to the input clock (f_{IN}) divided by the pre-scale counter (*n*). Therefore, the feedback clock (f_{FB}) applied to one input of the PFD is locked to the f_{REF} that is applied to the other input of the PFD.

The VCO output can feed up to six post-scale counters (`C0`, `C1`, `C2`, `C3`, `C4`, and `C5`). These post-scale counters allow a number of harmonically related frequencies to be produced within the PLL.

Figure 7–3 shows a simplified block diagram of the major components of the Stratix II and Stratix II GX enhanced PLL. Figure 7–4 shows the enhanced PLL's outputs and dedicated clock outputs.

Figure 7–3. Stratix II and Stratix II GX Enhanced PLL



Notes to Figure 7–3:

(1) Each clock source can come from any of the four clock pins located on the same side of the device as the PLL.

(2) PLLs 5, 6, 11, and 12 each have six single-ended dedicated clock outputs or three differential dedicated clock outputs.

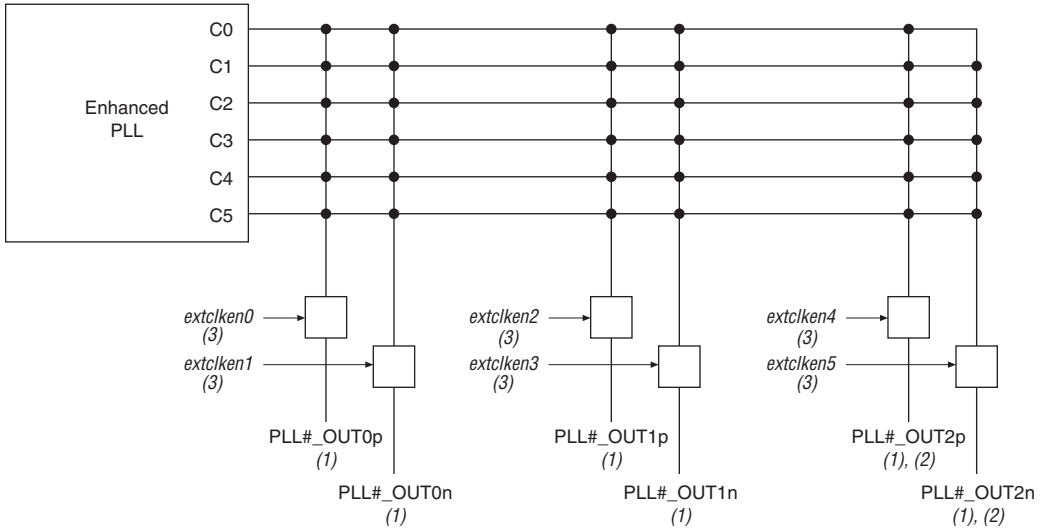
(3) If the design uses external feedback input pins, you will lose one (or two, if f_{BIN} is differential) dedicated output clock pin. Every Stratix II and Stratix II GX device has at least two enhanced PLLs with one single-ended or differential external feedback input per PLL.

(4) The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

External Clock Outputs

Enhanced PLLs 5, 6, 11, and 12 each support up to six single-ended clock outputs (or three differential pairs). See Figure 7–4.

Figure 7–4. External Clock Outputs for Enhanced PLLs 5, 6, 11 and 12



Notes to Figure 7–4:

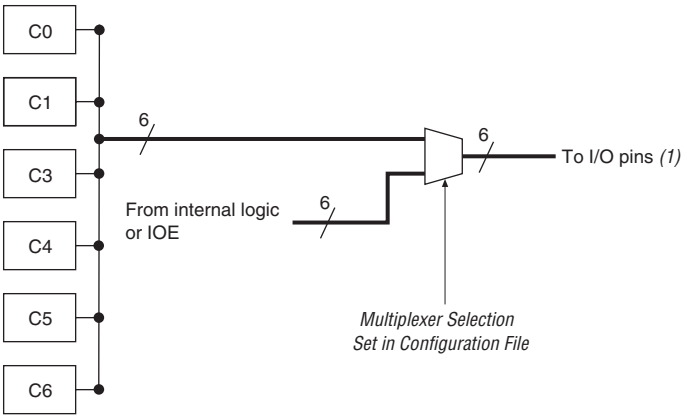
(1) These clock output pins can be fed by any one of the `C[5..0]` counters.

(2) These clock output pins are used as either external clock outputs or for external feedback. If the design uses external feedback input pins, you will lose one (or two, if f_{BIN} is differential) dedicated output clock pin.

(3) These external clock enable signals are available only when using the `altclkctrl` megafunction.

Any of the six output counters `C[5..0]` can feed the dedicated external clock outputs, as shown in Figure 7–5. Therefore, one counter or frequency can drive all output pins available from a given PLL. The dedicated output clock pins (`PLL_OUT`) from each enhanced PLL are powered by a separate power pin (e.g., `VCC_PLL5_OUT`, `VCC_PLL6_OUT`, etc.), reducing the overall output jitter by providing improved isolation from switching I/O pins.

Figure 7–5. External Clock Output Connectivity to PLL Output Counters for Enhanced PLLs 5, 6, 11 and 12 Note (1)



Note to Figure 7–5:

(1) The design can use each external clock output pin as a general-purpose output pin from the logic array. These pins are multiplexed with I/O element (IOE) outputs.

Each pin of a single-ended output pair can either be in phase or 180° out of phase. The Quartus II software places the NOT gate in the design into the IOE to implement 180° phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins (in the top and bottom banks) as well as LVDS, LVPECL, differential HSTL, and differential SSTL. See Table 7–6, in the "Enhanced PLL Pins" section on page 7–12 to determine which I/O standards the enhanced PLL clock pins support.

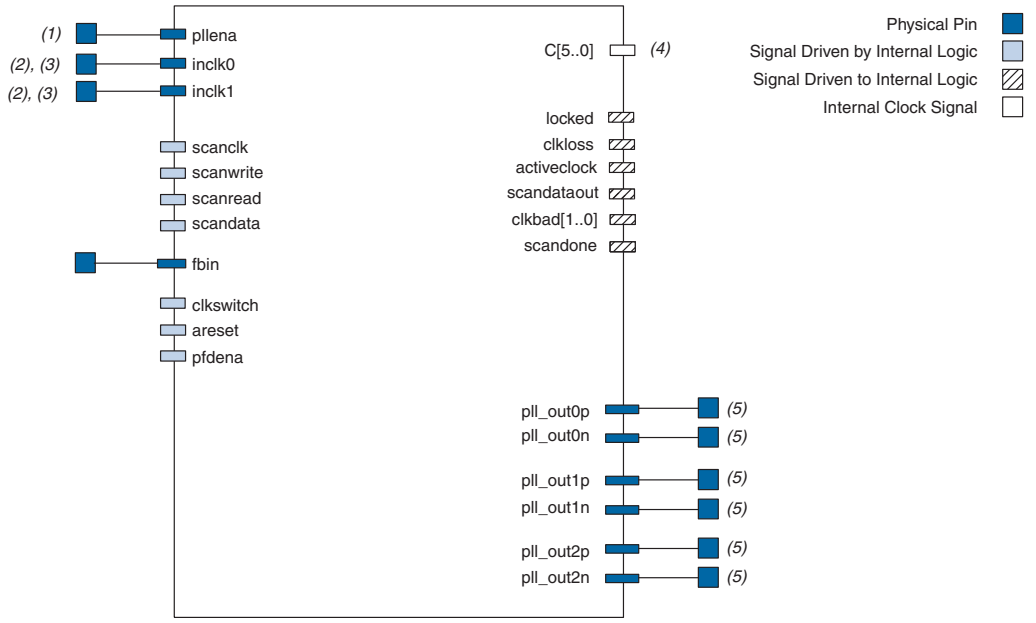
When in single-ended or differential mode, one power pin supports six single-ended or three differential outputs. Both outputs use the same I/O standard in single-ended mode to maintain performance. You can also use the external clock output pins as user output pins if external enhanced PLL clocking is not needed.

The enhanced PLL can also drive out to any regular I/O pin through the global or regional clock network.

Enhanced PLL Software Overview

Stratix II and Stratix II GX enhanced PLLs are enabled in the Quartus II software by using the altpll megafunction. Figure 7–6 shows the available ports (as they are named in the Quartus II altpll megafunction) of the Stratix II and Stratix II GX enhanced PLL.

Figure 7–6. Enhanced PLL Ports



Notes to Figure 7–6:

(1) Enhanced and fast PLLs share this input pin.
(2) These are either single-ended or differential pins.
(3) The primary and secondary clock input can be fed from any one of four clock pins located on the same side of the device as the PLL.
(4) Can drive to the global or regional clock networks or the dedicated external clock output pins.
(5) These dedicated output clocks are fed by the `C[5..0]` counters.

Tables 7–4 and 7–5 describe all the enhanced PLL ports.

Table 7–4. Enhanced PLL Input Signals (Part 1 of 2)

Port	Description	Source	Destination
`inclk0`	Primary clock input to the PLL.	Pin or another PLL	*n* counter
`inclk1`	Secondary clock input to the PLL.	Pin or another PLL	*n* counter
`fbin`	External feedback input to the PLL.	Pin	PFD
`pllena`	Enable pin for enabling or disabling all or a set of PLLs. Active high.	Pin	General PLL control signal
`clkswitch`	Switch-over signal used to initiate external clock switch-over control. Active high.	Logic array	PLL switch-over circuit

Table 7–4. Enhanced PLL Input Signals (Part 2 of 2)

Port	Description	Source	Destination
areset	Signal used to reset the PLL which resynchronizes all the counter outputs. Active high.	Logic array	General PLL control signal
pfdena	Enables the outputs from the phase frequency detector. Active high.	Logic array	PFD
scanclk	Serial clock signal for the real-time PLL reconfiguration feature.	Logic array	Reconfiguration circuit
scandata	Serial input data stream for the real-time PLL reconfiguration feature.	Logic array	Reconfiguration circuit
scanwrite	Enables writing the data in the scan chain into the PLL. Active high.	Logic array	Reconfiguration circuit
scanread	Enables scan data to be written into the scan chain. Active high.	Logic array	Reconfiguration circuit

Table 7–5. Enhanced PLL Output Signals (Part 1 of 2)

Port	Description	Source	Destination
c[5..0]	PLL output counters driving regional, global or external clocks.	PLL counter	Internal or external clock
pll_out [2..0]p pll_out [2..0]n	These are three differential or six single-ended external clock output pins fed from the C[5..0] PLL counters, and every output can be driven by any counter. *p* and *n* are the positive (*p*) and negative (*n*) pins for differential pins.	PLL counter	Pin(s)
clkloss	Signal indicating the switch-over circuit detected a switch-over condition.	PLL switch-over circuit	Logic array
clkbad[1..0]	Signals indicating which reference clock is no longer toggling. clkbad1 indicates inclk1 status, clkbad0 indicates inclk0 status. 1= good; 0=bad	PLL switch-over circuit	Logic array
locked	Lock or gated lock output from lock detect circuit. Active high.	PLL lock detect	Logic array
activeclock	Signal to indicate which clock (0 = inclk0 or 1 = inclk1) is driving the PLL. If this signal is low, inclk0 drives the PLL, If this signal is high, inclk1 drives the PLL	PLL clock multiplexer	Logic array

Table 7–5. Enhanced PLL Output Signals (Part 2 of 2)			
Port	**Description**	**Source**	**Destination**
scandataout	Output of the last shift register in the scan chain.	PLL scan chain	Logic array
scandone	Signal indicating when the PLL has completed reconfiguration. 1 to 0 transition indicates that the PLL has been reconfigured.	PLL scan chain	Logic array

Enhanced PLL Pins

Table 7–6 lists the I/O standards support by the enhanced PLL clock outputs.

Table 7–6. I/O Standards Supported for Enhanced PLL Pins (Part 1 of 2) Note (1)			
I/O Standard	**Input**		**Output**
	INCLK	**FBIN**	**EXTCLK**
LVTTL	✓	✓	✓
LVCMOS	✓	✓	✓
2.5 V	✓	✓	✓
1.8 V	✓	✓	✓
1.5 V	✓	✓	✓
3.3-V PCI	✓	✓	✓
3.3-V PCI-X	✓	✓	✓
SSTL-2 Class I	✓	✓	✓
SSTL-2 Class II	✓	✓	✓
SSTL-18 Class I	✓	✓	✓
SSTL-18 Class II	✓	✓	✓
1.8-V HSTL Class I	✓	✓	✓
1.8-V HSTL Class II	✓	✓	✓
1.5-V HSTL Class I	✓	✓	✓
1.5-V HSTL Class II	✓	✓	✓
1.2-V HSTL Class I	✓	✓	✓
1.2-V HSTL Class II	✓	✓	✓

Table 7–6. I/O Standards Supported for Enhanced PLL Pins (Part 2 of 2) *Note (1)*

I/O Standard	Input		Output
	INCLK	FBIN	EXTCLK
Differential SSTL-2 Class I	✓	✓	✓
Differential SSTL-2 Class II	✓	✓	✓
Differential SSTL-18 Class I	✓	✓	✓
Differential SSTL-18 Class II	✓	✓	✓
1.8-V differential HSTL Class I	✓	✓	✓
1.8-V differential HSTL Class II	✓	✓	✓
1.5-V differential HSTL Class I	✓	✓	✓
1.5-V differential HSTL Class II	✓	✓	✓
LVDS	✓	✓	✓
HyperTransport technology			
Differential LVPECL	✓	✓	✓

Note to Table 7–6:
(1) The enhanced PLL external clock output bank does not allow a mixture of both single-ended and differential I/O standards.

Table 7–7 shows the physical pins and their purpose for the Stratix II and Stratix II GX enhanced PLLs. For `inclk` port connections to pins see "Clock Control Block" on page 7–86.

Table 7–7. Stratix II and Stratix II GX Enhanced PLL Pins (Part 1 of 3) *Note (1)*

Pin	Description
CLK4p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 6 or 12.
CLK5p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 6 or 12.
CLK6p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 6 or 12.
CLK7p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 6 or 12.
CLK12p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 5 or 11.
CLK13p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 5 or 11.
CLK14p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 5 or 11.
CLK15p/n	Single-ended or differential pins that can drive the `inclk` port for PLLs 5 or 11.
PLL5_FBp/n	Single-ended or differential pins that can drive the `fbin` port for PLL 5.

Table 7–7. Stratix II and Stratix II GX Enhanced PLL Pins (Part 2 of 3) *Note (1)*

Pin	Description
PLL6_FBp/n	Single-ended or differential pins that can drive the fbin port for PLL 6.
PLL11_FBp/n	Single-ended or differential pins that can drive the fbin port for PLL 11.
PLL12_FBp/n	Single-ended or differential pins that can drive the fbin port for PLL 12.
PLL_ENA	Dedicated input pin that drives the pllena port of all or a set of PLLs. If you do not use this pin, connect it to ground.
PLL5_OUT[2..0]p/n	Single-ended or differential pins driven by C[5..0] ports from PLL 5.
PLL6_OUT[2..0]p/n	Single-ended or differential pins driven by C[5..0] ports from PLL 6.
PLL11_OUT[2..0]p/n	Single-ended or differential pins driven by C[5..0] ports from PLL 11.
PLL12_OUT[2..0]p/n	Single-ended or differential pins driven by C[5..0] ports from PLL 12.
VCCA_PLL5	Analog power for PLL 5. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL5	Analog ground for PLL 5. You can connect this pin to the GND plane on the board.
VCCA_PLL6	Analog power for PLL 6. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL6	Analog ground for PLL 6. You can connect this pin to the GND plane on the board.
VCCA_PLL11	Analog power for PLL 11. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL11	Analog ground for PLL 11. You can connect this pin to the GND plane on the board.
VCCA_PLL12	Analog power for PLL 12. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL12	Analog ground for PLL 12. You can connect this pin to the GND plane on the board.
VCCD_PLL	Digital power for PLLs. You must connect this pin to 1.2 V, even if the PLL is not used.
VCC_PLL5_OUT	External clock output V_{CCIO} power for PLL5_OUT0p, PLL5_OUT0n, PLL5_OUT1p, PLL5_OUT1n, PLL5_OUT2p, and PLL5_OUT2n outputs from PLL 5.
VCC_PLL6_OUT	External clock output V_{CCIO} power for PLL6_OUT0p, PLL6_OUT0n, PLL6_OUT1p, PLL6_OUT1n and PLL6_OUT2p, PLL6_OUT2n outputs from PLL 6.
VCC_PLL11_OUT	External clock output V_{CCIO} power for PLL11_OUT0p, PLL11_OUT0n, PLL11_OUT1p, PLL11_OUT1n and PLL11_OUT2p, PLL11_OUT2n outputs from PLL 11.

Table 7–7. Stratix II and Stratix II GX Enhanced PLL Pins (Part 3 of 3) Note (1)	
Pin	**Description**
VCC_PLL12_OUT	External clock output V_{CCIO} power for PLL12_OUT0p, PLL12_OUT0n, PLL12_OUT1p, PLL12_OUT1n and PLL12_OUT2p, PLL12_OUT2n outputs from PLL 12.

Note to Table 7–7:
(1) The negative leg pins (CLKn, PLL_FBn, and PLL_OUTn) are only required with differential signaling.

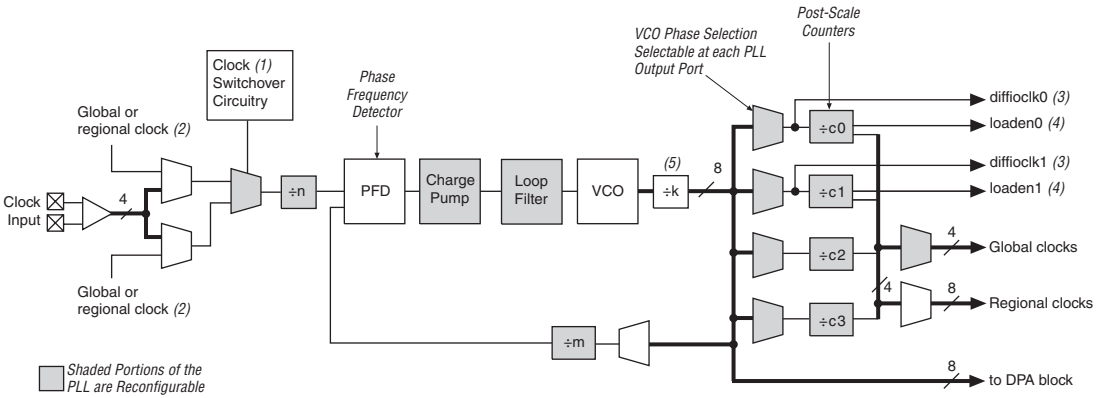
Fast PLLs

Stratix II devices contain up to eight fast PLLs and Stratix II GX devices contain up to four fast PLLs. Fast PLLs have high-speed differential I/O interface capability along with general-purpose features.

Fast PLL Hardware Overview

Figure 7–7 shows a diagram of the fast PLL.

Figure 7–7. Stratix II and Stratix II GX Fast PLL Block Diagram



Notes to Figure 7–7:
(1) Stratix II and Stratix II GX fast PLLs only support manual clock switchover.
(2) The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.
(3) In high-speed differential I/O support mode, this high-speed PLL clock feeds the SERDES. Stratix II devices only support one rate of data transfer per fast PLL in high-speed differential I/O support mode.
(4) This signal is a high-speed differential I/O support SERDES control signal.
(5) If the design enables this ÷2 counter, then the device can use a VCO frequency range of 150 to 520 MHz.

External Clock Outputs

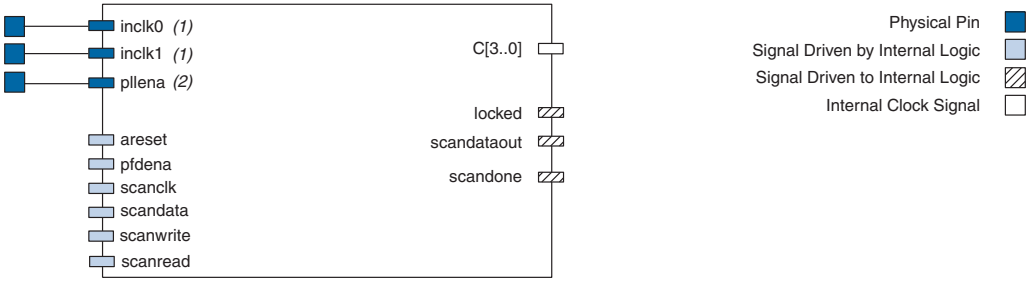
Each fast PLL supports differential or single-ended outputs for source-synchronous transmitters or for general-purpose external clocks. There are no dedicated external clock output pins. The fast PLL global or regional outputs can drive any I/O pin as an external clock output pin. The I/O standards supported by any particular bank determines what standards are possible for an external clock output driven by the fast PLL in that bank.

For more information, see the *Selectable I/O Standards in Stratix II and Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* (or the *Stratix II Device Handbook*).

Fast PLL Software Overview

Stratix II and Stratix II GX fast PLLs are enabled in the Quartus II software by using the `altpll` megafunction. Figure 7–8 shows the available ports (as they are named in the Quartus II `altpll` megafunction) of the Stratix II or Stratix II GX fast PLL.

Figure 7–8. Stratix II and Stratix II GX Fast PLL Ports and Physical Destinations



Notes to Figure 7–8:
(1) This input pin is either single-ended or differential.
(2) This input pin is shared by all enhanced and fast PLLs.

Tables 7–8 and 7–9 show the description of all fast PLL ports.

Table 7–8. Fast PLL Input Signals (Part 1 of 2)

Name	Description	Source	Destination
`inclk0`	Primary clock input to the fast PLL.	Pin or another PLL	*n* counter
`inclk1`	Secondary clock input to the fast PLL.	Pin or another PLL	*n* counter

Table 7–8. Fast PLL Input Signals (Part 2 of 2)

Name	Description	Source	Destination
pllena	Enable pin for enabling or disabling all or a set of PLLs. Active high.	Pin	PLL control signal
clkswitch	Switch-over signal used to initiate external clock switch-over control. Active high.	Logic array	Reconfiguration circuit
areset	Enables the up/down outputs from the phase-frequency detector. Active high.	Logic array	PLL control signal
pfdena	Enables the up/down outputs from the phase-frequency detector. Active high.	Logic array	PFD
scanclk	Serial clock signal for the real-time PLL control feature.	Logic array	Reconfiguration circuit
scandata	Serial input data stream for the real-time PLL control feature.	Logic array	Reconfiguration circuit
scanwrite	Enables writing the data in the scan chain into the PLL Active high.	Logic array	Reconfiguration circuit
scanread	Enables scan data to be written into the scan chain Active high.	Logic array	Reconfiguration circuit

Table 7–9. Fast PLL Output Signals

Name	Description	Source	Destination
c[3..0]	PLL outputs driving regional or global clock.	PLL counter	Internal clock
locked	Lock or gated lock output from lock detect circuit. Active high.	PLL lock detect	Logic array
scandataout	Output of the last shift register in the scan chain.	PLL scan chain	Logic array
scandone	Signal indicating when the PLL has completed reconfiguration. 1 to 0 transition indicates the PLL has been reconfigured.	PLL scan chain	Logic array

Fast PLL Pins

Table 7–10 shows the I/O standards supported by the fast PLL input pins.

Table 7–10. I/O Standards Supported for Stratix II and Stratix II GX Fast PLL Pins

I/O Standard	INCLK
LVTTL	✓
LVCMOS	✓
2.5 V	✓
1.8 V	✓
1.5 V	✓
3.3-V PCI	
3.3-V PCI-X	
SSTL-2 Class I	✓
SSTL-2 Class II	✓
SSTL-18 Class I	✓
SSTL-18 Class II	✓
1.8-V HSTL Class I	✓
1.8-V HSTL Class II	✓
1.5-V HSTL Class I	✓
1.5-V HSTL Class II	✓
Differential SSTL-2 Class I	
Differential SSTL-2 Class II	
Differential SSTL-18 Class I	
Differential SSTL-18 Class II	
1.8-V differential HSTL Class I	
1.8-V differential HSTL Class II	
1.5-V differential HSTL Class I	
1.5-V differential HSTL Class II	
LVDS	✓
HyperTransport technology	✓
Differential LVPECL	

Table 7–11 shows the physical pins and their purpose for the fast PLLs. For inclk port connections to pins, see "Clocking" on page 7–62.

Table 7–11. Fast PLL Pins (Part 1 of 2) *Note (1)*

Pin	Description
CLK0p/n	Single-ended or differential pins that can drive the inclk port for PLLs 1, 2, 7 or 8.
CLK1p/n	Single-ended or differential pins that can drive the inclk port for PLLs 1, 2, 7 or 8.
CLK2p/n	Single-ended or differential pins that can drive the inclk port for PLLs 1, 2, 7 or 8.
CLK3p/n	Single-ended or differential pins that can drive the inclk port for PLLs 1, 2, 7 or 8.
CLK8p/n	Single-ended or differential pins that can drive the inclk port for PLLs 3, 4, 9 or 10.
CLK9p/n	Single-ended or differential pins that can drive the inclk port for PLLs 3, 4, 9 or 10.
CLK10p/n	Single-ended or differential pins that can drive the inclk port for PLLs 3, 4, 9 or 10.
CLK11p/n	Single-ended or differential pins that can drive the inclk port for PLLs 3, 4, 9 or 10.
FPLL7CLKp/n	Single-ended or differential pins that can drive the inclk port for PLL 7.
FPLL8CLKp/n	Single-ended or differential pins that can drive the inclk port for PLL 8.
FPLL9CLKp/n	Single-ended or differential pins that can drive the inclk port for PLL 9.
FPLL10CLKp/n	Single-ended or differential pins that can drive the inclk port for PLL 10.
PLL_ENA	Dedicated input pin that drives the pllena port of all or a set of PLLs. If you do not use this pin, connect it to GND.
VCCD_PLL	Digital power for PLLs. You must connect this pin to 1.2 V, even if the PLL is not used.
VCCA_PLL1	Analog power for PLL 1. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL1	Analog ground for PLL 1. Your can connect this pin to the GND plane on the board.
VCCA_PLL2	Analog power for PLL 2. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL2	Analog ground for PLL 2. You can connect this pin to the GND plane on the board.
VCCA_PLL3	Analog power for PLL 3. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL3	Analog ground for PLL 3. You can connect this pin to the GND plane on the board.
VCCA_PLL4	Analog power for PLL 4. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL4	Analog ground for PLL 4. You can connect this pin to the GND plane on the board.
GNDA_PLL7	Analog ground for PLL 7. You can connect this pin to the GND plane on the board.
VCCA_PLL8	Analog power for PLL 8. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL8	Analog ground for PLL 8. You can connect this pin to the GND plane on the board.
VCCA_PLL9	Analog power for PLL 9. You must connect this pin to 1.2 V, even if the PLL is not used.
GNDA_PLL9	Analog ground for PLL 9. You can connect this pin to the GND plane on the board.
VCCA_PLL10	Analog power for PLL 10. You must connect this pin to 1.2 V, even if the PLL is not used.

Table 7–11. Fast PLL Pins (Part 2 of 2) Note (1)	
Pin	**Description**
GNDA_PLL10	Analog ground for PLL 10. You can connect this pin to the GND plane on the board.

Note to Table 7–11:
(1) The negative leg pins (CLKn and FPLL_CLKn) are only required with differential signaling.

Clock Feedback Modes

Stratix II and Stratix II GX PLLs support up to five different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle. Each PLL must be driven by one of its own dedicated clock input pins for proper clock compensation. The clock input pin connections for each PLL are listed in Table 7–20 on page 7–70.

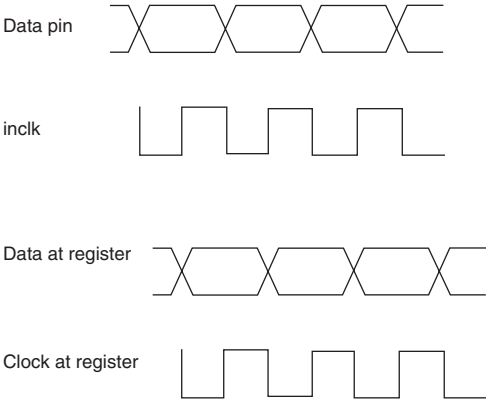
Table 7–12 shows which modes are supported by which PLL type.

Table 7–12. Clock Feedback Mode Availability		
Clock Feedback Mode	**Mode Available in**	
	Enhanced PLLs	**Fast PLLs**
Source synchronous mode	Yes	Yes
No compensation mode	Yes	Yes
Normal mode	Yes	Yes
Zero delay buffer mode	Yes	No
External feedback mode	Yes	No

Source-Synchronous Mode

If data and clock arrive at the same time at the input pins, they are guaranteed to keep the same phase relationship at the clock and data ports of any IOE input register. Figure 7–9 shows an example waveform of the clock and data in this mode. This mode is recommended for source-synchronous data transfers. Data and clock signals at the IOE experience similar buffer delays as long as the same I/O standard is used.

Figure 7–9. Phase Relationship Between Clock and Data in Source-Synchronous Mode



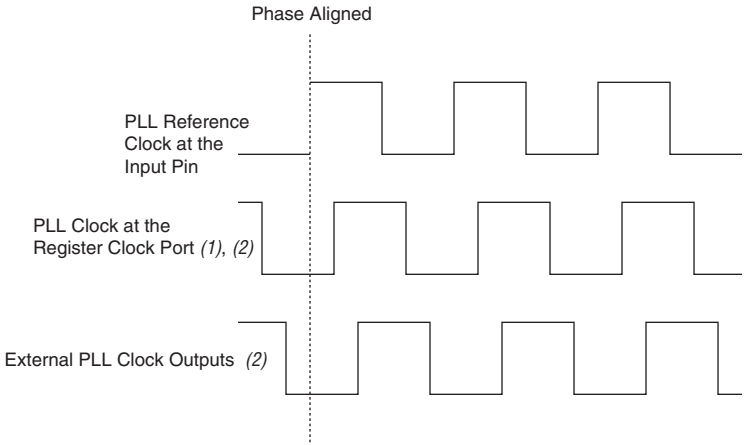
In source-synchronous mode, enhanced PLLs compensate for clock delay to the top and bottom IO registers and fast PLLs compensate for clock delay to the side IO registers. While implementing source-synchronous receivers in these IO banks, use the corresponding PLL type for best matching between clock and data delays (from input pins to register ports).

Set the input pin to the register delay chain within the IOE to zero in the Quartus II software for all data pins clocked by a source-synchronous mode PLL.

No Compensation Mode

In this mode, the PLL does not compensate for any clock networks. This provides better jitter performance because the clock feedback into the PFD does not pass through as much circuitry. Both the PLL internal and external clock outputs are phase shifted with respect to the PLL clock input. Figure 7–10 shows an example waveform of the PLL clocks' phase relationship in this mode.

Figure 7–10. Phase Relationship between PLL Clocks in No Compensation Mode

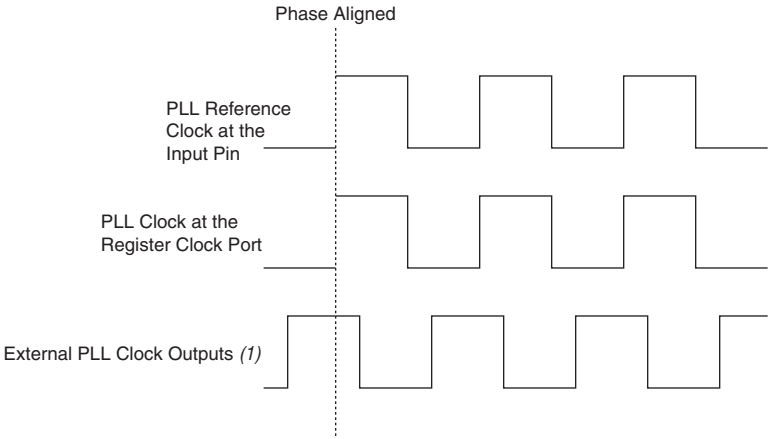


Notes to Figure 7–10.
(1) Internal clocks fed by the PLL are phase-aligned to each other.
(2) The PLL clock outputs can lead or lag the PLL input clocks.

Normal Mode

An internal clock in normal mode is phase-aligned to the input clock pin. The external clock output pin will have a phase delay relative to the clock input pin if connected in this mode. In normal mode, the delay introduced by the GCLK or RCLK network is fully compensated. Figure 7–11 shows an example waveform of the PLL clocks' phase relationship in this mode.

Figure 7–11. Phase Relationship Between PLL Clocks in Normal Mode

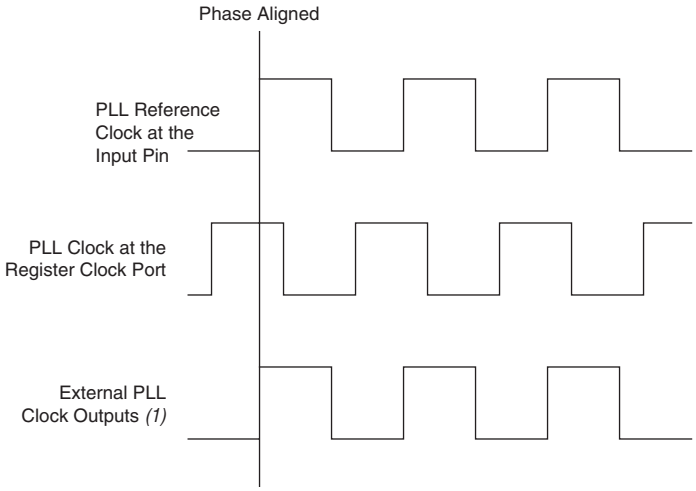


Note to Figure 7–11:
(1) The external clock output can lead or lag the PLL internal clock signals.

Zero Delay Buffer Mode

In the zero delay buffer mode, the external clock output pin is phase-aligned with the clock input pin for zero delay through the device. Figure 7–12 shows an example waveform of the PLL clocks' phase relationship in this mode. When using this mode, Altera requires that you use the same I/O standard on the input clock, and output clocks. When using single-ended I/O standards, the `inclk` port of the PLL must be fed by the dedicated `CLKp` input pin.

Figure 7–12. Phase Relationship Between PLL Clocks in Zero Delay Buffer Mode

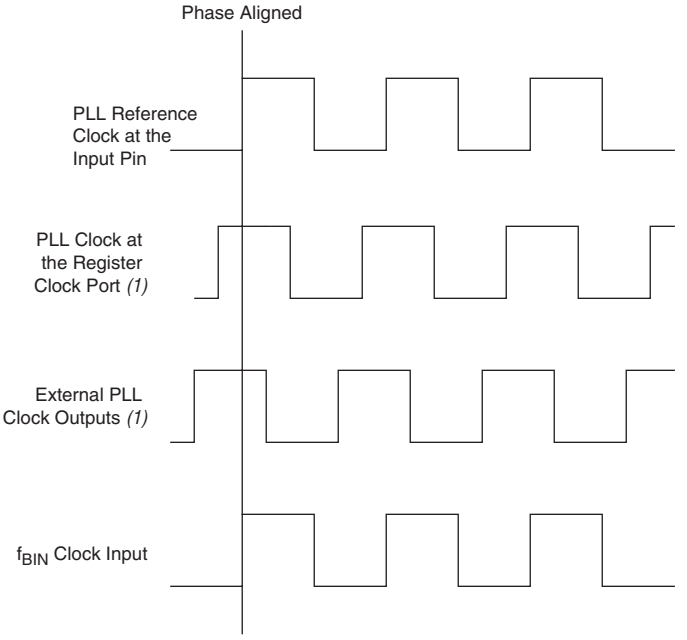


Note to Figure 7–12:
(1) The internal PLL clock output can lead or lag the external PLL clock outputs.

External Feedback Mode

In the external feedback mode, the external feedback input pin, `fbin`, is phase-aligned with the clock input pin, (see Figure 7–13). Aligning these clocks allows you to remove clock delay and skew between devices. This mode is possible on all enhanced PLLs. PLLs 5, 6, 11, and 12 support feedback for one of the dedicated external outputs, either one single-ended or one differential pair. In this mode, one *C* counter feeds back to the PLL `fbin` input, becoming part of the feedback loop. In this mode, you will be using one of the dedicated external clock outputs (two if a differential I/O standard is used) as the PLL `fbin` input pin. When using this mode, Altera requires that you use the same I/O standard on the input clock, feedback input, and output clocks. When using single-ended I/O standards, the `inclk` port of the PLL must be fed by the dedicated `CLKp` input pin.

Figure 7–13. Phase Relationship Between PLL Clocks in External Feedback Mode



Note to Figure 7–13:
(1) The PLL clock outputs can lead or lag the f_{BIN} clock input.

Hardware Features

Stratix II and Stratix II GX PLLs support a number of features for general-purpose clock management. This section discusses clock multiplication and division implementation, phase-shifting implementations and programmable duty cycles. Table 7–13 shows which feature is available in which type of Stratix II or Stratix II GX PLL.

Table 7–13. Stratix II and Stratix II GX PLL Hardware Features (Part 1 of 2)

Hardware Features	Availability	
	Enhanced PLL	Fast PLL
Clock multiplication and division	*m* (*n* × post-scale counter)	*m* (*n* × post-scale counter)
m counter value	Ranges from 1 through 512	Ranges from 1 through 32
n counter value	Ranges from 1 through 512	Ranges from 1 through 4
Post-scale counter values	Ranges from 1 through 512 *(1)*	Ranges from 1 through 32 *(2)*

Table 7–13. Stratix II and Stratix II GX PLL Hardware Features (Part 2 of 2)		
Hardware Features	Availability	
	Enhanced PLL	Fast PLL
Phase shift	Down to 125-ps increments (3)	Down to 125-ps increments (3)
Programmable duty cycle	Yes	Yes

Notes to Table 7–13:

(1) Post-scale counters range from 1 through 512 if the output clock uses a 50% duty cycle. For any output clocks using a non-50% duty cycle, the post-scale counters range from 1 through 256.

(2) Post-scale counters range from 1 through 32 if the output clock uses a 50% duty cycle. For any output clocks using a non-50% duty cycle, the post-scale counters range from 1 through 16.

(3) The smallest phase shift is determined by the VCO period divided by 8. For degree increments, the Stratix II device can shift all output frequencies in increments of at least 45° . Smaller degree increments are possible depending on the frequency and divide parameters.

Clock Multiplication and Division

Each Stratix II PLL provides clock synthesis for PLL output ports using $m/(n \times$ post-scale counter) scaling factors. The input clock is divided by a pre-scale factor, n, and is then multiplied by the m feedback factor. The control loop drives the VCO to match f_{IN} (m/n). Each output port has a unique post-scale counter that divides down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO is set to the least common multiple of the output frequencies that meets its frequency specifications. For example, if output frequencies required from one PLL are 33 and 66 MHz, then the Quartus II software sets the VCO to 660 MHz (the least common multiple of 33 and 66 MHz within the VCO range). Then, the post-scale counters scale down the VCO frequency for each output port.

There is one pre-scale counter, n, and one multiply counter, m, per PLL, with a range of 1 to 512 for both m and n in enhanced PLLs. For fast PLLs, m ranges from 1 to 32 while n ranges from 1 to 4. There are six generic post-scale counters in enhanced PLLs that can feed regional clocks, global clocks, or external clock outputs, all ranging from 1 to 512 with a 50% duty cycle setting for each PLL. The post-scale counters range from 1 to 256 with any non-50% duty cycle setting. In fast PLLs, there are four post-scale counters (`C0`, `C1`, `C2`, `C3`) for the regional and global clock output ports. All post-scale counters range from 1 to 32 with a 50% duty cycle setting. For non-50% duty cycle clock outputs, the post-scale counters range from 1 to 16. If the design uses a high-speed I/O interface, you can connect the dedicated `dffioclk` clock output port to allow the high-speed VCO frequency to drive the serializer/deserializer (SERDES).

The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the `altpll` megafunction.

Phase-Shift Implementation

Phase shift is used to implement a robust solution for clock delays in Stratix II and Stratix II GX devices. Phase shift is implemented by using a combination of the VCO phase output and the counter starting time. The VCO phase output and counter starting time is the most accurate method of inserting delays, since it is purely based on counter settings, which are independent of process, voltage, and temperature.

Stratix II and Stratix II GX PLLs do not support programmable delay elements because these delay elements require considerable area on the die and are sensitive to process, voltage, and temperature.

You can phase shift the output clocks from the Stratix II or Stratix II GX enhanced PLL in either:

- Fine resolution using VCO phase taps
- Coarse resolution using counter starting time

The VCO phase tap and counter starting time is implemented by allowing any of the output counters (`C[5..0]` or `m`) to use any of the eight phases of the VCO as the reference clock. This allows you to adjust the delay time with a fine resolution. The minimum delay time that you can insert using this method is defined by:

$$\Phi_{fine} = \frac{1}{8} T_{VCO} = \frac{1}{8 f_{VCO}} = \frac{N}{8 M f_{REF}}$$

where f_{REF} is input reference clock frequency.

For example, if f_{REF} is 100 MHz, *n* is 1, and *m* is 8, then f_{VCO} is 800 MHz and Φ_{FINE} equals 156.25 ps. This phase shift is defined by the PLL operating frequency, which is governed by the reference clock frequency and the counter settings.

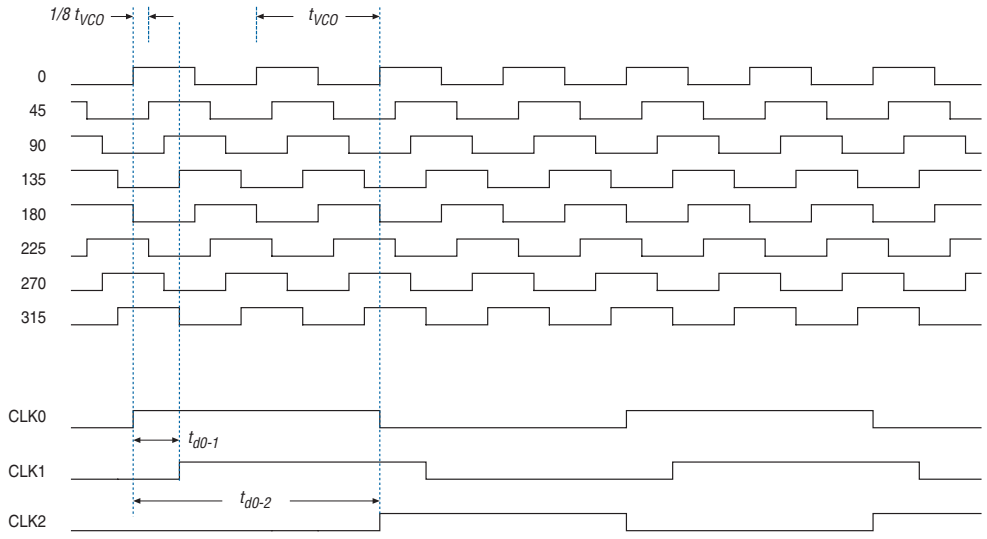
You can also delay the start of the counters for a predetermined number of counter clocks. You can express phase shift as:

$$\Phi_{coarse} = \frac{C - 1}{f_{Vco}} = \frac{(C - 1)N}{M f_{REF}}$$

where *C* is the count value set for the counter delay time, (this is the initial setting in the PLL usage section of the compilation report in the Quartus II software). If the initial value is 1, C – 1 = 0° phase shift.

Figure 7–14 shows an example of phase shift insertion using the fine resolution using VCO phase taps method. The eight phases from the VCO are shown and labeled for reference. For this example, CLK0 is based off the 0 phase from the VCO and has the *C* value for the counter set to one. The CLK1 signal is divided by four, two VCO clocks for high time and two VCO clocks for low time. CLK1 is based off the 135° phase tap from the VCO and also has the *C* value for the counter set to one. The CLK1 signal is also divided by 4. In this case, the two clocks are offset by 3 Φ_{FINE}. CLK2 is based off the 0phase from the VCO but has the *C* value for the counter set to three. This creates a delay of 2 Φ_{COARSE}, (two complete VCO periods).

Figure 7–14. Delay Insertion Using VCO Phase Output and Counter Delay Time



You can use the coarse and fine phase shifts as described above to implement clock delays in Stratix II and Stratix II GX devices. The phase-shift parameters are set in the Quartus II software.

Programmable Duty Cycle

The programmable duty cycle allows enhanced and fast PLLs to generate clock outputs with a variable duty cycle. This feature is supported on each enhanced and fast PLL post-scale counter `C[]`. The duty cycle setting is achieved by a low and high time count setting for the post-scale counters. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices. The post-scale counter value determines the precision of the duty cycle. The precision is defined by 50% divided by the post-scale counter value. The closest value to 100% is not achievable for a given counter value. For example, if the `C0` counter is ten, then steps of 5% are possible for duty cycle choices between 5 to 90%.

If the device uses external feedback, you must set the duty cycle for the counter driving the `fbin` pin to 50%. Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

Advanced Clear and Enable Control

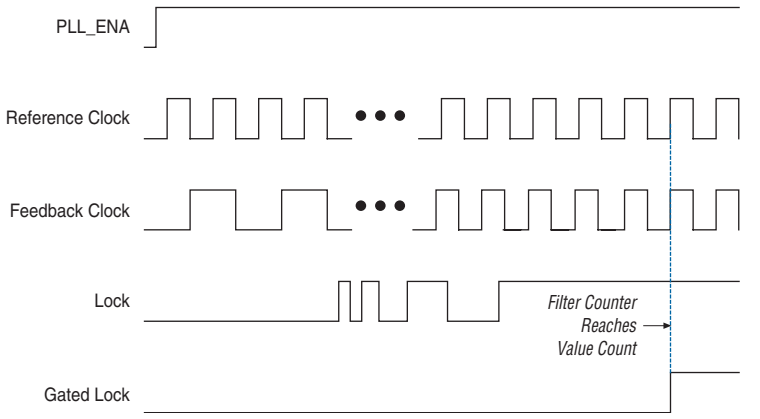
There are several control signals for clearing and enabling PLLs and their outputs. You can use these signals to control PLL resynchronization and gate PLL output clocks for low-power applications.

Enhanced Lock Detect Circuit

The lock output indicates that the PLL has locked onto the reference clock. Without any additional circuitry, the lock signal may toggle as the PLL begins tracking the reference clock. You may need to gate the lock signal for use as a system control. Either a gated lock signal or an ungated lock signal from the locked port can drive the logic array or an output pin. The Stratix II and Stratix II GX enhanced and fast PLLs include a programmable counter that holds the lock signal low for a user-selected number of input clock transitions. This allows the PLL to lock before enabling the lock signal. You can use the Quartus II software to set the 20-bit counter value.

Figure 7–15 shows the timing waveform for the lock and gated lock signals.

Figure 7–15. Timing Waveform for Lock and Gated Lock Signals



The device resets and enables both the counter and the PLL simultaneously when the `pllena` signal is asserted or the `areset` signal is de-asserted. Enhanced PLLs and fast PLLs support this feature. To ensure correct circuit operation, and to ensure that the output clocks have the correct phase relationship with respect to the input clock, Altera recommends that the input clock be running before the Stratix II device is finished configuring.

PLL_ENA

The `PLL_ENA` pin is a dedicated pin that enables or disables all PLLs on the Stratix II or Stratix II GX device. When the `PLL_ENA` pin is low, the clock output ports are driven low and all the PLLs go out of lock. When the `PLL_ENA` pin goes high again, the PLLs relock and resynchronize to the input clocks. You can choose which PLLs are controlled by the `pllena` signal by connecting the `pllena` input port of the `altpll` megafunction to the common `PLL_ENA` input pin.

Also, whenever the PLL loses lock for any reason (be it excessive `inclk` jitter, clock switchover, PLL reconfiguration, power supply noise, etc.), the PLL must be reset with the `areset` signal to guarantee correct phase relationship between the PLL output clocks. If the phase relationship between the input clock versus output clock, and between different output clocks from the PLL is not important in your design, the PLL need not be reset.

The level of the VCCSEL pin selects the PLL_ENA input buffer power. Therefore, if VCCSEL is high, the PLL_ENA pin's 1.8/1.5-V input buffer is powered by V_{CCIO} of the bank that PLL_ENA resides in. If VCCSEL is low (GND), the PLL_ENA pin's 3.3/2.5-V input buffer is powered by V_{CCPD}.

For more information on the VCCSEL pin, refer to the *Configuring Stratix II and Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* (or the *Stratix II Device Handbook)*.

pfdena

The pfdena signals control the phase frequency detector (PFD) output with a programmable gate. If you disable the PFD, the VCO operates at its last set value of control voltage and frequency with some long-term drift to a lower frequency. The system continues running when the PLL goes out of lock or the input clock is disabled. By maintaining the last locked frequency, the system has time to store its current settings before shutting down. You can either use your own control signal or clkloss or gated locked status signals, to trigger pfdena.

areset

The areset signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals. When driven high, the PLL counters reset, clearing the PLL output and placing the PLL out of lock. The VCO is set back to its nominal setting (~700 MHz). When driven low again, the PLL will resynchronize to its input as it relocks. If the target VCO frequency is below this nominal frequency, then the output frequency starts at a higher value than desired as the PLL locks.

The areset signal should be asserted every time the PLL loses lock to guarantee correct phase relationship between the PLL input clock and output clocks. Users should include the areset signal in designs if any of the following conditions are true:

- PLL reconfiguration or clock switchover enabled in the design.
- Phase relationships between the PLL input clock and output clocks need to be maintained after a loss of lock condition.
- If the input clock to the PLL is not toggling or is unstable upon power up, assert the areset signal after the input clock is toggling, making sure to stay within the input jitter specification.

Altera recommends that you use the areset and locked signals in your designs to control and observe the status of your PLL.

clkena

If the system cannot tolerate the higher output frequencies when using pfdena higher value, the clkena signals can disable the output clocks until the PLL locks. The clkena signals control the regional, global, and external clock outputs. The clkena signals are registered on the falling edge of the counter output clock to enable or disable the clock without glitches. See Figure 7–56 in the "Clock Control Block" section on page 7–86 of this document for more information on the clkena signals.

Advanced Features

Stratix II and Stratix II GX PLLs offer a variety of advanced features, such as counter cascading, clock switchover, PLL reconfiguration, reconfigurable bandwidth, and spread-spectrum clocking. Table 7–14 shows which advanced features are available in which type of Stratix II or Stratix II GX PLL.

Table 7–14. Stratix II and Stratix II GX PLL Advanced Features

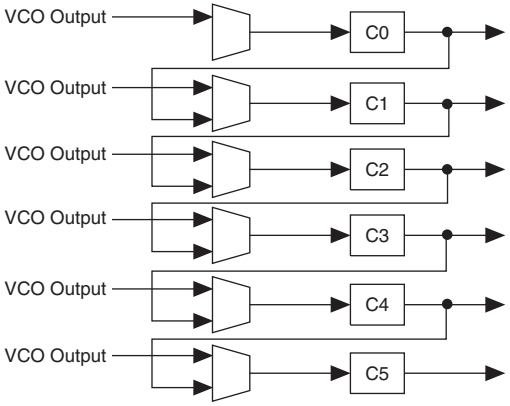
Advanced Feature	Availability	
	Enhanced PLLs	Fast PLLs *(1)*
Counter cascading	✓	
Clock switchover	✓	✓
PLL reconfiguration	✓	✓
Reconfigurable bandwidth	✓	✓
Spread-spectrum clocking	✓	

Note to Table 7–14:
(1) Stratix II and Stratix II GX fast PLLs only support manual clock switchover, not automatic clock switchover.

Counter Cascading

The Stratix II and Stratix II GX enhanced PLL supports counter cascading to create post-scale counters larger than 512. This is implemented by feeding the output of one counter into the input of the next counter in a cascade chain, as shown in Figure 7–16.

Figure 7–16. Counter Cascading



When cascading counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings. For example, if C0 = 4 and C1 = 2, then the cascaded value is C0 × C1 = 8.

The Stratix II and Stratix II GX fast PLLs does not support counter cascading.

Counter cascading is set in the configuration file, meaning they can not be cascaded using PLL reconfiguration.

Clock Switchover

The clock switchover feature allows the PLL to switch between two reference input clocks. Use this feature for clock redundancy or for a dual clock domain application such as in a system that turns on the redundant clock if the primary clock stops running. The design can perform clock switchover automatically, when the clock is no longer toggling, or based on a user control signal, `clkswitch`.

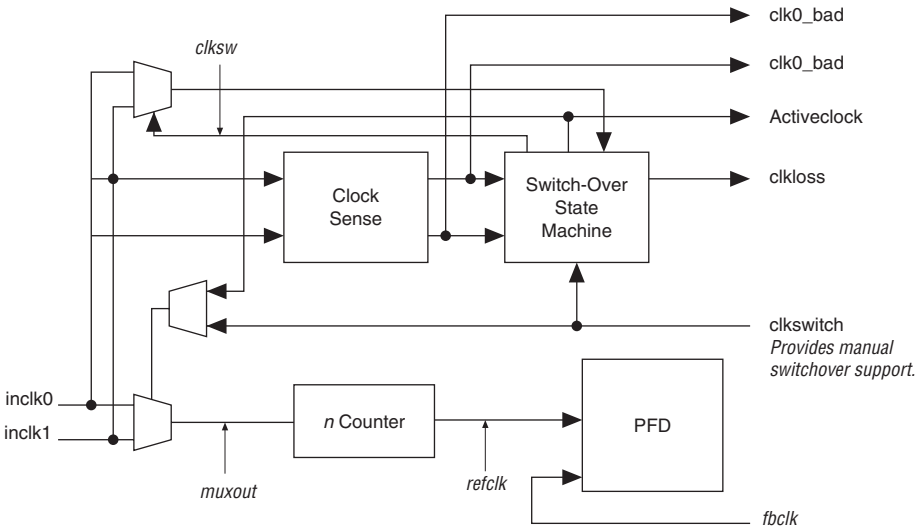
Enhanced PLLs support both automatic and manual switchover, while fast PLLs only support manual switchover.

Automatic Clock Switchover

Stratix II and Stratix II GX device PLLs support a fully configurable clock switchover capability. Figure 7–17 shows the block diagram of the switch-over circuit built into the enhanced PLL. When the primary clock signal is not present, the clock sense block automatically switches from

the primary to the secondary clock for PLL reference. The design sends out the clk0_bad, clk1_bad, and the clk_loss signals from the PLL to implement a custom switchover circuit.

Figure 7–17. Automatic Clock Switchover Circuit Block Diagram



There are two possible ways to use the clock switchover feature.

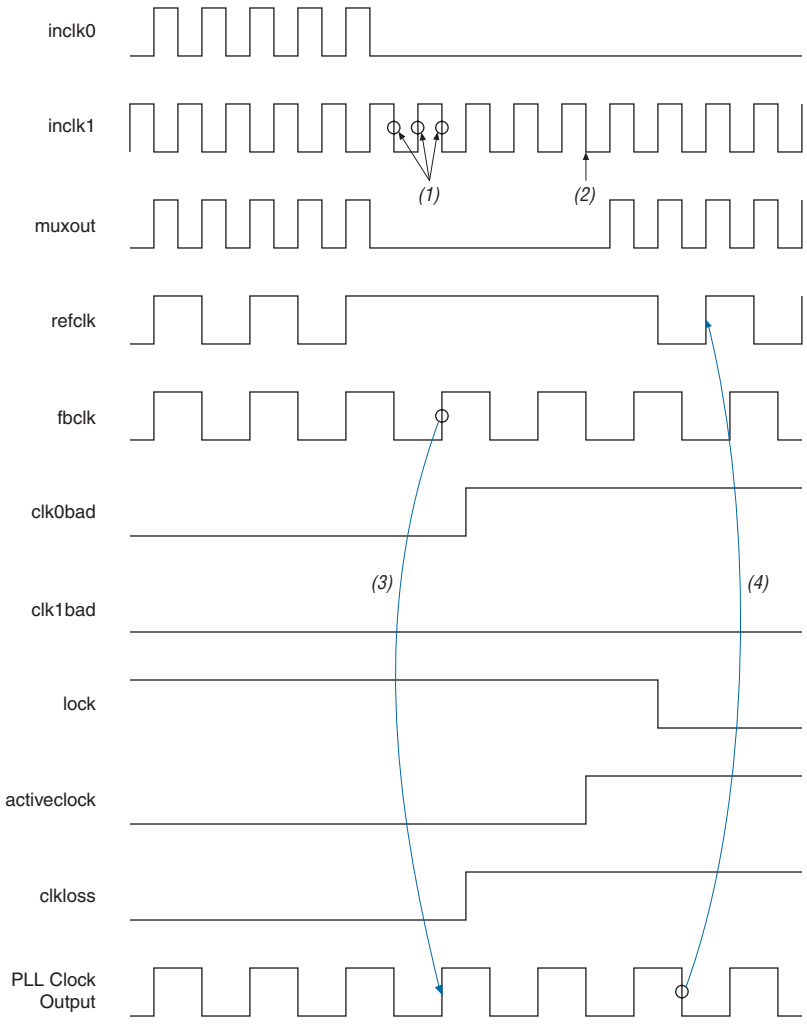
- Use the switchover circuitry for switching from a primary to secondary input of the same frequency. For example, in applications that require a redundant clock with the same frequency as the primary clock, the switchover state machine generates a signal that controls the multiplexer select input shown on the bottom of Figure 7–17. In this case, the secondary clock becomes the reference clock for the PLL. This automatic switchover feature only works for switching from the primary to secondary clock.

- Use the CLKSWITCH input for user- or system-controlled switch conditions. This is possible for same-frequency switchover or to switch between inputs of different frequencies. For example, if inclk0 is 66 MHz and inclk1 is 100 MHz, you must control the switchover because the automatic clock-sense circuitry cannot monitor primary and secondary clock frequencies with a frequency difference of more than 20%. This feature is useful when clock sources can originate from multiple cards on the backplane, requiring a system-controlled switchover between frequencies of operation. You should choose the secondary clock frequency so the

VCO operates within the recommended range of 500 to 1,000 MHz. You should also set the *m* and *n* counters accordingly to keep the VCO operating frequency in the recommended range.

Figure 7–18 shows an example waveform of the switchover feature when using the automatic `clkloss` detection. Here, the `inclk0` signal gets stuck low. After the `inclk0` signal is stuck at low for approximately two clock cycles, the clock sense circuitry drives the `clk0_bad` signal high. Also, because the reference clock signal is not toggling, the `clk_loss` signal goes low, indicating a switch condition. Then, the switchover state machine controls the multiplexer through the `clksw` signal to switch to the secondary clock.

Figure 7–18. Automatic Switchover Upon Clock Loss Detection

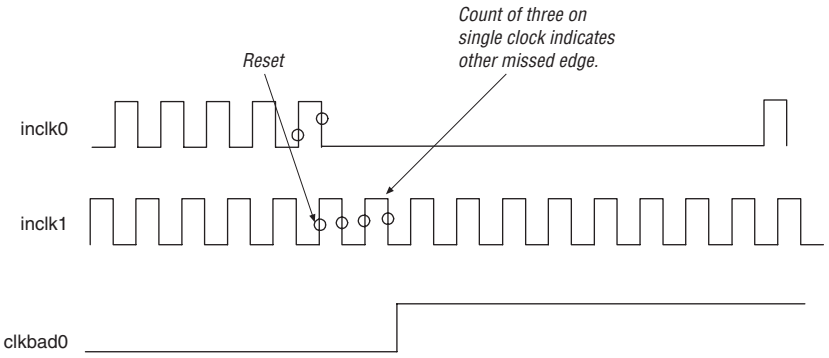


Notes to Figure 7–18:

(1) The number of clock edges before allowing switchover is determined by the counter setting.
(2) Switchover is enabled on the falling edge of INCLK1.
(3) The rising edge of FBCLK causes the VCO frequency to decrease.
(4) The rising edge of REFCLK starts the PLL lock process again, and the VCO frequency increases.

The switch-over state machine has two counters that count the edges of the primary and the secondary clocks; counter0 counts the number of inclk0 edges and counter1 counts the number of inclk1 edges. The counters get reset to zero when the count values reach 1, 1; 1, 2; 2, 1; or 2, 2 for inclock0 and inclock1, respectively. For example, if counter0 counts two edges, its count is set to two and if counter1 counts two edges before the counter0 sees another edge, they are both reset to 0. If for some reason one of the counters counts to three, it means the other clock missed an edge. The clkbad0 or clkbad1 signal goes high, and the switchover circuitry signals a switch condition. See Figure 7–19.

Figure 7–19. Clock-Edge Detection for Switchover



Manual Override

When using automatic switchover, you can switch input clocks by using the manual override feature with the clkswitch input.

The manual override feature available in automatic clock switchover is different from manual clock switchover.

Figure 7–20 shows an example of a waveform illustrating the switchover feature when controlled by clkswitch. In this case, both clock sources are functional and inclk0 is selected as the primary clock. clkswitch goes high, which starts the switchover sequence. On the falling edge of inclk0, the counter's reference clock, muxout, is gated off to prevent any clock glitching. On the falling edge of inclk1, the reference clock multiplexer switches from inclk0 to inclk1 as the PLL reference. This is also when the clksw signal changes to indicate which clock is selected as primary and which is secondary.

The clkloss signal mirrors the clkswitch signal and activeclock mirrors clksw in this mode. Since both clocks are still functional during the manual switch, neither clk_bad signal goes high. Since the

switchover circuit is edge-sensitive, the falling edge of the clkswitch signal does not cause the circuit to switch back from inclk1 to inclk0. When the clkswitch signal goes high again, the process repeats. clkswitch and automatic switch only work if the clock being switched to is available. If the clock is not available, the state machine waits until the clock is available.

Figure 7–20. Clock Switchover Using the CLKSWITCH Control

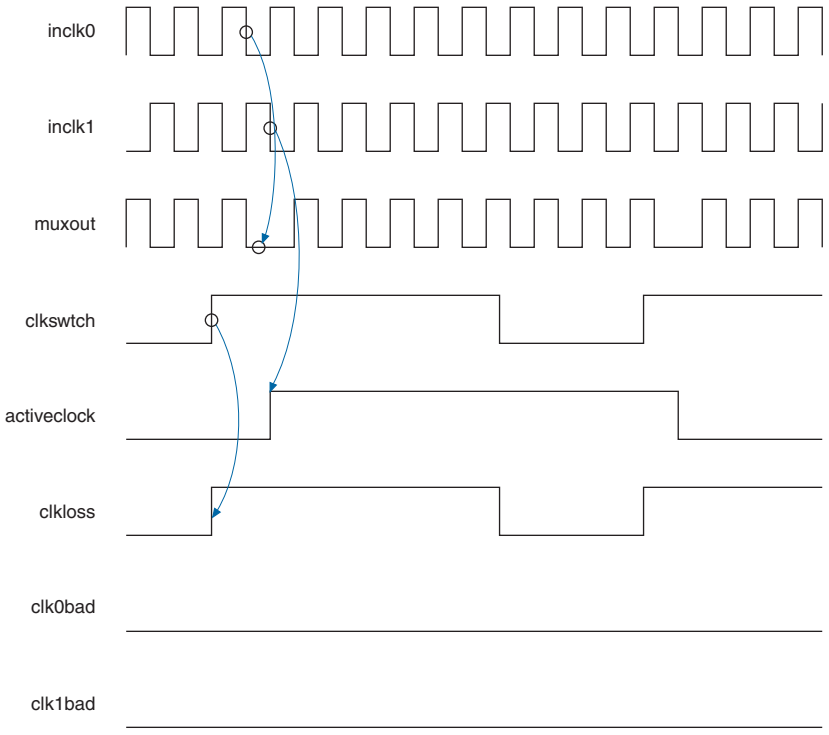
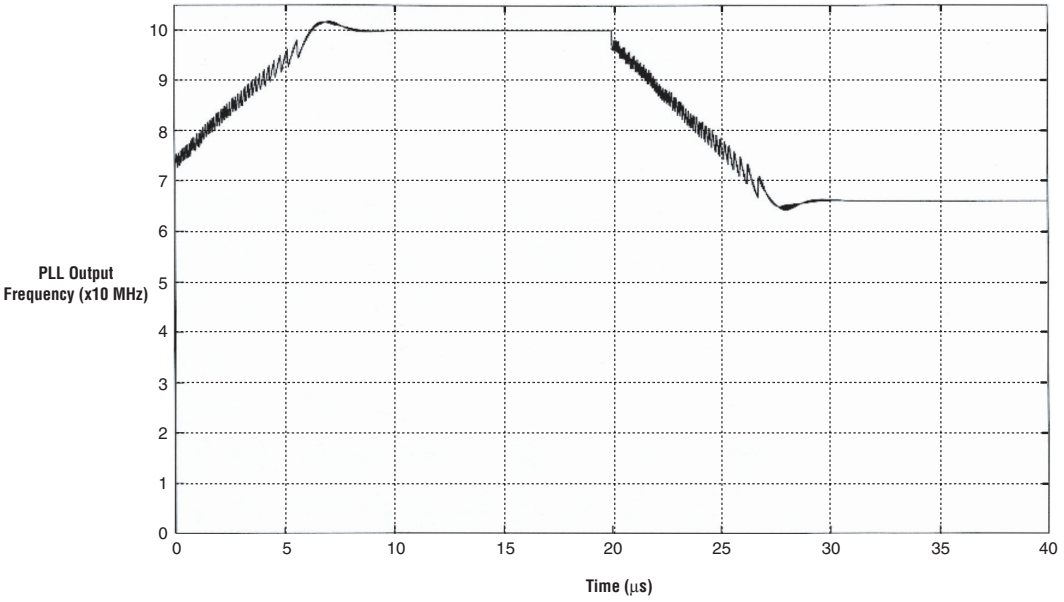


Figure 7–21 shows a simulation of using switchover for two different reference frequencies. In this example simulation, the reference clock is either 100 or 66 MHz. The PLL begins with f_{IN} = 100 MHz and is allowed to lock. At 20 µs, the clock is switched to the secondary clock, which is at 66 MHz.

Figure 7–21. Switchover Simulation *Note (1)*



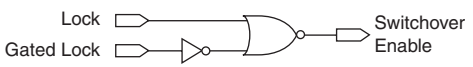
Note to Figure 7–21:

(1) This simulation was performed under the following conditions: the *n* counter is set to 2, the *m* counter is set to 16, and the output counter is set to 8. Therefore, the VCO operates at 800 MHz for the 100-MHz input references and at 528 MHz for the 66-MHz reference input.

Lock Signal-Based Switchover

The lock circuitry can initiate the automatic switchover. This is useful for cases where the input clock is still clocking, but its characteristics have changed so that the PLL is not locked to it. The switchover enable is based on both the gated and ungated lock signals. If the ungated lock is low, the switchover is not enabled until the gated lock has reached its terminal count. You must activate the switchover enable if the gated lock is high, but the ungated lock goes low. The switchover timing for this mode is similar to the waveform shown in Figure 7–20 for `clkswitch` control, except the switchover enable replaces `clkswitch`. Figure 7–17 shows the switchover enable circuit when controlled by lock and gated lock.

Figure 7–22. Switchover Enable Circuit



Manual Clock Switchover

Stratix II and Stratix II GX enhanced and fast PLLs support manual switchover, where the clkswitch signal controls whether inclk0 or inclk1 is the input clock to the PLL. If clkswitch is low, then inclk0 is selected; if clkswitch is high, then inclk1 is selected. Figure 7–23 shows the block diagram of the manual switchover circuit in fast PLLs. The block diagram of the manual switchover circuit in enhanced PLLs is shown in Figure 7–23.

Figure 7–23. Manual Clock Switchover Circuitry in Fast PLLs

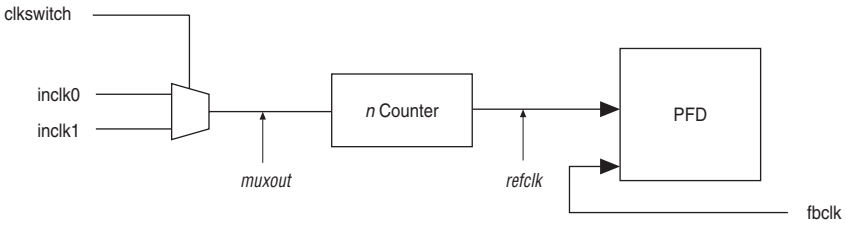
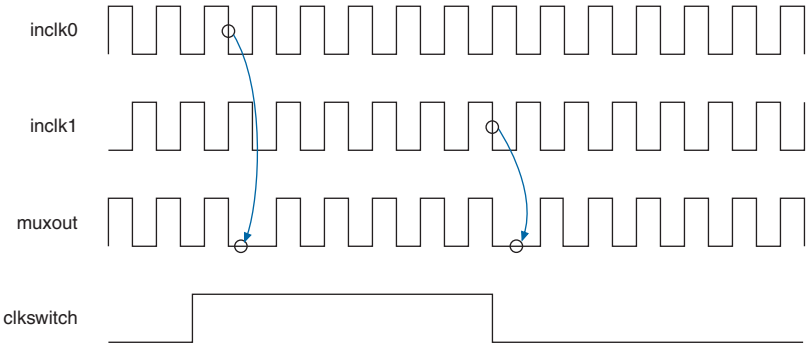


Figure 7–24 shows an example of a waveform illustrating the switchover feature when controlled by clkswitch. In this case, both clock sources are functional and inclk0 is selected as the primary clock. clkswitch goes high, which starts the switch-over sequence. On the falling edge of inclk0, the counter's reference clock, muxout, is gated off to prevent any clock glitching. On the rising edge of inclk1, the reference clock multiplex switches from inclk0 to inclk1 as the PLL reference. When the clkswitch signal goes low, the process repeats, causing the circuit to switch back from inclk1 to inclk0.

Figure 7–24. Manual Switchover



Software Support

Table 7–15 summarizes the signals used for clock switchover.

Table 7–15. altpll Megafunction Clock Switchover Signals

Port	Description	Source	Destination
inclk0	Reference `clk0` to the PLL.	I/O pin	Clock switchover circuit
inclk1	Reference `clk1` to the PLL.	I/O pin	Clock switchover circuit
clkbad0 *(1)*	Signal indicating that `inclk0` is no longer toggling.	Clock switchover circuit	Logic array
clkbad1 *(1)*	Signal indicating that `inclk1` is no longer toggling.	Clock switchover circuit	Logic array
clkswitch	Switchover signal used to initiate clock switchover asynchronously. When used in manual switchover, clkswitch is used as a select signal between inclk0 and inclk1 `clswitch = 0 inclk0` is selected and vice versa.	Logic array or I/O pin	Clock switchover circuit
clkloss *(1)*	Signal indicating that the switchover circuit detected a switch condition.	Clock switchover circuit	Logic array
locked	Signal indicating that the PLL has lost lock.	PLL	Clock switchover circuit
activeclock *(1)*	Signal to indicate which clock (0 = `inclk0`, 1= `inclk1`) is driving the PLL.	PLL	Logic array

Note for Table 7–15:
(1) These ports are only available for enhanced PLLs and in auto mode and when using automatic switchover.

All the switchover ports shown in Table 7–15 are supported in the `altpll` megafunction in the Quartus II software. The `altpll` megafunction supports two methods for clock switchover:

- When selecting an enhanced PLL, you can enable both the automatic and the manual switchover, making all the clock switchover ports available.
- When selecting a fast PLL, you can use only enable the manual clock switchover option to select between `inclk0` or `inclk1`. The `clkloss`, `activeclock` and the `clkbad0`, and `clkbad1` signals are not available when manual switchover is selected.

If the primary and secondary clock frequencies are different, the Quartus II software selects the proper parameters to keep the VCO within the recommended frequency range.

For more information on PLL software support in the Quartus II software, see the *altpll Megafunction User Guide*.

Guidelines

Use the following guidelines to design with clock switchover in PLLs.

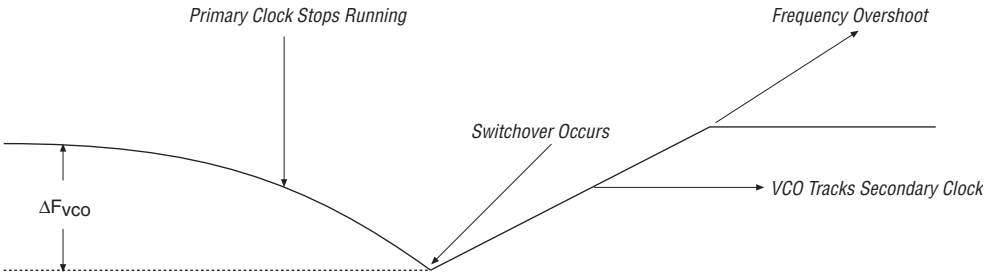
- When using automatic switchover, the `clkswitch` signal has a minimum pulse width based on the two reference clock periods. The `CLKSWITCH` pulse width must be greater than or equal to the period of the current reference clock (t_{from_clk}) multiplied by two plus the rounded-up version of the ratio of the two reference clock periods. For example, if t_{to_clk} is equal to t_{from_clk}, then the `CLKSWITCH` pulse width should be at least three times the period of the clock pulse.

  $$t_{CLKSWITCHCHmin} \geq t_{from\_clk} \times [2 + int_{round\_up} (t_{to\_clk} \div t_{from\_clk})]$$

- Applications that require a clock switchover feature and a small frequency drift should use a low-bandwidth PLL. The low-bandwidth PLL reacts slower than a high-bandwidth PLL to reference input clock changes. When the switchover happens, a low-bandwidth PLL propagates the stopping of the clock to the output slower than a high-bandwidth PLL. A low-bandwidth PLL filters out jitter on the reference clock. However, be aware that the low-bandwidth PLL also increases lock time.
- Stratix II and Stratix II GX device PLLs can use both the automatic clock switchover and the `clkswitch` input simultaneously. Therefore, the switchover circuitry can automatically switch from the primary to the secondary clock. Once the primary clock stabilizes again, the `clkswitch` signal can switch back to the primary clock. During switchover, the `PLL_VCO` continues to run and slows down, generating frequency drift on the PLL outputs. The `clkswitch` signal controls switchover with its rising edge only.
- If the clock switchover event is glitch-free, after the switch occurs, there is still a finite resynchronization period to lock onto a new clock as the VCO ramps up. The exact amount of time it takes for the PLL to relock is dependent on the PLL configuration. Use the PLL programmable bandwidth feature to adjust the relock time.
- If the phase relationship between the input clock to the PLL and output clock from the PLL is important in your design, assert `areset` for 10ns after performing a clock switchover. Wait for the locked signal (or gated lock) to go high before re-enabling the output clocks from the PLL.

- Figure 7–25 shows how the VCO frequency gradually decreases when the primary clock is lost and then increases as the VCO locks on to the secondary clock. After the VCO locks on to the secondary clock, some overshoot can occur (an over-frequency condition) in the VCO frequency.

Figure 7–25. VCO Switchover Operating Frequency



- Disable the system during switchover if it is not tolerant to frequency variations during the PLL resynchronization period. There are two ways to disable the system. First, the system may require some time to stop before switchover occurs. The switchover circuitry includes an optional five-bit counter to delay when the reference clock is switched. You have the option to control the time-out setting on this counter (up to 32 cycles of latency) before the clock source switches. You can use these cycles for disaster recovery. The clock output frequency varies slightly during those 32 cycles since the VCO can still drift without an input clock. Programmable bandwidth can control the PLL response to limit drift during this 32 cycle period.
- A second option available is the ability to use the PFD enable signal (`pfdena`) along with user-defined control logic. In this case you can use `clk0_bad` and `clk1_bad` status signals to turn off the PFD so the VCO maintains its last frequency. You can also use the state machine to switch over to the secondary clock. Upon re-enabling the PFD, output clock enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. Once the lock indication is stable, the system can re-enable the output clock(s).

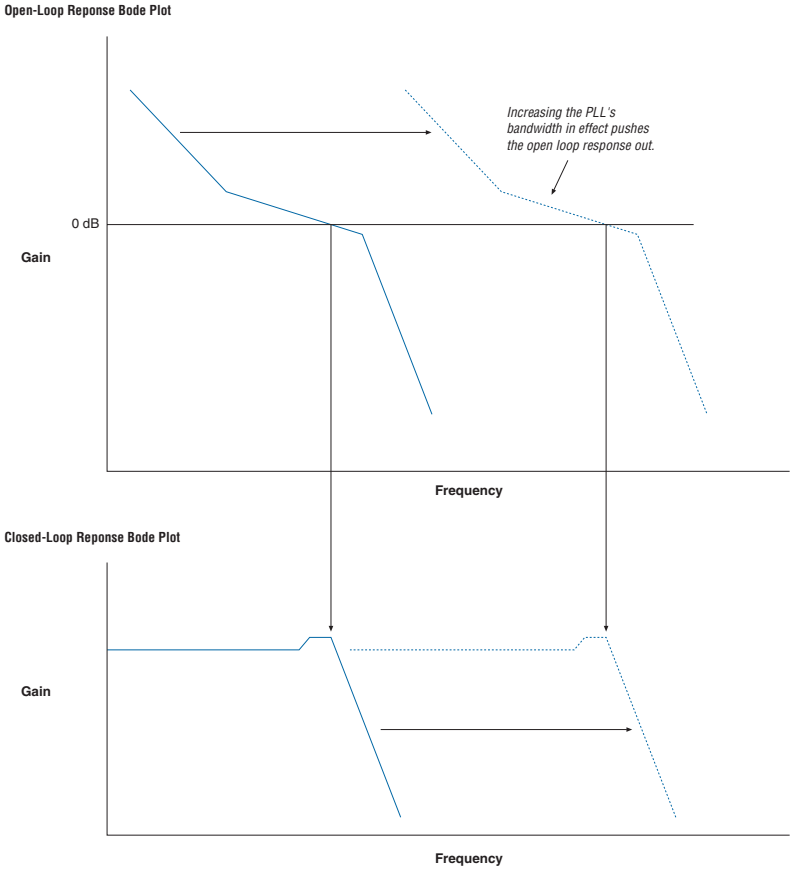
Reconfigurable Bandwidth

Stratix II and Stratix II GX enhanced and fast PLLs provide advanced control of the PLL bandwidth using the PLL loop's programmable characteristics, including loop filter and charge pump.

Background

PLL bandwidth is the measure of the PLL's ability to track the input clock and jitter. The closed-loop gain 3-dB frequency in the PLL determines the PLL bandwidth. The bandwidth is approximately the unity gain point for open loop PLL response. As Figure 7–26 shows, these points correspond to approximately the same frequency.

Figure 7–26. Open- and Closed-Loop Response Bode Plots



A high-bandwidth PLL provides a fast lock time and tracks jitter on the reference clock source, passing it through to the PLL output. A low-bandwidth PLL filters out reference clock, but increases lock time. Stratix II and Stratix II GX enhanced and fast PLLs allow you to control the bandwidth over a finite range to customize the PLL characteristics for a particular application. The programmable bandwidth feature in Stratix II and Stratix II GX PLLs benefits applications requiring clock switchover (e.g., TDMA frequency hopping wireless, and redundant clocking).

The bandwidth and stability of such a system is determined by the charge pump current, the loop filter resistor value, the high-frequency capacitor value (in the loop filter), and the *m*-counter value. You can use the Quartus II software to control these factors and to set the bandwidth to the desired value within a given range.

You can set the bandwidth to the appropriate value to balance the need for jitter filtering and lock time. Figures 7–27 and 7–28 show the output of a low- and high-bandwidth PLL, respectively, as it locks onto the input clock.

Figure 7–27. Low-Bandwidth PLL Lock Time

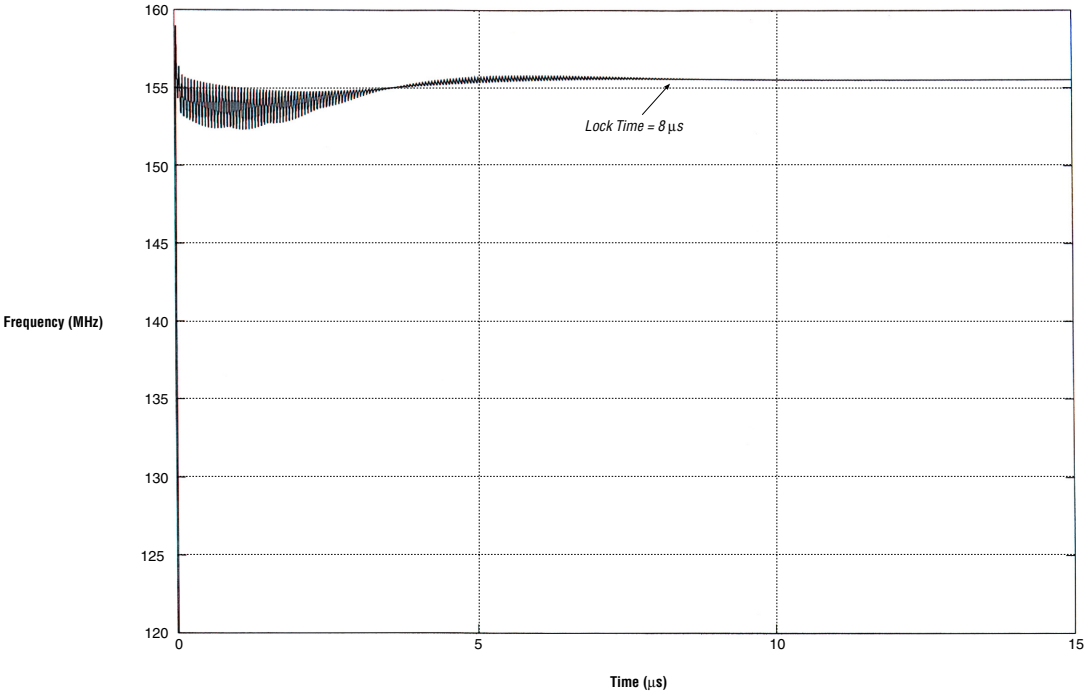
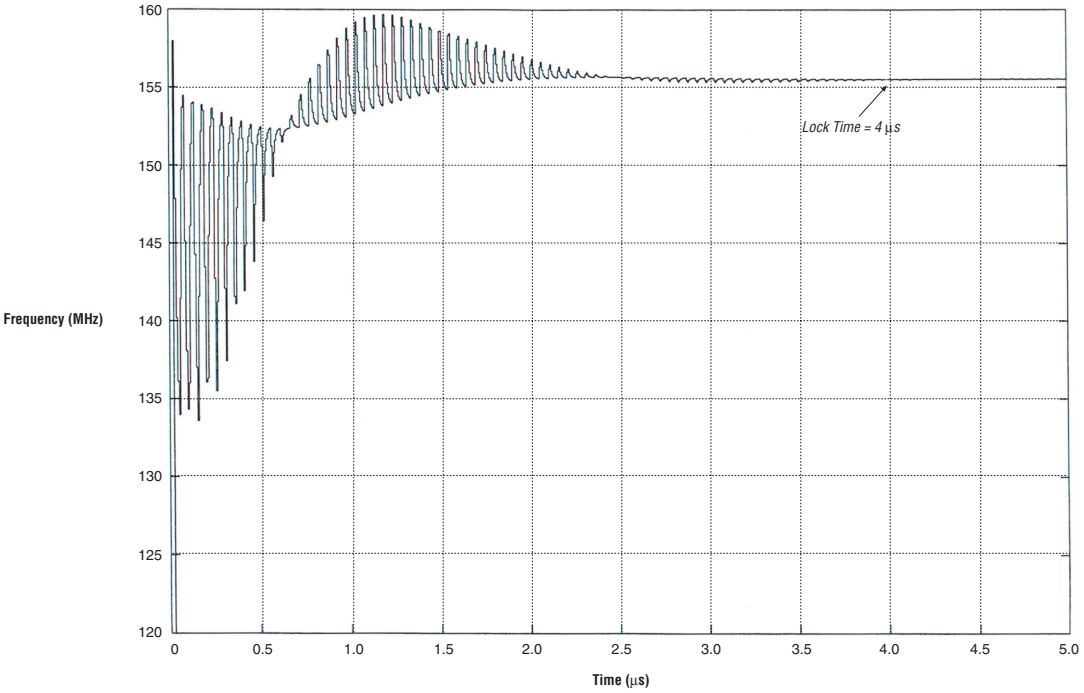


Figure 7–28. High-Bandwidth PLL Lock Time



A high-bandwidth PLL can benefit a system that has two cascaded PLLs. If the first PLL uses spread spectrum (as user-induced jitter), the second PLL can track the jitter that is feeding it by using a high-bandwidth setting. A low-bandwidth PLL can, in this case, lose lock due to the spread-spectrum-induced jitter on the input clock.

A low-bandwidth PLL benefits a system using clock switchover. When the clock switchover happens, the PLL input temporarily stops. A low-bandwidth PLL would react more slowly to changes to its input clock and take longer to drift to a lower frequency (caused by the input stopping) than a high-bandwidth PLL. Figures 7–29 and 7–30 demonstrate this property. The two plots show the effects of clock switchover with a low- or high-bandwidth PLL. When the clock switchover happens, the output of the low-bandwidth PLL (see Figure 7–29) drifts to a lower frequency more slowly than the high-bandwidth PLL output (see Figure 7–30).

Figure 7–29. Effect of Low Bandwidth on Clock Switchover

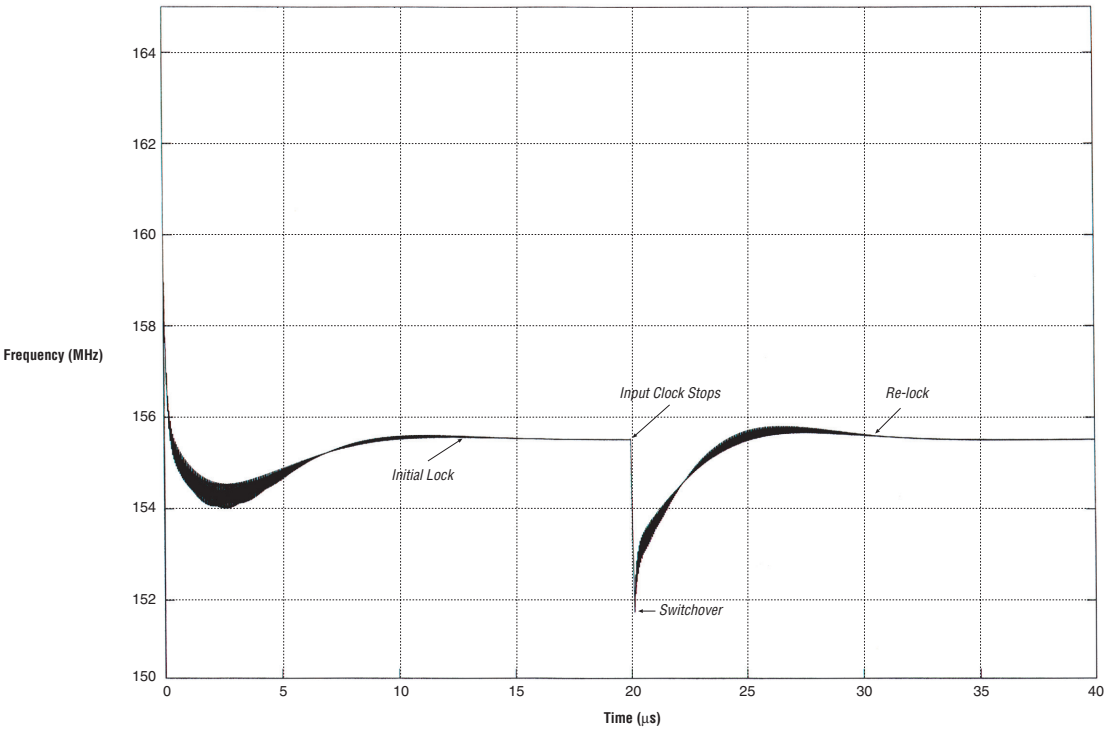
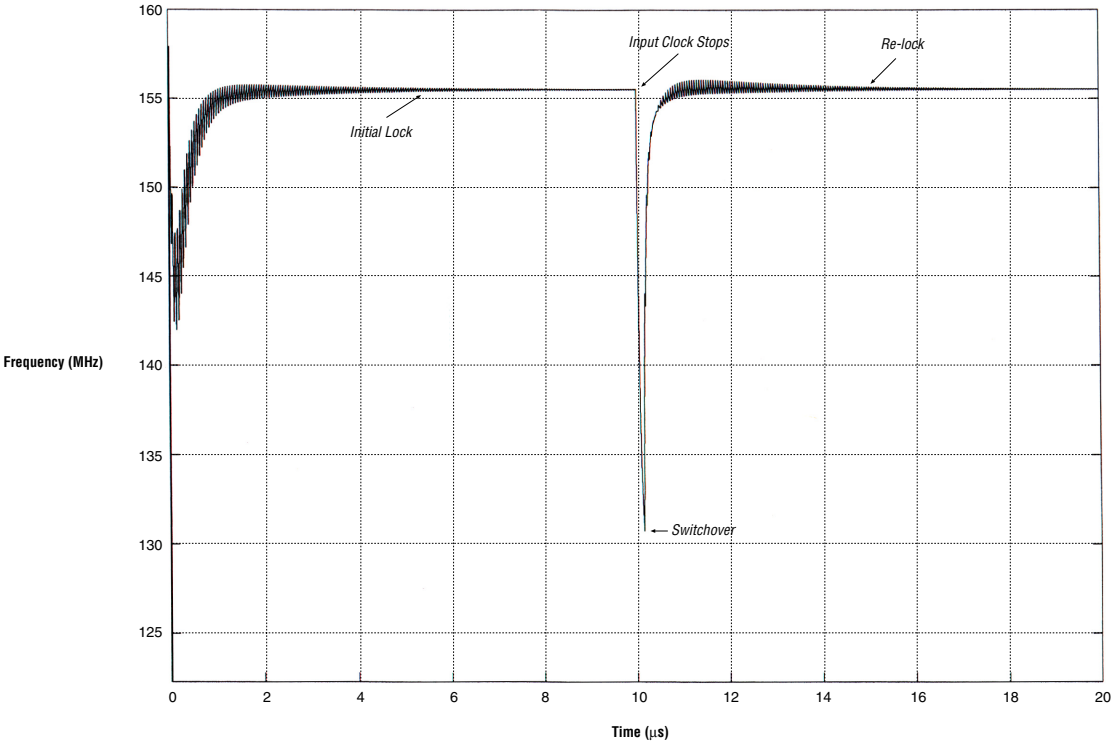


Figure 7–30. Effect of High Bandwidth on Clock Switchover



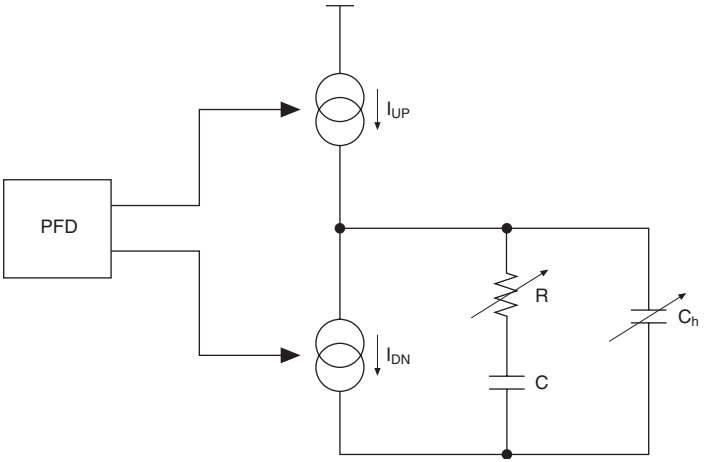
Implementation

Traditionally, external components such as the VCO or loop filter control a PLL's bandwidth. Most loop filters are made up of passive components such as resistors and capacitors that take up unnecessary board space and increase cost. With Stratix II and Stratix II GX PLLs, all the components are contained within the device to increase performance and decrease cost.

Stratix II and Stratix II GX device PLLs implement reconfigurable bandwidth by giving you control of the charge pump current and loop filter resistor (R) and high-frequency capacitor C_H values (see Table 7–16). The Stratix II and Stratix II GX device enhanced PLL bandwidth ranges from 130 kHz to 16.9 MHz. The Stratix II and Stratix II GX device fast PLL bandwidth ranges from 1.16 to 28 MHz.

The charge pump current directly affects the PLL bandwidth. The higher the charge pump current, the higher the PLL bandwidth. You can choose from a fixed set of values for the charge pump current. Figure 7–31 shows the loop filter and the components that can be set through the Quartus II software. The components are the loop filter resistor, R, and the high frequency capacitor, C_H, and the charge pump current, I_{UP} or I_{DN}.

Figure 7–31. Loop Filter Programmable Components



Software Support

The Quartus II software provides two levels of bandwidth control.

Megafunction-Based Bandwidth Setting

The first level of programmable bandwidth allows you to enter a value for the desired bandwidth directly into the Quartus II software using the `altpll` megafunction. You can also set the bandwidth parameter in the `altpll` megafunction to the desired bandwidth. The Quartus II software selects the best bandwidth parameters available to match your bandwidth request. If the individual bandwidth setting request is not available, the Quartus II software selects the closest achievable value.

Advanced Bandwidth Setting

An advanced level of control is also possible using advanced loop filter parameters. You can dynamically change the charge pump current, loop filter resistor value, and the loop filter (high frequency) capacitor value.

The parameters for these changes are: `charge_pump_current`, `loop_filter_r`, and `loop_filter_c`. Each parameter supports the specific range of values listed in Table 7–16.

Table 7–16. Advanced Loop Filter Parameters

Parameter	Values
Resistor values (kΩ)	*(1)*
High-frequency capacitance values (pF)	*(1)*
Charge pump current settings (µA)	*(1)*

Note to Table 7–16:
(1) For more information, see *AN 367: Implementing PLL Reconfiguration in Stratix II Devices*.

For more information on Quartus II software support of reconfigurable bandwidth, see the *Design Example: Dynamic PLL Reconfiguration* section in volume 3, Verification, of the *Quartus II Development Software Handbook*.

PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In Stratix II and Stratix II GX enhanced and fast PLLs, the counter value and phase are configurable in real time. In addition, you can change the loop filter and charge pump components, which affect the PLL bandwidth, on the fly. You can control these PLL components to update the output clock frequency, PLL bandwidth, and phase-shift variation in real time, without the need to reconfigure the entire FPGA.

For more information on PLL reconfiguration, see *AN 367: Implementing PLL Reconfiguration in Stratix II Devices*.

Spread-Spectrum Clocking

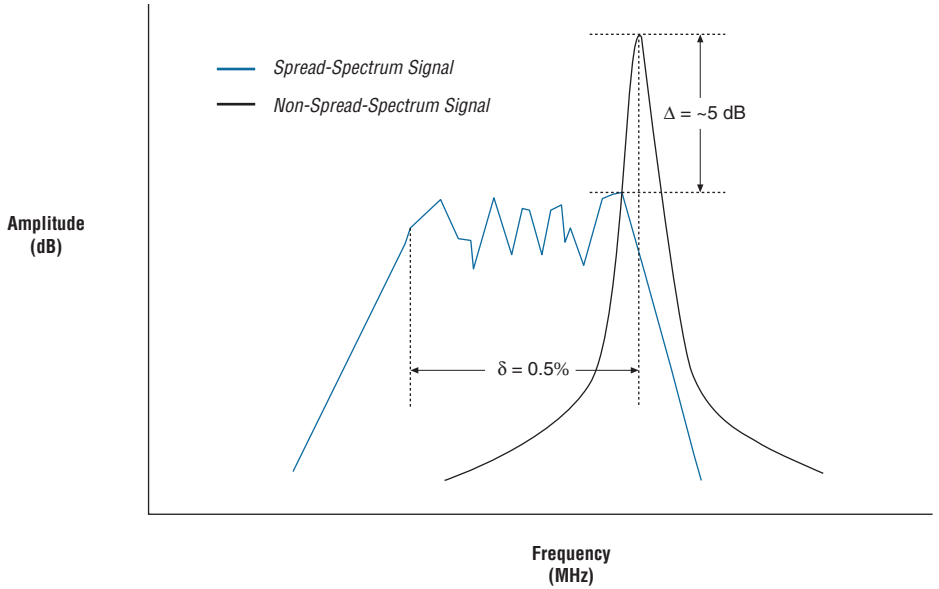
Digital clocks are square waves with short rise times and a 50% duty cycle. These high-speed clocks concentrate a significant amount of energy in a narrow bandwidth at the target frequency and at the higher frequency harmonics. This results in high energy peaks and increased electromagnetic interference (EMI). The radiated noise from the energy peaks travels in free air and, if not minimized, can lead to corrupted data and intermittent system errors, which can jeopardize system reliability.

Traditional methods for limiting EMI include shielding, filtering, and multi-layer printed circuit boards (PCBs). However, these methods significantly increase the overall system cost and sometimes are not

enough to meet EMI compliance. Spread-spectrum technology provides you with a simple and effective technique for reducing EMI without additional cost and the trouble of re-designing a board.

Spread-spectrum technology modulates the target frequency over a small range. For example, if a 100-MHz signal has a 0.5% down-spread modulation, then the frequency is swept from 99.5 to 100 MHz. Figure 7–32 gives a graphical representation of the energy present in a spread-spectrum signal vs. a non-spread spectrum-signal. It is apparent that instead of concentrating the energy at the target frequency, the energy is re-distributed across a wider band of frequencies, which reduces peak energy. Not only is there a reduction in the fundamental peak EMI components, but there is also a reduction in EMI of the higher order harmonics. Since some regulations focus on peak EMI emissions, rather than average EMI emissions, spread-spectrum technology is a valuable method of EMI reduction.

Figure 7–32. Spread-Spectrum Signal Energy Versus Non-Spread-Spectrum Signal Energy



Spread-spectrum technology would benefit a design with high EMI emissions and/or strict EMI requirements. Device-generated EMI is dependent on frequency and output voltage swing amplitude and edge rate. For example, a design using LVDS already has low EMI emissions

because of the low-voltage swing. The differential LVDS signal also allows for EMI rejection within the signal. Therefore, this situation may not require spread-spectrum technology.

Spread-spectrum clocking is only supported in Stratix II enhanced PLLs, not fast PLLs.

Implementation

Stratix II and Stratix II GX device enhanced PLLs feature spread-spectrum technology to reduce the EMIs emitted from the device. The enhanced PLL provides approximately 0.5% down spread using a triangular, also known as linear, modulation profile. The modulation frequency is programmable and ranges from approximately 100 to 500 kHz. The spread percentage is based on the clock input to the PLL and the *m* and *n* settings. Spread-spectrum technology reduces the peak energy by four to six dB at the target frequency. However, this number is dependent on bandwidth and the *m* and *n* counter values and can vary from design to design.

Spread percentage, also known as modulation width, is defined as the percentage that the design modulates the target frequency. A negative (–) percentage indicates a down spread, a positive (+) percentage indicates an up spread, and a (±) indicates a center spread. Modulation frequency is the frequency of the spreading signal, or how fast the signal sweeps from the minimum to the maximum frequency. Down-spread modulation shifts the target frequency down by half the spread percentage, centering the modulated waveforms on a new target frequency.

The *m* and *n* counter values are toggled at the same time between two fixed values. The loop filter then slowly changes the VCO frequency to provide the spreading effect, which results in a triangular modulation. An additional spread-spectrum counter (shown in Figure 7–33) sets the modulation frequency. Figure 7–33 shows how spread-spectrum technology is implemented in the Stratix II and Stratix II GX device enhanced PLL.

Figure 7–33. Stratix II and Stratix II GX Spread-Spectrum Circuit Block Diagram

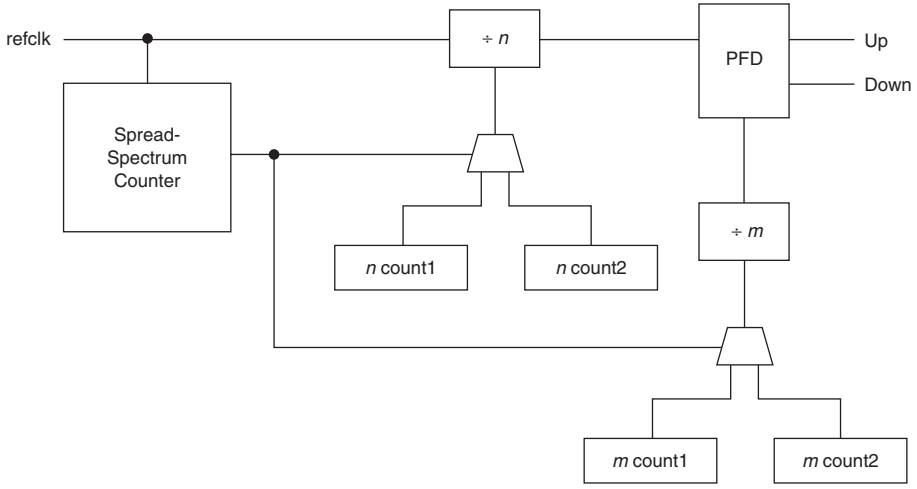


Figure 7–34 shows a VCO frequency waveform when toggling between different counter values. Since the enhanced PLL switches between two different m and n values, the result is a straight line between two frequencies, which gives a linear modulation. The magnitude of modulation is determined by the ratio of two m/n sets. The percent spread is determined by:

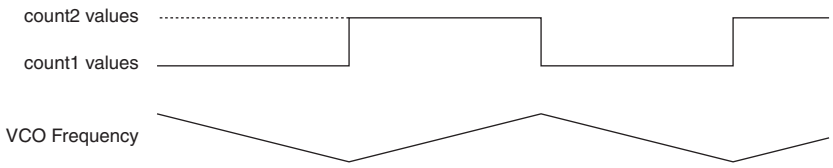
$$\text{percent spread} = (f_{VCOmax} - f_{VCOmin})/f_{VCOmax} = 1\ [(m_2 \times n_1)/(m_1 \times n_2)].$$

The maximum and minimum VCO frequency is defined as:

$$f_{VCOmax} = (m_1/n_1) \times f_{REF}$$

$$f_{VCOmin} = (m_2/n_2) \times f_{REF}$$

Figure 7–34. VCO Frequency Modulation Waveform



Software Support

You can enter the desired down-spread percentage and modulation frequency in the `altpll` megafunction through the Quartus II software. Alternatively, the `downspread` parameter in the `altpll` megafunction can be set to the desired down-spread percentage. Timing analysis ensures the design operates at the maximum spread frequency and meets all timing requirements.

For more information on PLL software support in the Quartus II software, see the *altpll Megafunction User Guide*.

Guidelines

If the design cascades PLLs, the source (upstream) PLL should have a low-bandwidth setting, while the destination (downstream) PLL should have a high-bandwidth setting. The upstream PLL must have a low-bandwidth setting because a PLL does not generate jitter higher than its bandwidth. The downstream PLL must have a high bandwidth setting to track the jitter. The design must use the spread-spectrum feature in a low-bandwidth PLL, and, therefore, the Quartus II software automatically sets the spread-spectrum PLL bandwidth to low.

If the programmable or reconfigurable bandwidth features are used, then you cannot use spread spectrum.

Stratix II and Stratix II GX devices can accept a spread-spectrum input with typical modulation frequencies. However, the device cannot automatically detect that the input is a spread-spectrum signal. Instead, the input signal looks like deterministic jitter at the input of the downstream PLL.

Spread spectrum can have a minor effect on the output clock by increasing the period jitter. Period jitter is the deviation of a clock's cycle time from its previous cycle position. Period jitter measures the variation of the clock output transition from its ideal position over consecutive edges.

With down-spread modulation, the peak of the modulated waveform is the actual target frequency. Therefore, the system never exceeds the maximum clock speed. To maintain reliable communication, the entire system and subsystem should use the Stratix II and Stratix II GX device as the clock source. Communication could fail if the Stratix II or Stratix II GX logic array is clocked by the spread-spectrum clock, but the data it receives from another device is not clocked by the spread spectrum.

Since spread spectrum affects the *m* counter values, all spread-spectrum PLL outputs are effected. Therefore, if only one spread-spectrum signal is needed, the clock signal should use a separate PLL without other outputs from that PLL.

No special considerations are needed when using spread spectrum with the clock switchover feature. This is because the clock switchover feature does not affect the *m* and *n* counter values, which are the counter values switching when using spread spectrum.

Board Layout

The enhanced and fast PLL circuits in Stratix II and Stratix II GX devices contain analog components embedded in a digital device. These analog components have separate power and ground pins to minimize noise generated by the digital components. Stratix II and Stratix II GX enhanced and fast PLLs use separate V_{CC} and ground pins to isolate circuitry and improve noise resistance.

V_{CCA} and GNDA

Each enhanced and fast PLL uses separate V_{CC} and ground pin pairs for their analog circuitry. The analog circuit power and ground pin for each PLL is called `VCCA_PLL`<*PLL number*> and `GNDA_PLL`<*PLL number*>. Connect the V_{CCA} power pin to a 1.2-V power supply, even if you do not use the PLL. Isolate the power connected to V_{CCA} from the power to the rest of the Stratix II or Stratix II GX device or any other digital device on the board. You can use one of three different methods of isolating the V_{CCA} pin: separate V_{CCA} power planes, a partitioned V_{CCA} island within the V_{CCINT} plane, and thick V_{CCA} traces.

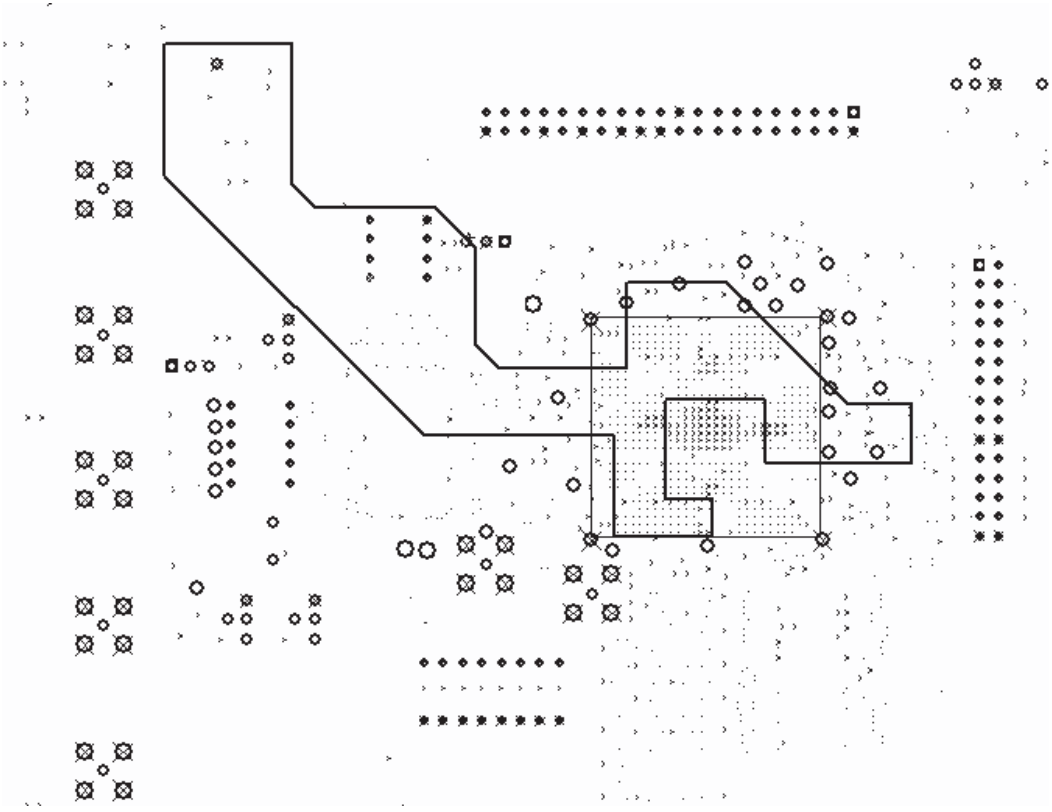
Separate V_{CCA} Power Plane

A mixed signal system is already partitioned into analog and digital sections, each with its own power planes on the board. To isolate the V_{CCA} pin using a separate V_{CCA} power plane, connect the V_{CCA} pin to the analog 1.2-V power plane.

Partitioned V_{CCA} Island Within V_{CCINT} Plane

Fully digital systems do not have a separate analog power plane on the board. Since it is expensive to add new planes to the board, you can create islands for `VCCA_PLL`. Figure 7–35 shows an example board layout with an analog power island. The dielectric boundary that creates the island should be 25 mils thick. Figure 7–36 shows a partitioned plane within V_{CCINT} for V_{CCA}.

Figure 7–35. V_{CCINT} Plane Partitioned for V_{CCA} Island

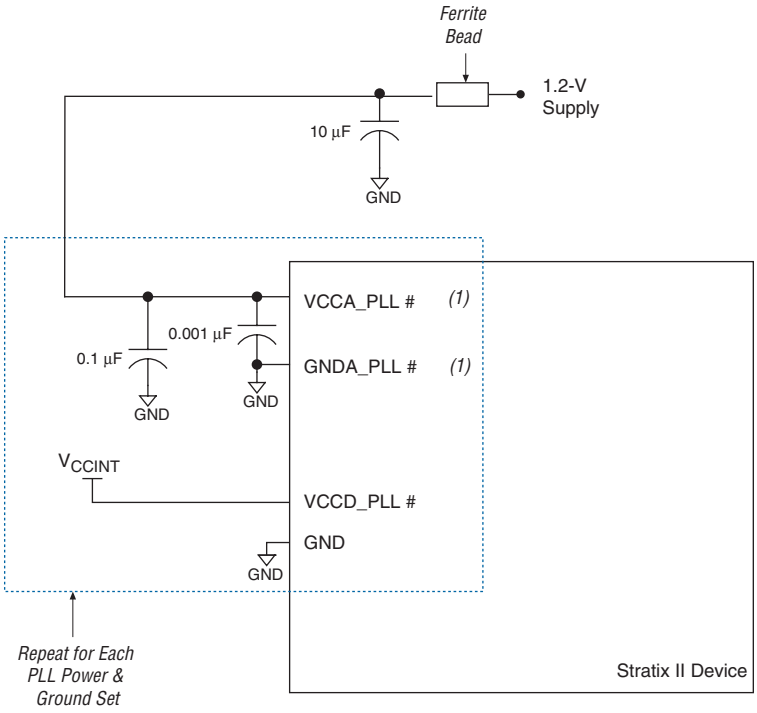


Thick V_{CCA} Trace

Because of board constraints, you may not be able to partition a V_{CCA} island. Instead, run a thick trace from the power supply to each V_{CCA} pin. The traces should be at least 20 mils thick.

In each of these three cases, you should filter each VCCA_PLL pin with a decoupling circuit, as shown in Figure 7–36. Place a ferrite bead that exhibits high impedance at frequencies of 50 MHz or higher and a 10-µF tantalum parallel capacitor where the power enters the board. Decouple each VCCA_PLL pin with a 0.1-µF and 0.001-µF parallel combination of ceramic capacitors located as close as possible to the Stratix II or Stratix II GX device. You can connect the GNDA_PLL pins directly to the same ground plane as the device's digital ground.

Figure 7–36. PLL Power Schematic for Stratix II and Stratix II GX PLLs



Note to Figure 7–36
(1) Applies to PLLs 1 through 12.

V_{CCD}

The digital power and ground pins are labeled VCCD_PLL<*PLL number*> and GND. The VCCD pin supplies the power for the digital circuitry in the PLL. Connect these VCCD pins to the quietest digital supply on the board. In most systems, this is the digital 1.2-V supply supplied to the device's V_{CCINT} pins. Connect the VCCD pins to a power supply even if you do not use the PLL. When connecting the V_{CCD} pins to V_{CCINT}, you do not need any filtering or isolation. You can connect the GND pins directly to the same ground plane as the device's digital ground. See Figure 7–36.

External Clock Output Power

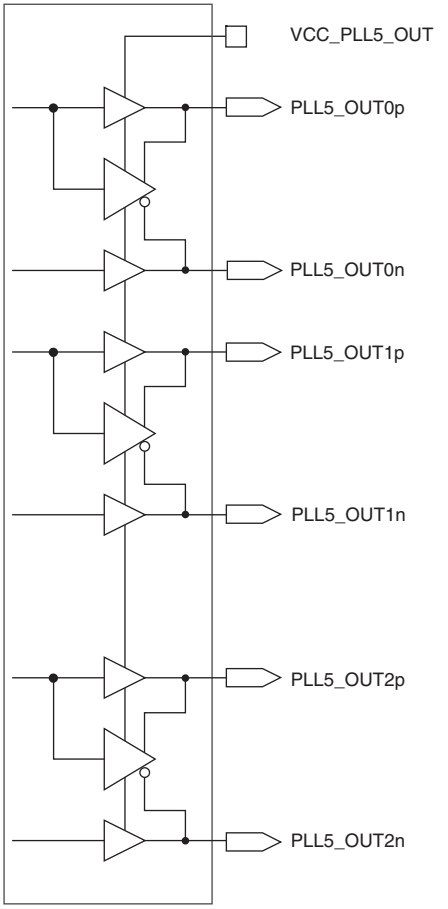
Enhanced PLLs 5, 6, 11, and 12 also have isolated power pins for their dedicated external clock outputs (VCC_PLL5_OUT, VCC_PLL6_OUT, VCC_PLL11_OUT and VCC_PLL12_OUT, respectively). Since the

dedicated external clock outputs from a particular enhanced PLL are powered by separate power pins, they are less susceptible to noise. They also reduce the overall jitter of the output clock by providing improved isolation from switching I/O pins.

I/O pins that reside in PLL banks 9 through 12 are powered by the VCC_PLL<*5, 6, 11, or 12*>_OUT pins, respectively. The EP2S60F484, EP2S60F780, EP2S90H484, EP2S90F780, and EP2S130F780 devices do not support PLLs 11 and 12. Therefore, any I/O pins that reside in bank 11 are powered by the VCCIO3 pin, and any I/O pins that reside in bank 12 are powered by the VCCIO8 pin.

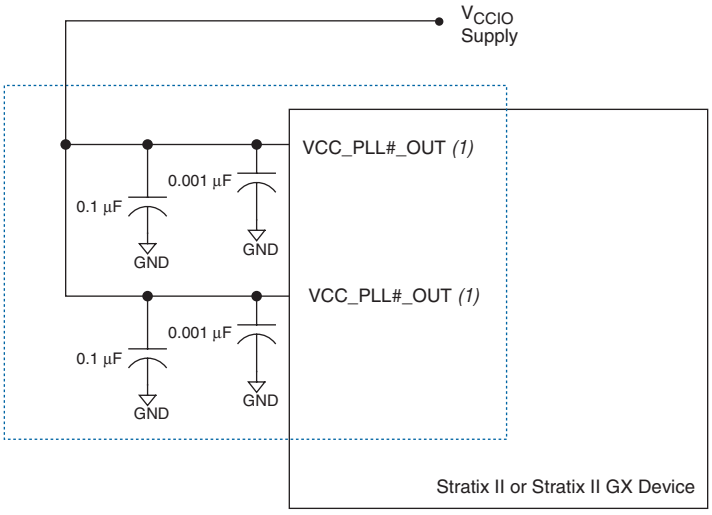
The VCC_PLL_OUT pins can by powered by 3.3, 2.5, 1.8, or 1.5 V, depending on the I/O standard for the clock output from a particular enhanced PLL, as shown in Figure 7–37.

Figure 7–37. External Clock Output Pin Association with Output Power



Filter each isolated power pin with a decoupling circuit shown in Figure 7–38. Decouple the isolated power pins with parallel combination of 0.1- and 0.001-µF ceramic capacitors located as close as possible to the Stratix II or Stratix II GX device.

Figure 7–38. Stratix II and Stratix II GX PLL External Clock Output Power Ball Connection *Note (1)*



Note to Figure 7–38:
(1) Applies only to enhanced PLLs 5, 6, 11, and 12.

Guidelines

Use the following guidelines for optimal jitter performance on the external clock outputs from enhanced PLLs 5, 6, 11, and 12. If all outputs are running at the same frequency, these guidelines are not necessary to improve performance.

- Use phase shift to ensure edges are not coincident on all the clock outputs.
- Use phase shift to skew clock edges with respect to each other for best jitter performance.

If you cannot drive multiple clocks of different frequencies and phase shifts or isolate banks, you should control the drive capability on the lower-frequency clock. Reducing how much current the output buffer has to supply can reduce the noise. Minimize capacitive load on the slower frequency output and configure the output buffer to lower current strength. The higher-frequency output should have an improved performance, but this may degrade the performance of your lower-frequency clock output.

PLL Specifications

See the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook* (or the *Stratix II Device Handbook*) for information on PLL timing specifications

Clocking

Stratix II and Stratix II GX devices provide a hierarchical clock structure and multiple PLLs with advanced features. The large number of clocking resources in combination with the clock synthesis precision provided by enhanced and fast PLLs provides a complete clock-management solution.

Global and Hierarchical Clocking

Stratix II and Stratix II GX devices provide 16 dedicated global clock networks and 32 regional clock networks. These clocks are organized into a hierarchical clock structure that allows for 24 unique clock sources per device quadrant with low skew and delay. This hierarchical clocking scheme provides up to 48 unique clock domains within the entire Stratix II or Stratix II GX device. Table 7–17 lists the clock resources available on Stratix II devices.

There are 16 dedicated clock pins (`CLK[15..0]`) on Stratix II and Stratix II GX devices to drive either the global or regional clock networks. Four clock pins drive each side of the Stratix II device, as shown in Figures 7–39 and 7–40**.** Enhanced and fast PLL outputs can also drive the global and regional clock networks.

Table 7–17. Clock Resource Availability in Stratix II and Stratix II GX Devices (Part 1 of 2)

Description	Stratix II Device Availability	Stratix II GX Device Availability
Number of clock input pins	24	12
Number of global clock networks	16	16
Number of regional clock networks	32	32
Global clock input sources	Clock input pins, PLL outputs, logic array	Clock input pins, PLL outputs, logic array, inter-transceiver clocks
Regional clock input sources	Clock input pins, PLL outputs, logic array	Clock input pins, PLL outputs, logic array, inter-transceiver clocks
Number of unique clock sources in a quadrant	24 (16 global clocks and 8 regional clocks)	24 (16 GCLK and 8 RCLK clocks)
Number of unique clock sources in the entire device	48 (16 global clocks and 32 regional clocks)	48 (16 GCLK and 32 RCLK clocks)

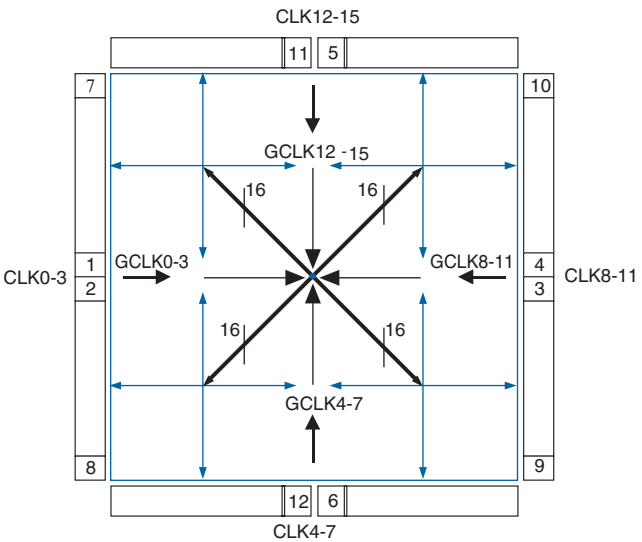
Table 7–17. Clock Resource Availability in Stratix II and Stratix II GX Devices (Part 2 of 2)

Description	Stratix II Device Availability	Stratix II GX Device Availability
Power-down mode	Global clock networks, regional clock networks, dual-regional clock region	GCLK, RCLK networks, dual-regional clock region
Clocking regions for high fan-out applications	Quadrant region, dual-regional, entire device via global clock or regional clock networks	Quadrant region, dual-regional, entire device via GCLK or RCLK networks

Global Clock Network

Global clocks drive throughout the entire device, feeding all device quadrants. All resources within the device IOEs, adaptive logic modules (ALMs), digital signal processing (DSP) blocks, and all memory blocks can use the global clock networks as clock sources. These resources can also be used for control signals, such as clock enables and synchronous or asynchronous clears fed by an external pin. Internal logic can also drive the global clock networks for internally generated global clocks and asynchronous clears, clock enables, or other control signals with large fanout. Figure 7–39 shows the 16 dedicated CLK pins driving global clock networks.

Figure 7–39. Global Clocking *Note (1)*



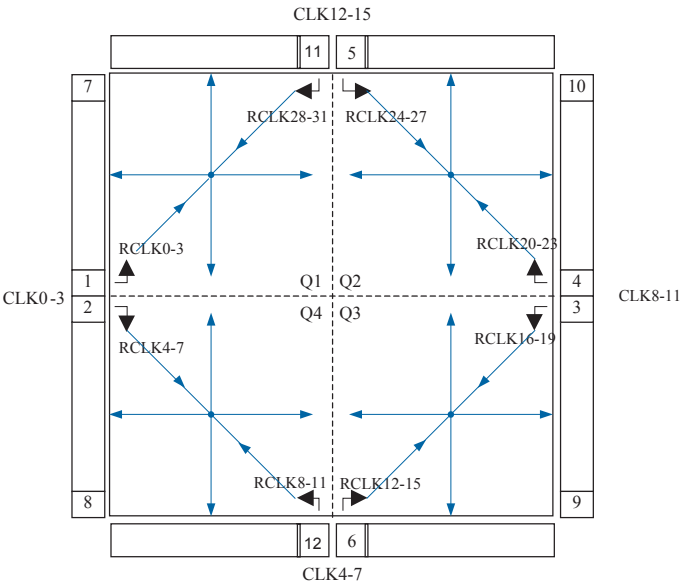
Note to Figure 7–39:

(1) Stratix II GX devices do not have PLLs 3, 4, 9, and 10 or clock pins 8, 9, 10, and 11.

Regional Clock Network

Eight regional clock networks within each quadrant of the Stratix II and Stratix II GX device are driven by the dedicated CLK[15..0] input pins or from PLL outputs. The regional clock networks only pertain to the quadrant they drive into. The regional clock networks provide the lowest clock delay and skew for logic contained within a single quadrant. Internal logic can also drive the regional clock networks for internally generated regional clocks and asynchronous clears, clock enables, or other control signals with large fanout.The CLK clock pins symmetrically drive the RCLK networks within a particular quadrant, as shown in Figure 7–40. Refer to Table 7–18 on page 7–67 and Table 7–19 on page 7–68 for RCLK connections from CLK pins and PLLs.

Figure 7–40. Regional Clocking *Note (1)*

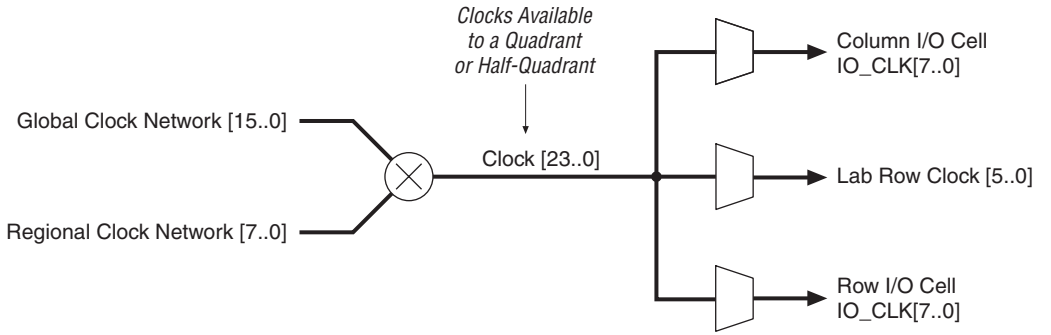


Note to Figure 7–40:
(1) Stratix II GX devices do not have PLLs 3, 4, 9, and 10 or clock pins 8, 9, 10, and 11.

Clock Sources Per Region

Each Stratix II and Stratix II GX device has 16 global clock networks and 32 regional clock networks that provide 48 unique clock domains for the entire device. There are 24 unique clocks available in each quadrant (16 global clocks and 8 regional clocks) as the input resources for registers (see Figure 7–41).

Figure 7–41. Hierarchical Clock Networks Per Quadrant



Stratix II and Stratix II GX clock networks provide three different clocking regions:

- Entire device clock region
- Quadrant clock region
- Dual-regional clock region

These clock network options provide more flexibility for routing signals that have high fan-out to improve the interface timing. By having various sized clock regions, it is possible to prioritize the number of registers the network can reach versus the total delay of the network.

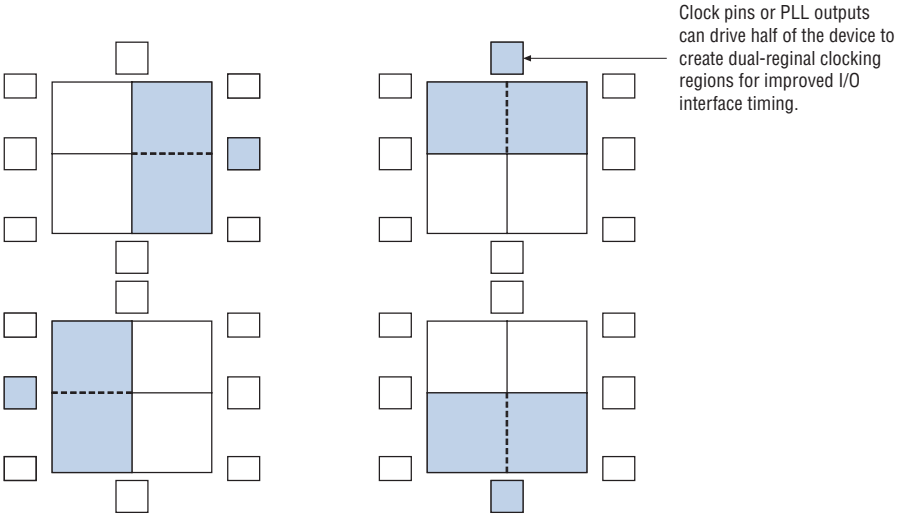
In the first clock scheme, a source (not necessarily a clock signal) drives a global clock network that can be routed through the entire device. This has the maximum delay for a low skew high fan-out signal but allows the signal to reach every block within the device. This is a good option for routing global resets or clear signals.

In the second clock scheme, a source drives a single-quadrant region. This represents the fastest, low-skew, high-fan-out signal-routing resource within a quadrant. The limitation to this resource is that it only covers a single quadrant.

In the third clock scheme, a single source (clock pin or PLL output) can generate a dual-regional clock by driving two regional clock network lines (one from each quadrant). This allows logic that spans multiple quadrants to utilize the same low-skew clock. The routing of this signal on an entire side has approximately the same speed as in a quadrant clock region. The internal logic-array routing that can drive a regional clock also supports this feature. This means internal logic can drive a

dual-regional clock network. Corner fast PLL outputs only span one quadrant and hence cannot form a dual-regional clock network. Figure 7–42 shows this feature pictorially.

Figure 7–42. Stratix II and Stratix II GX Dual-Regional Clock Region



The 16 clock input pins, enhanced or fast PLL outputs, and internal logic array can be the clock input sources to drive onto either global or regional clock networks. The CLKn pins also drive the global clock network as shown in Table 7–22 on page 7–72. Tables 7–18 and 7–19 for the connectivity between CLK pins as well as the global and regional clock networks.

Clock Inputs

The clock input pins CLK[15..0] are also used for high fan-out control signals, such as asynchronous clears, presets, clock enables, or protocol signals such as TRDY and IRDY for PCI through global or regional clock networks.

Internal Logic Array

Each global and regional clock network can also be driven by logic-array routing to enable internal logic to drive a high fan-out, low-skew signal.

PLL Outputs

All clock networks can be driven by the PLL counter outputs.

Table 7–18 shows the connection of the clock pins to the global clock resources. The reason for the higher level of connectivity is to support user controllable global clock multiplexing.

Table 7–18. Clock Input Pin Connectivity to Global Clock Networks

Clock Resource	CLK(p) (Pin)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
GCLK0	✓	✓														
GCLK1	✓	✓														
GCLK2			✓	✓												
GCLK3			✓	✓												
GCLK4					✓	✓										
GCLK5					✓	✓										
GCLK6							✓	✓								
GCLK7							✓	✓								
GCLK8											✓ (1)	✓ (1)				
GCLK9											✓ (1)	✓ (1)				
GCLK10									✓ (1)	✓ (1)						
GCLK11									✓ (1)	✓ (1)						
GCLK12															✓	✓
GCLK13															✓	✓
GCLK14													✓	✓		
GCLK15													✓	✓		

Note to Table 7–18:

(1) Clock pins 8, 9, 10, and 11 are not available in Stratix II GX devices. Therefore, these connections do not exist in Stratix II GX devices.

Table 7–19 summarizes the connectivity between the clock pins and the regional clock networks. Here, each clock pin can drive two regional clock networks, facilitating stitching of the clock networks to support the ability to drive two quadrants with the same clock or signal.

Table 7–19. Clock Input Pin Connectivity to Regional Clock Networks (Part 1 of 2)

Clock Resource	CLK(p) (Pin)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK0	✓															
RCLK1		✓														
RCLK2			✓													
RCLK3				✓												
RCLK4	✓															
RCLK5		✓														
RCLK6			✓													
RCLK7				✓												
RCLK8					✓											
RCLK9						✓										
RCLK10							✓									
RCLK11								✓								
RCLK12					✓											
RCLK13						✓										
RCLK14							✓									
RCLK15								✓								
RCLK16											✓ (1)					
RCLK17												✓ (1)				
RCLK18									✓ (1)							
RCLK19										✓ (1)						
RCLK20											✓ (1)					
RCLK21												✓ (1)				
RCLK22									✓ (1)							

Table 7–19. Clock Input Pin Connectivity to Regional Clock Networks (Part 2 of 2)

Clock Resource	CLK(p) (Pin)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK23										✓ (1)						
RCLK24															✓	
RCLK25																✓
RCLK26													✓			
RCLK27														✓		
RCLK28															✓	
RCLK29																✓
RCLK30													✓			
RCLK31														✓		

Note to Table 7–19:

(1) Clock pins 8, 9, 10, and 11 are not available in Stratix II GX devices. Therefore, these connections do not exist in Stratix II GX devices.

Clock Input Connections

Four CLK pins drive each enhanced PLL. You can use any of the pins for clock switchover inputs into the PLL. The CLK pins are the primary clock source for clock switchover, which is controlled in the Quartus II software. Enhanced PLLs 5, 6, 11, and 12 also have feedback input pins, as shown in Table 7–20.

Input clocks for fast PLLs 1, 2, 3, and 4 come from CLK pins. A multiplexer chooses one of two possible CLK pins to drive each PLL. This multiplexer is not a clock switchover multiplexer and is only used for clock input connectivity.

Either an FPLLCLK input pin or a CLK pin can drive the fast PLLs in the corners (7, 8, 9, and 10) when used for general-purpose applications. CLK pins cannot drive these fast PLLs in high-speed differential I/O mode.

Tables 7–20 and 7–21 show which PLLs are available in each Stratix II and Stratix II GX device, respectively, and which input clock pin drives which PLLs.

Table 7–20. Stratix II Device PLLs and PLL Clock Pin Drivers (Part 1 of 2)

Input Pin	All Devices						EP2S60 to EP2S180 Devices					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3	4	5	6	7	8	9	10	11	12
CLK0	✓	✓					✓ (1)	✓ (1)				
CLK1 (2)	✓	✓					✓ (1)	✓ (1)				
CLK2	✓	✓					✓ (1)	✓ (1)				
CLK3 (2)	✓	✓					✓ (1)	✓ (1)				
CLK4						✓						✓
CLK5						✓						✓
CLK6						✓						✓
CLK7						✓						✓
CLK8			✓	✓					✓ (1)	✓ (1)		
CLK9 (2)			✓	✓					✓ (1)	✓ (1)		
CLK10			✓	✓					✓ (1)	✓ (1)		
CLK11 (2)			✓	✓					✓ (1)	✓ (1)		
CLK12					✓						✓	
CLK13					✓						✓	
CLK14					✓						✓	
CLK15					✓						✓	
PLL5_FB					✓							
PLL6_FB						✓						
PLL11_FB											✓	
PLL12_FB												✓
PLL_ENA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
FPLL7CLK (2)							✓					
FPLL8CLK (2)								✓				
FPLL9CLK (2)									✓			

Table 7–20. Stratix II Device PLLs and PLL Clock Pin Drivers (Part 2 of 2)

Input Pin	All Devices						EP2S60 to EP2S180 Devices					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3	4	5	6	7	8	9	10	11	12
FPLL10CLK *(2)*										✓		

Notes to Table 7–20:
(1) Clock connection is available. For more information on the maximum frequency, contact Altera Applications.
(2) This is a dedicated high-speed clock input. For more information on the maximum frequency, contact Altera Applications.

Table 7–21. Stratix II GX Device PLLs and PLL Clock Pin Drivers (Part 1 of 2)

Input Pin	All Devices						EP2SGX60 to EP2SGX130 Devices					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3 *(1)*	4 *(1)*	5	6	7	8	9 *(1)*	10 *(1)*	11	12
CLK0	✓	✓					✓*(2)*	✓*(2)*				
CLK1 *(2)*	✓	✓					✓*(2)*	✓*(2)*				
CLK2	✓	✓					✓*(2)*	✓*(2)*				
CLK3 *(2)*	✓	✓					✓*(2)*	✓*(2)*				
CLK4						✓						✓
CLK5						✓						✓
CLK6						✓						✓
CLK7						✓						✓
CLK8 *(4)*												
CLK9 *(3)*, *(4)*												
CLK10 *(4)*												
CLK11 *(3)*, *(4)*												
CLK12					✓						✓	
CLK13					✓						✓	
CLK14					✓						✓	
CLK15					✓						✓	
PLL5_FB					✓							
PLL6_FB						✓						
PLL11_FB											✓	

Table 7–21. Stratix II GX Device PLLs and PLL Clock Pin Drivers (Part 2 of 2)

Input Pin	All Devices						EP2SGX60 to EP2SGX130 Devices					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3 *(1)*	4 *(1)*	5	6	7	8	9 *(1)*	10 *(1)*	11	12
PLL12_FB												✓
PLL_ENA	✓	✓			✓	✓	✓	✓			✓	✓
FPLL7CLK *(3)*							✓					
FPLL8CLK *(3)*								✓				
FPLL9CLK *(3)*												
FPLL10CLK *(3)*												

Notes to Table 7–21:

(1) PLLs 3, 4, 9, and 10 are not available in Stratix II GX devices.
(2) Clock connection is available. For more information on the maximum frequency, contact Altera Applications.
(3) This is a dedicated high-speed clock input. For more information on the maximum frequency, contact Altera Applications.
(4) Input pins `CLK[11..8]` are not available in Stratix II GX devices.

CLK(n) Pin Connectivity to Global Clock Networks

In Stratix II and Stratix II GX devices, the `clk(n)` pins can also feed the global clock network. Table 7–22 shows the `clk(n)` pin connectivity to global clock networks.

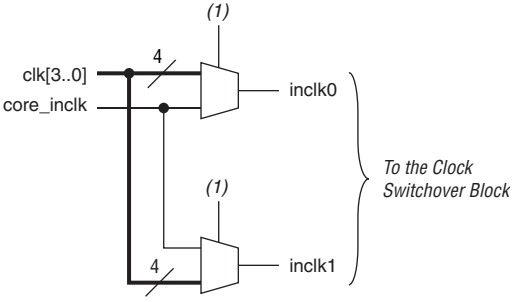
Table 7–22. CLK(n) Pin Connectivity to Global Clock Network

Clock Resource	CLK(n) pin							
	4	5	6	7	12	13	14	15
GCLK4	✓							
GCLK5		✓						
GCLK6			✓					
GCLK7				✓				
GCLK12							✓	
GCLK13								✓
GCLK14					✓			
GCLK15						✓		

Clock Source Control For Enhanced PLLs

The clock input multiplexer for enhanced PLLs is shown in Figure 7–43. This block allows selection of the PLL clock reference from several different sources. The clock source to an enhanced PLL can come from any one of four clock input pins `CLK[3..0]`, or from a logic-array clock, provided the logic array clock is driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL. The clock input pin connections to the respective enhanced PLLs are shown in Table 7–20 above. The multiplexer select lines are set in the configuration file only. Once programmed, this block cannot be changed without loading a new configuration file. The Quartus II software automatically sets the multiplexer select signals depending on the clock sources that a user selects in the design.

Figure 7–43. Enhanced PLL Clock Input Multiplex Logic



Note to Figure 7–43:

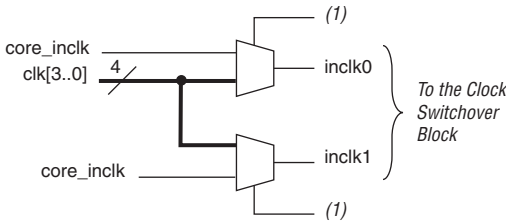
(1) The input clock multiplexing is controlled through a configuration file only and cannot be dynamically controlled in user mode.

Clock Source Control for Fast PLLs

Each center fast PLL has five clock input sources, four from clock input pins, and one from a logic array signal, provided the logic array signal is driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL. When using clock input pins as the clock source, you can perform manual clock switchover among the input clock sources.

The clock input multiplexer control signals for performing clock switchover are from core signals. Figure 7–44 shows the clock input multiplexer control circuit for a center fast PLL.

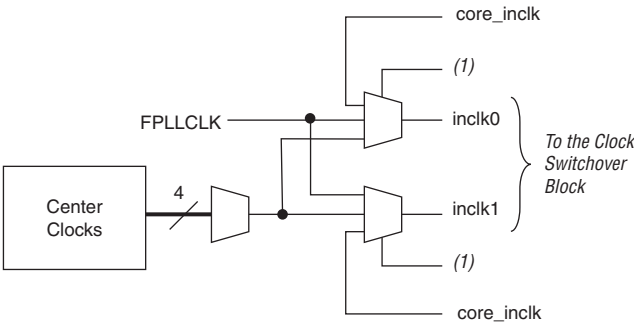
Figure 7–44. Center Fast PLL Clock Input Multiplexer Control



Note to Figure 7–44:
(1) The input clock multiplexing is controlled through a configuration file only and cannot be dynamically controlled in user mode.

Each corner fast PLL has three clock input sources, one from a dedicated corner clock input pin, one from a center clock input pin, and one from a logic array clock, provided the logic array signal is driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL. Figure 7–45 shows a block diagram showing the clock input multiplexer control circuit for a corner fast PLL. Only the corner FPLLCLK pin is fully compensated.

Figure 7–45. Corner Fast PLL Clock Input Multiplexer Control



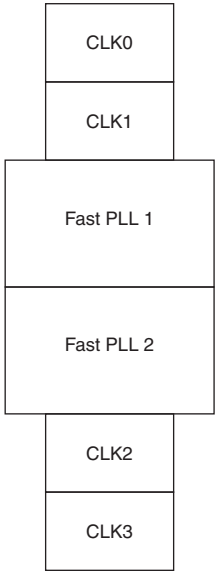
Note to Figure 7–45:
(1) The input clock multiplexing is controlled through a configuration file only and cannot be dynamically controlled in user mode.

Delay Compensation for Fast PLLs

Each center fast PLL can be fed by any one of four possible input clock pins. Among the four clock input signals, only two are fully compensated, i.e., the clock delay to the fast PLL matches the delay in the data input path when used in the LVDS receiver mode. The two clock inputs that match the data input path are located right next to the fast PLL. The two clock inputs that do not match the data input path are located next to the neighboring fast PLL. Figure 7–46 shows the above description for the left-side center fast PLL pair. If the PLL is used in non-LVDS modes, then any of the four dedicated clock inputs can be used and are compensated.

Fast PLL 1 and PLL 2 can choose among `CLK[3..0]` as the clock input source. However, for fast PLL 1, only `CLK0` and `CLK1` have their delay matched to the data input path delay when used in the LVDS receiver mode operation. The delay from `CLK2` or `CLK3` to fast PLL 1 does not match the data input delay. For fast PLL 2, only `CLK2` and `CLK3` have their delay matched to the data input path delay in LVDS receiver mode operation. The delay from `CLK0` or `CLK1` to fast PLL 2 does not match the data input delay. The same arrangement applies to the right side center fast PLL pair. For corner fast PLLs, only the corner `FPLLCLK` pins are fully compensated. For LVDS receiver operation, it is recommended to use the delay compensated clock pins only.

Figure 7–46. Delay Compensated Clock Input Pins for Center Fast PLL Pair



Clock Output Connections

Enhanced PLLs have outputs for eight regional clock outputs and four global clock outputs. There is line sharing between clock pins, global and regional clock networks and all PLL outputs. See Tables 7–18 through 7–23 and Figures 7–47 through 7–53 to validate your clocking scheme. The Quartus II software automatically maps to regional and global clocks to avoid any restrictions. Enhanced PLLs 5, 6, 11, and 12 drive out to single-ended pins as shown in Table 7–23.

You can connect each fast PLL 1, 2, 3, or 4 output (`C0`, `C1`, `C2`, and `C3`) to either a global or a regional clock. There is line sharing between clock pins, `FPLLCLK` pins, global and regional clock networks, and all PLL outputs. The Quartus II software will automatically map to regional and global clocks to avoid any restrictions.

Figure 7–47 shows the clock input and output connections from the enhanced PLLs.

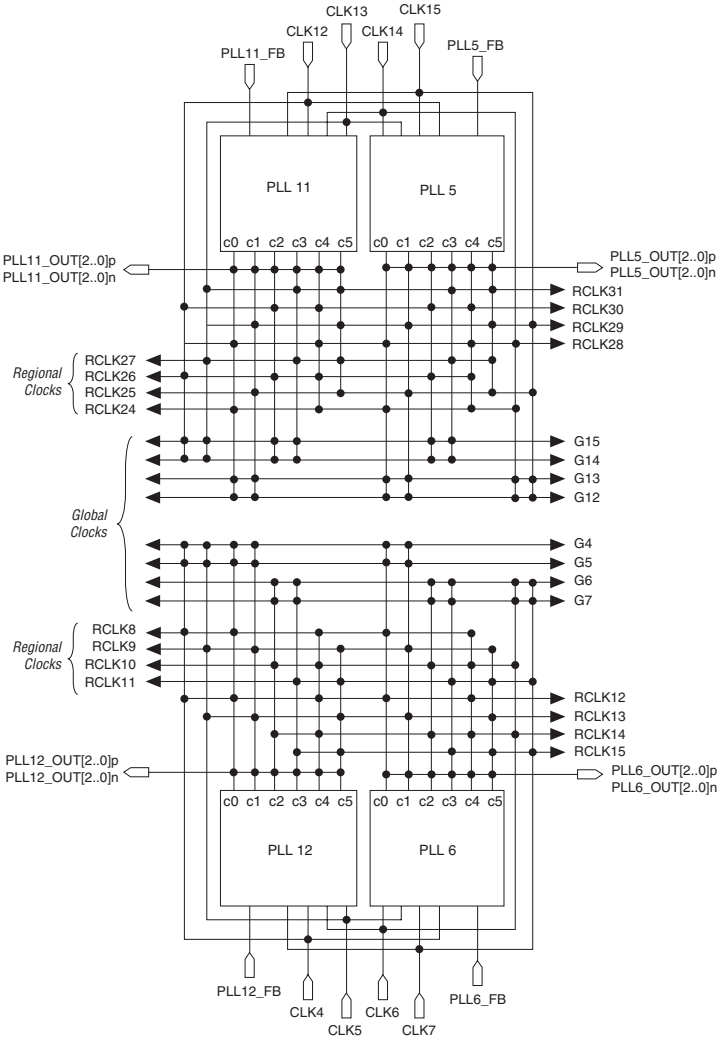
EP2S15, EP2S30, and EP2SGX30 devices have only two enhanced PLLs (5, 6), but the connectivity from these two PLLs to the global or regional clock networks remains the same.

The EP2S60 device in the 1,020-pin package contains 12 PLLs. EP2S60 devices in the 484-pin and 672-pin packages contain fast PLLs 1–4 and enhanced PLLs 5 and 6.

EP2S90 devices in the 1020-pin and 1508-pin packages contain 12 PLLs. EP2S90 devices in the 484-pin and 780-pin packages contain fast PLLs 1–4 and enhanced PLLs 5 and 6.

EP2S130 devices in the 1020-pin and 1508-pin packages contain 12 PLLs. The EP2S130 device in the 780-pin package contains fast PLLs 1–4 and enhanced PLLs 5 and 6.

Figure 7–47. Stratix II and Stratix II GX Top and Bottom Enhanced PLLs, Clock Pin and Logic Array Signal Connectivity to Global and Regional Clock Networks *Notes (1) and (2)*



Note to Figure 7–47:

(1) The redundant connection dots facilitate stitching of the clock networks to support the ability to drive two quadrants with the same clock.

(2) The enhanced PLLs can also be driven through the global or regional clock networks. The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

Tables 7–23 and 7–24 show the global and regional clocks that the PLL outputs drive.

Table 7–23. Stratix II Global and Regional Clock Outputs From PLLs (Part 1 of 3)

Clock Network	PLL Number and Type											
	EP2S15 through EP2S30 Devices						EP2S60 through EP2S180 Devices					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3	4	5	6	7	8	9	10	11	12
GCLK0	✓	✓					✓	✓				
GCLK1	✓	✓					✓	✓				
GCLK2	✓	✓					✓	✓				
GCLK3	✓	✓					✓	✓				
GCLK4						✓						✓
GCLK5						✓						✓
GCLK6						✓						✓
GCLK7						✓						✓
GCLK8			✓	✓					✓	✓		
GCLK9			✓	✓					✓	✓		
GCLK10			✓	✓					✓	✓		
GCLK11			✓	✓					✓	✓		
GCLK12					✓						✓	
GCLK13					✓						✓	
GCLK14					✓						✓	
GCLK15					✓						✓	
RCLK0	✓	✓					✓					
RCLK1	✓	✓					✓					
RCLK2	✓	✓					✓					
RCLK3	✓	✓					✓					
RCLK4	✓	✓						✓				
RCLK5	✓	✓						✓				
RCLK6	✓	✓						✓				
RCLK7	✓	✓						✓				
RCLK8						✓						✓
RCLK9						✓						✓

Table 7–23. Stratix II Global and Regional Clock Outputs From PLLs (Part 2 of 3)

Clock Network	PLL Number and Type											
	EP2S15 through EP2S30 Devices						EP2S60 through EP2S180 Devices					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3	4	5	6	7	8	9	10	11	12
RCLK10						✓						✓
RCLK11						✓						✓
RCLK12						✓						✓
RCLK13						✓						✓
RCLK14						✓						✓
RCLK15						✓						✓
RCLK16			✓	✓					✓			
RCLK17			✓	✓					✓			
RCLK18			✓	✓					✓			
RCLK19			✓	✓					✓			
RCLK20			✓	✓						✓		
RCLK21			✓	✓						✓		
RCLK22			✓	✓						✓		
RCLK23			✓	✓						✓		
RCLK24					✓						✓	
RCLK25					✓						✓	
RCLK26					✓						✓	
RCLK27					✓						✓	
RCLK28					✓						✓	
RCLK29					✓						✓	
RCLK30					✓						✓	
RCLK31					✓						✓	
External Clock Output												
PLL5_OUT[3..0]p/n					✓							
PLL6_OUT[3..0]p/n						✓						

Table 7–23. Stratix II Global and Regional Clock Outputs From PLLs (Part 3 of 3)

Clock Network	PLL Number and Type											
	EP2S15 through EP2S30 Devices						EP2S60 through EP2S180 Devices					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3	4	5	6	7	8	9	10	11	12
PLL11_OUT[3..0]p/n											✓	
PLL12_OUT[3..0]p/n												✓

Table 7–24. Stratix II GX Global and Regional Clock Outputs From PLLs (Part 1 of 3)

Clock Network	PLL Number and Type											
	EP2SGX30 Devices						EP2SGX60 through EP2SGX130 Devices *Notes (2)*, *(3)*, and *(4)*					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3 *(1)*	4 *(1)*	5	6	7	8	9 *(1)*	10 *(1)*	11	12
GCLK0	✓	✓					✓	✓				
GCLK1	✓	✓					✓	✓				
GCLK2	✓	✓					✓	✓				
GCLK3	✓	✓					✓	✓				
GCLK4						✓						✓
GCLK5						✓						✓
GCLK6						✓						✓
GCLK7						✓						✓
GCLK8												
GCLK9												
GCLK10												
GCLK11												
GCLK12					✓						✓	
GCLK13					✓						✓	
GCLK14					✓						✓	
GCLK15					✓						✓	

Table 7–24. Stratix II GX Global and Regional Clock Outputs From PLLs (Part 2 of 3)

Clock Network	PLL Number and Type											
	EP2SGX30 Devices						EP2SGX60 through EP2SGX130 Devices *Notes (2), (3), and (4)*					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3 *(1)*	4 *(1)*	5	6	7	8	9 *(1)*	10 *(1)*	11	12
RCLK0	✓	✓					✓					
RCLK1	✓	✓					✓					
RCLK2	✓	✓					✓					
RCLK3	✓	✓					✓					
RCLK4	✓	✓						✓				
RCLK5	✓	✓						✓				
RCLK6	✓	✓						✓				
RCLK7	✓	✓						✓				
RCLK8						✓						✓
RCLK9						✓						✓
RCLK10						✓						✓
RCLK11						✓						✓
RCLK12						✓						✓
RCLK13						✓						✓
RCLK14						✓						✓
RCLK15						✓						✓
RCLK16												
RCLK17												
RCLK18												
RCLK19												
RCLK20												
RCLK21												
RCLK22												
RCLK23												
RCLK24					✓						✓	
RCLK25					✓						✓	
RCLK26					✓						✓	
RCLK27					✓						✓	

Table 7–24. Stratix II GX Global and Regional Clock Outputs From PLLs (Part 3 of 3)

Clock Network	PLL Number and Type											
	EP2SGX30 Devices						EP2SGX60 through EP2SGX130 Devices *Notes (2)*, *(3)*, and *(4)*					
	Fast PLLs				Enhanced PLLs		Fast PLLs				Enhanced PLLs	
	1	2	3 *(1)*	4 *(1)*	5	6	7	8	9 *(1)*	10 *(1)*	11	12
RCLK28					✓						✓	
RCLK29					✓						✓	
RCLK30					✓						✓	
RCLK31					✓						✓	
External Clock Output												
PLL5_OUT[3..0]p/n					✓							
PLL6_OUT[3..0]p/n						✓						
PLL11_OUT[3..0]p/n											✓	
PLL12_OUT[3..0]p/n												✓

Note to Table 7–24:

(1) PLLs 3, 4, 9, and 10 are not available in Stratix II GX devices.

(2) The EP2S60 device in the 1,020-pin package contains 12 PLLs. EP2S60 devices in the 484-pin and 672-pin packages contain fast PLLs 1–4 and enhanced PLLs 5 and 6.

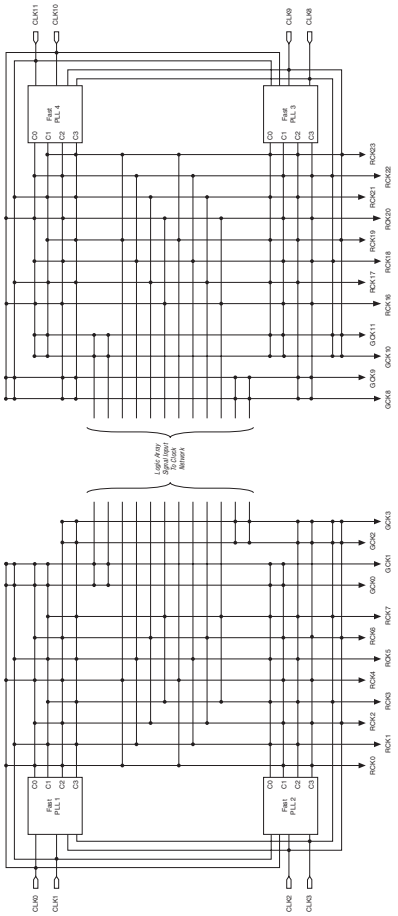
(3) EP2S90 devices in the 1020-pin and 1508-pin packages contain 12 PLLs. EP2S90 devices in the 484-pin and 780-pin packages contain fast PLLs 1–4 and enhanced PLLs 5 and 6.

(4) EP2S130 devices in the 1020-pin and 1508-pin packages contain 12 PLLs. The EP2S130 device in the 780-pin package contains fast PLLs 1–4 and enhanced PLLs 5 and 6.

The fast PLLs also drive high-speed SERDES clocks for differential I/O interfacing. For information on these FPLLCLK pins, contact Altera Applications.

Figures 7–48 through 7–51 show the global and regional clock input and output connections from the Stratix II fast PLLs.

Figure 7–48. Stratix II Center Fast PLLs, Clock Pin and Logic Array Signal Connectivity to Global and Regional Clock Networks *Notes (1) and (2)*

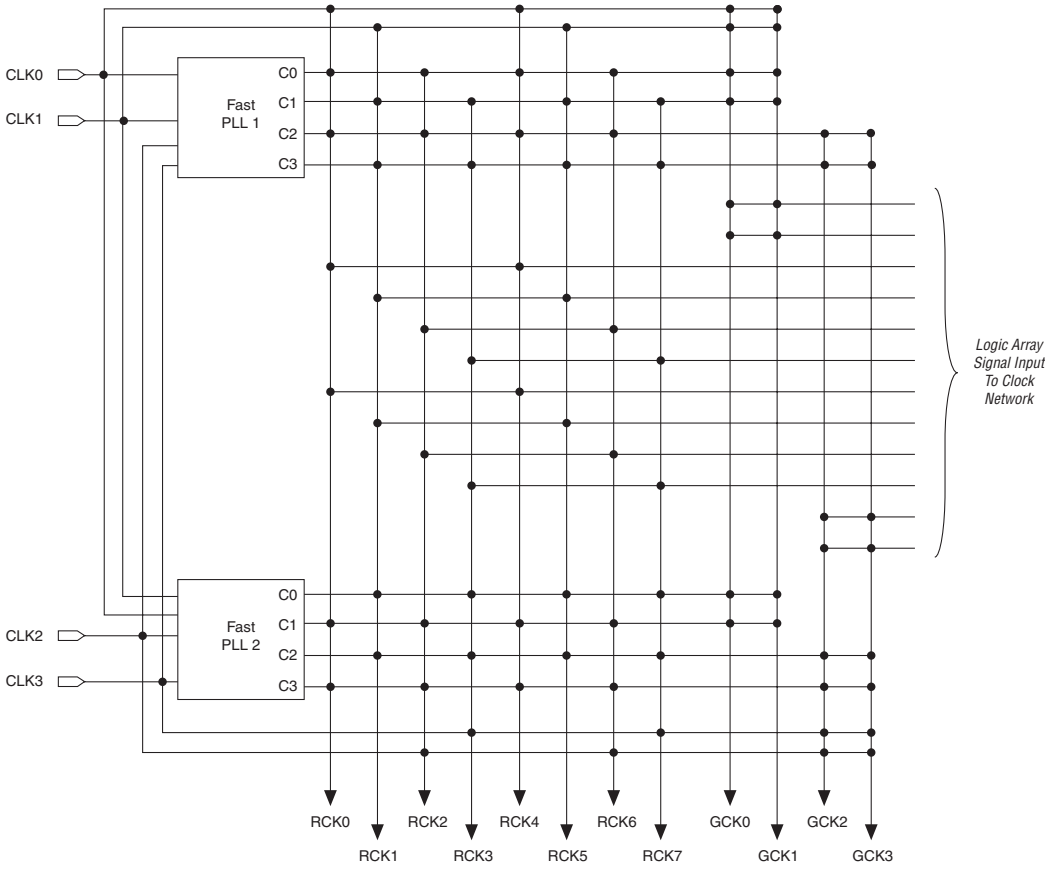


Note to Figure 7–48:

(1) The redundant connection dots facilitate stitching of the clock networks to support the ability to drive two quadrants with the same clock.

(2) The global or regional clocks in a fast PLL's quadrant can drive the fast PLL input. The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

Figure 7–49. Stratix II GX Center Fast PLLs, Clock Pin and Logic Array Signal Connectivity to Global and Regional Clock Networks *Notes (1) and (2)*

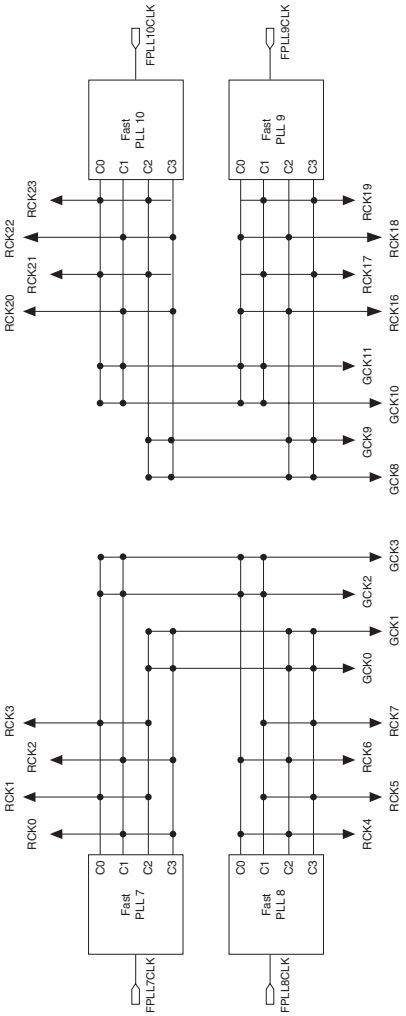


Note to Figure 7–49:

(1) The redundant connection dots facilitate stitching of the clock networks to support the ability to drive two quadrants with the same clock.

(2) The global or regional clocks in a fast PLL's quadrant can drive the fast PLL input. The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

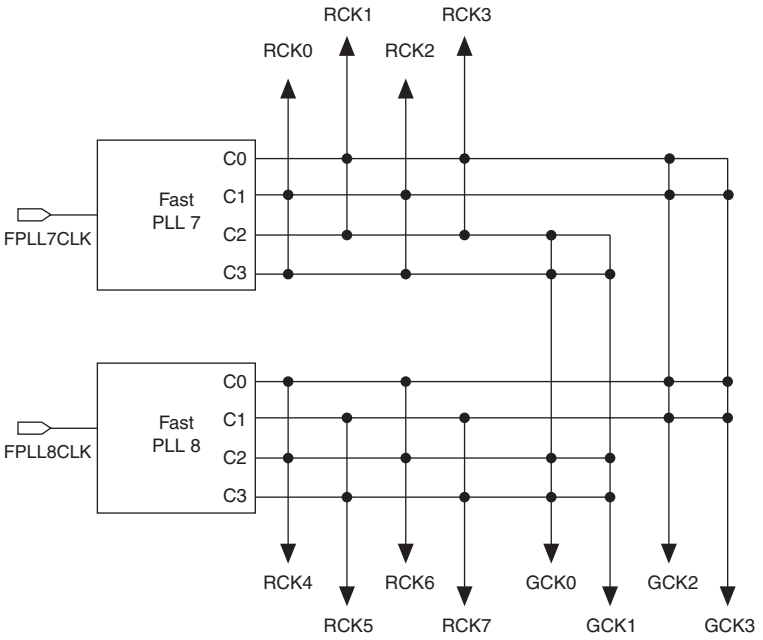
Figure 7–50. Stratix II Corner Fast PLLs, Clock Pin and Logic Array Signal Connectivity to Global and Regional Clock Networks *Note (1)*



Note to Figure 7–50:

(1) The corner FPLLs can also be driven through the global or regional clock networks. The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

Figure 7–51. Stratix II GX Corner Fast PLLs, Clock Pin and Logic Array Signal Connectivity to Global and Regional Clock Networks *Note (1)*



Note to Figure 7–51:

(1) The corner FPLLs can also be driven through the global or regional clock networks. The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block, provided the clock control block is fed by an output from another PLL or a pin-driven dedicated global or regional clock. An internally generated global signal cannot drive the PLL.

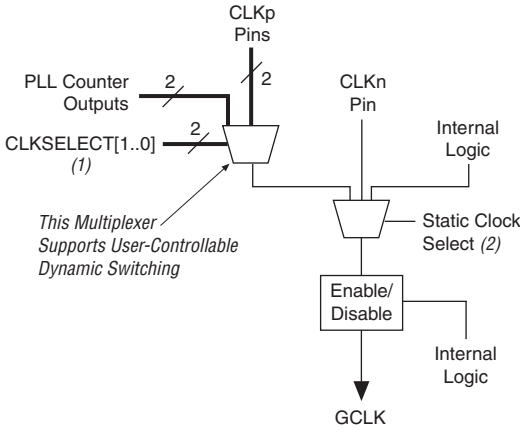
Clock Control Block

Each global and regional clock has its own clock control block. The control block has two functions:

- Clock source selection (dynamic selection for global clocks)
- Clock power-down (dynamic clock enable or disable)

Figures 7–52 and 7–53 show the global clock and regional clock select blocks, respectively.

Figure 7–52. Stratix II Global Clock Control Block

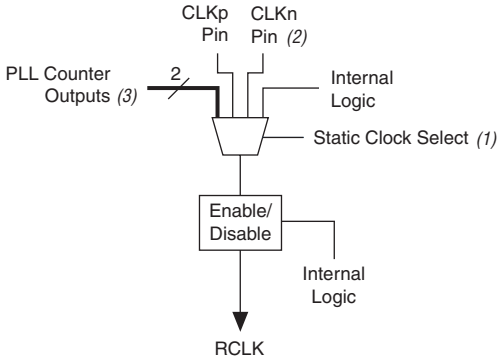


Notes to Figure 7–52:

(1) These clock select signals can only be dynamically controlled through internal logic when the device is operating in user mode.

(2) These clock select signals can only be set through a configuration file and cannot be dynamically controlled during user-mode operation.

Figure 7–53. Regional Clock Control Block



Notes to Figure 7–53:

(1) These clock select signals can only be dynamically controlled through a configuration file and cannot be dynamically controlled during user-mode operation.

(2) Only the CLK*n* pins on the top and bottom for the device feed to regional clock select blocks.

For the global clock select block, the clock source selection can be controlled either statically or dynamically. You have the option to statically select the clock source in configuration file generated by the Quartus II software, or you can control the selection dynamically by using internal logic to drive the multiplexer select inputs. When selecting statically, the clock source can be set to any of the inputs to the select multiplexer. When selecting the clock source dynamically, you can either select two PLL outputs (such as `CLK0` or `CLK1`), or a combination of clock pins or PLL outputs.

When using the `altclkctrl` megafunction to implement clock source (dynamics) selection, the inputs from the clock pins feed the `inclock[0..1]` ports of the multiplexer, while the PLL outputs feed the `inclock[2..3]` ports. You can choose from among these inputs using the `CLKSELECT[1..0]` signal.

For the regional clock select block, the clock source selection can only be controlled statically using configuration bits. Any of the inputs to the clock select multiplexer can be set as the clock source.

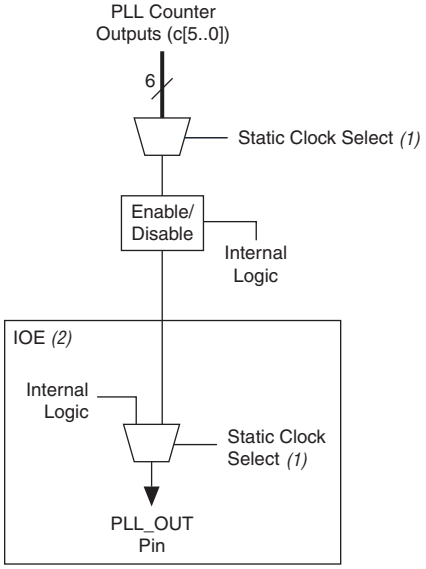
The Stratix II and Stratix II GX clock networks can be disabled (powered down) by both static and dynamic approaches. When a clock net is powered down, all the logic fed by the clock net is in an off-state, thereby reducing the overall power consumption of the device.

The global and regional clock networks that are not used are automatically powered down through configuration bit settings in the configuration file (SRAM Object File (**.sof**) or Programmer Object File (**.pof**)) generated by the Quartus II software.

The dynamic clock enable or disable feature allows the internal logic to control power up or down synchronously on GCLK and RCLK nets, including dual-regional clock regions. This function is independent of the PLL and is applied directly on the clock network, as shown in Figure 7–52 on page 7–87 and Figure 7–53 on page 7–88.

The input clock sources and the clkena signals for the global and regional clock network multiplexers can be set through the Quartus II software using the altclkctrl megafunction. The dedicated external clock output pins can also be enabled or disabled using the altclkctrl megafunction. Figure 7–54 shows the external PLL output clock control block.

Figure 7–54. Stratix II External PLL Output Clock Control Block



Notes to Figure 7–54:

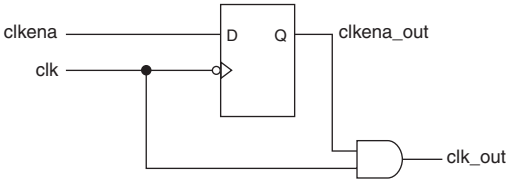
(1) These clock select signals can only be set through a configuration file and cannot be dynamically controlled during user mode operation.

(2) The clock control block feeds to a multiplexer within the PLL_OUT pin's IOE. The PLL_OUT pin is a dual-purpose pin. Therefore, this multiplexer selects either an internal signal or the output of the clock control block.

clkena Signals

Figure 7–55 shows how clkena is implemented.

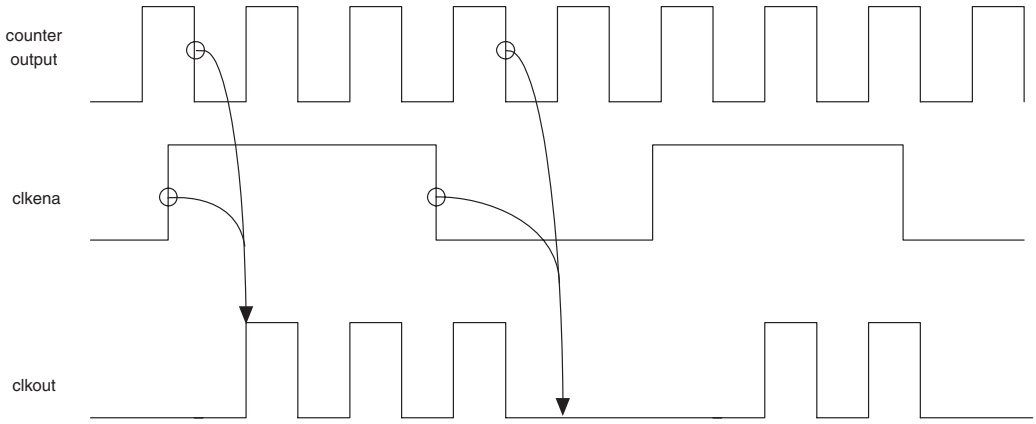
Figure 7–55. clkena Implementation



In Stratix II devices, the clkena signals are supported at the clock network level. This allows you to gate off the clock even when a PLL is not being used.

The clkena signals can also be used to control the dedicated external clocks from enhanced PLLs. Upon re-enabling, the PLL does not need a resynchronization or relock period unless the PLL is using external feedback mode. Figure 7–56 shows the waveform example for a clock output enable. clkena is synchronous to the falling edge of the counter output.

Figure 7–56. Clkena Signals



Note to Figure 7–56

(1) The clkena signals can be used to enable or disable the global and regional networks or the PLL_OUT pins.

The PLL can remain locked independent of the clkena signals since the loop-related counters are not affected. This feature is useful for applications that require a low power or sleep mode. Upon re-enabling, the PLL does not need a resynchronization or relock period. The clkena signal can also disable clock outputs if the system is not tolerant to frequency overshoot during resynchronization.

Conclusion

Stratix II and Stratix II GX device enhanced and fast PLLs provide you with complete control of device clocks and system timing. These PLLs are capable of offering flexible system-level clock management that was previously only available in discrete PLL devices. The embedded PLLs meet and exceed the features offered by these high-end discrete devices, reducing the need for other timing devices in the system.

Referenced Documents

This chapter references the following documents:

- *altpll Megafunction User Guide*
- *AN 367: Implementing PLL Reconfiguration in Stratix II Devices*
- *Configuring Stratix II and Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* (or the *Stratix II Device Handbook)*
- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook* (or the *Stratix II Device Handbook*)
- *Selectable I/O Standards in Stratix II and Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* (or the *Stratix II Device Handbook*)
- Verification, volume 3 of the *Quartus II Development Software Handbook*

Document Revision History

Table 7–25 shows the revision history for this chapter.

Table 7–25. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v4.5	Updated "External Clock Outputs" section.	—
	Added the "Referenced Documents" section.	—
	Minor text edis.	—

Table 7–25. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 6. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
May 2007, v4.4	Updated Table 7–6.	—
	Updated notes to: • Figure 7–7 • Figure 7–47 • Figure 7–48 • Figure 7–49 • Figure 7–50 • Figure 7–51	—
	Updated the “Clock Source Control For Enhanced PLLs” section.	—
	Updated the “Clock Source Control for Fast PLLs” section.	—
February 2007 v4.3	Added “Document Revision History” section to this chapter.	—
	Deleted paragraph beginning with “The Stratix II GX PLLs have the ability...” in the “Enhanced Lock Detect Circuit” section.	—
April 2006, v4.2	Chapter updated as part of the *Stratix II Device Handbook* update.	—
No change	Formerly chapter 5. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
December 2005, v4.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—
October 2005 v4.0	Added chapter to the *Stratix II GX Device Handbook*.	—

Section III. Memory

This section provides information on the TriMatrix™ embedded memory blocks internal to Stratix® II GX devices and the supported external memory interfaces.

This section contains the following chapters:

- Chapter 8, TriMatrix Embedded Memory Blocks in Stratix II and Stratix II GX Devices
- Chapter 9, External Memory Interfaces in Stratix II and Stratix II GX Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

8. TriMatrix Embedded Memory Blocks in Stratix II and Stratix II GX Devices

SII52002-4.5

Introduction

Stratix® II and Stratix II GX devices feature the TriMatrix™ memory structure, consisting of three sizes of embedded RAM blocks that efficiently address the memory needs of FPGA designs.

TriMatrix memory includes 512-bit M512 blocks, 4-Kbit M4K blocks, and 512-Kbit M-RAM blocks, which are each configurable to support many features. TriMatrix memory provides up to 9 megabits of RAM at up to 550 MHz operation, and up to 16 terabits per second of total memory bandwidth per device. This chapter describes TriMatrix memory blocks, modes, and features.

TriMatrix Memory Overview

The TriMatrix architecture provides complex memory functions for different applications in FPGA designs. For example, M512 blocks are used for first-in first-out (FIFO) functions and clock domain buffering where memory bandwidth is critical; M4K blocks are ideal for applications requiring medium-sized memory, such as asynchronous transfer mode (ATM) cell processing; and M-RAM blocks are suitable for large buffering applications, such as internet protocol (IP) packet buffering and system cache.

The TriMatrix memory blocks support various memory configurations, including single-port, simple dual-port, true dual-port (also known as bidirectional dual-port), shift register, and read-only memory (ROM) modes. The TriMatrix memory architecture also includes advanced features and capabilities, such as parity-bit support, byte enable support, pack mode support, address clock enable support, mixed port width support, and mixed clock mode support.

When applied to input registers, the asynchronous clear signal for the TriMatrix embedded memory immediately clears the input registers. However, the output of the memory block does not show the effects until the next clock edge. When applied to output registers, the asynchronous clear signal clears the output registers and the effects are seen immediately.

Table 8–1 summarizes the features supported by the three sizes of TriMatrix memory.

Table 8–1. Summary of TriMatrix Memory Features

Feature	M512 Blocks	M4K Blocks	M-RAM Blocks
Maximum performance	500 MHz	550 MHz	420 MHz
Total RAM bits (including parity bits)	576	4,608	589,824
Configurations	512 × 1 256 × 2 128 × 4 64 × 8 64 × 9 32 × 16 32 × 18	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 8K × 64 8K × 72 4K × 128 4K × 144
Parity bits	✓	✓	✓
Byte enable	✓	✓	✓
Pack mode		✓	✓
Address clock enable		✓	✓
Single-port memory	✓	✓	✓
Simple dual-port memory	✓	✓	✓
True dual-port memory		✓	✓
Embedded shift register	✓	✓	
ROM	✓	✓	
FIFO buffer	✓	✓	✓
Simple dual-port mixed width support	✓	✓	✓
True dual-port mixed width support		✓	✓
Memory initialization file (**.mif**)	✓	✓	
Mixed-clock mode	✓	✓	✓
Power-up condition	Outputs cleared	Outputs cleared	Outputs unknown
Register clears	Output registers only	Output registers only	Output registers only
Same-port read-during-write	New data available at positive clock edge	New data available at positive clock edge	New data available at positive clock edge
Mixed-port read-during-write	Outputs set to unknown or old data	Outputs set to unknown or old data	Unknown output

Tables 8–2 and 8–3 show the capacity and distribution of the TriMatrix memory blocks in each Stratix II and Stratix II GX family member, respectively.

Table 8–2. TriMatrix Memory Capacity and Distribution in Stratix II Devices

Device	M512 Columns/Blocks	M4K Columns/Blocks	M-RAM Blocks	Total RAM Bits
EP2S15	4/104	3/78	0	419,328
EP2S30	6/202	4/144	1	1,369,728
EP2S60	7/329	5/255	2	2,544,192
EP2S90	8/488	6/408	4	4,520,448
EP2S130	9/699	7/609	6	6,747,840
EP2S180	11/930	8/768	9	9,383,040

Table 8–3. TriMatrix Memory Capacity and Distribution in Stratix II GX Devices

Device	M512 Columns/Blocks	M4K Columns/Blocks	M-RAM Blocks	Total RAM Bits
EP2SGX30C EP2SGX30D	6/202	4/144	1	1,369,728
EP2SGX60C EP2SGX60D EP2SGX60E	7/329	5/255	2	2,544,192
EP2SGX90E EP2SGX90F	8/488	6/408	4	4,520,448
EP2SGX130G	9/699	7/609	6	6,747,840

Parity Bit Support

All TriMatrix memory blocks (M512, M4K, and M-RAM) support one parity bit for each byte.

Parity bits add to the amount of memory in each random access memory (RAM) block. For example, the M512 block has 576 bits, 64 of which are optionally used for parity bit storage. The parity bit, along with logic implemented in adaptive logic modules (ALMs), implements parity checking for error detection to ensure data integrity. Parity-size data words can also be used for other purposes such as storing user-specified control bits.

Refer to the *Using Parity to Detect Memory Errors* white paper for more information on using the parity bit to detect memory errors.

Byte Enable Support

All TriMatrix memory blocks support byte enables that mask the input data so that only specific bytes, nibbles, or bits of data are written. The unwritten bytes or bits retain the previous written value. The write enable (`wren`) signals, along with the byte enable (`byteena`) signals, control the RAM blocks' write operations. The default value for the byte enable signals is high (enabled), in which case writing is controlled only by the write enable signals. There is no clear port to the byte enable registers.

M512 Blocks

M512 blocks support byte enables for data widths of 16 and 18 bits only. For memory block configurations with widths of less than two bytes (×16/×18), the byte-enable feature is not supported. For memory configurations less than two bytes wide, the write enable or clock enable signals can optionally be used to control the write operation. Table 8–4 summarizes the byte selection.

Table 8–4. Byte Enable for Stratix II and Stratix II GX M512 Blocks *Note (1)*

byteena[1..0]	data ×16	data ×18
[0] = 1	[7..0]	[8..0]
[1] = 1	[15..8]	[17..9]

Note to Table 8–4:
(1) Any combination of byte enables is possible.

M4K Blocks

M4K blocks support byte enables for any combination of data widths of 16, 18, 32, and 36 bits only. For memory block configurations with widths of less than two bytes (×16/×18), the byte-enable feature is not supported. For memory configurations less than two bytes wide, the write enable or clock enable signals can optionally be used to control the write operation.

Table 8–5 summarizes the byte selection.

Table 8–5. Byte Enable for Stratix II and Stratix II GX M4K Blocks *Note (1)*

byteena [3..0]	data ×16	data ×18	data ×32	data ×36
[0] = 1	[7..0]	[8..0]	[7..0]	[8..0]
[1] = 1	[15..8]	[17..9]	[15..8]	[17..9]
[2] = 1	-	-	[23..16]	[26..18]
[3] = 1	-	-	[31..24]	[35..27]

Note to Table 8–5:
(1) Any combination of byte enables is possible.

M-RAM Blocks

M-RAM blocks support byte enables for any combination of data widths of 16, 18, 32, 36, 64, and 72 bits. For memory block configurations with widths of less than two bytes (×16/×18), the byte-enable feature is not supported. In the ×128 and ×144 simple dual-port modes, the two sets of byte enable signals (`byteena_a` and `byteena_b`) combine to form the necessary 16 byte enables. In ×128 and ×144 modes, byte enables are only supported when using single clock mode. However, the Quartus II software can implement byte enables in other clocking modes for ×128 or ×144 widths but will use twice as many M-RAM resources. If clock enables are used in ×128 or ×144 mode, you must use the same clock enable setting for both the A and B ports. Table 8–6 summarizes the byte selection for M-RAM blocks.

Table 8–6. Byte Enable for Stratix II and Stratix II GX M-RAM Blocks *Note (1)*

byteena	data ×16	data ×18	data ×32	data ×36	data ×64	data ×72
[0] = 1	[7..0]	[8..0]	[7..0]	[8..0]	[7..0]	[8..0]
[1] = 1	[15..8]	[17..9]	[15..8]	[17..9]	[15..8]	[17..9]
[2] = 1	-	-	[23..16]	[26..18]	[23..16]	[26..18]
[3] = 1	-	-	[31..24]	[35..27]	[31..24]	[35..27]
[4] = 1	-	-	-	-	[39..32]	[44..36]
[5] = 1	-	-	-	-	[47..40]	[53..45]
[6] = 1	-	-	-	-	[55..48]	[62..54]
[7] = 1	-	-	-	-	[63..56]	[71..63]

Note to Table 8–6:
(1) Any combination of byte enables is possible.

Table 8–7 summarizes the byte selection for ×144 mode.

Table 8–7. Stratix II and Stratix II GX M-RAM Combined Byte Selection for ×144 Mode *Note (1)*

byteena	data ×128	data ×144
[0] = 1	[7..0]	[8..0]
[1] = 1	[15..8]	[17..9]
[2] = 1	[23..16]	[26..18]
[3] = 1	[31..24]	[35..27]
[4] = 1	[39..32]	[44..36]
[5] = 1	[47..40]	[53..45]
[6] = 1	[55..48]	[62..54]
[7] = 1	[63..56]	[71..63]
[8] = 1	[71..64]	[80..72]
[9] = 1	[79..72]	[89..73]
[10] = 1	[87..80]	[98..90]
[11] = 1	[95..88]	[107..99]
[12] = 1	[103..96]	[116..108]
[13] = 1	[111..104]	[125..117]
[14] = 1	[119..112]	[134..126]
[15] = 1	[127..120]	[143..135]

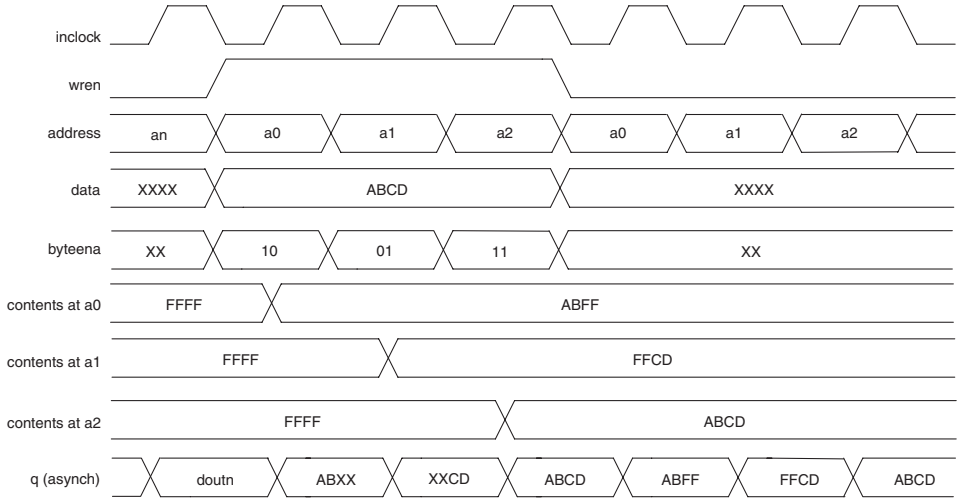
Note to Table 8–7:
(1) Any combination of byte enables is possible.

Byte Enable Functional Waveform

Figure 8–1 shows how the write enable (`wren`) and byte enable (`byteena`) signals control the operations of the RAM.

When a byte enable bit is de-asserted during a write cycle, the corresponding data byte output appears as a "don't care" or unknown value. When a byte enable bit is asserted during a write cycle, the corresponding data byte output will be the newly written data.

Figure 8–1. Stratix II and Stratix II GX Byte Enable Functional Waveform



For more information about MRAM and byte enable for the Stratix II device family, refer to the *Stratix II FPGA Errata Sheet* at the Altera web site at www.altera.com.

Pack Mode Support

Stratix II and Stratix II GX M4K and M-RAM memory blocks support pack mode. In M4K and M-RAM memory blocks, two single-port memory blocks can be implemented in a single block under the following conditions:

- Each of the two independent block sizes is equal to or less than half of the M4K or M-RAM block size.
- Each of the single-port memory blocks is configured in single-clock mode.

Thus, each of the single-port memory blocks access up to half of the M4K or M-RAM memory resources such as clock, clock enables, and asynchronous clear signals.

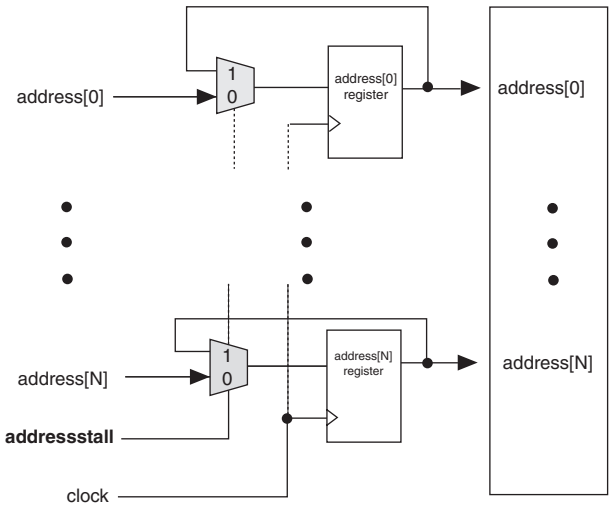
Refer to "Single-Port Mode" on page 8–10 and "Single-Clock Mode" on page 8–28 for more information.

Address Clock Enable Support

Stratix II and Stratix II GX M4K and M-RAM memory blocks support address clock enable, which is used to hold the previous address value for as long as the signal is enabled. When the memory blocks are configured in dual-port mode, each port has its own independent address clock enable.

Figure 8–2 shows an address clock enable block diagram. Placed in the address register, the address signal output by the address register is fed back to the input of the register via a multiplexer. The multiplexer output is selected by the address clock enable (`addressstall`) signal. Address latching is enabled when the `addressstall` signal turns high. The output of the address register is then continuously fed into the input of the register; therefore, the address value can be held until the `addressstall` signal turns low.

Figure 8–2. Stratix II and Stratix II GX Address Clock Enable Block Diagram



Address clock enable is typically used for cache memory applications, which require one port for read and another port for write. The default value for the address clock enable signals is low (disabled). Figures 8–3 and 8–4 show the address clock enable waveform during the read and write cycles, respectively.

Figure 8–3. Stratix II and Stratix II GX Address Clock Enable During Read Cycle Waveform

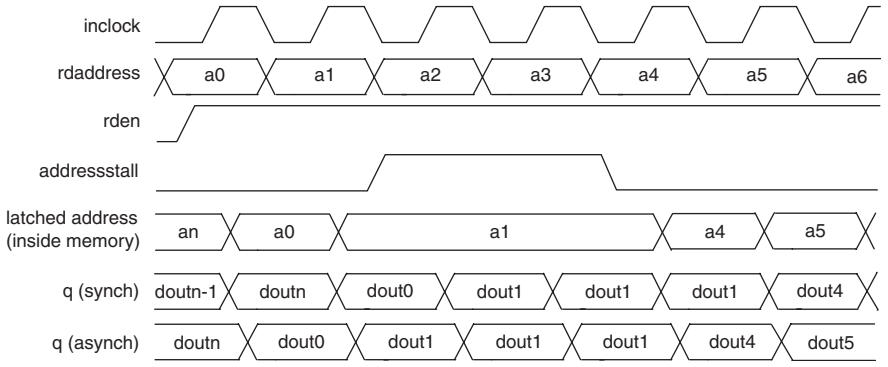
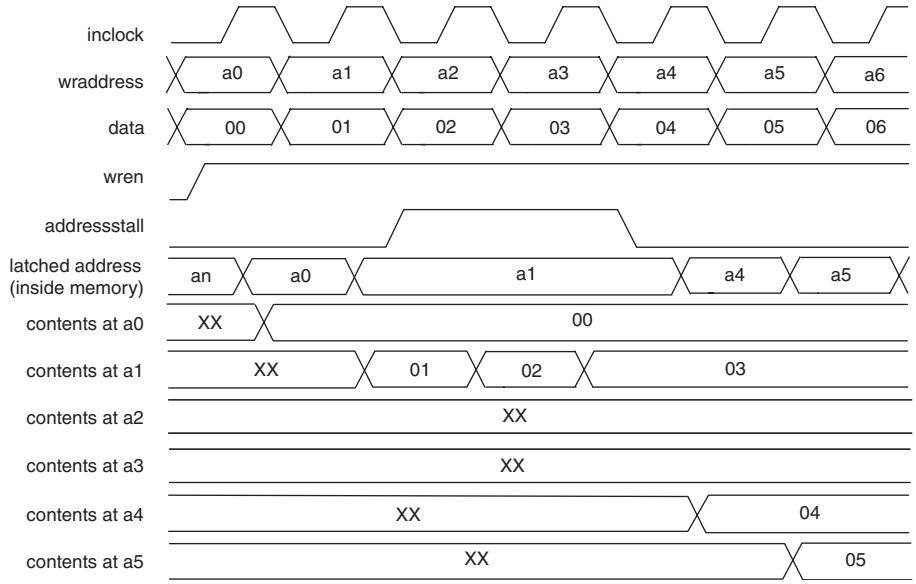


Figure 8–4. Stratix II and Stratix II GX Address Clock Enable During Write Cycle Waveform



Memory Modes

Stratix II and Stratix II GX TriMatrix memory blocks include input registers that synchronize writes, and output registers to pipeline data to improve system performance. All TriMatrix memory blocks are fully synchronous, meaning that all inputs are registered, but outputs can be either registered or unregistered.

TriMatrix memory does not support asynchronous memory (unregistered inputs).

Depending on which TriMatrix memory block you use, the memory has various modes, including:

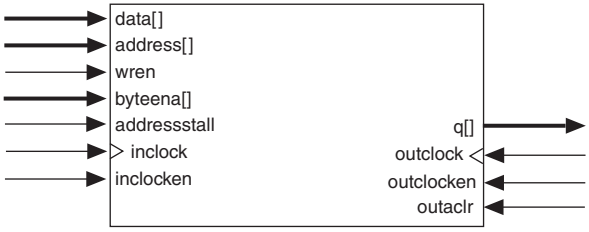
- Single-port
- Simple dual-port
- True dual-port (bidirectional dual-port)
- Shift-register
- ROM
- FIFO

Violating the setup or hold time on the memory block address registers could corrupt memory contents. This applies to both read and write operations.

Single-Port Mode

All TriMatrix memory blocks support the single-port mode that supports non-simultaneous read and write operations. Figure 8–5 shows the single-port memory configuration for TriMatrix memory.

Figure 8–5. Single-Port Memory *Note (1)*



Note to Figure 8–5:
(1) Two single-port memory blocks can be implemented in a single M4K or M-RAM block.

M4K and M-RAM memory blocks can also be halved and used for two independent single-port RAM blocks. The Altera® Quartus® II software automatically uses this single-port memory packing when running low on memory resources. To force two single-port memories into one M4K or M-RAM block, first ensure that each of the two independent RAM blocks is equal to or less than half the size of the M4K or M-RAM block. Secondly, assign both single-port RAMs to the same M4K or M-RAM block.

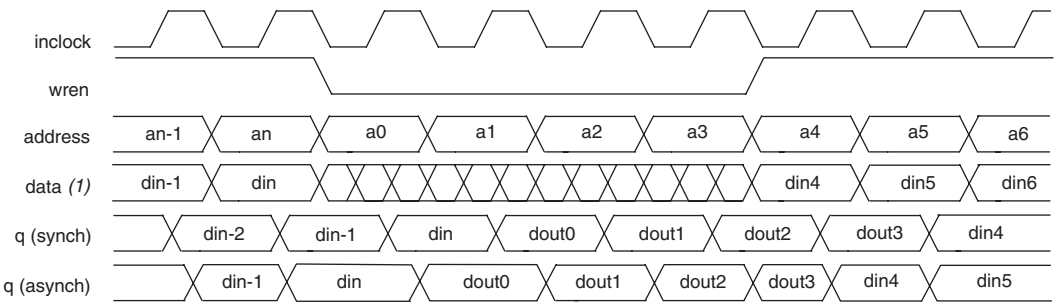
In single-port RAM configuration, the outputs can only be in read-during-write mode, which means that during the write operation, data written to the RAM flows through to the RAM outputs. When the output registers are bypassed, the new data is available on the rising edge of the same clock cycle on which it was written. Refer to "Read-During-Write Operation at the Same Address" on page 8–33 for more information about read-during-write mode. Table 8–8 shows the port width configurations for TriMatrix blocks in single-port mode.

Table 8–8. Stratix II and Stratix II GX Port Width Configurations for M512, M4K, and M-RAM Blocks (Single-Port Mode)

	M512 Blocks	M4K Blocks	M-RAM Blocks
Port Width Configurations	512 × 1 256 × 2 128 × 4 64 × 8 64 × 9 32 × 16 32 × 18	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 16K × 36 8K × 64 8K × 72 4K × 128 4K × 144

Figure 8–6 shows timing waveforms for read and write operations in single-port mode.

Figure 8–6. Stratix II and Stratix II GX Single-Port Timing Waveforms

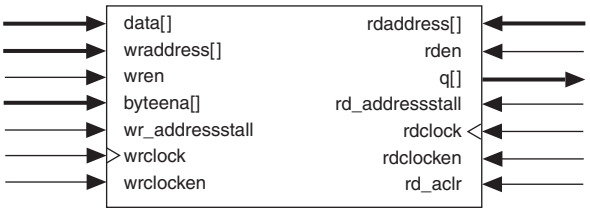


Note to Figure 8–6:
(1) The crosses in the `data` waveform during read mean "don't care."

Simple Dual-Port Mode

All TriMatrix memory blocks support simple dual-port mode which supports a simultaneous read and write operation. Figure 8–7 shows the simple dual-port memory configuration for TriMatrix memory.

Figure 8–7. Stratix II and Stratix II GX Simple Dual-Port Memory *Note (1)*



Note to Figure 8–7:

(1) Simple dual-port RAM supports input/output clock mode in addition to the read/write clock mode shown.

TriMatrix memory supports mixed-width configurations, allowing different read and write port widths. Tables 8–9 through 8–11 show the mixed width configurations for the M512, M4K, and M-RAM blocks, respectively.

Table 8–9. Stratix II and Stratix II GX M512 Block Mixed-Width Configurations (Simple Dual-Port Mode)

Read Port	Write Port						
	512 × 1	256 × 2	128 × 4	64 × 8	32 × 16	64 × 9	32 × 18
512 × 1	✓	✓	✓	✓	✓		
256 × 2	✓	✓	✓	✓	✓		
128 × 4	✓	✓	✓	✓	✓		
64 × 8	✓	✓	✓	✓	✓		
32 × 16	✓	✓	✓	✓	✓		
64 × 9						✓	✓
32 × 18						✓	✓

Table 8–10. Stratix II and Stratix II GX M4K Block Mixed-Width Configurations (Simple Dual-Port Mode)

Read Port	Write Port								
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	128 × 32	512 × 9	256 × 18	128 × 36
4K × 1	✓	✓	✓	✓	✓	✓			
2K × 2	✓	✓	✓	✓	✓	✓			
1K × 4	✓	✓	✓	✓	✓	✓			
512 × 8	✓	✓	✓	✓	✓	✓			
256 × 16	✓	✓	✓	✓	✓	✓			
128 × 32	✓	✓	✓	✓	✓	✓			
512 × 9							✓	✓	✓
256 × 18							✓	✓	✓
128 × 36							✓	✓	✓

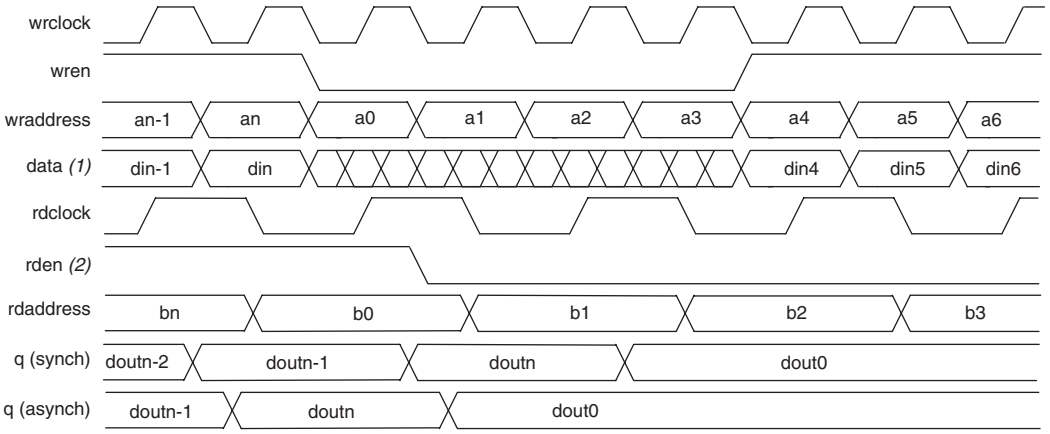
Table 8–11. Stratix II and Stratix II GX M-RAM Block Mixed-Width Configurations (Simple Dual-Port Mode)

Read Port	Write Port				
	64K × 9	32K × 18	18K × 36	8K × 72	4K × 144
64K × 9	✓	✓	✓	✓	
32K × 18	✓	✓	✓	✓	
18K × 36	✓	✓	✓	✓	
8K × 72	✓	✓	✓	✓	
4K × 144					✓

In simple dual-port mode, M512 and M4K blocks have one write enable and one read enable signal. However, M-RAM blocks contain only a write-enable signal, which is held high to perform a write operation. M-RAM blocks are always enabled for read operations. If the read address and the write address select the same address location during a write operation, M-RAM block output is unknown.

TriMatrix memory blocks do not support a clear port on the write enable and read enable registers. When the read enable is deactivated, the current data is retained at the output ports. If the read enable is activated during a write operation with the same address location selected, the simple dual-port RAM output is either unknown or can be set to output the old data stored at the memory address. Refer to "Read-During-Write Operation at the Same Address" on page 8–33 for more information. Figure 8–8 shows timing waveforms for read and write operations in simple dual-port mode.

Figure 8–8. Stratix II and Stratix II GX Simple Dual-Port Timing Waveforms

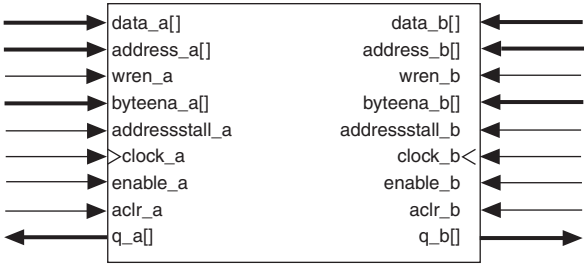


Notes to Figure 8–8:
(1) The crosses in the `data` waveform during read mean "don't care."
(2) The read enable `rden` signal is not available in M-RAM blocks. The M-RAM block in simple dual-port mode always reads out the data stored at the current read address location.

True Dual-Port Mode

Stratix II and Stratix II GX M4K and M-RAM memory blocks support the true dual-port mode. True dual-port mode supports any combination of two-port operations: two reads, two writes, or one read and one write at two different clock frequencies. Figure 8–9 shows Stratix II and Stratix II GX true dual-port memory configuration.

Figure 8–9. Stratix II and Stratix II GX True Dual-Port Memory *Note (1)*



Note to Figure 8–9:
(1) True dual-port memory supports input/output clock mode in addition to the independent clock mode shown.

The widest bit configuration of the M4K and M-RAM blocks in true dual-port mode is as follows:

- 256 × 16-bit (×18-bit with parity) (M4K)
- 8K × 64-bit (×72-bit with parity) (M-RAM)

The 128 × 32-bit (×36-bit with parity) configuration of the M4K block and the 4K × 128-bit (×144-bit with parity) configuration of the M-RAM block are unavailable because the number of output drivers is equivalent to the maximum bit width of the respective memory block. Because true dual-port RAM has outputs on two ports, the maximum width of the true dual-port RAM equals half of the total number of output drivers. Table 8–12 lists the possible M4K block mixed-port width configurations.

Table 8–12. Stratix II and Stratix II GX M4K Block Mixed-Port Width Configurations (True Dual-Port)

Read Port	Write Port						
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	512 × 9	256 × 18
4K × 1	✓	✓	✓	✓	✓		
2K × 2	✓	✓	✓	✓	✓		
1K × 4	✓	✓	✓	✓	✓		
512 × 8	✓	✓	✓	✓	✓		
256 × 16	✓	✓	✓	✓	✓		
512 × 9						✓	✓
256 × 18						✓	✓

Table 8–13 lists the possible M-RAM block mixed-port width configurations.

Table 8–13. Stratix II and Stratix II GX M-RAM Block Mixed-Port Width Configurations (True Dual-Port)

Read Port	Write Port			
	64K × 9	32K × 18	18K × 36	8K × 72
64K × 9	✓	✓	✓	✓
32K × 18	✓	✓	✓	✓
18K × 36	✓	✓	✓	✓
8K × 72	✓	✓	✓	✓

In true dual-port configuration, the RAM outputs can only be configured for read-during-write mode. This means that during write operation, data being written to the A or B port of the RAM flows through to the A or B outputs, respectively. When the output registers are bypassed, the new data is available on the rising edge of the same clock cycle on which it was written. Refer to "Read-During-Write Operation at the Same Address" on page 8–33 for waveforms and information on mixed-port read-during-write mode.

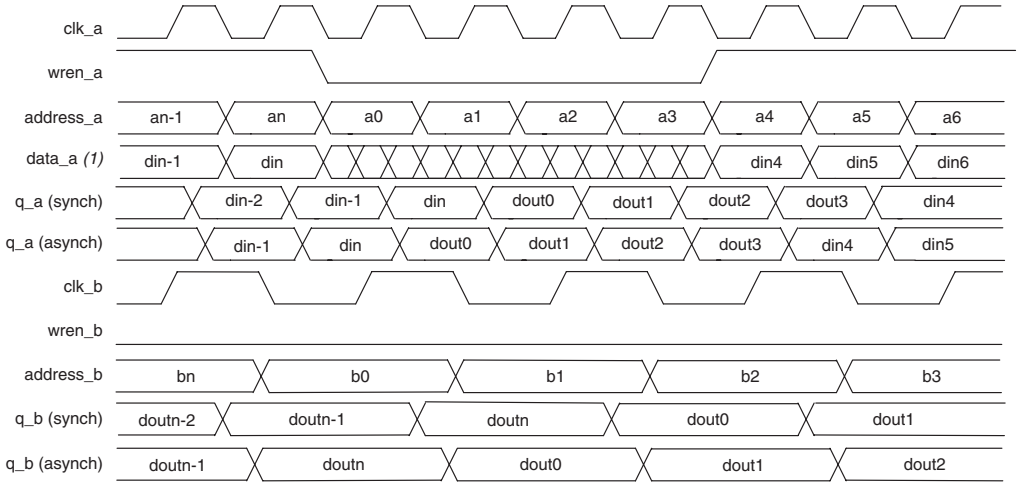
Potential write contentions must be resolved external to the RAM because writing to the same address location at both ports results in unknown data storage at that location. For a valid write operation to the same address of the M-RAM block, the rising edge of the write clock for port A must occur following the maximum write cycle time interval after the rising edge of the write clock for port B. Data is written on the rising edge of the write clock for the M-RAM block.

Because data is written into the M512 and M4K blocks at the falling edge of the write clock, the rising edge of the write clock for port A should occur following half of the maximum write cycle time interval after the falling edge of the write clock for port B. If this timing is not met, the data stored in that particular address will be invalid.

Refer to the *Stratix II Device Family Data Sheet* (volume 1) of the *Stratix II Device Handbook* or the *Stratix II GX Device Family Data Sheet* (volume 1) of the *Stratix II GX Device Handbook* for the maximum synchronous write cycle time.

Figure 8–10 shows true dual-port timing waveforms for the write operation at port A and the read operation at port B.

Figure 8–10. Stratix II and Stratix II GX True Dual-Port Timing Waveforms



Note to Figure 8–10:
(1) The crosses in the `data_a` waveform during write mean "don't care."

Shift-Register Mode

All Stratix II memory blocks support the shift register mode.

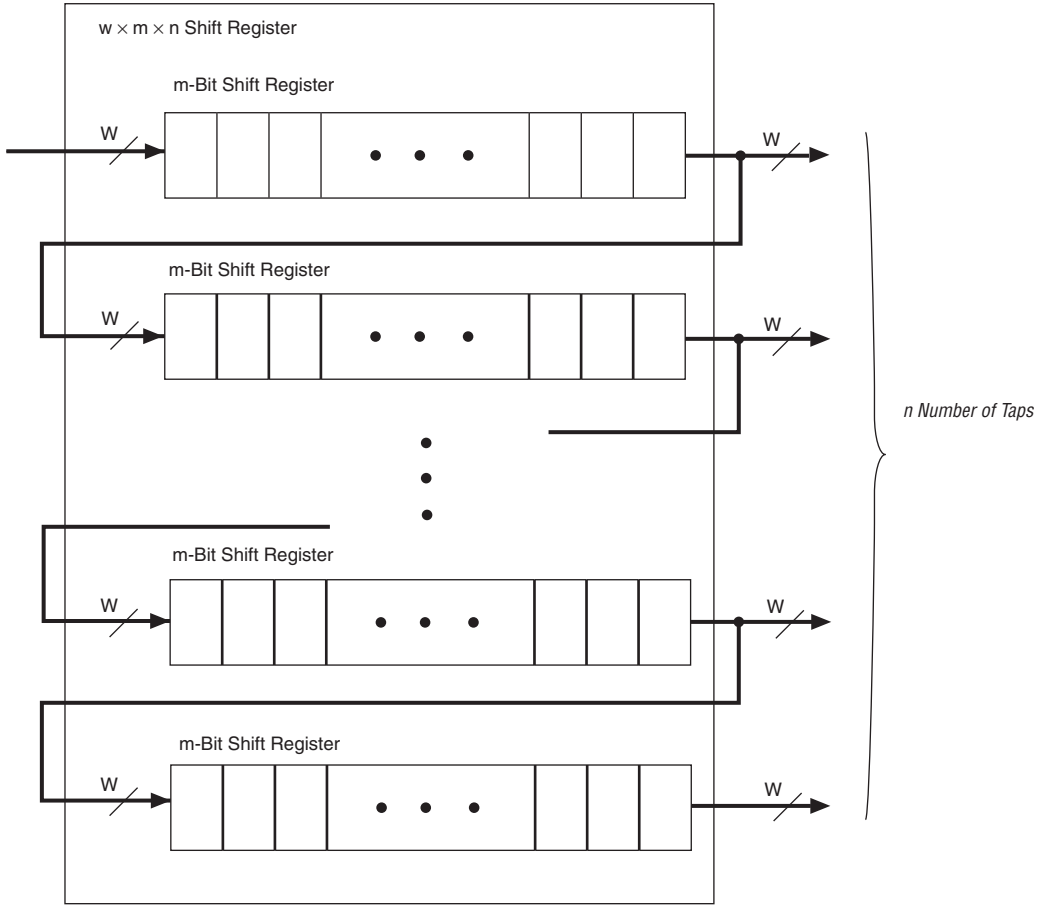
Embedded memory block configurations can implement shift registers for digital signal processing (DSP) applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto-correlation and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flip-flops that quickly exhaust many logic cells for large shift registers. A more efficient alternative is to use embedded memory as a shift-register block, which saves logic cell and routing resources.

The size of a (w × m × n) shift register is determined by the input data width (w), the length of the taps (m), and the number of taps (n), and must be less than or equal to the maximum number of memory bits in the respective block: 576 bits for the M512 block, 4,608 bits for the M4K block, and 589,824 bits for the MRAM block. In addition, the size of w × n must be less than or equal to the maximum width of the respective block: 18

bits for the M512 block, 36 bits for the M4K block, and 144 bits for the MRAM block. If a larger shift register is required, the memory blocks can be cascaded.

In M512 and M4K blocks, data is written into each address location at the falling edge of the clock and read from the address at the rising edge of the clock. The shift-register mode logic automatically controls the positive and negative edge clocking to shift the data in one clock cycle. The MRAM block performs reads and writes on the rising edge. Figure 8–11 shows the TriMatrix memory block in the shift-register mode.

Figure 8–11. Stratix II and Stratix II GX Shift-Register Memory Configuration



ROM Mode

M512 and M4K memory blocks support ROM mode. A memory initialization file (**.mif**) initializes the ROM contents of these blocks. The address lines of the ROM are registered. The outputs can be registered or unregistered. The ROM read operation is identical to the read operation in the single-port RAM configuration.

FIFO Buffers Mode

TriMatrix memory blocks support the FIFO mode. M512 memory blocks are ideal for designs with many shallow FIFO buffers. All memory configurations have synchronous inputs; however, the FIFO buffer outputs are always combinational. Simultaneous read and write from an empty FIFO buffer is not supported.

Refer to the *Single- and Dual-Clock FIFO Megafunctions User Guide* and *FIFO Partitioner Megafunction User Guide* for more information on FIFO buffers.

Clock Modes

Depending on which TriMatrix memory mode is selected, the following clock modes are available:

- Independent
- Input/output
- Read/write
- Single-clock

Table 8–14 shows these clock modes supported by all TriMatrix blocks when configured as respective memory modes.

Table 8–14. Stratix II and Stratix II GX TriMatrix Memory Clock Modes

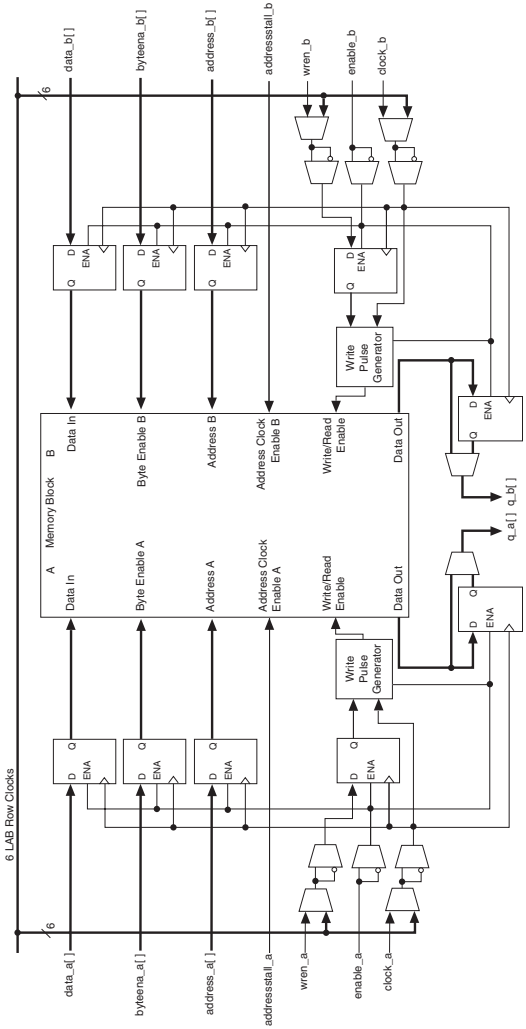
Clocking Modes	True Dual-Port Mode	Simple Dual-Port Mode	Single-Port Mode
Independent	✓		
Input/output	✓	✓	✓
Read/write		✓	
Single clock	✓	✓	✓

Independent Clock Mode

The TriMatrix memory blocks can implement independent clock mode for true dual-port memory. In this mode, a separate clock is available for each port (A and B). Clock A controls all registers on the port A side, while clock B controls all registers on the port B side. Each port also supports independent clock enables for port A and B registers. Asynchronous clear signals for the registers, however, are supported.

Figure 8–12 shows a TriMatrix memory block in independent clock mode.

Figure 8–12. Stratix II and Stratix II GX TriMatrix Memory Block in Independent Clock Mode *Note (1)*



Note to Figure 8–12:

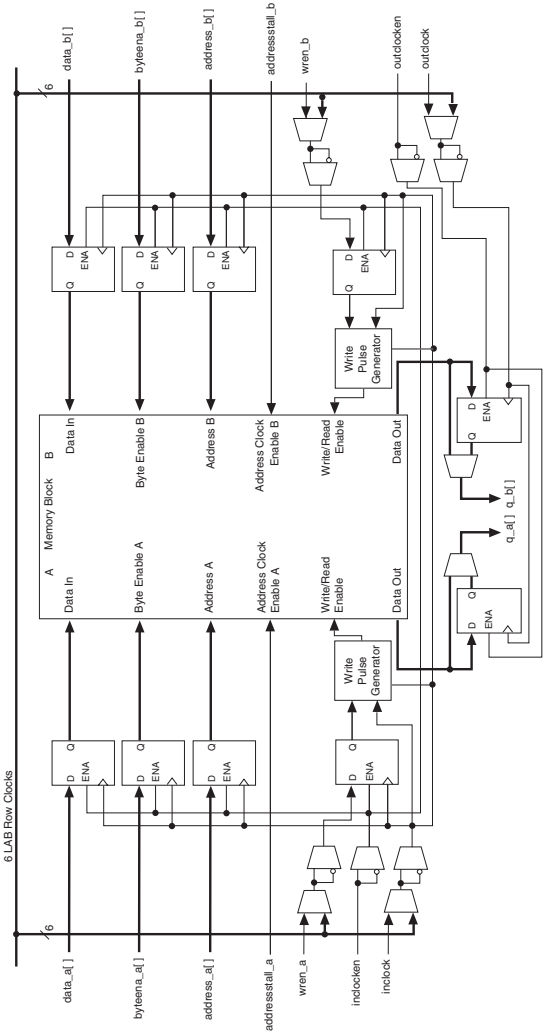
(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

Input/Output Clock Mode

Stratix II and Stratix II GX TriMatrix memory blocks can implement input/output clock mode for true and simple dual-port memory. On each of the two ports, A and B, one clock controls all registers for the following inputs into the memory block: data input, write enable, and address. The other clock controls the blocks' data output registers. Each memory block port also supports independent clock enables for input and output registers. Asynchronous clear signals for the registers, however, are not supported.

Figures 8–13 through 8–15 show the memory block in input/output clock mode for true dual-port, simple dual-port, and single-port modes, respectively.

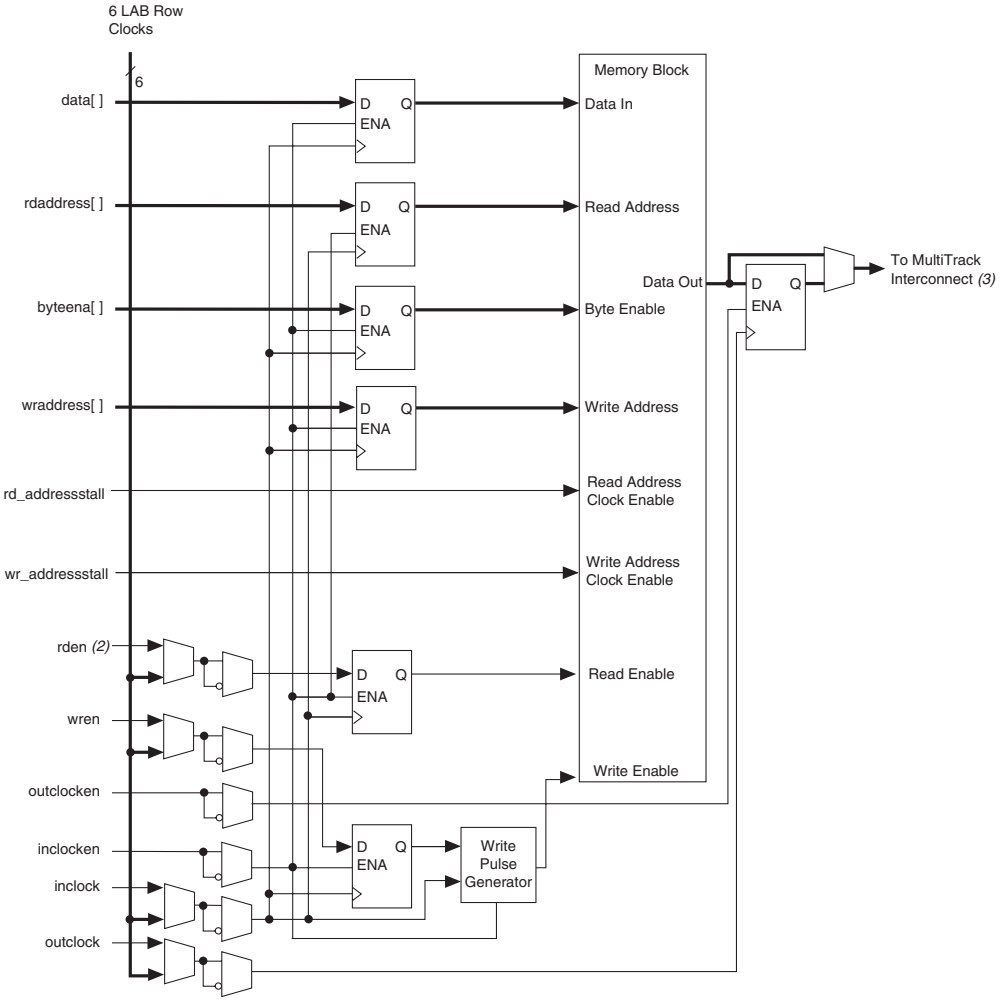
Figure 8–13. Stratix II and Stratix II GX Input/Output Clock Mode in True Dual-Port Mode *Note (1)*



Note to Figure 8–13:

(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

Figure 8–14. Stratix II and Stratix II GX Input/Output Clock Mode in Simple Dual-Port Mode *Note (1)*



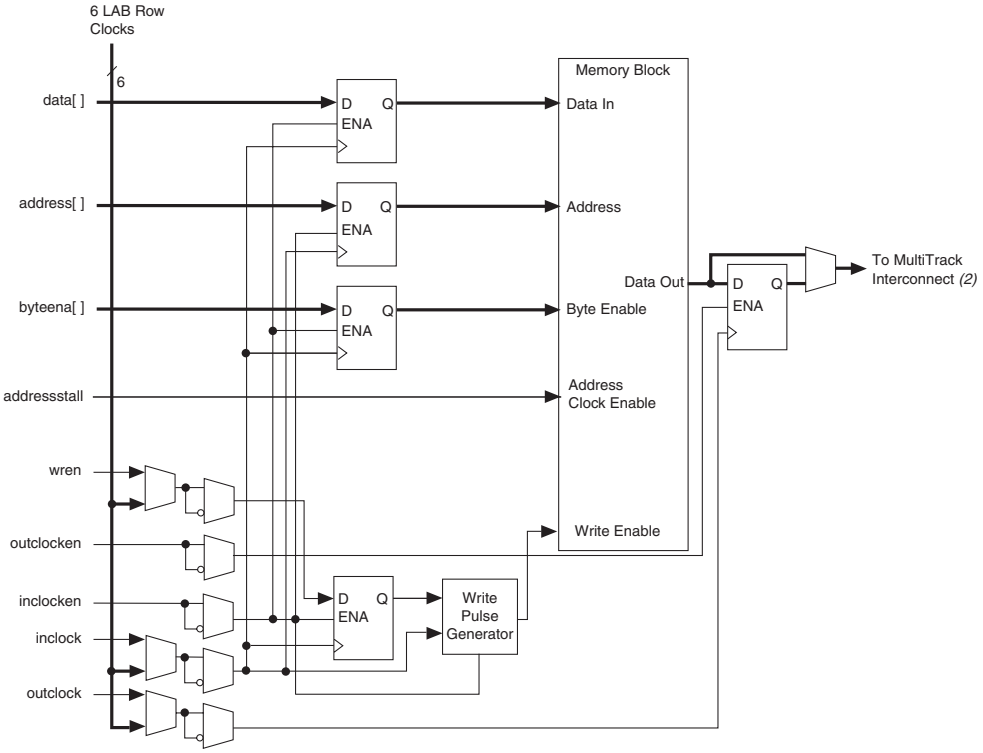
Notes to Figure 8–14:

(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

(2) The read enable `rden` signal is not available in the M-RAM block. An M-RAM block in simple dual-port mode is always reading out the data stored at the current read address location.

(3) Refer to the *Stratix II Device Family Data Sheet* (volume 1) of the *Stratix II Device Handbook* or the *Stratix II GX Device Family Data Sheet* (volume 1) of the *Stratix II GX Device Handbook* for more information on the MultiTrack™ interconnect.

Figure 8–15. Stratix II and Stratix II GX Input/Output Clock Mode in Single-Port Mode *Note (1)*



Notes to Figure 8–15:

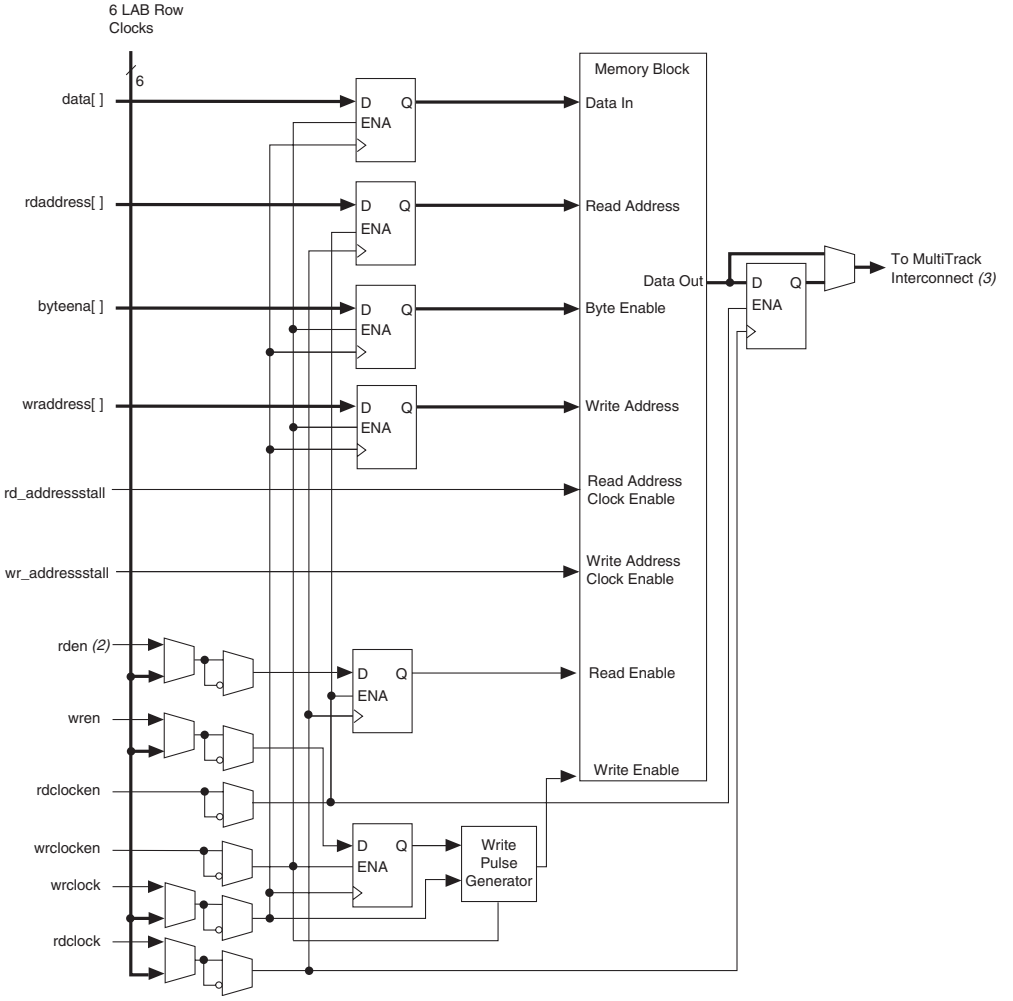
(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

(2) Refer to the *Stratix II Device Family Data Sheet* (volume 1) of the *Stratix II Device Handbook* or the *Stratix II GX Device Family Data Sheet* (volume 1) of the *Stratix II GX Device Handbook* for more information on the MultiTrack interconnect.

Read/Write Clock Mode

Stratix II and Stratix II GX TriMatrix memory blocks can implement read/write clock mode for simple dual-port memory. This mode uses up to two clocks. The write clock controls the blocks' data inputs, write address, and write enable signals. The read clock controls the data output, read address, and read enable signals. The memory blocks support independent clock enables for each clock for the read- and write-side registers. Asynchronous clear signals for the registers, however, are not supported. Figure 8–16 shows a memory block in read/write clock mode.

Figure 8–16. Stratix II and Stratix II GX Read/Write Clock Mode *Note (1)*



Notes to Figure 8–16:

(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

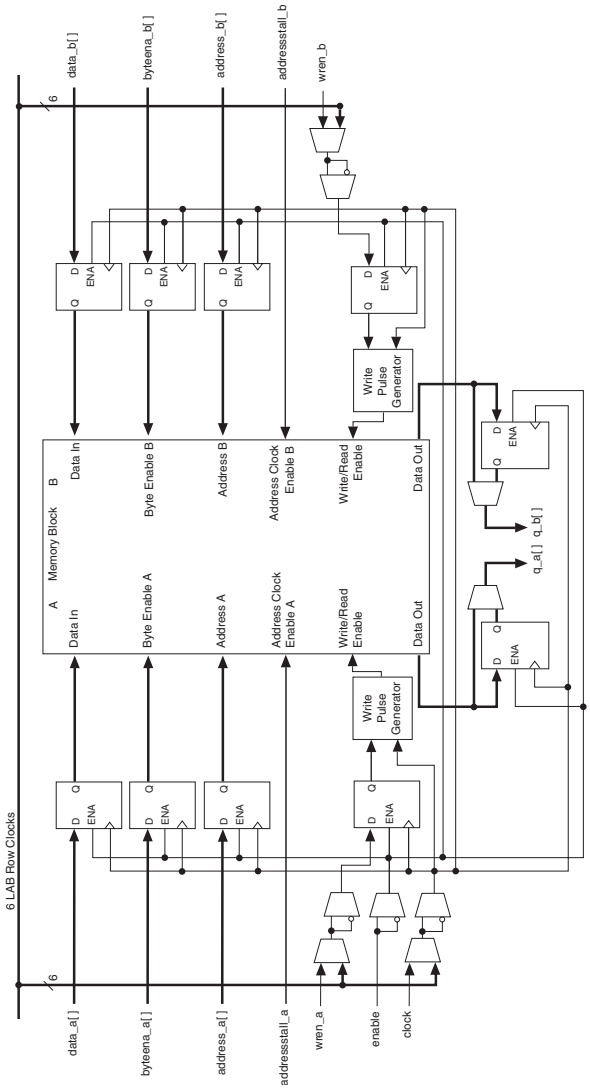
(2) The read enable `rden` signal is not available in the M-RAM block. An M-RAM block in simple dual-port mode is always reading the data stored at the current read address location.

(3) Refer to the *Stratix II Device Family Data Sheet* (volume 1) of the *Stratix II Device Handbook* or the *Stratix II GX Device Family Data Sheet* (volume 1) of the *Stratix II GX Device Handbook* for more information on the MultiTrack interconnect.

Single-Clock Mode

Stratix II and Stratix II GX TriMatrix memory blocks implement single-clock mode for true dual-port, simple dual-port, and single-port memory. In this mode, a single clock, together with clock enable, is used to control all registers of the memory block. Asynchronous clear signals for the registers, however, are not supported. Figures 8–17 through 8–19 show the memory block in single-clock mode for true dual-port, simple dual-port, and single-port modes, respectively.

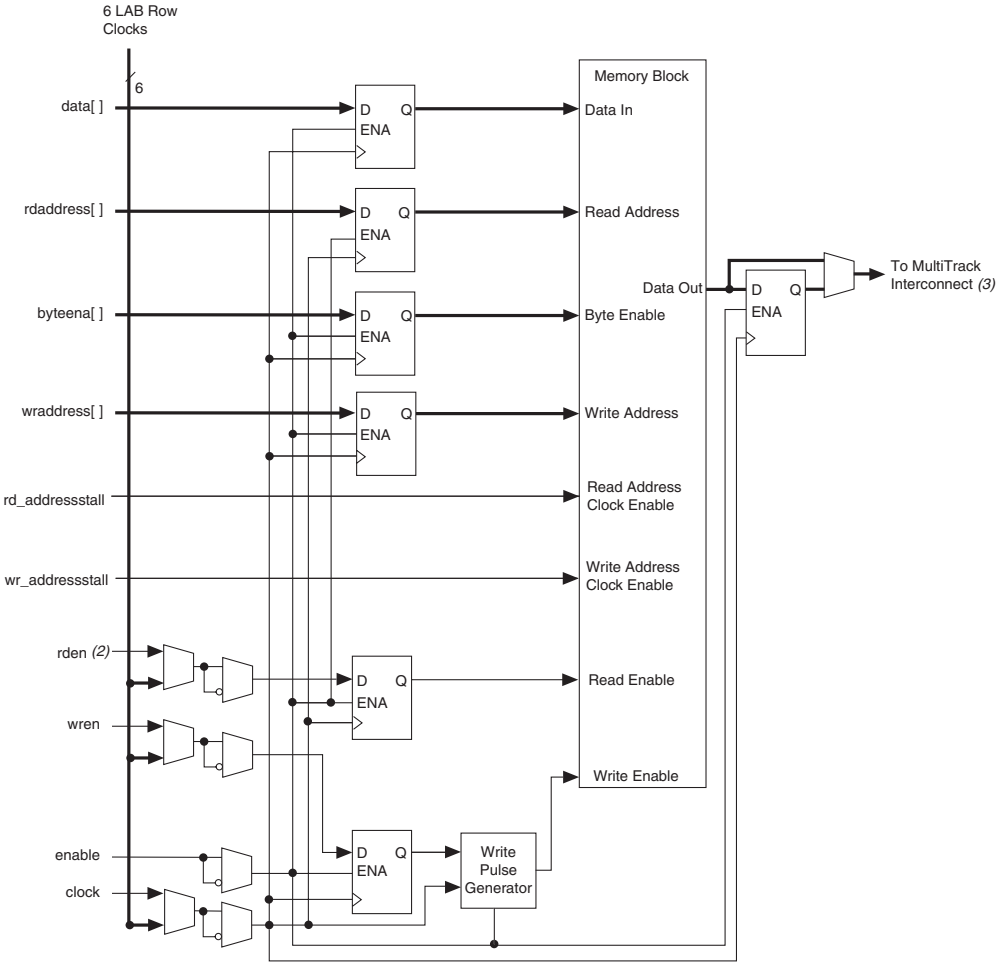
Figure 8–17. Stratix II and Stratix II GX Single-Clock Mode in True Dual-Port Mode *Note (1)*



Note to Figure 8–17:

(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

Figure 8–18. Stratix II andStratix II GX Single-Clock Mode in Simple Dual-Port Mode *Note (1)*



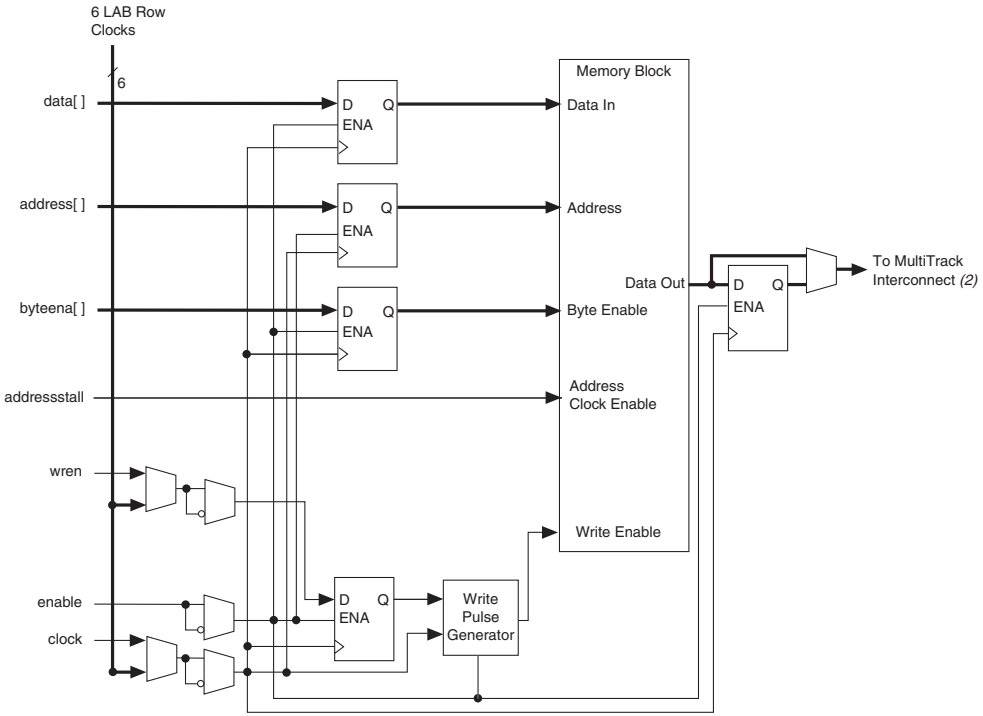
Notes to Figure 8–18:

(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

(2) The read enable `rden` signal is not available in the M-RAM block. An M-RAM block in simple dual-port mode is always reading the data stored at the current read address location.

(3) Refer to the *Stratix II Device Family Data Sheet* (volume 1) of the *Stratix II Device Handbook* or the *Stratix II GX Device Family Data Sheet* (volume 1) of the *Stratix II GX Device Handbook* for more information on the MultiTrack interconnect.

Figure 8–19. Stratix II and Stratix II GX Single-Clock Mode in Single-Port Mode *Note (1)*



Notes to Figure 8–19:

(1) Violating the setup or hold time on the memory block address registers could corrupt the memory contents. This applies to both read and write operations.

(2) Refer to the *Stratix II Device Family Data Sheet* (volume 1) of the *Stratix II Device Handbook* or the *Stratix II GX Device Family Data Sheet* (volume 1) of the *Stratix II GX Device Handbook* for more information on the MultiTrack interconnect.

Designing With TriMatrix Memory

When instantiating TriMatrix memory, it is important to understand the features that set it apart from other memory architectures. The following sections describe the unique attributes and functionality of TriMatrix memory.

Selecting TriMatrix Memory Blocks

The Quartus II software automatically partitions user-defined memory into embedded memory blocks using the most efficient size combinations. The memory can also be manually assigned to a specific block size or a mixture of block sizes. Table 8–1 on page 8–2 is a guide for selecting a TriMatrix memory block size based on supported features.

Refer to *AN 207: TriMatrix Memory Selection Using the Quartus II Software* for more information on selecting the appropriate memory block.

Synchronous and Pseudo-Asynchronous Modes

The TriMatrix memory architecture implements synchronous RAM by registering the input and output signals to the RAM block. The inputs to all TriMatrix memory blocks are registered providing synchronous write cycles, while the output registers can be bypassed. In a synchronous operation, RAM generates its own self-timed strobe write enable signal derived from the global or regional clock. In contrast, a circuit using asynchronous RAM must generate the RAM write enable signal while ensuring that its data and address signals meet setup and hold time specifications relative to the write enable signal. During a synchronous operation, the RAM is used in pipelined mode (inputs and outputs registered) or flow-through mode (only inputs registered). However, in an asynchronous memory, neither the input nor the output is registered.

While Stratix II and Stratix II GX devices do not support asynchronous memory, they do support a pseudo-asynchronous read where the output data is available during the clock cycle when the read address is driven into it. Pseudo-asynchronous reading is possible in the simple and true dual-port modes of the M512 and M4K blocks by clocking the read enable and read address registers on the negative clock edge and bypassing the output registers.

Refer to *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs* for more information.

Power-up Conditions and Memory Initialization

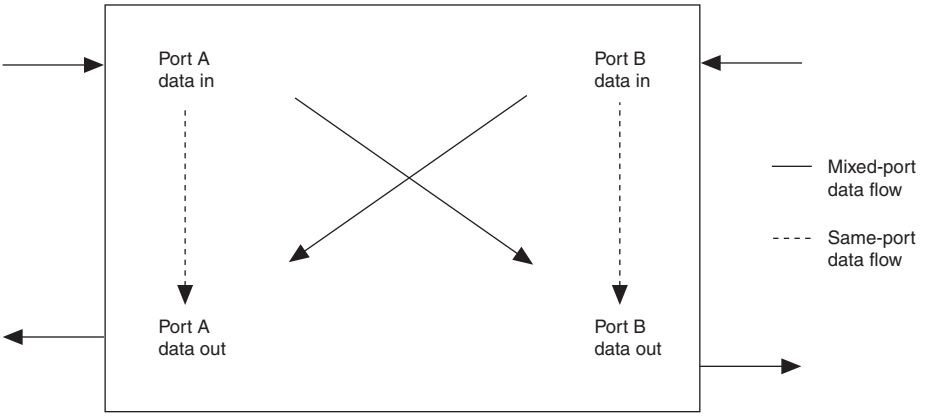
Upon power up, TriMatrix memory is in an idle state. The M512 and M4K block outputs always power-up to zero, regardless of whether the output registers are used or bypassed. Even if an MIF is used to pre-load the contents of the RAM block, the outputs will still power-up as cleared. For example, if address 0 is pre-initialized to FF, the M512 and M4K blocks power up with the output at 00.

M-RAM blocks do not support MIFs; therefore, they cannot be pre-loaded with data upon power up. M-RAM blocks asynchronous outputs and memory controls always power up to an unknown state. If M-RAM block outputs are registered, the registers power up as cleared. When a read is performed immediately after power up, the output from the read operation will be undefined since the M-RAM contents are not initialized. The read operation will continue to be undefined for a given address until a write operation is performed for that address.

Read-During-Write Operation at the Same Address

The "Same-Port Read-During-Write Mode" on page 8–33 and "Mixed-Port Read-During-Write Mode" on page 8–34 sections describe the functionality of the various RAM configurations when reading from an address during a write operation at that same address. There are two read-during-write data flows: same-port and mixed-port. Figure 8–20 shows the difference between these flows.

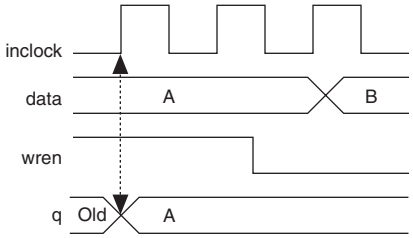
Figure 8–20. Stratix II and Stratix II GX Read-During-Write Data Flow



Same-Port Read-During-Write Mode

For read-during-write operation of a single-port RAM or the same port of a true dual-port RAM, the new data is available on the rising edge of the same clock cycle on which it was written. This behavior is valid on all memory block sizes. Figure 8–21 shows a sample functional waveform. When using byte enables in true dual-port RAM mode, the outputs for the masked bytes on the same port are unknown (refer to Figure 8–1 on page 8–7). The non-masked bytes are read out as shown in Figure 8–21.

Figure 8–21. Stratix II and Stratix II GX Same-Port Read-During-Write Functionality *Note (1)*



Note to Figure 8–21:
(1) Outputs are not registered.

Mixed-Port Read-During-Write Mode

This mode is used when a RAM in simple or true dual-port mode has one port reading and the other port writing to the same address location with the same clock.

The `READ_DURING_WRITE_MODE_MIXED_PORTS` parameter for M512 and M4K memory blocks determines whether to output the old data at the address or a "don't care" value. Setting this parameter to `OLD_DATA` outputs the old data at that address. Setting this parameter to `DONT_CARE` outputs a "don't care" or unknown value. Figures 8–22 and 8–23 show sample functional waveforms where both ports have the same address. These figures assume that the outputs are not registered.

The `DONT_CARE` setting allows memory implementation in any TriMatrix memory block, whereas the `OLD_DATA` setting restricts memory implementation to only M512 or M4K memory blocks. Selecting `DONT_CARE` gives the compiler more flexibility when placing memory functions into TriMatrix memory.

The RAM outputs are unknown for a mixed-port read-during-write operation of the same address location of an M-RAM block, as shown in Figure 8–23.

Figure 8–22. Stratix II and Stratix II GX Mixed-Port Read-During-Write: OLD_DATA

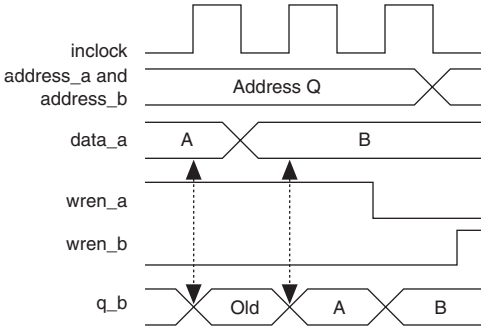
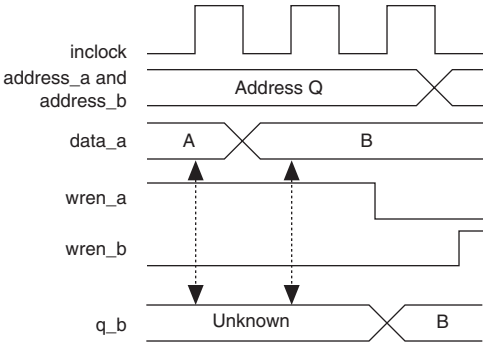


Figure 8–23. Stratix II and Stratix II GX Mixed-Port Read-During-Write: DONT_CARE



Mixed-port read-during-write is not supported when two different clocks are used in a dual-port RAM. The output value is unknown during a mixed-port read-during-write operation.

Conclusion

The TriMatrix memory structure of Stratix II and Stratix II GX devices provides an enhanced RAM architecture with high memory bandwidth. It addresses the needs of different memory applications in FPGA designs with features such as different memory block sizes and modes, byte enables, parity bit storage, address clock enables, mixed clock mode, shift register mode, mixed-port width support, and true dual-port mode.

Referenced Documents

This chapter references the following documents:

- *AN 207: TriMatrix Memory Selection Using the Quartus II Software*
- *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs*
- *FIFO Partitioner Megafunction User Guide*
- *Single- and Dual-Clock FIFO Megafunctions User Guide*
- *Stratix II Device Family Data Sheet* (volume 1) of the *Stratix II Device Handbook*
- *Stratix II GX Device Family Data Sheet* (volume 1) of the *Stratix II GX Device Handbook*
- *Using Parity to Detect Memory Errors* white paper

Document Revision History

Table 8–15 shows the revision history for this chapter.

Table 8–15. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007, v4.5	Added "Referenced Documents" section.	—
	Minor text edits.	—
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 7. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
May 2007, v4.4	Added note to "Byte Enable Functional Waveform" section.	—
	Updated "Byte Enable Support" section.	—
February 2007 v4.3	Added the "Document Revision History" section to this chapter.	—
April 2006, v4.2	Chapter updated as part of the *Stratix II Device Handbook* update.	—
No change	Formerly chapter 6. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
December 2005, v4.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—
October 2005 v4.0	Added chapter to the *Stratix II GX Device Handbook*.	—

9. External Memory Interfaces in Stratix II and Stratix II GX Devices

SII52003-4.5

Introduction

Stratix® II and Stratix II GX devices support a broad range of external memory interfaces such as double data rate (DDR) SDRAM, DDR2 SDRAM, RLDRAM II, QDRII SRAM, and single data rate (SDR) SDRAM. Its dedicated phase-shift circuitry allows the Stratix II or Stratix II GX device to interface with an external memory at twice the system clock speed (up to 300 MHz/600 megabits per second (Mbps) with RLDRAM II). In addition to external memory interfaces, you can also use the dedicated phase-shift circuitry for other applications that require a shifted input signal.

Typical I/O architectures transmit a single data word on each positive clock edge and are limited to the associated clock speed. To achieve a 400-Mbps transfer rate, a SDR system requires a 400-MHz clock. Many new applications have introduced a DDR I/O architecture as an alternative to SDR architectures. While SDR architectures capture data on one edge of a clock, the DDR architectures captures data on both the rising and falling edges of the clock, doubling the throughput for a given clock frequency and accelerating performance. For example, a 200-MHz clock can capture a 400-Mbps data stream, enhancing system performance and simplifying board design.

Most new memory architectures use a DDR I/O interface. Although Stratix II and Stratix II GX devices also support the mature and well established SDR external memory, this chapter focuses on DDR memory standards. These DDR memory standards cover a broad range of applications for embedded processor systems, image processing, storage, communications, and networking.

Stratix II devices offer external memory support in every I/O bank. The side I/O banks support the PLL-based interfaces running at up to 200 MHz, while the top and bottom I/O banks support PLL- and DLL-based interfaces. Figure 9–1 shows Stratix II device memory support.

Figure 9–1. External Memory Support

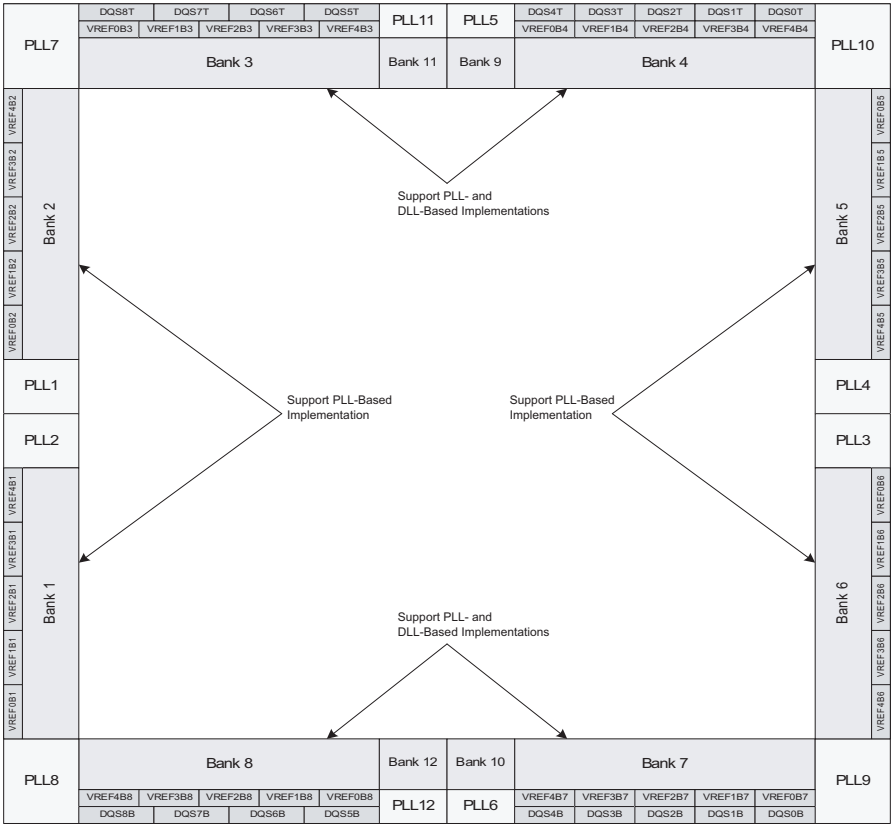


Table 9–1 summarizes the maximum clock rate Stratix II and Stratix II GX devices can support with external memory devices.

Table 9–1. Stratix II and Stratix II GX Maximum Clock Rate Support for External Memory Interfaces Notes (1), (2)

Memory Standards	–3 Speed Grade (MHz)		–4 Speed Grade (MHz)		–5 Speed Grade (MHz)	
	DLL-Based	PLL-Based	DLL-Based	PLL-Based	DLL-Based	PLL-Based
DDR2 SDRAM (3), (5)	333	200	267	167	233	167
DDR SDRAM (3)	200	150	200	133	200	100
RLDRAM II	300	200	250 (4)	175	200	175
QDRII SRAM	300	200	250	167	250	167
QDRII+ SRAM	300	(6)	250	(6)	250	(6)

Notes to Table 9–1:

(1) Memory interface timing specifications are dependent on the memory, board, physical interface, and core logic. Refer to each memory interface application note for more details on how each specification was generated.

(2) The respective Altera MegaCore function and the EP2S60F1020C3 timing information featured in the Quartus® II software version 6.0 was used to define these clock rates.

(3) This applies for interfaces with both modules and components.

(4) You must underclock a 300-MHz RLDRAM II device to achieve this clock rate.

(5) To achieve speeds greater than 267 MHz (533 Mbps) up to 333 MHz (667 Mbps), you must use the Altera DDR2 SDRAM Controller MegaCore function that features a new dynamic auto-calibration circuit in the data path for resynchronization. For more information, see the Altera web site at www.altera.com. For interfaces running at 267 MHz or below, continue to use the static resynchronization data path currently supported by the released version of the MegaCore function.

(6) The lowest frequency at which a QDRII+ SRAM device can operate is 238 MHz. Therefore, the PLL-based implementation does not support the QDRII+ SRAM interface.

This chapter describes the hardware features in Stratix II and Stratix II GX devices that facilitate the high-speed memory interfacing for each DDR memory standard. This chapter focuses primarily on the DLL-based implementation. The PLL-based implementation is described in application notes. It then lists the Stratix II and Stratix II GX feature enhancements from Stratix devices and briefly explains how each memory standard uses the Stratix II and Stratix II GX features.

You can use this document with the following documents:

- AN 325: Interfacing RLDRAM II with Stratix II & Stratix GX Devices
- AN 326: Interfacing QDRII & QDRII+ SRAM with Stratix II, Stratix, & Stratix GX Devices
- AN 327: Interfacing DDR SDRAM with Stratix II Devices
- AN 328: Interfacing DDR2 SDRAM with Stratix II Devices

External Memory Standards

The following sections briefly describe the external memory standards supported by Stratix II and Stratix II GX devices. Altera offers a complete solution for these memories, including clear-text data path, memory controller, and timing analysis.

DDR and DDR2 SDRAM

DDR SDRAM is a memory architecture that transmits and receives data at twice the clock speed. These devices transfer data on both the rising and falling edge of the clock signal. DDR2 SDRAM is a second generation memory based on the DDR SDRAM architecture and transfers data to Stratix II and Stratix II GX devices at up to 333 MHz/667 Mbps. Stratix II and Stratix II GX devices can support DDR SDRAM at up to 200 MHz/400 Mbps. For PLL-based implementations, Stratix II and Stratix II GX devices support DDR and DDR2 SDRAM up to 150 MHz and 200 MHz, respectively.

Interface Pins

DDR and DDR2 SDRAM devices use interface pins such as data (DQ), data strobe (DQS), clock, command, and address pins. Data is sent and captured at twice the system clock rate by transferring data on the clock's positive and negative edge. The commands and addresses still only use one active (positive) edge of a clock. DDR and DDR2 SDRAM use single-ended data strobes (DQS). DDR2 SDRAM can also use optional differential data strobes (DQS and DQS#). However, Stratix II and Stratix II GX devices do not use the optional differential data strobes for DDR2 SDRAM interfaces since DQS and DQSn pins in Stratix II and Stratix II GX devices are not differential. You can leave the DDR SDRAM memory DQS# pin unconnected. Only the shifted DQS signal from the DQS logic block is used to capture data.

DDR and DDR2 SDRAM ×16 devices use two DQS pins, and each DQS pin is associated with eight DQ pins. However, this is not the same as the ×16/×18 mode in Stratix II and Stratix II GX devices (see "Data and Data Strobe Pins" on page 9–14). To support a ×16 DDR SDRAM device, you need to configure Stratix II and Stratix II GX devices to use two sets of DQ pins in ×8/×9 mode. Similarly if your ×32 memory device uses four DQS pins where each DQS pin is associated with eight DQ pins, you need to configure Stratix II and Stratix II GX devices to use four sets of DQS/DQ groups in ×8/×9 mode.

Connect the memory device's DQ and DQS pins to Stratix II and Stratix II GX DQ and DQS pins, respectively, as listed in Stratix II and Stratix II GX pin tables. DDR and DDR2 SDRAM also uses active-high data mask, DM, pins for writes. You can connect the memory's DM pins

to any of Stratix II and Stratix II GX I/O pins in the same bank as the DQ pins of the FPGA. There is one DM pin per DQS/DQ group in a DDR or DDR2 SDRAM device.

You can also use I/O pins in banks 1, 2, 5, or 6 to interface with DDR and DDR2 SDRAM devices. These banks do not have dedicated circuitry, though, and can only support DDR SDRAM at speeds up to 150 MHz and DDR2 SDRAM at speeds up to 200 MHz. DDR2 SDRAM interfaces using these banks are supported using the SSTL-18 Class I I/O standard.

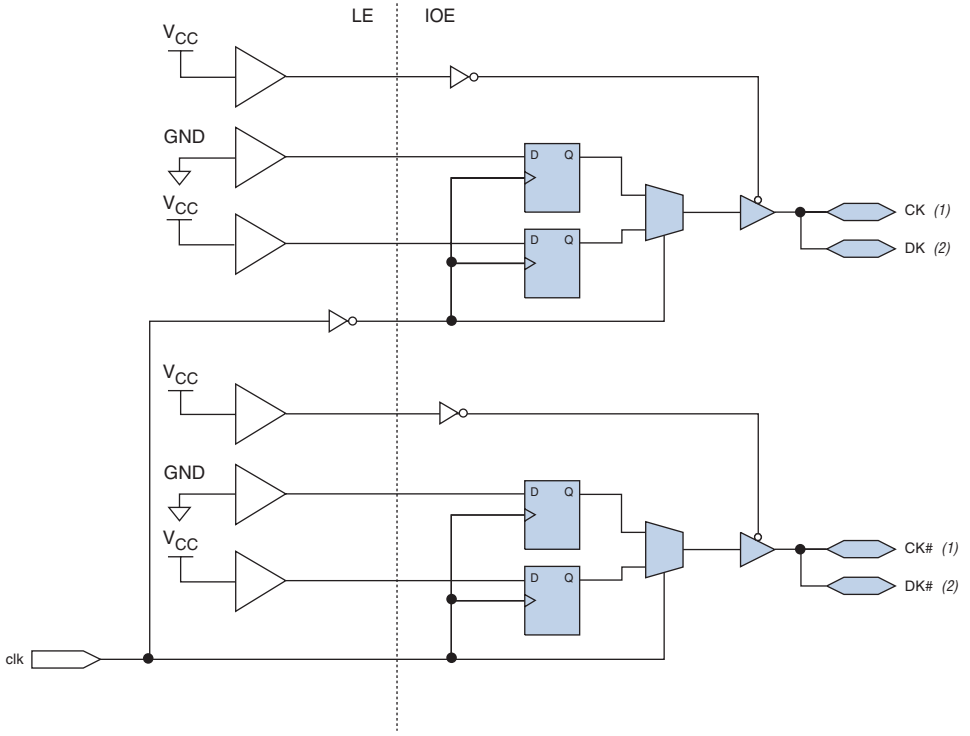
For more information, see AN 327: Interfacing DDR SDRAM with Stratix II Devices and AN 328: Interfacing DDR2 SDRAM with Stratix II Devices.

If the DDR or DDR2 SDRAM device supports error correction coding (ECC), the design will use an extra DQS/DQ group for the ECC pins.

You can use any of the user I/O pins for commands and addresses to the DDR and DDR2 SDRAM. You may need to generate these signals from the system clock's negative edge.

The clocks to the SDRAM device are called CK and CK# pins. Use any of the user I/O pins via the DDR registers to generate the CK and CK# signals to meet the DDR SDRAM or DDR2 SDRAM device's t_{DQSS} requirement. The memory device's t_{DQSS} specification requires that the write DQS signal's positive edge must be within 25% of the positive edge of the DDR SDRAM or DDR2 SDRAM clock input. Using regular I/O pins for CK and CK# also ensures that any PVT variations on the DQS signals are tracked the same way by these CK and CK# pins. Figure 9–2 shows a diagram that illustrates how to generate these clocks.

Figure 9–2. Clock Generation for External Memory Interfaces in Stratix II and Stratix II GX Devices



Notes to Figure 9–2:

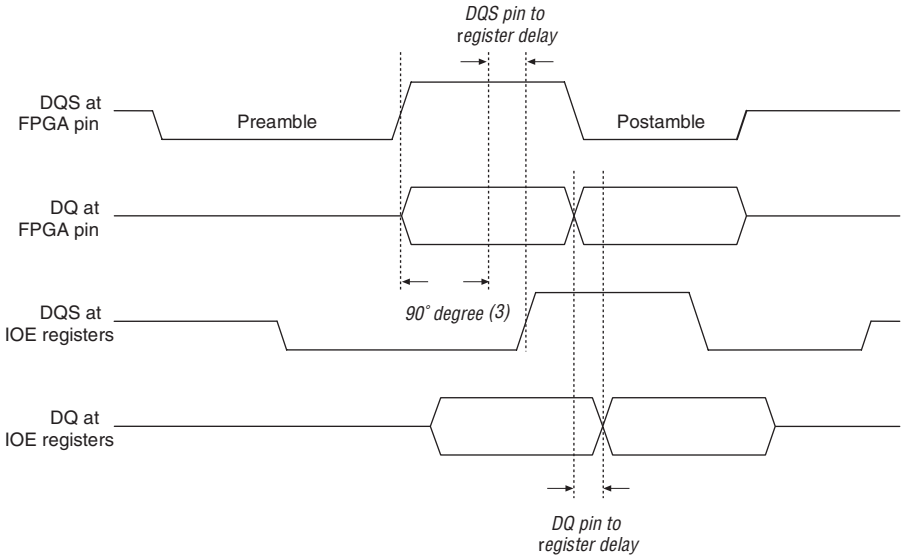
(1) CK and CK# are the clocks to the memory devices.

(2) DK and DK# are for RLDRAM II interfaces. You can generate DK# and DK from separate pins if the difference of the Quartus II software's reported clock-to-out time for these pins meets the RLDRAM II device's t_{CKDK} specification.

Read and Write Operations

When reading from the memory, DDR and DDR2 SDRAM devices send the data edge-aligned with respect to the data strobe. To properly read the data in, the data strobe needs to be center-aligned with respect to the data inside the FPGA. Stratix II and Stratix II GX devices feature dedicated circuitry to shift this data strobe to the middle of the data window. Figure 9–3 shows an example of how the memory sends out the data and data strobe for a burst-of-two operation.

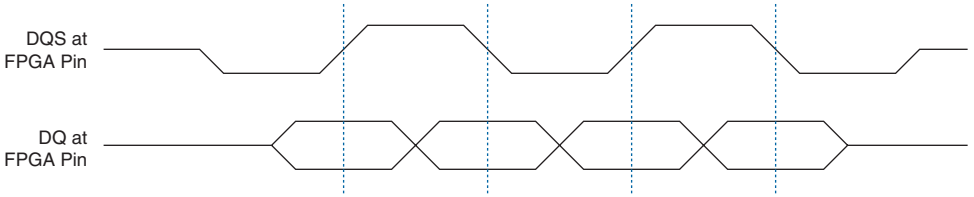
Figure 9–3. Example of a 90° Shift on the DQS Signal *Notes (1), (2)*



Notes to Figure 9–3:
(1) RLDRAM II and QDRII SRAM memory interfaces do not have preamble and postamble specifications.
(2) DDR2 SDRAM does not support a burst length of two.
(3) The phase shift required for your system should be based on your timing analysis and may not be 90°.

During write operations to a DDR or DDR2 SDRAM device, the FPGA needs to send the data to the memory center-aligned with respect to the data strobe. Stratix II and Stratix II GX devices use a PLL to center-align the data by generating a 0° phase-shifted system clock for the write data strobes and a –90° phase-shifted write clock for the write data pins for DDR and DDR2 SDRAM. Figure 9–4 shows an example of the relationship between the data and data strobe during a burst-of-four write.

Figure 9–4. DQ and DQS Relationship During a DDR and DDR2 SDRAM Write *Notes (1), (2)*



Notes to Figure 9–4:

(1) This example shows a write for a burst length of four. DDR SDRAM also supports burst lengths of two.

(2) The write clock signals never go to hi-Z state on RLDRAM II and QDRII SRAM memory interfaces because they use free-running clocks. However, the general timing relationship between data and the read clock shown in this figure still applies.

For more information on DDR SDRAM and DDR2 SDRAM specifications, refer to JEDEC standard publications JESD79C and JESD79-2, respectively, from www.jedec.org, or see AN 327: Interfacing DDR SDRAM with Stratix II Devices and AN 327: Interfacing DDR SDRAM with Stratix II Devices.

RLDRAM II

RLDRAM II provides fast random access as well as high bandwidth and high density, making this memory technology ideal for high-speed network and communication data storage applications. The fast random access speeds in RLDRAM II devices make them a viable alternative to SRAM devices at a lower cost. Additionally, RLDRAM II devices have minimal latency to support designs that require fast response times.

Interface Pins

RLDRAM II devices use interface pins such as data, clock, command, and address pins. There are two types of RLDRAM II memory: common I/O (CIO) and separate I/O (SIO). The data pins in a RLDRAM II CIO device are bidirectional while the data pins in a RLDRAM II SIO device are unidirectional. Instead of bidirectional data strobes, RLDRAM II uses differential free-running read and write clocks to accompany the data. As in DDR or DDR2 SDRAM, data is sent and captured at twice the system clock rate by transferring data on the clock's positive and negative edge. The commands and addresses still only use one active (positive) edge of a clock.

If the data pins are bidirectional, as in RLDRAM II CIO devices, connect them to Stratix II and Stratix II GX DQ pins. If the data pins are unidirectional, as in RLDRAM II SIO devices, connect the RLDRAM II device Q ports to the Stratix II and Stratix II GX device DQ pins and

connect the D ports to any user I/O pins in I/O banks 3, 4, 7, or 8 for optimal performance. RLDRAM II also uses active-high data mask, DM, pins for writes. You can connect DM pins to any of the I/O pins in the same bank as the DQ pins of the FPGA when interfacing with RLDRAM II CIO devices to any of the I/O pins in the same bank as the D pins when interfacing with RLDRAM II SIO devices. There is one DM pin per RLDRAM II device. You can also use I/O pins in banks 1, 2, 5, or 6 to interface with RLDRAM II devices. However, these banks do not have dedicated circuitry and can only support RLDRAM II devices at speeds up to 200 MHz. RLDRAM II interfaces using these banks are supported using the 1.8-V HSTL Class I I/O support.

Connect the RLDRAM II device's read clock pins (QK) to Stratix II or Stratix II GX DQS pins. Because of software requirements, you must configure the DQS signals as bidirectional pins. However, since QK pins are output-only pins from the memory, RLDRAM II memory interfacing in Stratix II and Stratix II GX devices requires that you ground the DQS pin output enables. Stratix II and Stratix II GX devices use the shifted QK signal from the DQS logic block to capture data. You can leave the QK# signal of the RLDRAM II device unconnected, as DQS and DQSn in Stratix II and Stratix II GX devices are not differential pins.

RLDRAM II devices also have input clocks (CK and CK#) and write clocks (DK and DK#).

You can use any of the user I/O pins for commands and addresses. RLDRAM II also offers QVLD pins to indicate the read data availability. Connect the QVLD pins to the Stratix II or Stratix II GX DQVLD pins, listed in the pin table.

Because the Quartus II software treats the DQVLD pins like DQ pins, you should ensure that the DQVLD pin is assigned to the pin table's recommended pin.

Read and Write Operations

When reading from the RLDRAM II device, data is sent edge-aligned with the read clock QK and QK#. When writing to the RLDRAM II device, data must be center-aligned with the write clock (DK and DK#). The RLDRAM II interface uses the same scheme as in DDR or DDR2 SDRAM interfaces, where the dedicated circuitry is used during reads to center-align the data and the read clock inside the FPGA and the PLL center-aligns the data and write clock outputs. The data and clock relationship for reads and writes in RLDRAM II is similar to those in DDR and DDR2 SDRAM as shown in Figures 9–3 and 9–4.

For details on RLDRAM II, see AN 325: Interfacing RLDRAM II with Stratix II & Stratix GX Devices.

QDRII SRAM

QDRII SRAM is the second generation of QDR SRAM devices. Both devices can transfer four words per clock cycle, fulfilling the requirements facing next-generation communications system designers. QDRII SRAM devices provide concurrent reads and writes, zero latency, and increased data throughput, allowing simultaneous access to the same address location. QDRII SRAM is available in burst-of-2 and burst-of-4 devices. Burst-of-2 devices support two-word data transfer on all read and write transactions, and burst-of-4 devices support four-word data transfer

Interface Pins

QDRII SRAM uses two separate, unidirectional data ports for read and write operations, enabling QDR data transfer. QDRII SRAM uses shared address lines for reads and writes. QDRII SRAM burst-of-two devices sample the read address on the rising edge of the clock and sample the write address on the falling edge of the clock while QDRII SRAM burst-of-four devices sample both read and write addresses on the clock's rising edge. Connect the memory device's Q ports (read data) to the Stratix II or Stratix II GX DQ pins. You can use any of the Stratix II or Stratix II GX device user I/O pins in I/O banks 3, 4, 7, or 8 for the D ports (write data), commands, and addresses. The control signals are sampled on the rising edge of the clock. You can also use I/O pins in banks 1, 2, 5, or 6 to interface with QDRII SRAM devices. However, these banks do not have dedicated circuitry and can only support QDRII SRAM devices at speeds up to 200 MHz. QDRII SRAM interfaces using these banks are supported using the 1.8-V HSTL Class I I/O support.

QDRII SRAM uses the following clock signals:

- Input clocks K and K#
- Output clocks C and C#
- Echo clocks CQ and CQ#

Clocks C#, K#, and CQ# are logical complements of clocks C, K, and CQ, respectively. Clocks C, C#, K, and K# are inputs to the QDRII SRAM while clocks CQ and CQ# are outputs from the QDRII SRAM. Stratix II and Stratix II GX devices use single-clock mode for single-device QDRII SRAM interfacing where the K and K# are used for write operations, and CQ and CQ# are used for read operations. You should use both C or C# and K or K# clocks when interfacing with a bank of multiple QDRII SRAM devices with a single controller.

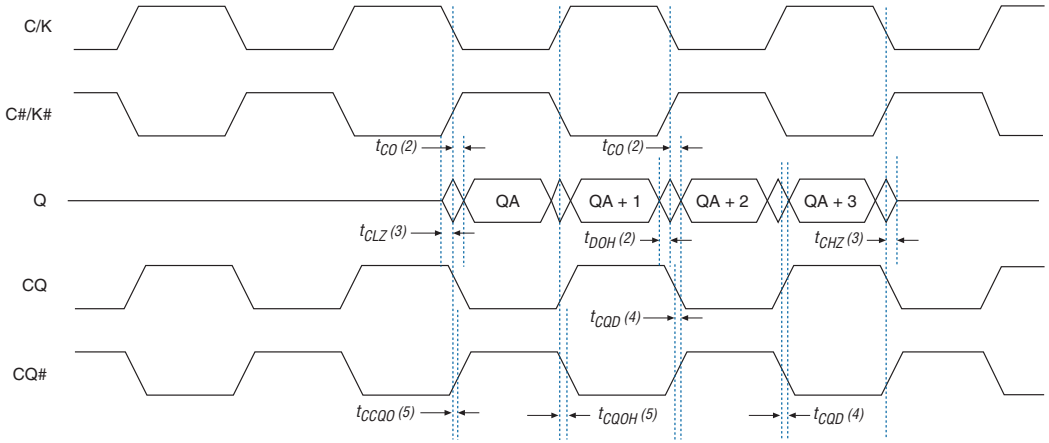
You can generate C, C#, K, and K# clocks using any of the I/O registers via the DDR registers. Because of strict skew requirements between K and K# signals, use adjacent pins to generate the clock pair.

Connect CQ and CQ# pins to the Stratix II or Stratix II GX DQS and DQSn pins for DLL-based implementations. You must configure DQS and DQSn as bidirectional pins. However, since CQ and CQ# pins are output-only pins from the memory, the Stratix II or Stratix II GX device QDRII SRAM memory interface requires that you ground the DQS and DQSn output enable. To capture data presented by the memory, connect the shifted CQ signal to the input latch and connect the active-high input registers and the shifted CQ# signal is connected to the active-low input register. For PLL-based implementations, connect QK to the input of the read PLL and leave QK# unconnected.

Read and Write Operations

Figure 9–5 shows the data and clock relationships in QDRII SRAM devices at the memory pins during reads. Data is output one-and-a-half clock cycles after a read command is latched into memory. QDRII SRAM devices send data within a t_{CO} time after each rising edge of the read clock C or C# in multi-clock mode, or the input clock K or K# in single clock mode. Data is valid until t_{DOH} time after each rising edge of the read clock C or C# in multi-clock mode or the input clock K or K# in single clock mode. The CQ and CQ# clocks are edge-aligned with the read data signal. These clocks accompany the read data for data capture in Stratix II and Stratix II GX devices.

Figure 9–5. Data and Clock Relationship During a QDRII SRAM Read *Note (1)*



Notes to Figure 9–5:

(1) This relationship is at the memory device. The timing parameter nomenclature is based on the Cypress QDRII SRAM data sheet for CY7C1313V18.

(2) t_{CO} is the data clock-to-out time and t_{DOH} is the data output hold time between burst.

(3) t_{CLZ} and t_{CHZ} are bus turn-on and turn-off times respectively.

(4) t_{CQD} is the skew between the rising edge of CQ or CQ# and the data edges.

(5) t_{CCQO} and t_{CQOH} are skew measurements between the C or C# clocks (or the K or K# clocks in single-clock mode) and the CQ or CQ# clocks.

When reading from the QDRII SRAM, data is sent edge-aligned with the rising edge of the echo clocks CQ and CQ#. Both CQ and CQ# are shifted inside the FPGA using DQS and DQSn logic blocks to capture the data in the DDR IOE registers in DLL-based implementations. In PLL-based implementations, CQ feeds a PLL, which generates the clock to capture the data in the DDR IOE registers.

When writing to QDRII SRAM devices, data is generated by the write clock while the K clock is 90° shifted from the write clock, creating a center-aligned arrangement.

Read and write operations occur during the same clock cycle on independent read and write data paths along with the cycle-shared address bus. Performing concurrent reads and writes does not change the functionality of either transaction. If a read request occurs simultaneously with a write request at the same address, the new data on D is forwarded to Q. Therefore, latency is not required to access valid data.

For more information on QDRII SRAM, go to www.qdrsram.com or see AN 326: Interfacing QDRII & QDRII+ SRAM with Stratix II, Stratix, & Stratix GX Devices.

Stratix II and Stratix II GX DDR Memory Support Overview

This section describes Stratix II and Stratix II GX features that enable high-speed memory interfacing. It first describes Stratix II and Stratix II GX memory pins and then the DQS phase-shift circuitry and the DDR I/O registers. Table 9–2 shows the I/O standard associated with the external memory interfaces.

Table 9–2. External Memory Support in Stratix II and Stratix II GX Devices

Memory Standard	I/O Standard
DDR SDRAM	SSTL-2 Class II
DDR2 SDRAM	SSTL-18 Class II *(1)*
RLDRAM II *(2)*	1.8-V HSTL Class I or II *(1)*
QDRII SRAM *(2)*	1.8-V HSTL Class I or II *(1)*

Notes to Table 9–2:

(1) Stratix II and Stratix II GX devices support 1.8-V HSTL/SSTL-18 Class I and II I/O standards in I/O banks 3, 4, 7, and 8. In I/O banks 1, 2, 5, and 6, Class I is supported for both input and output operations, while Class II is only supported for input operations for these I/O standards.

(2) For maximum performance, Altera recommends using the 1.8-V HSTL I/O standard. RLDRAM II and QDRII SRAM devices also support the 1.5-V HSTL I/O standard.

Stratix II and Stratix II GX devices support the data strobe or read clock signal (DQS) used in DDR SDRAM, DDR2 SDRAM, RLDRAM II, and QDRII SRAM devices with dedicated circuitry. Stratix II and Stratix II GX devices also support the DQSn signal (the DQS complement signal) for external memory types that require them, for example QDRII SRAM. DQS and DQSn signals are usually associated with a group of data (DQ) pins. However, these are not differential buffers and cannot be used in DDR2 SDRAM or RLDRAM II interfaces.

You can also interface with these external memory devices without the use of dedicated circuitry at a lower performance.

For more information, see the appropriate Stratix II or Stratix II GX memory interfaces application note available at www.altera.com.

Stratix II and Stratix II GX devices contain dedicated circuitry to shift the incoming DQS signals by 0°, 22.5°, 30°, 36°, 45°, 60°, 67.5°, 72°, 90°, 108°, 120°, or 144°, depending on the delay-locked loop (DLL) mode. There are four DLL modes. The DQS phase-shift circuitry uses a frequency reference to dynamically generate control signals for the delay chains in each of the DQS and DQSn pins, allowing it to compensate for process,

voltage, and temperature (PVT) variations. This phase-shift circuitry has been enhanced in Stratix II and Stratix II GX devices to support more phase-shift options with less jitter.

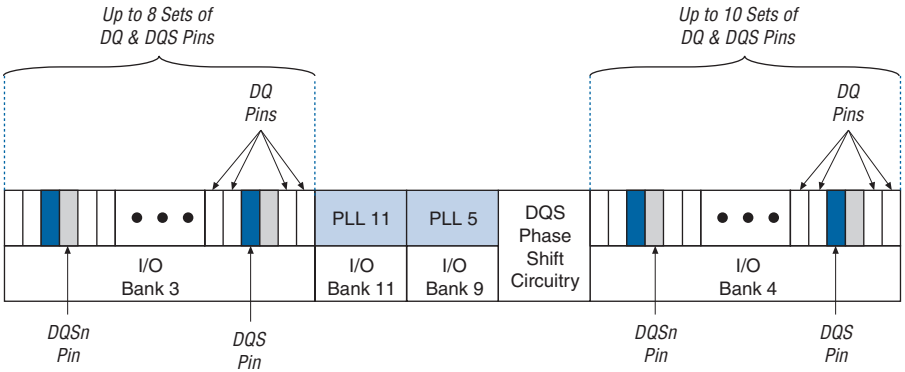
Besides the DQS dedicated phase-shift circuitry, each DQS and DQSn pin has its own DQS logic block that sets the delay for the signal input to the pin. Using the DQS dedicated phase-shift circuitry with the DQS logic block allows for phase-shift fine-tuning. Additionally, every IOE in a Stratix II or Stratix II GX device contains six registers and one latch to achieve DDR operation.

DDR Memory Interface Pins

Stratix II and Stratix II GX devices use data (DQ), data strobe (DQS and DQSn), and clock pins to interface with external memory.

Figure 9–6 shows the DQ, DQS, and DQSn pins in the Stratix II or Stratix II GX I/O banks on the top of the device. A similar arrangement is repeated at the bottom of the device.

Figure 9–6. DQ and DQS Pins Per I/O Bank



Data and Data Strobe Pins

Stratix II and Stratix II GX data pins for the DDR memory interfaces are called DQ pins. Stratix II and Stratix II GX devices can use either bidirectional data strobes or unidirectional read clocks. Depending on the external memory interface, either the memory device's read data strobes or read clocks feed the Stratix II or Stratix II GX DQS (and DQSn) pins.

Stratix II and Stratix II GX DQS pins connect to the DQS pins in DDR and DDR2 SDRAM interfaces or to the QK pins in RLDRAM II interfaces. The DQSn pins are not used in these interfaces. Connect the Stratix II or Stratix II GX DQS and DQSn pins to the QDRII SRAM CQ and CQ# pins, respectively.

In every Stratix II or Stratix II GX device, the I/O banks at the top (I/O banks 3 and 4) and bottom (I/O banks 7 and 8) of the device support DDR memory up to 300 MHz/600 Mbps (with RLDRAM II). These I/O banks support DQS signals and its complement DQSn signals with DQ bus modes of ×4, ×8/×9, ×16/×18, or ×32/×36.

In ×4 mode, each DQS/DQSn pin drives up to four DQ pins within that group. In ×8/×9 mode, each DQS/DQSn pin drives up to nine DQ pins within that group to support one parity bit and the eight data bits. If the parity bit or any data bit is not used, the extra DQ pins can be used as regular user I/O pins. Similarly, with ×16/×18 and ×32/×36 modes, each DQS/DQSn pin drives up to 18 and 36 DQ pins respectively. There are two parity bits in the ×16/×18 mode and four parity bits in the ×32/×36 mode. Tables 9–3 through 9–6 show the number of DQS/DQ groups and non-DQS /DQ supported in each Stratix II or Stratix II GX density/package combination, respectively, for DLL-based implementations.

Table 9–3. Stratix II DQS and DQ Bus Mode Support (Part 1 of 2) *Note (1)*

Device	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2S15	484-pin FineLine BGA	8	4	0	0
	672-pin FineLine BGA	18	8	4	0
EP2S30	484-pin FineLine BGA	8	4	0	0
	672-pin FineLine BGA	18	8	4	0
EP2S60	484-pin FineLine BGA	8	4	0	0
	672-pin FineLine BGA	18	8	4	0
	1,020-pin FineLine BGA	36	18	8	4
EP2S90	484-pin Hybrid FineLine BGA	8	4	0	0
	780-pin FineLine BGA	18	8	4	0
	1,020-pin FineLine BGA	36	18	8	4
	1,508-pin FineLine BGA	36	18	8	4
EP2S130	780-pin FineLine BGA	18	8	4	0
	1,020-pin FineLine BGA	36	18	8	4
	1,508-pin FineLine BGA	36	18	8	4

Table 9–3. Stratix II DQS and DQ Bus Mode Support (Part 2 of 2) *Note (1)*

Device	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2S180	1,020-pin FineLine BGA	36	18	8	4
	1,508-pin FineLine BGA	36	18	8	4

Note to Table 9–3:
(1) Check the pin table for each DQS/DQ group in the different modes.

Table 9–4. Stratix II non-DQS and DQ Bus Mode Support *Note (1)*

Device	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2S15	484-pin FineLine BGA	13	7	3	1
	672-pin FineLine BGA	24	9	4	2
EP2S30	484-pin FineLine BGA	13	7	3	1
	672-pin FineLine BGA	36	15	7	3
EP2S60	484-pin FineLine BGA	13	7	3	1
	672-pin FineLine BGA	36	15	7	3
	1,020-pin FineLine BGA	51	26	13	6
EP2S90	780-pin FineLine BGA	40	24	12	6
	1,020-pin FineLine BGA	51	25	12	6
	1,508-pin FineLine BGA	51	25	12	6
EP2S130	780-pin FineLine BGA	40	24	12	6
	1,020-pin FineLine BGA	51	25	12	6
	1,508-pin FineLine BGA	51	25	12	6
EP2S180	1,020-pin FineLine BGA	51	25	12	6
	1,508-pin FineLine BGA	51	25	12	6

Note to Table 9–4:
(1) Check the pin table for each DQS/DQ group in the different modes.

Table 9–5. Stratix II GX DQS and DQ Bus Mode Support *Note (1)*

Device	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2SGX30C EP2SGX30D	780-pin FineLine BGA	18	8	4	0
EP2SGX60C EP2SGX60D	780-pin FineLine BGA	18	8	4	0
EP2SGX60E	1,152-pin FineLine BGA	36	18	8	4
EP2SGX90E	1,152-pin FineLine BGA	36	18	8	4
EP2SGX90F	1,508-pin FineLine BGA	36	18	8	4
EP2SGX130G	1,508-pin FineLine BGA	36	18	8	4

Note to Table 9–5:
(1) Check the pin table for each DQS/DQ group in the different modes.

Table 9–6. Stratix II GX Non-DQS and DQ Bus Mode Support *Note (1)*

Device	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2SGX30	780-pin FineLine BGA	18	8	4	2
EP2SGX60	780-pin FineLine BGA	18	8	4	2
	1,152-pin FineLine BGA	25	13	6	3
EP2SGX90	1,152-pin FineLine BGA	25	13	6	3
	1,508-pin FineLine BGA	25	12	6	3
EP2SGX130	1,508-pin FineLine BGA	25	12	6	3

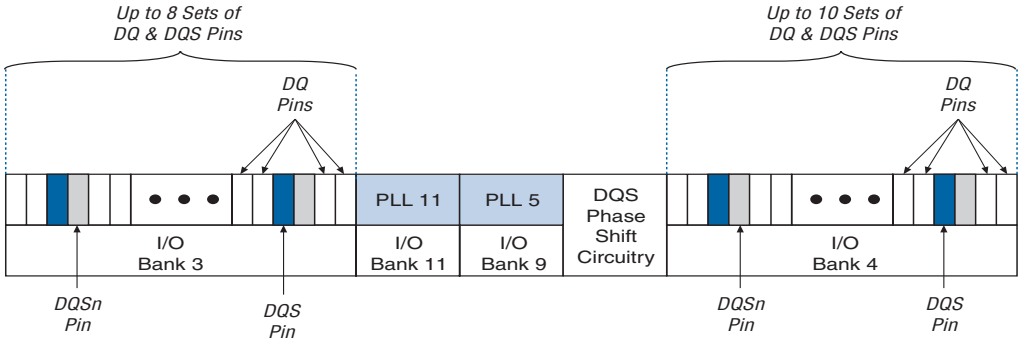
Note to Table 9–6:
(1) Check the pin table for each DQS/DQ group in the different modes.

To support the RLDRAM II QVLD pin, some of the unused ×4 DQS pins, whose DQ pins were combined to make the bigger ×8/×9, ×16/×18, or ×32/×36 groups, are listed as DQVLD pins in the Stratix II or Stratix II GX pin table. DQVLD pins are for input-only operations. The signal coming into this pin can be captured by the shifted DQS signal like any of the DQ pins.

The DQS pins are listed in the Stratix II or Stratix II GX pin tables as DQS[17..0]T or DQS[17..0]B. The T denotes pins on the top of the device and the B denotes pins on the bottom of the device. The complement DQSn pins are marked as DQSn[17..0]T or DQSn[17..0]B. The corresponding DQ pins are marked as DQ[17..0]T[3..0], where [17..0] indicates which DQS group the pins belong to. Similarly, the corresponding DQVLD pins are marked as DQVLD[8..0]T, where [8..0] indicates which DQS group the pins belong to. The numbering scheme starts from right to left on the package bottom view. When not used as DQ, DQS, or DQSn pins, these pins are available as regular I/O pins. Figure 9–7 shows the DQS pins in Stratix II or Stratix II GX I/O banks.

The Quartus II software treats DQVLD pins as regular DQ pins. Therefore, you must ensure that the DQVLD pin assigned in your design corresponds to the pin table's recommended DQVLD pins.

Figure 9–7. DQS Pins in Stratix II and Stratix II GX I/O Banks *Notes (1), (2), (3)*



Notes to Figure 9–7:

(1) There are up to 18 pairs of DQS and DQSn pins on both the top and bottom of the device. See Table 9–3 for the exact number of DQS and DQSn pin pairs in each device package.

(2) See Table 9–7 for the available DQS and DQSn pins in each mode and package.

(3) Each DQS pin has a complement DQSn pin. DQS and DQSn pins are not differential.

The DQ pin numbering is based on ×4 mode. There are up to 8 DQS/DQ groups in ×4 mode in I/O banks 3 and 8 and up to 10 DQS/DQ groups in ×4 mode in I/O banks 4 and 7. In ×8/×9 mode, two adjacent ×4 DQS/DQ groups plus one parity pin are combined; one pair of DQS/DQSn pins from the combined groups can drive all the DQ and parity pins. Since there is an even number of DQS/DQ groups in an I/O bank, combining groups is efficient. Similarly, in ×16/×18 mode, four adjacent ×4 DQS/DQ groups plus two parity pins are combined and one pair of DQS/DQSn pins from the combined groups can drive all the DQ and parity pins. In

×32/×36 mode, eight adjacent DQS/DQ groups are combined and one pair of DQS/DQSn pins can drive all the DQ and parity pins in the combined groups.

Table 9–7 shows which DQS and DQSn pins are available in each mode and package in the Stratix II or Stratix II GX device family.

Table 9–7. Available DQS and DQSn Pins in Each Mode and Package *Note (1)*

Mode	Package		
	484-Pin FineLine BGA 484-Pin Hybrid FineLine BGA	672-Pin FineLine BGA 780-Pin FineLine BGA	1,020-Pin FineLine BGA 1,508-Pin FineLine BGA
×4	7, 9, 11, 13	Odd-numbered pins only	All DQS and DQSn pins
×8/×9	7,11	3, 7, 11, 15	Even-numbered pins only
×16/×18	N/A	5, 13	3, 7, 11, 15
×32/×36	N/A	N/A	5, 13

Note to Table 9–7:

(1) The numbers correspond to the DQS and DQSn pin numbering in the Stratix II or Stratix II GX pin table. There are two sets of DQS/DQ groups, one corresponding with the top side of the device and one with the bottom side of the device.

On the top and bottom side of the device, the DQ and DQS pins must be configured as bidirectional DDR pins to enable the DQS phase-shift circuitry. The DQSn pins can be configured as input, output, or bidirectional pins. You can use the `altdq` and `altdqs` megafunctions to configure the DQ and DQS/DQSn paths, respectively. However, Altera highly recommends that you use the respective Altera memory controller IP Tool Bench for your external memory interface data paths. The data path is clear-text and free to use. You are responsible for your own timing analysis if you use your own data path. If you only want to use the DQ and/or DQS pins as inputs, you need to set the output enable of the DQ and/or DQS pins to ground.

Stratix II or Stratix II GX side I/O banks (I/O banks 1, 2, 5, and 6) support all the memory interfaces supported in the top and bottom I/O banks. For optimal performance, use the Altera memory controller IP Tool Bench to pick the data and strobe pins for these interfaces. Since these I/O banks do not have any dedicated circuitry for memory interfacing, they can support DDR SDRAM at speeds up to 150 MHz and other DDR memories at speeds up to 200 MHz. You need to use the SSTL-18 Class I I/O standard when interfacing with DDR2 SDRAM devices using pins in I/O bank 1, 2, 5, or 6. These I/O banks do not support the SSTL-18 Class II and

1.8-V HSTL Class II I/O standards on output and bidirectional pins, but you can use SSTL-18 Class I or 1.8-V HSTL Class I I/O standards for memory interfaces.

The Altera memory controller IP Tool Bench generates the optimal pin constraints that allow you to interface these memories at high frequency.

Table 9–8 shows the maximum clock rate supported for the DDR SDRAM interface in the Stratix II or Stratix II GX device side I/O banks.

Table 9–8. Maximum Clock Rate for DDR and DDR2 SDRAM in Stratix II or Stratix II GX Side I/O Banks

Stratix II or Stratix II GX Device Speed Grade	DDR SDRAM (MHz)	DDR2 SDRAM (MHz)	QDRII SRAM (MHz)	RLDRAM II (MHz)
-3	150	200	200	200
-4	133	167	167	175
-5	133	167	167	175

Clock Pins

You can use any of the DDR I/O registers to generate clocks to the memory device. For better performance, use the same I/O bank as the data and address/command pins.

Command and Address Pins

You can use any of the user I/O pins in the top or bottom bank of the device for commands and addresses. For better performance, use the same I/O bank as the data pins.

Other Pins (Parity, DM, ECC and QVLD Pins)

You can use any of the DQ pins for the parity pins in Stratix II and Stratix II GX devices. The Stratix II or Stratix II GX device family has support for parity in the ×8/×9, ×16/×18, and ×32/×36 mode. There is one parity bit available per 8 bits of data pins.

The data mask, DM, pins are only required when writing to DDR SDRAM, DDR2 SDRAM, and RLDRAM II devices. A low signal on the DM pins indicates that the write is valid. If the DM signal is high, the memory will mask the DQ signals. You can use any of the I/O pins in the same bank as the DQ pins (or the RLDRAM II SIO's and QDRII SRAM's D pins) for the DM signals. Each group of DQS and DQ signals in DDR

and DDR2 SDRAM devices requires a DM pin. There is one DM pin per RLDRAM II device. The DDR I/O output registers, clocked by the –90° shifted clock, creates the DM signals, similar to DQ output signals.

Perform timing analysis to calculate your write-clock phase shift.

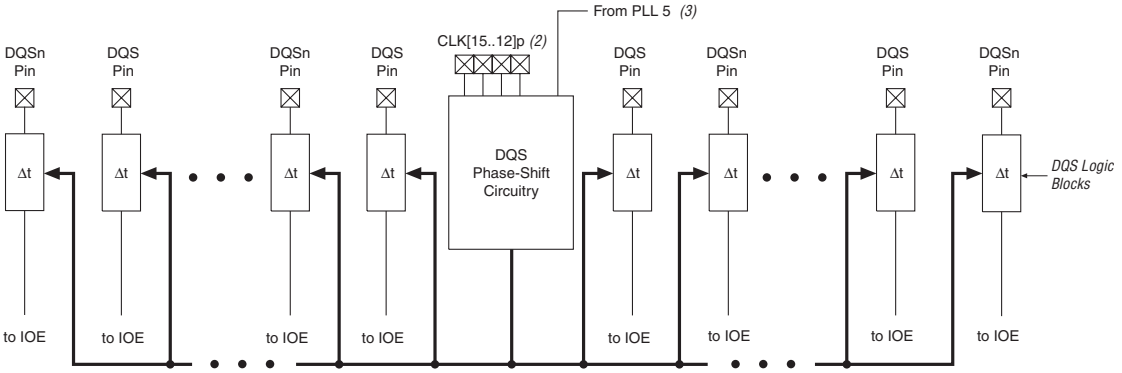
Some DDR SDRAM and DDR2 SDRAM devices support error correction coding (ECC), which is a method of detecting and automatically correcting errors in data transmission. In a 72-bit DDR SDRAM interface, there are eight ECC pins in addition to the 64 data pins. Connect the DDR and DDR2 SDRAM ECC pins to a Stratix II or Stratix II GX device DQS/DQ group. The memory controller needs extra logic to encode and decode the ECC data.

QVLD pins are used in RLDRAM II interfacing to indicate the read data availability. There is one QVLD pin per RLDRAM II device. A high on QVLD indicates that the memory is outputting the data requested. Similar to DQ inputs, this signal is edge-aligned with QK/QK# signals and is sent half a clock cycle before data starts coming out of the memory. You need to connect QVLD pins to the DQVLD pin on the Stratix II or Stratix II GX device. The DQVLD pin can be used as a regular user I/O pin if not used for QVLD. Because the Quartus II software does not differentiate DQVLD pins from DQ pins, you must ensure that your design uses the pin table's recommended DQVLD pin.

DQS Phase-Shift Circuitry

The Stratix II or Stratix II GX phase-shift circuitry and the DQS logic block control the DQS and DQSn pins. Each Stratix II or Stratix II GX device contains two phase-shifting circuits. There is one circuit for I/O banks 3 and 4, and another circuit for I/O banks 7 and 8. The phase-shifting circuit on the top of the device can control all the DQS and DQSn pins in the top I/O banks and the phase-shifting circuit on the bottom of the device can control all the DQS and DQSn pins in the bottom I/O banks. Figure 9–8 shows the DQS and DQSn pin connections to the DQS logic block and the DQS phase-shift circuitry.

Figure 9–8. DQS and DQSn Pins and the DQS Phase-Shift Circuitry *Note (1)*



Notes to Figure 9–8:

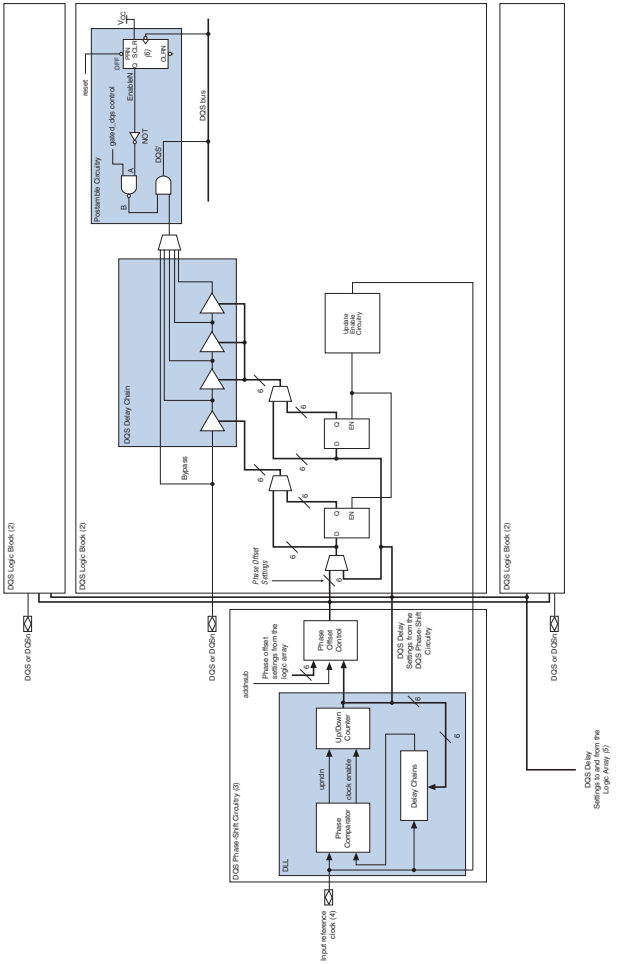
(1) There are up to 18 pairs of DQS and DQSn pins available on the top or the bottom of the Stratix II or Stratix II GX device, up to 8 on the left side of the DQS phase-shift circuitry (I/O banks 3 and 8), and up to 10 on the right side (I/O bank 4 and 7).

(2) Clock pins `CLK[15..12]p` feed the phase-shift circuitry on the top of the device and clock pins `CLK[7..4]p` feed the phase-shift circuitry on the bottom of the device. You can also use a phase-locked loop (PLL) clock output as a reference clock to the phase-shift circuitry. The reference clock can also be used in the logic array.

(3) You can only use PLL 5 to feed the DQS phase-shift circuitry on the top of the device and PLL 6 to feed the DQS phase-shift circuitry on the bottom of the device.

Figure 9–9 shows the connections between the DQS phase-shift circuitry and the DQS logic block.

Figure 9–9. DQS Phase-Shift Circuitry and DQS Logic Block Connections *Note (1)*



Notes to Figure 9–9:

(1) All features of the DQS phase-shift circuitry and the DQS logic block are accessible from the `altdqs` megafunction in the Quartus II software. You should, however, use Altera's memory controller IP Tool Bench to generate the data path for your memory interface.

(2) DQS logic block is available on every DQS and DQSn pin.

(3) There is one DQS phase-shift circuit on the top and bottom side of the device.

(4) The input reference clock can come from `CLK[15..12]p` or PLL 5 for the DQS phase-shift circuitry on the top side of the device or from `CLK[7..4]p` or PLL 6 for the DQS phase-shift circuitry on the bottom side of the device.

(5) Each individual DQS and DQSn pair can have individual DQS delay settings to and from the logic array.

(6) This register is one of the DQS IOE input registers.

The phase-shift circuitry is only used during read transactions where the DQS and DQSn pins are acting as input clocks or strobes. The phase-shift circuitry can shift the incoming DQS signal by 0°, 22.5°, 30°, 36°, 45°, 60°, 67.5°, 72°, 90°, 108°, 120°, or 144°. The shifted DQS signal is then used as clocks at the DQ IOE input registers.

Figure 9–3 shows an example where the DQS signal is shifted by 90°. The DQS signals goes through the 90° shift delay set by the DQS phase-shift circuitry and the DQS logic block and some routing delay from the DQS pin to the DQ IOE registers. The DQ signals only goes through routing delay from the DQ pin to the DQ IOE registers and maintains the 90° relationship between the DQS and DQ signals at the DQ IOE registers since the software will automatically set delay chains to match the routing delay between the pins and the IOE registers for the DQ and DQS input paths.

All 18 DQS and DQSn pins on either the top or bottom of the device can have their input signal phase shifted by a different degree amount but all must be referenced at one particular frequency. For example you can have a 90° phase shift on `DQS0T` and have a 60° phase shift on `DQS1T` both referenced from a 200-MHz clock. Not all phase-shift combinations are supported, however. The phase shifts on the same side of the device must all be a multiple of 22.5° (up to 90°), a multiple of 30° (up to 120°), or a multiple of 36° (up to 144°).

In order to generate the correct phase shift with the DLL used, you must provide a clock signal of the same frequency as the DQS signal to the DQS phase-shift circuitry. Any of the `CLK[15..12]p` clock pins can feed the phase circuitry on the top of the device (I/O banks 3 and 4) or any of the `CLK[7..4]p` clock pins can feed the phase circuitry on the bottom of the device (I/O banks 7 and 8). Stratix II and Stratix II GX devices can also use PLLs 5 or 6 as the reference clock to the DQS phase-shift circuitry on the top or bottom of the device, respectively. PLL 5 is connected to the DQS phase-shift circuitry on the top side of the device and PLL 6 is connected to the DQS phase-shift circuitry on the bottom side of the device. Both the top and bottom phase-shift circuits need unique clock pins or PLL clock outputs for the reference clock.

When you have a PLL dedicated only to generate the DLL input reference clock, you must set the PLL mode to "No Compensation" or the Quartus® II software will change it automatically. Because there are no other PLL outputs used, the PLL doesn't need to compensate for any clock paths.

DLL

The DQS phase-shift circuitry uses a delay-locked loop (DLL) to dynamically measure the clock period needed by the DQS/DQSn pin (see Figure 9–10). The DQS phase-shift circuitry then uses the clock period to generate the correct phase shift. The DLL in the Stratix II or Stratix II GX DQS phase-shift circuitry can operate between 100 and 400 MHz. The phase-shift circuitry needs a maximum of 256 clock cycles to calculate the correct input clock period. Data sent during these clock cycles may not be properly captured.

Although the DLL can run up to 400 MHz, other factors may prevent you from interfacing with a 400-MHz external memory device.

You can still use the DQS phase-shift circuitry for any memory interfaces that are less than 100 MHz. The DQS signal will be shifted by 2.5 ns and you can add more shift by using the phase offset module. Even if the DQS signal is not shifted exactly to the middle of the DQ valid window, the IOE should still be able to capture the data in this low frequency application.

There are four different frequency modes for the Stratix II or Stratix II GX DLL. Each frequency mode provides different phase shift, as shown in Table 9–9.

Table 9–9. Stratix II and Stratix II GS DLL Frequency Modes

Frequency Mode	Frequency Range (MHz)	Available Phase Shift	Number of Delay Chains
0	100–175	30, 60, 90, 120	12
1	150–230	22.5, 45, 67.5, 90	16
2	200–310	30, 60, 90, 120	12
3	240–400 (C3 speed grade) 240–350 (C4 and C5 speed grades)	36, 72, 108, 144	10

In frequency mode 0, Stratix II devices use a 6-bit setting to implement the phase-shift delay. In frequency modes 1, 2, and 3, Stratix II devices only use a 5-bit setting to implement the phase-shift delay.

The DLL can be reset from either the logic array or a user I/O pin. This signal is not shown in Figure 9–10. Each time the DLL is reset, you must wait for 256 clock cycles before you can capture the data properly.

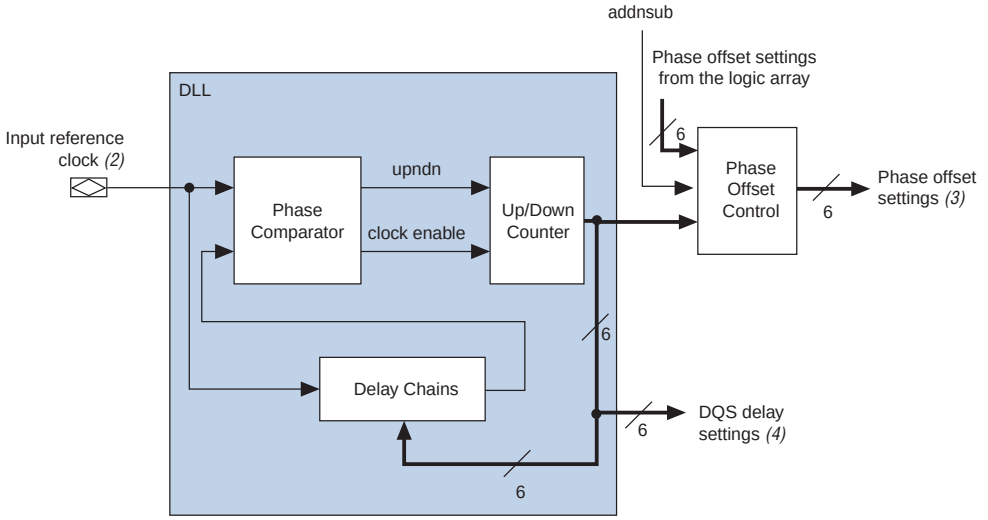
The input reference clock for the DQS phase-shift circuitry on the top side of the device can come from CLK[15..12]p or PLL 5. The input reference clock for the DQS phase-shift circuitry on the bottom side of the device can come from CLK[7..4]p or PLL 6.

Table 9–10 lists the maximum delay in the fast timing model for the Stratix II DQS delay buffer. Multiply the number of delay buffers that you are using in the DQS logic block to get the maximum delay achievable in your system. For example, if you implement a 90° phase shift at 200 MHz, you use three delay buffers in mode 2. The maximum achievable delay from the DQS block is then 3 × .416 ps = 1.248 ns.

Table 9–10. DQS Delay Buffer Maximum Delay in Fast Timing Model

Frequency Mode	Maximum Delay Per Delay Buffer (Fast Timing Model)	Unit
0	0.833	ns
1, 2, 3	0.416	ns

Figure 9–10. Simplified Diagram of the DQS Phase-Shift Circuitry *Note (1)*



Notes to Figure 9–10:

(1) All features of the DQS phase-shift circuitry are accessible from the altdqs megafunction in the Quartus II software. You should; however, use Altera's memory controller IP Tool Bench to generate the data path for your memory interface.

(2) The input reference clock for the DQS phase-shift circuitry on the top side of the device can come from `CLK[15..12]p` or PLL 5. The input reference clock for the DQS phase-shift circuitry on the bottom side of the device can come from `CLK[7..4]p` or PLL 6.

(3) Phase offset settings can only go to the DQS logic blocks.

(4) DQS delay settings can go to the logic array and/or to the DQS logic block.

The input reference clock goes into the DLL to a chain of up to 16 delay elements. The phase comparator compares the signal coming out of the end of the delay element chain to the input reference clock. The phase comparator then issues the `upndn` signal to the up/down counter. This signal increments or decrements a six-bit delay setting (DQS delay settings) that will increase or decrease the delay through the delay element chain to bring the input reference clock and the signals coming out of the delay element chain in phase.

The DQS delay settings contain the control bits to shift the signal on the input DQS pin by the amount set in the `altdqs` megafunction. For the 0° shift, both the DLL and the DQS logic block are bypassed. Since Stratix II and Stratix II GX DQS and DQ pins are designed such that the pin to IOE delays are matched, the skew between the DQ and DQS pin at the DQ IOE registers is negligible when the 0° shift is implemented. You can feed the DQS delay settings to the DQS logic block and the logic array.

Phase Offset Control

The DQS phase-shift circuitry also contains a phase offset control module that can add or subtract a phase offset amount from the DQS delay setting (phase offset settings from the logic array in Figure 9–10). You should use the phase offset control module for making small shifts to the input signal and use the DQS phase-shift circuitry for larger signal shifts. For example, if you need the input signal to be shifted by 75°, you can set the `altdqs` megafunction to generate a 72° phase shift with a phase offset of +3°.

You can either use a static phase offset or a dynamic phase offset to implement the additional phase shift. The available additional phase shift is implemented in 2s-complement between settings –64 to +63 for frequency mode 0, and between settings –32 to +31 for frequency modes 1, 2, and 3. However, the DQS delay settings are at the maximum at setting 64 for frequency mode 0, and at the maximum at setting 32 for frequency modes 1, 2, and 3. Therefore, the actual physical offset setting range will be 64 or 32 subtracted by the DQS delay settings from the DLL.

For example, if the DLL determines that to achieve 30° you will need a DQS delay setting of 28, you can subtract up to 28 phase offset settings and you can add up to 36 phase offset settings to achieve the optimal delay.

Each phase offset setting translates to a certain delay, as specified in the *DC & Switching Characteristics of Stratix III Devices* chapter in volume 2 of the *Stratix III Device Handbook*.

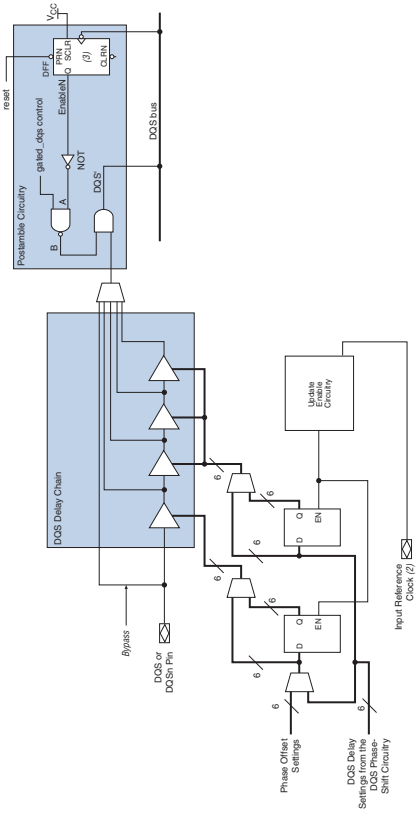
When using the static phase offset, you can specify the phase offset amount in the `altdqs` megafunction as a positive number for addition or a negative number for subtraction. You can also have a dynamic phase offset that is always added to, subtracted from, or both added to and subtracted from the DLL phase shift. When you always add or subtract, you can dynamically input the phase offset amount into the `dll_offset[5..0]` port. When you want to both add and subtract dynamically, you control the `addnsub` signal in addition to the `dll_offset[5..0]` signals.

DQS Logic Block

Each DQS and DQSn pin is connected to a separate DQS logic block (see Figure 9–11). The logic block contains DQS delay chains and postamble circuitry.

The input reference clock for the DQS phase-shift circuitry on the top side of the device can come from CLK[15..12]p or PLL 5. The input reference clock for the DQS phase-shift circuitry on the bottom side of the device can come from CLK[7..4]p or PLL 6.

Figure 9–11. Simplified Diagram of the DQS Logic Block *Note (1)*



Notes to Figure 9–11:

(1) All features of the DQS logic block are accessible from the altdqs megafunction in the Quartus II software. You should; however, use Altera's memory controller IP Tool Bench to generate the data path for your memory interface.

(2) The input reference clock for the DQS phase-shift circuitry on the top side of the device can come from CLK[15..12]p or PLL 5. The input reference clock for the DQS phase-shift circuitry on the top side of the device can come from CLK[7..4]p or PLL 6.

(3) This register is one of the DQS IOE input registers.

DQS Delay Chains

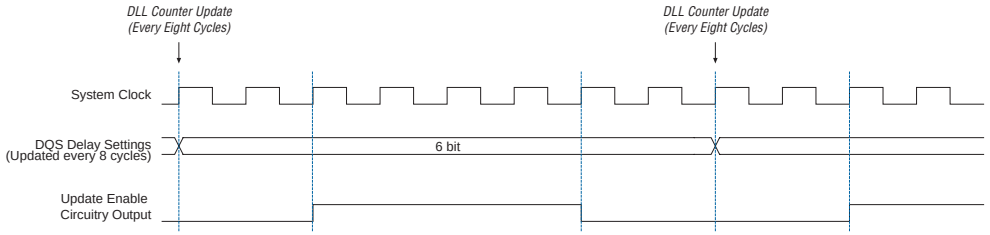
The DQS delay chains consist of a set of variable delay elements to allow the input DQS and DQSn signals to be shifted by the amount given by the DQS phase-shift circuitry or the logic array. There are four delay elements in the DQS delay chain; the first delay chain closest to the DQS pin can either be shifted by the DQS delay settings or by the sum of the DQS delay setting and the phase-offset setting. The number of delay chains used is transparent to the users because the `altdqs` megafunction automatically sets it. The DQS delay settings can come from the DQS phase-shift circuitry on the same side of the device as the target DQS logic block or from the logic array. When you apply a 0° shift in the `altdqs` megafunction, the DQS delay chains are bypassed.

The delay elements in the DQS logic block mimic the delay elements in the DLL. When the DLL is not used to control the DQS delay chains, you can input your own 6- or 5-bit settings using the `dqs_delayctrlin[5..0]` signals available in the `altdqs` megafunction. These settings control 1, 2, 3, or all 4 delay elements in the DQS delay chains. The amount of delay is equal to the sum of the delay element's intrinsic delay and the product of the number of delay steps and the value of the delay steps.

Both the DQS delay settings and the phase-offset settings pass through a latch before going into the DQS delay chains. The latches are controlled by the update enable circuitry to allow enough time for any changes in the DQS delay setting bits to arrive to all the delay elements. This allows them to be adjusted at the same time. The update enable circuitry enables the latch to allow enough time for the DQS delay settings to travel from the DQS phase-shift circuitry to all the DQS logic blocks before the next change. It uses the input reference clock to generate the update enable output. The `altdqs` megafunction uses this circuit by default. See Figure 9–12 for an example waveform of the update enable circuitry output.

The shifted DQS signal then goes to the DQS bus to clock the IOE input registers of the DQ pins. It can also go into the logic array for resynchronization purposes. The shifted DQSn signal can only go to the active-low input register in the DQ IOE and is only used for QDRII SRAM interfaces.

Figure 9–12. DQS Update Enable Waveform



DQS Postamble Circuitry

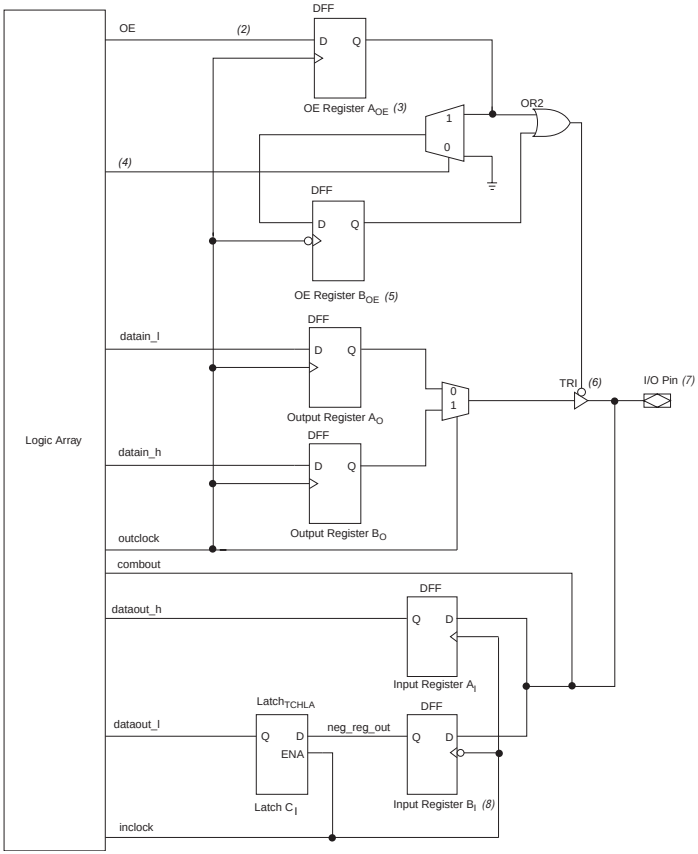
For external memory interfaces that use a bidirectional read strobe like DDR and DDR2 SDRAM, the DQS signal is low before going to or coming from a high-impedance state. See Figure 9–3. The state where DQS is low, just after a high-impedance state, is called the preamble and the state where DQS is low, just before it returns to a high-impedance state, is called the postamble. There are preamble and postamble specifications for both read and write operations in DDR and DDR2 SDRAM. The DQS postamble circuitry ensures data is not lost when there is noise on the DQS line at the end of a read postamble time. It is to be used with one of the DQS IOE input registers such that the DQS postamble control signal can ground the shifted DQS signal used to clock the DQ input registers at the end of a read operation. This ensures that any glitches on the DQS input signals at the end of the read postamble time do not affect the DQ IOE registers.

See *AN 327: Interfacing DDR SDRAM with Stratix II Devices* and *AN 328: Interfacing DDR2 SDRAM with Stratix II Devices* for more details.

DDR Registers

Each IOE in a Stratix II or Stratix II GX device contains six registers and one latch. Two registers and a latch are used for input, two registers are used for output, and two registers are used for output enable control. The second output enable register provides the write preamble for the DQS strobe in the DDR external memory interfaces. This active low output enable register extends the high-impedance state of the pin by a half clock cycle to provide the external memory's DQS write preamble time specification. Figure 9–13 shows the six registers and the latch in the Stratix II or Stratix II GX IOE and Figure 9–14 shows how the second OE register extends the DQS high-impedance state by half a clock cycle during a write operation.

Figure 9–13. Bidirectional DDR I/O Path in Stratix II and Stratix II GX Devices *Note (1)*



Notes to Figure 9–13:

(1) All control signals can be inverted at the IOE. The signal names used here match with Quartus II software naming convention.

(2) The OE signal is active low, but the Quartus II software implements this as active high and automatically adds an inverter before input to the A_{OE} register during compilation.

(3) The A_{OE} register generates the enable signal for general-purpose DDR I/O applications.

(4) This select line is to choose whether the OE signal should be delayed by half-a-clock cycle.

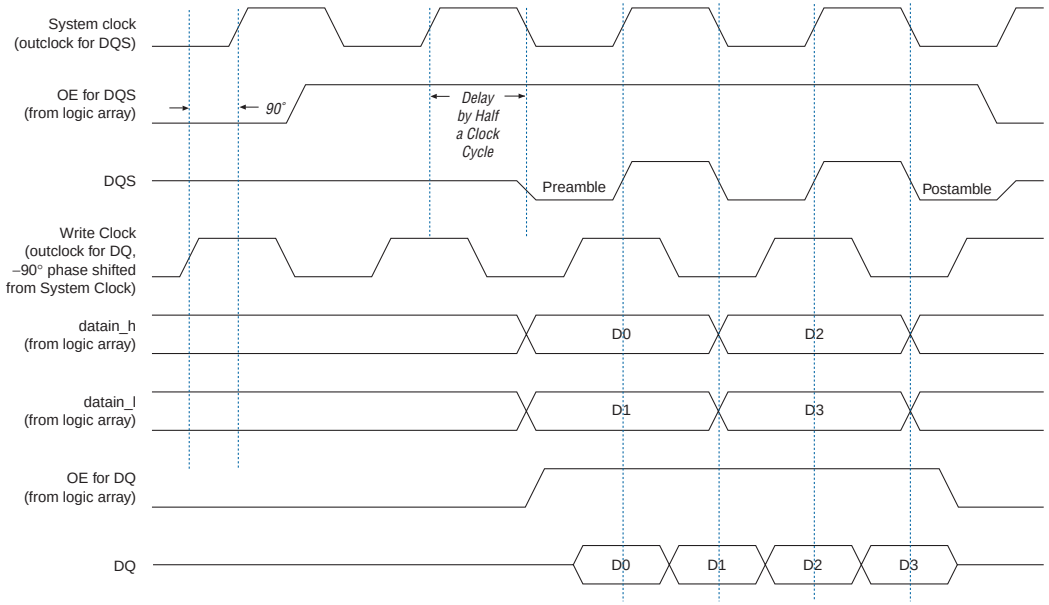
(5) The B_{OE} register generates the delayed enable signal for the write strobes or write clocks for memory interfaces.

(6) The tristate enable is by default active low. You can, however, design it to be active high. The combinational control path for the tristate is not shown in this diagram.

(7) You can also have combinational output to the I/O pin; this path is not shown in the diagram.

(8) On the top and bottom I/O banks, the clock to this register can be an inverted register A's clock or a separate clock (inverted or non-inverted). On the side I/O banks, you can only use the inverted register A's clock for this port.

Figure 9–14. Extending the OE Disable by Half-a-Clock Cycle for a Write Transaction *Note (1)*

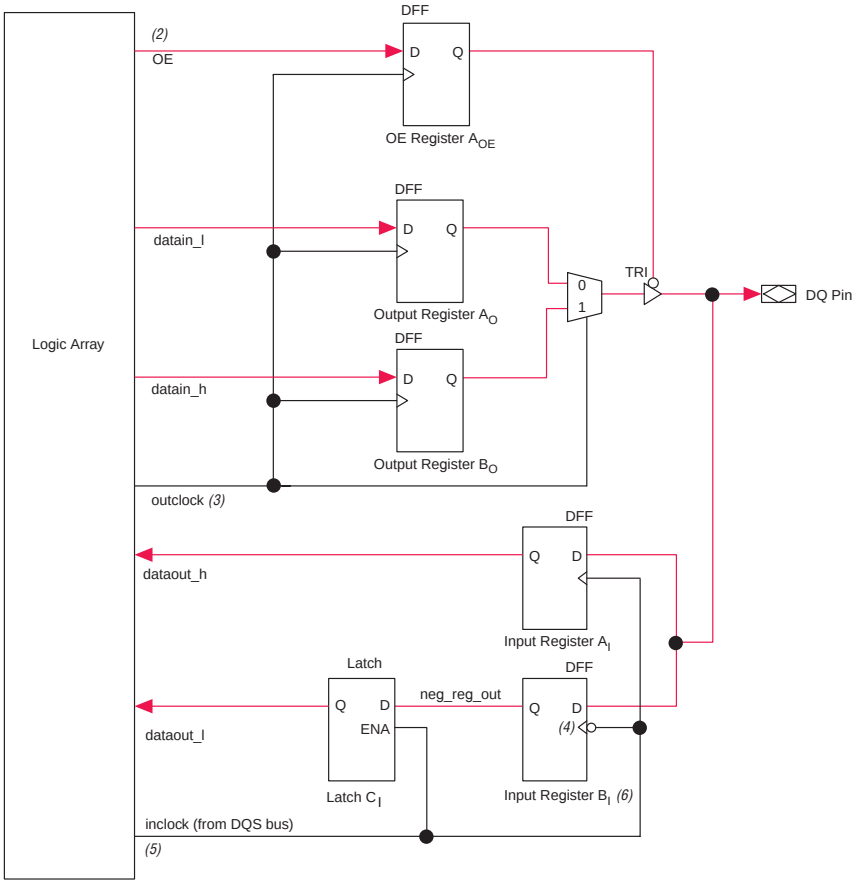


Note to Figure 9–14:

(1) The waveform reflects the software simulation result. The OE signal is an active low on the device. However, the Quartus II software implements this signal as an active high and automatically adds an inverter before the A_{OE} register D input.

Figures 9–15 and 9–16 summarize the IOE registers used for the DQ and DQS signals.

Figure 9–15. DQ Configuration in Stratix II or Stratix II GX IOE *Note (1)*



Notes to Figure 9–15:

(1) You can use the `altdq` megafunction to generate the DQ signals. You should, however, use Altera's memory controller IP Tool Bench to generate the data path for your memory interface. The signal names used here match with Quartus II software naming convention.

(2) The `OE` signal is active low, but the Quartus II software implements this as active high and automatically adds an inverter before the `OE` register A_{OE} during compilation.

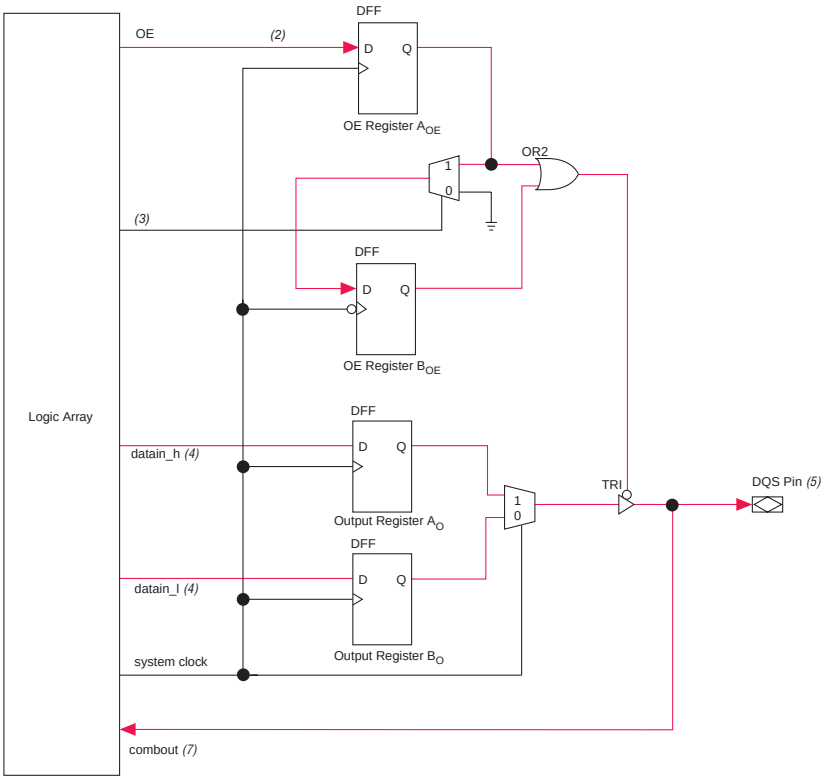
(3) The `outclock` signal for DDR, DDR2 SDRAM, and QDRII SRAM interfaces has a 90° phase-shift relationship with the system clock. For 300-MHz RLDRAM II interfaces with EP2S60F1020C3, Altera recommends a 75° phase-shift relationship.

(4) The shifted DQS or DQSn signal can clock this register. Only use the DQSn signal for QDRII SRAM interfaces.

(5) The shifted DQS signal must be inverted before going to the DQ IOE. The inversion is automatic if you use the `altdq` megafunction to generate the DQ signals. Connect this port to the `combout` port in the `altdqs` megafunction.

(6) On the top and bottom I/O banks, the clock to this register can be an inverted register A's clock or a separate clock (inverted or non-inverted). On the side I/O banks, you can only use the inverted register A's clock for this port.

Figure 9–16. DQS Configuration in Stratix II or Stratix II GX IOE *Note (1)*



Notes to Figure 9–16:

(1) You can use the `altdqs` megafunction to generate the DQS signals. You should, however, use Altera's memory controller IP Tool Bench to generate the data path for your memory interface. The signal names used here match with Quartus II software naming convention.

(2) The `OE` signal is active low, but the Quartus II software implements this as active high and automatically adds an inverter before `OE` register A_{OE} during compilation. In RLDRAM II and QDRII SRAM, the `OE` signal is always disabled.

(3) The select line can be chosen in the `altdqs` megafunction.

(4) The `datain_l` and `datain_h` pins are usually connected to ground and V_{CC}, respectively.

(5) DQS postamble circuitry and handling is not shown in this diagram. For more information, see *AN 327: Interfacing DDR SDRAM with Stratix II Devices* and *AN 328: Interfacing DDR2 SDRAM with Stratix II Devices*.

(6) DQS logic blocks are only available with DQS and DQSn pins.

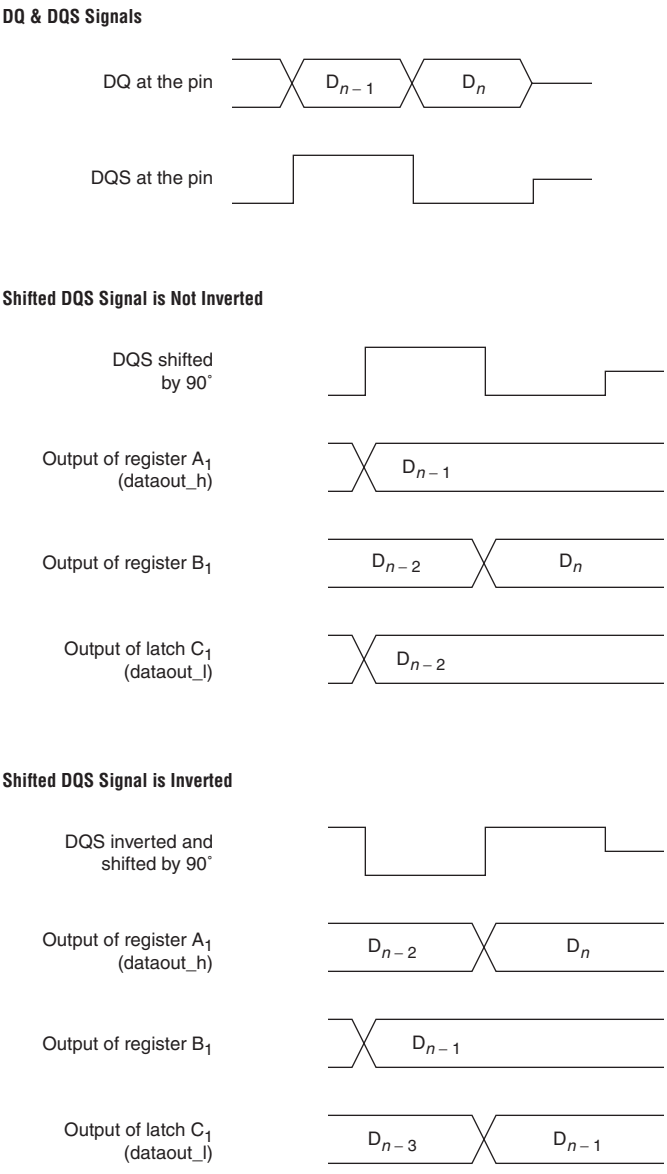
(7) You must invert this signal before it reaches the DQ IOE. This signal is automatically inverted if you use the `altdq` megafunction to generate the DQ signals. Connect this port to the `inclock` port in the `altdq` megafunction.

For interfaces to DDR SDRAM, DDR2 SDRAM, and RLDRAM II, the Stratix II or Stratix II GX DDR IOE structure requires you to invert the incoming DQS signal to ensure proper data transfer. This is not required for QDRII SRAM interfaces if the CQ signal is wired to the DQS pin and the CQ# signal is wired to the DQSn pin. The `altdq` megafunction, by default, adds the inverter to the `inclock` port when it generates DQ blocks. The megafunction also includes an option to remove the inverter for QDRII SRAM interfaces. As shown in Figure 9–13, the `inclock` signal's rising edge clocks the A_I register, `inclock` signal's falling edge clocks the B_I register, and latch C_I is opened when `inclock` is 1. In a DDR memory read operation, the last data coincides with DQS being low. If you do not invert the DQS pin, you will not get this last data as the latch does not open until the next rising edge of the DQS signal.

Figure 9–17 shows waveforms of the circuit shown in Figure 9–15.

The first set of waveforms in Figure 9–17 shows the edge-aligned relationship between the DQ and DQS signals at the Stratix II or Stratix II GX device pins. The second set of waveforms in Figure 9–17 shows what happens if the shifted DQS signal is not inverted; the last data, D_n, does not get latched into the logic array as DQS goes to tristate after the read postamble time. The third set of waveforms in Figure 9–17 shows a proper read operation with the DQS signal inverted after the 90° shift; the last data, D_n, does get latched. In this case the outputs of register A_I and latch C_I, which correspond to `dataout_h` and `dataout_l` ports, are now switched because of the DQS inversion. Register A_I, register B_I, and latch C_I refer to the nomenclature in Figure 9–15.

Figure 9–17. DQ Captures with Non-Inverted and Inverted Shifted DQS



PLL

When using the Stratix II and Stratix II GX top and bottom I/O banks (I/O banks 3, 4, 7, or 8) to interface with a DDR memory, at least one PLL with two outputs is needed to generate the system clock and the write clock. The system clock generates the DQS write signals, commands, and addresses. The write clock is either shifted by –90° or 90° from the system clock and is used to generate the DQ signals during writes.

For DDR and DDR2 SDRAM interfaces above 200 MHz, Altera also recommends a second read PLL to help ease resynchronization.

When using the Stratix II and Stratix II GX side I/O banks 1, 2, 5, or 6 to interface with DDR SDRAM devices, two PLLs may be needed per I/O bank for best performance. Since the side I/O banks do not have dedicated circuitry, one PLL captures data from the DDR SDRAM and another PLL generates the write signals, commands, and addresses to the DDR SDRAM device. Stratix II and Stratix II GX side I/O banks can support DDR SDRAM up to 150 MHz.

Enhancements In Stratix II and Stratix II GX Devices

Stratix II and Stratix II GX external memory interfaces support differs from Stratix external memory interfaces support in the following ways:

- A PLL output can now be used as the input reference clock to the DLL.
- The shifted DQS signal can now go into the logic array.
- The DLL in Stratix II and Stratix II GX devices has more phase-shift options than in Stratix devices. It also has the option to add phase offset settings.
- Stratix II and Stratix II GX devices have DQS logic blocks with each DQS pin that helps with fine tuning the phase shift.
- The DQS delay settings can be routed from the DLL into the logic array. You can also bypass the DLL and send the DQS delay settings from the logic array to the DQS logic block.
- Stratix II and Stratix II GX devices support DQSn pins.
- The DQS/DQ groups now support ×4, ×9, ×18, and ×36 bus modes.
- The DQS pins have been enhanced with the DQS postamble circuitry.

Conclusion

Stratix II and Stratix II GX devices support SDR SDRAM, DDR SDRAM, DDR2 SDRAM, RLDRAM II, and QDRII SRAM external memories. Stratix II and Stratix II GX devices feature high-speed interfaces that transfer data between external memory devices at up to 300 MHz/600 Mbps. DQS phase-shift circuitry and DQS logic blocks within the Stratix II and Stratix II GX devices allow you to fine-tune the phase shifts for the input clocks or strobes to properly align clock edges as needed to capture data.

Referenced Documents

This chapter references the following documents:

- *AN 325: Interfacing RLDRAM II with Stratix II & Stratix GX Devices*
- *AN 326: Interfacing QDRII & QDRII+ SRAM with Stratix II, Stratix, & Stratix GX Devices*
- *AN 327: Interfacing DDR SDRAM with Stratix II Devices*
- *AN 328: Interfacing DDR2 SDRAM with Stratix II Devices*
- *DC & Switching Characteristics of Stratix III Devices* chapter in volume 2 of the *Stratix III Device Handbook*

Document Revision History

Table 9–11 shows the revision history for this chapter.

Table 9–11. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007, v4.5	Added the "Referenced Documents" section.	—
	Minor text edits.	
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 8. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
May 2007, v4.4	Updated the "Phase Offset Control" section.	—
	Updated Figure 9–2.	—
	Updated Table 9–1.	—
	Added Table 9–4 and Table 9–6.	—
	Updated *Note (1)* to Figure 9–10.	—
February 2007 v4.3	Added the "Document Revision History" section to this chapter.	—
April 2006, v4.2	Chapter updated as part of the *Stratix II Device Handbook* update.	—
No change	Formerly chapter 7. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
December 2005 v4.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—
October 2005 v4.0	Added chapter to the *Stratix II GX Device Handbook*.	—

Section IV. I/O Standards

This section provides information on Stratix® II GX single-ended, voltage-referenced, and differential I/O standards.

This section contains the following chapters:

- Chapter 10, Selectable I/O Standards in Stratix II and Stratix II GX Devices
- Chapter 11, High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

10. Selectable I/O Standards in Stratix II and Stratix II GX Devices

SII52004-4.6

Introduction

This chapter provides guidelines for using industry I/O standards in Stratix® II and Stratix II GX devices, including:

- I/O features
- I/O standards
- External memory interfaces
- I/O banks
- Design considerations

Stratix II and Stratix II GX I/O Features

Stratix II and the Stratix II GX devices contain an abundance of adaptive logic modules (ALMs), embedded memory, high-bandwidth digital signal processing (DSP) blocks, and extensive routing resources, all of which can operate at very high core speed.

Stratix II and Stratix II GX devices I/O structure is designed to ensure that these internal capabilities are fully utilized. There are numerous I/O features to assist in high-speed data transfer into and out of the device including:

- Single-ended, non-voltage-referenced and voltage-referenced I/O standards
- High-speed differential I/O standards featuring serializer/deserializer (SERDES), dynamic phase alignment (DPA), capable of 1 gigabit per second (Gbps) performance for low-voltage differential signaling (LVDS), Hypertransport technology, HSTL, SSTL, and LVPECL

  HSTL and SSTL I/O standards are used only for PLL clock inputs and outputs in differential mode. LVPECL is supported on clock input and outputs of the top and bottom I/O banks.

- Double data rate (DDR) I/O pins
- Programmable output drive strength for voltage-referenced and non-voltage-referenced single-ended I/O standards
- Programmable bus-hold
- Programmable pull-up resistor
- Open-drain output
- On-chip series termination
- On-chip parallel termination

- On-chip differential termination
- Peripheral component interconnect (PCI) clamping diode
- Hot socketing

For a detailed description of each I/O feature, refer to the *Stratix II Architecture* chapter in volume 1 of the *Stratix II Device Handbook* or the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Stratix II and Stratix II GX I/O Standards Support

Stratix II and Stratix II GX devices support a wide range of industry I/O standards. Table 10–1 shows which I/O standards Stratix II devices support as well as typical applications.

Table 10–1. Stratix II and Stratix II GX I/O Standard Applications (Part 1 of 2)

I/O Standard	Application
LVTTL	General purpose
LVCMOS	General purpose
2.5 V	General purpose
1.8 V	General purpose
1.5 V	General purpose
3.3-V PCI	PC and embedded system
3.3-V PCI-X	PC and embedded system
SSTL-2 Class I	DDR SDRAM
SSTL-2 Class II	DDR SDRAM
SSTL-18 Class I	DDR2 SDRAM
SSTL-18 Class II	DDR2 SDRAM
1.8-V HSTL Class I	QDRII SRAM/RLDRAM II/SRAM
1.8-V HSTL Class II	QDRII SRAM/RLDRAM II/SRAM
1.5-V HSTL Class I	QDRII SRAM/SRAM
1.5-V HSTL Class II	QDRII SRAM/SRAM
1.2-V HSTL	General purpose
Differential SSTL-2 Class I	DDR SDRAM
Differential SSTL-2 Class II	DDR SDRAM
Differential SSTL-18 Class I	DDR2 SDRAM
Differential SSTL-18 Class II	DDR2 SDRAM
1.8-V differential HSTL Class I	Clock interfaces
1.8-V differential HSTL Class II	Clock interfaces

Table 10–1. Stratix II and Stratix II GX I/O Standard Applications (Part 2 of 2)

I/O Standard	Application
1.5-V differential HSTL Class I	Clock interfaces
1.5-V differential HSTL Class II	Clock interfaces
LVDS	High-speed communications
HyperTransport™ technology	PCB interfaces
Differential LVPECL	Video graphics and clock distribution

Single-Ended I/O Standards

In non-voltage-referenced single-ended I/O standards, the voltage at the input must be above a set voltage to be considered "on" (high, or logic value 1) or below another voltage to be considered "off" (low, or logic value 0). Voltages between the limits are undefined logically, and may fall into either a logic value 0 or 1. The non-voltage-referenced single-ended I/O standards supported by Stratix II and Stratix II GX devices are:

- Low-voltage transistor-transistor logic (LVTTL)
- Low-voltage complementary metal-oxide semiconductor (LVCMOS)
- 1.5 V
- 1.8 V
- 2.5 V
- 3.3-V PCI
- 3.3-V PCI-X

Voltage-referenced, single-ended I/O standards provide faster data rates. These standards use a constant reference voltage at the input levels. The incoming signals are compared with this constant voltage and the difference between the two defines "on" and "off" states.

Stratix II and Stratix II GX devices support stub series terminated logic (SSTL) and high-speed transceiver logic (HSTL) voltage-referenced I/O standards.

LVTTL

The LVTTL standard is formulated under EIA/JEDEC Standard, JESD8-B (Revision of JESD8-A): Interface Standard for Nominal 3-V/3.3-V Supply Digital Integrated Circuits.

The standard defines DC interface parameters for digital circuits operating from a 3.0- or 3.3-V power supply and driving or being driven by LVTTL-compatible devices. The 3.3-V LVTTL standard is a

general-purpose, single-ended standard used for 3.3-V applications. This I/O standard does not require input reference voltages (V_{REF}) or termination voltages (V_{TT}).

Stratix II and Stratix II GX devices support both input and output levels for 3.3-V LVTTL operation.

Stratix II Stratix II GX devices support a V_{CCIO} voltage level of 3.3 V ±5% as specified as the narrow range for the voltage supply by the EIA/JEDEC standard.

LVCMOS

The LVCMOS standard is formulated under EIA/JEDEC Standard, JESD8-B (Revision of JESD8-A): Interface Standard for Nominal 3-V/3.3-V Supply Digital Integrated Circuits.

The standard defines DC interface parameters for digital circuits operating from a 3.0- or 3.3-V power supply and driving or being driven by LVCMOS-compatible devices. The 3.3-V LVCMOS I/O standard is a general-purpose, single-ended standard used for 3.3-V applications. While LVCMOS has its own output specification, it specifies the same input voltage requirements as LVTTL. These I/O standards do not require V_{REF} or V_{TT}.

Stratix II and Stratix II GX devices support both input and output levels for 3.3-V LVCMOS operation.

Stratix II and Stratix II GX devices support a V_{CCIO} voltage level of 3.3 V ±5% as specified as the narrow range for the voltage supply by the EIA/JEDEC standard.

2.5 V

The 2.5-V I/O standard is formulated under EIA/JEDEC Standard, EIA/JESD8-5: 2.5-V± 0.2-V (Normal Range), and 1.8-V – 2.7-V (Wide Range) Power Supply Voltage and Interface Standard for Non-Terminated Digital Integrated Circuit.

The standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 2.5-V devices. This standard is a general-purpose, single-ended standard used for 2.5-V applications. It does not require the use of a V_{REF} or a V_{TT}.

Stratix II and Stratix II GX devices support both input and output levels for 2.5-V operation with V_{CCIO} voltage level support of 2.5 V ± 5%, which is narrower than defined in the Normal Range of the EIA/JEDEC standard.

1.8 V

The 1.8-V I/O standard is formulated under EIA/JEDEC Standard, EIA/JESD8-7: 1.8-V± 0.15-V (Normal Range), and 1.2-V – 1.95-V (Wide Range) Power Supply Voltage and Interface Standard for Non-Terminated Digital Integrated Circuit.

The standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 1.8-V devices. This standard is a general-purpose, single-ended standard used for 1.8-V applications. It does not require the use of a V_{REF} or a V_{TT}.

Stratix II and Stratix II GX devices support both input and output levels for 1.8-V operation with V_{CCIO} voltage level support of 1.8 V ± 5%, which is narrower than defined in the Normal Range of the EIA/JEDEC standard.

1.5 V

The 1.5-V I/O standard is formulated under EIA/JEDEC Standard, JESD8-11: 1.5-V± 0.1-V (Normal Range) and 0.9-V – 1.6-V (Wide Range) Power Supply Voltage and Interface Standard for Non-Terminated Digital Integrated Circuit.

The standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 1.5-V devices. This standard is a general-purpose, single-ended standard used for 1.5-V applications. It does not require the use of a V_{REF} or a V_{TT}.

Stratix II and Stratix II GX devices support both input and output levels for 1.5-V operation V_{CCIO} voltage level support of 1.5 V ± 5%, which is narrower than defined in the Normal Range of the EIA/JEDEC standard.

3.3-V PCI

The 3.3-V PCI I/O standard is formulated under PCI Local Bus Specification Revision 2.2 developed by the PCI Special Interest Group (SIG).

The PCI local bus specification is used for applications that interface to the PCI local bus, which provides a processor-independent data path between highly integrated peripheral controller components, peripheral add-in boards, and processor/memory systems. The conventional PCI specification revision 2.2 defines the PCI hardware environment including the protocol, electrical, mechanical, and configuration specifications for the PCI devices and expansion boards. This standard requires 3.3-V $V_{CCIO.}$ Stratix II and Stratix II GX devices are fully compliant with the 3.3-V PCI Local Bus Specification Revision 2.2 and meet 64-bit/66-MHz operating frequency and timing requirements.

The 3.3-V PCI standard does not require input reference voltages or board terminations. Stratix II and Stratix II GX devices support both input and output levels.

3.3-V PCI-X

The 3.3-V PCI-X I/O standard is formulated under PCI-X Local Bus Specification Revision 1.0a developed by the PCI SIG.

The PCI-X 1.0 standard is used for applications that interface to the PCI local bus. The standard enables the design of systems and devices that operate at clock speeds up to 133 MHz, or 1 Gbps for a 64-bit bus. The PCI-X 1.0 protocol enhancements enable devices to operate much more efficiently, providing more usable bandwidth at any clock frequency. By using the PCI-X 1.0 standard, you can design devices to meet PCI-X 1.0 requirements and operate as conventional 33- and 66-MHz PCI devices when installed in those systems. This standard requires 3.3-V V_{CCIO}. Stratix II and Stratix II GX devices are fully compliant with the 3.3-V PCI-X Specification Revision 1.0a and meet the 133-MHz operating frequency and timing requirements. The 3.3-V PCI-X standard does not require input reference voltages or board terminations.

Stratix II and Stratix II GX devices support both input and output levels operation.

SSTL-2 Class I and SSTL-2 Class II

The 2.5-V SSTL-2 standard is formulated under JEDEC Standard, JESD8-9A: Stub Series Terminated Logic for 2.5-V (SSTL_2).

The SSTL-2 I/O standard is a 2.5-V memory bus standard used for applications such as high-speed DDR SDRAM interfaces. This standard defines the input and output specifications for devices that operate in the SSTL-2 logic switching range of 0.0 to 2.5 V. This standard improves

operation in conditions where a bus must be isolated from large stubs. SSTL-2 requires a 1.25-V V_{REF} and a 1.25-V V_{TT} to which the series and termination resistors are connected (Figures 10–1 and 10–2).

Stratix II and Stratix II GX devices support both input and output levels operation.

Figure 10–1. 2.5-V SSTL Class I Termination

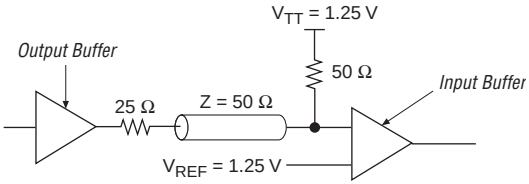
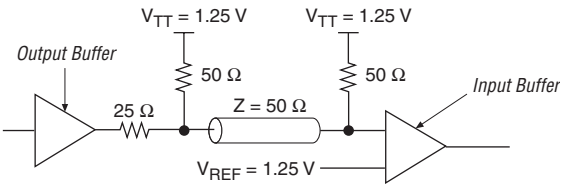


Figure 10–2. 2.5-V SSTL Class II Termination



SSTL-18 Class I and SSTL-18 Class II

The 1.8-V SSTL-18 standard is formulated under JEDEC Standard, JESD8-15: Stub Series Terminated Logic for 1.8-V (SSTL_18).

The SSTL-18 I/O standard is a 1.8-V memory bus standard used for applications such as high-speed DDR2 SDRAM interfaces. This standard is similar to SSTL-2 and defines input and output specifications for devices that are designed to operate in the SSTL-18 logic switching range 0.0 to 1.8 V. SSTL-18 requires a 0.9-V V_{REF} and a 0.9-V V_{TT} to which the series and termination resistors are connected.

There are no class definitions for the SSTL-18 standard in the JEDEC specification. The specification of this I/O standard is based on an environment that consists of both series and parallel terminating resistors. Altera provides solutions to two derived applications in JEDEC specification, and names them Class I and Class II to be consistent with other SSTL standards. Figures 10–3 and 10–4 show SSTL-18 Class I and II termination, respectively.

Stratix II and Stratix II GX devices support both input and output levels operation.

Figure 10–3. 1.8-V SSTL Class I Termination

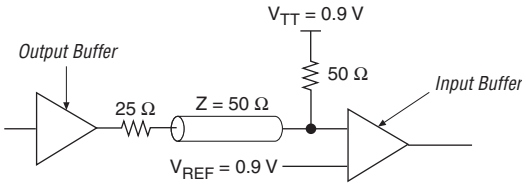
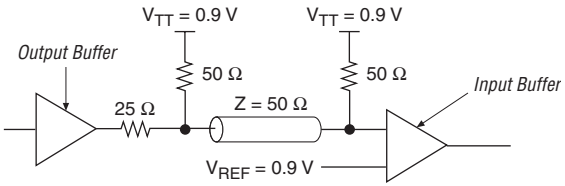


Figure 10–4. 1.8-V SSTL Class II Termination



1.8-V HSTL Class I and 1.8-V HSTL Class II

The HSTL standard is a technology-independent I/O standard developed by JEDEC to provide voltage scalability. It is used for applications designed to operate in the 0.0- to 1.8-V HSTL logic switching range such as quad data rate (QDR) memory clock interfaces.

Although JEDEC specifies a maximum V_{CCIO} value of 1.6 V, there are various memory chip vendors with HSTL standards that require a V_{CCIO} of 1.8 V. Stratix II and Stratix II GX devices support interfaces to chips with V_{CCIO} of 1.8 V for HSTL. Figures 10–5 and 10–6 show the nominal V_{REF} and V_{TT} required to track the higher value of V_{CCIO}. The value of V_{REF} is selected to provide optimum noise margin in the system.

Stratix II and Stratix II GX devices support both input and output levels operation.

Figure 10–5. 1.8-V HSTL Class I Termination

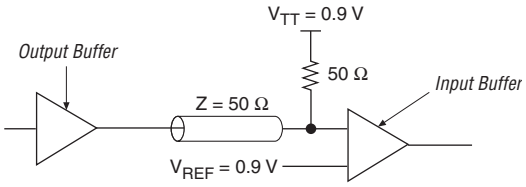
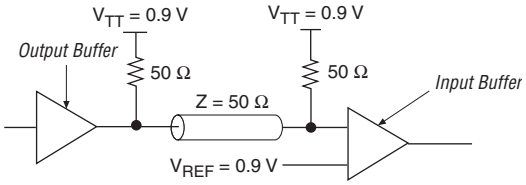


Figure 10–6. 1.8-V HSTL Class II Termination



1.5-V HSTL Class I and 1.5-V HSTL Class II

The 1.5-V HSTL standard is formulated under EIA/JEDEC Standard, EIA/JESD8-6: A 1.5-V Output Buffer Supply Voltage Based Interface Standard for Digital Integrated Circuits.

The 1.5-V HSTL I/O standard is used for applications designed to operate in the 0.0- to 1.5-V HSTL logic nominal switching range. This standard defines single-ended input and output specifications for all HSTL-compliant digital integrated circuits. The 1.5-V HSTL I/O standard in Stratix II and Stratix II GX devices are compatible with the 1.8-V HSTL I/O standard in APEX™ 20KE, APEX 20KC, and in Stratix II and Stratix II GX devices themselves because the input and output voltage thresholds are compatible (Figures 10–7 and 10–8).

☞ Stratix II and Stratix II GX devices support both input and output levels with V_{REF} and V_{TT}.

Figure 10–7. 1.5-V HSTL Class I Termination

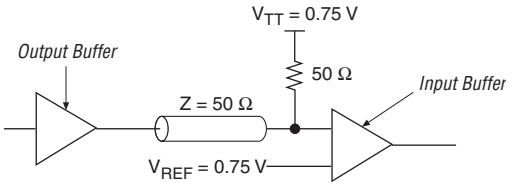
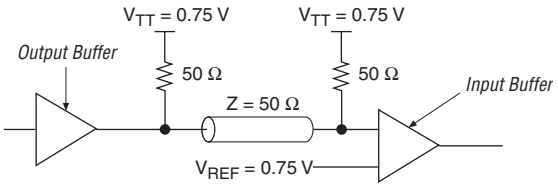


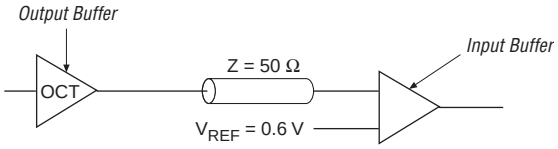
Figure 10–8. 1.5-V HSTL Class II Termination



1.2-V HSTL

Although there is no EIA/JEDEC standard available for the 1.2-V HSTL standard, Altera supports it for applications that operate in the 0.0 to 1.2-V HSTL logic nominal switching range. 1.2-V HSTL can be terminated through series or parallel on-chip termination (OCT). Figure 10–9 shows the termination scheme.

Figure 10–9. 1.2-V HSTL Termination



Differential I/O Standards

Differential I/O standards are used to achieve even faster data rates with higher noise immunity. Apart from LVDS, LVPECL, and HyperTransport technology, Stratix II and Stratix II GX devices also support differential versions of SSTL and HSTL standards.

For detailed information on differential I/O standards, refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Differential SSTL-2 Class I and Differential SSTL-2 Class II

The 2.5-V differential SSTL-2 standard is formulated under JEDEC Standard, JESD8-9A: Stub Series Terminated Logic for 2.5-V (SSTL_2).

This I/O standard is a 2.5-V standard used for applications such as high-speed DDR SDRAM clock interfaces. This standard supports differential signals in systems using the SSTL-2 standard and supplements the SSTL-2 standard for differential clocks. Stratix II and Stratix II GX devices support both input and output levels. Figures 10–10 and 10–11 shows details on differential SSTL-2 termination.

Stratix II and Stratix II GX devices support differential SSTL-2 I/O standards in pseudo-differential mode, which is implemented by using two SSTL-2 single-ended buffers.

The Quartus® II software only supports pseudo-differential standards on the `INCLK`, `FBIN` and `EXTCLK` ports of enhanced PLL, as well as on DQS pins when DQS megafunction (ALTDQS, Bidirectional Data Strobe) is used. Two single-ended output buffers are automatically programmed to have opposite polarity so as to implement a pseudo-differential output. A proper V_{REF} voltage is required for the two single-ended input buffers to implement a pseudo-differential input. In this case, only the positive polarity input is used in the speed path while the negative input is not connected internally. In other words, only the non-inverted pin is required to be specified in your design, while the Quartus II software automatically generates the inverted pin for you.

Although the Quartus II software does not support pseudo-differential SSTL-2 I/O standards on the left and right I/O banks, you can implement these standards at these banks. You need to create two pins in the designs and configure the pins with single-ended SSTL-2 standards. However, this is limited only to pins that support the differential pin-pair I/O function and is dependent on the single-ended SSTL-2 standards support at these banks.

Figure 10–10. Differential SSTL-2 Class I Termination

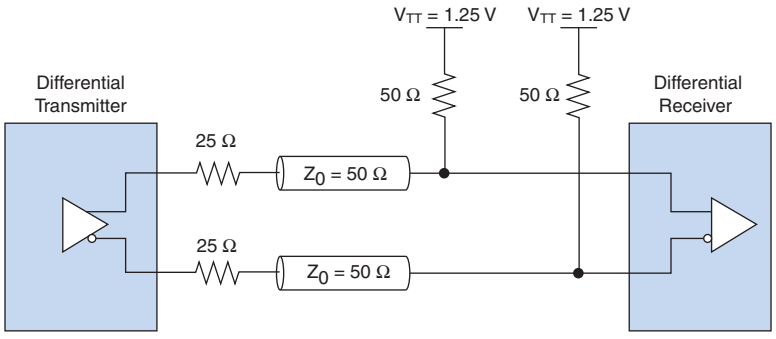
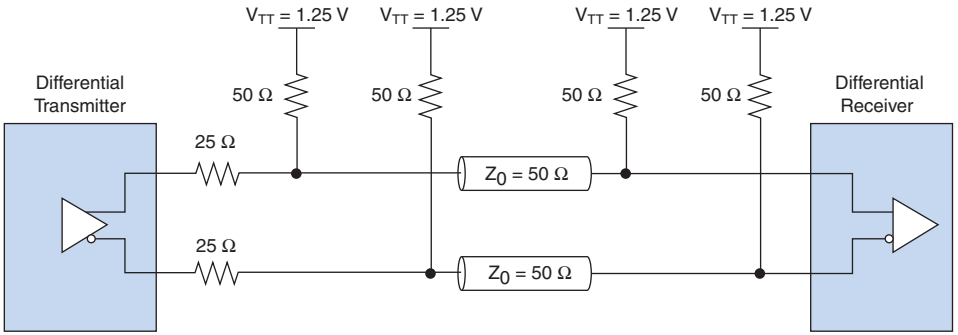


Figure 10–11. Differential SSTL-2 Class II Termination



Differential SSTL-18 Class I and Differential SSTL-18 Class II

The 1.8-V differential SSTL-18 standard is formulated under JEDEC Standard, JESD8-15: Stub Series Terminated Logic for 1.8-V (SSTL_18).

The differential SSTL-18 I/O standard is a 1.8-V standard used for applications such as high-speed DDR2 SDRAM interfaces. This standard supports differential signals in systems using the SSTL-18 standard and supplements the SSTL-18 standard for differential clocks.

Stratix II and Stratix II GX devices support both input and output levels operation.

Figures 10–12 and 10–13 shows details on differential SSTL-18 termination. Stratix II and Stratix II GX devices support differential SSTL-18 I/O standards in pseudo-differential mode, which is implemented by using two SSTL-18 single-ended buffers.

The Quartus II software only supports pseudo-differential standards on the INCLK, FBIN and EXTCLK ports of enhanced PLL, as well as on DQS pins when DQS megafunction (ALTDQS, Bidirectional Data Strobe) is used. Two single-ended output buffers are automatically programmed to have opposite polarity so as to implement a pseudo-differential output. A proper V_{REF} voltage is required for the two single-ended input buffers to implement a pseudo-differential input. In this case, only the positive polarity input is used in the speed path while the negative input is not connected internally. In other words, only the non-inverted pin is required to be specified in your design, while the Quartus II software automatically generates the inverted pin for you.

Although the Quartus II software does not support pseudo-differential SSTL-18 I/O standards on the left and right I/O banks, you can implement these standards at these banks. You need to create two pins in the designs and configure the pins with single-ended SSTL-18 standards. However, this is limited only to pins that support the differential pin-pair I/O function and is dependent on the single-ended SSTL-18 standards support at these banks.

Figure 10–12. Differential SSTL-18 Class I Termination

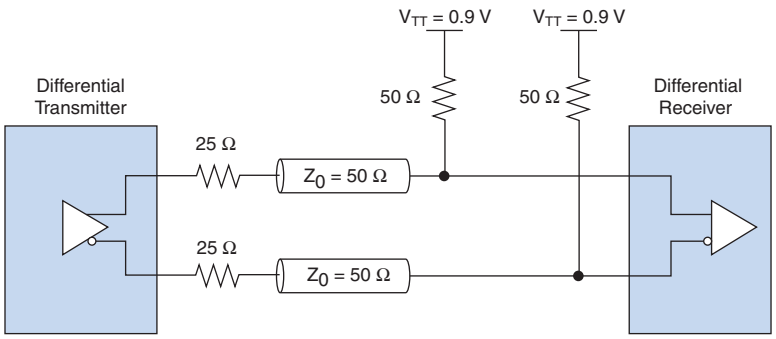
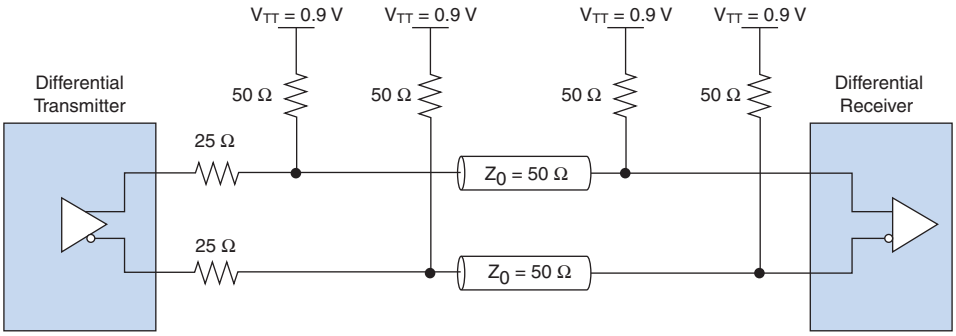


Figure 10–13. Differential SSTL-18 Class II Termination



1.8-V Differential HSTL Class I and 1.8-V Differential HSTL Class II

The 1.8-V differential HSTL specification is the same as the 1.8-V single-ended HSTL specification. It is used for applications designed to operate in the 0.0- to 1.8-V HSTL logic switching range such as QDR memory clock interfaces. Stratix II and Stratix II GX devices support both input and output levels operation. Figures 10–14 and 10–15 show details on 1.8-V differential HSTL termination.

Stratix II and Stratix II GX devices support 1.8-V differential HSTL I/O standards in pseudo-differential mode, which is implemented by using two 1.8-V HSTL single-ended buffers.

The Quartus II software only supports pseudo-differential standards on the `INCLK`, `FBIN` and `EXTCLK` ports of enhanced PLL, as well as on DQS pins when DQS megafunction (ALTDQS, Bidirectional Data Strobe) is used. Two single-ended output buffers are automatically programmed to have opposite polarity so as to implement a pseudo-differential output. A proper V_{REF} voltage is required for the two single-ended input buffers to implement a pseudo-differential input. In this case, only the positive polarity input is used in the speed path while the negative input is not connected internally. In other words, only the non-inverted pin is required to be specified in your design, while the Quartus II software automatically generates the inverted pin for you.

Although the Quartus II software does not support 1.8-V pseudo-differential HSTL I/O standards on left/right I/O banks, you can implement these standards at these banks. You need to create two pins in the designs and configure the pins with single-ended 1.8-V HSTL standards. However, this is limited only to pins that support the differential pin-pair I/O function and is dependent on the single-ended 1.8-V HSTL standards support at these banks.

Figure 10–14. 1.8-V Differential HSTL Class I Termination

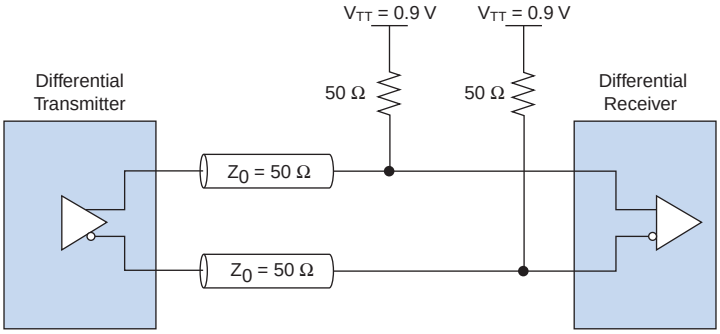
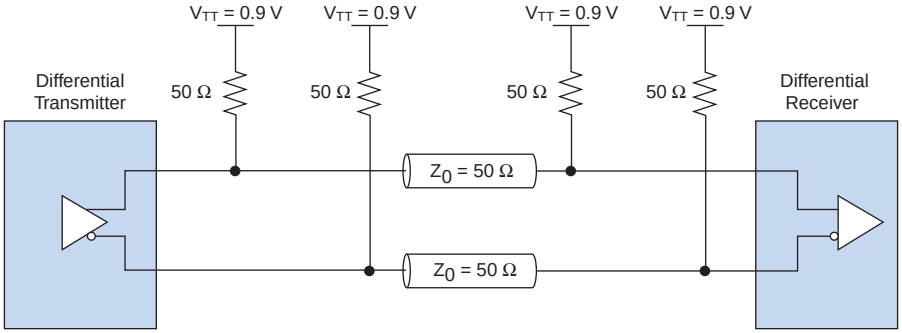


Figure 10–15. 1.8-V Differential HSTL Class II Termination



1.5-V Differential HSTL Class I and 1.5-V Differential HSTL Class II

The 1.5-V differential HSTL standard is formulated under EIA/JEDEC Standard, EIA/JESD8-6: A 1.5-V Output Buffer Supply Voltage Based Interface Standard for Digital Integrated Circuits.

The 1.5-V differential HSTL specification is the same as the 1.5-V single-ended HSTL specification. It is used for applications designed to operate in the 0.0- to 1.5-V HSTL logic switching range, such as QDR memory clock interfaces. Stratix II and Stratix II GX devices support both input and output levels operation. Figures 10–16 and 10–17 show details on the 1.5-V differential HSTL termination.

Stratix II and Stratix II GX devices support 1.5-V differential HSTL I/O standards in pseudo-differential mode, which is implemented by using two 1.5-V HSTL single-ended buffers.

The Quartus II software only supports pseudo-differential standards on the INCLK, FBIN and EXTCLK ports of enhanced PLL, as well as on DQS pins when DQS megafunction (ALTDQS, Bidirectional Data Strobe) is used. Two single-ended output buffers are automatically programmed to have opposite polarity so as to implement a pseudo-differential output. A proper V_{REF} voltage is required for the two single-ended input buffers to implement a pseudo-differential input. In this case, only the positive polarity input is used in the speed path while the negative input is not connected internally. In other words, only the non-inverted pin is required to be specified in your design, while the Quartus II software automatically generates the inverted pin for you.

Although the Quartus II software does not support 1.5-V pseudo-differential HSTL I/O standards on left/right I/O banks, you can implement these standards at these banks. You need to create two pins in the designs and configure the pins with single-ended 1.5-V HSTL standards. However, this is limited only to pins that support the differential pin-pair I/O function and is dependent on the single-ended 1.8-V HSTL standards support at these banks.

Figure 10–16. 1.5-V Differential HSTL Class I Termination

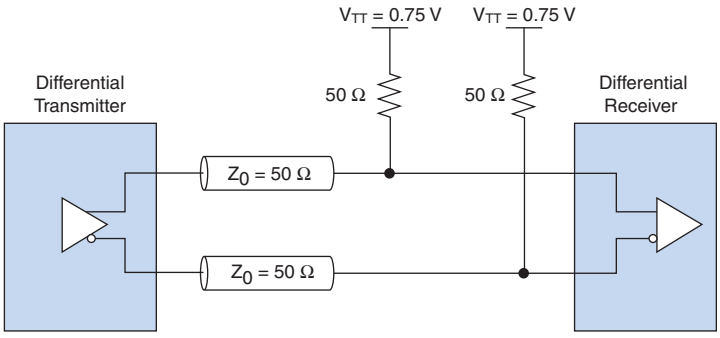
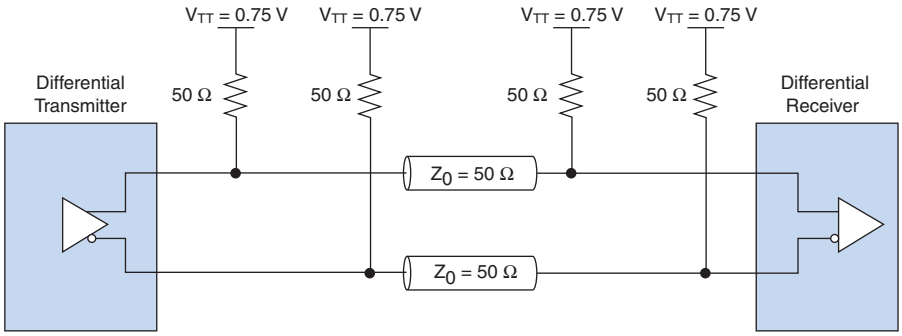


Figure 10–17. 1.5-V Differential HSTL Class II Termination



LVDS

The LVDS standard is formulated under ANSI/TIA/EIA Standard, ANSI/TIA/EIA-644: Electrical Characteristics of Low Voltage Differential Signaling Interface Circuits.

The LVDS I/O standard is a differential high-speed, low-voltage swing, low-power, general-purpose I/O interface standard. In Stratix II devices, the LVDS I/O standard requires a 2.5-V V_{CCIO} level for the side I/O pins in banks 1, 2, 5, and 6. The top and bottom banks have different V_{CCIO} requirements for the LVDS I/O standard. The LVDS clock I/O pins in banks 9 through 12 require a 3.3-V V_{CCIO} level. Within these banks, the `PLL[5,6,11,12]_OUT[1,2]` pins support output only LVDS operations. The `PLL[5,6,11,12]_FB/OUT2` pins support LVDS input or output operations but cannot be configured for bidirectional LVDS operations. The LVDS clock input pins in banks 4, 5, 7, and 8 use V_{CCINT} and have no dependency on the V_{CCIO} voltage level. This standard is used in applications requiring high-bandwidth data transfer, backplane drivers, and clock distribution. The ANSI/TIA/EIA-644 standard specifies LVDS transmitters and receivers capable of operating at recommended maximum data signaling rates of 655 megabit per second (Mbps). However, devices can operate at slower speeds if needed, and there is a theoretical maximum of 1.923 Gbps. Stratix II and Stratix II GX devices are capable of running at a maximum data rate of 1 Gbps and still meet the ANSI/TIA/EIA-644 standard.

Because of the low-voltage swing of the LVDS I/O standard, the electromagnetic interference (EMI) effects are much smaller than complementary metal-oxide semiconductor (CMOS), transistor-to-transistor logic (TTL), and positive (or psuedo) emitter coupled logic (PECL). This low EMI makes LVDS ideal for applications

with low EMI requirements or noise immunity requirements. The LVDS standard does not require an input reference voltage. However, it does require a 100-Ω termination resistor between the two signals at the input buffer. Stratix II and Stratix II GX devices provide an optional 100-Ω differential LVDS termination resistor in the device using on-chip differential termination. Stratix II and Stratix II GX devices support both input and output levels operation.

Differential LVPECL

The low-voltage positive (or pseudo) emitter coupled logic (LVPECL) standard is a differential interface standard requiring a 3.3-V V_{CCIO}. The standard is used in applications involving video graphics, telecommunications, data communications, and clock distribution. The high-speed, low-voltage swing LVPECL I/O standard uses a positive power supply and is similar to LVDS. However, LVPECL has a larger differential output voltage swing than LVDS. The LVPECL standard does not require an input reference voltage, but it does require a 100-Ω termination resistor between the two signals at the input buffer. Figures 10–18 and 10–19 show two alternate termination schemes for LVPECL.

Stratix II and Stratix II GX devices support both input and output levels operation.

Figure 10–18. LVPECL DC Coupled Termination

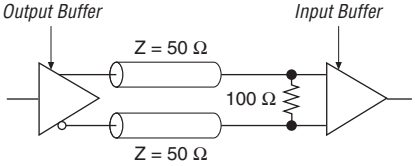
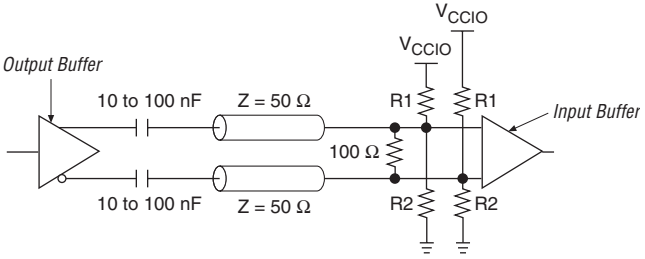


Figure 10–19. LVPECL AC Coupled Termination



HyperTransport Technology

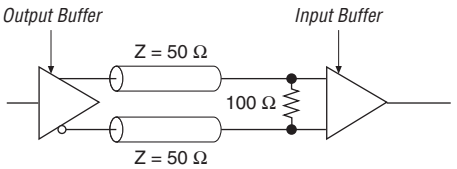
The HyperTransport standard is formulated by the HyperTransport Consortium.

The HyperTransport I/O standard is a differential high-speed, high-performance I/O interface standard requiring a 2.5- or 3.3-V V_{CCIO}. This standard is used in applications such as high-performance networking, telecommunications, embedded systems, consumer electronics, and Internet connectivity devices. The HyperTransport I/O standard is a point-to-point standard in which each HyperTransport bus consists of two point-to-point unidirectional links. Each link is 2 to 32 bits.

The HyperTransport standard does not require an input reference voltage. However, it does require a 100-Ω termination resistor between the two signals at the input buffer. Figure 10–20 shows HyperTransport termination. Stratix II and Stratix II GX devices include an optional 100-Ω differential HyperTransport termination resistor in the device using on-chip differential termination.

Stratix II and Stratix II GX devices support both input and output levels operation.

Figure 10–20. HyperTransport Termination



Stratix II and Stratix II GX External Memory Interface

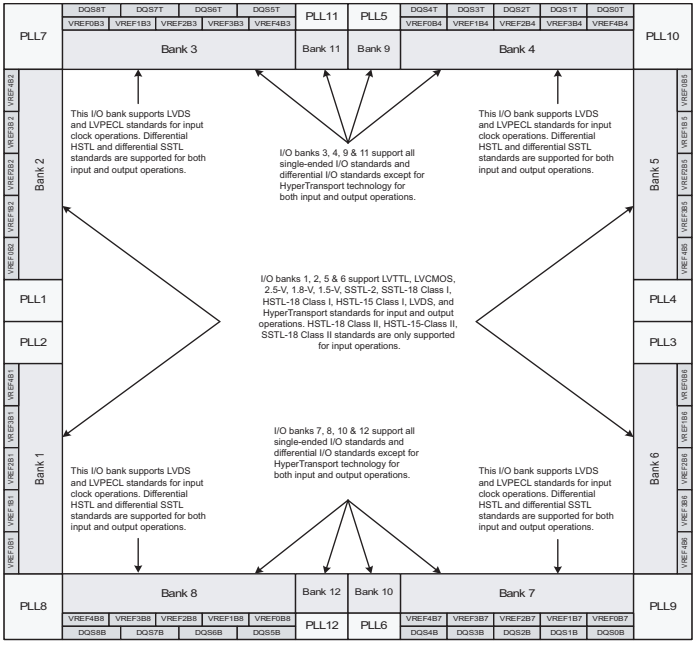
The increasing demand for higher-performance data processing systems often requires memory-intensive applications. Stratix II and Stratix II GX devices can interface with many types of external memory.

Refer to the *External Memory Interfaces in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *External Memory Interfaces in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook* for more information on the external memory interface support in Stratix II or Stratix II GX devices.

Stratix II and Stratix II GX I/O Banks

Stratix II devices have eight general I/O banks and four enhanced phase-locked loop (PLL) external clock output banks (Figure 10–21). I/O banks 1, 2, 5, and 6 are on the left or right sides of the device and I/O banks 3, 4, and 7 through 12 are at the top or bottom of the device.

Figure 10–21. Stratix II I/O Banks *Notes (1), (2), (3), (4), (5), (6), (7)*



Notes to Figure 10–21:

(1) Figure 10–21 is a top view of the silicon die that corresponds to a reverse view for flip-chip packages. It is a graphical representation only. Refer to the pin list and Quartus II software for exact locations.

(2) Depending on the size of the device, different device members have different numbers of V_{REF} groups.

(3) Banks 9 through 12 are enhanced PLL external clock output banks. These PLL banks utilize the adjacent V_{REF} group when voltage-referenced standards are implemented. For example, if an SSTL input is implemented in PLL bank 10, the voltage level at VREFB7 is the reference voltage level for the SSTL input.

(4) Differential HSTL and differential SSTL standards are available for bidirectional operations on DQS pin and input-only operations on PLL clock input pins; LVDS, LVPECL, and HyperTransport standards are available for input-only operations on PLL clock input pins. Refer to the "Differential I/O Standards" on page 10–10 for more details.

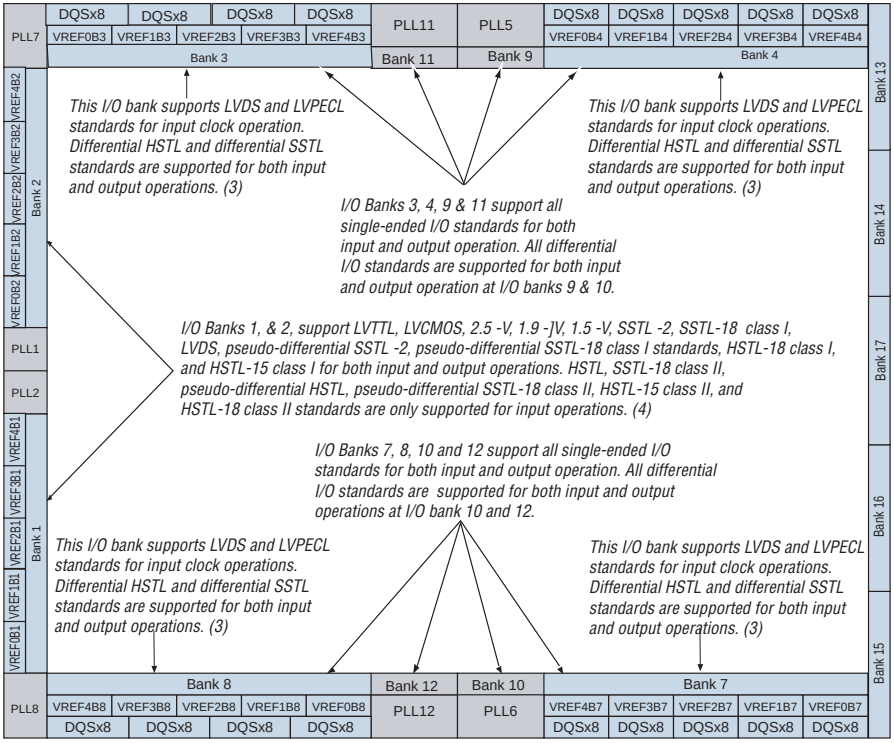
(5) Quartus II software does not support differential SSTL and differential HSTL standards at left/right I/O banks. Refer to the "Differential I/O Standards" on page 10–10 if you need to implement these standards at these I/O banks.

(6) Banks 11 and 12 are available only in EP2S60, EP2S90, EP2S130, and EP2S180 devices.

(7) PLLs 7, 8, 9 10, 11, and 12 are available only in EP2S60, EP2S90, EP2S130, and EP2S180 devices.

Stratix II GX devices have 6 general I/O banks and 4 enhanced phase-locked loop (PLL) external clock output banks (Figure 10–22). I/O banks 9 through 12 are enhanced PLL external clock output banks located on the top and bottom of the device.

Figure 10–22. Stratix II GX I/O Banks *Notes (1), (2), (3), (4)*



Notes to Figure 10–22:

(1) Figure 10–22 is a top view of the silicon die which corresponds to a reverse view for flip-chip packages. It is a graphical representation only.

(2) Depending on size of the device, different device members have different number of V_{REF} groups. Refer to the pin list and the Quartus II software for exact locations.

(3) Banks 9 through 12 are enhanced PLL external clock output banks.

(4) Horizontal I/O banks feature transceiver and DPA circuitry for high speed differential I/O standards. Refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*, or the *Stratix II GX Transceiver User Guide* (volume 1) of the *Stratix II GX Device Handbook* for more information on differential I/O standards.

(5) Quartus II software does not support differential SSTL and differential HSTL standards at left/right I/O banks. Refer to the "Differential I/O Standards" on page 10–10 if you need to implement these standards at these I/O banks.

(6) Banks 11 and 12 are available only in EP2SGX60C/D/E, EP2SGX90E/F, and EP2SGX130G.

(7) PLLs 7,8,11, and 12 are available only in EP2SGX60C/D/E, EP2SGXE/F, and EP2SGX130G.

Programmable I/O Standards

Stratix II and Stratix II GX device programmable I/O standards deliver high-speed and high-performance solutions in many complex design systems. This section discusses the I/O standard support in the I/O banks of Stratix II and Stratix II GX devices.

Regular I/O Pins

Most Stratix II and Stratix II GX device pins are multi-function pins. These pins support regular inputs and outputs as their primary function, and offer an optional function such as DQS, differential pin-pair, or PLL external clock outputs. For example, you can configure a multi-function pin in the enhanced PLL external clock output bank as a PLL external clock output when it is not used as a regular I/O pin.

I/O pins that reside in PLL banks 9 through 12 are powered by the VCC_PLL<*5*, *6*, *11*, *or 12*>_OUT pins, respectively. The EP2S60F484, EP2S60F780, EP2S90H484, EP2S90F780, and EP2S130F780 devices do not support PLLs 11 and 12. Therefore, any I/O pins that reside in bank 11 are powered by the VCCIO3 pin, and any I/O pins that reside in bank 12 are powered by the VCCIO8 pin.

Table 10–2 shows the I/O standards supported when a pin is used as a regular I/O pin in the I/O banks of Stratix II and Stratix II GX devices.

Table 10–2. Stratix II and Stratix II GX Regular I/O Standards Support (Part 1 of 2)

I/O Standard	General I/O Bank								Enhanced PLL External Clock Output Bank *(2)*			
	1	2	3	4	5*(1)*	6*(1)*	7	8	9	10	11	12
LVTTL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LVCMOS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2.5 V	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1.8 V	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1.5 V	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3.3-V PCI			✓	✓			✓	✓	✓	✓	✓	✓
3.3-V PCI-X			✓	✓			✓	✓	✓	✓	✓	✓
SSTL-2 Class I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SSTL-2 Class II	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 10–2. Stratix II and Stratix II GX Regular I/O Standards Support (Part 2 of 2)

I/O Standard	General I/O Bank								Enhanced PLL External Clock Output Bank *(2)*			
	1	2	3	4	5*(1)*	6*(1)*	7	8	9	10	11	12
SSTL-18 Class I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SSTL-18 Class II	*(3)*	*(3)*	✓	✓	*(3)*	*(3)*	✓	✓	✓	✓	✓	✓
1.8-V HSTL Class I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1.8-V HSTL Class II	*(3)*	*(3)*	✓	✓	*(3)*	*(3)*	✓	✓	✓	✓	✓	✓
1.5-V HSTL Class I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1.5-V HSTL Class II	*(3)*	*(3)*	✓	✓	*(3)*	*(3)*	✓	✓	✓	✓	✓	✓
1.2-V HSTL				✓			✓	✓				
Differential SSTL-2 Class I	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
Differential SSTL-2 Class II	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
Differential SSTL-18 Class I	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
Differential SSTL-18 Class II	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
1.8-V differential HSTL Class I	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
1.8-V differential HSTL Class II	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
1.5-V differential HSTL Class I	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
1.5-V differential HSTL Class II	*(4)*	*(4)*	*(5)*	*(5)*	*(4)*	*(4)*	*(5)*	*(5)*				
LVDS	✓	✓	*(6)*	*(6)*	✓	✓	*(6)*	*(6)*	✓	✓	✓	✓
HyperTransport technology	✓	✓			✓	✓						
Differential LVPECL			*(6)*	*(6)*			*(6)*	*(6)*	✓	✓	✓	✓

Notes to Table 10–2:

(1) This bank is not available in Stratix II GX Devices.

(2) A mixture of single-ended and differential I/O standards is not allowed in enhanced PLL external clock output bank.

(3) This I/O standard is only supported for the input operation in this I/O bank.

(4) Although the Quartus II software does not support pseudo-differential SSTL/HSTL I/O standards on the left and right I/O banks, you can implement these standards at these banks. Refer to the "Differential I/O Standards" on page 10–10 for details.

(5) This I/O standard is supported for both input and output operations for pins that support the DQS function. Refer to the "Differential I/O Standards" on page 10–10 for details.

(6) This I/O standard is only supported for the input operation for pins that support PLL `INCLK` function in this I/O bank.

Clock I/O Pins

The PLL clock I/O pins consist of clock inputs (INCLK), external feedback inputs (FBIN), and external clock outputs (EXTCLK). Clock inputs are located at the left and right I/O banks (banks 1, 2, 5, and 6) to support fast PLLs, and at the top and bottom I/O banks (banks 3, 4, 7, and 8) to support enhanced PLLs. Both external clock outputs and external feedback inputs are located at enhanced PLL external clock output banks (banks 9, 10, 11, and 12) to support enhanced PLLs. Table 10–3 shows the PLL clock I/O support in the I/O banks of Stratix II and Stratix II GX devices.

Table 10–3. I/O Standards Supported for Stratix II and Stratix II GX PLL Pins (Part 1 of 2)

I/O Standard (2)	Enhanced PLL (1)			Fast PLL
	Input		Output	Input
	INCLK	FBIN	EXTCLK	INCLK
LVTTL	✓	✓	✓	✓
LVCMOS	✓	✓	✓	✓
2.5 V	✓	✓	✓	✓
1.8 V	✓	✓	✓	✓
1.5 V	✓	✓	✓	✓
3.3-V PCI	✓	✓	✓	
3.3-V PCI-X	✓	✓	✓	
SSTL-2 Class I	✓	✓	✓	✓
SSTL-2 Class II	✓	✓	✓	✓
SSTL-18 Class I	✓	✓	✓	✓
SSTL-18 Class II	✓	✓	✓	✓
1.8-V HSTL Class I	✓	✓	✓	✓
1.8-V HSTL Class II	✓	✓	✓	✓
1.5-V HSTL Class I	✓	✓	✓	✓
1.5-V HSTL Class II	✓	✓	✓	✓
Differential SSTL-2 Class I	✓	✓	✓	
Differential SSTL-2 Class II	✓	✓	✓	
Differential SSTL-18 Class I	✓	✓	✓	
Differential SSTL-18 Class II	✓	✓	✓	

Table 10–3. I/O Standards Supported for Stratix II and Stratix II GX PLL Pins (Part 2 of 2)

I/O Standard (2)	Enhanced PLL (1)			Fast PLL
	Input		Output	Input
	INCLK	FBIN	EXTCLK	INCLK
1.8-V differential HSTL Class I	✓	✓	✓	
1.8-V differential HSTL Class II	✓	✓	✓	
1.5-V differential HSTL Class I	✓	✓	✓	
1.5-V differential HSTL Class II	✓	✓	✓	
LVDS	✓	✓	✓	✓
HyperTransport technology				✓
Differential LVPECL	✓	✓	✓	

Note to Table 10–3:
(1) The enhanced PLL external clock output bank does not allow a mixture of both single-ended and differential I/O standards.
(2) Altera does not support 1.2-V HSTL for PLL input pins on column I/O pins.

For more information, refer to the *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *PLLs in Stratix II & Straix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook.*

Voltage Levels

Stratix II device specify a range of allowed voltage levels for supported I/O standards. Table 10–4 shows only typical values for input and output V_{CCIO}, V_{REF}, as well as the board V_{TT}.

Table 10–4. Stratix II and Stratix II GX I/O Standards and Voltage Levels (Part 1 of 3) Note (1)

I/O Standard	Stratix II and Stratix II GX					
	V_{CCIO} (V)				V_{REF} (V)	V_{TT} (V)
	Input Operation		Output Operation		Input	Termination
	Top and Bottom I/O Banks	Left and Right I/O Banks (3)	Top and Bottom I/O Banks	Left and Right I/O Banks (3)		
LVTTL	3.3/2.5	3.3/2.5	3.3	3.3	NA	NA
LVCMOS	3.3/2.5	3.3/2.5	3.3	3.3	NA	NA

Table 10–4. Stratix II and Stratix II GX I/O Standards and Voltage Levels (Part 2 of 3) *Note (1)*

I/O Standard	Stratix II and Stratix II GX					
	V_{CCIO} (V)				V_{REF} (V)	V_{TT} (V)
	Input Operation		Output Operation		Input	Termination
	Top and Bottom I/O Banks	Left and Right I/O Banks *(3)*	Top and Bottom I/O Banks	Left and Right I/O Banks *(3)*		
2.5 V	3.3/2.5	3.3/2.5	2.5	2.5	NA	NA
1.8 V	1.8/1.5	1.8/1.5	1.8	1.8	NA	NA
1.5 V	1.8/1.5	1.8/1.5	1.5	1.5	NA	NA
3.3-V PCI	3.3	NA	3.3	NA	NA	NA
3.3-V PCI-X	3.3	NA	3.3	NA	NA	NA
SSTL-2 Class I	2.5	2.5	2.5	2.5	1.25	1.25
SSTL-2 Class II	2.5	2.5	2.5	2.5	1.25	1.25
SSTL-18 Class I	1.8	1.8	1.8	1.8	0.90	0.90
SSTL-18 Class II	1.8	1.8	1.8	NA	0.90	0.90
1.8-V HSTL Class I	1.8	1.8	1.8	1.8	0.90	0.90
1.8-V HSTL Class II	1.8	1.8	1.8	NA	0.90	0.90
1.5-V HSTL Class I	1.5	1.5	1.5	1.5	0.75	0.75
1.5-V HSTL Class II	1.5	1.5	1.5	NA	0.75	0.75
1.2-V HSTL *(4)*	1.2	NA	1.2	NA	0.6	NA
Differential SSTL-2 Class I	2.5	2.5	2.5	2.5	1.25	1.25
Differential SSTL-2 Class II	2.5	2.5	2.5	2.5	1.25	1.25
Differential SSTL-18 Class I	1.8	1.8	1.8	1.8	0.90	0.90
Differential SSTL-18 Class II	1.8	1.8	1.8	NA	0.90	0.90
1.8-V differential HSTL Class I	1.8	1.8	1.8	NA	0.90	0.90
1.8-V differential HSTL Class II	1.8	1.8	1.8	NA	0.90	0.90
1.5-V differential HSTL Class I	1.5	1.5	1.5	NA	0.75	0.75
1.5-V differential HSTL Class II	1.5	1.5	1.5	NA	0.75	0.75
LVDS *(2)*	3.3/2.5/1.8/1.5	2.5	3.3	2.5	NA	NA

Table 10–4. Stratix II and Stratix II GX I/O Standards and Voltage Levels (Part 3 of 3) Note (1)

I/O Standard	Stratix II and Stratix II GX					
	V_{CCIO} (V)				V_{REF} (V)	V_{TT} (V)
	Input Operation		Output Operation		Input	Termination
	Top and Bottom I/O Banks	Left and Right I/O Banks (3)	Top and Bottom I/O Banks	Left and Right I/O Banks (3)		
HyperTransport technology	NA	2.5	NA	2.5	NA	NA
Differential LVPECL (2)	3.3/2.5/1.8/1.5	NA	3.3	NA	NA	NA

Notes to Table 10–4:

(1) Any input pins with PCI-clamping diode will clamp the V_{CCIO} to 3.3 V.

(2) LVDS and LVPECL output operation in the top and bottom banks is only supported in PLL banks 9-12. The V_{CCIO} level for differential output operation in the PLL banks is 3.3 V. The V_{CCIO} level for output operation in the left and right I/O banks is 2.5 V.

(3) The right I/O bank does not apply to the Stratix II GX. The right I/O Bank on Stratix II GX devices consists of transceivers.

(4) 1.2-V HSTL is only supported in I/O banks 4,7, and 8.

Refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* or the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook* for detailed electrical characteristics of each I/O standard.

On-Chip Termination

Stratix II and Stratix II GX devices feature on-chip termination to provide I/O impedance matching and termination capabilities. Apart from maintaining signal integrity, this feature also minimizes the need for external resistor networks, thereby saving board space and reducing costs.

Stratix II and Stratix II GX devices support on-chip series (R_S) and parallel (R_T) termination for single-ended I/O standards and on-chip differential termination (R_D) for differential I/O standards. This section discusses the on-chip series termination support.

For more information on differential on-chip termination, Refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook.*

The Stratix II and Stratix II GX devices supports I/O driver on-chip series (R_S) and parallel (R_T) termination through drive strength control for single-ended I/Os. There are three ways to implement the R_S and (R_T) in Stratix II and Stratix II GX devices:

- R_S without calibration for both row I/Os and column I/Os
- R_S with calibration only for column I/Os
- R_T with calibration only for column I/Os

On-Chip Series Termination without Calibration

Stratix II and Stratix II GX devices support driver impedance matching to provide the I/O driver with controlled output impedance that closely matches the impedance of the transmission line. As a result, reflections can be significantly reduced. Stratix II and Stratix II GX devices support on-chip series termination for single-ended I/O standards (see Figure 10–23). The R_S shown in Figure 10–23 is the intrinsic impedance of transistors. The typical R_S values are 25Ω and 50Ω Once matching impedance is selected, current drive strength is no longer selectable.

On-chip series termination without calibration is supported on output pins or on the output function of bidirectional pins.

Figure 10–23. Stratix II and Stratix II GX On-Chip Series Termination without Calibration

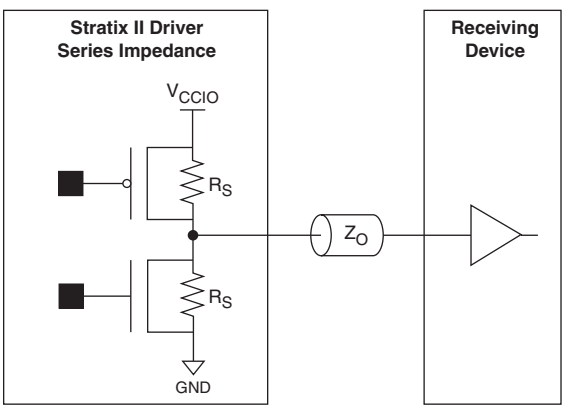


Table 10–5 shows the list of output standards that support on-chip series termination without calibration.

Table 10–5. Selectable I/O Drivers with On-Chip Series Termination without Calibration

I/O Standard	On-chip Series Termination Setting		
	Row I/O	Column I/O	Unit
3.3-V LVTTL	50	50	Ω
	25	25	Ω
3.3-V LVCMOS	50	50	Ω
	25	25	Ω
2.5-V LVTTL	50	50	Ω
	25	25	Ω
2.5-V LVCMOS	50	50	Ω
	25	25	Ω
1.8-V LVTTL	50	50	Ω
		25	Ω
1.8-V LVCMOS	50	50	Ω
		25	Ω
1.5-V LVTTL	50	50	Ω
1.5-V LVCMOS	50	50	Ω
SSTL-2 Class I	50	50	Ω
SSTL-2 Class II	25	25	Ω
SSTL-18 Class I	50	50	Ω
SSTL-18 Class II		25	Ω
1.8-V HSTL Class I	50	50	Ω
1.8-V HSTL Class II		25	Ω
1.5-V HSTL Class I	50	50	Ω
1.2-V HSTL *(1)*		50	Ω

Note to Table 10–5:
(1) 1.2-V HSTL is only supported in I/O banks 4,7, and 8.

To use on-chip termination for the SSTL Class I standard, users should select the 50-Ω on-chip series termination setting for replacing the external 25-Ω R_S (to match the 50-Ω transmission line). For the SSTL Class II standard, users should select the 25-Ω on-chip series termination setting (to match the 50-Ω transmission line and the near end 50-Ω pull-up to V_{TT}).

For more information on tolerance specifications for on-chip termination without calibration, refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* or the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

On-Chip Series Termination with Calibration

Stratix II and Stratix II GX devices support on-chip series termination with calibration in column I/Os in top and bottom banks. Every column I/O buffer consists of a group of transistors in parallel. Each transistor can be individually enabled or disabled. The on-chip series termination calibration circuit compares the total impedance of the transistor group to the external 25-Ωor 50-Ωresistors connected to the RUP and RDN pins, and dynamically enables or disables the transistors until they match (as shown in Figure 10–24). The R_S shown in Figure 10–24 is the intrinsic impedance of transistors. Calibration happens at the end of device configuration. Once the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers.

On-chip series termination with calibration is supported on output pins or on the output function of bidirectional pins.

Figure 10–24. Stratix II and Stratix II GX On-Chip Series Termination with Calibration

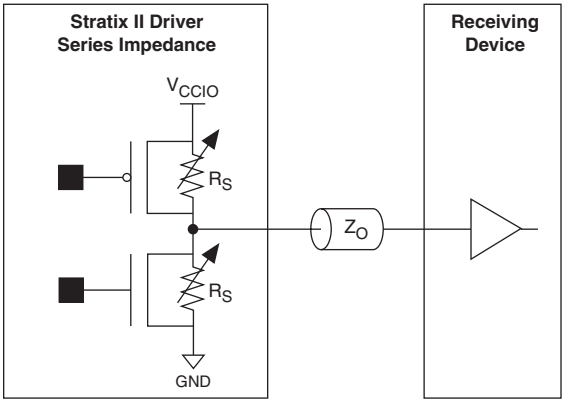


Table 10–6 shows the list of output standards that support on-chip series termination with calibration.

Table 10–6. Selectable I/O Drivers with On-Chip Series Termination with Calibration

I/O Standard	On-Chip Series Termination Setting (Column I/O)	Unit
3.3-V LVTTL	50	Ω
	25	Ω
3.3-V LVCMOS	50	Ω
	25	Ω
2.5-V LVTTL	50	Ω
	25	Ω
2.5-V LVCMOS	50	Ω
	25	Ω
1.8-V LVTTL	50	Ω
	25	Ω
1.8-V LVCMOS	50	Ω
	25	Ω
1.5 LVTTL	50	Ω
1.5 LVCMOS	50	Ω
SSTL-2 Class I	50	Ω
SSTL-2 Class II	25	Ω
SSTL-18 Class I	50	Ω
SSTL-18 Class II	25	Ω
1.8-V HSTL Class I	50	Ω
1.8-V HSTL Class II	25	Ω
1.5-V HSTL Class I	50	Ω
1.2-V HSTL *(1)*	50	Ω

Note to Table 10–6:
(1) 1.2-V HSTL is only supported in I/O banks 4,7, and 8.

On-Chip Parallel Termination with Calibration

Stratix II and Stratix II GX devices support on-chip parallel termination with calibration in column I/Os in top and bottom banks. Every column I/O buffer consists of a group of transistors in parallel. Each transistor can be individually enabled or disabled. The on-chip parallel termination calibration circuit compares the total impedance of the transistor group to

the external 50-Ω resistors connected to the RUP and RDN pins and dynamically enables or disables the transistors until they match. Calibration happens at the end of the device configuration. Once the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers.

Table 10–7. Selectable I/O Drivers with On-Chip Parallel Termination with Calibration

I/O Standard	On-Chip Parallel Termination Setting (Column I/O)	Unit
SSTL-2 Class I	50	Ω
SSTL-2 Class II	50	Ω
SSTL-18 Class I	50	Ω
SSTL-18 Class II	50	Ω
1.8-V HSTL Class I	50	Ω
1.8-V HSTL Class II	50	Ω
1.5-V HSTL Class I	50	Ω
1.5-V HSTL Class II	50	Ω
1.2-V HSTL *(1)*	50	Ω

Note to Table 10–7:
(1) 1.2-V HSTL is only supported in I/O banks 4,7, and 8.

There are two separate sets of calibration circuits in the Stratix II and Stratix II GX devices:

- One calibration circuit for top banks 3 and 4
- One calibration circuit for bottom banks 7 and 8

Calibration circuits rely on the external pull-up reference resistor (R_{UP}) and pull-down reference resistor (R_{DN}) to achieve accurate on-chip series and parallel termination. There is one pair of RUP and RDN pins in bank 4 for the calibration circuit for top I/O banks 3 and 4. Similarly, there is one pair of RUP and RDN pins in bank 7 for the calibration circuit for bottom I/O banks 7 and 8. Two banks share the same calibration circuitry, so they must have the same V_{CCIO} voltage if both banks enable on-chip series or parallel termination with calibration. If banks 3 and 4 have different V_{CCIO} voltages, only bank 4 can enable on-chip series or parallel termination with calibration because the RUP and RDN pins are located in bank 4. Bank 3 still can use on-chip series termination, but without calibration. The same rule applies to banks 7 and 8.

On-chip parallel termination with calibration is only supported for input pins. Pins configured as bidirectional do not support on-chip parallel termination.

The RUP and RDN pins are dual-purpose I/Os, which means they can be used as regular I/Os if the calibration circuit is not used. When used for calibration, the RUP pin is connected to V_{CCIO} through an external 25-Ω or 50-Ω resistor for an on-chip series termination value of 25 Ω or 50 Ω, respectively. The RDN pin is connected to GND through an external 25-Ω or 50-Ω resistor for an on-chip series termination value of 25 Ω or 50 Ω, respectively. For on-chip parallel termination, the RUP pin is connected to V_{CCIO} through an external 50-Ω resistor, and RDN is connected to GND through an external 50-Ω resistor.

For more information on tolerance specifications for on-chip termination with calibration, refer to the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* or the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Design Considerations

While Stratix II and Stratix II GX devices feature various I/O capabilities for high-performance and high-speed system designs, there are several other considerations that require attention to ensure the success of those designs.

I/O Termination

I/O termination requirements for single-ended and differential I/O standards are discussed in this section.

Single-Ended I/O Standards

Although single-ended, non-voltage-referenced I/O standards do not require termination, impedance matching is necessary to reduce reflections and improve signal integrity.

Voltage-referenced I/O standards require both an input reference voltage, V_{REF}, and a termination voltage, V_{TT}. The reference voltage of the receiving device tracks the termination voltage of the transmitting device. Each voltage-referenced I/O standard requires a unique termination setup. For example, a proper resistive signal termination scheme is critical in SSTL standards to produce a reliable DDR memory system with superior noise margin.

Stratix II and Stratix II GX on-chip series and parallel termination provides the convenience of no external components. External pull-up resistors can be used to terminate the voltage-referenced I/O standards such as SSTL-2 and HSTL.

Refer to the "Stratix II and Stratix II GX I/O Standards Support" on page 10–2 for more information on the termination scheme of various single-ended I/O standards.

Differential I/O Standards

Differential I/O standards typically require a termination resistor between the two signals at the receiver. The termination resistor must match the differential load impedance of the bus. Stratix II and Stratix II GX devices provide an optional differential on-chip resistor when using LVDS and HyperTransport standards.

I/O Banks Restrictions

Each I/O bank can simultaneously support multiple I/O standards. The following sections provide guidelines for mixing non-voltage-referenced and voltage-referenced I/O standards in Stratix II and Stratix II GX devices.

Non-Voltage-Referenced Standards

Each Stratix II and Stratix II GX device I/O bank has its own V_{CCIO} pins and supports only one V_{CCIO}, either 1.5, 1.8, 2.5, or 3.3 V. An I/O bank can simultaneously support any number of input signals with different I/O standard assignments, as shown in Table 10–8.

For output signals, a single I/O bank supports non-voltage-referenced output signals that are driving at the same voltage as V_{CCIO}. Since an I/O bank can only have one V_{CCIO} value, it can only drive out that one value for non-voltage-referenced signals. For example, an I/O bank with a 2.5-V V_{CCIO} setting can support 2.5-V standard inputs and outputs and 3.3-V LVCMOS inputs (not output or bidirectional pins).

Table 10–8. Acceptable Input Levels for LVTTL and LVCMOS (Part 1 of 2)

Bank V_{CCIO} (V)	Acceptable Input Levels (V)			
	3.3	2.5	1.8	1.5
3.3	✓	✓ *(1)*		
2.5	✓	✓		

Table 10–8. Acceptable Input Levels for LVTTL and LVCMOS (Part 2 of 2)

Bank V_{CCIO} (V)	Acceptable Input Levels (V)			
	3.3	2.5	1.8	1.5
1.8	✓ (2)	✓ (2)	✓	✓ (1)
1.5	✓ (2)	✓ (2)	✓	✓

Notes to Table 10–8:

(1) Because the input signal does not drive to the rail, the input buffer does not completely shut off, and the I/O current is slightly higher than the default value.

(2) These input values overdrive the input buffer, so the pin leakage current is slightly higher than the default value. To drive inputs higher than V_{CCIO} but less than 4.0 V, disable the PCI clamping diode and select the **Allow LVTTL and LVCMOS input levels to overdrive input buffer** option in the Quartus II software.

Voltage-Referenced Standards

To accommodate voltage-referenced I/O standards, each Stratix II or Stratix II GX device's I/O bank supports multiple V_{REF} pins feeding a common V_{REF} bus. The number of available V_{REF} pins increases as device density increases. If these pins are not used as V_{REF} pins, they cannot be used as generic I/O pins. However, each bank can only have a single V_{CCIO} voltage level and a single V_{REF} voltage level at a given time.

An I/O bank featuring single-ended or differential standards can support voltage-referenced standards as long as all voltage-referenced standards use the same V_{REF} setting.

Because of performance reasons, voltage-referenced input standards use their own V_{CCIO} level as the power source. For example, you can only place 1.5-V HSTL input pins in an I/O bank with a 1.5-V V_{CCIO}.

Refer to the "Stratix II and Stratix II GX I/O Banks" on page 10–20 for details on input V_{CCIO} for voltage-referenced standards.

Voltage-referenced bidirectional and output signals must be the same as the I/O bank's V_{CCIO} voltage. For example, you can only place SSTL-2 output pins in an I/O bank with a 2.5-V V_{CCIO}.

Refer to the "I/O Placement Guidelines" on page 10–36 for details on voltage-referenced I/O standards placement.

Mixing Voltage-Referenced and Non-Voltage-Referenced Standards

An I/O bank can support both non-voltage-referenced and voltage-referenced pins by applying each of the rule sets individually. For example, an I/O bank can support SSTL-18 inputs and 1.8-V inputs and outputs with a 1.8-V V_{CCIO} and a 0.9-V V_{REF}. Similarly, an I/O bank can support 1.5-V standards, 2.5-V (inputs, but not outputs), and HSTL I/O standards with a 1.5-V V_{CCIO} and 0.75-V V_{REF}.

I/O Placement Guidelines

The I/O placement guidelines help to reduce noise issues that may be associated with a design such that Stratix II and Stratix II GX FPGAs can maintain an acceptable noise level on the V_{CCIO} supply. Because Stratix II and Stratix II GX devices require each bank to be powered separately for V_{CCIO}, these noise issues have no effect when crossing bank boundaries and, as such, these rules need not be applied.

This section provides I/O placement guidelines for the programmable I/O standards supported by Stratix II and Stratix II GX devices and includes essential information for designing systems using their devices' selectable I/O capabilities.

V_{REF} Pin Placement Restrictions

There are at least two dedicated V_{REF} pins per I/O bank to drive the V_{REF} bus. Larger Stratix II and Stratix II GX devices have more V_{REF} pins per I/O bank. All V_{REF} pins within one I/O bank are shorted together at device die level.

There are limits to the number of pins that a V_{REF} pin can support. For example, each output pin adds some noise to the V_{REF} level and an excessive number of outputs make the level too unstable to be used for incoming signals.

Restrictions on the placement of single-ended voltage-referenced I/O pads with respect to V_{REF} pins help maintain an acceptable noise level on the V_{CCIO} supply and prevent output switching noise from shifting the V_{REF} rail.

Input Pins
Each V_{REF} pin supports a maximum of 40 input pads.

Output Pins

When a voltage-referenced input or bidirectional pad does not exist in a bank, the number of output pads that can be used in that bank depends on the total number of available pads in that same bank. However, when a voltage-referenced input exists, a design can use up to 20 output pads per V_{REF} pin in a bank.

Bidirectional Pins

Bidirectional pads must satisfy both input and output guidelines simultaneously. The general formulas for input and output rules are shown in Table 10–9.

Table 10–9. Bidirectional Pin Limitation Formulas

Rules	Formulas
Input	<Total number of bidirectional pins> + <Total number of V_{REF} input pins, if any> ≤40 per V_{REF} pin
Output	<Total number of bidirectional pins> + <Total number of output pins, if any> – <Total number of pins from smallest OE group, if more than one OE groups> ≤20 per V_{REF} pin

- If the same output enable (OE) controls all the bidirectional pads (bidirectional pads in the same OE group are driving in and out at the same time) and there are no other outputs or voltage-referenced inputs in the bank, then the voltage-referenced input is never active at the same time as an output. Therefore, the output limitation rule does not apply. However, since the bidirectional pads are linked to the same OE, the bidirectional pads will all act as inputs at the same time. Therefore, there is a limit of 40 input pads, as follows:

  <Total number of bidirectional pins> + *<Total number of V_{REF} input pins>* ≤40 per V_{REF} pin

- If any of the bidirectional pads are controlled by different OE and there are no other outputs or voltage-referenced inputs in the bank, then one group of bidirectional pads can be used as inputs and another group is used as outputs. In such cases, the formula for the output rule is simplified, as follows:

  <Total number of bidirectional pins> – *<Total number of pins from smallest OE group>* ≤20 per V_{REF} pin

- Consider a case where eight bidirectional pads are controlled by OE1, eight bidirectional pads are controlled by OE2, six bidirectional pads are controlled by OE3, and there are no other outputs or voltage-referenced inputs in the bank. While this totals 22 bidirectional pads, it is safely allowable because there would be a possible maximum of 16 outputs per V_{REF} pin, assuming the worst case where OE1 and OE2 are active and OE3 is inactive. This is useful for DDR SDRAM applications.

- When at least one additional voltage-referenced input and no other outputs exist in the same V_{REF} group, the bidirectional pad limitation must simultaneously adhere to the input and output limitations. The input rule becomes:

  <Total number of bidirectional pins> + *<Total number of V_{REF} input pins>* ≤40 per V_{REF} pin

  Whereas the output rule is simplified as:

  <Total number of bidirectional pins> ≤20 per V_{REF} pin

- When at least one additional output exists but no voltage-referenced inputs exist, the output rule becomes:

  <Total number of bidirectional pins> + *<Total number of output pins>* – *<Total number of pins from smallest OE group>* ≤20 per V_{REF} pin

- When additional voltage-referenced inputs and other outputs exist in the same V_{REF} group, then the bidirectional pad limitation must again simultaneously adhere to the input and output limitations. The input rule is:

  <Total number of bidirectional pins> + *<Total number of V_{REF} input pins>* ≤40 per V_{REF} pin

  Whereas the output rule is given as:

  <Total number of bidirectional pins> + *<Total number of output pins>* – *<Total number of pins from smallest OE group>* ≤20 per V_{REF} pin

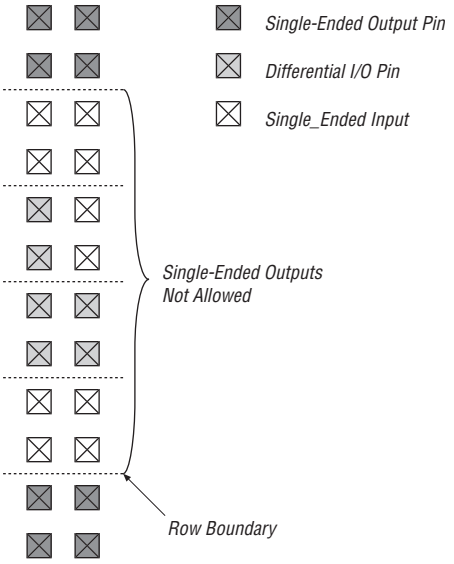
I/O Pin Placement with Respect to High-Speed Differential I/O Pins

Regardless of whether or not the SERDES circuitry is utilized, there is a restriction on the placement of single-ended output pins with respect to high-speed differential I/O pins. As shown in Figure 10–25, all

single-ended outputs must be placed at least one LAB row away from the differential I/O pins. There are no restrictions on the placement of single-ended input pins with respect to differential I/O pins. Single-ended input pins may be placed within the same LAB row as differential I/O pins. However, the single-ended input's IOE register is not available. The input must be implemented within the core logic.

This single-ended output pin placement restriction only applies when using the LVDS or HyperTransport I/O standards in the left and right I/O banks. There are no restrictions for single-ended output pin placement with respect to differential clock pins in the top and bottom I/O banks.

Figure 10–25. Single-Ended Output Pin Placement with Respect to Differential I/O Pins



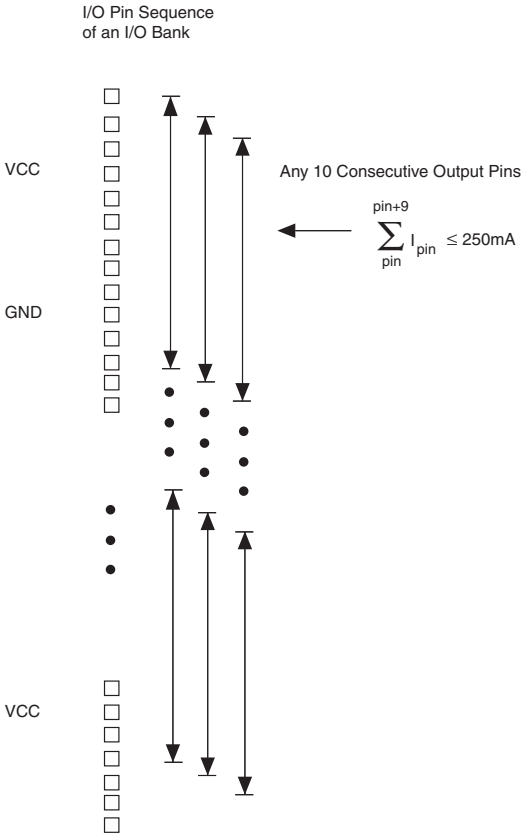
DC Guidelines

Power budgets are essential to ensure the reliability and functionality of a system application. You are often required to perform power dissipation analysis on each device in the system to come out with the total power dissipated in that system, which is composed of a static component and a dynamic component.

The static power consumption of a device is the total DC current flowing from V_{CCIO} to ground.

For any ten consecutive pads in an I/O bank of Stratix II and Stratix II GX devices, Altera recommends a maximum current of 250 mA, as shown in Figure 10–26, because the placement of V_{CCIO}/ground (GND) bumps are regular, 10 I/O pins per pair of power pins. This limit is on the static power consumed by an I/O standard, as shown in Table 10–10. Limiting static power is a way to improve reliability over the lifetime of the device.

Figure 10–26. DC Current Density Restriction *Notes (1), (2)*



Notes to Figure 10–26:

(1) The consecutive pads do not cross I/O banks.

(2) V_{REF} pins do not affect DC current calculation because there are no V_{REF} pads.

Table 10–10 shows the I/O standard DC current specification.

Table 10–10. Stratix II and Stratix II GX I/O Standard DC Current Specification (Part 1 of 2) *Note (1)*

I/O Standard	I_{PIN} (mA), Top and Bottom I/O Banks	I_{PIN} (mA), Left and Right I/O Banks *(2)*
LVTTL	*(3)*	*(3)*
LVCMOS	*(3)*	*(3)*
2.5 V	*(3)*	*(3)*
1.8 V	*(3)*	*(3)*
1.5 V	*(3)*	*(3)*
3.3-V PCI	1.5	NA
3.3-V PCI-X	1.5	NA
SSTL-2 Class I	12 *(4)*	12 *(4)*
SSTL-2 Class II	24 *(4)*	16 *(4)*
SSTL-18 Class I	12 *(4)*	10 *(4)*
SSTL-18 Class II	20 *(4)*	NA
1.8-V HSTL Class I	12 *(4)*	12
1.8-V HSTL Class II	20 *(4)*	NA
1.5-V HSTL Class I	12 *(4)*	8
1.5-V HSTL Class II	20 *(4)*	NA
Differential SSTL-2 Class I	12	12
Differential SSTL-2 Class II	24	16
Differential SSTL-18 Class I	12	10
Differential SSTL-18 Class II	20	NA
1.8-V differential HSTL Class I	12	12
1.8-V differential HSTL Class II	20	NA
1.5-V differential HSTL Class I	12	8
1.5-V differential HSTL Class II	20	NA

Table 10–10. Stratix II and Stratix II GX I/O Standard DC Current Specification (Part 2 of 2) Note (1)		
I/O Standard	I_{PIN} (mA), Top and Bottom I/O Banks	I_{PIN} (mA), Left and Right I/O Banks (2)

Notes to Table 10–10:

(1) The current value obtained for differential HSTL and differential SSTL standards is per pin and not per differential pair, as opposed to the per-pair current value of LVDS and HyperTransport standards.

(2) This does not apply to the right I/O banks of Stratix II GX devices. Stratix II GX devices have transceivers on the right I/O banks.

(3) The DC power specification of each I/O standard depends on the current sourcing and sinking capabilities of the I/O buffer programmed with that standard, as well as the load being driven. LVTTL, LVCMOS, 2.5-V, 1.8-V, and 1.5-V outputs are not included in the static power calculations because they normally do not have resistor loads in real applications. The voltage swing is rail-to-rail with capacitive load only. There is no DC current in the system.

(4) This I_{PIN} value represents the DC current specification for the default current strength of the I/O standard. The I_{PIN} varies with programmable drive strength and is the same as the drive strength as set in Quartus II software. Refer to the *Stratix II Architecture* chapter in volume 1 of the *Stratix II Device Handbook* or the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook* for a detailed description of the programmable drive strength feature of voltage-referenced I/O standards.

Table 10–10 only shows the limit on the static power consumed by an I/O standard. The amount of power used at any moment could be much higher, and is based on the switching activities.

Conclusion

Stratix II and Stratix II GX devices provide I/O capabilities that allow you to work in compliance with current and emerging I/O standards and requirements. With the Stratix II or Stratix II GX devices features, such as programmable driver strength, you can reduce board design interface costs and increase the development flexibility.

References

Refer to the following references for more information:

- Interface Standard for Nominal 3V/ 3.3-V Supply Digital Integrated Circuits, JESD8-B, Electronic Industries Association, September 1999.
- 2.5-V +/- 0.2V (Normal Range) and 1.8-V to 2.7V (Wide Range) Power Supply Voltage and Interface Standard for Non-terminated Digital Integrated Circuits, JESD8-5, Electronic Industries Association, October 1995.
- 1.8-V +/- 0.15 V (Normal Range) and 1.2 V - 1.95 V (Wide Range) Power Supply Voltage and Interface Standard for Non-terminated Digital Integrated Circuits, JESD8-7, Electronic Industries Association, February 1997.
- 1.5-V +/- 0.1 V (Normal Range) and 0.9 V - 1.6 V (Wide Range) Power Supply Voltage and Interface Standard for Non-terminated Digital Integrated Circuits, JESD8-11, Electronic Industries Association, October 2000.

- PCI Local Bus Specification, Revision 2.2, PCI Special Interest Group, December 1998.
- PCI-X Local Bus Specification, Revision 1.0a, PCI Special Interest Group.
- Stub Series Terminated Logic for 2.5-V (SSTL-2), JESD8-9A, Electronic Industries Association, December 2000.
- Stub Series Terminated Logic for 1.8 V (SSTL-18), Preliminary JC42.3, Electronic Industries Association.
- High-Speed Transceiver Logic (HSTL)—A 1.5-V Output Buffer Supply Voltage Based Interface Standard for Digital Integrated Circuits, EIA/JESD8-6, Electronic Industries Association, August 1995.
- Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits, ANSI/TIA/EIA-644, American National Standards Institute/Telecommunications Industry/Electronic Industries Association, October 1995.

Referenced Documents

This chapter references the following documents:

- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook*
- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *External Memory Interfaces in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook*
- *External Memory Interfaces in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook*
- *High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook*
- *PLLs in Stratix II & Straix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Stratix II Architecture* chapter in volume 1 of the *Stratix II Device Handbook*
- *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *Stratix II GX Transceiver User Guide* (volume 1) of the *Stratix II GX Device Handbook*

Document Revision History

Table 10–11 shows the revision history for this chapter.

Table 10–11. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2007 v4.6	Updated Figure 10–22.	—
	Updated Note 4 to Table 10–2.	—
	Added "Referenced Documents" section.	—
	Minor text edits.	—
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 9. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
May 2007 v4.5	Added a note to the "On-Chip Series Termination with Calibration" section.	—
	Added a note to the "On-Chip Series Termination without Calibration" section	—
	Updated note to the "Stratix II and Stratix II GX I/O Features" section.	—
	Updated the "LVDS" section.	—
	Updated note to "1.5 V" section	—
	• Updated Note (1) for Table 10–4 • Updated Note (2) for Table 10–3	—
	Updated Table 10–2, column heading for columns 9 and 10.	—
	Updated Table 10–10.	—
	Fixed typo in the "Stratix II and Stratix II GX I/O Features" section	—
February 2007 v4.4	Added the "Document Revision History" section to this chapter.	—
August 2006 v4.3	Updated Table 9–2, Table 9–4, Table 9–5, Table 9–6, and Table 9–7.	—
April 2006 v4.2	Chapter updated as part of the *Stratix II Device Handbook* update.	—
No change	Formerly chapter 8. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—

Table 10–11. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
December 2005 v4.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—
October 2005 v4.0	Added chapter to the *Stratix II GX Device Handbook*.	—

11. High-Speed Differential I/O Interfaces with DPA in Stratix II & Stratix II GX Devices

SII52005-2.3

Introduction

Stratix® II and Stratix® II GX device family offers up to 1-Gbps differential I/O capabilities to support source-synchronous communication protocols such as HyperTransport™ technology, Rapid I/O, XSBI, and SPI.

Stratix II and Stratix II GX devices have the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmit serializer
- Receive deserializer
- Data realignment circuit
- Dynamic phase aligner (DPA)
- Synchronizer (FIFO buffer)
- Analog PLLs (fast PLLs)

For high-speed differential interfaces, Stratix II and Stratix II GX devices can accommodate different differential I/O standards, including the following:

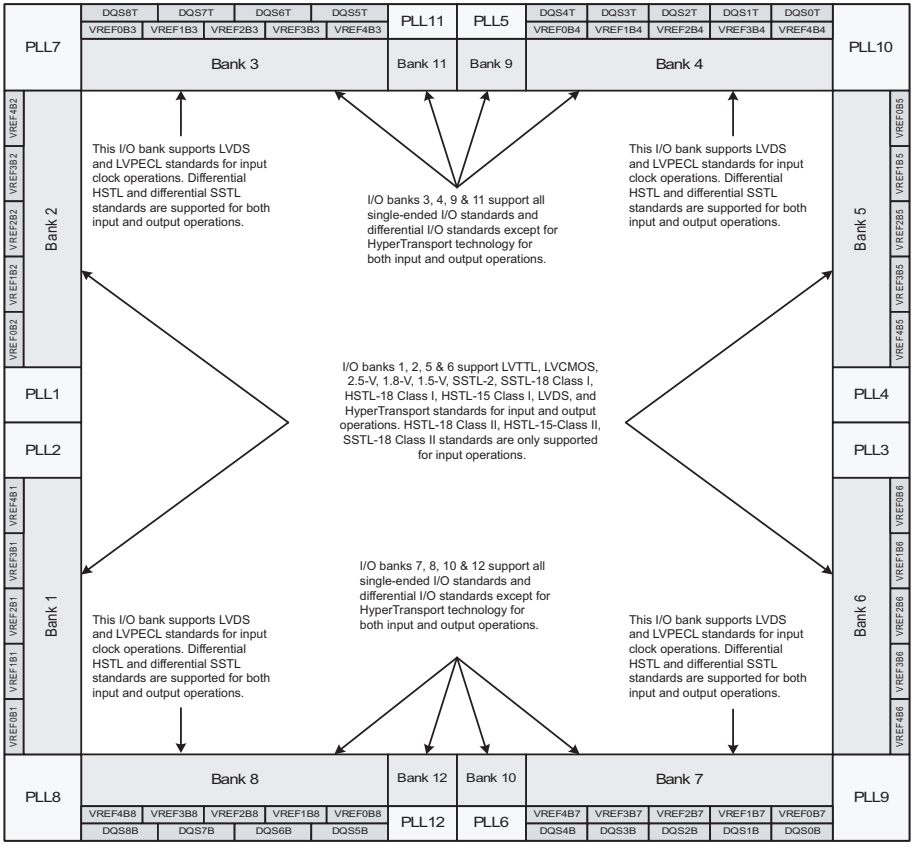
- LVDS
- HyperTransport technology
- HSTL
- SSTL
- LVPECL

HSTL, SSTL, and LVPECL I/O standards can be used only for PLL clock inputs and outputs in differential mode.

I/O Banks

Stratix II and Stratix II GX inputs and outputs are partitioned into banks located on the periphery of the die. The inputs and outputs that support LVDS and HyperTransport technology are located in row I/O banks, two on the left and two on the right side of the Stratix II device and two on the left side of the Stratix II GX device. LVPECL, HSTL, and SSTL standards are supported on certain top and bottom banks of the die (banks 9 to 12) when used as differential clock inputs/outputs. Differential HSTL and SSTL standards can be supported on banks 3, 4, 7, and 8 if the pins on these banks are used as DQS/DQSn pins. Figures 11–1 and 11–2 show where the banks and the PLLs are located on the die.

Figure 11–1. Stratix II I/O Banks *Note (1), (2), (3), (4), (5), (6), and (7)*



Notes to Figure 11–1:

(1) Figure 11–1 is a top view of the silicon die that corresponds to a reverse view for flip-chip packages. It is a graphical representation only. See the pin list and Quartus II software for exact locations.

(2) Depending on the size of the device, different device members have different numbers of V_{REF} groups.

(3) Banks 9 through 12 are enhanced PLL external clock output banks. These PLL banks utilize the adjacent V_{REF} group when voltage-referenced standards are implemented. For example, if an SSTL input is implemented in PLL bank 10, the voltage level at VREFB7 is the reference voltage level for the SSTL input.

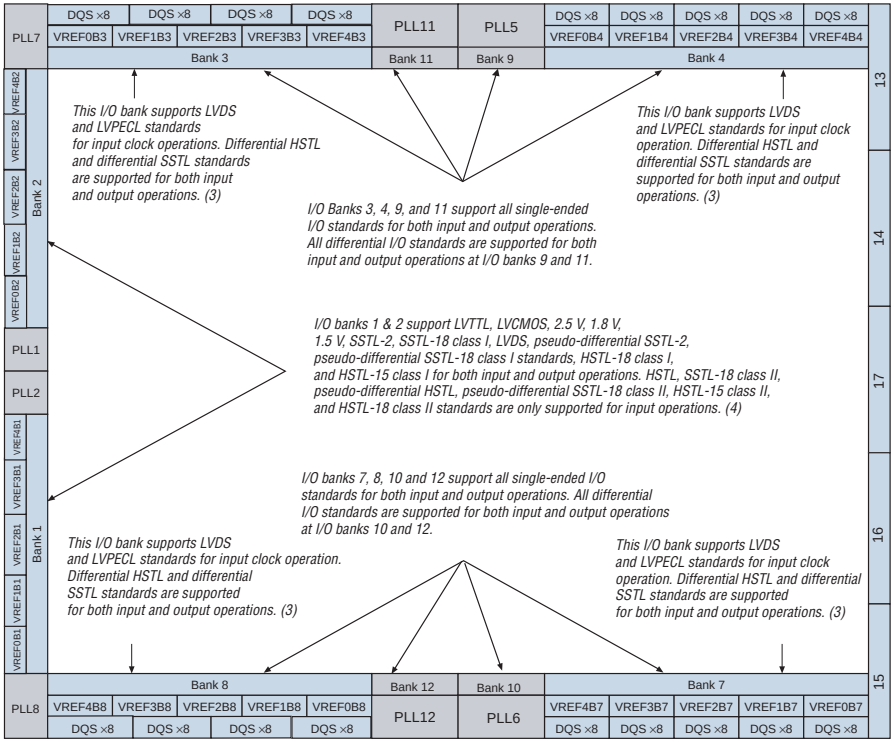
(4) Differential HSTL and differential SSTL standards are available for bidirectional operations on DQS pin and input-only operations on PLL clock input pins; LVDS, LVPECL, and HyperTransport standards are available for input-only operations on PLL clock input pins. See the *Selectable I/O standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* for more details.

(5) Quartus II software does not support differential SSTL and differential HSTL standards at left/right I/O banks. See the *Selectable I/O standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* if you need to implement these standards at these I/O banks.

(6) Banks 11 and 12 are available only in EP2S60, EP2S90, EP2S130, and EP2S180 devices.

(7) PLLs 7, 8, 9, 10, 11, and 12 are available only in EP2S60, EP2S90, EP2S130, and EP2S180 devices.

***Figure 11–2. Stratix II GX I/O Banks** Note (1), (2), (3), (4), (5), (6), and (7)*



Notes to Figure 11–2:

(1) Figure 11–2 is a top view of the silicon die which corresponds to a reverse view for flip-chip packages. It is a graphical representation only.

(2) Depending on size of the device, different device members have different number of V_{REF} groups. Refer to the pin list and the Quartus II software for exact locations.

(3) Banks 9 through 12 are enhanced PLL external clock output banks.

(4) Horizontal I/O banks feature transceiver and DPA circuitry for high speed differential I/O standards.

(5) Quartus II software does not support differential SSTL and differential HSTL standards at left/right I/O banks. Refer to the "Differential Pin Placement Guidelines" on page 11–21 if you need to implement these standards at these I/O banks.

(6) Banks 11 and 12 are available only in EP2SGX60C/D/E, EP2SGX90E/F, and EP2SGX130G.

(7) PLLs 7, 8, 11, and 12 are available only in EP2SGX60C/D/E, EP2SGXE/F, and EP2SGX130G.

Table 11–1 lists the differential I/O standards supported by each bank.

Table 11–1. Supported Differential I/O Types

Bank	Row I/O (Banks 1, 2, 5 and 6) (2)			Column I/O (Banks, 3, 4 and 7 through 12)		
Type	Clock Inputs	Clock Outputs	Data or Regular I/O Pins	Clock Inputs	Clock Outputs	Data or Regular I/O Pins
Differential HSTL				✓	✓	(1)
Differential SSTL				✓	✓	(1)
LVPECL				✓	✓	
LVDS	✓	✓	✓	✓	✓	
HyperTransport technology	✓	✓	✓			

Note to Table 11–1:
(1) Used as both inputs and outputs on the DQS/DQSn pins.
(2) Banks 5 and 6 are not available in Stratix II GX devices.

Table 11–2 shows the total number of differential channels available in Stratix II devices. The available channels are divided evenly between the left and right banks of the die. Non-dedicated clocks in the left and right banks can also be used as data receiver channels. The total number of receiver channels includes these four non-dedicated clock channels. Pin migration is available for different size devices in the same package.

Table 11–2. Differential Channels in Stratix II Devices (Part 1 of 2) Notes (1), (2), and (3)

Device	484-Pin FineLine BGA	484-Pin Hybrid FineLine BGA	672-Pin FineLine BGA	780-Pin FineLine BGA	1,020-Pin FineLine BGA	1,508-Pin FineLine BGA Within the 1,508-pin Fin
EP2S15	38 transmitters 42 receivers		38 transmitters 42 receivers			
EP2S30	38 transmitters 42 receivers		58 transmitters 62 receivers			
EP2S60	38 transmitters 42 receivers		58 transmitters 62 receivers		84 transmitters 84 receivers	
EP2S90		38 transmitters 42 receivers		64 transmitters 68 receivers	90 transmitters 94 receivers	118 transmitters 118 receivers
EP2S130				64 transmitters 68 receivers	88 transmitters 92 receivers	156 transmitters 156 receivers

Table 11–2. Differential Channels in Stratix II Devices (Part 2 of 2) Notes (1), (2), and (3)

Device	484-Pin FineLine BGA	484-Pin Hybrid FineLine BGA	672-Pin FineLine BGA	780-Pin FineLine BGA	1,020-Pin FineLine BGA	1,508-Pin FineLine BGA Within the 1,508-pin Fin
EP2S180					88 transmitters 92 receivers	156 transmitters 156 receivers

Notes to Table 11–2:

(1) Pin count does not include dedicated PLL input pins.

(2) The total number of receiver channels includes the four non-dedicated clock channels that can optionally be used as data channels.

(3) Within the 1,508-pin FineLine BGA package, 92 receiver channels and 92 transmitter channels are vertically migratable.

Table 11–3 shows the total number of differential channels available in Stratix II GX devices. Non-dedicated clocks in the left bank can also be used as data receiver channels. The total number of receiver channels includes these four non-dedicated clock channels. Pin migration is available for different size devices in the same package.

Table 11–3. Differential Channels in Stratix II GX Devices Notes (1), (2), (3)

Device	780-Pin FineLine BGA	1,152-Pin FineLine BGA	1,508-Pin FineLine BGA
EP2SGX30	29 transmitters 31receivers		
EP2SGX60	29 transmitters 31 receivers	42 transmitters 42 receivers	
EP2SGX90		45 transmitters 47 receivers	59 transmitters 59 receivers
EP2SGX130			71 transmitters 73 receivers

Notes to Table 11–3:

(1) Pin count does not include dedicated PLL input pins.

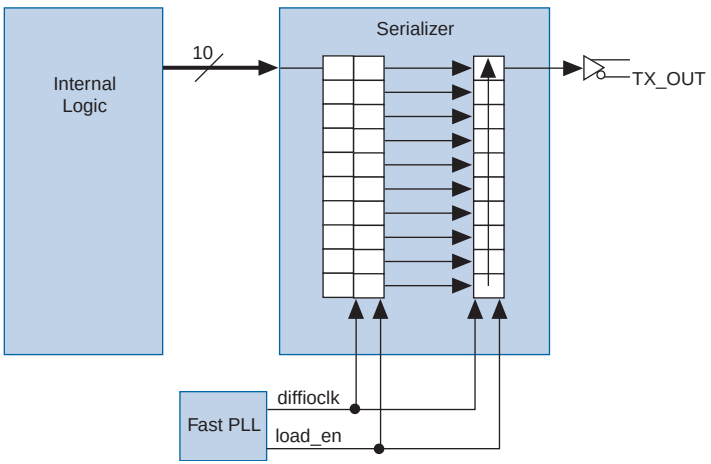
(2) The total number of receiver channels includes the four non-dedicated clock channels that can optionally be used as data channels.

(3) EP2SGX30CF780 devices with four transceiver channels are vertically migratable to EP2SGX60CF780 devices with four transceiver channels. EP2SGX30DF780 devices with eight transceiver channels are vertically migratable to EP2SGX60DF780 devices with eight transceiver channels. EP2SGX60EF1152 devices with 12 transceiver channels are vertically migratable to EP2SGX90EF1152 devices with 12 transceiver channels. EP2SGX90FF1508 devices with 16 transceiver channels are vertically migratable to EP2SGX130GF1508 devices with 20 transceiver channels.

Differential Transmitter

The Stratix II and Stratix II GX transmitter has dedicated circuitry to provide support for LVDS and HyperTransport signaling. The dedicated circuitry consists of a differential buffer, a serializer, and a shared fast PLL. The differential buffer can drive out LVDS or HyperTransport signal levels that are statically set in the Quartus® II software. The serializer takes data from a parallel bus up to 10 bits wide from the internal logic, clocks it into the load registers, and serializes it using the shift registers before sending the data to the differential buffer. The most significant bit (MSB) is transmitted first. The load and shift registers are clocked by the `diffioclk` (a fast PLL clock running at the serial rate) and controlled by the load enable signal generated from the fast PLL. The serialization factor can be statically set to ×4, ×5, ×6, ×7, ×8, ×9, or ×10 using the Quartus II software. The load enable signal is automatically generated by the fast PLL and is derived from the serialization factor setting. Figure 11–3 is a block diagram of the Stratix II transmitter.

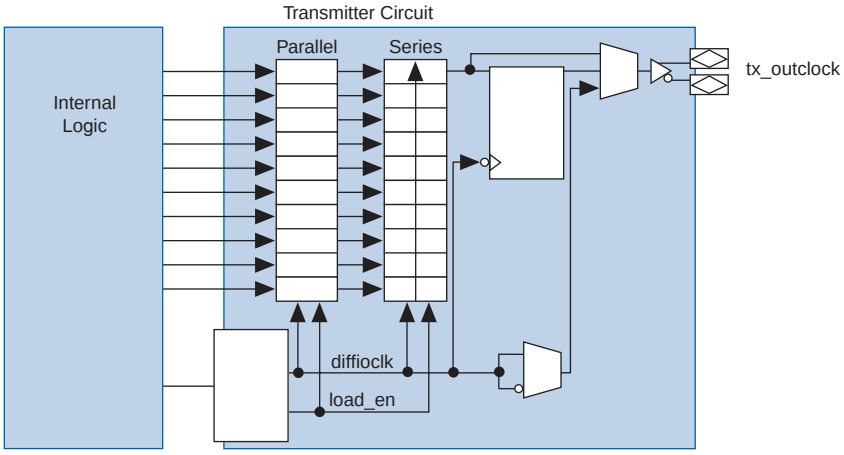
Figure 11–3. Transmitter Block Diagram



Each Stratix II and Stratix II GX transmitter data channel can be configured to operate as a transmitter clock output. This flexibility allows the designer to place the output clock near the data outputs to simplify board layout and reduce clock-to-data skew. Different applications often require specific clock to data alignments or specific data rate to clock rate factors. The transmitter can output a clock signal at the same rate as the data with a maximum frequency of 717 MHz. The output clock can also be divided by a factor of 2, 4, 8, or 10, depending on the serialization factor. The phase of the clock in relation to the data can be set at 0° or 180° (edge or center aligned). The fast PLL provides additional support for

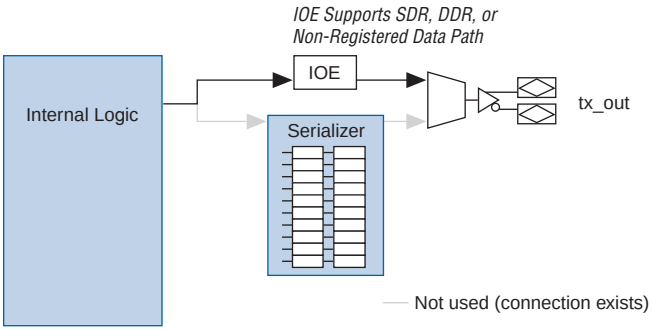
other phase shifts in 45° increments. These settings are made statically in the Quartus II MegaWizard® software. Figure 11–4 shows the transmitter in clock output mode.

Figure 11–4. Transmitter in Clock Output Mode



The serializer can be bypassed to support DDR (×2) and SDR (×1) operations. The I/O element (IOE) contains two data output registers that each can operate in either DDR or SDR mode. The clock source for the registers in the IOE can come from any routing resource, from the fast PLL, or from the enhanced PLL. Figure 11–5 shows the bypass path.

Figure 11–5. Serializer Bypass



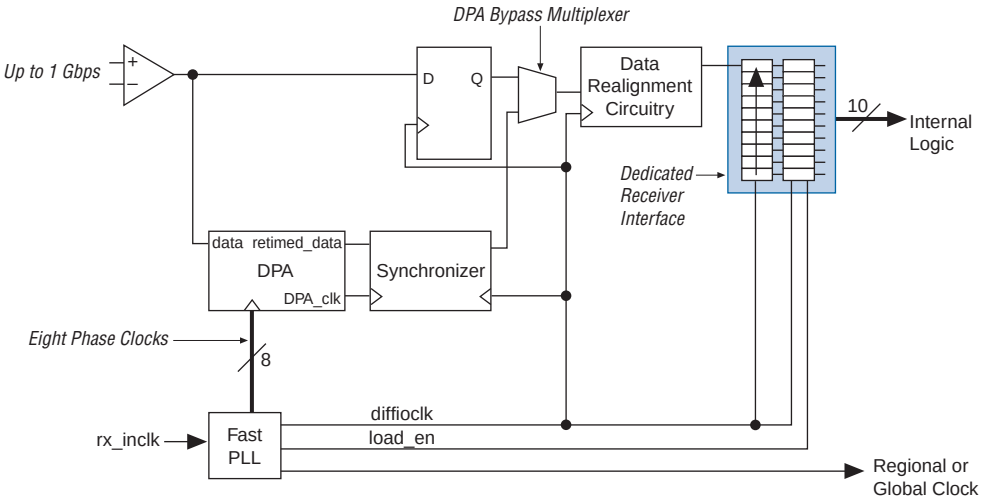
Differential Receiver

The receiver has dedicated circuitry to support high-speed LVDS and HyperTransport signaling, along with enhanced data reception. Each receiver consists of a differential buffer, dynamic phase aligner (DPA), synchronization FIFO buffer, data realignment circuit, deserializer, and a shared fast PLL. The differential buffer receives LVDS or HyperTransport signal levels, which are statically set by the Quartus II software. The DPA block aligns the incoming data to one of eight clock phases to maximize the receiver's skew margin. The DPA circuit can be bypassed on a channel-by-channel basis if it is not needed. Set the DPA bypass statically in the Quartus II MegaWizard Plug-In Manager or dynamically by using the optional `RX_DPLL_ENABLE` port.

The synchronizer circuit is a 1-bit wide by 6-bit deep FIFO buffer that compensates for any phase difference between the DPA block and the deserializer. If necessary, the data realignment circuit inserts a single bit of latency in the serial bit stream to align the word boundary. The deserializer includes shift registers and parallel load registers, and sends a maximum of 10 bits to the internal logic. The data path in the receiver is clocked by either the `diffioclk` signal or the DPA recovered clock. The deserialization factor can be statically set to 4, 5, 6, 7, 8, 9, or 10 by using the Quartus II software. The fast PLL automatically generates the load enable signal, which is derived from the deserialization factor setting.

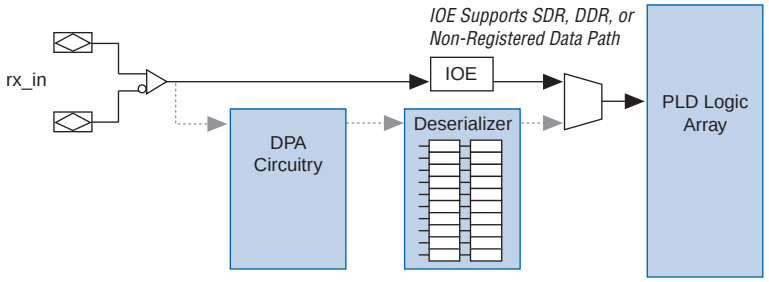
Figure 11–6 shows a block diagram of the receiver.

Figure 11–6. Receiver Block Diagram



The deserializer, like the serializer, can also be bypassed to support DDR (×2) and SDR (×1) operations. The DPA and data realignment circuit cannot be used when the deserializer is bypassed. The IOE contains two data input registers that can operate in DDR or SDR mode. The clock source for the registers in the IOE can come from any routing resource, from the fast PLL, or from the enhanced PLL. Figure 11–7 shows the bypass path.

Figure 11–7. Deserializer Bypass



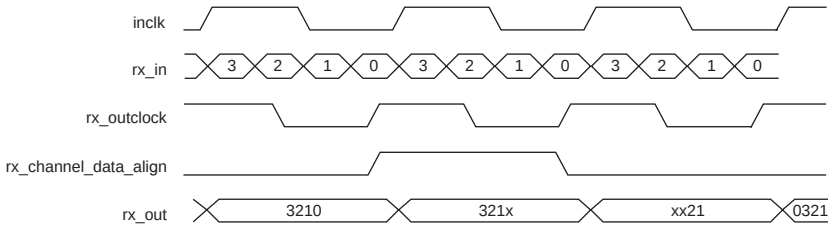
Receiver Data Realignment Circuit

The data realignment circuit aligns the word boundary of the incoming data by inserting bit latencies into the serial stream. An optional RX_CHANNEL_DATA_ALIGN port controls the bit insertion of each receiver independently controlled from the internal logic. The data slips one bit for every pulse on the RX_CHANNEL_DATA_ALIGN port. The following are requirements for the RX_CHANNEL_DATA_ALIGN port:

- The minimum pulse width is one period of the parallel clock in the logic array.
- The minimum low time between pulses is one period of parallel clock.
- There is no maximum high or low time.
- Valid data is available two parallel clock cycles after the rising edge of RX_CHANNEL_DATA_ALIGN.

Figure 11–8 shows receiver output (RX_OUT) after one bit slip pulse with the deserialization factor set to 4.

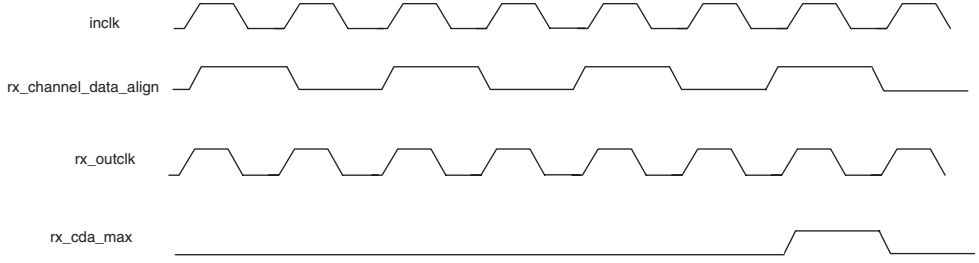
Figure 11–8. Data Realignment Timing



The data realignment circuit can have up to 11 bit-times of insertion before a rollover occurs. The programmable bit rollover point can be from 1 to 11 bit-times independent of the deserialization factor. An optional status port, rx_cda_max, is available to the FPGA from each channel to indicate when the preset rollover point is reached.

Figure 11–9 illustrates a preset value of four bit-times before rollover occurs. The rx_cda_max signal pulses for one rx_outclk cycle to indicate that the rollover has occurred.

Figure 11–9. Receiver Data Re-alignment Rollover

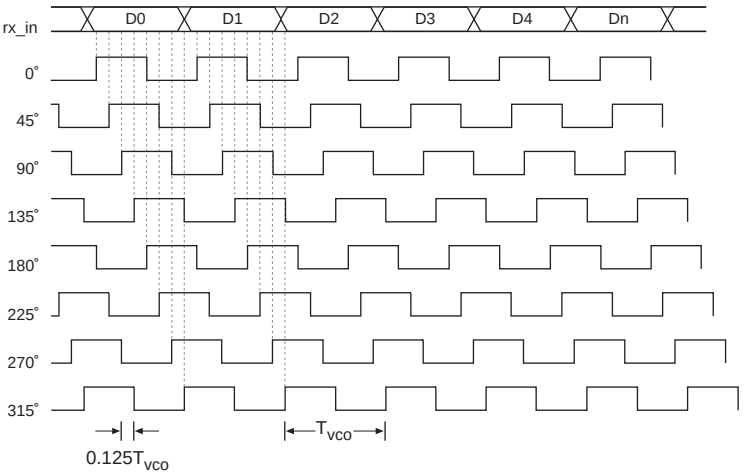


Dynamic Phase Aligner

The DPA block takes in high-speed serial data from the differential input buffer and selects one of eight phase clocks to sample the data. The DPA chooses a phase closest to the phase of the serial data. The maximum phase offset between the data and the phase-aligned clock is 1/8 UI, which is the maximum quantization error of the DPA. The eight phases

are equally divided, giving a 45-degree resolution. Figure 11–10 shows the possible phase relationships between the DPA clocks and the incoming serial data.

Figure 11–10. DPA Clock Phase to Data Bit Relationship



Each DPA block continuously monitors the phase of the incoming data stream and selects a new clock phase if needed. The selection of a new clock phase can be prevented by the optional RX_DPLL_HOLD port, which is available for each channel.

The DPA block requires a training pattern and a training sequence of at least 256 repetitions of the training pattern. The training pattern is not fixed, so you can use any training pattern with at least one transition on each channel. An optional output port, RX_DPA_LOCKED, is available to the internal logic, to indicate when the DPA block has settled on the closest phase to the incoming data phase. The RX_DPA_LOCKED de-asserts, depending on what is selected in the Quartus II MegaWizard Plug-In, when either a new phase is selected, or when the DPA has moved two phases in the same direction. The data may still be valid even when the RX_DPA_LOCKED is deasserted. Use data checkers to validate the data when RX_DPA_LOCKED is deasserted.

An independent reset port, RX_RESET, is available to reset the DPA circuitry. The DPA circuit must be retrained after reset.

Synchronizer

The synchronizer is a 1-bit × 6-bit deep FIFO buffer that compensates for the phase difference between the recovered clock from the DPA circuit and the `diffioclk` that clocks the rest of the logic in the receiver. The synchronizer can only compensate for phase differences, not frequency differences between the data and the receiver's `INCLK`. An optional port, `RX_FIFO_RESET`, is available to the internal logic to reset the synchronizer. The synchronizer is automatically reset when the DPA first locks to the incoming data. Altera® recommends using `RX_FIFO_RESET` to reset the synchronizer when the DPA signals a loss-of-lock condition beyond the initial locking condition.

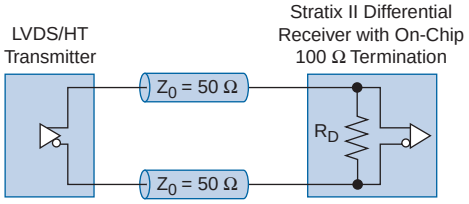
Differential I/O Termination

Stratix II and Stratix II GX devices provide an on-chip 100-Ω differential termination option on each differential receiver channel for LVDS and HyperTransport standards. The on-chip termination eliminates the need to supply an external termination resistor, simplifying the board design and reducing reflections caused by stubs between the buffer and the termination resistor. You can enable on-chip termination in the Quartus II assignments editor. Differential on-chip termination is supported across the full range of supported differential data rates.

For more information, refer to the High-Speed I/O Specifications section of the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* or the High-Speed I/O Specifications section of the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

Figure 11–11 illustrates on-chip termination.

Figure 11–11. On-Chip Differential Termination



On-chip differential termination is supported on all row I/O pins and on clock pins `CLK[0, 2, 8, 10]`. The clock pins `CLK[1, 3, 9, 11]`, and `FPLL[7..10]CLK`, and the clocks in the top and bottom I/O banks (`CLK[4..7, 12..15]`) do not support differential on-chip termination.

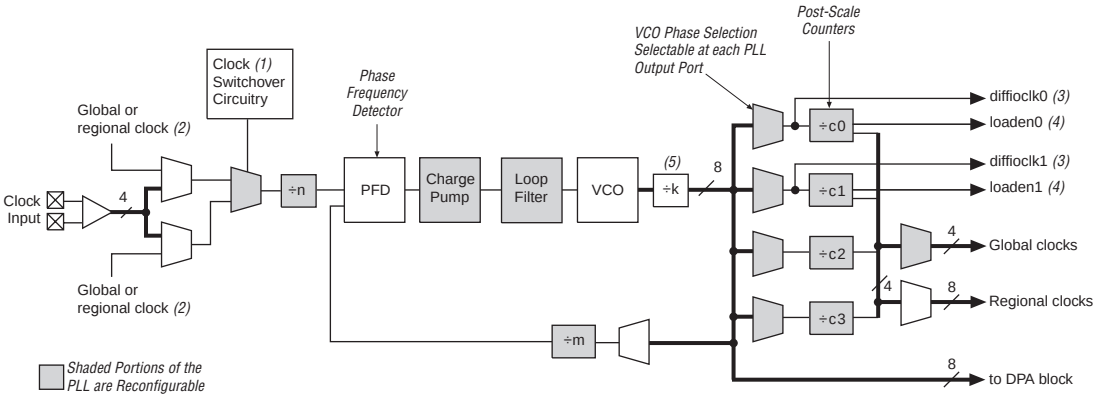
Fast PLL

The high-speed differential I/O receiver and transmitter channels use the fast PLL to generate the parallel global clocks (`rx-` or `tx-` clock) and high-speed clocks (`diffioclk`). Figure 11–12 shows the locations of the fast PLLs. The fast PLL VCO operates at the clock frequency of the data rate. Each fast PLL offers a single serial data rate support, but up to two separate serialization and/or deserialization factors (from the C0 and C1 fast PLL clock outputs) can be used. Clock switchover and dynamic fast PLL reconfiguration is available in high-speed differential I/O support mode.

For additional information on the fast PLL, refer to the *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook* or the *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*.

Figure 11–12 shows a block diagram of the fast PLL in high-speed differential I/O support mode.

Figure 11–12. Fast PLL Block Diagram



Notes to Figure 11–12:

(1) Stratix II fast PLLs only support manual clock switchover.

(2) The global or regional clock input can be driven by an output from another PLL, a pin-driven dedicated global or regional clock, or through a clock control block provided the clock control block is fed by an output from another PLL or pin-driven dedicated global or regional clock.

(3) In high-speed differential I/O support mode, this high-speed PLL clock feeds the SERDES. Stratix II devices only support one rate of data transfer per fast PLL in high-speed differential I/O support mode.

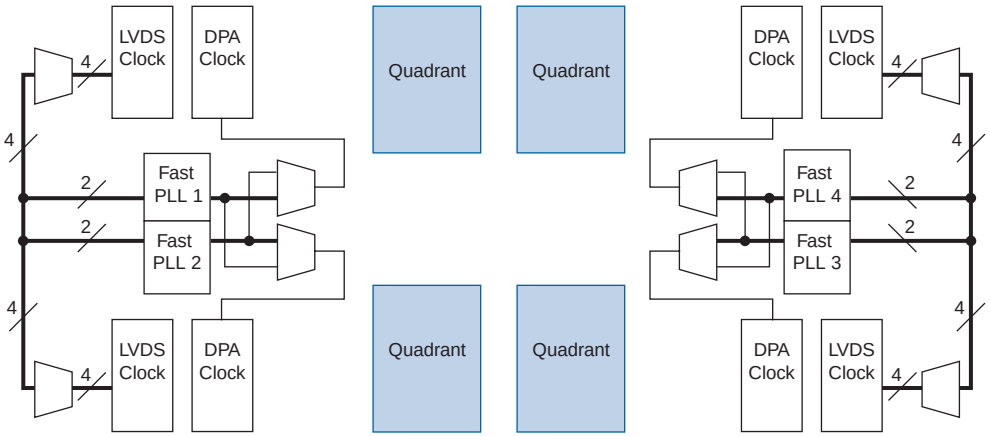
(4) This signal is a high-speed differential I/O support SERDES control signal.

(5) If the design enables this ÷2 counter, the device can use a VCO frequency range of 150 to 520 MHz.

Clocking

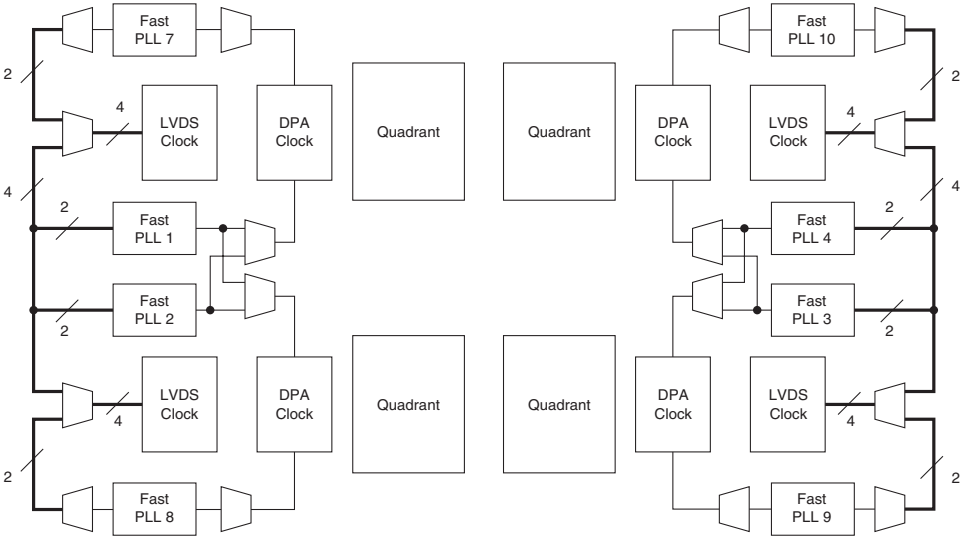
The fast PLLs feed in to the differential receiver and transmitter channels through the LVDS/DPA clock network. The center fast PLLs can independently feed the banks above and below them. The corner PLLs can feed only the banks adjacent to them. Figures 11–13 and 11–14 show the LVDS and DPA clock networks of the Stratix II devices.

Figure 11–13. Fast PLL and LVDS/DPA Clock for EP2S15, EP2S30, and EP2S60 Devices *Note (1)*



Note to Figure 11–13:
(1) Figure 11–13 applies to EP2S60 devices in the 484 and 672 pin packages.

Figure 11–14. Fast PLL and LVDS/DPA Clocks for EP2S60, EP2S90, EP2S130 and EP2S180 Devices Note (1)



Note to Figure 11–14:
(1) Figure 11–14 applies only to the EP2S60 in the 1020 Stratix II GX device.

Figures 11–15 and 11–16 show the Fast PLL and LVDS/DPA clock of the Stratix II GX devices.

Figure 11–15. Fast PLL and LVDS/DPA Clock for EP2SGX30C/D and EP2SGX60C/D Devices

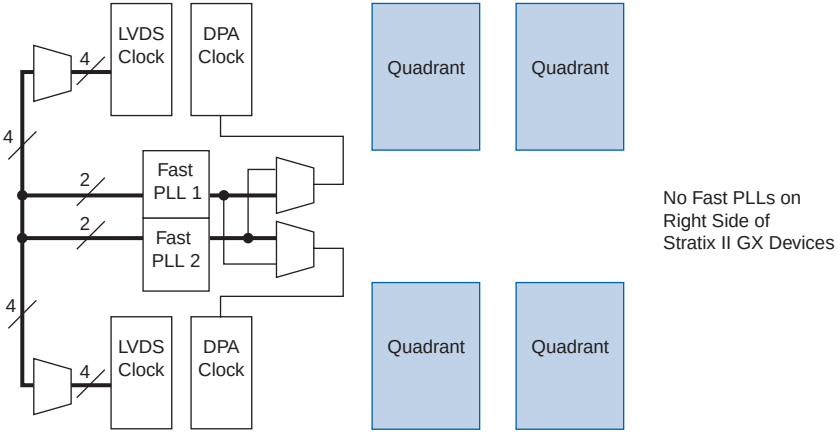
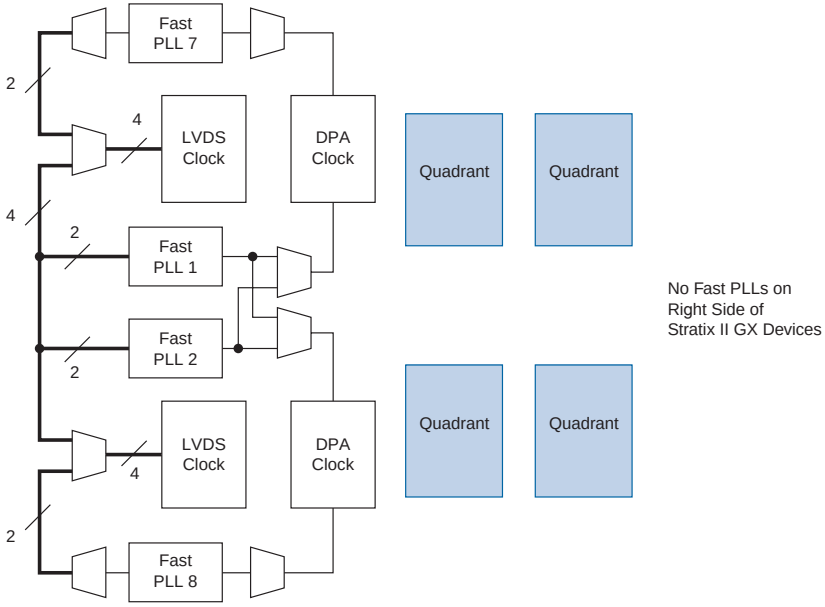


Figure 11–16. Fast PLL and LVDS/DPA Clocks for EP2SGX60E, EP2SGX90 and EP2SGX130 Devices



Source Synchronous Timing Budget

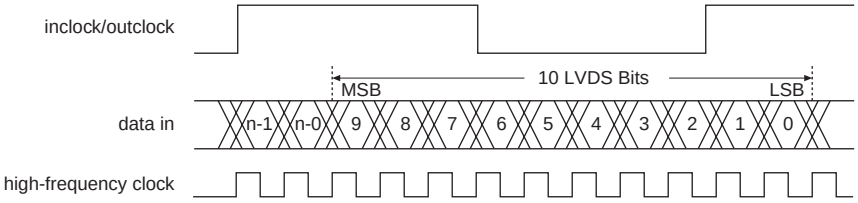
This section discusses the timing budget, waveforms, and specifications for source-synchronous signaling in Stratix II and Stratix II GX devices. LVDS and HyperTransport I/O standards enable high-speed data transmission. This high data transmission rate results in better overall system performance. To take advantage of fast system performance, it is important to understand how to analyze timing for these high-speed signals. Timing analysis for the differential block is different from traditional synchronous timing analysis techniques.

Rather than focusing on clock-to-output and setup times, source-synchronous timing analysis is based on the skew between the data and the clock signals. High-speed differential data transmission requires the use of timing parameters provided by IC vendors and is strongly influenced by board skew, cable skew, and clock jitter. This section defines the source-synchronous differential data orientation timing parameters, the timing budget definitions for Stratix II and Stratix II GX devices, and how to use these timing parameters to determine a design's maximum performance.

Differential Data Orientation

There is a set relationship between an external clock and the incoming data. For operation at 1 Gbps and SERDES factor of 10, the external clock is multiplied by 10, and phase-alignment can be set in the PLL to coincide with the sampling window of each data bit. The data is sampled on the falling edge of the multiplied clock. Figure 11–17 shows the data bit orientation of the ×10 mode.

Figure 11–17. Bit Orientation in the Quartus II Software



Differential I/O Bit Position

Data synchronization is necessary for successful data transmission at high frequencies. Figure 11–18 shows the data bit orientation for a channel operation. These figures are based on the following:

- SERDES factor equals clock multiplication factor
- Edge alignment is selected for phase alignment
- Implemented in hard SERDES

For other serialization factors use the Quartus II software tools and find the bit position within the word. The bit positions after deserialization are listed in Table 11–4.

Figure 11–18 also shows a functional waveform. Timing waveforms may produce different results. Altera recommends performing a timing simulation to predict actual device behavior.

Figure 11–18. Bit Order for One Channel of Differential Data

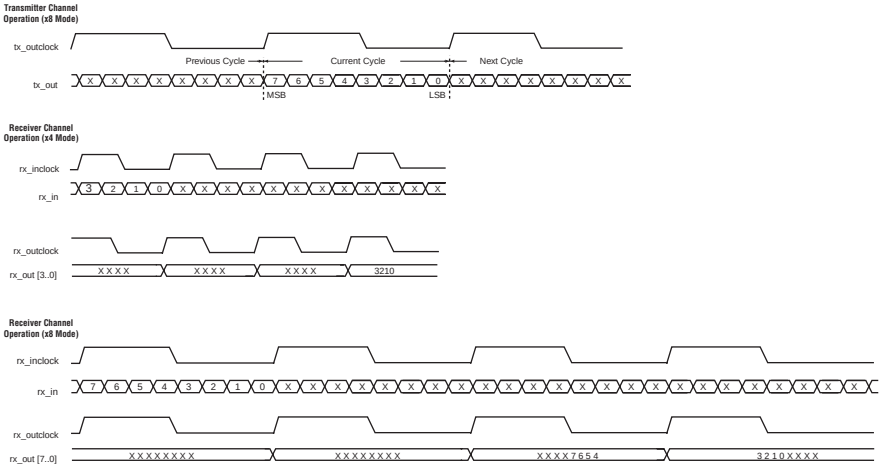


Table 11–4 shows the conventions for differential bit naming for 18 differential channels. The MSB and LSB positions increase with the number of channels used in a system.

Table 11–4. LVDS Bit Naming

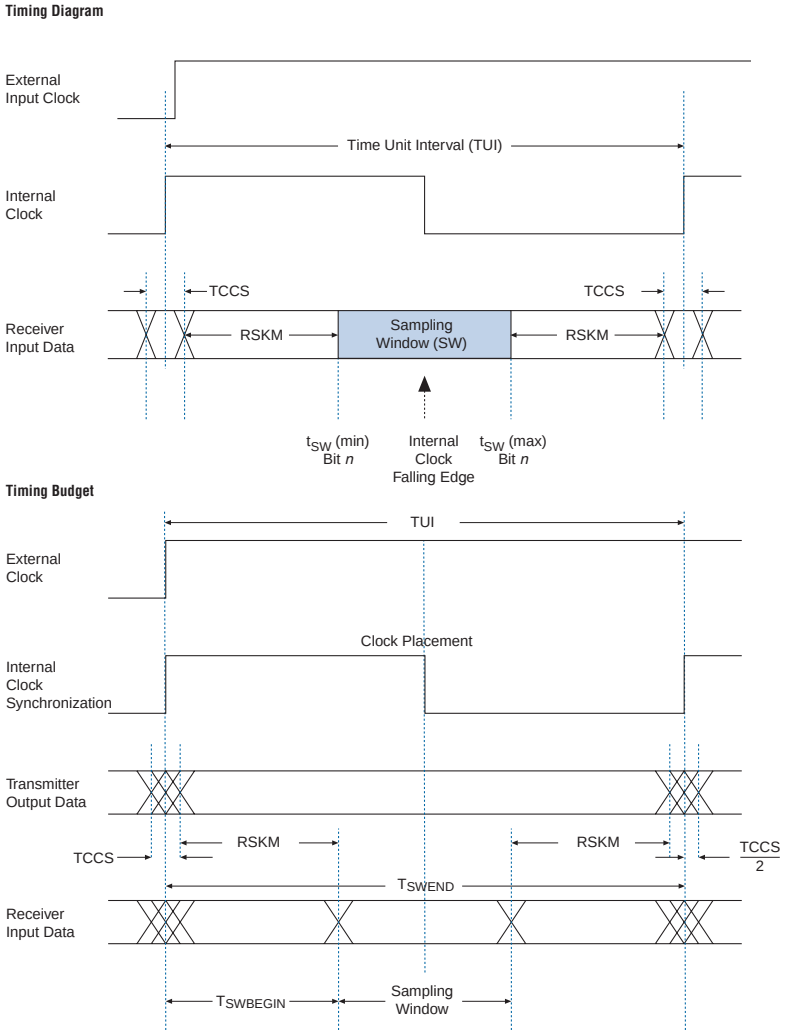
Receiver Channel Data Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
1	7	0
2	15	8
3	23	16
4	31	24
5	39	32
6	47	40
7	55	48
8	63	56
9	71	64
10	79	72
11	87	80
12	95	88
13	103	96
14	111	104
15	119	112
16	127	120
17	135	128
18	143	136

Receiver Skew Margin for Non-DPA

Changes in system environment, such as temperature, media (cable, connector, or PCB) loading effect, the receiver's setup and hold times, and internal skew, reduce the sampling window for the receiver. The timing margin between the receiver's clock input and the data input sampling window is called Receiver Skew Margin (RSKM). Figure 11–19 shows the relationship between the RSKM and the receiver's sampling window.

TCCS, RSKM, and the sampling window specifications are used for high-speed source-synchronous differential signals without DPA. When using DPA, these specifications are exchanged for the simpler single DPA jitter tolerance specification. For instance, the receiver skew is why each input with DPA selects a different phase of the clock, thus removing the requirement for this margin.

Figure 11–19. Differential High-Speed Timing Diagram and Timing Budget for Non-DPA



Differential Pin Placement Guidelines

In order to ensure proper high-speed operation, differential pin placement guidelines have been established. The Quartus II compiler automatically checks that these guidelines are followed and will issue an error message if these guidelines are not met. PLL driving distance information is separated into guidelines with and without DPA usage.

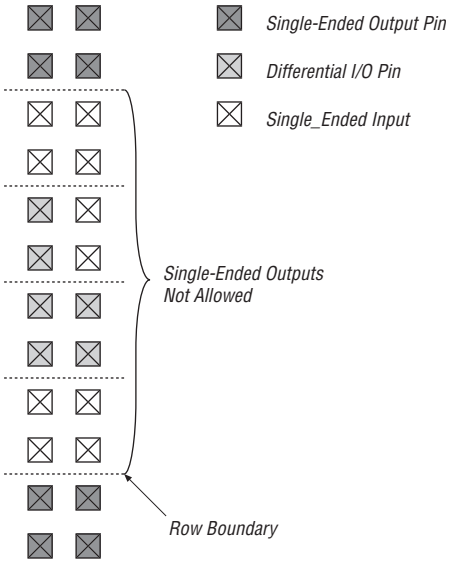
High-Speed Differential I/Os and Single-Ended I/Os

When a differential channel or channels of side banks are used (with or without DPA), you must adhere to the guidelines described in the following sections.

- Single-ended I/Os are allowed in the same bank as the LVDS channels (with or without DPA) as long as the single-ended I/O standard uses the same V_{CCIO} as the LVDS bank.
- Single-ended inputs can be in the same LAB row. Outputs cannot be on the same LAB row with LVDS I/Os. If input registers are used in the IOE, single-ended inputs cannot be in the same LAB row as an LVDS SERDES block.
- LVDS (non-SERDES) I/Os are allowed in the same row as LVDS SERDES but the use of IOE registers are not allowed.
- Single-ended outputs are limited to 120 mA drive strength on LVDS banks (with or without DPA).
  - LVTTL equation for maximum number of I/Os in an LVDS bank:
    - 120 mA = (number of LVTTL outputs) × (drive strength of each LVTTL output)
  - SSTL-2 equation:
    - 120 mA = (number of SSTL-2 I/Os) × (drive strength of each output) ÷ 2
  - LVTTL and SSTL-2 mix equation:
    - 120 mA= (total drive strength of all LVTTL outputs) + (total drive strength of all SSTL2 outputs) ÷ 2
- Single-ended inputs can be in the same LAB row as a differential channel using the SERDES circuitry; however, IOE input registers are not available for the single-ended I/Os placed in the same LAB row as differential I/Os. The same rule for input registers applies for non-SERDES differential inputs placed within the same LAB row as a SERDES differential channel. The input register must be implemented within the core logic. The same rule for input registers applies for non-SERDES differential inputs placed within the same LAB row as a SERDES differential channel.

- Single-ended output pins must be at least one LAB row away from differential output pins, as shown in Figure 11–20.

Figure 11–20. Single-Ended Output Pin Placement with Respect to Differential I/O Pins



DPA Usage Guidelines

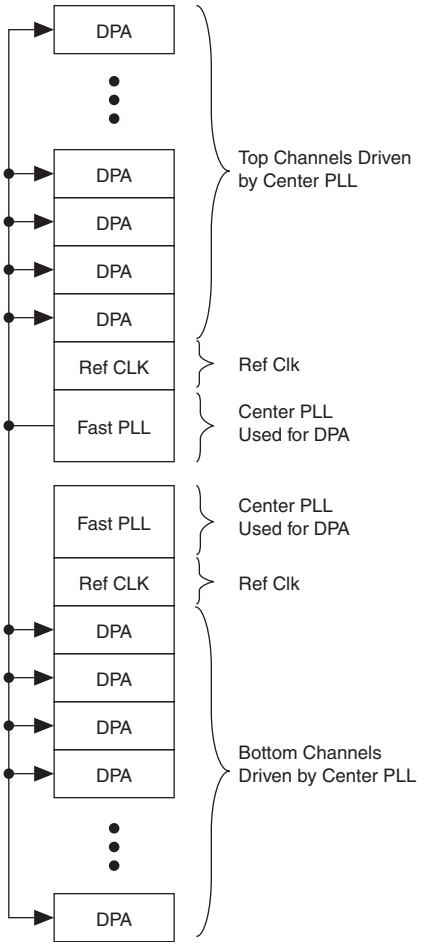
The Stratix II and Stratix II GX device have differential receivers and transmitters on the Row banks of the device. Each receiver has a dedicated DPA circuit to align the phase of the clock to the data phase of its associated channel. When a channel or channels of left or right banks are used in DPA mode, the guidelines listed below must be adhered to.

Fast PLL/DPA Channel Driving Distance

- Each fast PLL can drive up to 25 contiguous rows in DPA mode in a single bank (not including the reference clock row). The unbonded SERDES I/O rows are included in the 25 row calculation. These channels can be anywhere in the bank, their distance from the PLL is not relevant, but the channels must be within 25 rows of each other.

- Unused channels can be within the 25 row span, but all used channels must be in DPA mode from the same fast PLL. Center fast PLLs can drive two I/O banks simultaneously, up to 50 channels (25 on the upper bank and 25 on the lower bank) as shown in Figure 11–21.
- If one center fast PLL drives DPA channels in the upper and lower banks, the other center fast PLL cannot be used for DPA.

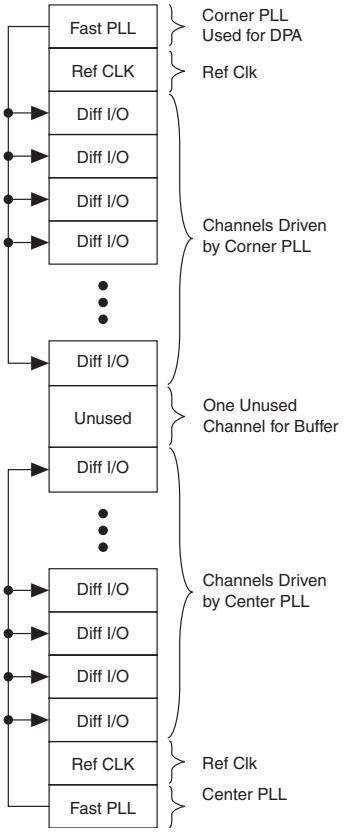
Figure 11–21. Driving Capabilities of a Center Fast PLL



Using Corner and Center Fast PLLs

- If a differential bank is being driven by two fast PLLs, where the corner PLL is driving one group and the center fast PLL is driving another group, there must be at least 1 row of separation between the two groups of DPA channels (see Figure 11–22). The two groups can operate at independent frequencies. Not all the channels are bonded out of the die. Each LAB row is considered a channel, whether or not it has I/O support.
- No separation is necessary if a single fast PLL is driving DPA channels as well as non-DPA channels as long as the DPA channels are contiguous.

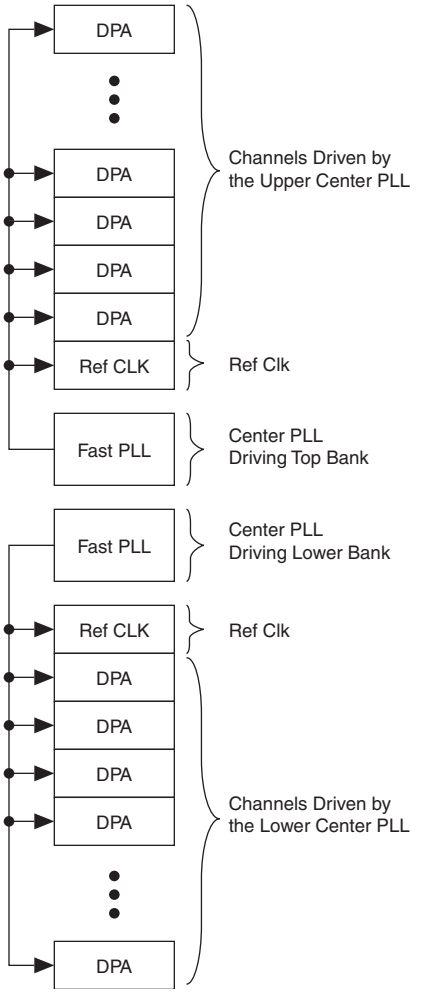
Figure 11–22. Usage of Corner and Center Fast PLLs Driving DPA Channels in a Single Bank



Using Both Center Fast PLLs

- Both center fast PLLs can be used for DPA as long as they drive DPA channels in their adjacent quadrant only. See Figure 11–23.
- Both center fast PLLs cannot be used for DPA if one of the fast PLLs drives the top and bottom banks, or if they are driving cross banks (e.g., the lower fast PLL drives the top bank and the top fast PLL drives the lower bank).

Figure 11–23. Center Fast PLL Usage When Driving DPA Channels



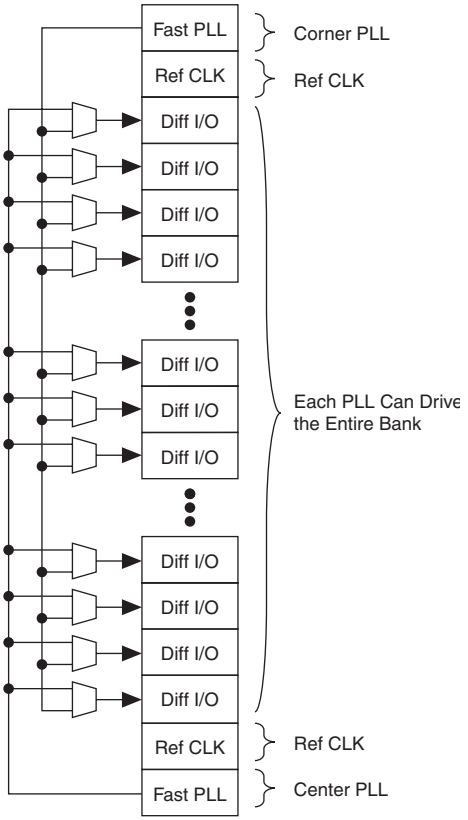
Non-DPA Differential I/O Usage Guidelines

When a differential channel or channels of left or right banks are used in non-DPA mode, you must adhere to the guidelines in the following sections.

Fast PLL/Differential I/O Driving Distance

- As shown in Figure 11–24, each fast PLL can drive all the channels in the entire bank.

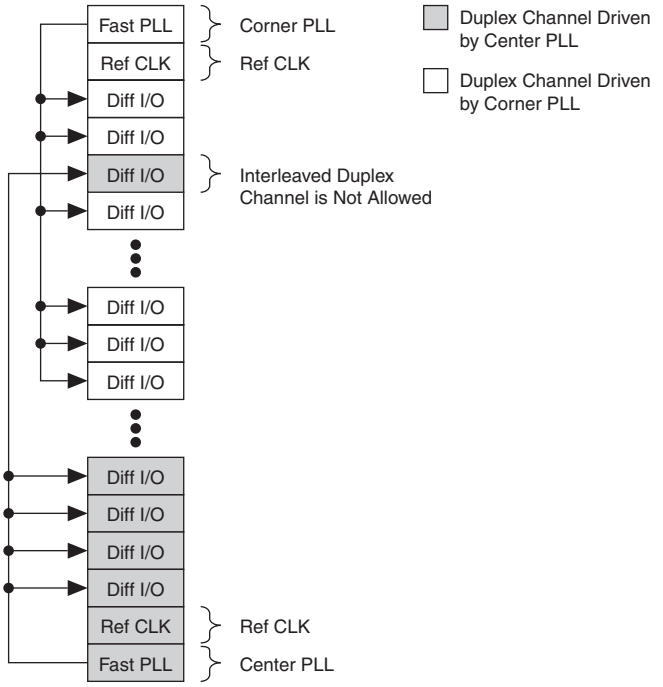
Figure 11–24. Fast PLL Driving Capability When Driving Non-DPA Differential Channels



Using Corner and Center Fast PLLs

- The corner and center fast PLLs can be used as long as the channels driven by separate fast PLLs do not have their transmitter or receiver channels interleaved. Figure 11–25 shows illegal placement of differential channels when using corner and center fast PLLs.
- If one fast PLL is driving transmitter channels only, and the other fast PLL drives receiver channels only, the channels driven by those fast PLLs can overlap each other.
- Center fast PLLs can be used for both transmitter and receiver channels.

Figure 11–25. Illegal Placement of Interlaced Duplex Channels in an I/O Bank



Board Design Considerations

This section explains how to achieve the optimal performance from the Stratix II and Stratix II GX high-speed I/O block and ensure first-time success in implementing a functional design with optimal signal quality.

For more information on board layout recommendations and I/O pin terminations, refer to *AN 224: High-Speed Board Layout Guidelines*.

To achieve the best performance from the device, pay attention to the impedances of traces and connectors, differential routing, and termination techniques.

Use this section together with the *Stratix II Device Family Data Sheet* in volume 1 of the *Stratix II Device Handbook*.

The Stratix II and Stratix II GX high-speed module generates signals that travel over the media at frequencies as high as one Gbps. Board designers should use the following guidelines:

- Base board designs on controlled differential impedance. Calculate and compare all parameters such as trace width, trace thickness, and the distance between two differential traces.
- Place external reference resistors as close to receiver input pins as possible.
- Use surface mount components.
- Avoid 90° or 45° corners.
- Use high-performance connectors such as HMZD or VHDM connectors for backplane designs. Two suppliers of high-performance connectors are Teradyne Corp (**www.teradyne.com**) and Tyco International Ltd. (**www.tyco.com**).
- Design backplane and card traces so that trace impedance matches the connector's or the termination's impedance.
- Keep an equal number of vias for both signal traces.
- Create equal trace lengths to avoid skew between signals. Unequal trace lengths also result in misplaced crossing points and system margins when the TCCS value increases.
- Limit vias, because they cause impedance discontinuities.
- Use the common bypass capacitor values such as 0.001, 0.01, and 0.1 µF to decouple the fast PLL power and ground planes. You can also use 0.0047 µF and 0.047 µF.
- Keep switching TTL signals away from differential signals to avoid possible noise coupling.
- Do not route TTL clock signals to areas under or above the differential signals.
- Route signals on adjacent layers orthogonally to each other.

Conclusion

Stratix II and Stratix II GX high-speed differential inputs and outputs, with their DPA and data realignment circuitry, allow users to build a robust multi-Gigabit system. The DPA circuitry allows users to compensate for any timing skews resulting from physical layouts. The data realignment circuitry allows the devices to align the data packet between the transmitter and receiver. Together with the on-chip differential termination, Stratix II and Stratix II GX devices can be used as a single-chip solution for high-speed applications.

Referenced Documents

This chapter references the following documents:

- *AN 224: High-Speed Board Layout Guidelines*
- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook*
- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook*
- *PLLs in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*
- *Selectable I/O standards in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook*
- *Stratix II Device Family Data Sheet* in volume 1 of the *Stratix II Device Handbook*

Document Revision History

Table 11–5 shows the revision history for this chapter.

Table 11–5. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v2.3	Updated Figure 11–2.	—
	Added "Referenced Documents" section.	—
	Minor text edits.	—
August 2007, v2.2	Added Figure 11–9.	—
	Updated "Receiver Data Realignment Circuit".	—
	For the *Stratix II GX Device Handbook* only: Formerly chapter 10. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter.	—
May 2007, v2.1	Updated entire chapter to include Stratix II GX information.	—
	Changed chapter part number.	—
	Fixed two types in "High-Speed Differential I/Os and Single-Ended I/Os" section	—

Table 11–5. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
February 2007 v2.0	This chapter changed from High-Speed, Source-Synchronous Differential I/O Interfaces in Stratix II GX Devices to "High-Speed Differential I/O Interfaces with DPA in Stratix II and Stratix II GX Devices".	—
	Added the "Document Revision History" section to this chapter.	—
	Added "and Stratix II GX" after each instance of "Stratix II".	—
	Updated Figures 10–4, 10–20, 10–22.	—
	Updated Note (4) of Figure 10–2.	—
	Updated Table 10–1.	—
	Updated the following sections: • "I/O Banks" • "Differential I/O Termination" • "Fast PLL " • "Differential I/O Bit Position" • "DPA Usage Guidelines" • "Fast PLL/DPA Channel Driving Distance"	—
	Updated Note (1) of Tables 10–2 and 10–3.	—
	Added Note (5) to Figure 10–11.	—
	Added Table 10–3.	—
	Added Figures 10–14, 10–15, 10–19.	—
	Deleted old section called High-Speed Differential I/Os and Single-Ended I/Os and added a new "High-Speed Differential I/Os and Single-Ended I/Os" section.	—
	Deleted DPA and Single-Ended I/Os section.	—
	Updated title and added Note (1) to Figure 10–12.	—
	Added Note (1) to Figure 10–13.	—
April 2006, v1.2	• Updated all the MegaWizard Plug-In Manager figures to match the Quartus II software GUI. • Updated "Dedicated Source-Synchronous Circuitry" section, including Table 10–3.	—
February 2006, v1.1	• Updated chapter number from 9 to 10. • Updated Figures 10–11 and 10–12.	—
October 2005 v1.0	Added chapter to the *Stratix II GX Device Handbook*.	—

Section V. Digital Signal Processing (DSP)

This section provides information for design and optimization of digital signal processing (DSP) functions and arithmetic operations in the on-chip DSP blocks.

This section contains the following chapter:

- Chapter 12, DSP Blocks in Stratix II & Stratix II GX Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

12. DSP Blocks in Stratix II & Stratix II GX Devices

SII52006-2.2

Introduction

Stratix® II and Stratix II GX devices have dedicated digital signal processing (DSP) blocks optimized for DSP applications requiring high data throughput. These DSP blocks combined with the flexibility of programmable logic devices (PLDs), provide you with the ability to implement various high performance DSP functions easily. Complex systems such as CDMA2000, voice over Internet protocol (VoIP), high-definition television (HDTV) require high performance DSP blocks to process data. These system designs typically use DSP blocks as finite impulse response (FIR) filters, complex FIR filters, fast Fourier transform (FFT) functions, discrete cosine transform (DCT) functions, and correlators.

Stratix II and Stratix II GX DSP blocks consist of a combination of dedicated blocks that perform multiplication, addition, subtraction, accumulation, and summation operations. You can configure these blocks to implement arithmetic functions like multipliers, multiply-adders and multiply-accumulators which are necessary for most DSP functions.

Along with the DSP blocks, the TriMatrix™ memory structures in Stratix II and Stratix II GX devices also support various soft multiplier implementations. The combination of soft multipliers and dedicated DSP blocks increases the number of multipliers available in Stratix II and Stratix II GX devices and provides you with a wide variety of implementation options and flexibility when designing your systems.

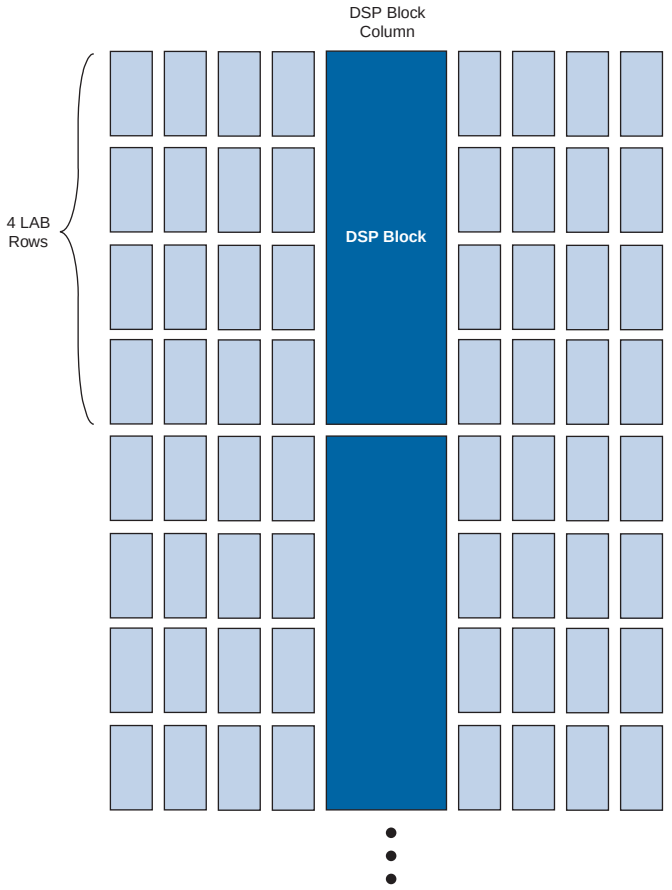
See the *Stratix II Device Family Data Sheet* in volume 1 of the *Stratix II Device Handbook* or the *Stratix II GX Device Family Data Sheet* in volume 1 of the *Stratix II GX Device Handbook* for more information on Stratix II and Stratix II GX devices, respectively.

DSP Block Overview

Each Stratix II and Stratix II GX device has two to four columns of DSP blocks that efficiently implement multiplication, multiply-accumulate (MAC) and multiply-add functions. Figure 12–1 shows the arrangement of one of the DSP block columns with the surrounding LABs. Each DSP block can be configured to support:

- Eight 9 × 9-bit multipliers
- Four 18 × 18-bit multipliers
- One 36 × 36-bit multiplier

Figure 12–1. DSP Blocks Arranged in Columns with Adjacent LABs



The multipliers then feed an adder or accumulator block within the DSP block. Stratix II and Stratix II GX device multipliers support rounding and saturation on Q1.15 input formats. The DSP block also has input registers that can be configured to operate in a shift register chain for efficient implementation of functions like FIR filters. The accumulator within the DSP block can be initialized to any value and supports rounding and saturation on Q1.15 input formats to the multiplier. A single DSP block can be broken down to operate different configuration modes simultaneously.

For more information on Q1.15 formatting, see "Saturation and Rounding" on page 12–13.

The number of DSP blocks per column and the number of columns available increases with device density. Table 12–1 shows the number of DSP blocks in each Stratix II device and the multipliers that you can implement.

Table 12–1. Number of DSP Blocks in Stratix II Devices *Note (1)*

Device	DSP Blocks	9 × 9 Multipliers	18 × 18 Multipliers	36 × 36 Multipliers
EP2S15	12	96	48	12
EP2S30	16	128	64	16
EP2S60	36	288	144	36
EP2S90	48	384	192	48
EP2S130	63	504	252	63
EP2S180	96	768	384	96

Note to Table 12–1:

(1) Each device has either the number of 9 × 9-, 18 × 18-, or 36 × 36-bit multipliers shown. The total number of multipliers for each device is not the sum of all the multipliers.

Table 12–2 shows the number of DSP blocks in each Stratix II GX device and the multipliers that you can implement.

Table 12–2. Number of DSP Blocks in Stratix II GX Devices *Note (1)*

Device	DSP Blocks	9 × 9 Multipliers	18 × 18 Multipliers	36 × 36 Multipliers
EP2SGX30C EP2SGX30D	16	128	64	16
EP2SGX60C EP2SGX60D EP2SGX60E	36	288	144	36
EP2SGX90E EP2SGX90F	48	384	192	48
EP2SGX130G	63	504	252	63

Note to Table 12–2:

(1) Each device has either the number of 9 × 9-, 18 × 18-, or 36 × 36-bit multipliers shown. The total number of multipliers for each device is not the sum of all the multipliers.

In addition to the DSP block multipliers, you can use the Stratix II or Stratix II GX device's TriMatrix memory blocks for soft multipliers. The availability of soft multipliers increases the number of multipliers available within the device. Table 12–3 shows the total number of multipliers available in Stratix II devices using DSP blocks and soft multipliers.

Table 12–3. Number of Multipliers in Stratix II Devices

Device	DSP Blocks (18 × 18)	Soft Multipliers (16 × 16) *(1)*, *(2)*	Total Multipliers *(3)*, *(4)*
EP2S15	48	100	148 (3.08)
EP2S30	64	189	253 (3.95)
EP2S60	144	325	469 (3.26)
EP2S90	192	509	701 (3.65)
EP2S130	252	750	1,002 (3.98)
EP2S130	384	962	1,346 (3.51)

Notes to Table 12–3:

(1) Soft multipliers implemented in sum of multiplication mode. RAM blocks are configured with 18-bit data widths and sum of coefficients up to 18-bits.

(2) Soft multipliers are only implemented in M4K and M512 TriMatrix memory blocks, not M-RAM blocks.

(3) The number in parentheses represents the increase factor, which is the total number of multipliers with soft multipliers divided by the number of 18 × 18 multipliers supported by DSP blocks only.

(4) The total number of multipliers may vary according to the multiplier mode used.

Table 12–4 shows the total number of multipliers available in Stratix II GX devices using DSP blocks and soft multipliers.

Table 12–4. Number of Multipliers in Stratix II GX Devices

Device	DSP Blocks (18 × 18)	Soft Multipliers (16 × 16) *(1)*, *(2)*	Total Multipliers *(3)*, *(4)*
EP2SGX30C EP2SGX30D	64	189	253 (3.95)
EP2SGX60C EP2SGX60D EP2SGX60E	144	325	469 (3.26)
EP2SGX90E EP2SGX90F	192	509	701 (3.65)
EP2SGX130G	252	750	1,002 (3.98)

Notes to Table 12–4:

(1) Soft multipliers implemented in sum of multiplication mode. RAM blocks are configured with 18-bit data widths and sum of coefficients up to 18-bits.

(2) Soft multipliers are only implemented in M4K and M512 TriMatrix memory blocks, not M-RAM blocks.

(3) The number in parentheses represents the increase factor, which is the total number of multipliers with soft multipliers divided by the number of 18 × 18 multipliers supported by DSP blocks only.

(4) The total number of multipliers may vary according to the multiplier mode used.

Refer to the *Stratix II Architecture* chapter in volume 1 of the *Stratix II Device Handbook* or the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook* for more information on Stratix II or Stratix II GX TriMatrix memory blocks. Refer to *AN 306: Implementing Multipliers in FPGA Devices* for more information on soft multipliers.

Figure 12–2 shows the DSP block configured for 18 × 18 multiplier mode. Figure 12–3 shows the 9 × 9 multiplier configuration of the DSP block.

Figure 12–2. DSP Block in 18 × 18 Mode

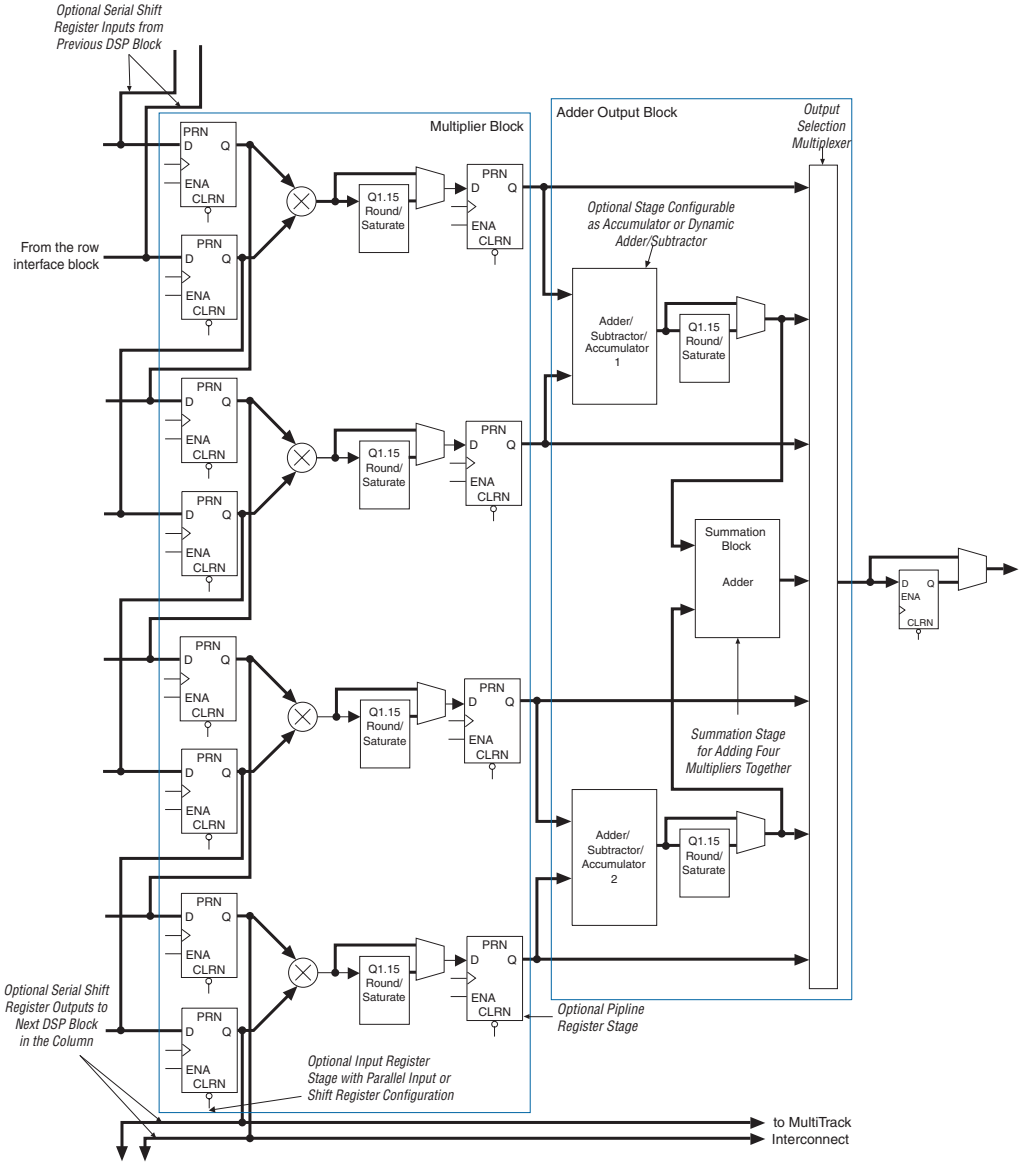
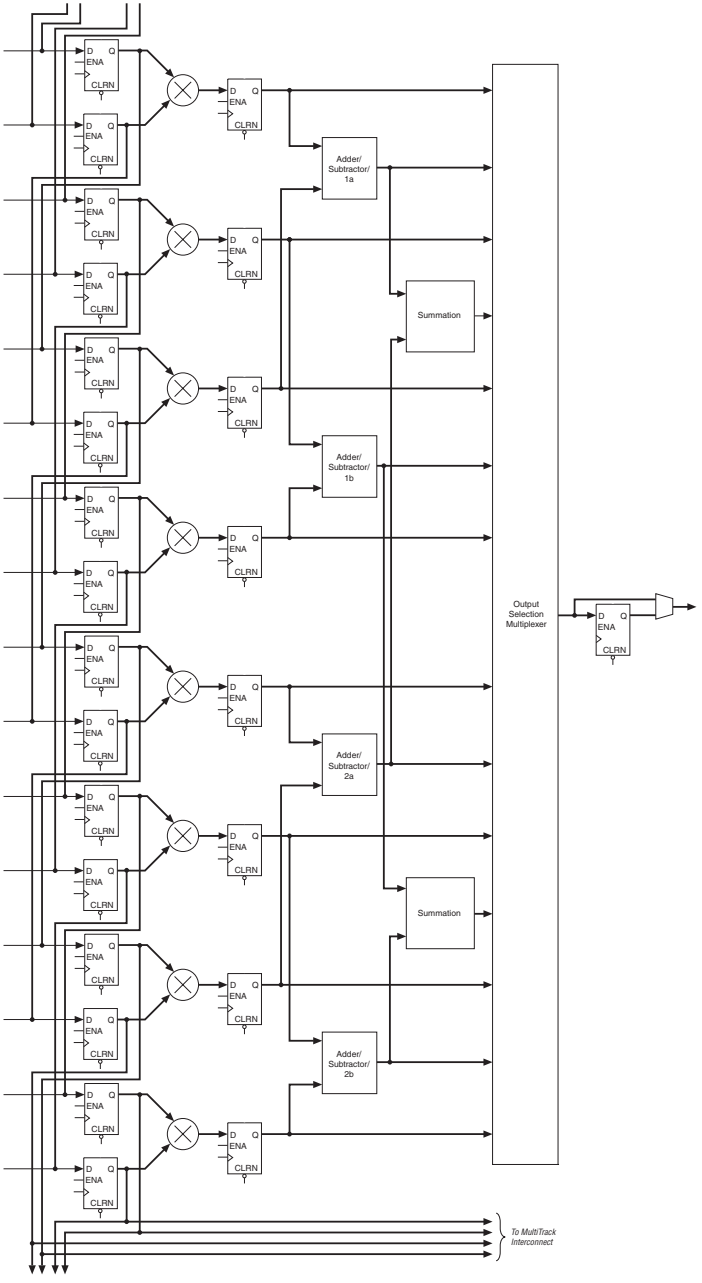


Figure 12–3. DSP Block in 9 × 9 Mode



Architecture

The DSP block consists of the following elements:

- A multiplier block
- An adder/subtractor/accumulator block
- A summation block
- Input and output interfaces
- Input and output registers

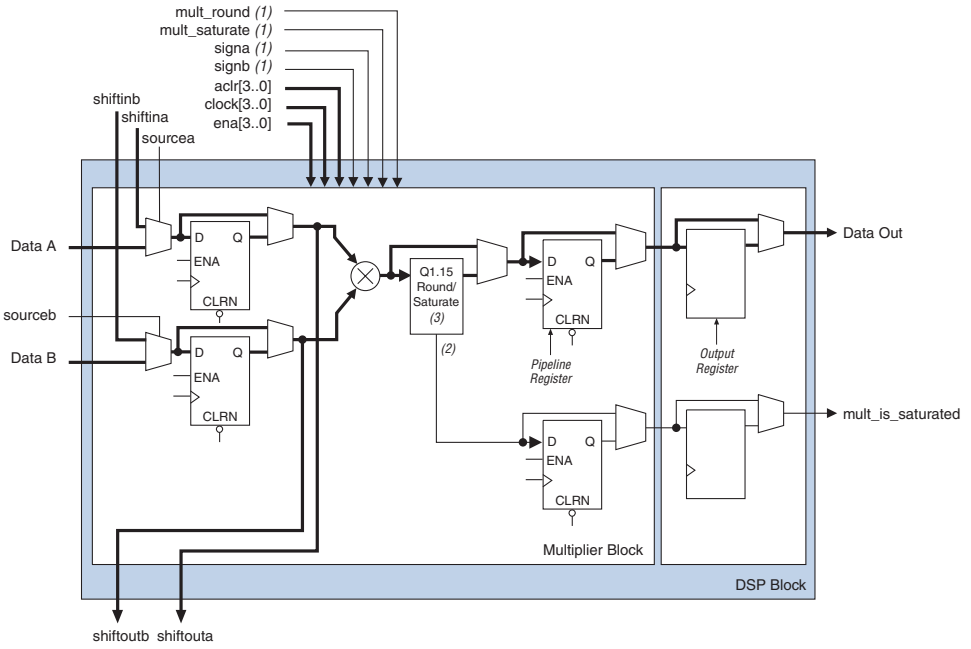
Multiplier Block

Each multiplier block has the following elements:

- Input registers
- A multiplier block
- A rounding and/or saturation stage for Q1.15 input formats
- A pipeline output register

Figure 12–4 shows the multiplier block architecture.

Figure 12–4. Multiplier Block Architecture



Notes to Figure 12–4:

(1) These signals are not registered or registered once to match the data path pipeline.

(2) You can send these signals through either one or two pipeline registers.

(3) The rounding and/or saturation is only supported in 18 × 18-bit signed multiplication for Q1.15 inputs.

Input Registers

Each multiplier operand can feed an input register or directly to the multiplier. The following DSP block signals control each input register within the DSP block:

- `clock[3..0]`
- `ena[3..0]`
- `aclr[3..0]`

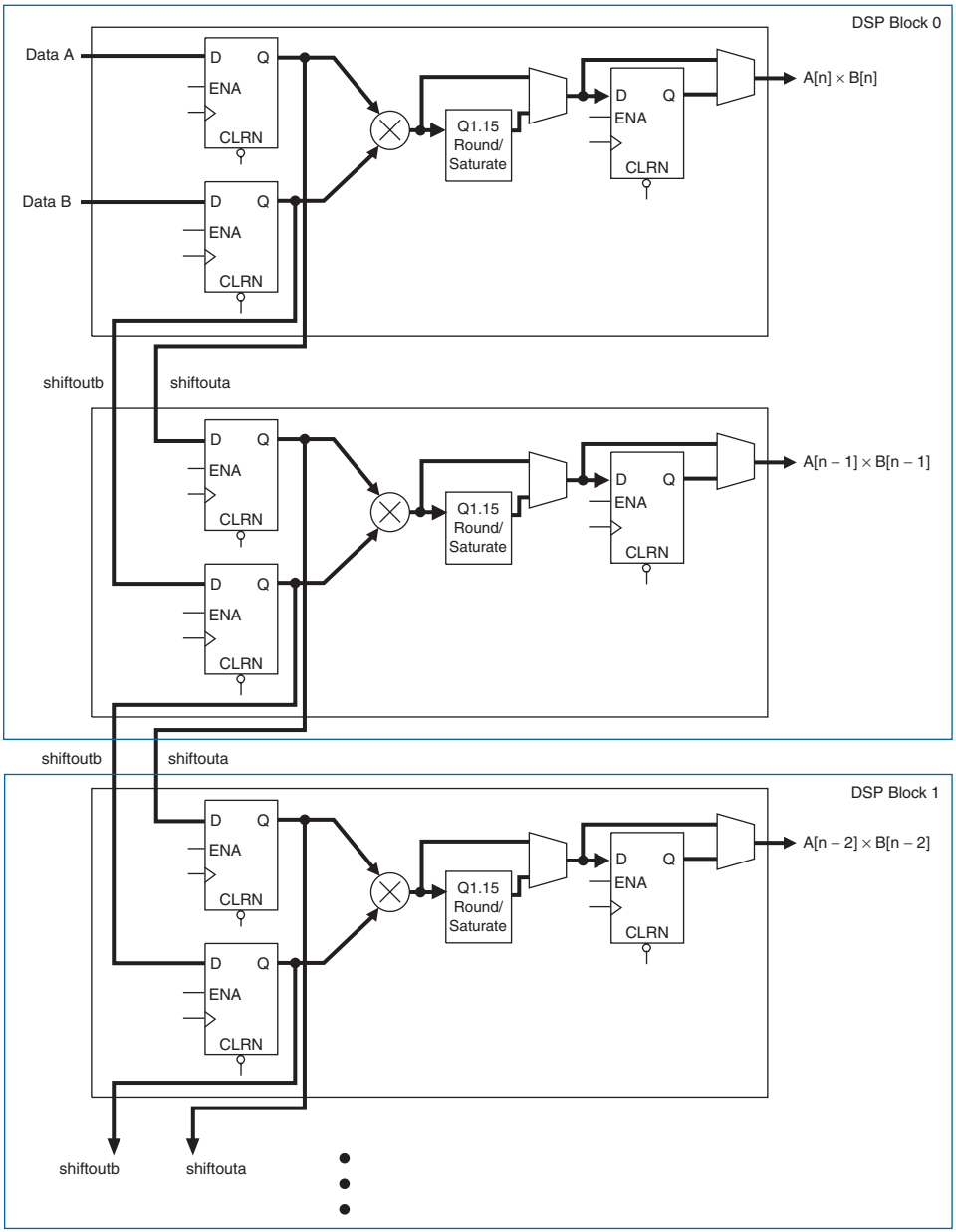
The input registers feed the multiplier and drive two dedicated shift output lines, `shiftouta` and `shiftoutb`. The dedicated shift outputs from one multiplier block directly feed input registers of the adjacent multiplier below it within the same DSP block or the first multiplier in the next DSP block to form a shift register chain, as shown in Figure 12–5. The

dedicated shift register chain spans a single column but longer shift register chains requiring multiple columns can be implemented using regular FPGA routing resources. Therefore, this shift register chain can be of any length up to 768 registers in the largest member of the Stratix II or Stratix II GX device family.

Shift registers are useful in DSP functions like FIR filters. When implementing 9 × 9 and 18 × 18 multipliers, you do not need external logic to create the shift register chain because the input shift registers are internal to the DSP block. This implementation significantly reduces the LE resources required, avoids routing congestion, and results in predictable timing.

Stratix II and Stratix II GX DSP blocks allow you to dynamically select whether a particular multiplier operand is fed by regular data input or the dedicated shift register input using the `sourcea` and `sourceb` signals. A logic 1 value on the `sourcea` signal indicates that data A is fed by the dedicated scan-chain; a logic 0 value indicates that it is fed by regular data input. This feature allows the implementation of a dynamically loadable shift register where the shift register operates normally using the scan-chains and can also be loaded dynamically in parallel using the data input value.

Figure 12–5. Shift Register Chain *Note (1)*



Note to Figure 12–5:
(1) Either Data A or Data B input can be set to a parallel input for constant coefficient multiplication.

Table 12–5 shows the summary of input register modes for the DSP block.

Table 12–5. Input Register Modes

Register Input Mode	9 × 9	18 × 18	36 × 36
Parallel input	✓	✓	✓
Shift register input	✓	✓	

Multiplier Stage

The multiplier stage supports 9 × 9, 18 × 18, or 36 × 36 multipliers as well as other smaller multipliers in between these configurations. See "Operational Modes" on page 12–21 for details. Depending on the data width of the multiplier, a single DSP block can perform many multiplications in parallel.

Each multiplier operand can be a unique signed or unsigned number. Two signals, `signa` and `signb`, control the representation of each operand respectively. A logic 1 value on the `signa` signal indicates that data A is a signed number while a logic 0 value indicates an unsigned number. Table 12–6 shows the sign of the multiplication result for the various operand sign representations. The result of the multiplication is signed if any one of the operands is a signed value.

Table 12–6. Multiplier Sign Representation

Data A (signa Value)	Data B (signb Value)	Result
Unsigned (logic 0)	Unsigned (logic 0)	Unsigned
Unsigned (logic 0)	Signed (logic 1)	Signed
Signed (logic 1)	Unsigned (logic 0)	Signed
Signed (logic 1)	Signed (logic 1)	Signed

There is only one `signa` and one `signb` signal for each DSP block. Therefore, all of the data A inputs feeding the same DSP block must have the same sign representation. Similarly, all of the data B inputs feeding the same DSP block must have the same sign representation. The multiplier offers full precision regardless of the sign representation.

When the `signa` and `signb` signals are unused, the Quartus® II software sets the multiplier to perform unsigned multiplication by default.

Saturation and Rounding

The DSP blocks have hardware support to perform optional saturation and rounding after each 18 × 18 multiplier for Q1.15 input formats.

Designs must use 18 × 18 multipliers for the saturation and rounding options because the Q1.15 input format requires 16-bit input widths.

Q1.15 input format multiplication requires signed multipliers. The most significant bit (MSB) in the Q1.15 input format represents the value's sign bit. Use signed multipliers to ensure the proper sign extension during multiplication.

The Q1.15 format uses 16 bits to represent each fixed point input. The MSB is the sign bit, and the remaining 15-bits are used to represent the value after the decimal place (or the fractional value). This Q1.15 value is equivalent to an integer number representation of the 16-bits divided by 2^{15}, as shown in the following equations.

$$-\frac{1}{2} = 1\ 100\ 0000\ 0000\ 0000 = -\frac{0x4000}{2^{15}}$$

$$\frac{1}{8} = 0\ 001\ 0000\ 0000\ 0000 = \frac{0x1000}{2^{15}}$$

All Q1.15 numbers are between –1 and 1.

When performing multiplication, even though the Q1.15 input only uses 16 of the 18 multiplier inputs, the entire 18-bit input bus is transmitted to the multiplier. This is like a 1.17 input, where the two least significant bits (LSBs) are always 0.

The multiplier output will be a 2.34 value (36 bits total) before performing any rounding or saturation. The two MSBs are sign bits. Since the output only requires one sign bit, you can ignore one of the two MSBs, resulting in a Q1.34 value before rounding or saturation.

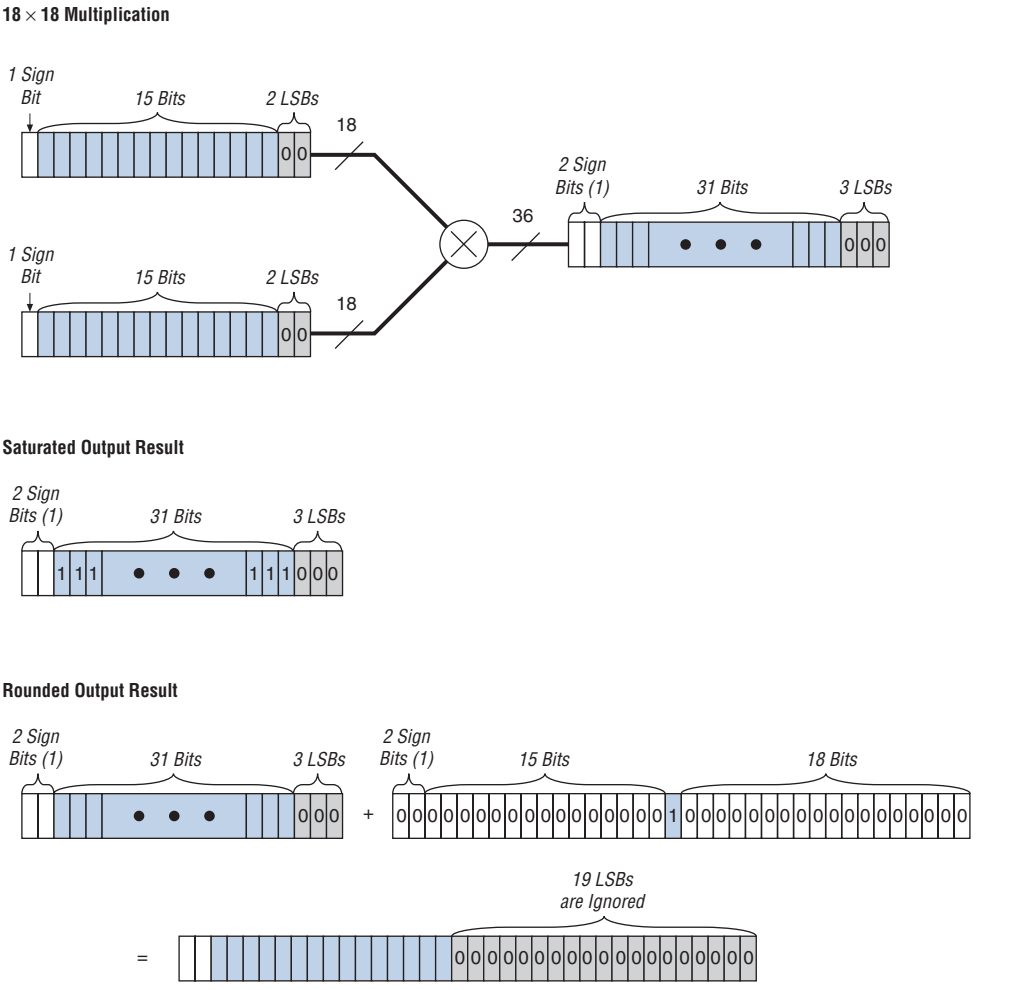
When the design performs saturation, the multiplier output gets saturated to 0x7FFFFFFF in a 1.31 format. This uses bits `[34..3]` of the overall 36-bit multiplier output. The three LSBs are set to 0.

The DSP block obtains the `mult_is_saturated` or `accum_is_saturated` overflow signal value from the LSB of the multiplier or accumulator output. Therefore, whenever saturation occurs, the LSB of the multiplier or accumulator output will send a `1` to the

mult_is_saturated or accum_is_saturated overflow signal. At all other times, this overflow signal is 0 when saturation is enabled or reflects the value of the LSB of the multiplier or accumulator output.

When the design performs rounding, it adds 0x00008000 in 1.31 format to the multiplier output, and it only uses bits [34..15] of the overall 36-bit multiplier output. Adding 0x00008000 in 1.31 format to the 36-bit multiplier result is equivalent to adding 0x0 0004 0000 in 2.34 format. The 16 LSBs are set to 0. Figure 12–6 shows which bits are used when the design performs rounding and saturation for the multiplication.

Figure 12–6. Rounding and Saturation Bits



Note to Figure 12–6:
(1) Both sign bits are the same. The design only uses one sign bit, and the other one is ignored.

If the design performs a `multiply_accumulate` or `multiply_add` operation, the multiplier output is input to the adder/subtractor/accumulator blocks as a 2.31 value, and the three LSBs are 0.

Pipeline Registers

The output from the multiplier can feed a pipeline register or this register can be bypassed. Pipeline registers may be implemented for any multiplier size and increase the DSP block's maximum performance, especially when using the subsequent DSP block adder stages. Pipeline registers split up the long signal path between the adder/subtractor/accumulator block and the adder/output block, creating two shorter paths.

Adder/Output Block

The adder/output block has the following elements:

- An adder/subtractor/accumulator block
- A summation block
- An output select multiplexer
- Output registers

Figure 12–7 shows the adder/output block architecture.

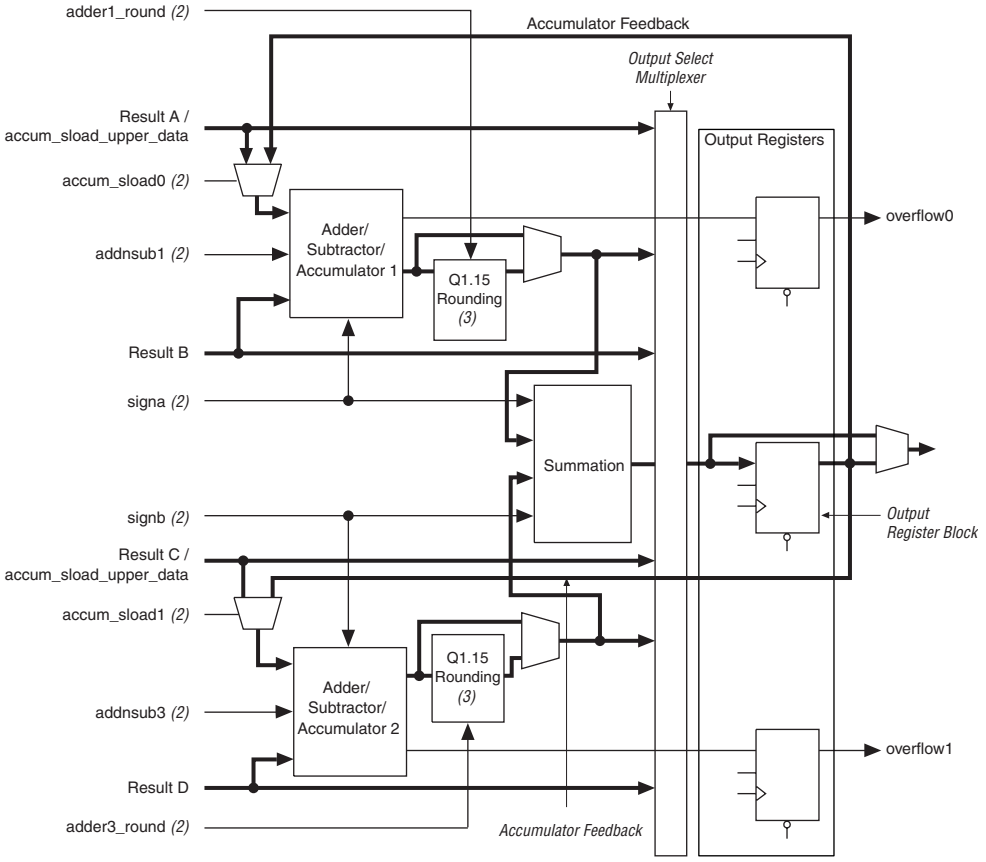
The adder/output block can be configured as:

- An output interface
- An accumulator which can be optionally loaded
- A one-level adder
- A two-level adder with dynamic addition/subtraction control on the first-level adder
- The final stage of a 36-bit multiplier, 9 × 9 complex multiplier, or 18 × 18 complex multiplier

The output select multiplexer sets the output configuration of the DSP block. The output registers can be used to register the output of the adder/output block.

The adder/output block cannot be used independently from the multiplier.

Figure 12–7. Adder/Output Block Architecture *Note (1)*



Notes to Figure 12–7:

(1) The adder/output block is in 18 × 18 mode. In 9 × 9 mode, there are four adder/subtractor blocks and two summation blocks.

(2) You can send these signals through a pipeline register. The pipeline length can be set to 1 or 2.

(3) Q1.15 inputs are not available in 9 × 9 or 36 × 36 modes.

Adder/Subtractor/Accumulator Block

The adder/subtractor/accumulator block is the first level adder stage of the adder/output block. This block can be configured as an accumulator or as an adder/subtractor.

Accumulator

When the adder/subtractor/accumulator is configured as an accumulator, the output of the adder/output block feeds back to the accumulator as shown in Figure 12–7. The accumulator can be set up to perform addition only, subtraction only or the addnsub signal can be used to dynamically control the accumulation direction. A logic 1 value on the addnsub signal indicates that the accumulator is performing addition while a logic 0 value indicates subtraction.

Each accumulator can be cleared by either clearing the DSP block output register or by using the accum_sload signal. The accumulator clear using the accum_sload signal is independent from the resetting of the output registers so the accumulation can be cleared and a new one can begin without losing any clock cycles. The accum_sload signal controls a feedback multiplexer that specifies that the output of the multiplier should be summed with a zero instead of the accumulator feedback path.

The accumulator can also be initialized/preloaded with a non-zero value using the accum_sload signal and the accum_sload_upper_data bus with one clock cycle latency. Preloading the accumulator is done by adding the result of the multiplier with the value specified on the accum_sload_upper_data bus. As in the case of the accumulator clearing, the accum_sload signal specifies to the feedback multiplexer that the accum_sload_upper_data signal should feed the accumulator instead of the accumulator feedback signal. The accum_sload_upper_data signal only loads the upper 36-bits of the accumulator. To load the entire accumulator, the value for the lower 16-bits must be sent through the multiplier feeding that accumulator with the multiplier set to perform a multiplication by one.

The overflow signal will go high on the positive edge of the clock when the accumulator detects an overflow or underflow. The overflow signal will stay high for only one clock cycle after an overflow or underflow is detected even if the overflow or underflow condition is still present. A latch external to the DSP block has to be used to preserve the overflow signal indefinitely or until the latch is cleared.

The DSP blocks support Q1.15 input format saturation and rounding in each accumulator. The following signals are available that can control if saturation or rounding or both is performed to the output of the accumulator:

- accum_round
- accum_saturation
- accum_is_saturated output

Each DSP block has two sets of accum_round and accum_saturation signals which control if rounding or saturation is performed on the accumulator output respectively (one set of signals for each accumulator). Rounding and saturation of the accumulator output is only available when implementing an 16 × 16 multiplier-accumulator to conform to the bit widths required for Q1.15 input format computation. A logic 1 value on the accum_round and accum_saturation signal indicates that rounding or saturation is performed while a logic 0 indicates that no rounding or saturation is performed. A logic 1 value on the accum_is_saturated output signal tells you that saturation has occurred to the result of the accumulator.

Figure 12–10 shows the DSP block configured to perform multiplier-accumulator operations.

Adder/Subtractor

The addnsub1 or addnsub3 signals specify whether you are performing addition or subtraction. A logic 1 value on the addnsub1 or addnsub3 signals indicates that the adder/subtractor is performing addition while a logic 0 value indicates subtraction. These signals can be dynamically controlled using logic external to the DSP block. If the first stage is configured as a subtractor, the output is A – B and C – D.

The adder/subtractor block share the same signa and signb signals as the multiplier block. The signa and signb signals can be pipelined with a latency of one or two clock cycles or not.

The DSP blocks support Q1.15 input format rounding (not saturation) after each adder/subtractor. The addnsub1_round and addnsub3_round signals determine if rounding is performed to the output of the adder/subtractor.

The addnsub1_round signal controls the rounding of the top adder/subtractor and the addnsub3_round signal controls the rounding of the bottom adder/subtractor. Rounding of the adder output is only available when implementing an 16 × 16 multiplier-adder to conform to the bit widths required for Q1.15 input format computation. A logic 1 value on the addnsub_round signal indicates that rounding is performed while a logic 0 indicates that no rounding is performed.

Summation Block

The output of the adder/subtractor block feeds an optional summation block, which is an adder block that sums the outputs of both adder/subtractor blocks. The summation block is used when more than two multiplier results are summed. This is useful in applications such as FIR filtering.

Output Select Multiplexer

The outputs of the different elements of the adder/output block are routed through an output select multiplexer. Depending on the operational mode of the DSP block, the output multiplexer selects whether the outputs of the DSP blocks comes from the outputs of the multiplier block, the outputs of the adder/subtractor/accumulator, or the output of the summation block. The output select multiplier configuration is set automatically by software, based on the DSP block operational mode you specify.

Output Registers

You can use the output registers to register the DSP block output. The following signals can control each output register within the DSP block:

- `clock[3..0]`
- `ena[3..0]`
- `aclr[3..0]`

The output registers can be used in any DSP block operational mode.

The output registers form part of the accumulator in the multiply-accumulate mode.

Refer to the *Stratix II Architecture* chapter in volume 1 of the *Stratix II Device Handbook* or the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook* for more information on the DSP block routing and interface.

Operational Modes

The DSP block can be used in one of four basic operational modes, or a combination of two modes, depending on the application needs. Table 12–7 shows the four basic operational modes and the number of multipliers that can be implemented within a single DSP block depending on the mode.

Table 12–7. DSP Block Operational Modes

Mode	Number of Multipliers		
	9 × 9	18 × 18	36 × 36
Simple multiplier	Eight multipliers with eight product outputs	Four multipliers with four product outputs	One multiplier
Multiply accumulate	-	Two 52-bit multiply-accumulate blocks	-
Two-multiplier adder	Four two-multiplier adder (two 9 × 9 complex multiply)	Two two-multiplier adder (one 18 × 18 complex multiply)	-
Four-multiplier adder	Two four-multiplier adder	One four-multiplier adder	-

The Quartus II software includes megafunctions used to control the mode of operation of the multipliers. After you make the appropriate parameter settings using the megafunction's MegaWizard® Plug-In Manager, the Quartus II software automatically configures the DSP block.

Stratix II and Stratix II GX DSP blocks can operate in different modes simultaneously. For example, a single DSP block can be broken down to operate a 9 × 9 multiplier as well as an 18 × 18 multiplier-adder where both multiplier's input a and input b have the same sign representations. This increases DSP block resource efficiency and allows you to implement more multipliers within a Stratix II or Stratix II GX device. The Quartus II software automatically places multipliers that can share the same DSP block resources within the same block.

Additionally, you can set up each Stratix II or Stratix II GX DSP block to dynamically switch between the following three modes:

- Up to four 18-bit independent multipliers
- Up to two 18-bit multiplier-accumulators
- One 36-bit multiplier

Each half of a Stratix II or Stratix II GX DSP block has separate mode control signals, which allows you to implement multiple 18-bit multipliers or multiplier-accumulators within the same DSP block and dynamically switch them independently (if they are in separate DSP block halves). If the design requires a 36-bit multiplier, you must switch the entire DSP block to accommodate the it since the multiplier requires the entire DSP block. The smallest input bit width that supports dynamic mode switching is 18 bits.

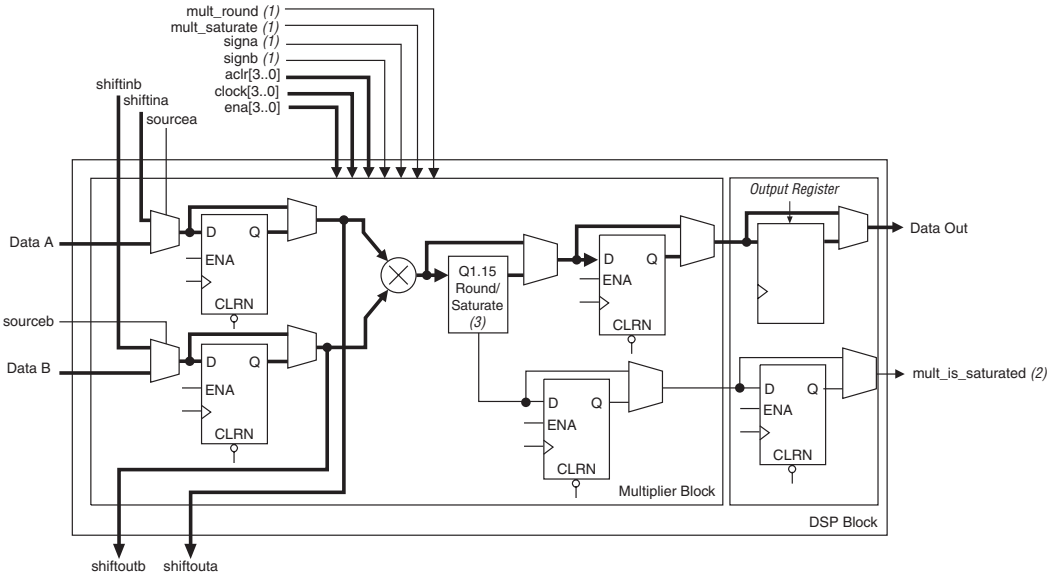
Simple Multiplier Mode

In simple multiplier mode, the DSP block performs individual multiplication operations for general-purpose multipliers and for applications such as computing equalizer coefficient updates which require many individual multiplication operations.

9- and 18-Bit Multipliers

Each DSP block multiplier can be configured for 9- or 18-bit multiplication. A single DSP block can support up to eight individual 9 × 9 multipliers or up to four individual 18 × 18 multipliers. For operand widths up to 9-bits, a 9 × 9 multiplier will be implemented and for operand widths from 10- to 18-bits, an 18 × 18 multiplier will be implemented. Figure 12–8 shows the DSP block in the simple multiplier operation mode.

Figure 12–8. Simple Multiplier Mode



Notes to Figure 12–8:

(1) These signals are not registered or registered once to match the data path pipeline.

(2) This signal has the same latency as the data path.

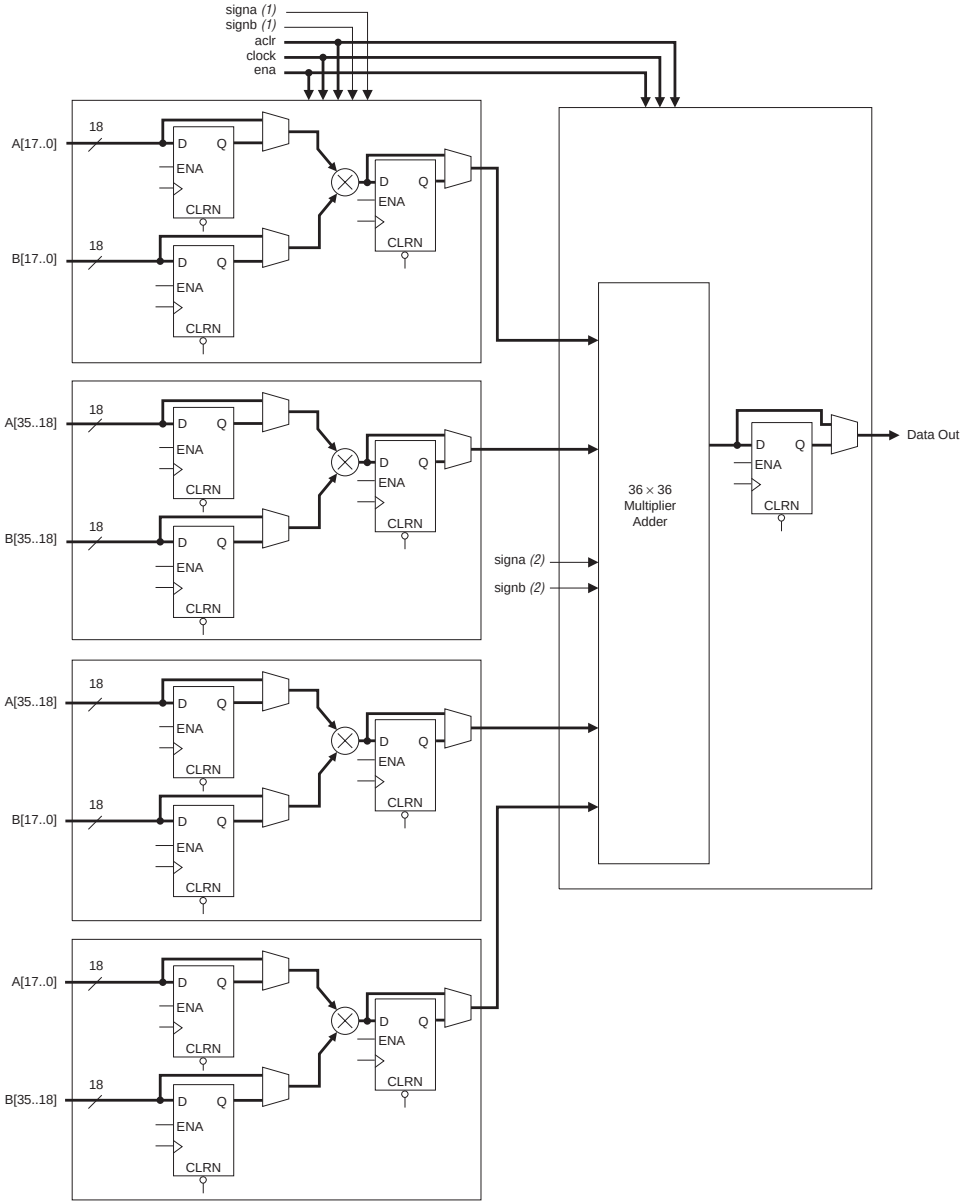
(3) The rounding and saturation is only supported in 18- × 18-bit signed multiplication for Q1.15 inputs.

The multiplier operands can accept signed integers, unsigned integers or a combination of both. The `signa` and `signb` signals can be changed dynamically and can be registered in the DSP block. Additionally, the multiplier inputs and result can be registered independently. The pipeline registers within the DSP block can be used to pipeline the multiplier result, increasing the performance of the DSP block.

36-Bit Multiplier

The 36-bit multiplier is also a simple multiplier mode but uses the entire DSP block, including the adder/output block to implement the 36 × 36-bit multiplication operation. The device inputs 18-bit sections of the 36-bit input into the four 18-bit multipliers. The adder/output block adds the partial products obtained from the multipliers using the summation block. Pipeline registers can be used between the multiplier stage and the summation block to speed up the multiplication. The 36 × 36-bit multiplier supports signed, unsigned as well as mixed sign multiplication. Figure 12–9 shows the DSP block configured to implement a 36-bit multiplier.

Figure 12–9. 36-Bit Multiplier



Notes to Figure 12–9:

(1) These signals are either not registered or registered once to match the pipeline.

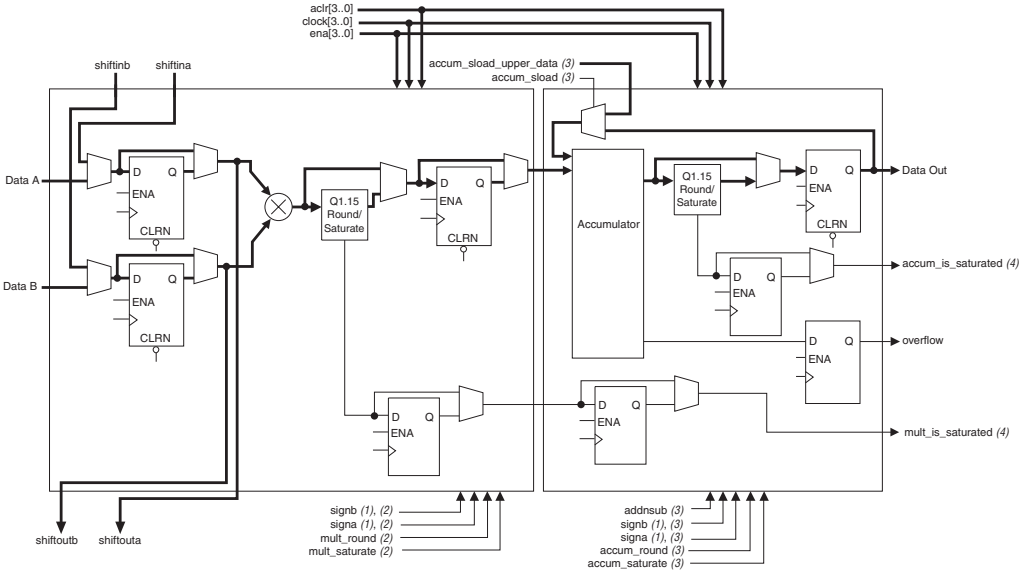
(2) These signals are either not registered, registered once, or registered twice to match the data path pipeline.

The 36-bit multiplier is useful for applications requiring more than 18-bit precision, for example, for mantissa multiplication of precision floating-point arithmetic applications.

Multiply Accumulate Mode

In multiply accumulate mode, the output of the multiplier stage feeds the adder/output block which is configured as an accumulator or subtractor. Figure 12–10 shows the DSP block configured to operate in multiply accumulate mode.

Figure 12–10. Multiply Accumulate Mode



Notes to Figure 12–10:

(1) The `signa` and `signb` signals are the same in the multiplier stage and the adder/output block.
(2) These signals are not registered or registered once to match the data path pipeline.
(3) You can send these signals through either one or two pipeline registers.
(4) These signals match the latency of the data path.

A single DSP block can implement up to two independent 18-bit multiplier accumulators. The Quartus II software implements smaller multiplier accumulators by tying the unused lower-order bits of the 18-bit multiplier to ground.

The multiplier accumulator output can be up to 52-bits wide to account for a 36-bit multiplier result with 16-bits of accumulation. In this mode, the DSP block uses output registers and the `accum_sload` and overflow

signals. The accum_sload signal can be used to clear the accumulator so that a new accumulation operation can begin without losing any clock cycles. This signal can be unregistered or registered once or twice. The accum_sload signal can also be used to preload the accumulator with a value specified on the accum_sload_upper_data signal with a one clock cycle penalty. The accum_sload_upper_data signal only loads the upper 36-bits (bits [51..16] of the accumulator). To load the entire accumulator, the value for the lower 16-bits (bits [15..0]) must be sent through the multiplier feeding that accumulator with the multiplier set to perform a multiplication by one. Bits [17..16] are overlapped by both the accum_sload_upper_data signal and the multiplier output. Either one of these signals can be used to load bits [17..16].

The overflow signal indicates an overflow or underflow in the accumulator. This signal gets updated every clock cycle due to a new accumulation operation every cycle. To preserve the signal, an external latch can be used. The addnsub signal can be used to specify if an accumulation or subtraction is performed dynamically.

The DSP block can implement just an accumulator (without multiplication) by specifying a multiply by one at the multiplier stage followed by an accumulator to force the Quartus II software to implement the function within the DSP block.

Multiply Add Mode

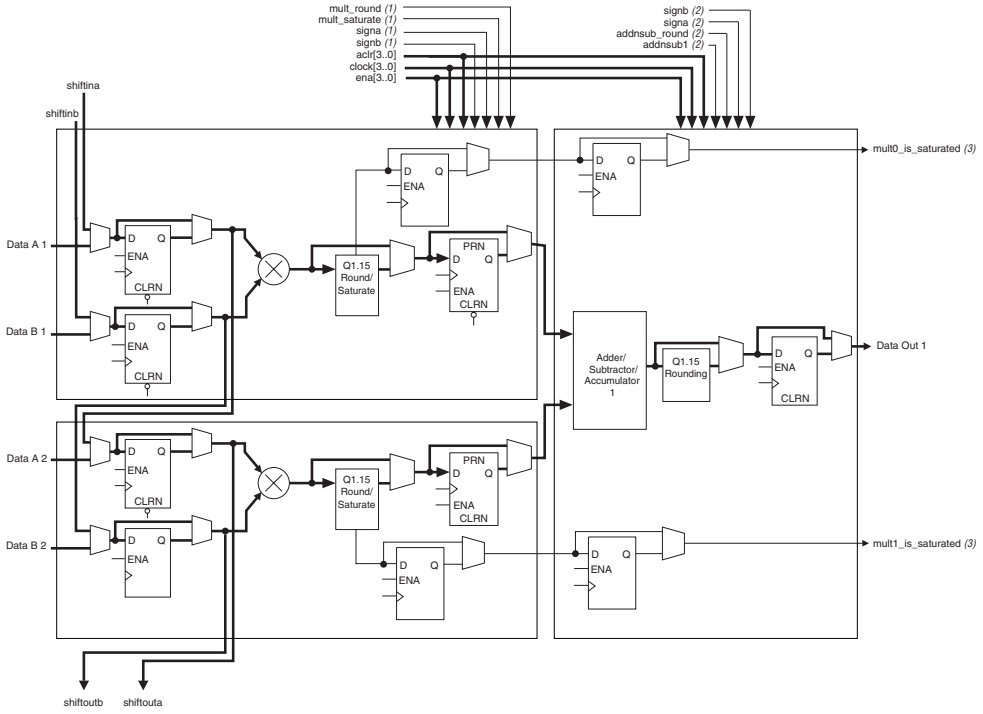
In multiply add mode, the output of the multiplier stage feeds the adder/output block which is configured as an adder or subtractor to sum or subtract the outputs of two or more multipliers. The DSP block can be configured to implement either a two-multiply add (where the outputs of two multipliers are added/subtracted together) or a four-multiply add function (where the outputs of four multipliers are added or subtracted together).

The adder block within the DSP block can only be used if it follows multiplication operations.

Two-Multiplier Adder

In the two-multiplier adder configuration, the DSP block can implement four 9-bit or smaller multiplier adders or two 18-bit multiplier adders. The adders can be configured to take the sum of both multiplier outputs or the difference of both multiplier outputs. You have the option to vary the summation/subtraction operation dynamically. These multiply add functions are useful for applications such as FFTs and complex FIR filters. Figure 12–11 shows the DSP block configured in the two-multiplier adder mode.

Figure 12–11. Two-Multiplier Adder Mode



Notes to Figure 12–11:

(1) These signals are not registered or registered once to match the data path pipeline.
(2) You can send these signals through a pipeline register. The pipeline length can be set to 1 or 2.
(3) These signals match the latency of the data path.

Complex Multiply

The DSP block can be configured to implement complex multipliers using the two-multiplier adder mode. A single DSP block can implement one 18 × 18-bit complex multiplier or two 9 × 9-bit complex multipliers.

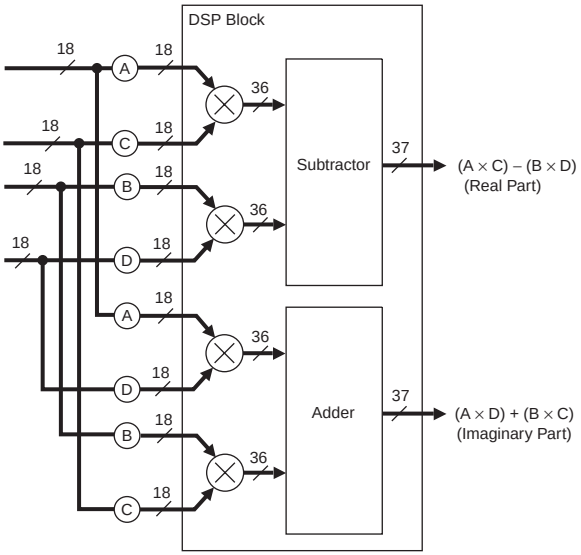
A complex multiplication can be written as:

$$(a + jb) \times (c + jd) = ((a \times c) - (b \times d)) + j\,((a \times d) + (b \times c))$$

To implement this complex multiplication within the DSP block, the real part $((a \times c) - (b \times d))$ is implemented using two multipliers feeding one subtractor block while the imaginary part $((a \times d) + (b \times c))$ is implemented using another two multipliers feeding an adder block, for data up to 18-bits. Figure 12–12 shows an 18-bit complex multiplication. For data widths up to 9-bits, a DSP block can perform two separate complex

multiplication operations using eight 9-bit multipliers feeding four adder/subtractor/accumulator blocks. Resources external to the DSP block must be used to route the correct real and imaginary input components to the appropriate multiplier inputs to perform the correct computation for the complex multiplication operation.

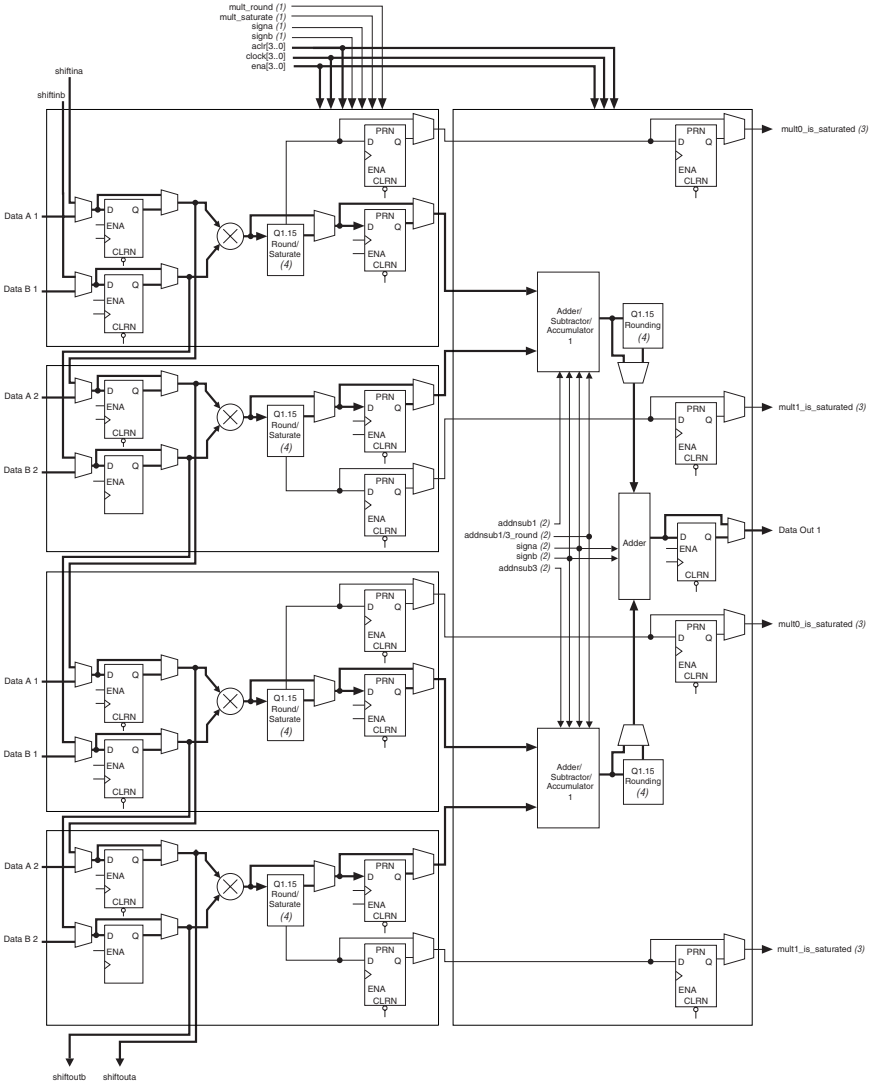
Figure 12–12. Complex Multiplier Using Two-Multiplier Adder Mode



Four-Multiplier Adder

In the four-multiplier adder configuration, the DSP block can implement one 18 × 18 or two individual 9 × 9 multiplier adders. These modes are useful for implementing one-dimensional and two-dimensional filtering applications. The four-multiplier adder is performed in two addition stages. The outputs of two of the four multipliers are initially summed in the two first-stage adder/subtractor/accumulator blocks. The results of these two adder/subtractor/accumulator blocks are then summed in the final stage summation block to produce the final four-multiplier adder result. Figure 12–13 shows the DSP block configured in the four-multiplier adder mode.

Figure 12–13. Four-Multiplier Adder Mode



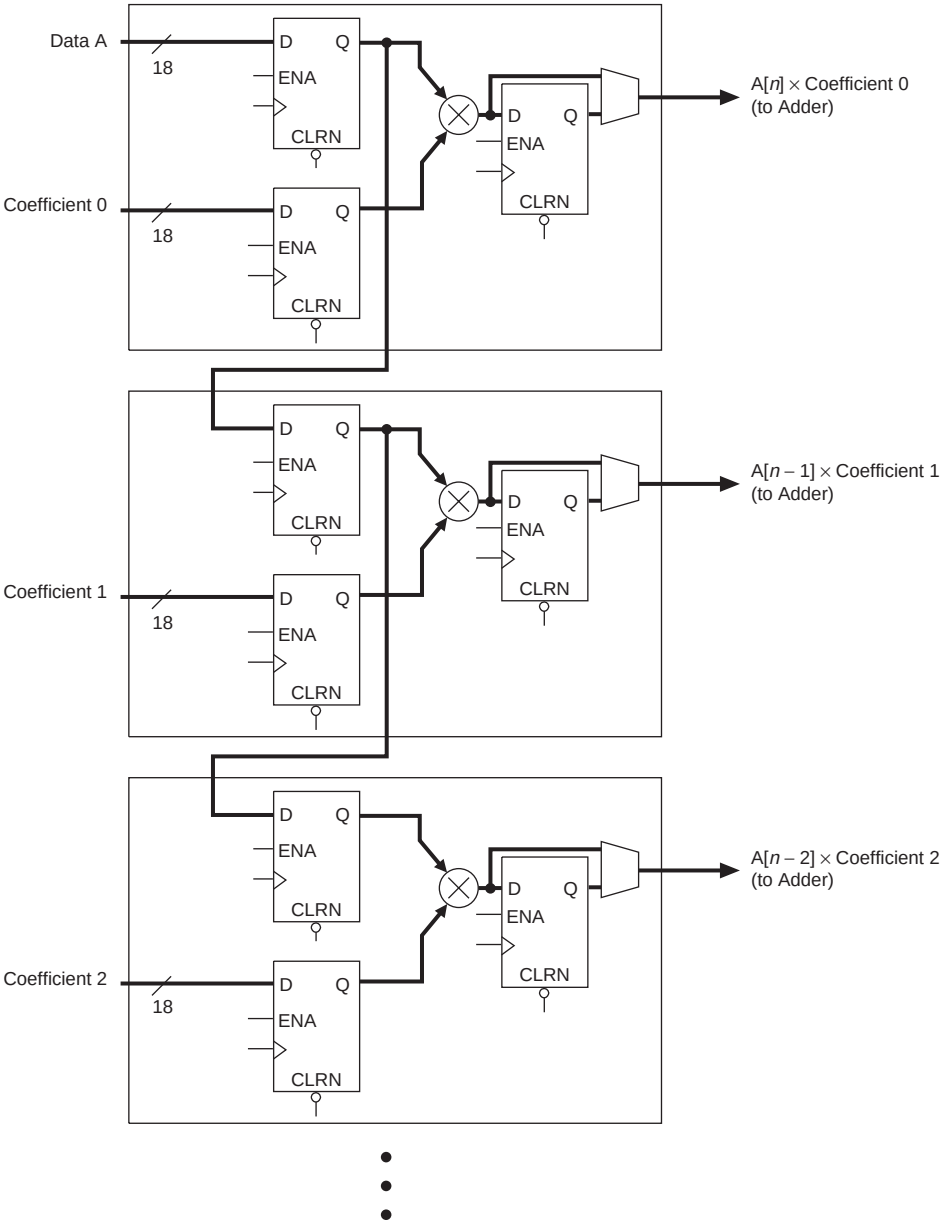
Notes to Figure 12–13:

(1) These signals are not registered or registered once to match the data path pipeline.
(2) You should send these signals through the pipeline register to match the latency of the data path.
(3) These signals match the latency of the data path.
(4) The rounding and saturation is only supported in 18- × 18-bit signed multiplication for Q1.15 inputs.

FIR Filter

The four-multiplier adder mode can be used to implement FIR filter and complex FIR filter applications. To do this, the DSP block is set up in a four-multiplier adder mode with one set of input registers configured as shift registers using the dedicated shift register chain. The set of input registers configured as shift registers will contain the input data while the inputs configured as regular inputs will hold the filter coefficients. Figure 12–14 shows the DSP block configured in the four-multiplier adder mode using input shift registers.

Figure 12–14. FIR Filter Implemented Using the Four-Multiplier Adder Mode with Input Shift Registers



The built-in input shift register chain within the DSP block eliminates the need for shift registers externally to the DSP block in logic elements (LEs). This architecture feature simplifies the filter design and improves the filter performance because all the filter circuitry is localized within the DSP block.

Input shift registers for the 36-bit simple multiplier mode have to be implemented using external registers to the DSP block.

A single DSP block can implement a four tap 18-bit FIR filter. For filters larger than four taps, the DSP blocks can be cascaded with additional adder stages implemented using LEs.

Software Support

Altera provides two distinct methods for implementing various modes of the DSP block in your design: instantiation and inference. Both methods use the following three Quartus II megafunctions:

- `lpm_mult`
- `altmult_add`
- `altmult_accum`

You can instantiate the megafunctions in the Quartus II software to use the DSP block. Alternatively, with inference, you can create a HDL design an synthesize it using a third-party synthesis tool like LeonardoSpectrum or Synplify or Quartus II Native Synthesis that infers the appropriate megafunction by recognizing multipliers, multiplier adders, and multiplier accumulators. Using either method, the Quartus II software maps the functionality to the DSP blocks during compilation.

See Quartus II On-Line Help for instructions on using the megafunctions and the MegaWizard Plug-In Manager.

For more information, see the Synthesis section in *Design and Synthesis* (volume 1) of the *Quartus II Development Software Handbook*.

Conclusion

The Stratix II and Stratix II GX device DSP blocks are optimized to support DSP applications requiring high data throughput such as FIR filters, FFT functions and encoders. These DSP blocks are flexible and can be configured to implement one of several operational modes to suit a particular application. The built-in shift register chain, adder/subtractor/accumulator block and the summation block minimizes the amount of external logic required to implement these functions, resulting in efficient resource utilization and improved performance and data throughput for DSP applications. The Quartus II

software, together with the LeonardoSpectrum™ and Synplify software provide a complete and easy-to-use flow for implementing these multiplier functions in the DSP blocks.

Referenced Documents

This chapter references the following documents:

- *AN 306: Implementing Multipliers in FPGA Devices*
- *Design and Synthesis* (volume 1) of the *Quartus II Development Software Handbook*
- *Stratix II Architecture* chapter in volume 1 of the *Stratix II Device Handbook*
- *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *Stratix II Device Family Data Sheet* in volume 1 of the *Stratix II Device Handbook*
- *Stratix II GX Device Family Data Sheet* in volume 1 of the *Stratix II GX Device Handbook*

Document Revision History

Table 12–8 shows the revision history for this chapter.

Table 12–8. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007 v2.2	Added the "Referenced Documents" section.	—
	Minor text edits.	—
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 11. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
February 2007 v2.1	Added the "Document Revision History" section to this chapter.	—
No change	Formerly chapter 10. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
October 2005 v2.0	Added chapter to the *Stratix II GX Device Handbook*.	—

Section VI. Configuration & Remote System Upgrades

This section provides configuration information for all of the supported configuration schemes for Stratix® II GX devices. These configuration schemes use either a microprocessor, configuration device, or download cable. There is detailed information on how to design with Altera enhanced configuration devices which includes information on how to manage multiple configuration files and access the on-chip FLASH memory space. The last chapter shows designers how to perform remote and local upgrades for their designs.

This section contains the following chapters:

- Chapter 13, Configuring Stratix II & Stratix II GX Devices
- Chapter 14, Remote System Upgrades with Stratix II & Stratix II GX Devices
- Chapter 15, IEEE 1149.1 (JTAG) Boundary-Scan Testing for Stratix II & Stratix II GX Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

13. Configuring Stratix II & Stratix II GX Devices

SII52007-4.5

Introduction

Stratix® II and Stratix II GX devices use SRAM cells to store configuration data. Because SRAM memory is volatile, configuration data must be downloaded to Stratix II and Stratix II GX devices each time the device powers up. Stratix II and Stratix II GX devices can be configured using one of five configuration schemes: the fast passive parallel (FPP), active serial (AS), passive serial (PS), passive parallel asynchronous (PPA), and Joint Test Action Group (JTAG) configuration schemes. All configuration schemes use either an external controller (for example, a MAX® II device or microprocessor) or a configuration device.

Configuration Devices

The Altera enhanced configuration devices (EPC16, EPC8, and EPC4) support a single-device configuration solution for high-density devices and can be used in the FPP and PS configuration schemes. They are ISP-capable through its JTAG interface. The enhanced configuration devices are divided into two major blocks, the controller and the flash memory.

For information on enhanced configuration devices, refer to the *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* in volume 2 of the *Configuration Handbook*.

The Altera serial configuration devices (EPCS64, EPCS16, and EPCS4) support a single-device configuration solution for Stratix II and Stratix II GX devices and are used in the AS configuration scheme. Serial configuration devices offer a low cost, low pin count configuration solution.

For information on serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

The EPC2 configuration devices provide configuration support for the PS configuration scheme. The EPC2 device is ISP-capable through its JTAG interface. The EPC2 device can be cascaded to hold large configuration files.

For more information on EPC2 configuration devices, refer to the *Configuration Devices for SRAM-Based LUT Devices Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

The configuration scheme is selected by driving the Stratix II or Stratix II GX device MSEL pins either high or low as shown in Table 13–1. The MSEL pins are powered by the V_{CCIO} power supply of the bank they reside in. The MSEL[3..0] pins have 9-kΩ internal pull-down resistors that are always active. During power-on reset (POR) and during reconfiguration, the MSEL pins have to be at LVTTL V_{IL} and V_{IH} levels to be considered a logic low and logic high.

To avoid any problems with detecting an incorrect configuration scheme, hard-wire the MSEL[] pins to V_{CCPD} and GND, without any pull-up or pull-down resistors. Do not drive the MSEL[] pins by a microprocessor or another device.

Table 13–1. Stratix II and Stratix II GX Configuration Schemes (Part 1 of 2)

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0
Fast passive parallel (FPP)	0	0	0	0
Passive parallel asynchronous (PPA)	0	0	0	1
Passive serial (PS)	0	0	1	0
Remote system upgrade FPP *(1)*	0	1	0	0
Remote system upgrade PPA *(1)*	0	1	0	1
Remote system upgrade PS *(1)*	0	1	1	0
Fast AS (40 MHz) *(2)*	1	0	0	0
Remote system upgrade fast AS (40 MHz) *(2)*	1	0	0	1
FPP with decompression and/or design security feature enabled *(3)*	1	0	1	1
Remote system upgrade FPP with decompression and/or design security feature enabled *(1)*, *(3)*	1	1	0	0
AS (20 MHz) *(2)*	1	1	0	1
Remote system upgrade AS (20 MHz) *(2)*	1	1	1	0
JTAG-based configuration *(5)*	*(4)*	*(4)*	*(4)*	*(4)*

Table 13–1. Stratix II and Stratix II GX Configuration Schemes (Part 2 of 2)				
Configuration Scheme	**MSEL3**	**MSEL2**	**MSEL1**	**MSEL0**

Notes to Table 13–1:
(1) These schemes require that you drive the RUnLU pin to specify either remote update or local update. For more information about remote system upgrades in Stratix II devices, refer to the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.
(2) Only the EPCS16 and EPCS64 devices support up to a 40 MHz DCLK. Other EPCS devices support up to a 20 MHz DCLK. Refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* for more information.
(3) These modes are only supported when using a MAX II device or a microprocessor with flash memory for configuration. In these modes, the host system must output a DCLK that is 4× the data rate.
(4) Do not leave the MSEL pins floating. Connect them to V_{CCPD} or ground. These pins support the non-JTAG configuration scheme used in production. If only JTAG configuration is used, you should connect the MSEL pins to ground.
(5) JTAG-based configuration takes precedence over other configuration schemes, which means MSEL pin settings are ignored.

Stratix II and Stratix II GX devices offer design security, decompression, and remote system upgrade features. Design security using configuration bitstream encryption is available in Stratix II and Stratix II GX devices, which protects your designs. Stratix II and Stratix II GX devices can receive a compressed configuration bit stream and decompress this data in real-time, reducing storage requirements and configuration time. You can make real-time system upgrades from remote locations of your Stratix II and Stratix II GX designs with the remote system upgrade feature.

Table 13–2 and Table 13–3 show the uncompressed configuration file sizes for Stratix II and Stratix II GX devices, respectively.

Table 13–2. Stratix II Uncompressed .rbf Sizes Notes (1), (2)		
Device	**Data Size (Bits)**	**Data Size (MBytes)**
EP2S15	4,721,544	0.590
EP2S30	9,640,672	1.205
EP2S60	16,951,824	2.119
EP2S90	25,699,104	3.212
EP2S130	37,325,760	4.666
EP2S180	49,814,760	6.227

Notes to Table 13–2:
(1) These values are final.
(2) **.rbf**: Raw Binary File.

Table 13–3. Stratix II GX Uncompressed .rbf Sizes Note (1)		
Device	**Data Size (Bits)**	**Data Size (MBytes)**
EP2SGX30C EP2SGX30D	9,640,672	1.205
EP2SGX60C EP2SGX60D EP2SGX60E	16,951,824	2.119
EP2SGX90E EP2SGX90F	25,699,104	3.212
EP2SGX130G	37,325,760	4.666

Note to Table 13–3:
(1) **.rbf**: Raw Binary File.

Use the data in Table 13–2 to estimate the file size before design compilation. Different configuration file formats, such as a Hexidecimal (**.hex**) or Tabular Text File (**.ttf**) format, will have different file sizes. However, for any specific version of the Quartus® II software, any design targeted for the same device will have the same uncompressed configuration file size. If you are using compression, the file size can vary after each compilation because the compression ratio is dependent on the design.

This chapter explains the Stratix II and Stratix II GX device configuration features and describes how to configure Stratix II and Stratix II GX devices using the supported configuration schemes. This chapter provides configuration pin descriptions and the Stratix II and Stratix II GX device configuration file formats. In this chapter, the generic term device(s) includes all Stratix II and Stratix II GX devices.

For more information on setting device configuration options or creating configuration files, refer to *Software Settings* in volume 2 of the *Configuration Handbook*.

Configuration Features

Stratix II and Stratix II GX devices offer configuration data decompression to reduce configuration file storage, design security using data encryption to protect your designs, and remote system upgrades to allow for remotely updating your Stratix II and Stratix II GX designs. Table 13–4 summarizes which configuration features can be used in each configuration scheme.

Table 13–4. Stratix II and Stratix II GX Configuration Features

Configuration Scheme	Configuration Method	Design Security	Decompression	Remote System Upgrade
FPP	MAX II device or a Microprocessor with flash memory	✓ (1)	✓ (1)	✓
	Enhanced Configuration Device		✓ (2)	✓
AS	Serial Configuration Device	✓	✓	✓ (3)
PS	MAX II device or a Microprocessor with flash memory	✓	✓	✓
	Enhanced Configuration Device	✓	✓	✓
	Download cable	✓	✓	
PPA	MAX II device or a Microprocessor with flash memory			✓
JTAG	MAX II device or a Microprocessor with flash memory			

Notes to Table 13–4:

(1) In these modes, the host system must send a DCLK that is 4× the data rate.

(2) The enhanced configuration device decompression feature is available, while the Stratix II and Stratix II GX decompression feature is not available.

(3) Only remote update mode is supported when using the AS configuration scheme. Local update mode is not supported.

Configuration Data Decompression

Stratix II and Stratix II GX devices support configuration data decompression, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other memory and transmit this compressed bit stream to Stratix II and Stratix II GX devices. During configuration, Stratix II and Stratix II GX devices automatically recognize the compressed file format and decompresses the bit stream in real time and programs its SRAM cells.

Data indicates that compression typically reduces configuration bit stream size by 35 to 55%.

Stratix II and Stratix II GX devices support decompression in the FPP (when using a MAX II device/microprocessor + flash), AS, and PS configuration schemes. Decompression is not supported in the PPA configuration scheme nor in JTAG-based configuration.

When using FPP mode, the intelligent host must provide a DCLK that is 4× the data rate. Therefore, the configuration data must be valid for four DCLK cycles.

The decompression feature supported by Stratix II and Stratix II GX devices is different from the decompression feature in enhanced configuration devices (EPC16, EPC8, and EPC4 devices), although they both use the same compression algorithm. The data decompression feature in the enhanced configuration devices allows them to store compressed data and decompress the bitstream before transmitting it to the target devices. When using Stratix II and Stratix II GX devices in FPP mode with enhanced configuration devices, the decompression feature is available only in the enhanced configuration device, not the Stratix II or Stratix II GX device.

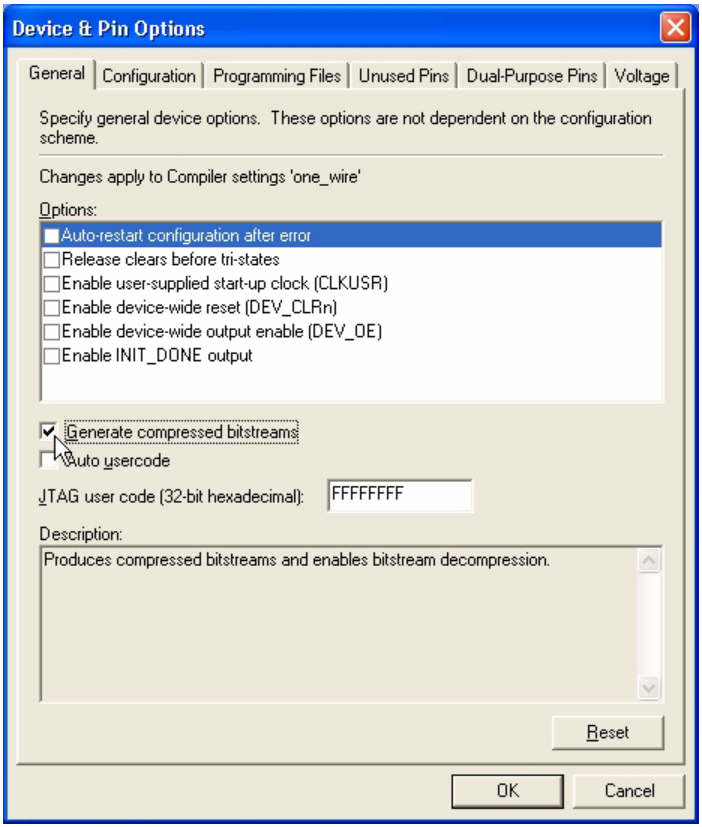
In PS mode, use the Stratix II or Stratix II GX decompression feature because sending compressed configuration data reduces configuration time. Do not use both the Stratix II or Stratix II GX device and the enhanced configuration device decompression features simultaneously. The compression algorithm is not intended to be recursive and could expand the configuration file instead of compressing it further.

When you enable compression, the Quartus II software generates configuration files with compressed configuration data. This compressed file reduces the storage requirements in the configuration device or flash memory, and decreases the time needed to transmit the bitstream to the Stratix II or Stratix II GX device. The time required by a Stratix II or Stratix II GX device to decompress a configuration file is less than the time needed to transmit the configuration data to the device.

There are two ways to enable compression for Stratix II and Stratix II GX bitstreams: before design compilation (in the **Compiler Settings** menu) and after design compilation (in the **Convert Programming Files** window).

To enable compression in the project's compiler settings, select **Device** under the **Assignments** menu to bring up the **Settings** window. After selecting your Stratix II or Stratix II GX device, open the **Device & Pin Options** window, and in the **General** settings tab enable the check box for **Generate compressed bitstreams** (as shown in Figure 13–1).

Figure 13–1. Enabling Compression for Stratix II and Stratix II GX Bitstreams in Compiler Settings



Compression can also be enabled when creating programming files from the **Convert Programming Files** window.

1. Click **Convert Programming Files** (File menu).
2. Select the programming file type (POF, SRAM HEXOUT, RBF, or TTF).
3. For POF output files, select a configuration device.
4. In the **Input files to convert** box, select **SOF Data**.
5. Select **Add File** and add a Stratix II or Stratix II GX device SOF(s).

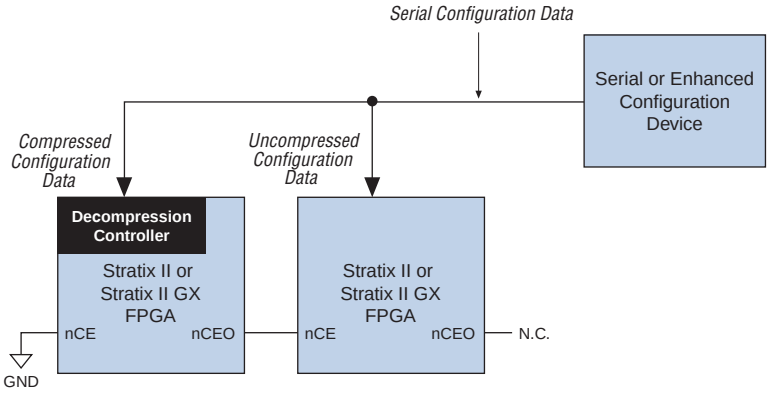
6. Select the name of the file you added to the **SOF Data** area and click **Properties**.

7. Check the **Compression** check box.

When multiple Stratix II or Stratix II GX devices are cascaded, you can selectively enable the compression feature for each device in the chain if you are using a serial configuration scheme. Figure 13–2 depicts a chain of two Stratix II or Stratix II GX devices. The first Stratix II or Stratix II GX device has compression enabled and therefore receives a compressed bit stream from the configuration device. The second Stratix II or Stratix II GX device has the compression feature disabled and receives uncompressed data.

In a multi-device FPP configuration chain all Stratix II or Stratix II GX devices in the chain must either enable of disable the decompression feature. You can not selectively enable the compression feature for each device in the chain because of the DATA and DCLK relationship.

Figure 13–2. Compressed and Uncompressed Configuration Data in the Same Configuration File



You can generate programming files for this setup from the **Convert Programming Files** window (File menu) in the Quartus II software.

Design Security Using Configuration Bitstream Encryption

Stratix II and Stratix II GX devices are the industry's first devices with the ability to decrypt a configuration bitstream using the Advanced Encryption Standard (AES) algorithm—the most advanced encryption algorithm available today. When using the design security feature, a

128-bit security key is stored in the Stratix II or Stratix II GX device. In order to successfully configure a Stratix II or Stratix II GX device that has the design security feature enabled, it must be configured with a configuration file that was encrypted using the same 128-bit security key. The security key can be stored in non-volatile memory inside the Stratix II or Stratix II GX device. This non-volatile memory does not require any external devices, such as a battery back-up, for storage.

When using a serial configuration scheme such as passive serial (PS) or active serial (AS), configuration time is the same whether or not the design security feature is enabled. If the fast passive parallel (FPP) scheme is used with the design security or decompression feature, a 4× DCLK is required. This results in a slower configuration time when compared to the configuration time of an FPGA that has neither the design security, nor decompression feature enabled. For more information about this feature, contact Altera Applications group.

Remote System Upgrade

Stratix II and Stratix II GX devices feature remote and local update.

For more information about this feature, refer to the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*

Power-On Reset Circuit

The POR circuit keeps the entire system in reset until the power supply voltage levels have stabilized on power-up. Upon power-up, the device does not release nSTATUS until V_{CCINT}, V_{CCPD}, and V_{CCIO} of banks 3, 4, 7, and 8 are above the device's POR trip point. On power down, V_{CCINT} is monitored for brown-out conditions.

The passive serial (PS) mode (MSEL[3,2,1,0] = 0010) and the Fast passive parallel (FPP) mode (MSEL[3,2,1,0] = 0000) always set bank 3 to use the lower POR trip point consistent with 1.8- and 1.5-V signaling, regardless of the VCCSEL setting. For all other configuration modes, VCCSEL selects the POR trip-point level. Refer to the section "VCCSEL Pin" on page 13–10 for more details.

In Stratix II devices, a pin-selectable option PORSEL is provided that allows you to select between a typical POR time setting of 12 ms or 100 ms. In both cases, you can extend the POR time by using an external component to assert the nSTATUS pin low.

V_{CCPD} Pins

Stratix II and Stratix II GX devices also offer a new power supply, V_{CCPD}, which must be connected to 3.3-V in order to power the 3.3-V/2.5-V buffer available on the configuration input pins and JTAG pins. V_{CCPD} applies to all the JTAG input pins (TCK, TMS, TDI, and TRST) and the configuration pins when VCCSEL is connected to ground. Refer to Table 13–5 for information on the pins affected by VCCSEL.

V_{CCPD} must ramp-up from 0-V to 3.3-V within 100 ms. If V_{CCPD} is not ramped up within this specified time, your Stratix II or Stratix II GX device will not configure successfully. If your system does not allow for a V_{CCPD} ramp-up time of 100 ms or less, you must hold nCONFIG low until all power supplies are stable.

VCCSEL Pin

The VCCSEL pin selects the type of input buffer used on configuration input pins and it selects the POR trip point voltage level for V_{CCIO} bank 3 powered by VCCIO3 pins.

For more information, refer to Table 13–24 on page 13–105.

The configuration input pins and the PLL_ENA pin (Table 13–5) have a dual buffer design. These pins have a 3.3-V/2.5-V input buffer and a 1.8-V/1.5-V input buffer. The VCCSEL input pin selects which input buffer is used during configuration. The 3.3-V/2.5-V input buffer is powered by V_{CCPD}, while the 1.8-V/1.5-V input buffer is powered by V_{CCIO}. After configuration, the dual-purpose configuration pins are powered by the V_{CCIO} pins of the bank in which they reside. Table 13–5 shows the pins affected by VCCSEL.

Table 13–5. Pins Affected by the Voltage Level at VCCSEL

Pin	VCCSEL = LOW (connected to GND)	VCCSEL = HIGH (connected to V_{CCPD})
nSTATUS (when used as an input)	3.3/2.5-V input buffer is selected. Input buffer is powered by V_{CCPD}.	1.8/1.5-V input buffer is selected. Input buffer is powered by V_{CCIO} of the I/O bank. These input buffers are 3.3 V tolerant.
nCONFIG		
CONF_DONE (when used as an input)		
DATA[7..0]		
nCE		
DCLK (when used as an input)		
CS		
nWS		
nRS		
nCS		
CLKUSR		
DEV_OE		
DEV_CLRn		
RUnLU		
PLL_ENA		

VCCSEL is sampled during power-up. Therefore, the VCCSEL setting cannot change on the fly or during a reconfiguration. The VCCSEL input buffer is powered by V_{CCINT} and has an internal 5-kΩ pull-down resistor that is always active.

VCCSEL must be hardwired to V_{CCPD} or GND.

A logic high selects the 1.8-V/1.5-V input buffer, and a logic low selects the 3.3-V/2.5-V input buffer. VCCSEL should be set to comply with the logic levels driven out of the configuration device or MAX II device or a microprocessor with flash memory.

VCCSEL also sets the POR trip point for I/O bank 3 to ensure that this I/O bank has powered up to the appropriate voltage levels before configuration begins. For passive serial (PS) mode (MSEL[3..0] = 0010) and for Fast passive parallel (FPP) mode (MSEL[3..0] = 0000) the POR circuitry selects the trip point associated with 1.5-V/1.8-V signaling. For all other configuration modes defined by MSEL[3..0] settings (other

than 00X0 (MSEL[1] = X, "don't care"), VCCSEL=GND selects the higher I/O bank 3 POR trip point for 2.5-V/3.3-V signaling and VCCSEL=VCCPD selects the lower I/O bank 3 POR trip point associated with 1.5-V/1.8-V signaling.

For all configuration modes with MSEL[3..0] not equal to 00X0 (MSEL[1] = X, "don't care"), if VCCIO of configuration bank 3 is powered by 1.8-V or 1.5-V and VCCSEL = GND, the voltage supplied to this I/O bank(s) may never reach the POR trip point, which prevents the device from beginning configuration.

If the VCCIO of I/O bank 3 is powered by 1.5- or 1.8-V and the configuration signals used require 3.3- or 2.5-V signaling, you should set VCCSEL to VCCPD to enable the 1.8-/1.5-V input buffers for configuration. The 1.8-V/1.5-V input buffers are 3.3-V tolerant.

The fast passive parallel (FPP) and passive serial (PS) modes always enable bank 3 to use the POR trip point to be consistent with 1.8- and 1.5-V signaling, regardless of the VCCSEL setting.

Table 13–6 shows how you should set VCCSEL depending on the configuration mode, the voltage level on VCCIO3 pins that power bank 3, and the supported configuration input voltages.

Table 13–6. Supported V_{CCSEL} Setting Based on Mode, VCCIO3, and Input Configuration Voltage

Configuration Mode	V_{CCIO} (Bank 3)	Supported Configuration Input Voltages	V_{CCSEL}
All modes	3.3-V/2.5-V	3.3-V/2.5-V	GND
All modes	1.8-V/1.5-V	3.3-V/2.5-V	V_{CCPD} *(1)*
All modes	1.8-V/1.5-V	1.8-V/1.5-V	V_{CCPD}
—	3.3-V/2.5-V	1.8-V/1.5-V	Not Supported

Note to Table 13–6:

(1) The VCCSEL pin can also be connected to GND for PS (MSEL[3..0]=0010) and FPP (MSEL[3..0]=0000) modes.

Table 13–7 shows the configuration mode support for banks 4, 7, and 8.

Table 13–7. Stratix II Configuration Mode Support for Banks 4, 7 and 8

Configuration Mode	Configuration Voltage/V_{CCIO} Support for Banks 4, 7, and 8		
	3.3/3.3	1.8/1.8	3.3/1.8
	VCCSEL = GND	VCCSEL = VCCPD	VCCSEL = GND
Fast passive parallel	Y	Y	Y
Passive parallel asynchronous	Y	Y	Y
Passive serial	Y	Y	Y
Remote system upgrade FPP	Y	Y	Y
Remote system upgrade PPA	Y	Y	Y
Remote system upgrade PS	Y	Y	Y
Fast AS (40 MHz)	Y	Y	Y
Remote system upgrade fast AS (40 MHz)	Y	Y	Y
FPP with decompression and/or design security	Y	Y	Y
Remote system upgrade FPP with decompression and/or design security feature enabled	Y	Y	Y
AS (20 MHz)	Y	Y	Y
Remote system upgrade AS (20 MHz)	Y	Y	Y

Output Configuration Pins

You must verify that the configuration output pins for your chosen configuration modes meet the V_{IH} of the configuration device. Refer to Table 13–22 on page 13–94 for a consolidated list of configuration output pins.

The V_{IH} of 3.3 V or 2.5 V configuration devices will not be met when the V_{CCIO} of the output configuration pins are 1.8 V or 1.5 V. Level shifters will be required to meet the input high level voltage threshold V_{IH}.

Note that AS mode is only applicable for 3.3-V configurations. If I/O bank 3 is less than 3.3 V, level shifters are required on the output pins (DCLK, nCSO, ASDO) from the Stratix II or Stratix II GX device back to the EPCS device.

The key is to ensure the VCCIO voltage of bank 3 is high enough to trip the VCCIO3 POR trip point on power-up. Also, to make sure the configuration device meets the V_{IH} for the configuration input pins based on the selected input buffer.

Fast Passive Parallel Configuration

Fast passive parallel (FPP) configuration in Stratix II and Stratix II GX devices is designed to meet the continuously increasing demand for faster configuration times. Stratix II and Stratix II GX devices are designed with the capability of receiving byte-wide configuration data per clock cycle. Table 13–8 shows the MSEL pin settings when using the FFP configuration scheme.

Table 13–8. Stratix II and Stratix II GX MSEL Pin Settings for FPP Configuration Schemes Notes (1), (2), and (3)

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0
FPP when not using remote system upgrade or decompression and/or design security feature	0	0	0	0
FPP when using remote system upgrade (4)	0	1	0	0
FPP with decompression and/or design security feature enabled (5)	1	0	1	1
FPP when using remote system upgrade and decompression and/or design security feature (4), (5)	1	1	0	0

Notes to Table 13–8:

(1) You must verify the configuration output pins for your chosen configuraiton modes meet the V_{IH} of the configuration device. Refer to Table 13–22 for a consolidated list of configuration output pins.

(2) The V_{IH} of 3.3-V or 2.5-V configuration devices will not be met when the VCCIO of the output configuration pins is 1.8-V or 1.5-V. Level shifters will be required to meet the input high level voltage threshold V_{IH}.

(3) The VCCSEL signal does not control TDO or nCEO. During configuration, these pins drive out voltage levels corresponding to the VCCIO supply voltage that powers the I/O bank containing the pin. For more information about multi-volt support, including information about using TDO and nCEO in multi-volt systems, refer to the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.

(4) These schemes require that you drive the RUnLU pin to specify either remote update or local update. For more information about remote system upgrade in Stratix II devices, refer to the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

(5) These modes are only supported when using a MAX II device or a microprocessor with flash memory for configuration. In these modes, the host system must output a DCLK that is 4× the data rate.

FPP configuration of Stratix II and Stratix II GX devices can be performed using an intelligent host, such as a MAX II device, a microprocessor, or an Altera enhanced configuration device.

FPP Configuration Using a MAX II Device as an External Host

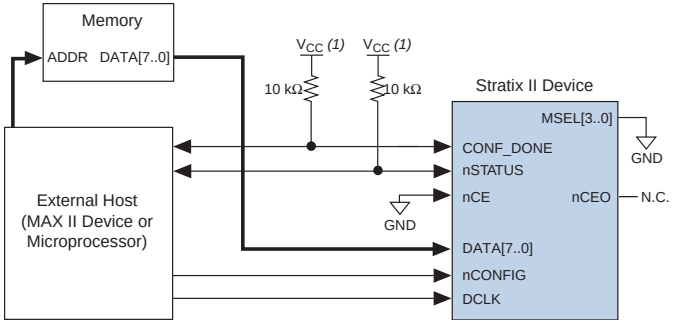
FPP configuration using compression and an external host provides the fastest method to configure Stratix II and Stratix II GX devices. In the FPP configuration scheme, a MAX II device can be used as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix II or Stratix II GX device. Configuration data can be stored in RBF, HEX, or TTF format. When using the MAX II devices as an intelligent host, a design that controls the configuration process, such as fetching the data from flash memory and sending it to the device, must be stored in the MAX II device.

If you are using the Stratix II or Stratix II GX decompression and/or design security feature, the external host must be able to send a DCLK frequency that is 4× the data rate.

The 4× DCLK signal does not require an additional pin and is sent on the DCLK pin. The maximum DCLK frequency is 100 MHz, which results in a maximum data rate of 200 Mbps. If you are not using the Stratix II or Stratix II GX decompression or design security features, the data rate is 8× the DCLK frequency.

Figure 13–3 shows the configuration interface connections between the Stratix II or Stratix II GX device and a MAX II device for single device configuration.

Figure 13–3. Single Device FPP Configuration Using an External Host



Note to Figure 13–3:

(1) The pull-up resistor should be connected to a supply that provides an acceptable input signal for the device. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

Upon power-up, the Stratix II and Stratix II GX devices go through a Power-On Reset (POR). The POR delay is dependent on the PORSEL pin setting; when PORSEL is driven low, the POR time is approximately 100 ms, if PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. Once the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

You can hold nConfig low in order to stop device configuration.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* or the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

The configuration cycle consists of three stages: reset, configuration and initialization. While nCONFIG or nSTATUS are low, the device is in the reset stage. To initiate configuration, the MAX II device must drive the nCONFIG pin from low-to-high.

V_{CCINT}, V_{CCIO}, and V_{CCPD} of the banks where the configuration and JTAG pins reside need to be fully powered to the appropriate voltage levels in order to begin the configuration process.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. Once nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the MAX II device places the configuration data one byte at a time on the DATA[7..0] pins.

Stratix II and Stratix II GX devices receive configuration data on the DATA[7..0] pins and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. If you are using the Stratix II or Stratix II GX decompression and/or design security feature, configuration data is latched on the rising edge of every fourth DCLK cycle. After the configuration data is latched in, it is processed during the following three DCLK cycles.

Data is continuously clocked into the target device until CONF_DONE goes high. The CONF_DONE pin goes high one byte early in parallel configuration (FPP and PPA) modes. The last byte is required for serial configuration (AS and PS) modes. After the device has received the next to last byte of the configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

In Stratix II and Stratix II GX devices, the initialization clock source is either the internal oscillator (typically 10 MHz) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If the internal oscillator is used, the Stratix II or Stratix II GX device provides itself with enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You can also synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. The **Enable user-supplied start-up clock** (**CLKUSR**) option can be turned on in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. Supplying a clock on CLKUSR does not affect the configuration process. The CONF_DONE pin goes high one byte early in parallel configuration (FPP and PPA) modes. The last byte is required for serial configuration (AS and PS) modes. After the CONF_DONE pin transitions high, CLKUSR is enabled after the time specified as t_{CD2CU}. After this time period elapses, Stratix II and Stratix II GX devices require 299 clock cycles to initialize properly and enter user mode. Stratix II and Stratix II GX devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. This **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If the INIT_DONE pin is used, it is high because of an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. Once the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high. The MAX II device must be able to detect this low-to-high transition, which signals the device has

entered user mode. When initialization is complete, the device enters user mode. In user-mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

To ensure DCLK and DATA[7..0] are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[7..0] pins are available as user I/O pins after configuration. When you select the FPP scheme in the Quartus II software, as a default, these I/O pins are tri-stated in user mode. To change this default option in the Quartus II software, select the **Pins** tab of the **Device & Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If you are using the Stratix II or Stratix II GX decompression and/or design security feature and need to stop DCLK, it can only be stopped three clock cycles after the last data byte was latched into the Stratix II or Stratix II GX device.

By stopping DCLK, the configuration circuit allows enough clock cycles to process the last byte of latched configuration data. When the clock restarts, the MAX II device must provide data on the DATA[7..0] pins prior to sending the first DCLK rising edge.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device & Pin Options** (dialog box) is turned on, the device releases nSTATUS after a reset time-out period (maximum of 100 µs). After nSTATUS is released and pulled high by a pull-up resistor, the MAX II device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

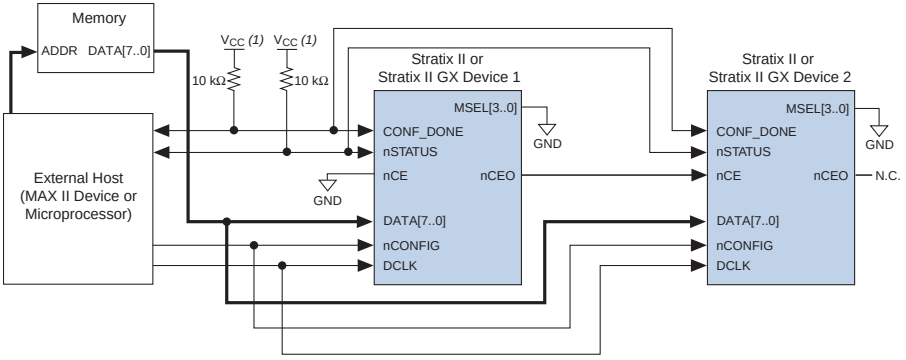
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The CONF_DONE pin must be monitored by the MAX II device to detect errors and determine when programming completes. If all configuration data is sent, but the CONF_DONE or INIT_DONE signals have not gone high, the MAX II device will reconfigure the target device.

If the optional CLKUSR pin is used and nCONFIG is pulled low to restart configuration during device initialization, you need to ensure CLKUSR continues toggling during the time nSTATUS is low (maximum of 100 µs).

When the device is in user-mode, initiating a reconfiguration is done by transitioning the nCONFIG pin low-to-high. The nCONFIG pin should be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Once nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 13–4 shows how to configure multiple devices using a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the Stratix II or Stratix II GX devices are cascaded for multi-device configuration.

Figure 13–4. Multi-Device FPP Configuration Using an External Host



Note to Figure 13–4:

(1) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O standard on the device and the external host.

In multi-device FPP configuration, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration within one clock cycle; therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in

the chain. The configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

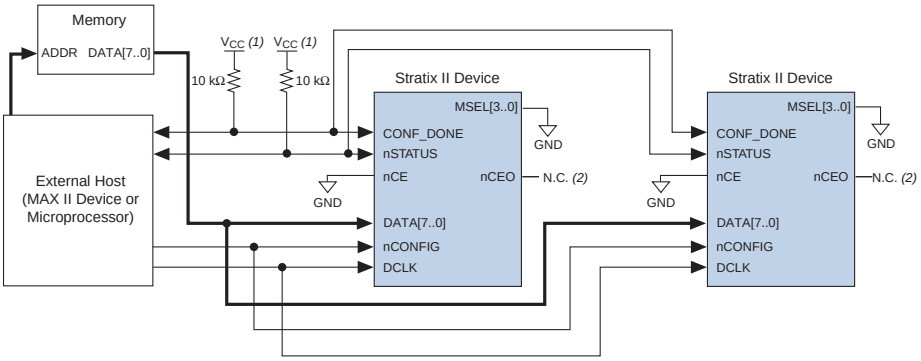
All nSTATUS and CONF_DONE pins are tied together and if any device detects an error, configuration stops for the entire chain and the entire chain must be reconfigured. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (maximum of 100 µs). After all nSTATUS pins are released and pulled high, the MAX II device can try to reconfigure the chain without pulsing nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

In a multi-device FPP configuration chain, all Stratix II or Stratix II GX devices in the chain must either enable or disable the decompression and/or design security feature. You can not selectively enable the decompression and/or design security feature for each device in the chain because of the DATA and DCLK relationship. If the chain contains devices that do not support design security, you should use a serial configuration scheme.

If a system has multiple devices that contain the same configuration data, tie all device nCE inputs to GND, and leave nCEO pins floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices start and complete configuration at the same time. Figure 13–5 shows multi-device FPP configuration when both Stratix II or Stratix II GX devices are receiving the same configuration data.

Figure 13–5. Multiple-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 13–5:

(1) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) The nCEO pins of both Stratix II or Stratix II GX devices are left unconnected when configuring the same configuration data into multiple devices.

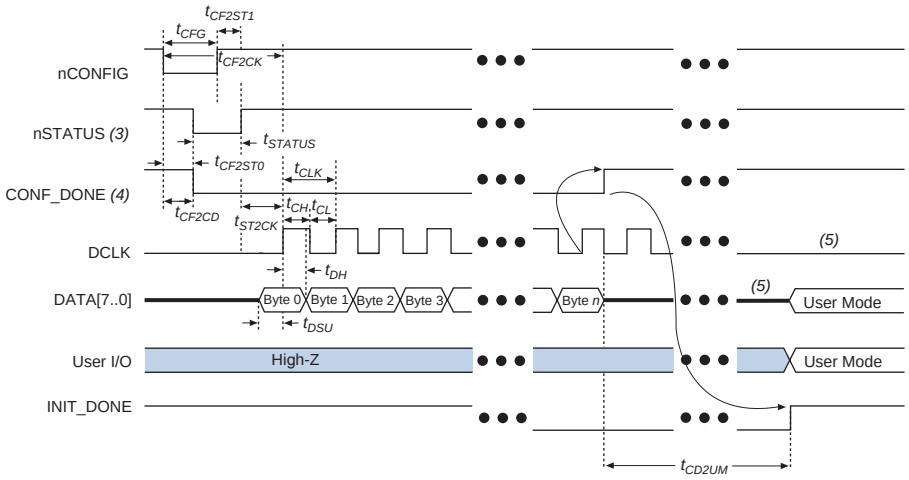
You can use a single configuration chain to configure Stratix II or Stratix II GX devices with other Altera devices that support FPP configuration, such as Stratix devices. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device initiates reconfiguration in all devices, tie all of the device CONF_DONE and nSTATUS pins together.

For more information on configuring multiple Altera devices in the same configuration chain, refer to *Configuring Mixed Altera FPGA Chains* in volume 2 of the *Configuration Handbook*.

FPP Configuration Timing

Figure 13–6 shows the timing waveform for FPP configuration when using a MAX II device as an external host. This waveform shows the timing when the decompression and the design security feature are not enabled.

Figure 13–6. FPP Configuration Timing Waveform *Notes (1), (2)*



Notes to Figure 13–6:

(1) This timing waveform should be used when the decompression and design security feature are not used.
(2) The beginning of this waveform shows the device in user-mode. In user-mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.
(3) Upon power-up, the Stratix II or Stratix II GX device holds nSTATUS low for the time of the POR delay.
(4) Upon power-up, before and during configuration, CONF_DONE is low.
(5) DCLK should not be left floating after configuration. It should be driven high or low, whichever is more convenient.
(6) DATA[7..0] are available as user I/O pins after configuration and the state of these pins depends on the dual-purpose pin settings.

Table 13–9 defines the timing parameters for Stratix II and Stratix II GX devices for FPP configuration when the decompression and the design security features are not enabled.

Table 13–9. FPP Timing Parameters for Stratix II and Stratix II GX Devices (Part 1 of 2) *Notes (1), (2)*

Symbol	Parameter	Min	Max	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low		800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low		800	ns
t_{CFG}	nCONFIG low pulse width	2		µs
t_{STATUS}	nSTATUS low pulse width	10	100 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high		100 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	100		µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2		µs

Table 13–9. FPP Timing Parameters for Stratix II and Stratix II GX Devices (Part 2 of 2) *Notes (1), (2)*

Symbol	Parameter	Min	Max	Units
t_{DSU}	Data setup time before rising edge on DCLK	5		ns
t_{DH}	Data hold time after rising edge on DCLK	0		ns
t_{CH}	DCLK high time	4		ns
t_{CL}	DCLK low time	4		ns
t_{CLK}	DCLK period	10		ns
f_{MAX}	DCLK frequency		100	MHz
t_R	Input rise time		40	ns
t_F	Input fall time		40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	20	100	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period		
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} +(299 × CLKUSR period)		

Notes to Table 13–9:

(1) This information is preliminary.

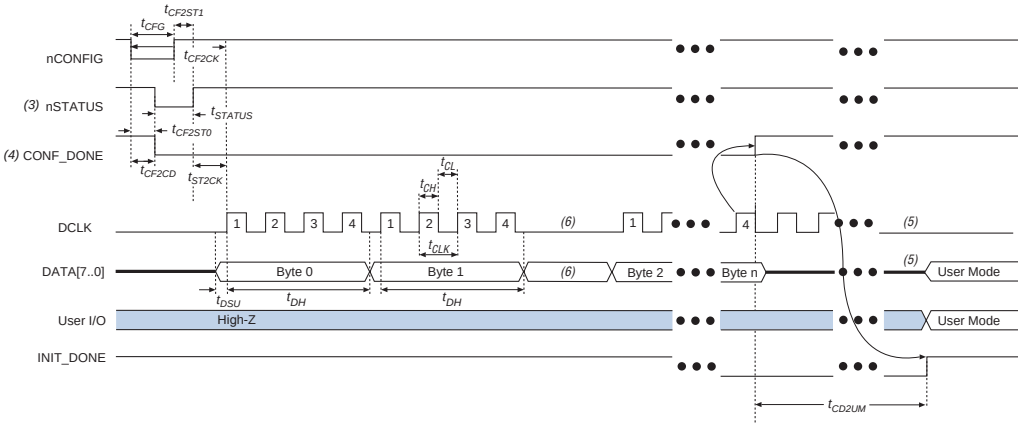
(2) These timing parameters should be used when the decompression and design security feature are not used.

(3) This value is obtainable if users do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(4) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device.

Figure 13–7 shows the timing waveform for FPP configuration when using a MAX II device as an external host. This waveform shows the timing when the decompression and/or the design security feature are enabled.

Figure 13–7. FPP Configuration Timing Waveform With Decompression or Design Security Feature Enabled *Notes (1), (2)*



Notes to Figure 13–7:

(1) This timing waveform should be used when the decompression and/or design security feature are used.
(2) The beginning of this waveform shows the device in user-mode. In user-mode, nCONFIG, nSTATUS and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.
(3) Upon power-up, the Stratix II or Stratix II GX device holds nSTATUS low for the time of the POR delay.
(4) Upon power-up, before and during configuration, CONF_DONE is low.
(5) DCLK should not be left floating after configuration. It should be driven high or low, whichever is more convenient.
(6) DATA[7..0] are available as user I/O pins after configuration and the state of these pins depends on the dual-purpose pin settings.
(7) If needed, DCLK can be paused by holding it low. When DCLK restarts, the external host must provide data on the DATA[7..0] pins prior to sending the first DCLK rising edge.

Table 13–10 defines the timing parameters for Stratix II and Stratix II GX devices for FPP configuration when the decompression and/or the design security feature are enabled.

Table 13–10. FPP Timing Parameters for Stratix II and Stratix II GX Devices With Decompression or Design Security Feature Enabled *Note (1)*

Symbol	Parameter	Min	Max	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low		800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low		800	ns
t_{CFG}	nCONFIG low pulse width	2		µs
t_{STATUS}	nSTATUS low pulse width	10	100 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high		100 *(2)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	100		µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2		µs
t_{DSU}	Data setup time before rising edge on DCLK	5		ns
t_{DH}	Data hold time after rising edge on DCLK	30		ns
t_{CH}	DCLK high time	4		ns
t_{CL}	DCLK low time	4		ns
t_{CLK}	DCLK period	10		ns
f_{MAX}	DCLK frequency		100	MHz
t_{DATA}	Data rate		200	Mbps
t_R	Input rise time		40	ns
t_F	Input fall time		40	ns
t_{CD2UM}	CONF_DONE high to user mode *(3)*	20	100	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period		
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (299 × CLKUSR period)		

Notes to Table 13–10:

(1) These timing parameters should be used when the decompression and design security feature are used.

(2) This value is obtainable if users do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(3) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device.

Device configuration options and how to create configuration files are discussed further in the *Software Settings* chapter in the *Configuration Handbook*.

FPP Configuration Using a Microprocessor

In the FPP configuration scheme, a microprocessor can control the transfer of configuration data from a storage device, such as flash memory, to the target Stratix II or Stratix II GX device.

All information in "FPP Configuration Using a MAX II Device as an External Host" on page 13–15 is also applicable when using a microprocessor as an external host. Refer to that section for all configuration and timing information.

FPP Configuration Using an Enhanced Configuration Device

In the FPP configuration scheme, an enhanced configuration device sends a byte of configuration data every DCLK cycle to the Stratix II or Stratix II GX device. Configuration data is stored in the configuration device.

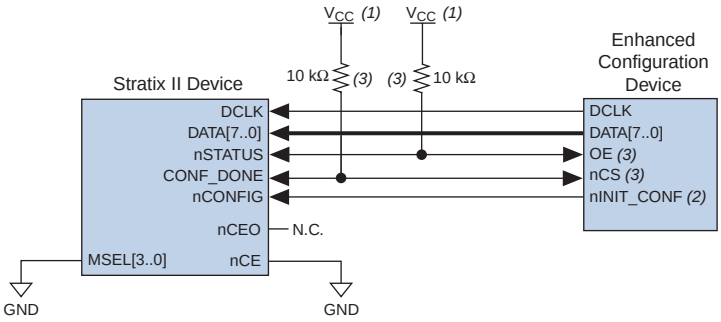
When configuring your Stratix II or Stratix II GX device using FPP mode and an enhanced configuration device, the enhanced configuration device decompression feature is available while the Stratix II and Stratix II GX decompression and design security features are not.

Figure 13–8 shows the configuration interface connections between a Stratix II or Stratix II GX device and the enhanced configuration device for single device configuration.

The figures in this chapter only show the configuration-related pins and the configuration pin connections between the configuration device and the device.

For more information on the enhanced configuration device and flash interface pins, such as PGM[2..0], EXCLK, PORSEL, A[20..0], and DQ[15..0], refer to the *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* in volume 2 of the *Configuration Handbook*.

Figure 13–8. Single Device FPP Configuration Using an Enhanced Configuration Device



Notes to Figure 13–8:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active. This means an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

The value of the internal pull-up resistors on the enhanced configuration devices can be found in the *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* in volume 2 of the *Configuration Handbook*.

When using enhanced configuration devices, you can connect the device's nCONFIG pin to nINIT_CONF pin of the enhanced configuration device, which allows the INIT_CONF JTAG instruction to initiate device configuration. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor. An internal pull-up resistor on the nINIT_CONF pin is always active in the enhanced configuration devices, which means an external pull-up resistor should not be used if nCONFIG is tied to nINIT_CONF.

Upon power-up, the Stratix II or Stratix II GX device goes through a POR. The POR delay is dependent on the PORSEL pin setting; when PORSEL is driven low, the POR time is approximately 100 ms, if PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS low, and tri-state all user I/O pins. The

configuration device also goes through a POR delay to allow the power supply to stabilize. The POR time for enhanced configuration devices can be set to either 100 ms or 2 ms, depending on its PORSEL pin setting. If the PORSEL pin is connected to GND, the POR delay is 100 ms. If the PORSEL pin is connected to V_{CC}, the POR delay is 2 ms. During this time, the configuration device drives its OE pin low. This low signal delays configuration because the OE pin is connected to the target device's nSTATUS pin.

When selecting a POR time, you need to ensure that the device completes power-up before the enhanced configuration device exits POR. Altera recommends that you use a 12-ms POR time for the Stratix II or Stratix II GX device, and use a 100-ms POR time for the enhanced configuration device.

When both devices complete POR, they release their open-drain OE or nSTATUS pin, which is then pulled high by a pull-up resistor. Once the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors, which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *Stratix II Device Handbook* or the *Stratix II GX Device Handbook*.

When the power supplies have reached the appropriate operating voltages, the target device senses the low-to-high transition on nCONFIG and initiates the configuration cycle. The configuration cycle consists of three stages: reset, configuration and initialization. While nCONFIG or nSTATUS are low, the device is in reset. The beginning of configuration can be delayed by holding the nCONFIG or nSTATUS pin low.

V_{CCINT}, V_{CCIO} and V_{CCPD} of the banks where the configuration and JTAG pins reside need to be fully powered to the appropriate voltage levels in order to begin the configuration process.

When nCONFIG goes high, the device comes out of reset and releases the nSTATUS pin, which is pulled high by a pull-up resistor. Enhanced configuration devices have an optional internal pull-up resistor on the OE pin. This option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If this internal pull-up resistor is not used, an external 10-kΩ pull-up resistor on the OE-nSTATUS line is required. Once nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins.

When nSTATUS is pulled high, the configuration device's OE pin also goes high and the configuration device clocks data out to the device using the Stratix II or Stratix II GX device's internal oscillator. The Stratix II and Stratix II GX devices receive configuration data on the DATA[7..0] pins and the clock is received on the DCLK pin. A byte of data is latched into the device on each rising edge of DCLK.

After the device has received all configuration data successfully, it releases the open-drain CONF_DONE pin which is pulled high by a pull-up resistor. Because CONF_DONE is tied to the configuration device's nCS pin, the configuration device is disabled when CONF_DONE goes high. Enhanced configuration devices have an optional internal pull-up resistor on the nCS pin. This option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If this internal pull-up resistor is not used, an external 10-kΩ pull-up resistor on the nCS-CONF_DONE line is required. A low to high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin.

In Stratix II and Stratix II GX devices, the initialization clock source is either the internal oscillator (typically 10 MHz) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If the internal oscillator is used, the Stratix II or Stratix II GX device provides itself with enough clock cycles for proper initialization. You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. The **Enable user-supplied start-up clock** (**CLKUSR**) option can be turned on in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. Supplying a clock on CLKUSR will not affect the configuration process. After all configuration data has been accepted and CONF_DONE goes high, CLKUSR will be enabled after the time specified as t_{CD2CU}. After this time period elapses, Stratix II and Stratix II GX devices require 299 clock cycles to initialize properly and enter user mode. Stratix II and Stratix II GX devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If the INIT_DONE pin is used, it will be high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. Once the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin will go low. When initialization is complete, the INIT_DONE pin will be released and pulled high. In user-mode, the user

I/O pins will no longer have weak pull-up resistors and will function as assigned in your design. The enhanced configuration device will drive DCLK low and DATA[7..0] high at the end of configuration.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. Since the nSTATUS pin is tied to OE, the configuration device will also be reset. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box) is turned on, the device will automatically initiate reconfiguration if an error occurs. The Stratix II or Stratix II GX device releases its nSTATUS pin after a reset time-out period (maximum of 100 µs). When the nSTATUS pin is released and pulled high by a pull-up resistor, the configuration device reconfigures the chain. If this option is turned off, the external system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration. The external system can pulse nCONFIG if nCONFIG is under system control rather than tied to V_{CC}.

In addition, if the configuration device sends all of its data and then detects that CONF_DONE has not gone high, it recognizes that the device has not configured successfully. Enhanced configuration devices wait for 64 DCLK cycles after the last configuration bit was sent for CONF_DONE to reach a high state. In this case, the configuration device pulls its OE pin low, which in turn drives the target device's nSTATUS pin low. If the Auto-restart configuration after error option is set in the software, the target device resets and then releases its nSTATUS pin after a reset time-out period (maximum of 100 µs). When nSTATUS returns to a logic high level, the configuration device will try to reconfigure the device.

When CONF_DONE is sensed low after configuration, the configuration device recognizes that the target device has not configured successfully. Therefore, your system should not pull CONF_DONE low to delay initialization. Instead, you should use the CLKUSR option to synchronize the initialization of multiple devices that are not in the same configuration chain. Devices in the same configuration chain will initialize together if their CONF_DONE pins are tied together.

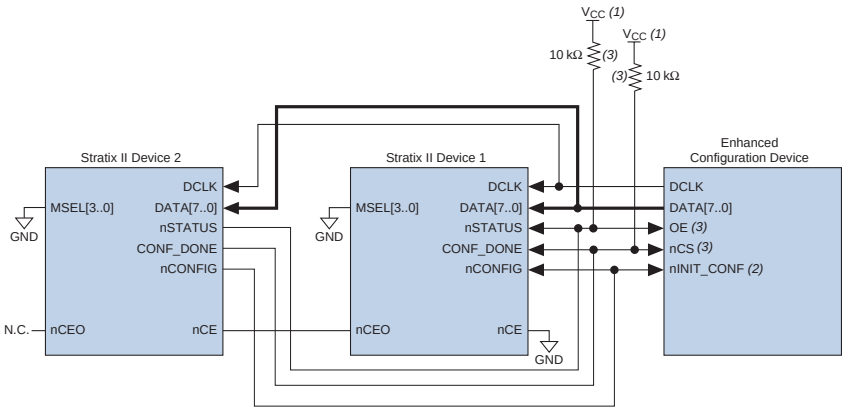
If the optional CLKUSR pin is used and nCONFIG is pulled low to restart configuration during device initialization, ensure CLKUSR continues toggling during the time nSTATUS is low (maximum of 100 µs).

When the device is in user-mode, a reconfiguration can be initiated by pulling the nCONFIG pin low. The nCONFIG pin should be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Because CONF_DONE is

pulled low, this activates the configuration device because it sees its nCS pin drive low. Once nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 13–9 shows how to configure multiple Stratix II or Stratix II GX devices with an enhanced configuration device. This circuit is similar to the configuration device circuit for a single device, except the Stratix II or Stratix II GX devices are cascaded for multi-device configuration.

Figure 13–9. Multi-Device FPP Configuration Using an Enhanced Configuration Device



Notes to Figure 13–9:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active. This means an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-up resistors on configuration device** option when generating programming files.

Enhanced configuration devices cannot be cascaded.

When performing multi-device configuration, you must generate the configuration device's POF from each project's SOF. You can combine multiple SOFs using the **Convert Programming Files** window in the Quartus II software.

For more information on how to create configuration files for multi-device configuration chains, refer to *Software Settings* in volume 2 of the *Configuration Handbook*.

In multi-device FPP configuration, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Pay special attention to the configuration signals because they may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device.

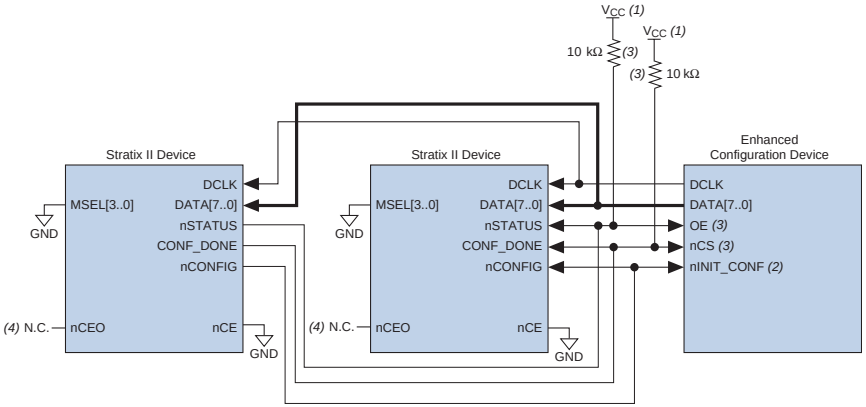
When configuring multiple devices, configuration does not begin until all devices release their OE or nSTATUS pins. Similarly, since all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

Since all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, configuration stops for the entire chain and the entire chain must be reconfigured. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This low signal drives the OE pin low on the enhanced configuration device and drives nSTATUS low on all devices, which causes them to enter a reset state. This behavior is similar to a single device detecting an error.

If the **Auto-restart configuration after error** option is turned on, the devices will automatically initiate reconfiguration if an error occurs. The devices will release their nSTATUS pins after a reset time-out period (maximum of 100 µs). When all the nSTATUS pins are released and pulled high, the configuration device tries to reconfigure the chain. If the **Auto-restart configuration after error** option is turned off, the external system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration. The external system can pulse nCONFIG if nCONFIG is under system control rather than tied to V_{CC}.

Your system may have multiple devices that contain the same configuration data. To support this configuration scheme, all device nCE inputs are tied to GND, while nCEO pins are left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices will start and complete configuration at the same time. Figure 13–10 shows multi-device FPP configuration when both Stratix II or Stratix II GX devices are receiving the same configuration data.

Figure 13–10. Multiple-Device FPP Configuration Using an Enhanced Configuration Device When Both devices Receive the Same Data



Notes to Figure 13–10:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active. This means an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

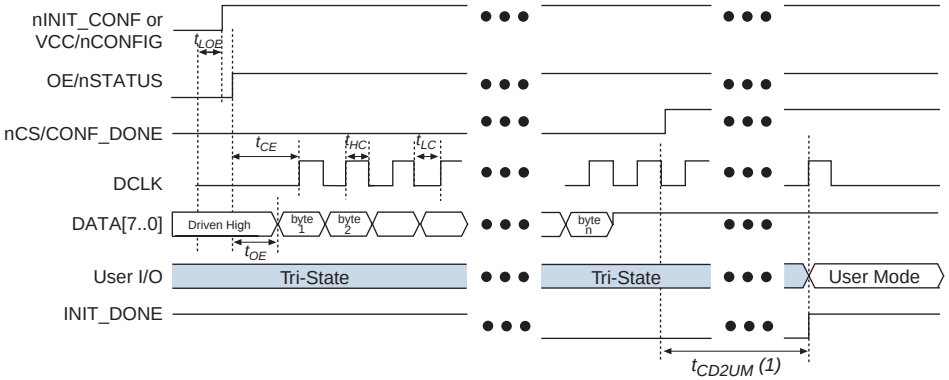
(4) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

You can use a single enhanced configuration chain to configure multiple Stratix II or Stratix II GX devices with other Altera devices that support FPP configuration, such as Stratix and Stratix GX devices. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device initiates reconfiguration in all devices, all of the device CONF_DONE and nSTATUS pins must be tied together.

For more information on configuring multiple Altera devices in the same configuration chain, refer to *Configuring Mixed Altera FPGA Chains* in the *Configuration Handbook*.

Figure 13–11 shows the timing waveform for the FPP configuration scheme using an enhanced configuration device.

Figure 13–11. Stratix II and Stratix II GX FPP Configuration Using an Enhanced Configuration Device Timing Waveform



Note to Figure 13–11:
(1) The initialization clock can come from the Stratix II or Stratix II GX device's internal oscillator or the CLKUSR pin.

For timing information, refer to the *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* in volume 2 of the *Configuration Handbook*.

Device configuration options and how to create configuration files are discussed further in the *Software Settings* section in volume 2 of the *Configuration Handbook*.

Active Serial Configuration (Serial Configuration Devices)

In the AS configuration scheme, Stratix II and Stratix II GX devices are configured using a serial configuration device. These configuration devices are low-cost devices with non-volatile memory that feature a simple four-pin interface and a small form factor. These features make serial configuration devices an ideal low-cost configuration solution.

Note that AS mode is only applicable for 3.3-V configurations. If I/O bank 3 is less than 3.3 V, level shifters are required on the output pins (DCLK, nCSO, ASDO) from the Stratix II or Stratix II GX device back to the EPCS device.

If VCCIO in bank 3 is set to 1.8 V, an external voltage level translator is needed to meet the V_{IH} of the EPCS device (3.3 V).

For more information on serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook*.

Serial configuration devices provide a serial interface to access configuration data. During device configuration, Stratix II and Stratix II GX devices read configuration data via the serial interface, decompress data if necessary, and configure their SRAM cells. This scheme is referred to as the AS configuration scheme because the device controls the configuration interface. This scheme contrasts with the PS configuration scheme, where the configuration device controls the interface.

The Stratix II and Stratix II GX decompression and design security features are fully available when configuring your Stratix II or Stratix II GX device using AS mode.

Table 13–11 shows the MSEL pin settings when using the AS configuration scheme.

Table 13–11. Stratix II and Stratix II GX MSEL Pin Settings for AS Configuration Schemes *Note (2)*

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0
Fast AS (40 MHz) *(1)*	1	0	0	0
Remote system upgrade fast AS (40 MHz) *(1)*	1	0	0	1
AS (20 MHz) *(1)*	1	1	0	1
Remote system upgrade AS (20 MHz) *(1)*	1	1	1	0

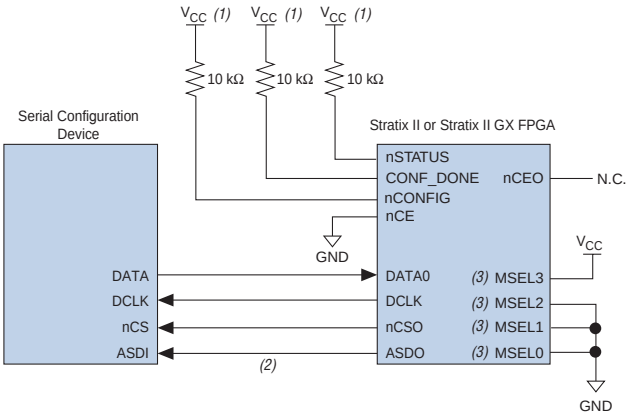
Notes to Table 13–11:

(1) Only the EPCS16 and EPCS64 devices support a DCLK up to 40 MHz clock; other EPCS devices support a DCLK up to 20 MHz. Refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook* for more information.

(2) Note that AS mode is only applicable for 3.3-V configuration. If I/O bank 3 is less than 3.3-V, level shifters are required on the output pins (DCLK,nCSO, and ASDO) from the Stratix II or Stratix II GX device back to the EPCS device.

Serial configuration devices have a four-pin interface: serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and an active-low chip select (nCS). This four-pin interface connects to Stratix II and Stratix II GX device pins, as shown in Figure 13–12.

Figure 13–12. Single Device AS Configuration



Notes to Figure 13–12:

(1) Connect the pull-up resistors to a 3.3-V supply.

(2) Stratix II and Stratix II GX devices use the ASDO to ASDI path to control the configuration device.

(3) If using an EPCS4 device, MSEL[3..0] should be set to 1101. Refer to Table 13–11 for more details.

Upon power-up, the Stratix II and Stratix II GX devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. If PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS and CONF_DONE low, and tri-state all user I/O pins. Once the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* and the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

The configuration cycle consists of three stages: reset, configuration and initialization. While nCONFIG or nSTATUS are low, the device is in reset. After POR, the Stratix II and Stratix II GX devices release nSTATUS, which is pulled high by an external 10-kΩ pull-up resistor, and enters configuration mode.

To begin configuration, power the V_{CCINT}, V_{CCIO}, and V_{CCPD} voltages (for the banks where the configuration and JTAG pins reside) to the appropriate voltage levels.

The serial clock (DCLK) generated by the Stratix II and Stratix II GX devices controls the entire configuration cycle and provides the timing for the serial interface. Stratix II and Stratix II GX devices use an internal oscillator to generate DCLK. Using the MSEL[] pins, you can select to use either a 40- or 20-MHz oscillator.

Only the EPCS16 and EPCS64 devices support a DCLK up to 40-MHz clock; other EPCS devices support a DCLK up to 20-MHz. Refer to the *Serial Configuration Devices Data Sheet* for more information. The EPCS4 device only supports the smallest Stratix II (EP2S15) device, which is when the SOF compression is enabled. Because of its insufficient memory capacity, the EPCS1 device does not support any Stratix II devices.

Table 13–12 shows the active serial DCLK output frequencies.

Table 13–12. Active Serial DCLK Output Frequency

Oscillator	Minimum	Typical	Maximum	Units
40 MHz *(1)*	20	26	40	MHz
20 MHz	10	13	20	MHz

Note to Table 13–12:

(1) Only the EPCS16 and EPCS64 devices support a DCLK up to 40-MHz clock; other EPCS devices support a DCLK up to 20-MHz. Refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS16, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook* for more information.

In both AS and fast AS configuration schemes, the serial configuration device latches input and control signals on the rising edge of DCLK and drives out configuration data on the falling edge. Stratix II and Stratix II GX devices drive out control signals on the falling edge of DCLK and latch configuration data on the falling edge of DCLK.

In configuration mode, Stratix II and Stratix II GX devices enable the serial configuration device by driving the nCSO output pin low, which connects to the chip select (nCS) pin of the configuration device. The Stratix II and Stratix II GX devices use the serial clock (DCLK) and serial data output (ASDO) pins to send operation commands and/or read address signals to the serial configuration device. The configuration device provides data on its serial data output (DATA) pin, which connects to the DATA0 input of the Stratix II and Stratix II GX devices.

After all configuration bits are received by the Stratix II or Stratix II GX device, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ resistor. Initialization begins only after the CONF_DONE signal reaches a logic high level. All AS configuration pins, DATA0, DCLK, nCSO, and ASDO, have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and are driven high by the weak internal pull-up resistors. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

In Stratix II and Stratix II GX devices, the initialization clock source is either the 10-MHz (typical) internal oscillator (separate from the active serial internal oscillator) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If the internal oscillator is used, the Stratix II or Stratix II GX device provides itself with enough clock cycles for proper initialization. You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. The **Enable user-supplied start-up clock** (**CLKUSR**) option can be turned on in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. When you **Enable** the **user supplied start-up clock** option, the CLKUSR pin is the initialization clock source. Supplying a clock on CLKUSR will not affect the configuration process. After all configuration data has been accepted and CONF_DONE goes high, CLKUSR is enabled after 600 ns. After this time period elapses, Stratix II and Stratix II GX devices require 299 clock cycles to initialize properly and enter user mode. Stratix II and Stratix II GX devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If the INIT_DONE pin is used, it will be high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. Once the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high. This low-to-high transition signals that the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

If an error occurs during configuration, Stratix II and Stratix II GX devices assert the nSTATUS signal low, indicating a data frame error, and the CONF_DONE signal stays low. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the

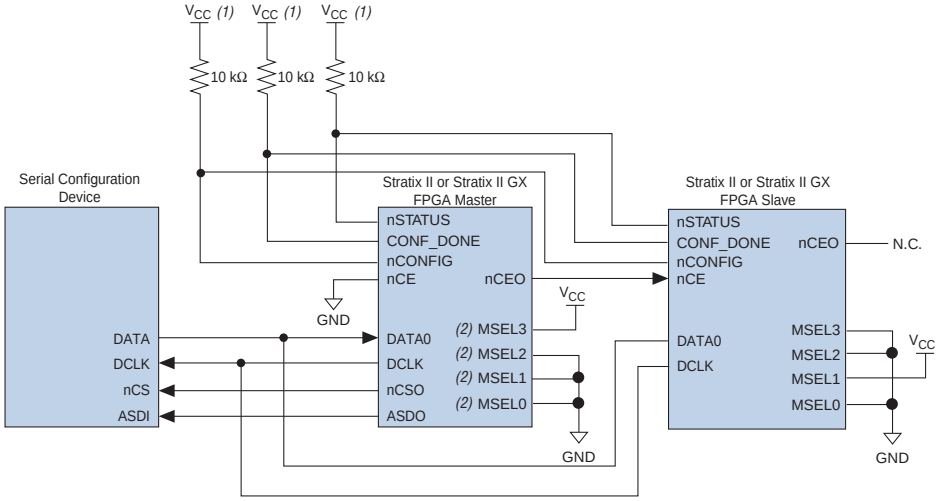
Device & Pin Options dialog box) is turned on, the Stratix II or Stratix II GX device resets the configuration device by pulsing nCSO, releases nSTATUS after a reset time-out period (maximum of 100 µs), and retries configuration. If this option is turned off, the system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration.

When the Stratix II or Stratix II GX device is in user mode, you can initiate reconfiguration by pulling the nCONFIG pin low. The nCONFIG pin should be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Once nCONFIG returns to a logic high level and nSTATUS is released by the Stratix II or Stratix II GX device, reconfiguration begins.

You can configure multiple Stratix II or Stratix II GX devices using a single serial configuration device. You can cascade multiple Stratix II or Stratix II GX devices using the chip-enable (nCE) and chip-enable-out (nCEO) pins. The first device in the chain must have its nCE pin connected to ground. You must connect its nCEO pin to the nCE pin of the next device in the chain. When the first device captures all of its configuration data from the bit stream, it drives the nCEO pin low, enabling the next device in the chain. You must leave the nCEO pin of the last device unconnected. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA0 pins of each device in the chain are connected (refer to Figure 13–13).

This first Stratix II or Stratix II GX device in the chain is the configuration master and controls configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Stratix II or Stratix II GX devices are configuration slaves and you must connect their MSEL pins to select the PS configuration scheme. Any other Altera device that supports PS configuration can also be part of the chain as a configuration slave. Figure 13–13 shows the pin connections for this setup.

Figure 13–13. Multi-Device AS Configuration



Notes to Figure 13–13:

(1) Connect the pull-up resistors to a 3.3-V supply.

(2) If using an EPCS4 device, `MSEL[3..0]` should be set to 1101. Refer to Table 13–11 for more details.

As shown in Figure 13–13, the `nSTATUS` and `CONF_DONE` pins on all target devices are connected together with external pull-up resistors. These pins are open-drain bidirectional pins on the devices. When the first device asserts `nCEO` (after receiving all of its configuration data), it releases its `CONF_DONE` pin. But the subsequent devices in the chain keep this shared `CONF_DONE` line low until they have received their configuration data. When all target devices in the chain have received their configuration data and have released `CONF_DONE`, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode.

If an error occurs at any point during configuration, the `nSTATUS` line is driven low by the failing device. If you enable the Auto-restart configuration after error option, reconfiguration of the entire chain begins after a reset time-out period (a maximum of 100 µs). If the **Auto-restart configuration after error** option is turned off, the external system must monitor `nSTATUS` for errors and then pulse `nCONFIG` low to restart configuration. The external system can pulse `nCONFIG` if it is under system control rather than tied to V_{CC}.

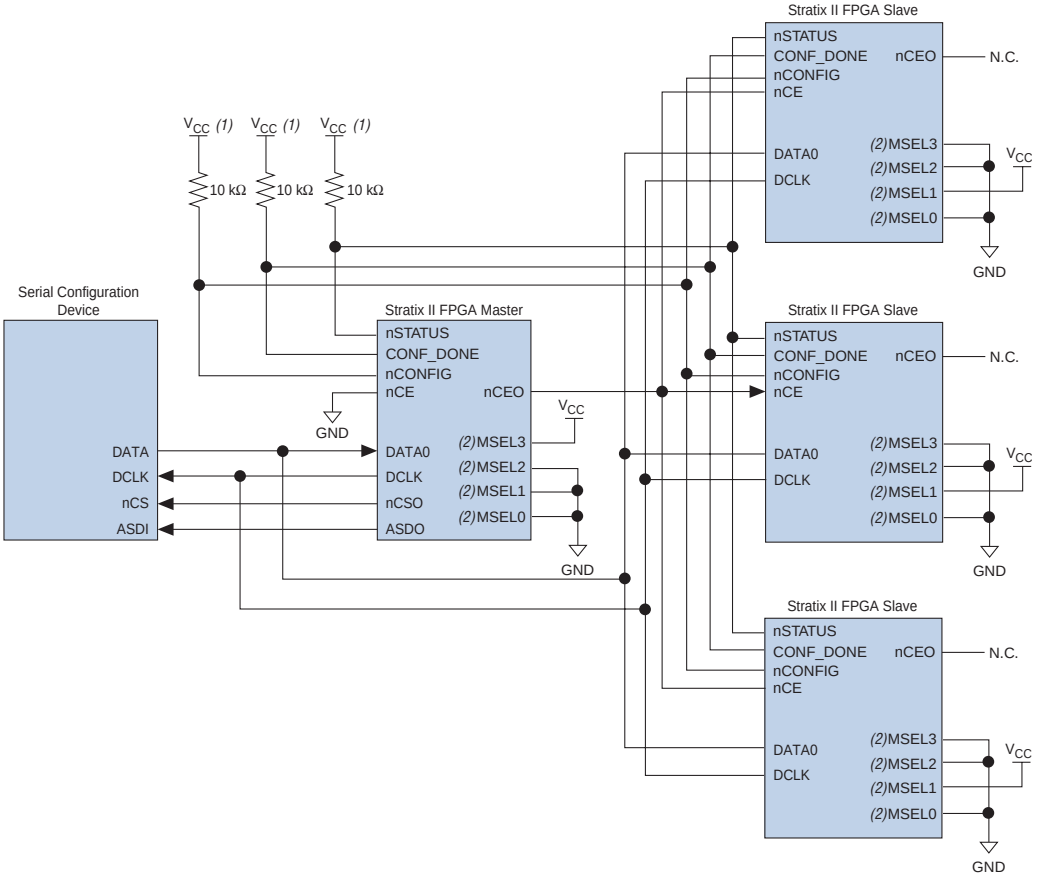
While you can cascade Stratix II or Stratix II GX devices, serial configuration devices cannot be cascaded or chained together.

If the configuration bit stream size exceeds the capacity of a serial configuration device, you must select a larger configuration device and/or enable the compression feature. When configuring multiple devices, the size of the bitstream is the sum of the individual devices' configuration bitstreams.

A system may have multiple devices that contain the same configuration data. In active serial chains, this can be implemented by storing two copies of the SOF in the serial configuration device. The first copy would configure the master Stratix II or Stratix II GX device, and the second copy would configure all remaining slave devices concurrently. All slave devices must be the same density and package. The setup is similar to Figure 13–13, where the master is set up in active serial mode and the slave devices are set up in passive serial mode.

To configure four identical Stratix II or Stratix II GX devices with the same SOF, you could set up the chain similar to the example shown in Figure 13–14. The first device is the master device and its MSEL pins should be set to select AS configuration. The other three slave devices are set up for concurrent configuration and its MSEL pins should be set to select PS configuration. The nCEO pin from the master device drives the nCE input pins on all three slave devices, and the DATA and DCLK pins connect in parallel to all four devices. During the first configuration cycle, the master device reads its configuration data from the serial configuration device while holding nCEO high. After completing its configuration cycle, the master drives nCE low and transmits the second copy of the configuration data to all three slave devices, configuring them simultaneously.

Figure 13–14. Multi-Device AS Configuration When devices Receive the Same Data



Notes to Figure 13–14:

(1) Connect the pull-up resistors to a 3.3-V supply.

(2) If using an EPCS4 device, `MSEL[3..0]` should be set to 1101. Refer to Table 13–11 for more details.

Estimating Active Serial Configuration Time

Active serial configuration time is dominated by the time it takes to transfer data from the serial configuration device to the Stratix II device. This serial interface is clocked by the Stratix II DCLK output (generated from an internal oscillator). As listed in Table 13–12 on page 13–37, the DCLK minimum frequency when choosing to use the 40-MHz oscillator is 20 MHz (50 ns). Therefore, the maximum configuration time estimate for an EP2S15 device (5 MBits of uncompressed data) is:

RBF Size (minimum DCLK period / 1 bit per DCLK cycle) = estimated maximum configuration time

5 Mbits × (50 ns / 1 bit) = 250 ms

To estimate the typical configuration time, use the typical DCLK period as listed in Table 13–12. With a typical DCLK period of 38.46 ns, the typical configuration time is 192 ms. Enabling compression reduces the amount of configuration data that is transmitted to the Stratix II or Stratix II GX device, which also reduces configuration time. On average, compression reduces configuration time by 50%.

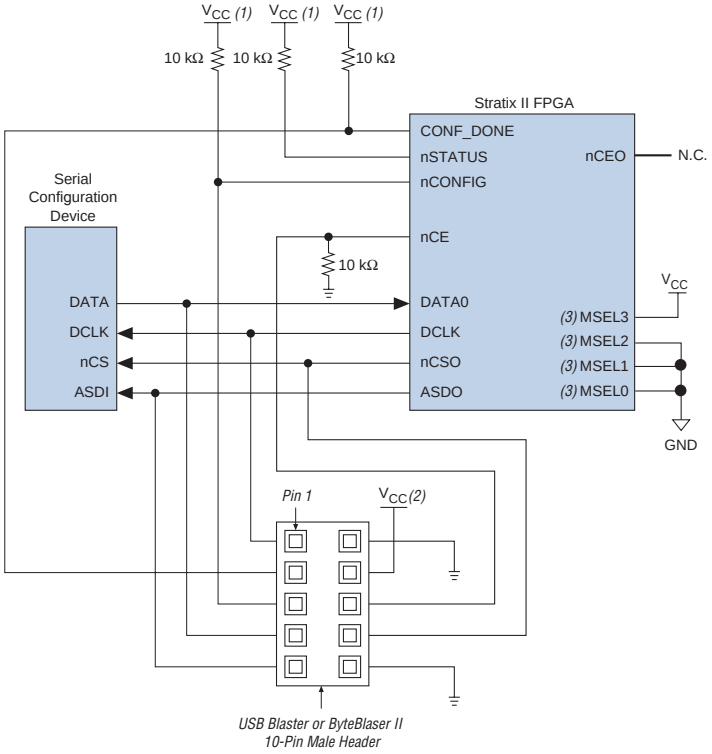
Programming Serial Configuration Devices

Serial configuration devices are non-volatile, flash-memory-based devices. You can program these devices in-system using the USB-Blaster™ or ByteBlaster™ II download cable. Alternatively, you can program them using the Altera Programming Unit (APU), supported third-party programmers, or a microprocessor with the SRunner software driver.

You can perform in-system programming of serial configuration devices via the AS programming interface. During in-system programming, the download cable disables device access to the AS interface by driving the nCE pin high. Stratix II and Stratix II GX devices are also held in reset by a low level on nCONFIG. After programming is complete, the download cable releases nCE and nCONFIG, allowing the pull-down and pull-up resistors to drive GND and V_{CC}, respectively. Figure 13–15 shows the download cable connections to the serial configuration device.

For more information on the USB Blaster download cable, refer to the *USB-Blaster Download Cable User Guide*. For more information on the ByteBlaster II cable, refer to the *ByteBlaster II Download Cable User Guide*.

Figure 13–15. In-System Programming of Serial Configuration Devices



Notes to Figure 13–15:

(1) Connect these pull-up resistors to 3.3-V supply.

(2) Power up the ByteBlaster II cable's V_{CC} with a 3.3-V supply.

(3) If using an EPCS4 device, `MSEL[3..0]` should be set to `1101`. Refer to Table 13–11 for more details.

You can program serial configuration devices with the Quartus II software with the Altera programming hardware (APU) and the appropriate configuration device programming adapter. The EPCS1 and EPCS4 devices are offered in an eight-pin small outline integrated circuit (SOIC) package.

In production environments, serial configuration devices can be programmed using multiple methods. Altera programming hardware or other third-party programming hardware can be used to program blank serial configuration devices before they are mounted onto printed circuit boards (PCBs). Alternatively, you can use an on-board microprocessor to program the serial configuration device in-system using C-based software drivers provided by Altera.

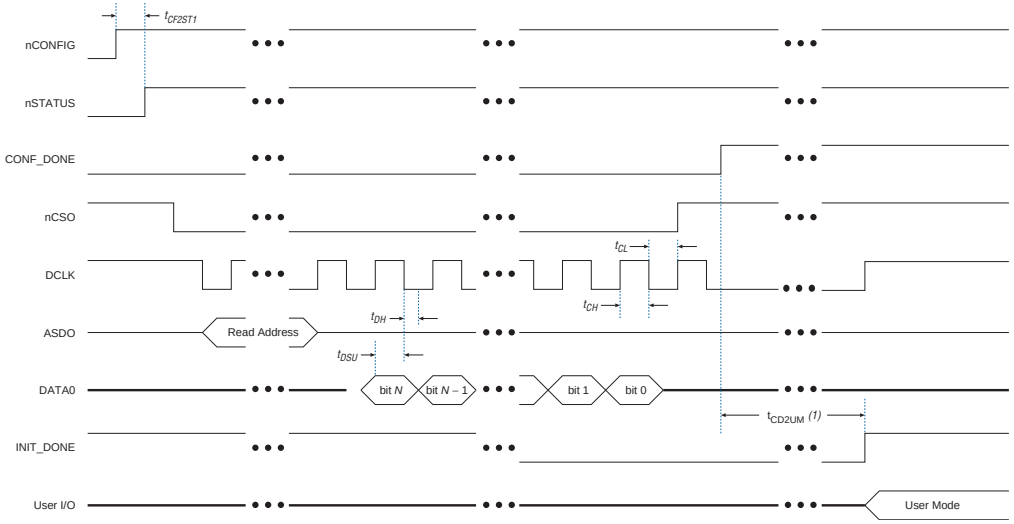
A serial configuration device can be programmed in-system by an external microprocessor using SRunner. SRunner is a software driver developed for embedded serial configuration device programming, which can be easily customized to fit in different embedded systems. SRunner is able to read a raw programming data (.**rpd**) file and write to the serial configuration devices. The serial configuration device programming time using SRunner is comparable to the programming time with the Quartus II software.

For more information about SRunner, refer to *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming* and the source code on the Altera web site at www.altera.com.

For more information on programming serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in the *Configuration Handbook*.

Figure 13–16 shows the timing waveform for the AS configuration scheme using a serial configuration device.

Figure 13–16. AS Configuration Timing



Note to Figure 13–16:

(1) The initialization clock can come from the Stratix II or Stratix II GX device's internal oscillator or the CLKUSR pin.

Table 13–13 shows the AS timing parameters for Stratix II devices.

Table 13–13. AS Timing Parameters for Stratix II Devices

Symbol	Parameter	Condition	Minimum	Typical	Maximum
t_{CF2ST1}	nCONFIG high to nSTATUS high				100
t_{DSU}	Data setup time before falling edge on DCLK		7		
t_{DH}	Data hold time after falling edge on DCLK		0		
t_{CH}	DCLK high time		10		
t_{CL}	DCLK low time		10		
t_{CD2UM}	CONF_DONE high to user mode		20		100

Passive Serial Configuration

PS configuration of Stratix II and Stratix II GX devices can be performed using an intelligent host, such as a MAX II device or microprocessor with flash memory, an Altera configuration device, or a download cable. In the PS scheme, an external host (MAX II device, embedded processor, configuration device, or host PC) controls configuration. Configuration data is clocked into the target Stratix II or Stratix II GX device via the DATA0 pin at each rising edge of DCLK.

The Stratix II and Stratix II GX decompression and design security features are fully available when configuring your Stratix II or Stratix II GX device using PS mode.

Table 13–14 shows the MSEL pin settings when using the PS configuration scheme.

Table 13–14. Stratix II and Stratix II GX MSEL Pin Settings for PS Configuration Schemes

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0
PS	0	0	1	0
PS when using Remote System Upgrade *(1)*	0	1	1	0

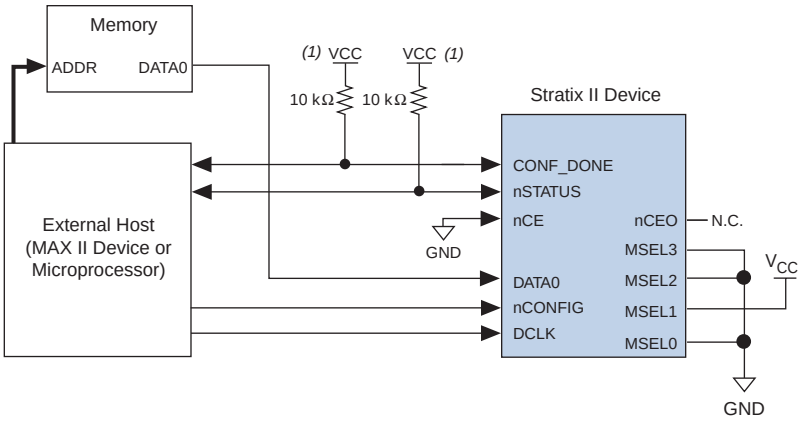
Note to Table 13–14:

(1) This scheme requires that you drive the RUnLU pin to specify either remote update or local update. For more information about remote system upgrade in Stratix II devices, refer to the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

PS Configuration Using a MAX II Device as an External Host

In the PS configuration scheme, a MAX II device can be used as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix II or Stratix II GX device. Configuration data can be stored in RBF, HEX, or TTF format. Figure 13–17 shows the configuration interface connections between a Stratix II or Stratix II GX device and a MAX II device for single device configuration.

Figure 13–17. Single Device PS Configuration Using an External Host



Note to Figure 13–17:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

Upon power-up, Stratix II and Stratix II GX devices go through a POR. The POR delay is dependent on the PORSEL pin setting; when PORSEL is driven low, the POR time is approximately 100 ms, if PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. Once the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

You can hold nConfig low in order to stop device configuration.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *Stratix II Device Handbook* or the *Stratix II GX Device Handbook*.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration, the MAX II device must generate a low-to-high transition on the nCONFIG pin.

V_{CCINT}, V_{CCIO}, and V_{CCPD} of the banks where the configuration and JTAG pins reside need to be fully powered to the appropriate voltage levels in order to begin the configuration process.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. Once nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the MAX II device should place the configuration data one bit at a time on the DATA0 pin. If you are using configuration data in RBF, HEX, or TTF format, you must send the least significant bit (LSB) of each data byte first. For example, if the RBF contains the byte sequence `02 1B EE 01 FA`, the serial bitstream you should transmit to the device is `0100-0000 1101-1000 0111-0111 1000-0000 0101-1111`.

The Stratix II and Stratix II GX devices receive configuration data on the DATA0 pin and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. Data is continuously clocked into the target device until CONF_DONE goes high. After the device has received all configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

In Stratix II and Stratix II GX devices, the initialization clock source is either the internal oscillator (typically 10 MHz) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If the internal oscillator is used, the Stratix II or Stratix II GX device provides itself with enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. The **Enable user-supplied start-up clock** (**CLKUSR**) option can be turned on in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. Supplying a clock on CLKUSR will not affect the configuration process. After all configuration data has been accepted and CONF_DONE goes high, CLKUSR will be enabled after the time specified as t_{CD2CU}. After this time period elapses, Stratix II and Stratix II GX devices require 299 clock cycles to initialize properly and enter user mode. Stratix II and Stratix II GX devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If the INIT_DONE pin is used it will be high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. Once the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin will go low. When initialization is complete, the INIT_DONE pin will be released and pulled high. The MAX II device must be able to detect this low-to-high transition which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user-mode, the user I/O pins will no longer have weak pull-up resistors and will function as assigned in your design.

To ensure DCLK and DATA0 are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When the PS scheme is chosen in the Quartus II software, as a default this I/O pin is tri-stated in user mode and should be driven by the MAX II device. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device & Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box) is turned on, the Stratix II or Stratix II GX device releases nSTATUS after a reset time-out period (maximum of 100 µs). After nSTATUS is released and

pulled high by a pull-up resistor, the MAX II device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

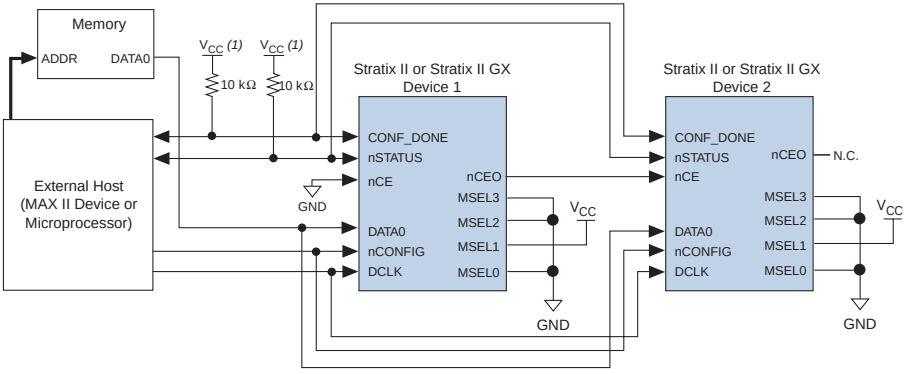
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The CONF_DONE pin must be monitored by the MAX II device to detect errors and determine when programming completes. If all configuration data is sent, but CONF_DONE or INIT_DONE have not gone high, the MAX II device must reconfigure the target device.

If the optional CLKUSR pin is being used and nCONFIG is pulled low to restart configuration during device initialization, you need to ensure that CLKUSR continues toggling during the time nSTATUS is low (maximum of 100 µs).

When the device is in user-mode, you can initiate a reconfiguration by transitioning the nCONFIG pin low-to-high. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Once nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 13–18 shows how to configure multiple devices using a MAX II device. This circuit is similar to the PS configuration circuit for a single device, except Stratix II or Stratix II GX devices are cascaded for multi-device configuration.

Figure 13–18. Multi-Device PS Configuration Using an External Host



Note to Figure 13–18:

(1) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

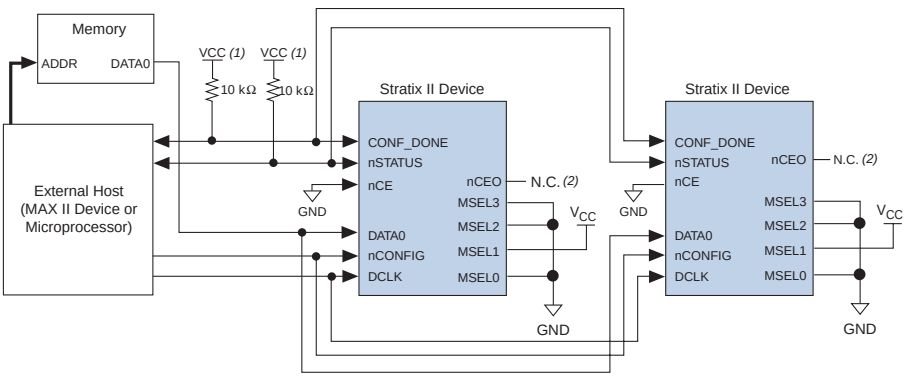
In multi-device PS configuration the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration within one clock cycle. Therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

Since all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, configuration stops for the entire chain and the entire chain must be reconfigured. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (maximum of 100 µs). After all nSTATUS pins are released and pulled high, the MAX II device can try to reconfigure the chain without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

In your system, you can have multiple devices that contain the same configuration data. To support this configuration scheme, all device nCE inputs are tied to GND, while nCEO pins are left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices will start and complete configuration at the same time. Figure 13–19 shows multi-device PS configuration when both Stratix II or Stratix II GX devices are receiving the same configuration data.

Figure 13–19. Multiple-Device PS Configuration When Both devices Receive the Same Data



Notes to Figure 13–19:

(1) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

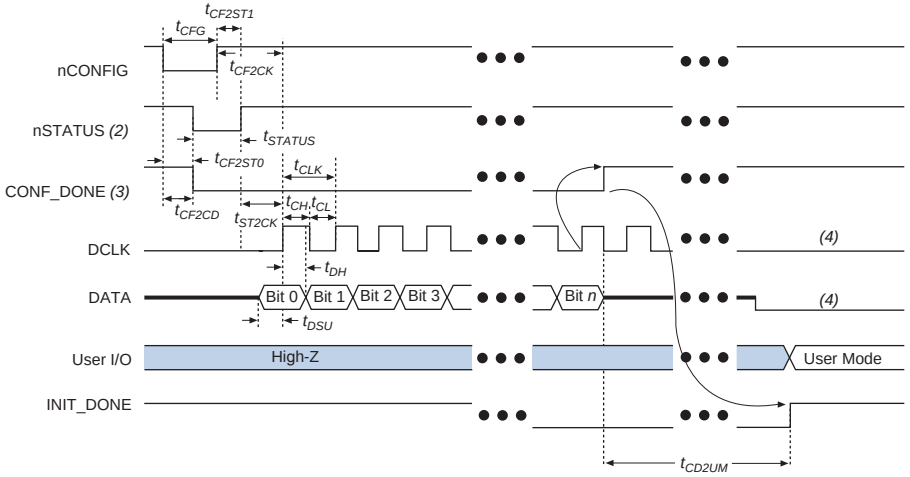
You can use a single configuration chain to configure Stratix II or Stratix II GX devices with other Altera devices. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device initiates reconfiguration in all devices, all of the device CONF_DONE and nSTATUS pins must be tied together.

For more information on configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

PS Configuration Timing

Figure 13–20 shows the timing waveform for PS configuration when using a MAX II device as an external host.

Figure 13–20. PS Configuration Timing Waveform *Note (1)*



Notes to Figure 13–20:

(1) The beginning of this waveform shows the device in user-mode. In user-mode, nCONFIG, nSTATUS and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.
(2) Upon power-up, the Stratix II or Stratix II GX device holds nSTATUS low for the time of the POR delay.
(3) Upon power-up, before and during configuration, CONF_DONE is low.
(4) DCLK should not be left floating after configuration. It should be driven high or low, whichever is more convenient. DATA[0] is available as a user I/O pin after configuration and the state of this pin depends on the dual-purpose pin settings.

Table 13–15 defines the timing parameters for Stratix II and Stratix II GX devices for PS configuration.

Table 13–15. PS Timing Parameters for Stratix II and Stratix II GX Devices (Part 1 of 2)

Symbol	Parameter	Min	Max	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low		800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low		800	ns

Table 13–15. PS Timing Parameters for Stratix II and Stratix II GX Devices (Part 2 of 2)

Symbol	Parameter	Min	Max	Units
t_{CFG}	nCONFIG low pulse width	2		µs
t_{STATUS}	nSTATUS low pulse width	10	100 *(1)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high		100 *(1)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	100		µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2		µs
t_{DSU}	Data setup time before rising edge on DCLK	5		ns
t_{DH}	Data hold time after rising edge on DCLK	0		ns
t_{CH}	DCLK high time	4		ns
t_{CL}	DCLK low time	4		ns
t_{CLK}	DCLK period	10		ns
f_{MAX}	DCLK frequency		100	MHz
t_R	Input rise time		40	ns
t_F	Input fall time		40	ns
t_{CD2UM}	CONF_DONE high to user mode *(2)*	20	100	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period		
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (299 × CLKUSR period)		

Notes to Table 13–15:

(1) This value is applicable if users do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(2) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting the device.

Device configuration options and how to create configuration files are discussed further in *Software Settings* in volume 2 of the *Configuration Handbook*.

An example PS design that uses a MAX II device as the external host for configuration will be available when devices are available.

PS Configuration Using a Microprocessor

In the PS configuration scheme, a microprocessor can control the transfer of configuration data from a storage device, such as flash memory, to the target Stratix II or Stratix II GX device.

All information in the "PS Configuration Using a MAX II Device as an External Host" section is also applicable when using a microprocessor as an external host. Refer to that section for all configuration and timing information.

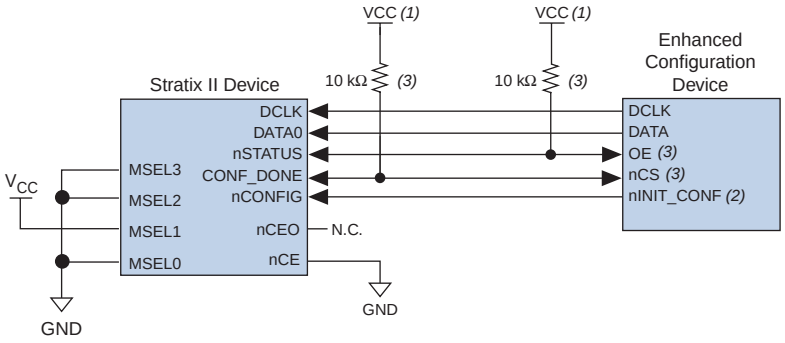
PS Configuration Using a Configuration Device

You can use an Altera configuration device, such as an enhanced configuration device, to configure Stratix II and Stratix II GX devices using a serial configuration bitstream. Configuration data is stored in the configuration device. Figure 13–21 shows the configuration interface connections between a Stratix II or Stratix II GX device and a configuration device.

The figures in this chapter only show the configuration-related pins and the configuration pin connections between the configuration device and the device.

For more information on the enhanced configuration device and flash interface pins (such as `PGM[2..0]`, `EXCLK`, `PORSEL`, `A[20..0]`, and `DQ[15..0]`), refer to the *Enhanced Configuration Devices (EPC4, EPC8, EPC16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

Figure 13–21. Single Device PS Configuration Using an Enhanced Configuration Device



Notes to Figure 13–21:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active, meaning an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

The value of the internal pull-up resistors on the enhanced configuration devices can be found in the Operating Conditions table of the *Enhanced Configuration Devices (EPC4, EPC8, & EPC16) Data Sheet* chapter in volume 2 of the *Configuration Handbook* or the *Configuration Devices for SRAM-based LUT Devices Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

When using enhanced configuration devices, nCONFIG of the device can be connected to nINIT_CONF of the configuration device, which allows the INIT_CONF JTAG instruction to initiate device configuration. The nINIT_CONF pin does not need to be connected if its functionality is not used. An internal pull-up resistor on the nINIT_CONF pin is always active in enhanced configuration devices, which means an external pull-up resistor should not be used if nCONFIG is tied to nINIT_CONF.

Upon power-up, the Stratix II and Stratix II GX devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. If PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS low, and tri-state all user I/O pins. The configuration device also goes through a POR delay to allow the power supply to stabilize. The POR time for EPC2 devices is 200 ms (maximum). The POR time for enhanced configuration devices can be set to either

100 ms or 2 ms, depending on its PORSEL pin setting. If the PORSEL pin is connected to GND, the POR delay is 100 ms. If the PORSEL pin is connected to V_{CC}, the POR delay is 2 ms. During this time, the configuration device drives its OE pin low. This low signal delays configuration because the OE pin is connected to the target device's nSTATUS pin.

When selecting a POR time, you need to ensure that the device completes power-up before the enhanced configuration device exits POR. Altera recommends that you choose a POR time for the Stratix II or Stratix II GX device of 12 ms, while selecting a POR time for the enhanced configuration device of 100 ms.

When both devices complete POR, they release their open-drain OE or nSTATUS pin, which is then pulled high by a pull-up resistor. Once the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *DC & Switching Characteristics* chapter in volume 2 of the *Stratix II Device Handbook* or the *DC & Switching Characteristics* chapter in volume 2 of the *Stratix II GX Device Handbook*.

When the power supplies have reached the appropriate operating voltages, the target device senses the low-to-high transition on nCONFIG and initiates the configuration cycle. The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. The beginning of configuration can be delayed by holding the nCONFIG or nSTATUS pin low.

To begin configuration, power the V_{CCINT}, V_{CCIO}, and V_{CCPD} voltages (for the banks where the configuration and JTAG pins reside) to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the nSTATUS pin, which is pulled high by a pull-up resistor. Enhanced configuration and EPC2 devices have an optional internal pull-up resistor on the OE pin. This option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If this internal pull-up resistor is not used, an external 10-kΩ pull-up resistor on the OE-nSTATUS line is required. Once nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins.

When nSTATUS is pulled high, OE of the configuration device also goes high and the configuration device clocks data out serially to the device using the Stratix II or Stratix II GX device's internal oscillator. The Stratix II and Stratix II GX devices receive configuration data on the DATA0 pin and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK.

After the device has received all configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by a pull-up resistor. Since CONF_DONE is tied to the configuration device's nCS pin, the configuration device is disabled when CONF_DONE goes high. Enhanced configuration and EPC2 devices have an optional internal pull-up resistor on the nCS pin. This option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If this internal pull-up resistor is not used, an external 10-kΩ pull-up resistor on the nCS-CONF_DONE line is required. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin.

In Stratix II and Stratix II GX devices, the initialization clock source is either the internal oscillator (typically 10 MHz) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you are using internal oscillator, the Stratix II or Stratix II GX device supplies itself with enough clock cycles for proper initialization. You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. Supplying a clock on CLKUSR will not affect the configuration process. After all configuration data has been accepted and CONF_DONE goes high, CLKUSR will be enabled after the time specified as t_{CD2CU}. After this time period elapses, the Stratix II and Stratix II GX devices require 299 clock cycles to initialize properly and enter user mode. Stratix II and Stratix II GX devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If you are using the INIT_DONE pin, it will be high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. Once the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high. This low-to-high transition signals that the device has entered user mode. In user-mode, the user I/O

pins will no longer have weak pull-up resistors and will function as assigned in your design. Enhanced configuration devices and EPC2 devices drive DCLK low and DATA0 high at the end of configuration.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. Since the nSTATUS pin is tied to OE, the configuration device will also be reset. If the **Auto-restart configuration after error** option, available in the Quartus II software, from the **General** tab of the **Device & Pin Options** dialog box is turned on, the device automatically initiates reconfiguration if an error occurs. The Stratix II and Stratix II GX devices release the nSTATUS pin after a reset time-out period (maximum of 100 µs). When the nSTATUS pin is released and pulled high by a pull-up resistor, the configuration device reconfigures the chain. If this option is turned off, the external system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration. The external system can pulse nCONFIG if nCONFIG is under system control rather than tied to V_{CC}.

In addition, if the configuration device sends all of its data and then detects that CONF_DONE has not gone high, it recognizes that the device has not configured successfully. Enhanced configuration devices wait for 64 DCLK cycles after the last configuration bit was sent for CONF_DONE to reach a high state. EPC2 devices wait for 16 DCLK cycles. In this case, the configuration device pulls its OE pin low, driving the target device's nSTATUS pin low. If the **Auto-restart configuration after error** option is set in the software, the target device resets and then releases its nSTATUS pin after a reset time-out period (maximum of 100 µs). When nSTATUS returns to a logic high level, the configuration device tries to reconfigure the device.

When CONF_DONE is sensed low after configuration, the configuration device recognizes that the target device has not configured successfully. Therefore, your system should not pull CONF_DONE low to delay initialization. Instead, use the CLKUSR option to synchronize the initialization of multiple devices that are not in the same configuration chain. Devices in the same configuration chain will initialize together if their CONF_DONE pins are tied together.

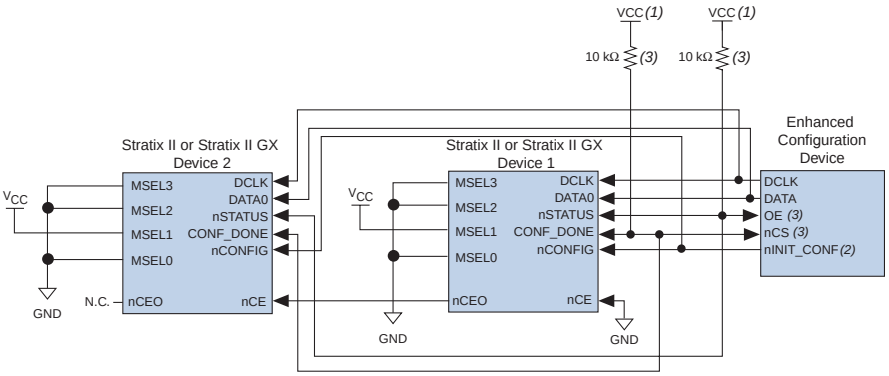
If you are using the optional CLKUSR pin and nCONFIG is pulled low to restart configuration during device initialization, you need to ensure that CLKUSR continues toggling during the time nSTATUS is low (maximum of 100 µs).

When the device is in user-mode, pulling the nCONFIG pin low initiates a reconfiguration. The nCONFIG pin should be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Because CONF_DONE is pulled low, this

activates the configuration device because it sees its nCS pin drive low. Once nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 13–22 shows how to configure multiple devices with an enhanced configuration device. This circuit is similar to the configuration device circuit for a single device, except Stratix II or Stratix II GX devices are cascaded for multi-device configuration.

Figure 13–22. Multi-Device PS Configuration Using an Enhanced Configuration Device



Notes to Figure 13–22:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active, meaning an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

Enhanced configuration devices cannot be cascaded.

When performing multi-device configuration, you must generate the configuration device's POF from each project's SOF. You can combine multiple SOFs using the **Convert Programming Files** window in the Quartus II software.

For more information on how to create configuration files for multi-device configuration chains, refer to the *Software Settings* chapter of the *Configuration Handbook*.

In multi-device PS configuration, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, prompting the second device to begin configuration. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device.

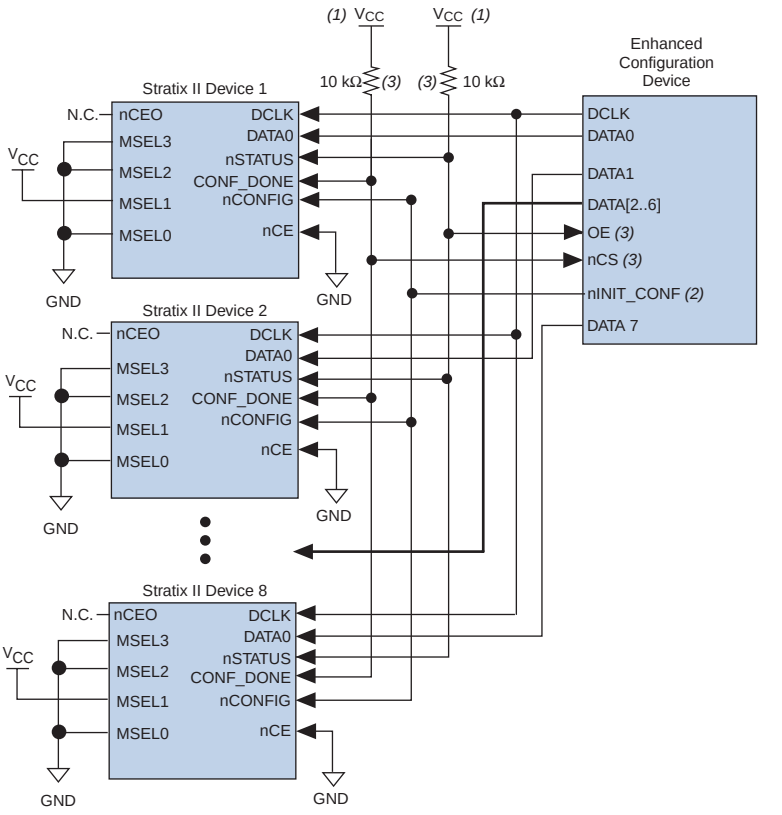
When configuring multiple devices, configuration does not begin until all devices release their OE or nSTATUS pins. Similarly, since all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

Since all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, configuration stops for the entire chain and the entire chain must be reconfigured. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This low signal drives the OE pin low on the enhanced configuration device and drives nSTATUS low on all devices, causing them to enter a reset state. This behavior is similar to a single device detecting an error.

If the **Auto-restart configuration after error** option is turned on, the devices will automatically initiate reconfiguration if an error occurs. The devices will release their nSTATUS pins after a reset time-out period (maximum of 100 µs). When all the nSTATUS pins are released and pulled high, the configuration device tries to reconfigure the chain. If the **Auto-restart configuration after error** option is turned off, the external system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration. The external system can pulse nCONFIG if nCONFIG is under system control rather than tied to V_{CC}.

The enhanced configuration devices also support parallel configuration of up to eight devices. The n-bit (n = 1, 2, 4, or 8) PS configuration mode allows enhanced configuration devices to concurrently configure devices or a chain of devices. In addition, these devices do not have to be the same device family or density as they can be any combination of Altera devices. An individual enhanced configuration device DATA line is available for each targeted device. Each DATA line can also feed a daisy chain of devices. Figure 13–23 shows how to concurrently configure multiple devices using an enhanced configuration device.

Figure 13–23. Concurrent PS Configuration of Multiple Devices Using an Enhanced Configuration Device



Notes to Figure 13–23:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active, meaning an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

The Quartus II software only allows the selection of n-bit PS configuration modes, where n must be 1, 2, 4, or 8. However, you can use these modes to configure any number of devices from 1 to 8. When configuring SRAM-based devices using n-bit PS modes, use Table 13–16 to select the appropriate configuration mode for the fastest configuration times.

Table 13–16. Recommended Configuration Using n-Bit PS Modes

Number of Devices *(1)*	Recommended Configuration Mode
1	1-bit PS
2	2-bit PS
3	4-bit PS
4	4-bit PS
5	8-bit PS
6	8-bit PS
7	8-bit PS
8	8-bit PS

Note to Table 13–16:

(1) Assume that each DATA line is only configuring one device, not a daisy chain of devices.

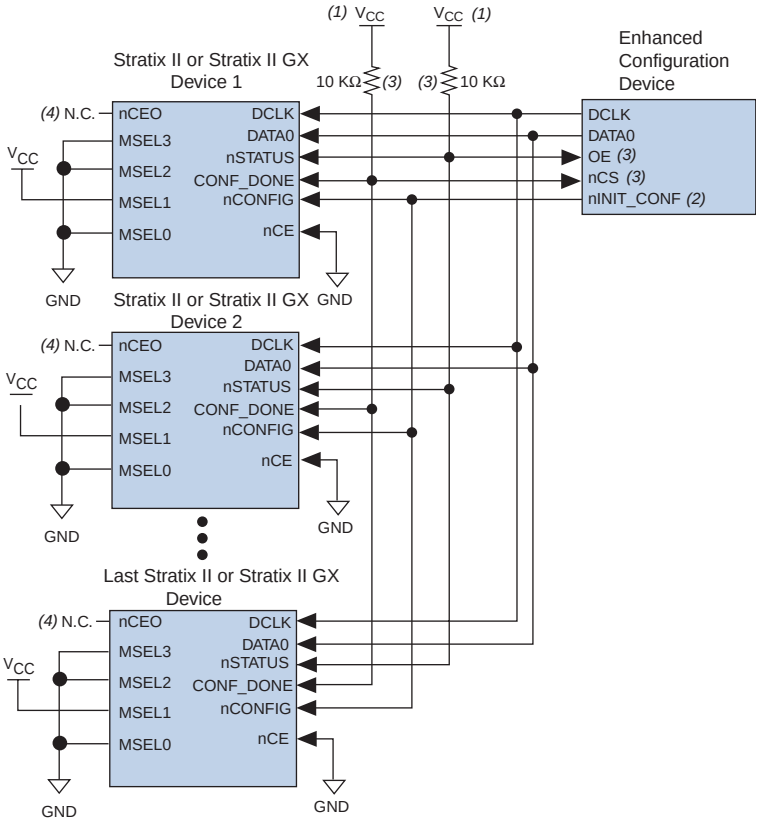
For example, if you configure three devices, you would use the 4-bit PS mode. For the DATA0, DATA1, and DATA2 lines, the corresponding SOF data is transmitted from the configuration device to the device. For DATA3, you can leave the corresponding Bit3 line blank in the Quartus II software. On the PCB, leave the DATA3 line from the enhanced configuration device unconnected.

Alternatively, you can daisy chain two devices to one DATA line while the other DATA lines drive one device each. For example, you could use the 2-bit PS mode to drive two devices with DATA Bit0 (two EP2S15 devices) and the third device (EP2S30 device) with DATA Bit1. This 2-bit PS configuration scheme requires less space in the configuration flash memory, but can increase the total system configuration time.

A system may have multiple devices that contain the same configuration data. To support this configuration scheme, all device nCE inputs are tied to GND, while nCEO pins are left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices will start and complete

configuration at the same time. Figure 13–24 shows multi-device PS configuration when the Stratix II or Stratix II GX devices are receiving the same configuration data.

Figure 13–24. Multiple-Device PS Configuration Using an Enhanced Configuration Device When devices Receive the Same Data



Notes to Figure 13–24:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin is available on enhanced configuration devices and has an internal pull-up resistor that is always active, meaning an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(3) The enhanced configuration devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

(4) The nCEO pins of all devices are left unconnected when configuring the same configuration data into multiple devices.

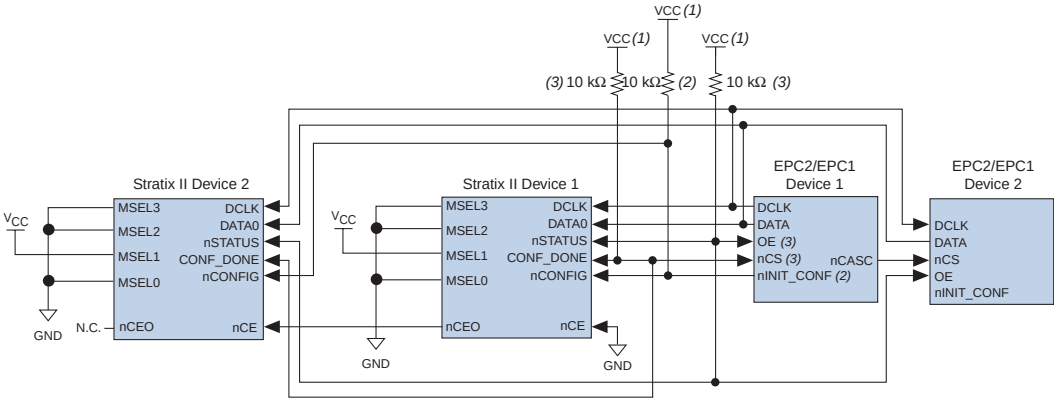
You can cascade several EPC2 devices to configure multiple Stratix II or Stratix II GX devices. The first configuration device in the chain is the master configuration device, while the subsequent devices are the slave devices. The master configuration device sends DCLK to the Stratix II or Stratix II GX devices and to the slave configuration devices. The first EPC device's nCS pin is connected to the CONF_DONE pins of the devices, while its nCASC pin is connected to nCS of the next configuration device in the chain. The last device's nCS input comes from the previous device, while its nCASC pin is left floating. When all data from the first configuration device is sent, it drives nCASC low, which in turn drives nCS on the next configuration device. A configuration device requires less than one clock cycle to activate a subsequent configuration device, so the data stream is uninterrupted.

Enhanced configuration devices cannot be cascaded.

Because all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, the master configuration device stops configuration for the entire chain and the entire chain must be reconfigured. For example, if the master configuration device does not detect CONF_DONE going high at the end of configuration, it resets the entire chain by pulling its OE pin low. This low signal drives the OE pin low on the slave configuration device(s) and drives nSTATUS low on all devices, causing them to enter a reset state. This behavior is similar to the device detecting an error in the configuration data.

Figure 13–25 shows how to configure multiple devices using cascaded EPC2 devices.

Figure 13–25. Multi-Device PS Configuration Using Cascaded EPC2 Devices



Notes to Figure 13–25:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin (available on enhanced configuration devices and EPC2 devices only) has an internal pull-up resistor that is always active, meaning an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used.

(3) The enhanced configuration devices' and EPC2 devices' OE and nCS pins have internal programmable pull-up resistors. External 10-kΩ pull-up resistors should be used. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

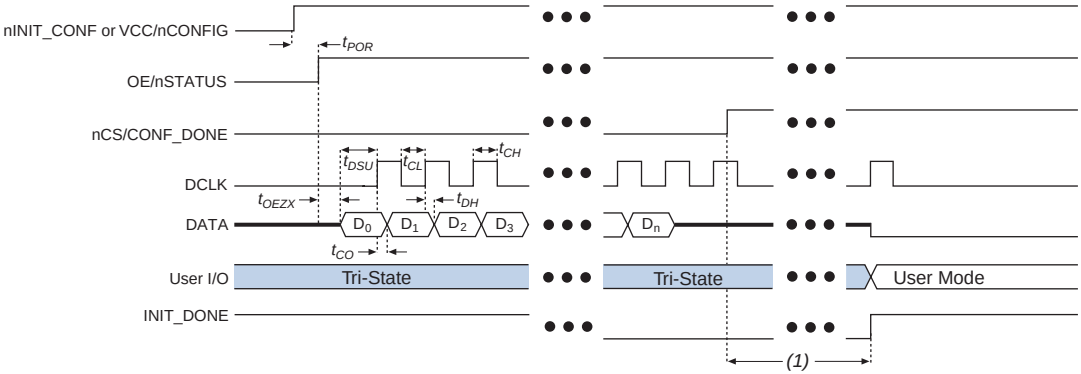
When using enhanced configuration devices or EPC2 devices, nCONFIG of the device can be connected to nINIT_CONF of the configuration device, allowing the INIT_CONF JTAG instruction to initiate device configuration. The nINIT_CONF pin does not need to be connected if its functionality is not used. An internal pull-up resistor on the nINIT_CONF pin is always active in the enhanced configuration devices and the EPC2 devices, which means that you shouldn't be using an external pull-up resistor if nCONFIG is tied to nINIT_CONF. If you are using multiple EPC2 devices to configure a Stratix II or Stratix II GX device(s), only the first EPC2 has its nINIT_CONF pin tied to the device's nCONFIG pin.

You can use a single configuration chain to configure Stratix II or Stratix II GX devices with other Altera devices. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device initiates reconfiguration in all devices, all of the device CONF_DONE and nSTATUS pins must be tied together.

For more information on configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in the *Configuration Handbook*.

Figure 13–26 shows the timing waveform for the PS configuration scheme using a configuration device.

Figure 13–26. Stratix II and Stratix II GX PS Configuration Using a Configuration Device Timing Waveform



Note to Figure 13–26:
(1) The initialization clock can come from the Stratix II or Stratix II GX device's internal oscillator or the CLKUSR pin.

For timing information, refer to the *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* chapter in volume 2 of the *Configuration Handbook* or the *Configuration Devices for SRAM-Based LUT Devices Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

Device configuration options and how to create configuration files are discussed further in the *Software Settings* chapter in volume 2 of the *Configuration Handbook*.

PS Configuration Using a Download Cable

In this section, the generic term "download cable" includes the Altera USB-Blaster™ universal serial bus (USB) port download cable, MasterBlaster™ serial/USB communications cable, ByteBlaster™ II parallel port download cable, and the ByteBlaster MV parallel port download cable.

In PS configuration with a download cable, an intelligent host (such as a PC) transfers data from a storage device to the device via the USB Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV cable.

Upon power-up, the Stratix II and Stratix II GX devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. If PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS low, and tri-state all user I/O pins. Once the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* and the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook.*

The configuration cycle consists of three stages: reset, configuration and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration in this scheme, the download cable generates a low-to-high transition on the nCONFIG pin.

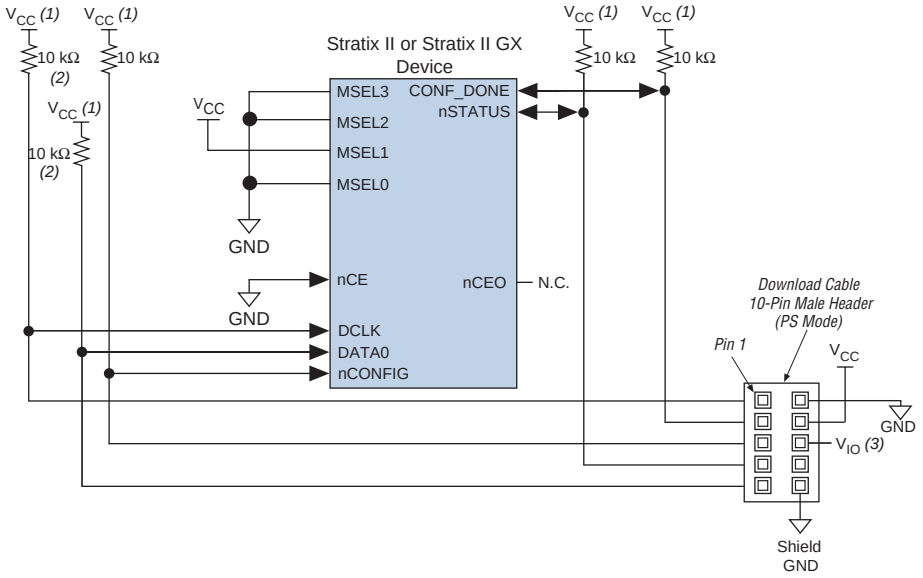
To begin configuration, power the V_{CCINT}, V_{CCIO}, and V_{CCPD} voltages (for the banks where the configuration and JTAG pins reside) to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. Once nSTATUS is released the device is ready to receive configuration data and the configuration stage begins. The programming hardware or download cable then places the configuration data one bit at a time on the device's DATA0 pin. The configuration data is clocked into the target device until CONF_DONE goes high. The CONF_DONE pin must have an external 10-kΩpull-up resistor in order for the device to initialize.

When using a download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs. Additionally, the **Enable user-supplied start-up clock** (**CLKUSR**) option has no affect on the device initialization since this option is disabled in the SOF when programming the device using the Quartus II programmer and download cable. Therefore, if you turn on the CLKUSR option, you do not need to provide a clock on CLKUSR when you are configuring the device with the Quartus II programmer and a

download cable. Figure 13–27 shows PS configuration for Stratix II or Stratix II GX devices using a USB Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV cable.

Figure 13–27. PS Configuration Using a USB Blaster, MasterBlaster, ByteBlaster II or ByteBlasterMV Cable



Notes to Figure 13–27:

(1) The pull-up resistor should be connected to the same supply voltage as the USB Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II or ByteBlasterMV cable.

(2) The pull-up resistors on DATA0 and DCLK are only needed if the download cable is the only configuration scheme used on your board. This ensures that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, the pull-up resistors on DATA0 and DCLK are not needed.

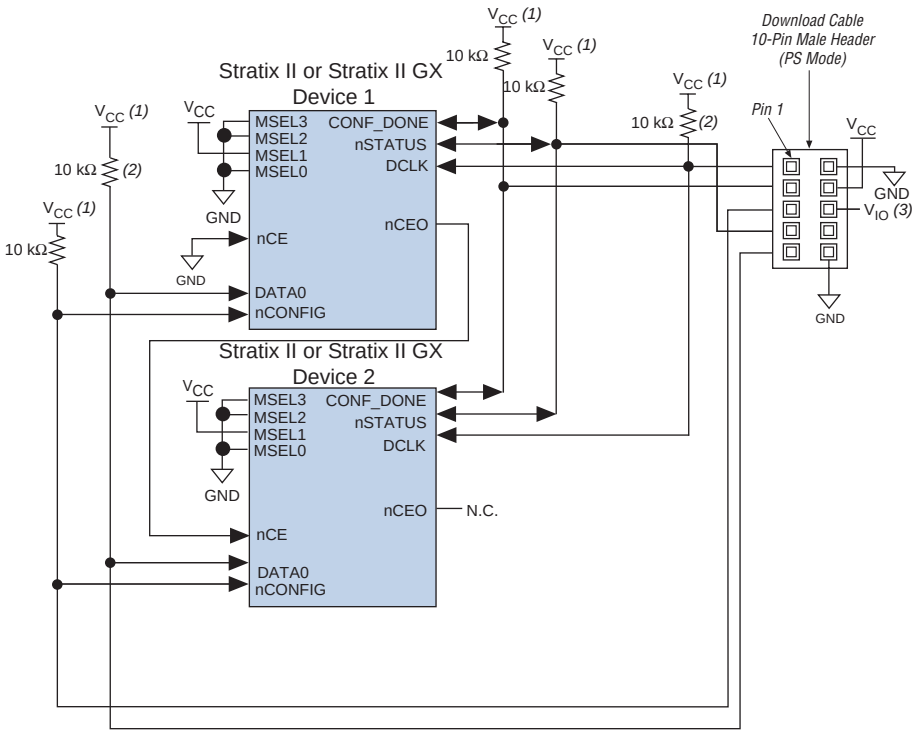
(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value. In the ByteBlasterMV cable, this pin is a no connect. In the USB Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for active serial programming, otherwise it is a no connect.

You can use a download cable to configure multiple Stratix II or Stratix II GX devices by connecting each device's nCEO pin to the subsequent device's nCE pin. The first device's nCE pin is connected to GND while its nCEO pin is connected to the nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. All other configuration pins, nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE are connected to every device in the chain. Because all CONF_DONE pins are tied together, all devices in the chain initialize and enter user mode at the same time.

In addition, because the nSTATUS pins are tied together, the entire chain halts configuration if any device detects an error. The **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs.

Figure 13–28 shows how to configure multiple Stratix II or Stratix II GX devices with a download cable.

Figure 13–28. Multi-Device PS Configuration using a USB Blaster, MasterBlaster, ByteBlaster II or ByteBlasterMV Cable



Notes to Figure 13–28:

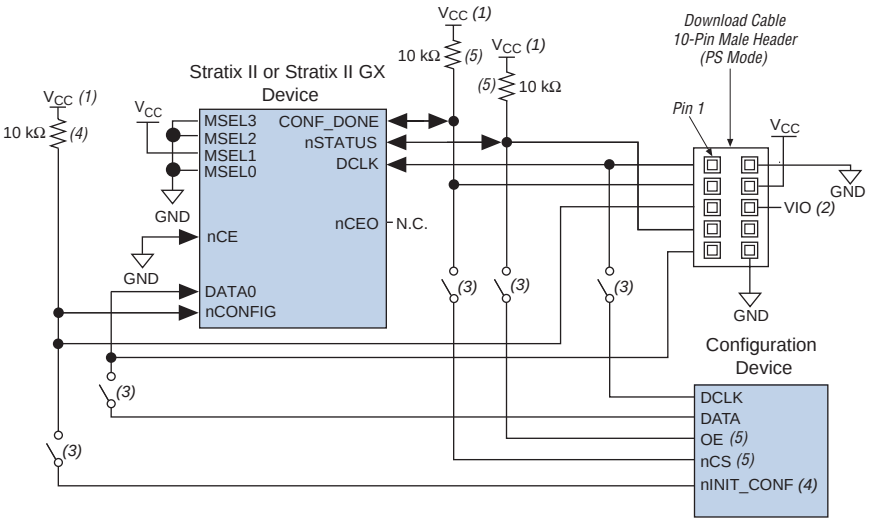
(1) The pull-up resistor should be connected to the same supply voltage as the USB Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II or ByteBlasterMV cable.

(2) The pull-up resistors on DATA0 and DCLK are only needed if the download cable is the only configuration scheme used on your board. This is to ensure that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, the pull-up resistors on DATA0 and DCLK are not needed.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value. In the ByteBlasterMV cable, this pin is a no connect. In the USB Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for active serial programming, otherwise it is a no connect.

If you are using a download cable to configure device(s) on a board that also has configuration devices, electrically isolate the configuration device from the target device(s) and cable. One way of isolating the configuration device is to add logic, such as a multiplexer, that can select between the configuration device and the cable. The multiplexer chip allows bidirectional transfers on the nSTATUS and CONF_DONE signals. Another option is to add switches to the five common signals (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) between the cable and the configuration device. The last option is to remove the configuration device from the board when configuring the device with the cable. Figure 13–29 shows a combination of a configuration device and a download cable to configure an device.

Figure 13–29. PS Configuration with a Download Cable and Configuration Device Circuit



Notes to Figure 13–29:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value. In the ByteBlasterMV cable, this pin is a no connect. In the USB Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for active serial programming, otherwise it is a no connect.

(3) You should not attempt configuration with a download cable while a configuration device is connected to a Stratix II or Stratix II GX device. Instead, you should either remove the configuration device from its socket when using the download cable or place a switch on the five common signals between the download cable and the configuration device.

(4) The nINIT_CONF pin (available on enhanced configuration devices and EPC2 devices only) has an internal pull-up resistor that is always active. This means an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its functionality is not used.

(5) The enhanced configuration devices' and EPC2 devices' OE and nCS pins have internal programmable pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. The internal pull-up resistors are used by default in the Quartus II software. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-up resistors on configuration device** option when generating programming files.

For more information on how to use the USB Blaster, MasterBlaster, ByteBlaster II or ByteBlasterMV cables, refer to the following data sheets:

- *USB Blaster Download Cable User Guide*
- *MasterBlaster Serial/USB Communications Cable User Guide*
- *ByteBlaster II Download Cable User Guide*
- *ByteBlasterMV Download Cable User Guide*

Passive Parallel Asynchronous Configuration

Passive parallel asynchronous (PPA) configuration uses an intelligent host, such as a microprocessor, to transfer configuration data from a storage device, such as flash memory, to the target Stratix II or Stratix II GX device.

Configuration data can be stored in RBF, HEX, or TTF format. The host system outputs byte-wide data and the accompanying strobe signals to the device. When using PPA, pull the DCLK pin high through a 10-kΩ pull-up resistor to prevent unused configuration input pins from floating.

You cannot use the Stratix II or Stratix II GX decompression and design security features if you are configuring your Stratix II or Stratix II GX device using PPA mode.

Table 13–17 shows the MSEL pin settings when using the PS configuration scheme.

Table 13–17. Stratix II and Stratix II GX MSEL Pin Settings for PPA Configuration Schemes

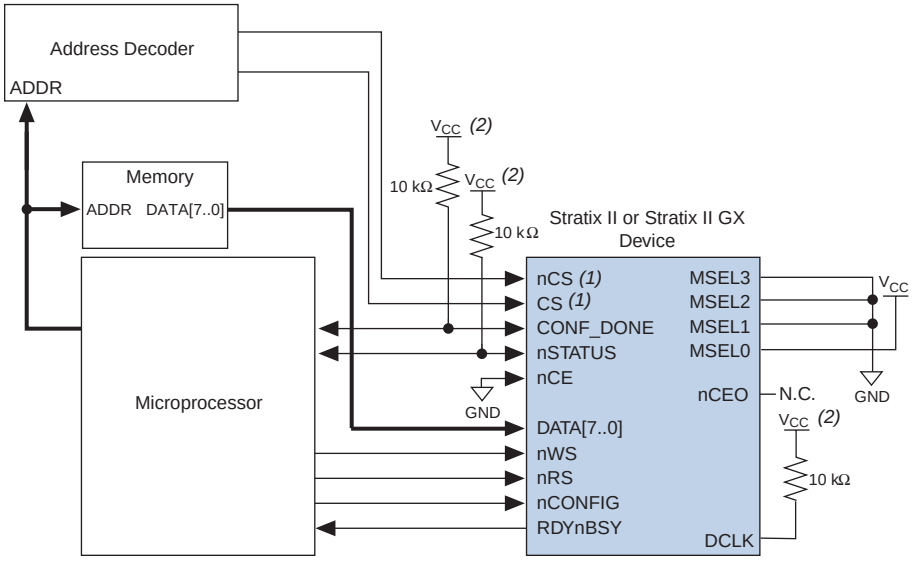
Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0
PPA	0	0	0	1
Remote System Upgrade PPA *(1)*	0	1	0	1

Note to Table 13–17:

(1) This scheme requires that you drive the RUnLU pin to specify either remote update or local update. For more information about remote system upgrades in Stratix II and Stratix II GX devices, refer to the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Figure 13–30 shows the configuration interface connections between the device and a microprocessor for single device PPA configuration. The microprocessor or an optional address decoder can control the device's chip select pins, nCS and CS. The address decoder allows the microprocessor to select the Stratix II or Stratix II GX device by accessing a particular address, which simplifies the configuration process. Hold the nCS and CS pins active during configuration and initialization.

Figure 13–30. Single Device PPA Configuration Using a Microprocessor



Notes to Figure 13–30:

(1) If not used, the CS pin can be connected to V_{CC} directly. If not used, the nCS pin can be connected to GND directly.

(2) The pull-up resistor should be connected to a supply that provides an acceptable input signal for the device. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

During PPA configuration, it is only required to use either the nCS or CS pin. Therefore, if you are using only one chip-select input, the other must be tied to the active state. For example, nCS can be tied to ground while CS is toggled to control configuration. The device's nCS or CS pins can be toggled during PPA configuration if the design meets the specifications set for t_{CSSU}, t_{WSP}, and t_{CSH} listed in Table 13–18.

Upon power-up, the Stratix II and Stratix II GX devices go through a POR. The POR delay is dependent on the PORSEL pin setting. When PORSEL is driven low, the POR time is approximately 100 ms. If PORSEL is driven high, the POR time is approximately 12 ms. During POR, the device will reset, hold nSTATUS low, and tri-state all user I/O pins. Once the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

You can hold nConfig low in order to stop device configuration.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook* and the *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*.

The configuration cycle consists of three stages: reset, configuration and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration, the microprocessor must generate a low-to-high transition on the nCONFIG pin.

To begin configuration, power the V_{CCINT}, V_{CCIO}, and V_{CCPD} voltages (for the banks where the configuration and JTAG pins reside) to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. Once nSTATUS is released the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the microprocessor should then assert the target device's nCS pin low and/or CS pin high. Next, the microprocessor places an 8-bit configuration word (one byte) on the target device's DATA[7..0] pins and pulses the nWS pin low.

On the rising edge of nWS, the target device latches in a byte of configuration data and drives its RDYnBSY signal low, which indicates it is processing the byte of configuration data. The microprocessor can then perform other system functions while the Stratix II or Stratix II GX device is processing the byte of configuration data.

During the time RDYnBSY is low, the Stratix II or Stratix II GX device internally processes the configuration data using its internal oscillator (typically 100 MHz). When the device is ready for the next byte of configuration data, it will drive RDYnBSY high. If the microprocessor senses a high signal when it polls RDYnBSY, the microprocessor sends the next byte of configuration data to the device.

Alternatively, the nRS signal can be strobed low, causing the RDYnBSY signal to appear on DATA7. Because RDYnBSY does not need to be monitored, this pin doesn't need to be connected to the microprocessor. Do not drive data onto the data bus while nRS is low because it will cause contention on the DATA7 pin. If you are not using the nRS pin to monitor configuration, it should be tied high.

To simplify configuration and save an I/O port, the microprocessor can wait for the total time of t_{BUSY} (max) + t_{RDY2WS} + t_{W2SB} before sending the next data byte. In this set-up, nRS should be tied high and RDYnBSY does not need to be connected to the microprocessor. The t_{BUSY}, t_{RDY2WS}, and t_{W2SB} timing specifications are listed in Table 13–18 on page 13–82.

Next, the microprocessor checks nSTATUS and CONF_DONE. If nSTATUS is not low and CONF_DONE is not high, the microprocessor sends the next data byte. However, if nSTATUS is not low and all the configuration data has been received, the device is ready for initialization. The CONF_DONE pin will go high one byte early in parallel configuration (FPP and PPA) modes. The last byte is required for serial configuration (AS and PS) modes. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The open-drain CONF_DONE pin is pulled high by an external 10-kΩ pull-up resistor. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

In Stratix II and Stratix II GX devices, the initialization clock source is either the internal oscillator (typically 10 MHz) or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If the internal oscillator is used, the Stratix II or Stratix II GX device provides itself with enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device.

You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. The **Enable user-supplied start-up clock** (**CLKUSR**) option can be turned on in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. Supplying a clock on CLKUSR does not affect the configuration process. After CONF_DONE goes high, CLKUSR is enabled after the time specified as t_{CD2CU}. After this time period elapses, the Stratix II and Stratix II GX devices require 299 clock cycles to initialize properly and enter user mode. Stratix II devices support a CLKUSR f_{MAX} of 100 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. This **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box. If the INIT_DONE pin is used it is high because of an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. Once the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the

INIT_DONE pin is released and pulled high. The microprocessor must be able to detect this low-to-high transition that signals the device has entered user mode. When initialization is complete, the device enters user mode. In user-mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

To ensure DATA[7..0] is not left floating at the end of configuration, the microprocessor must drive them either high or low, whichever is convenient on your board. After configuration, the nCS, CS, nRS, nWS, RDYnBSY, and DATA[7..0] pins can be used as user I/O pins. When choosing the PPA scheme in the Quartus II software as a default, these I/O pins are tri-stated in user mode and should be driven by the microprocessor. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device & Pin Options** dialog box.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the microprocessor that there is an error. If the **Auto-restart configuration after error** option-available in the Quartus II software from the **General** tab of the **Device & Pin Options** dialog box-is turned on, the device releases nSTATUS after a reset time-out period (maximum of 100 µs). After nSTATUS is released and pulled high by a pull-up resistor, the microprocessor can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the microprocessor must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

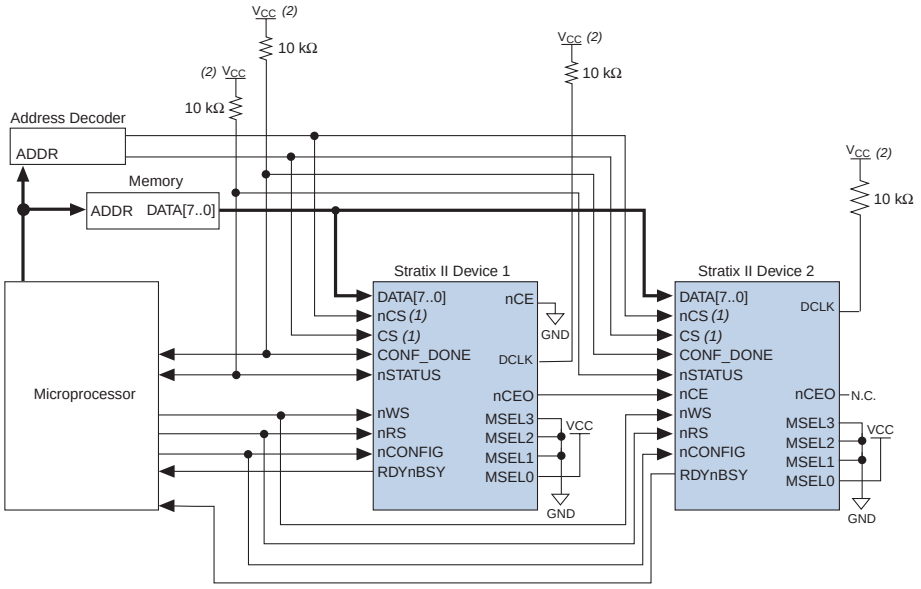
The microprocessor can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. To detect errors and determine when programming completes, monitor the CONF_DONE pin with the microprocessor. If the microprocessor sends all configuration data but CONF_DONE or INIT_DONE has not gone high, the microprocessor must reconfigure the target device.

If you are using the optional CLKUSR pin and nCONFIG is pulled low to restart configuration during device initialization, ensure CLKUSR continues toggling during the time nSTATUS is low (maximum of 100 µs).

When the device is in user-mode, a reconfiguration can be initiated by transitioning the nCONFIG pin low-to-high. The nCONFIG pin should go low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Once nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 13–31 shows how to configure multiple Stratix II or Stratix II GX devices using a microprocessor. This circuit is similar to the PPA configuration circuit for a single device, except the devices are cascaded for multi-device configuration.

Figure 13–31. Multi-Device PPA Configuration Using a Microprocessor



Notes to Figure 13–31:

(1) If not used, the CS pin can be connected to V_{CC} directly. If not used, the nCS pin can be connected to GND directly.

(2) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

In multi-device PPA configuration the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration within one clock cycle. Therefore, the transfer of data destinations is transparent to the microprocessor.

Each device's RDYnBSY pin can have a separate input to the microprocessor. Alternatively, if the microprocessor is pin limited, all the RDYnBSY pins can feed an AND gate and the output of the AND gate can feed the microprocessor. For example, if you have two devices in a PPA

configuration chain, the second device's RDYnBSY pin will be high during the time that the first device is being configured. When the first device has been successfully configured, it will drive nCEO low to activate the next device in the chain and drive its RDYnBSY pin high. Therefore, since RDYnBSY signal is driven high before configuration and after configuration before entering user-mode, the device being configured will govern the output of the AND gate.

The nRS signal can be used in multi-device PPA chain because the Stratix II and Stratix II GX devices tri-state the DATA[7..0] pins before configuration and after configuration before entering user-mode to avoid contention. Therefore, only the device that is currently being configured responds to the nRS strobe by asserting DATA7.

All other configuration pins (nCONFIG, nSTATUS, DATA[7..0], nCS, CS, nWS, nRS and CONF_DONE) are connected to every device in the chain. It is not necessary to tie nCS and CS together for every device in the chain, as each device's nCS and CS input can be driven by a separate source. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

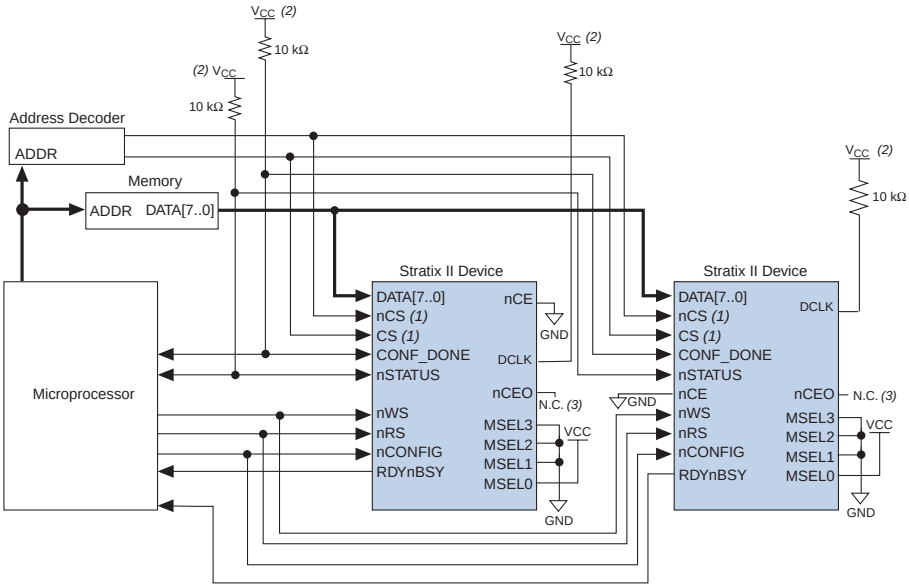
Since all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, configuration stops for the entire chain and the entire chain must be reconfigured. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (maximum of 100 µs). After all nSTATUS pins are released and pulled high, the microprocessor can try to reconfigure the chain without needing to pulse nCONFIG low. If this option is turned off, the microprocessor must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

In your system, you may have multiple devices that contain the same configuration data. To support this configuration scheme, all device nCE inputs are tied to GND, while nCEO pins are left floating. All other configuration pins (nCONFIG, nSTATUS, DATA[7..0], nCS, CS, nWS, nRS and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DATA lines are buffered for every fourth device. Devices must be the same density and package.

All devices start and complete configuration at the same time. Figure 13–32 shows multi-device PPA configuration when both devices are receiving the same configuration data.

Figure 13–32. Multiple-Device PPA Configuration Using a Microprocessor When Both devices Receive the Same Data



Notes to Figure 13–32:

(1) If not used, the CS pin can be connected to V_{CC} directly. If not used, the nCS pin can be connected to GND directly.

(2) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(3) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

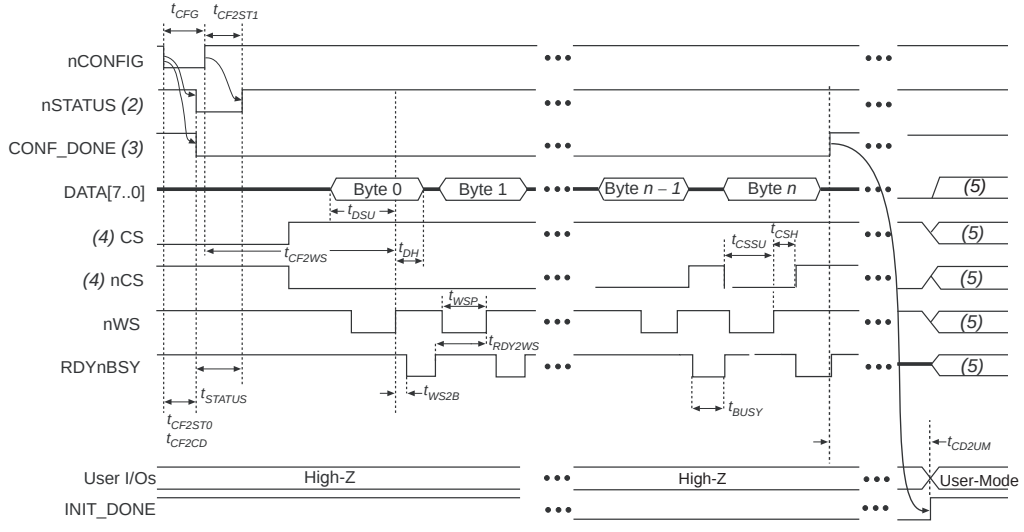
You can use a single configuration chain to configure Stratix II or Stratix II GX devices with other Altera devices that support PPA configuration, such as Stratix, Mercury™, APEX™ 20K, ACEX® 1K, and FLEX® 10KE devices. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device initiates reconfiguration in all devices, all of the device CONF_DONE and nSTATUS pins must be tied together.

For more information on configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

PPA Configuration Timing

Figure 13–33 shows the timing waveform for the PPA configuration scheme using a microprocessor.

Figure 13–33. Stratix II and Stratix II GX PPA Configuration Timing Waveform Using nWS *Note (1)*



Notes to Figure 13–33:

(1) The beginning of this waveform shows the device in user-mode. In user-mode, nCONFIG, nSTATUS and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(2) Upon power-up, Stratix II and Stratix II GX devices hold nSTATUS low for the time of the POR delay.

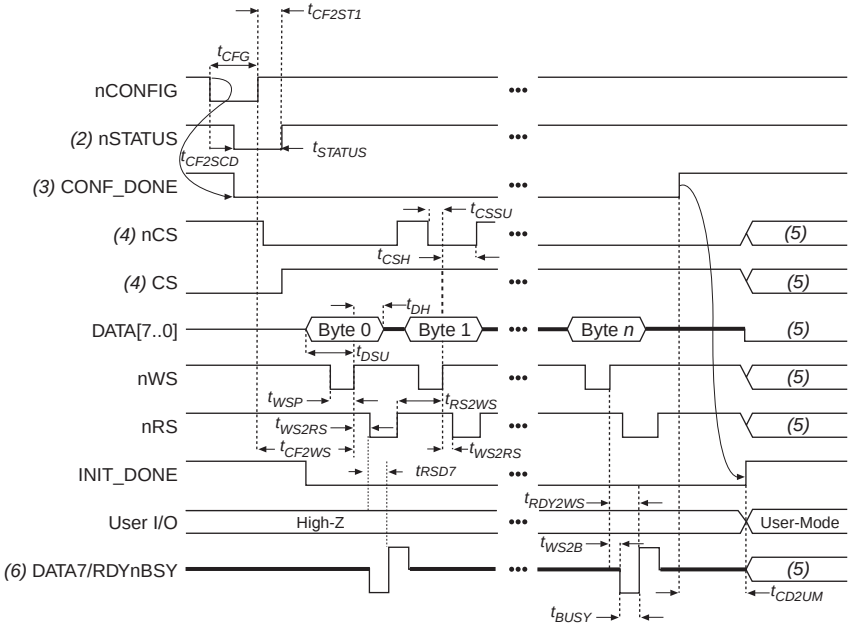
(3) Upon power-up, before and during configuration, CONF_DONE is low.

(4) The user can toggle nCS or CS during configuration if the design meets the specification for t_{CSSU}, t_{WSP}, and t_{CSH}.

(5) DATA[7..0], CS, nCS, nWS, nRS, and RDYnBSY are available as user I/O pins after configuration and the state of theses pins depends on the dual-purpose pin settings.

Figure 13–34 shows the timing waveform for the PPA configuration scheme when using a strobed nRS and nWS signal.

Figure 13–34. Stratix II and Stratix II GX PPA Configuration Timing Waveform Using nRS and nWS *Note (1)*



Notes to Figure 13–34:

(1) The beginning of this waveform shows the device in user-mode. In user-mode, nCONFIG, nSTATUS and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.
(2) Upon power-up, Stratix II and Stratix II GX devices hold nSTATUS low for the time of the POR delay.
(3) Upon power-up, before and during configuration, CONF_DONE is low.
(4) The user can toggle nCS or CS during configuration if the design meets the specification for t_{CSSU}, t_{WSP}, and t_{CSH}.
(5) DATA[7..0], CS, nCS, nWS, nRS, and RDYnBSY are available as user I/O pins after configuration and the state of theses pins depends on the dual-purpose pin settings.
(6) DATA7 is a bidirectional pin. It is an input for configuration data input, but it is an output to show the status of RDYnBSY.

Table 13–18 defines the timing parameters for Stratix II and Stratix II GX devices for PPA configuration.

Table 13–18. PPA Timing Parameters for Stratix II and Stratix II GX Devices (Part 1 of 2)

Symbol	Parameter	Min	Max	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low		800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low		800	ns

Table 13–18. PPA Timing Parameters for Stratix II and Stratix II GX Devices (Part 2 of 2)

Symbol	Parameter	Min	Max	Units
t_{CFG}	nCONFIG low pulse width	2		µs
t_{STATUS}	nSTATUS low pulse width	10	100 *(1)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high		100 *(1)*	µs
t_{CSSU}	Chip select setup time before rising edge on nWS	10		ns
t_{CSH}	Chip select hold time after rising edge on nWS	0		ns
t_{CF2WS}	nCONFIG high to first rising edge on nWS	100		µs
t_{ST2WS}	nSTATUS high to first rising edge on nWS	2		µs
t_{DSU}	Data setup time before rising edge on nWS	10		ns
t_{DH}	Data hold time after rising edge on nWS	0		ns
t_{WSP}	nWS low pulse width	15		ns
t_{WS2B}	nWS rising edge to RDYnBSY low		20	ns
t_{BUSY}	RDYnBSY low pulse width	7	45	ns
t_{RDY2WS}	RDYnBSY rising edge to nWS rising edge	15		ns
t_{WS2RS}	nWS rising edge to nRS falling edge	15		ns
t_{RS2WS}	nRS rising edge to nWS rising edge	15		ns
t_{RSD7}	nRS falling edge to DATA7 valid with RDYnBSY signal		20	ns
t_R	Input rise time		40	ns
t_F	Input fall time		40	ns
t_{CD2UM}	CONF_DONE high to user mode *(2)*	20	100	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	40		ns
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (299 × CLKUSR period)		

Notes to Table 13–18:

(1) This value is obtainable if users do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(2) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device.

Device configuration options and how to create configuration files are discussed further in the *Software Settings* chapter in volume 2 the *Configuration Handbook*.

JTAG Configuration

The JTAG has developed a specification for boundary-scan testing. This boundary-scan test (BST) architecture offers the capability to efficiently test components on PCBs with tight lead spacing. The BST architecture can test pin connections without using physical test probes and capture functional data while a device is operating normally. The JTAG circuitry can also be used to shift configuration data into the device. The Quartus II software automatically generates SOFs that can be used for JTAG configuration with a download cable in the Quartus II software programmer.

For more information on JTAG boundary-scan testing, refer to the following documents:

- *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Jam Programming Support - JTAG Technologies*

Stratix II and Stratix II GX devices are designed such that JTAG instructions have precedence over any device configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration of Stratix II or Stratix II GX devices during PS configuration, PS configuration is terminated and JTAG configuration begins.

You cannot use the Stratix II and Stratix II GX decompression or design security features if you are configuring your Stratix II or Stratix II GX device when using JTAG-based configuration.

A device operating in JTAG mode uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS`, and `TRST` pins have weak internal pull-up resistors (typically 25 kΩ). The `TDO` output pin is powered by V_{CCIO} in I/O bank 4. All of the JTAG input pins are powered by the 3.3-V V_{CCPD} pin. All user I/O pins are tri-stated during JTAG configuration. Table 13–19 explains each JTAG pin's function.

The `TDO` output is powered by the V_{CCIO} power supply of I/O bank 4.

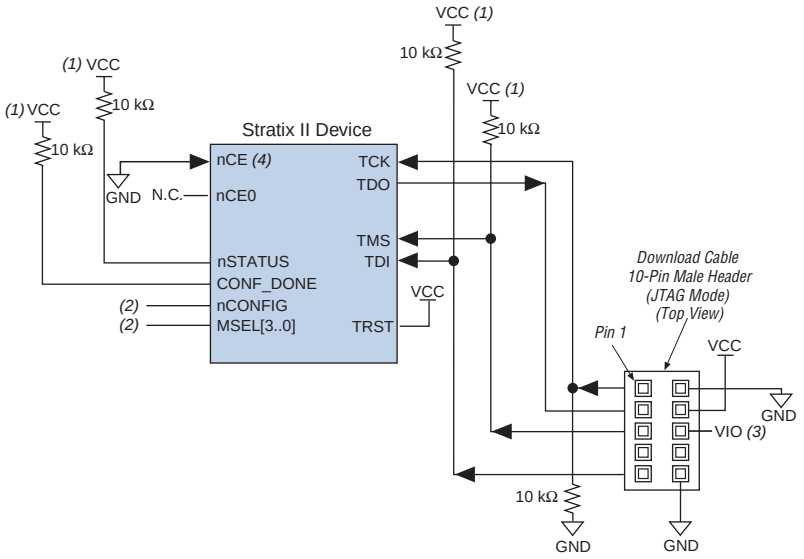
For recommendations on how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

Table 13–19. Dedicated JTAG Pins

Pin Name	Pin Type	Description
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to V_{CC}.
TDO	Test data output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the JTAG circuitry is not used, leave the TDO pin unconnected.
TMS	Test mode select	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions within the state machine occur on the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to V_{CC}.
TCK	Test clock input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to GND.
TRST	Test reset input (optional)	Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to IEEE Std. 1149.1. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to GND.

During JTAG configuration, data can be downloaded to the device on the PCB through the USB Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV download cable. Configuring devices through a cable is similar to programming devices in-system, except the TRST pin should be connected to V_{CC}. This ensures that the TAP controller is not reset. Figure 13–35 shows JTAG configuration of a single Stratix II or Stratix II GX device.

Figure 13–35. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 13–35:

(1) The pull-up resistor should be connected to the same supply voltage as the USB Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II, or ByteBlasterMV cable.

(2) The nCONFIG, MSEL[3..0] pins should be connected to support a non-JTAG configuration scheme. If only JTAG configuration is used, connect nCONFIG to V_{CC}, and MSEL[3..0] to ground. Pull DCLK either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value. In the ByteBlasterMV cable, this pin is a no connect. In the USB Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for active serial programming, otherwise it is a no connect.

(4) nCE must be connected to GND or driven low for successful JTAG configuration.

To configure a single device in a JTAG chain, the programming software places all other devices in bypass mode. In bypass mode, devices pass programming data from the TDI pin to the TDO pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the TDO pin one clock cycle later.

The Quartus II software verifies successful JTAG configuration upon completion. At the end of configuration, the software checks the state of CONF_DONE through the JTAG port. When Quartus II generates a (**.jam**) file for a multi-device chain, it contains instructions so that all the devices in the chain will be initialized at the same time. If CONF_DONE is not high, the Quartus II software indicates that configuration has failed. If

CONF_DONE is high, the software indicates that configuration was successful. After the configuration bit stream is transmitted serially via the JTAG TDI port, the TCK port is clocked an additional 299 cycles to perform device initialization.

Stratix II and Stratix II GX devices have dedicated JTAG pins that always function as JTAG pins. Not only can you perform JTAG testing on Stratix II and Stratix II GX devices before and after, but also during configuration. While other device families do not support JTAG testing during configuration, Stratix II and Stratix II GX devices support the bypass, idcode, and sample instructions during configuration without interrupting configuration. All other JTAG instructions may only be issued by first interrupting configuration and reprogramming I/O pins using the CONFIG_IO instruction.

The CONFIG_IO instruction allows I/O buffers to be configured via the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix II or Stratix II GX device or waiting for a configuration device to complete configuration. Once configuration has been interrupted and JTAG testing is complete, the part must be reconfigured via JTAG (PULSE_CONFIG instruction) or by pulsing nCONFIG low.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Stratix II and Stratix II GX devices do not affect JTAG boundary-scan or programming operations. Toggling these pins does not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration of Stratix II or Stratix II GX devices, consider the dedicated configuration pins. Table 13–20 shows how these pins should be connected during JTAG configuration.

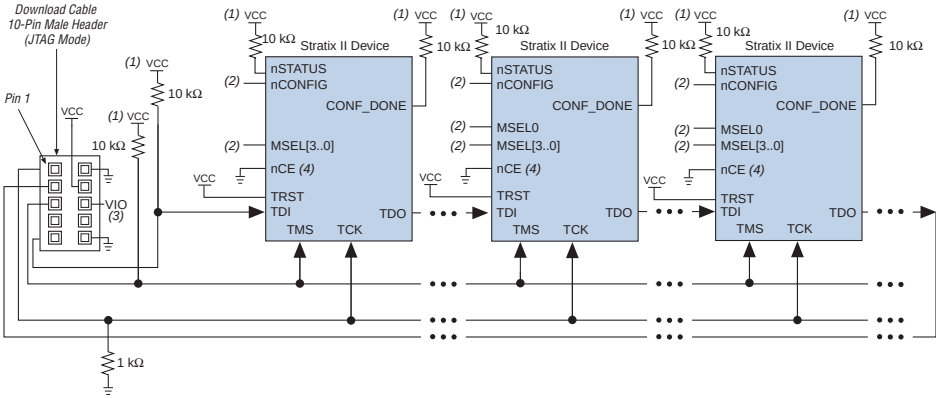
When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer.

Table 13–20. Dedicated Configuration Pin Connections During JTAG Configuration

Signal	Description
nCE	On all Stratix II or Stratix II GX devices in the chain, nCE should be driven low by connecting it to ground, pulling it low via a resistor, or driving it by some control circuitry. For devices that are also in multi-device FPP, AS, PS, or PPA configuration chains, the nCE pins should be connected to GND during JTAG configuration or JTAG configured in the same order as the configuration chain.
nCEO	On all Stratix II or Stratix II GX devices in the chain, nCEO can be left floating or connected to the nCE of the next device.
MSEL	These pins must not be left floating. These pins support whichever non-JTAG configuration is used in production. If only JTAG configuration is used, tie these pins to ground.
nCONFIG	Driven high by connecting to V_{CC}, pulling up via a resistor, or driven high by some control circuitry.
nSTATUS	Pull to V_{CC} via a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin should be pulled up to V_{CC} individually.
CONF_DONE	Pull to V_{CC} via a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin should be pulled up to V_{CC} individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration.
DCLK	Should not be left floating. Drive low or high, whichever is more convenient on your board.

JTAG-chain device programming is ideal when the system contains multiple devices, or when testing your system using JTAG BST circuitry. Figure 13–36 shows multi-device JTAG configuration.

Figure 13–36. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 13–36:

(1) The pull-up resistor should be connected to the same supply voltage as the USB Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II or ByteBlasterMV cable.

(2) The nCONFIG, MSEL[3..0] pins should be connected to support a non-JTAG configuration scheme. If only JTAG configuration is used, connect nCONFIG to V_{CC}, and MSEL[3..0] to ground. Pull DCLK either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value. In the ByteBlasterMV cable, this pin is a no connect. In the USB Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for active serial programming, otherwise it is a no connect.

(4) nCE must be connected to GND or driven low for successful JTAG configuration.

The nCE pin must be connected to GND or driven low during JTAG configuration. In multi-device FPP, AS, PS, and PPA configuration chains, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. In addition, the CONF_DONE and nSTATUS signals are all shared in multi-device FPP, AS, PS, or PPA configuration chains so the devices can enter user mode at the same time after configuration is complete. When the CONF_DONE and nSTATUS signals are shared among all the devices, every device must be configured when JTAG configuration is performed.

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 13–36, where each of the CONF_DONE and nSTATUS signals are isolated, so that each device can enter user mode individually.

After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, make sure the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the nCEO of the previous device will drive nCE of the next device low when it has successfully been JTAG configured.

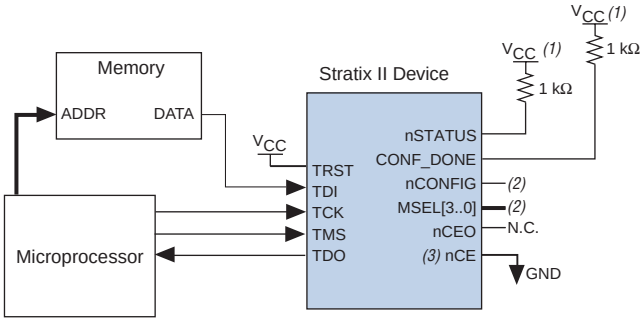
Other Altera devices that have JTAG support can be placed in the same JTAG chain for device programming and configuration.

Stratix, Stratix II, Stratix II GX, Cyclone™, and Cyclone II devices must be within the first 17 devices in a JTAG chain. All of these devices have the same JTAG controller. If any of the Stratix, Stratix II, Stratix II GX, Cyclone, and Cyclone II devices are in the 18th or after they will fail configuration. This does not affect SignalTap® II.

For more information on configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in the *Configuration Handbook*.

Figure 13–37 shows JTAG configuration of a Stratix II or Stratix II GX device with a microprocessor.

Figure 13–37. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 13–37:

(1) The pull-up resistor should be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} should be high enough to meet the V_{IH} specification of the I/O on the device.

(2) The `nCONFIG`, `MSEL[3..0]` pins should be connected to support a non-JTAG configuration scheme. If only JTAG configuration is used, connect `nCONFIG` to V_{CC}, and `MSEL[3..0]` to ground. Pull `DCLK` either high or low, whichever is convenient on your board.

(3) `nCE` must be connected to GND or driven low for successful JTAG configuration.

Jam STAPL

Jam STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface. Jam STAPL is a freely licensed open standard.

The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

For more information on JTAG and Jam STAPL in embedded environments, refer to *AN 122: Using Jam STAPL for ISP & ICR via an Embedded Processor*. To download the jam player, visit the Altera web site at www.altera.com.

Device Configuration Pins

The following tables describe the connections and functionality of all the configuration related pins on the Stratix II and Stratix II GX devices. Table 13–21 summarizes the Stratix II pin configuration.

Table 13–21. Stratix II Configuration Pin Summary (Part 1 of 2) *Note (1)*

Bank	Description	Input/Output	Dedicated	Powered By	Configuration Mode
3	PGM[2..0]	Output		*(2)*	PS, FPP, PPA, RU, LU
3	ASDO	Output		*(2)*	AS
3	nCSO	Output		*(2)*	AS
3	CRC_ERROR	Output		*(2)*	Optional, all modes
3	DATA0	Input		*(3)*	All modes except JTAG
3	DATA[7..1]	Input		*(3)*	FPP, PPA
3	DATA7	Bidirectional		*(2)*, *(3)*	PPA
3	RDYnBSY	Output		*(2)*	PPA
3	INIT_DONE	Output		Pull-up	Optional, all modes
3	nSTATUS	Bidirectional	Yes	Pull-up	All modes
3	nCE	Input	Yes	*(3)*	All modes
3	DCLK	Input	Yes	*(3)*	PS, FPP
		Output		*(2)*	AS
3	CONF_DONE	Bidirectional	Yes	Pull-up	All modes
8	TDI	Input	Yes	VCCPD	JTAG
8	TMS	Input	Yes	VCCPD	JTAG
8	TCK	Input	Yes	VCCPD	JTAG
8	TRST	Input	Yes	VCCPD	JTAG
8	nCONFIG	Input	Yes	*(3)*	All modes
8	VCCSEL	Input	Yes	VCCINT	All modes
8	CS	Input		*(3)*	PPA
8	CLKUSR	Input		*(3)*	Optional
8	nWS	Input		*(3)*	PPA
8	nRS	Input		*(3)*	PPA
8	RUnLU	Input		*(3)*	PS, FPP, PPA, RU, LU
8	nCS	Input		*(3)*	PPA
7	PORSEL	Input	Yes	VCCINT	All modes
7	nIO_PULLUP	Input	Yes	VCCINT	All modes

Table 13–21. Stratix II Configuration Pin Summary (Part 2 of 2) *Note (1)*

Bank	Description	Input/Output	Dedicated	Powered By	Configuration Mode
7	PLL_ENA	Input	Yes	*(3)*	Optional
7	nCEO	Output	Yes	*(2)*, *(4)*	All modes
4	MSEL[3..0]	Input	Yes	VCCINT	All modes
4	TDO	Output	Yes	*(2)*, *(4)*	JTAG

Notes to Table 13–21:

(1) Total number of pins is 41, total number of dedicated pins is 19.

(2) All outputs are powered by VCCIO except as noted.

(3) All inputs are powered by VCCIO or VCCPD, based on the VCCSEL setting, except as noted.

(4) An external pull-up resistor may be required for this configuration pin because of the multivolt I/O interface. Refer to the *Stratix II Architecture* chapter in volume 1 of the *Stratix II Device Handbook* for pull-up or level shifter recommendations for nCEO and TDO.

Figure 13–38 shows the I/O bank locations.

Figure 13–38. Stratix II I/O Bank Numbers

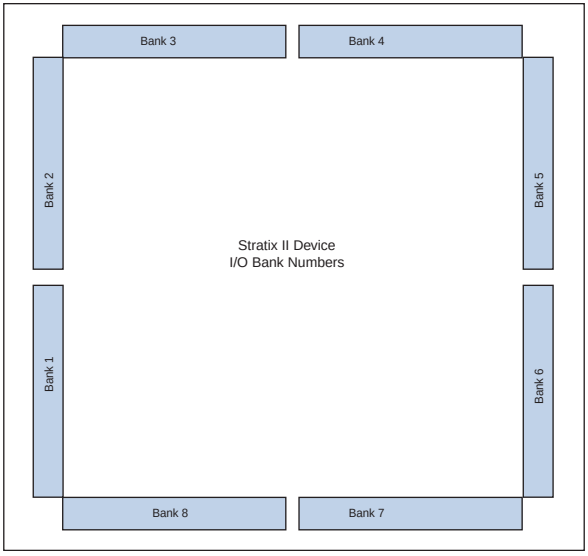


Table 13–22 describes the dedicated configuration pins, which are required to be connected properly on your board for successful configuration. Some of these pins may not be required for your configuration schemes.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 1 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
V_{CCPD}	N/A	All	Power	Dedicated power pin. This pin is used to power the I/O pre-drivers, the JTAG input pins, and the configuration input pins that are affected by the voltage level of VCCSEL. This pin must be connected to 3.3-V. V_{CCPD} must ramp-up from 0-V to 3.3-V within 100 ms. If V_{CCPD} is not ramped up within this specified time, your Stratix II or Stratix II GX device will not configure successfully. If your system does not allow for a V_{CCPD} ramp-up time of 100 ms or less, you must hold nCONFIG low until all power supplies are stable.
VCCSEL	N/A	All	Input	Dedicated input that selects which input buffer is used on the PLL_ENA pin and the configuration input pins; nCONFIG, DCLK (when used as an input), nSTATUS (when used as an input), CONF_DONE (when used as an input), DEV_OE, DEV_CLRn, DATA[7..0], RUnLU, nCE, nWS, nRS, CS, nCS, and CLKUSR. The 3.3-V/2.5-V input buffer is powered by V_{CCPD}, while the 1.8-V/1.5-V input buffer is powered by V_{CCIO}. The VCCSEL input buffer has an internal 5-kΩ pull-down resistor that is always active. The VCCSEL input buffer is powered by V_{CCINT} and must be hardwired to V_{CCPD} or ground. A logic high selects the 1.8-V/1.5-V input buffer, and a logic low selects the 3.3-V/2.5-V input buffer. For more information, refer to the "VCCSEL Pin" section.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 2 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
PORSEL	N/A	All	Input	Dedicated input which selects between a POR time of 12 ms or 100 ms. A logic high (1.5 V, 1.8 V, 2.5 V, 3.3 V) selects a POR time of about 12 ms and a logic low selects POR time of about 100 ms. The PORSEL input buffer is powered by V_{CCINT} and has an internal 5-kΩ pull-down resistor that is always active. The PORSEL pin should be tied directly to V_{CCPD} or GND.
nIO_PULLUP	N/A	All	Input	Dedicated input that chooses whether the internal pull-up resistors on the user I/O pins and dual-purpose I/O pins (nCSO, nASDO, DATA[7..0], nWS, nRS, RDYnBSY, nCS, CS, RUnLU, PGM[], CLKUSR, INIT_DONE, DEV_OE, DEV_CLR) are on or off before and during configuration. A logic high (1.5 V, 1.8 V, 2.5 V, 3.3 V) turns off the weak internal pull-up resistors, while a logic low turns them on. The nIO-PULLUP input buffer is powered by V_{CCPD} and has an internal 5-kΩ pull-down resistor that is always active. The nIO-PULLUP can be tied directly to V_{CCPD} or use a 1-kΩ pull-up resistor or tied directly to GND.
MSEL[3..0]	N/A	All	Input	4-bit configuration input that sets the Stratix II and Stratix II GX device configuration scheme. Refer to Table 13–1 for the appropriate connections. These pins must be hard-wired to V_{CCPD} or GND. The MSEL[3..0] pins have internal 5-kΩ pull-down resistors that are always active.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 3 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nCONFIG	N/A	All	Input	Configuration control input. Pulling this pin low during user-mode will cause the device to lose its configuration data, enter a reset state, tri-state all I/O pins. Returning this pin to a logic high level will initiate a reconfiguration. If your configuration scheme uses an enhanced configuration device or EPC2 device, nCONFIG can be tied directly to V_{CC} or to the configuration device's nINIT_CONF pin.
nSTATUS	N/A	All	Bidirectional open-drain	The device drives nSTATUS low immediately after power-up and releases it after the POR time. Status output. If an error occurs during configuration, nSTATUS is pulled low by the target device. Status input. If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. Driving nSTATUS low after configuration and initialization does not affect the configured device. If a configuration device is used, driving nSTATUS low will cause the configuration device to attempt to configure the device, but since the device ignores transitions on nSTATUS in user-mode, the device does not reconfigure. To initiate a reconfiguration, nCONFIG must be pulled low. The enhanced configuration devices' and EPC2 devices' OE and nCS pins have optional internal programmable pull-up resistors. If internal pull-up resistors on the enhanced configuration device are used, external 10-kΩ pull-up resistors should not be used on these pins. When using EPC2 devices, only external 10-kΩ pull-up resistors should be used.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 4 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nSTATUS (continued)				If VCCPD and VCCIO are not fully powered up, the following could occur: • VCCPD and VCCIO are powered high enough for the nSTATUS buffer to function properly, and nSTATUS is driven low. When VCCPD and VCCIO are ramped up, POR trips, and nSTATUS is released after POR expires. • VCCPD and VCCIO are not powered high enough for the nSTATUS buffer to function properly. In this situation, nSTATUS might appear logic high, triggering a configuration attempt that would fail because POR did not yet trip. When VCCPD and VCCIO are powered up, nSTATUS is pulled low because POR did not yet trip. When POR trips after VCCPD and VCCIO are powered up, nSTATUS is released and pulled high. At that point, reconfiguration is triggered and the device is configured.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 5 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
CONF_DONE	N/A	All	Bidirectional open-drain	Status output. The target device drives the CONF_DONE pin low before and during configuration. Once all configuration data is received without error and the initialization cycle starts, the target device releases CONF_DONE. Status input. After all data is received and CONF_DONE goes high, the target device initializes and enters user mode. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize. Driving CONF_DONE low after configuration and initialization does not affect the configured device. The enhanced configuration devices' and EPC2 devices' OE and nCS pins have optional internal programmable pull-up resistors. If internal pull-up resistors on the enhanced configuration device are used, external 10-kΩ pull-up resistors should not be used on these pins. When using EPC2 devices, only external 10-kΩ pull-up resistors should be used.
nCE	N/A	All	Input	Active-low chip enable. The nCE pin activates the device with a low signal to allow configuration. The nCE pin must be held low during configuration, initialization, and user mode. In single device configuration, it should be tied low. In multi-device configuration, nCE of the first device is tied low while its nCEO pin is connected to nCE of the next device in the chain. The nCE pin must also be held low for successful JTAG programming of the device.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 6 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nCEO	N/A	All	Output	Output that drives low when device configuration is complete. In single device configuration, this pin is left floating. In multi-device configuration, this pin feeds the next device's nCE pin. The nCEO of the last device in the chain is left floating. The nCEO pin is powered by V_{CCIO} in I/O bank 7. For recommendations on how to connect nCEO in a chain with multiple voltages across the devices in the chain, refer to the *Stratix II Architecture* chapter in volume 1 of the *Stratix II Handbook* or the *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*.
ASDO	N/A in AS mode I/O in non-AS mode	AS	Output	Control signal from the Stratix II or Stratix II GX device to the serial configuration device in AS mode used to read out configuration data. In AS mode, ASDO has an internal pull-up resistor that is always active.
nCSO	N/A in AS mode I/O in non-AS mode	AS	Output	Output control signal from the Stratix II or Stratix II GX device to the serial configuration device in AS mode that enables the configuration device. In AS mode, nCSO has an internal pull-up resistor that is always active.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 7 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DCLK	N/A	Synchronous configuration schemes (PS, FPP, AS)	Input (PS, FPP) Output (AS)	In PS and FPP configuration, DCLK is the clock input used to clock data from an external source into the target device. Data is latched into the device on the rising edge of DCLK. In AS mode, DCLK is an output from the Stratix II or Stratix II GX device that provides timing for the configuration interface. In AS mode, DCLK has an internal pull-up resistor (typically 25 kΩ) that is always active. In PPA mode, DCLK should be tied high to V_{CC} to prevent this pin from floating. After configuration, this pin is tri-stated. In schemes that use a configuration device, DCLK will be driven low after configuration is done. In schemes that use a control host, DCLK should be driven either high or low, whichever is more convenient. Toggling this pin after configuration does not affect the configured device.
DATA0	I/O	PS, FPP, PPA, AS	Input	Data input. In serial configuration modes, bit-wide configuration data is presented to the target device on the DATA0 pin. The V_{IH} and V_{IL} levels for this pin are dependent on the input buffer selected by the VCCSEL pin. Refer to the section "VCCSEL Pin" on page 13–10 for more information. In AS mode, DATA0 has an internal pull-up resistor that is always active. After configuration, DATA0 is available as a user I/O pin and the state of this pin depends on the **Dual-Purpose Pin** settings. After configuration, EPC1 and EPC1441 devices tri-state this pin, while enhanced configuration and EPC2 devices drive this pin high.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 8 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DATA[7..1]	I/O	Parallel configuration schemes (FPP and PPA)	Inputs	Data inputs. Byte-wide configuration data is presented to the target device on DATA[7..0]. The V_{IH} and V_{IL} levels for this pin are dependent on the input buffer selected by the VCCSEL pin. Refer to the section "VCCSEL Pin" on page 13–10 for more information. In serial configuration schemes, they function as user I/O pins during configuration, which means they are tri-stated. After PPA or FPP configuration, DATA[7..1] are available as user I/O pins and the state of these pin depends on the **Dual-Purpose Pin** settings.
DATA7	I/O	PPA	Bidirectional	In the PPA configuration scheme, the DATA7 pin presents the RDYnBSY signal after the nRS signal has been strobed low. The V_{IH} and V_{IL} levels for this pin are dependent on the input buffer selected by the VCCSEL pin. Refer to the section "VCCSEL Pin" on page 13–10 for more information. In serial configuration schemes, it functions as a user I/O pin during configuration, which means it is tri-stated. After PPA configuration, DATA7 is available as a user I/O and the state of this pin depends on the **Dual-Purpose Pin** settings.
nWS	I/O	PPA	Input	Write strobe input. A low-to-high transition causes the device to latch a byte of data on the DATA[7..0] pins. In non-PPA schemes, it functions as a user I/O pin during configuration, which means it is tri-stated. After PPA configuration, nWS is available as a user I/O pins and the state of this pin depends on the **Dual-Purpose Pin** settings.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 9 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nRS	I/O	PPA	Input	Read strobe input. A low input directs the device to drive the RDYnBSY signal on the DATA7 pin. If the nRS pin is not used in PPA mode, it should be tied high. In non-PPA schemes, it functions as a user I/O during configuration, which means it is tri-stated. After PPA configuration, nRS is available as a user I/O pin and the state of this pin depends on the **Dual-Purpose Pin** settings.
RDYnBSY	I/O	PPA	Output	Ready output. A high output indicates that the target device is ready to accept another data byte. A low output indicates that the target device is busy and not ready to receive another data byte. In PPA configuration schemes, this pin will drive out high after power-up, before configuration and after configuration before entering user-mode. In non-PPA schemes, it functions as a user I/O pin during configuration, which means it is tri-stated. After PPA configuration, RDYnBSY is available as a user I/O pin and the state of this pin depends on the **Dual-Purpose Pin** settings.

Table 13–22. Dedicated Configuration Pins on the Stratix II and Stratix II GX Device (Part 10 of 10)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nCS/CS	I/O	PPA	Input	Chip-select inputs. A low on nCS and a high on CS select the target device for configuration. The nCS and CS pins must be held active during configuration and initialization. During the PPA configuration mode, it is only required to use either the nCS or CS pin. Therefore, if only one chip-select input is used, the other must be tied to the active state. For example, nCS can be tied to ground while CS is toggled to control configuration. In non-PPA schemes, it functions as a user I/O pin during configuration, which means it is tri-stated. After PPA configuration, nCS and CS are available as user I/O pins and the state of these pins depends on the **Dual-Purpose Pin** settings.
RUnLU	N/A if using Remote System Upgrade I/O if not	Remote System Upgrade in FPP, PS or PPA	Input	Input that selects between remote update and local update. A logic high (1.5-V, 1.8-V, 2.5-V, 3.3-V) selects remote update and a logic low selects local update. When not using remote update or local update configuration modes, this pin is available as general-purpose user I/O pin. When using remote system upgrade in AS more, the RUnLU pin is available as a general-purpose I/O pin.
PGM[2..0]	N/A if using Remote System Upgrade I/O if not using	Remote System Upgrade in FPP, PS or PPA	Output	These output pins select one of eight pages in the memory (either flash or enhanced configuration device) when using a remote system upgrade mode. When not using remote update or local update configuration modes, these pins are available as general-purpose user I/O pins. When using remote system upgrade in AS more, the PGM[] pins are available as general-purpose I/O pins.

Table 13–23 describes the optional configuration pins. If these optional configuration pins are not enabled in the Quartus II software, they are available as general-purpose user I/O pins. Therefore, during configuration, these pins function as user I/O pins and are tri-stated with weak pull-up resistors.

Table 13–23. Optional Configuration Pins

Pin Name	User Mode	Pin Type	Description
CLKUSR	N/A if option is on. I/O if option is off.	Input	Optional user-supplied clock input synchronizes the initialization of one or more devices. This pin is enabled by turning on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software.
INIT_DONE	N/A if option is on. I/O if option is off.	Output open-drain	Status pin can be used to indicate when the device has initialized and is in user mode. When nCONFIG is low and during the beginning of configuration, the INIT_DONE pin is tri-stated and pulled high due to an external 10-kΩ pull-up resistor. Once the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin will go low. When initialization is complete, the INIT_DONE pin will be released and pulled high and the device enters user mode. Thus, the monitoring circuitry must be able to detect a low-to-high transition. This pin is enabled by turning on the **Enable INIT_DONE output** option in the Quartus II software.
DEV_OE	N/A if option is on. I/O if option is off.	Input	Optional pin that allows the user to override all tri-states on the device. When this pin is driven low, all I/O pins are tri-stated; when this pin is driven high, all I/O pins behave as programmed. This pin is enabled by turning on the **Enable device-wide output enable** (**DEV_OE**) option in the Quartus II software.
DEV_CLRn	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all clears on all device registers. When this pin is driven low, all registers are cleared; when this pin is driven high, all registers behave as programmed. This pin is enabled by turning on the **Enable device-wide reset** (**DEV_CLRn**) option in the Quartus II software.

Table 13–24 describes the dedicated JTAG pins. JTAG pins must be kept stable before and during configuration to prevent accidental loading of JTAG instructions. The TDI, TMS, and TRST have weak internal pull-up resistors (typically 25 kΩ) while TCK has a weak internal pull-down resistor. If you plan to use the SignalTap embedded logic array analyzer, you need to connect the JTAG pins of the Stratix II or Stratix II GX device to a JTAG header on your board.

Table 13–24. Dedicated JTAG Pins (Part 1 of 2)

Pin Name	User Mode	Pin Type	Description
TDI	N/A	Input	Serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK. The TDI pin is powered by the 3.3-V V_{CCPD} supply. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to V_{CC}.
TDO	N/A	Output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. The TDO pin is powered by V_{CCIO} in I/O bank 4. For recommendations on connecting a JTAG chain with multiple voltages across the devices in the chain, refer to the *IEEE 1149.1 (JTAG) Boundary Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook* or the *IEEE 1149.1 (JTAG) Boundary Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*. If the JTAG circuitry is not used, leave the TDO pin unconnected.
TMS	N/A	Input	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions within the state machine occur on the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. The TMS pin is powered by the 3.3-V V_{CCPD} supply. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to V_{CC}.

Table 13–24. Dedicated JTAG Pins (Part 2 of 2)

Pin Name	User Mode	Pin Type	Description
TCK	N/A	Input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. The TCK pin is powered by the 3.3-V V_{CCPD} supply. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting TCK to GND.
TRST	N/A	Input	Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to IEEE Std. 1149.1. The TRST pin is powered by the 3.3-V V_{CCPD} supply. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting the TRST pin to GND.

Conclusion

Stratix II and Stratix II GX devices can be configured in a number of different schemes to fit your system's need. In addition, configuration bitstream encryption, configuration data decompression, and remote system upgrade support supplement the Stratix II and Stratix II GX configuration solution.

Referenced Documents

This chapter references the following documents:

- *AN 122: Using Jam STAPL for ISP & ICR via an Embedded Processor*
- *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming*
- *ByteBlaster II Download Cable User Guide*
- *ByteBlasterMV Download Cable User Guide*
- *Configuration Devices for SRAM-Based LUT Devices Data Sheet* chapter in volume 2 of the *Configuration Handbook*.
- *Configuring Mixed Altera FPGA Chains* in volume 2 of the *Configuration Handbook*
- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II Device Handbook*
- *DC & Switching Characteristics* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* in volume 2 of the *Configuration Handbook*
- *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook*
- *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*
- *Jam Programming Support - JTAG Technologies*
- *MasterBlaster Serial/USB Communications Cable User Guide*

- *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook*
- *Remote System Upgrades With Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.
- *USB-Blaster Download Cable User Guide*
- *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.
- *Software Settings* in volume 2 of the *Configuration Handbook*
- *Stratix II GX Architecture* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *Stratix II Device Handbook*
- *Stratix II GX Device Handbook*

Document Revision History

Table 13–25 shows the revision history for this chapter.

Table 13–25. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v4.5	Updated handnote in "Configuration Data Decompression" section.	—
	Updated Notes in: • Table 13–3 • Table 13–10 • Table 13–12 • Table 13–15 • Table 13–18	—
	Updated TDO row for Tables 13–19 and 13–24.	—
	Added the "Referenced Documents" section.	—
	Minor text edits.	—
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 12. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—

Table 13–25. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
May 2007, v4.4	• Updated "Power-On Reset Circuit" section • Updated "VCCSEL Pin" section • Updated "Configuration Data Decompression" section • Updated "Active Serial Configuration (Serial Configuration Devices)" section • Updated "Output Configuration Pins" section	—
	Added notes to "FPP Configuration Using a MAX II Device as an External Host" and "Passive Parallel Asynchronous Configuration" section.	—
	• Updated Table 13–5 • Updated Table 13–6 • Updated Table 13–8 • Updated Table 13–11	—
	Removed Table 7-7.	—
	Added new "Output Configuration Pins" section.	—
	Corrected typo in "Configuration Devices" section.	—
	Corrected `CONF_DONE` in Table 13–22.	—
February 2007 v4.3	Added the "Document Revision History" section to this chapter.	—
April 2006, v4.2	Chapter updated as part of the *Stratix II Device Handbook* update.	—
No change	Formerly chapter 11. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
December 2005, v4.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—
October 2005 v4.0	Added chapter to the *Stratix II GX Device Handbook*.	—

14. Remote System Upgrades with Stratix II & Stratix II GX Devices

SII52008-4.5

Introduction

System designers today face difficult challenges such as shortened design cycles, evolving standards, and system deployments in remote locations. Stratix® II and Stratix II GX FPGAs help overcome these challenges with their inherent re-programmability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduce time-to-market, and extend product life.

Stratix II and Stratix II GX FPGAs feature dedicated remote system upgrade circuitry. Soft logic (either the Nios® embedded processor or user logic) implemented in a Stratix II or Stratix II GX device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information. This dedicated remote system upgrade circuitry is unique to Stratix, Stratix II, and Stratix II GX FPGAs and helps to avoid system downtime.

Remote system upgrade is supported in all Stratix II and Stratix II GX configuration schemes: fast passive parallel (FPP), active serial (AS), passive serial (PS), and passive parallel asynchronous (PPA). Remote system upgrade can also be implemented in conjunction with advanced Stratix II and Stratix II GX features such as real-time decompression of configuration data and design security using the advanced encryption standard (AES) for secure and efficient field upgrades.

This chapter describes the functionality and implementation of the dedicated remote system upgrade circuitry. It also defines several concepts related to remote system upgrade, including factory configuration, application configuration, remote update mode, local update mode, the user watchdog timer, and page mode operation. Additionally, this chapter provides design guidelines for implementing remote system upgrade with the various supported configuration schemes.

Functional Description

The dedicated remote system upgrade circuitry in Stratix II and Stratix II GX FPGAs manages remote configuration and provides error detection, recovery, and status information. User logic or a Nios processor implemented in the FPGA logic array provides access to the remote configuration data source and an interface to the system's configuration memory.

Stratix II and Stratix II GX FPGA's remote system upgrade process involves the following steps:

1. A Nios processor (or user logic) implemented in the FPGA logic array receives new configuration data from a remote location. The connection to the remote source is a communication protocol such as the transmission control protocol/Internet protocol (TCP/IP), peripheral component interconnect (PCI), user datagram protocol (UDP), universal asynchronous receiver/transmitter (UART), or a proprietary interface.

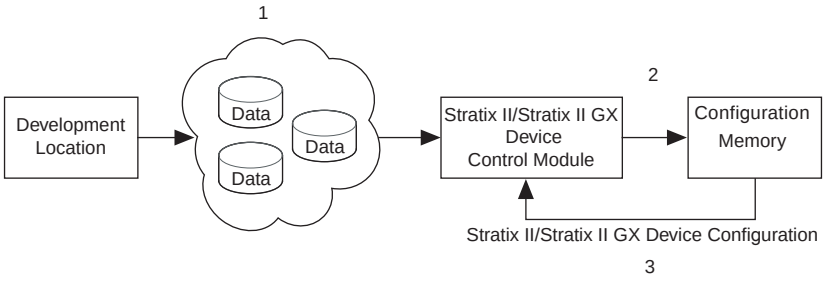
2. The Nios processor (or user logic) stores this new configuration data in non-volatile configuration memory. The non-volatile configuration memory can be any standard flash memory used in conjunction with an intelligent host (for example, a MAX® device or microprocessor), the serial configuration device, or the enhanced configuration device.

3. The Nios processor (or user logic) initiates a reconfiguration cycle with the new or updated configuration data.

4. The dedicated remote system upgrade circuitry detects and recovers from any error(s) that might occur during or after the reconfiguration cycle, and provides error status information to the user design.

Figure 14–1 shows the steps required for performing remote configuration updates. (The numbers in the figure below coincide with the steps above.)

Figure 14–1. Functional Diagram of Stratix II or Stratix II GX Remote System Upgrade



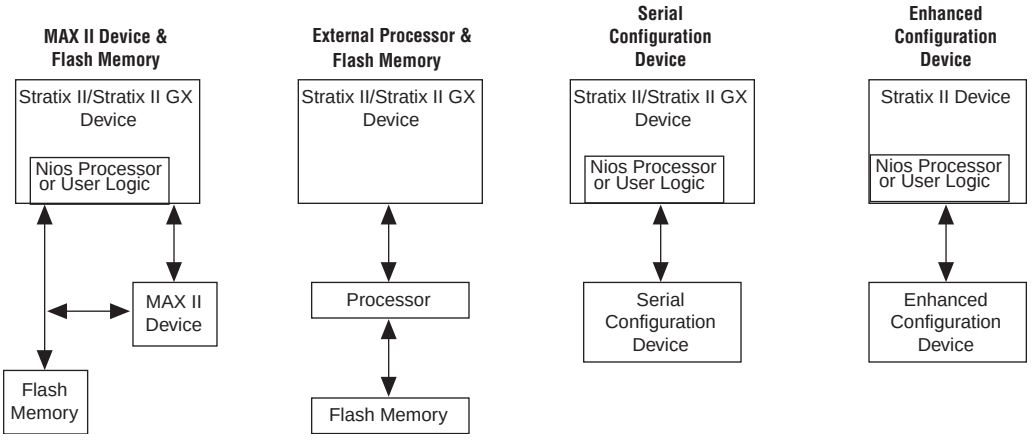
Stratix II and Stratix II GX FPGAs support remote system upgrade in the FPP, AS, PS, and PPA configuration schemes.

- Serial configuration devices use the AS scheme to configure Stratix II and Stratix II GX FPGAs.
- A MAX II device (or microprocessor and flash configuration schemes) uses FPP, PS, or PPA schemes to configure Stratix II and Stratix II GX FPGAs.
- Enhanced configuration devices use the FPP or PS configuration schemes to configure Stratix II and Stratix II GX FPGAs.

The JTAG-based configuration scheme does not support remote system upgrade.

Figure 14–2 shows the block diagrams for implementing remote system upgrade with the various Stratix II and Stratix II GX configuration schemes.

Figure 14–2. Remote System Upgrade Block Diagrams for Various Stratix II and Stratix II GS Configuration Schemes



For the active serial configuration scheme, the remote system upgrade only supports single device configurations.

You must set the mode select pins (`MSEL[3..0]`) and the `RUnLU` pin to select the configuration scheme and remote system upgrade mode best suited for your system. Table 14–1 lists the pin settings for Stratix II and Stratix II GX FPGAs. Standard configuration mode refers to normal FPGA configuration mode with no support for remote system upgrades, and the remote system upgrade circuitry is disabled. The following sections describe the local update and remote update remote system upgrade modes.

For more information on standard configuration schemes supported in Stratix II and Stratix II GX FPGAs, see the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook* and the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*.

Table 14–1. Stratix II and Stratix II GX Remote System Upgrade Modes

Configuration Scheme	MSEL[3..0]	RUnLU	Remote System Upgrade Mode
FPP	0000	-	Standard
	0100 *(1)*	0	Local update
	0100 *(1)*	1	Remote update
FPP with decompression and/or design security feature enabled *(2)*	1011	-	Standard
	1100 *(1)*	0	Local update
	1100 *(1)*	1	Remote update
Fast AS (40 MHz) *(3)*	1000	-	Standard
	1001	1	Remote update
AS (20 MHz) *(3)*	1101	-	Standard
	1110	1	Remote update
PS	0010	-	Standard
	0110 *(1)*	0	Local update
	0110 *(1)*	1	Remote update
PPA	0001	-	Standard
	0101 *(1)*	0	Local update
	0101 *(1)*	1	Remote update

Notes to Table 14–1:

(1) These schemes require that you drive the RUnLU pin to specify either remote update or local update mode. AS schemes only support the remote update mode.

(2) These modes are only supported when using a MAX II device or microprocessor and flash for configuration. In these modes, the host system must output a DCLK that is 4 x the data rate.

(3) The EPCS16 and EPCS64 serial configuration devices support a DCLK up to 40 MHz; other EPCS devices support a DCLK up to 20 MHz. See the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook* for more information.

Configuration Image Types and Pages

When using remote system upgrade, FPGA configuration bitstreams are classified as factory configuration images or application configuration images. An image, also referred to as a configuration, is a design loaded into the FPGA that performs certain user-defined functions. Each FPGA in your system requires one factory image and one or more application

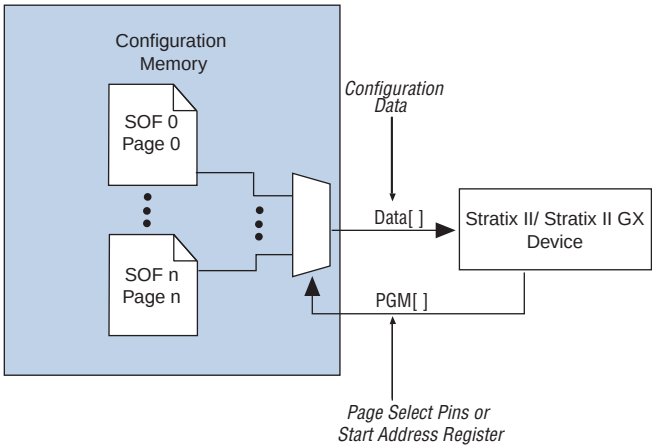
images. The factory image is a user-defined fall-back, or safe, configuration and is responsible for administering remote updates in conjunction with the dedicated circuitry. Application images implement user-defined functionality in the target FPGA.

A remote system update involves storing a new application configuration image or updating an existing one via the remote communication interface. After an application configuration image is stored or updated remotely, the user design in the FPGA initiates a reconfiguration cycle with the new image. Any errors during or after this cycle are detected by the dedicated remote system upgrade circuitry and cause the FPGA to automatically revert to the factory image. The factory image then performs error processing and recovery. While error processing functionality is limited to the factory configuration, both factory and application configurations can download and store remote updates and initiate system reconfiguration.

Stratix II and Stratix II GX FPGAs select between the different configuration images stored in the system configuration memory using the page address pins or start address registers. A page is a section of the configuration memory space that contains one configuration image for each FPGA in the system. One page stores one system configuration, regardless of the number of FPGAs in the system.

Page address pins select the configuration image within an enhanced configuration device or flash memory (MAX II device or microprocessor setup). Page start address registers are used when Stratix II and Stratix II GX FPGAs are configured in AS mode with serial configuration devices. Figure 14–3 illustrates page mode operation in Stratix II and Stratix II GX FPGAs.

Figure 14–3. Page Mode Operation in Stratix II & Stratix II GX FPGAs



Stratix II and Stratix II GX devices drive out three page address pins, PGM[2..0], to the MAX II device or microprocessor or enhanced configuration device. These page pins select between eight configuration pages. Page zero (PGM[2..0] = 000) must contain the factory configuration, and the other seven pages are application configurations. The PGM[] pins are pointers to the start address and length of each page, and the MAX II device, microprocessor, and enhanced configuration devices perform this translation.

When implementing remote system upgrade with an intelligent-host-based configuration, your MAX II device or microprocessor should emulate the page mode feature supported by the enhanced configuration device, which translates PGM pointers to a memory address in the configuration memory. Your MAX II device or microprocessor must provide a similar translation feature.

For more information about the enhanced configuration device page mode feature, refer to the Dynamic Configuration (Page Mode) Implementation section of the *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

When implementing remote system upgrade with AS configuration, a dedicated 7-bit page start address register inside Stratix II and Stratix II GX FPGAs determines the start addresses for configuration pages within the serial configuration device. The PGM[6..0] registers form bits [22..16] of the 24-bit start address while the other 17 bits are

set to zero: StAdd[23..0] = {1'b0, PGM[6..0], 16'b0}. During AS configuration, Stratix II and Stratix II GX FPGAs use this 24-bit page start address to obtain configuration data from the serial configuration devices.

Remote System Upgrade Modes

Remote system upgrade has two modes of operation: remote update mode and local update mode. The remote and local update modes allow you to determine the functionality of your system upon power up and offer different features. The RUnLU input pin selects between the remote update (logic high) and local update (logic low) modes.

Overview

In remote update mode, Stratix II and Stratix II GX FPGAs load the factory configuration image upon power up. The user-defined factory configuration should determine which application configuration is to be loaded and trigger a reconfiguration cycle. Remote update mode allows up to eight configuration images (one factory plus seven application images) when used with the MAX II device or microprocessor and flash-based configuration or an enhanced configuration device.

When used with serial configuration devices, the remote update mode allows an application configuration to start at any flash sector boundary. This translates to a maximum of 128 pages in the EPCS64 and 32 pages in the EPCS16 device, where the minimum size of each page is 512 KBits. Additionally, the remote update mode features a user watchdog timer that can detect functional errors in an application configuration.

Local update mode is a simplified version of the remote update mode. In this mode, Stratix II and Stratix II GX FPGAs directly load the application configuration, bypassing the factory configuration. This mode is useful if your system is required to boot into user mode with minimal startup time. It is also useful during system prototyping, as it allows you to verify functionality of the application configuration.

In local update mode, a maximum of two configuration images or pages is supported: one factory configuration, located at page address PGM[2..0] = 000, and one application configuration, located at page address PGM[2..0] = 001. Because the page address of the application configuration is fixed, the local update mode does not require the factory configuration image to determine which application is to be loaded. If any errors are encountered while loading the application configuration, Stratix II and Stratix II GX FPGAs revert to the factory configuration. The user watchdog timer feature is not supported in this mode.

Also, local update mode does not support AS configuration with the serial configuration devices because these devices don't support a dynamic pointer to page 001 start address location.

Table 14–2 details the differences between remote and local update modes.

Table 14–2. Differences Between Remote and Local Update Modes

Features	Remote Update Mode	Local Update Mode
RUnLU input pin setting	1	0
Page selection upon power up	PGM[2..0] = 000 (Factory)	PGM[2..0] = 001 (Application)
Supported configurations	MAX II device or microprocessor-based configuration, serial configuration, and enhanced configuration devices (FPP, PS, AS, PPA)	MAX II device or microprocessor-based configuration and enhanced configuration devices (FPP, PS, PPA)
Number of pages supported	Eight pages for external host or controller based configuration; up to 128 pages (512 KBits/page) for serial configuration device	Two pages
User watchdog timer	Available	Disabled
Remote system upgrade control and status register	Read/write access allowed in factory configuration. Read access in application configuration	Only status register read access allowed in local update mode (factory and application configurations). Write access to control register is disabled

Remote Update Mode

When Stratix II and Stratix II GX FPGAs are first powered up in remote update mode, it loads the factory configuration located at page zero (page address pins PGM[2..0] = "000"; page registers PGM[6..0] = "0000000"). You should always store the factory configuration image for your system at page address zero. A factory configuration image is a bitstream for the FPGA(s) in your system that is programmed during production and is the fall-back image when errors occur. This image is stored in non-volatile memory and is never updated or modified using

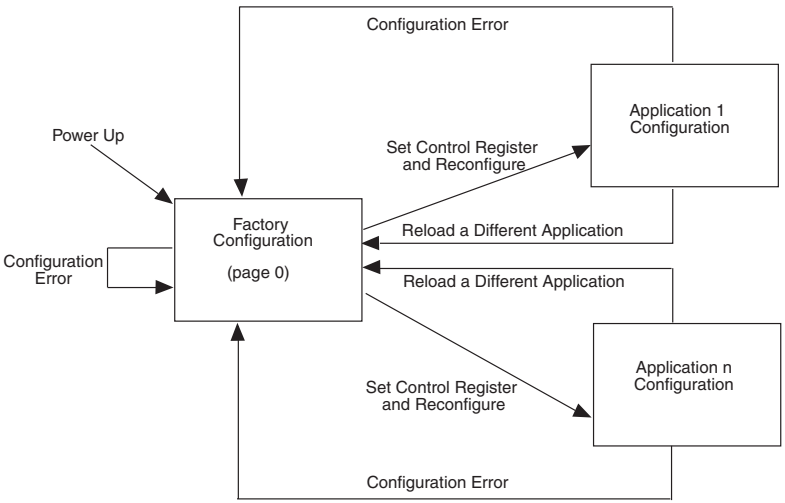
remote access. This corresponds to PGM[2..0] = 000 of the enhanced configuration device or standard flash memory, and start address location 0x000000 in the serial configuration device.

The factory image is user designed and contains soft logic to:

- Process any errors based on status information from the dedicated remote system upgrade circuitry
- Communicate with the remote host and receive new application configurations, and store this new configuration data in the local non-volatile memory device
- Determine which application configuration is to be loaded into the FPGA
- Enable or disable the user watchdog timer and load its time-out value (optional)
- Instruct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle

Figure 14–4 shows the transitions between the factory and application configurations in remote update mode.

Figure 14–4. Transitions Between Configurations in Remote Update Mode



After power up or a configuration error, the factory configuration logic should write the remote system upgrade control register to specify the page address of the application configuration to be loaded. The factory configuration should also specify whether or not to enable the user watchdog timer for the application configuration and, if enabled, specify the timer setting.

The user watchdog timer ensures that the application configuration is valid and functional. After confirming the system is healthy, the user-designed application configuration should reset the timer periodically during user-mode operation of an application configuration. This timer reset logic should be a user-designed hardware and/or software health monitoring signal that indicates error-free system operation. If the user application configuration detects a functional problem or if the system hangs, the timer is not reset in time and the dedicated circuitry updates the remote system upgrade status register, triggering the device to load the factory configuration. The user watchdog timer is automatically disabled for factory configurations.

Only valid application configurations designed for remote update mode include the logic to reset the timer in user mode.

For more information about the user watchdog timer, see "User Watchdog Timer" on page 14–20.

If there is an error while loading the application configuration, the remote system upgrade status register is written by the Stratix II or Stratix II GX FPGA's dedicated remote system upgrade circuitry, specifying the cause of the reconfiguration. Actions that cause the remote system upgrade status register to be written are:

- nSTATUS driven low externally
- Internal CRC error
- User watchdog timer time out
- A configuration reset (logic array nCONFIG signal or external nCONFIG pin assertion)

Stratix II and Stratix II GX FPGAs automatically load the factory configuration located at page address zero. This user-designed factory configuration should read the remote system upgrade status register to determine the reason for reconfiguration. The factory configuration should then take appropriate error recovery steps and write to the remote system upgrade control register to determine the next application configuration to be loaded.

When Stratix II or Stratix II GX devices successfully load the application configuration, they enter into user mode. In user mode, the soft logic (Nios processor or state machine and the remote communication interface) assists the Stratix II or Stratix II GX device in determining when a remote system update is arriving. When a remote system update arrives, the soft logic receives the incoming data, writes it to the configuration memory device, and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and initiates system reconfiguration.

Stratix II and Stratix II GX FPGAs support the remote update mode in the AS, FPP, PS, and PPA configuration schemes. In the FPP, PS, and PPA schemes, the MAX II device, microprocessor, or enhanced configuration device should sample the PGM[2..0] outputs from the Stratix II or Stratix II GX FPGA and transmit the appropriate configuration image. In the AS scheme, the Stratix II or Stratix II GX device uses the page addresses to read configuration data out of the serial configuration device.

Local Update Mode

Local update mode is a simplified version of the remote update mode. This feature allows systems to load an application configuration immediately upon power up without loading the factory configuration first. Local update mode does not require the factory configuration to determine which application configuration to load, because only one application configuration is allowed (at page address one (PGM [2..0] = 001). You can update this application configuration remotely. If an error occurs while loading the application configuration, the factory configuration is automatically loaded.

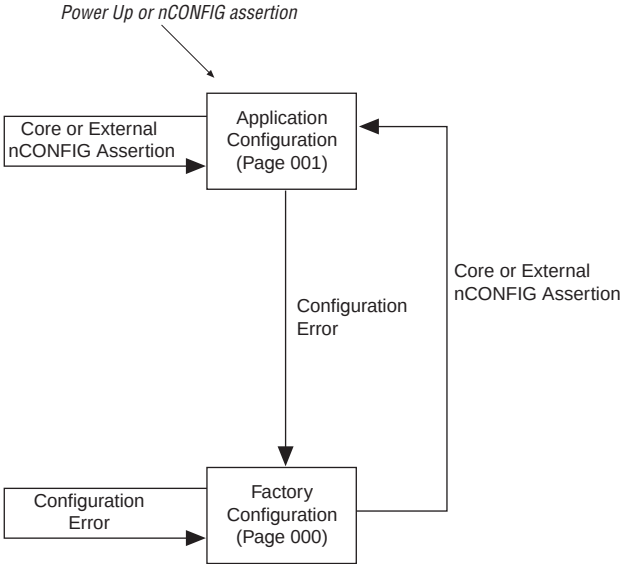
Upon power up or nCONFIG assertion, the dedicated remote system upgrade circuitry drives out "001" on the PGM[] pins selecting the application configuration stored in page one. If the device encounters any errors during the configuration cycle, the remote system upgrade circuitry retries configuration by driving PGM[2..0] to zero (PGM[2..0] = 000) to select the factory configuration image. The error conditions that trigger a return to the factory configuration are:

- An internal CRC error
- An external error signal (nSTATUS detected low)

When the remote system upgrade circuitry detects an external configuration reset (nCONFIG pulsed low) or internal configuration reset (logic array nCONFIG assertion), the device attempts to reload the application configuration from page one.

Figure 14–5 shows the transitions between configurations in local update mode.

Figure 14–5. Transitions Between Configurations in Local Update Mode



Stratix II and Stratix II GX FPGAs support local update mode in the FPP, PS, and PPA configuration schemes. In these schemes, the MAX II device, microprocessor, or enhanced configuration device should sample the PGM[2..0] outputs from the Stratix II or Stratix II GX FPGA and transmit the appropriate configuration image.

Local update mode is not supported with the AS configuration scheme, (or serial configuration device), because the Stratix II or Stratix II GX FPGA cannot determine the start address of the application configuration page upon power up. While the factory configuration is always located at memory address 0x000000, the application configuration can be located at any other sector boundary within the serial configuration device. The start address depends on the size of the factory configuration and is user selectable. Hence, only remote update mode is supported in the AS configuration scheme.

☞ Local update mode is not supported in the AS configuration scheme (with a serial configuration device).

Local update mode supports read access to the remote system upgrade status register. The factory configuration image can use this error status information to determine if a new application configuration must be downloaded from the remote source. After a remote update, the user design should assert the logic array configuration reset (nCONFIG) signal to load the new application configuration.

The device does not support write access to the remote system upgrade control register in local update mode. Write access is not required because this mode only supports one application configuration (eliminating the need to write in a page address) and does not support the user watchdog timer (eliminating the need to enable or disable the timer or specify its time-out value).

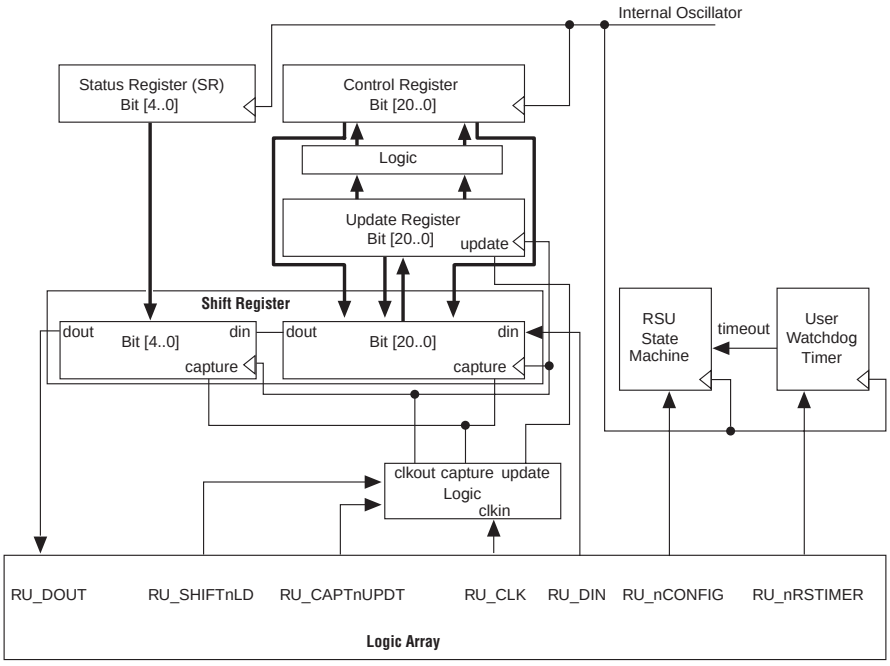
☞ The user watchdog timer is disabled in local update mode.

☞ Write access to the remote system upgrade control register is disabled in local update mode. However, the device supports read access to obtain error status information.

Dedicated Remote System Upgrade Circuitry

This section explains the implementation of the Stratix II or Stratix II GX remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces to the user-defined factory application configurations implemented in the FPGA logic array to provide the complete remote configuration solution. The remote system upgrade circuitry contains the remote system upgrade registers, a watchdog timer, and a state machine that controls those components. Figure 14–6 shows the remote system upgrade block's data path.

Figure 14–6. Remote System Upgrade Circuit Data Path



Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that store the page addresses, watchdog timer settings, and status information. These registers are detailed in Table 14–3.

Table 14–3. Remote System Upgrade Registers (Part 1 of 2)

Register	Description
Shift register	This register is accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic. Write access is enabled in remote update mode for factory configurations to allow writes to the update register. Write access is disabled in local update mode and for all application configurations in remote update mode.
Control register	This register contains the current page address, the user watchdog timer settings, and one bit specifying whether the current configuration is a factory configuration or an application configuration. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is initiated, the contents of the update register are written into the control register.

Table 14–3. Remote System Upgrade Registers (Part 2 of 2)

Register	Description
Update register	This register contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a read in a factory configuration, this register is read into the shift register.
Status register	This register is written to by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

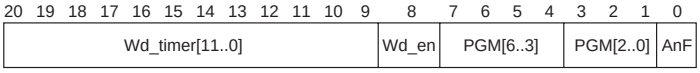
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (RU_CLK).

Remote System Upgrade Control Register

The remote system upgrade control register stores the application configuration page address and user watchdog timer settings. The control register functionality depends on the remote system upgrade mode selection. In remote update mode, the control register page address bits are set to all zeros (7'b0 = 0000_000) at power up in order to load the factory configuration. However, in local update mode the control register page address bits power up as (7'b1 = 0000_001) in order to select the application configuration. Additionally, the control register cannot be updated in local update mode, whereas a factory configuration in remote update mode has write access to this register.

The control register bit positions are shown in Figure 14–7 and defined in Table 14–4. In the figure, the numbers show the bit position of a setting within a register. For example, bit number 8 is the enable bit for the watchdog timer.

Figure 14–7. Remote System Upgrade Control Register



The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix II or Stratix II GX device is the factory configuration or an application configuration. This bit is set high at power up in local update mode, and is set low by the remote system upgrade circuitry when an error condition causes a fall-back to factory configuration. When the AnF bit is high, the control register access is limited to read operations. When the AnF bit is low, the register allows write operations and disables the watchdog timer.

In remote update mode, factory configuration design should set this bit high (1'b1) when updating the contents of the update register with application page address and watchdog timer settings.

Table 14–4. Remote System Upgrade Control Register Contents

Control Register Bit	Remote System Upgrade Mode	Value	Definition
AnF *(1)*	Local update Remote update	1'b1 1'b0	Application not factory
PGM[2..0]	Local update Remote update (FPP, PS, PPA)	3'b001 3'b000	Page mode select
	Remote update (AS)	3'b000	AS configuration start address (StAdd[18..16])
PGM[6..3]	Local update Remote update (FPP, PS, PPA)	4'b0000 4'b0000	Not used
	Remote update (AS)	4'b0000	AS configuration start address (StAdd[22..19])
Wd_en	Remote update	1'b0	User watchdog timer enable bit
Wd_timer[11..0]	Remote update	12'b000000000000	User watchdog time-out value (most significant 12 bits of 29-bit count value: {Wd_timer[11..0], 17'b0})

Note to Table 14–4:

(1) In remote update mode, the remote configuration block does not update the AnF bit automatically (you can update it manually). In local update mode, the remote configuration updates the AnF bit with 0 in the factory page and 1 in the application page.

Remote System Upgrade Status Register

The remote system upgrade status register specifies the reconfiguration trigger condition. The various trigger and error conditions include:

- CRC (cyclic redundancy check) error during application configuration
- nSTATUS assertion by an external device due to an error
- FPGA logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- External configuration reset (nCONFIG) assertion
- User watchdog timer time out

Figure 14–8 and Table 14–5 specify the contents of the status register. The numbers in the figure show the bit positions within a 5-bit register.

Figure 14–8. Remote System Upgrade Status Register

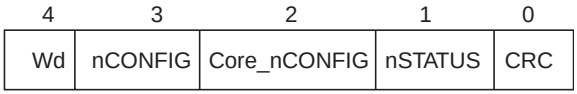


Table 14–5. Remote System Upgrade Status Register Contents

Status Register Bit	Definition	POR Reset Value
CRC (from configuration)	CRC error caused reconfiguration	1 bit '0'
nSTATUS	nSTATUS caused reconfiguration	1 bit '0'
CORE *(1)* CORE_nCONFIG	Device logic array caused reconfiguration	1 bit '0'
nCONFIG	nCONFIG caused reconfiguration	1 bit '0'
Wd	Watchdog timer caused reconfiguration	1 bit '0'

Note to Table 14–5:

(1) Logic array reconfiguration forces the system to load the application configuration data into the Stratix II or Stratix II GX device. This occurs after the factory configuration specifies the appropriate application configuration page address by updating the update register.

Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (see Table 14–3 on page 14–15). While both registers can only be updated when the FPGA is loaded with a factory configuration image, the update register writes are controlled by the user logic, and the control register writes are controlled by the remote system upgrade state machine.

In factory configurations, the user logic should send the AnF bit (set high), the page address, and watchdog timer settings for the next application configuration bit to the update register. When the logic array configuration reset (RU_nCONFIG) goes high, the remote system upgrade state machine updates the control register with the contents of the update register and initiates system reconfiguration from the new application page.

In the event of an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (page zero or page one, based on mode and error condition) by setting the control register accordingly. Table 14–6 lists the contents of the control register after such an event occurs for all possible error or trigger conditions.

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition but before the factory configuration is loaded.

Table 14–6. Control Register Contents After an Error or Reconfiguration Trigger Condition

Reconfiguration Error/Trigger	Control Register Setting	
	Remote Update	Local Update
nCONFIG reset	All bits are 0	PGM[6..0] = 7'b0000001 AnF = 1 All other bits are 0
nSTATUS error	All bits are 0	All bits are 0
CORE triggered reconfiguration	Update register	PGM[6..0] = 7'b0000001 AnF = 1 All other bits are 0
CRC error	All bits are 0	All bits are 0
Wd time out	All bits are 0	All bits are 0

Read operations during factory configuration access the contents of the update register. This feature is used by the user logic to verify that the page address and watchdog timer settings were written correctly. Read operations in application configurations access the contents of the control register. This information is used by the user logic in the application configuration.

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from stalling the device indefinitely. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the FPGA.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29-bits-wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The granularity of the timer setting is 2^{15} cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator. Table 14–7 specifies the operating range of the 10-MHz internal oscillator.

Table 14–7. 10-MHz Internal Oscillator Specifications

Minimum	Typical	Maximum	Units
5	6.5	10	MHz

The user watchdog timer begins counting once the application configuration enters FPGA user mode. This timer must be periodically reloaded or reset by the application configuration before the timer expires by asserting `RU_nRSTIMER`. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (`Wd`) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

The user watchdog timer is not enabled during the configuration cycle of the FPGA. Errors during configuration are detected by the CRC engine. Also, the timer is disabled for factory configurations. Functional errors should not exist in the factory configuration since it is stored and validated during production and is never updated remotely.

The user watchdog timer is disabled in factory configurations and during the configuration cycle of the application configuration. It is enabled after the application configuration enters user mode.

Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array

The dedicated remote system upgrade circuitry drives (or receives) seven signals to (or from) the FPGA logic array. The FPGA logic array uses these signals to read and write the remote system upgrade control, status, and update registers using the remote system upgrade shift register. Table 14–8 lists each of these seven signals and describes their functionality.

Except for RU_nRSTIMER and RU_CAPTnUPDT, the logic array signals are enabled for both remote and local update modes and for both factory and application configurations. RU_nRSTIMER is only valid for application configurations in remote update mode, since local update configurations and factory configurations have the user watchdog timer disabled. When RU_CAPTnUPDT is low, the device can write to the update register only for factory configurations in remote update mode, since this is the only case where the update register is written to by the user logic. When the RU_nCONFIG signal goes high, the contents of the update register are written into the control register for controlling the next configuration cycle.

Table 14–8. Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array (Part 1 of 3)

Signal Name	Signal Direction	Description
RU_nRSTIMER	Input to remote system upgrade block (driven by FPGA logic array)	Request from the application configuration to reset the user watchdog timer with its initial count. A falling edge of this signal triggers a reset of the user watchdog timer.
RU_nCONFIG	Input to remote system upgrade block (driven by FPGA logic array)	When driven low, this signal triggers the device to reconfigure. If asserted by the factory configuration in remote update mode, the application configuration specified in the remote update control register is loaded. If requested by the application configuration in remote update mode, the factory configuration is loaded. In the local updated mode, the application configuration is loaded whenever this signal is asserted.

Table 14–8. Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array (Part 2 of 3)

Signal Name	Signal Direction	Description
RU_CLK	Input to remote system upgrade block (driven by FPGA logic array)	Clocks the remote system upgrade shift register and update register so that the contents of the status, control, and update registers can be read, and so that the contents of the update register can be loaded. The shift register latches data on the rising edge of this clock signal.
RU_SHIFTnLD	Input to remote system upgrade block (driven by FPGA logic array)	This pin determines if the shift register contents are shifted over during the next clock edge or loaded in/out. When this signal is driven high (1'b1), the remote system upgrade shift register shifts data left on each rising edge of RU_CLK. When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven low (1'b0), the remote system upgrade update register is updated with the contents of the shift register on the rising edge of RU_CLK. When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven high (1'b1), the remote system upgrade shift register captures the status register and either the control or update register (depending on whether the current configuration is application or factory, respectively) on the rising edge of RU_CLK.
RU_CAPTnUPDT	Input to remote system upgrade block (driven by FPGA logic array)	This pin determines if the contents of the shift register are captured or updated on the next clock edge. When the RU_SHIFTnLD signal is driven high (1'b1), this input signal has no function. When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven high (1'b1), the remote system upgrade shift register captures the status register and either the control or update register (depending on whether the current configuration is application or factory, respectively) on the rising edge of RU_CLK. When RU_SHIFTnLD is driven low (1'b0) and RU_CAPTnUPDT is driven low (1'b0), the remote system upgrade update register is updated with the contents of the shift register on the rising edge of RU_CLK. In local update mode, a low input on RU_CAPTnUPDT has no function, because the update register cannot be updated in this mode.

Table 14–8. Interface Signals between Remote System Upgrade Circuitry and FPGA Logic Array (Part 3 of 3)

Signal Name	Signal Direction	Description
RU_DIN	Input to remote system upgrade block (driven by FPGA logic array)	Data to be written to the remote system upgrade shift register on the rising edge of RU_CLK. To load data into the shift register, RU_SHIFTnLD must be asserted.
RU_DOUT	Output from remote system upgrade block (driven to FPGA logic array)	Output data from the remote system upgrade shift register to be read by logic array logic. New data arrives on each rising edge of RU_CLK.

Remote System Upgrade Pin Descriptions

Table 14–9 describes the dedicated remote system upgrade configuration pins.

For descriptions of all the configuration pins, refer to the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook* and the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*.

Table 14–9. Stratix II and Stratix II GX Remote System Upgrade Pins

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
RUnLU	N/A if using remote system upgrade in FPP, PS, AS, or PPA modes. I/O if not using these modes.	Remote configuration in FPP, PS, or PPA	Input	Input that selects between remote update and local update. A logic high (1.5-V, 1.8-V, 2.5-V, 3.3-V) selects remote update, and a logic low selects local update. When not using remote update or local update configuration modes, this pin is available as a general-purpose user I/O pin. When using remote configuration in AS mode, set the RUnLU pin to high because AS does not support local update.
PGM[2..0]	N/A if using remote system upgrade in FPP, PS, or PPA modes. I/O if not using these modes.	Remote configuration in FPP, PS or PPA	Output	These output pins select one of eight pages in the memory (either flash or enhanced configuration device) when using remote update mode. When not using remote update or local update configuration modes, these pins are available as general-purpose user I/O pins.

Quartus II Software Support

Implementation in your design requires an remote system upgrade interface between the FPGA logic array and remote system upgrade circuitry. You also need to generate configuration files for production and remote programming of the system configuration memory. The Quartus® II software provides these features.

The two implementation options, altremote_update megafunction and remote system upgrade atom, are for the interface between the remote system upgrade circuitry and the FPGA logic array interface.

altremote_update Megafunction

The altremote_update megafunction provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read/write protocol in FPGA logic. This implementation is suitable for designs that implement the factory configuration functions using a Nios processor in the FPGA.

Tables 14–10 and 14–11 describe the input and output ports available on the altremote_update megafunction. Table 14–12 shows the param[2..0] bit settings.

Table 14–10. Input Ports of the altremote_update Megafunction (Part 1 of 2)

Port Name	Required	Source	Description
clock	Y	Logic Array	Clock input to the altremote_update block. All operations are performed with respects to the rising edge of this clock.
reset	Y	Logic Array	Asynchronous reset, which is used to initialize the remote update block. To ensure proper operation, the remote update block must be reset before first accessing the remote update block. This signal is not affected by the busy signal and will reset the remote update block even if busy is logic high. This means that if the reset signal is driven logic high during writing of a parameter, the parameter will not be properly written to the remote update block.
reconfig	Y	Logic Array	When driven logic high, reconfiguration of the device is initiated using the current parameter settings in the remote update block. If busy is asserted, this signal is ignored. This is to ensure all parameters are completely written before reconfiguration begins.
reset_timer	N	Logic Array	This signal is required if you are using the watchdog timer feature. A logic high resets the internal watchdog timer. This signal is not affected by the busy signal and can reset the timer even when the remote update block is busy. If this port is left connected, the default value is 0.
read_param	N	Logic Array	Once read_param is sampled as a logic high, the busy signal is asserted. While the parameter is being read, the busy signal remains asserted, and inputs on param[] are ignored. Once the busy signal is deactivated, the next parameter can be read. If this port is left unconnected, the default value is 0.
write_param	N	Logic Array	This signal is required if you intend on writing parameters to the remote update block. When driven logic high, the parameter specified on the param[] port should be written to the remote update block with the value on data_in[]. The number of valid bits on data_in[] is dependent on the parameter type. This signal is sampled on the rising edge of clock and should only be asserted for one clock cycle to prevent the parameter from being re-read on subsequent clock cycles. Once write_param is sampled as a logic high, the busy signal is asserted. While the parameter is being written, the busy signal remains asserted, and inputs on param[] and data_in[] are ignored. Once the busy signal is deactivated, the next parameter can be written. This signal is only valid when the Current_Configuration parameter is factory since parameters cannot be written in application configurations. If this port is left unconnected, the default value is 0.

Table 14–10. Input Ports of the altremote_update Megafunction (Part 2 of 2)

Port Name	Required	Source	Description
param[2..0]	N	Logic Array	3-bit bus that selects which parameter should be read or written. If this port is left unconnected, the default value is 0.
data_in[11..0]	N	Logic Array	This signal is required if you intend on writing parameters to the remote update block 12-bit bus used when writing parameters, which specifies the parameter value. The parameter value is requested using the param[] input and by driving the write_param signal logic high, at which point the busy signal goes logic high and the value of the parameter is captured from this bus. For some parameters, not all 12 bits are used, in which case only the least significant bits are used. This port is ignored if the Current_Configuration parameter is set to an application configuration since writing of parameters is only allowed in the factory configuration. If this port is left unconnected, the default value is 0.

Note to Table 14–10:
(1) Logic array source means that you can drive the port from internal logic or any general-purpose I/O pin.

Table 14–11. Output Ports of the altremote_update Megafunction (Part 1 of 2)

Port Name	Required	Destination	Description
busy	Y	Logic Array	When this signal is a logic high, the remote update block is busy either reading or writing a parameter. When the remote update block is busy, it ignores its data_in[], param[], and reconfig inputs. This signal goes high when read_param or write_param is asserted and remains asserted until the operation is complete.
pgm_out[2..0]	Y	PGM[2..0] pins	3-bit bus that specifies the page pointer of the configuration data to be loaded when the device is reconfigured. This port must be connected to the PGM[] output pins, which should be connected to the external configuration device.

Table 14–11. Output Ports of the altremote_update Megafunction (Part 2 of 2)

Port Name	Required	Destination	Description
data_out[11..0]	N	Logic Array	12-bit bus used when reading parameters, which reads out the parameter value. The parameter value is requested using the param[] input and by driving the read_param signal logic high, at which point the busy signal goes logic high. When the busy signal goes low, the value of the parameter is driven out on this bus. The data_out[] port is only valid after a read_param has been issued and once the busy signal is deasserted. At any other time, its output values are invalid. For example, even though the data_out[] port may toggle during a writing of a parameter, these values are not a valid representation of what was actually written to the remote update block. For some parameters, not all 12 bits are used, in which case only the least significant bits are used.

Note to Table 14–11:
(1) Logic array destination means that you can drive the port to internal logic or any general-purpose I/O pin.

Table 14–12. Parameter Settings for the altremote_update Megafunction (Part 1 of 2)

Selected Parameter	param[2..0] Bit Setting	Width of Parameter Value	POR Reset Value	Description
Status Register Contents	000	5	5 bit '0	Specifies the reason for re-configuration, which could be caused by a CRC error during configuration, nSTATUS being pulled low due to an error, the device core caused an error, nCONFIG pulled low, or the watchdog timer timed-out. This parameter can only be read.
Watchdog Timeout Value	010	12	12 bits '0	User watchdog timer time-out value. Writing of this parameter is only allowed when in the factory configuration.
Watchdog Enable	011	1	1 bit '0	User watchdog timer enable. Writing of this parameter is only allowed when in the factory configuration
Page select	100	3 (FPP, PS, PPA)	3 bit '001' - Local configuration	Page mode selection. Writing of this parameter is only allowed when in the factory configuration.
			3 bit '000' - Remote configuration	
		7 (AS)	7 bit '0000000' - Remote configuration	

Table 14–12. Parameter Settings for the altremote_update Megafunction (Part 2 of 2)

Selected Parameter	param[2..0] Bit Setting	Width of Parameter Value	POR Reset Value	Description
Current configuration (AnF)	101	1	1 bit '0' - Factory 1 bit '1' - Application	Specifies whether the current configuration is factory or and application configuration. This parameter can only be read.
Illegal values	001			
	110			
	111			

Remote System Upgrade Atom

The remote system upgrade atom is a WYSIWYG atom or primitive that can be instantiated in your design. The primitive is used to access the remote system upgrade shift register, logic array reset, and watchdog timer reset signals. The ports on this primitive are the same as those listed in Table 14–8. This implementation is suitable for designs that implement the factory configuration functions using state machines (without a processor).

System Design Guidelines

The following general guidelines are applicable when implementing remote system upgrade in Stratix II and Stratix II GX FPGAs. Guidelines for specific configuration schemes are also discussed in this section.

- After downloading a new application configuration, the soft logic implemented in the FPGA can validate the integrity of the data received over the remote communication interface. This optional step helps avoid configuration attempts with bad or incomplete configuration data. However, in the event that bad or incomplete configuration data is sent to the FPGA, it detects the data corruption using the CRC signature attached to each configuration frame.
- The auto-reconfigure on configuration error option bit is ignored when remote system upgrade is enabled in your system. This option is always enabled in remote configuration designs, allowing your system to return to the safe factory configuration in the event of an application configuration error or user watchdog timer time out.

Remote System Upgrade With Serial Configuration Devices

Remote system upgrade support in the AS configuration scheme is similar to support in other schemes, with the following exceptions:

- The remote system upgrade block provides the AS configuration controller inside the Stratix II or Stratix II GX FPGA with a 7-bit page start address (PGM[6..0]) instead of driving the 3-bit page mode pins (PGM[2..0]) used in FPP, PS, and PPA configuration schemes. This 7-bit address forms the 24-bit configuration start address (StAdd[23..0]). Table 14–13 illustrates the start address generation using the page address registers.
- The configuration start address for factory configuration is always set to 24'b0.
- PGM[2..0] pins on Stratix II devices are not used in AS configuration schemes and can not be used as regular I/O pins.
- The Nios ASMI peripheral can be used to update configuration data within the serial configuration device.

Table 14–13. AS Configuration Start Address Generation

Serial Configuration Device	Serial Configuration Device Density (MB)	Add[23]	PGM[6..0] (Add[22..16])	Add[15..0]
EPCS64	64	0	MSB[6..0]	All 0s
EPCS16	16	0	00, MSB[4..0]	All 0s
EPCS4	4	0	0000, MSB[2..0]	All 0s

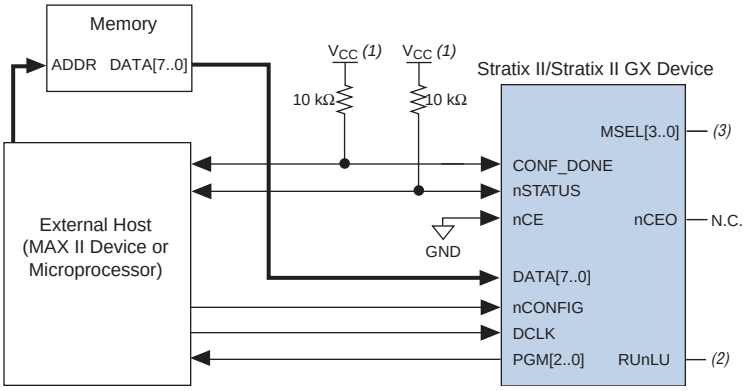
Remote System Upgrade With a MAX II Device or Microprocessor and Flash Device

This setup requires the MAX II device or microprocessor to support page addressing. MAX II or microprocessor devices implementing remote system upgrade should emulate the enhanced configuration device page mode feature. The PGM[2..0] output pins from the Stratix II or Stratix II GX device must be sampled to determine which configuration image is to be loaded into the FPGA.

If the FPGA does not release CONF_DONE after all data has been sent, the MAX II microprocessor should reset the FPGA back to the factory image by pulsing its nSTATUS pin low.

The MAX II device or microprocessor and flash configuration can use FPP, PS, or PPA. Decompression and design security features are supported in the FPP (requires 4× DCLK) and PS modes only. Figure 14–9 shows a system block diagram for remote system upgrade with the MAX II device or microprocessor and flash.

Figure 14–9. System Block Diagram for Remote System Upgrade With MAX II Device or Microprocessor and Flash Device



Notes to Figure 14–9:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device.

(2) Connect `RUnLU` to GND or V_{CC} to select between remote and local update modes.

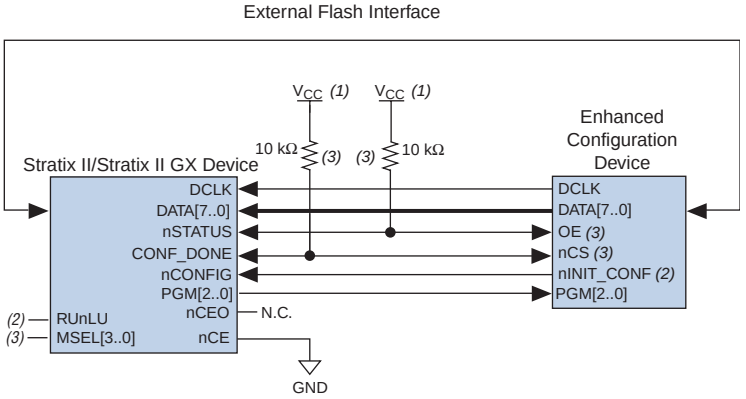
(3) Connect `MSEL[3..0]` to `0100` to enable remote update remote system upgrade mode.

Remote System Upgrade with Enhanced Configuration Devices

- Enhanced Configuration devices support remote system upgrade with FPP or PS configuration schemes. The Stratix II or Stratix II GX decompression and design security features are only supported in the PS mode. The enhanced configuration device's decompression feature is supported in both PS and FPP schemes.
- In remote update mode, neither the factory configuration nor the application configurations should alter the enhanced configuration device's option bits or the page 000 factory configuration data. This ensures that an error during remote update can always be resolved by reverting to the factory configuration located at page 000.

- The enhanced configuration device features an error checking mechanism to detect instances when the FPGA fails to detect the configuration preamble. In these instances, the enhanced configuration device pulses the nSTATUS signal low, and the remote system upgrade circuitry attempts to load the factory configuration. Figure 14–10 shows a system block diagram for remote system upgrade with enhanced configuration devices.

Figure 14–10. System Block Diagram for Remote System Upgrade with Enhanced Configuration Devices



Notes to Figure 14–10:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device.

(2) Connect RUnLU to GND or V_{CC} to select between remote and local update modes.

(3) Connect MSEL[3..0] to 0100 to enable remote update remote system upgrade mode.

Conclusion

Stratix II and Stratix II GX devices offer remote system upgrade capability, where you can upgrade a system in real-time through any network. Remote system upgrade helps to deliver feature enhancements and bug fixes without costly recalls, reduces time to market, and extends product life cycles. The dedicated remote system upgrade circuitry in Stratix II and Stratix II GX devices provides error detection, recovery, and status information to ensure reliable reconfiguration.

Referenced Documents

This chapter references the following documents:

- *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Handbook*

- *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Handbook*
- *Enhanced Configuration Devices (EPC4, EPC8 & EPC16) Data Sheet* chapter in volume 2 of the *Configuration Handbook*
- *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook*

Document Revision History

Table 14–14 shows the revision history for this chapter.

Table 14–14. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
October 2007, v4.5	Updated Table 14–7.	—
	Added the "Referenced Documents" section.	—
	Minor text edits.	—
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 13. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
May 2007, v4.4	Updated note to "Functional Description" section.	—
	Minor text edit to "Remote System Upgrade With Serial Configuration Devices" section.	—
February 2007 v4.3	Added the "Document Revision History" section to this chapter.	—
April 2006, v4.2	Chapter updated as part of the *Stratix II Device Handbook* update.	—
No change	Formerly chapter 12. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
December 2005, v4.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—
October 2005 v4.0	Added chapter to the *Stratix II GX Device Handbook*.	—

15. IEEE 1149.1 (JTAG) Boundary-Scan Testing for Stratix II & Stratix II GX Devices

SII52009-3.3

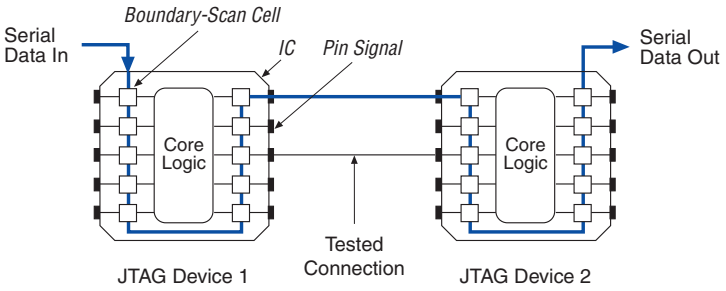
Introduction

As printed circuit boards (PCBs) become more complex, the need for thorough testing becomes increasingly important. Advances in surface-mount packaging and PCB manufacturing have resulted in smaller boards, making traditional test methods (such as; external test probes and "bed-of-nails" test fixture) harder to implement. As a result, cost savings from PCB space reductions increases the cost for traditional testing methods.

In the 1980s, the Joint Test Action Group (JTAG) developed a specification for boundary-scan testing that was later standardized as the IEEE Std. 1149.1 specification. This Boundary-Scan Test (BST) architecture offers the capability to test efficiently components on PCBs with tight lead spacing.

This BST architecture tests pin connections without using physical test probes and captures functional data while a device is operating normally. Boundary-scan cells in a device can force signals onto pins or capture data from pin or logic array signals. Forced test data is serially shifted into the boundary-scan cells. Captured data is serially shifted out and externally compared to expected results. Figure 15–1 illustrates the concept of BST.

Figure 15–1. IEEE Std. 1149.1 Boundary-Scan Testing



This chapter discusses how to use the IEEE Std. 1149.1 BST circuitry in Stratix® II and Stratix GX devices, including:

- IEEE Std. 1149.1 BST architecture
- IEEE Std. 1149.1 boundary-scan register
- IEEE Std. 1149.1 BST operation control
- I/O Voltage Support in JTAG Chain
- IEEE Std. 1149.1 BST circuitry utilization
- IEEE Std. 1149.1 BST circuitry disabling
- IEEE Std. 1149.1 BST guidelines
- Boundary-Scan Description Language (BSDL) support

In addition to BST, you can use the IEEE Std. 1149.1 controller for Stratix II and Stratix II GX device in-circuit reconfiguration (ICR). However, this chapter only discusses the BST feature of the IEEE Std. 1149.1 circuitry.

For information on configuring Stratix II devices via the IEEE Std. 1149.1 circuitry, refer to the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook*, or the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

When configuring via IJAG make sure that Stratix II, Stratix II GX, Stratix, Cyclone™ II, and Cyclone devices are within the first 17 devices in a JTAG chain. All of these devices have the same JTAG controller. If any of the Stratix II, Stratix II GX, Stratix, Cyclone II, and Cyclone devices are in the 18th or further position, configuration fails. This does not affect SignalTap® II or boundary-scan testing.

IEEE Std. 1149.1 BST Architecture

A Stratix II and Stratix II GX device operating in IEEE Std. 1149.1 BST mode uses four required pins, `TDI`, `TDO`, `TMS` and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS` and `TRST` pins have weak internal pull-ups. The `TDO` output pin is powered by V_{CCIO} in I/O bank 4. All of the JTAG input pins are powered by the 3.3-V V_{CCPD} supply. All user I/O pins are tri-stated during JTAG configuration.

For recommendations on how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to "I/O Voltage Support in JTAG Chain" on page 15–17.

Table 15–1 summarizes the functions of each of these pins.

Table 15–1. IEEE Std. 1149.1 Pin Descriptions

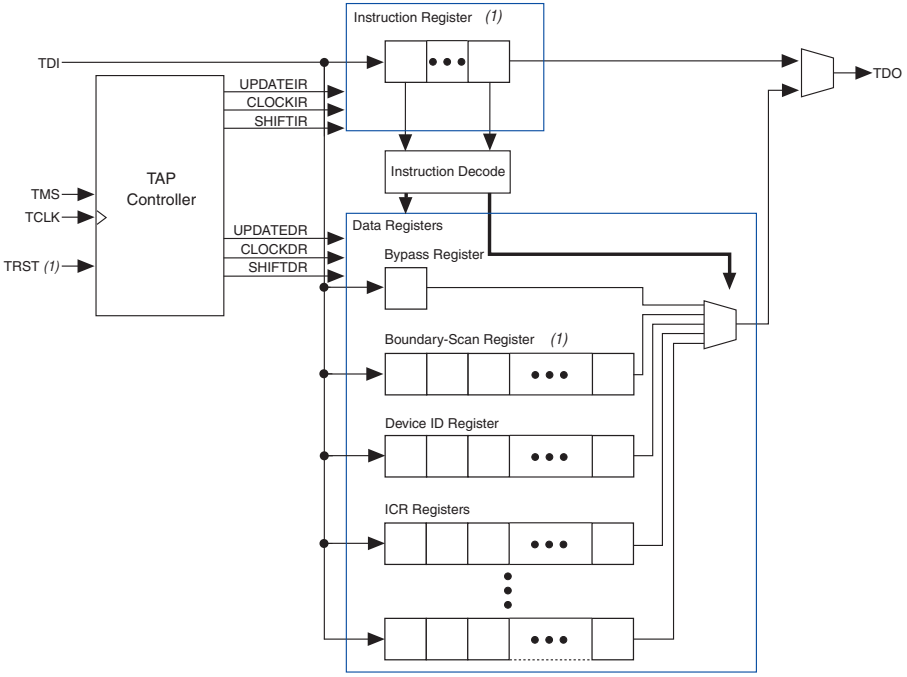
Pin	Description	Function
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK.
TDO	Test data output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device.
TMS	Test mode select	Input pin that provides the control signal to determine the transitions of the Test Access Port (TAP) controller state machine. Transitions within the state machine occur at the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK.
TCK	Test clock input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge.
TRST	Test reset input (optional)	Active-low input to asynchronously reset the boundary-scan circuit. This pin should be driven low when not in boundary-scan operation and for non-JTAG users the pin should be permanently tied to GND.

The IEEE Std. 1149.1 BST circuitry requires the following registers:

- The instruction register determines the action to be performed and the data register to be accessed.
- The bypass register is a 1-bit-long data register that provides a minimum-length serial path between TDI and TDO.
- The boundary-scan register is a shift register composed of all the boundary-scan cells of the device.

Figure 15–2 shows a functional model of the IEEE Std. 1149.1 circuitry.

Figure 15–2. IEEE Std. 1149.1 Circuitry



Note to Figure 15–2:
(1) Refer to the appropriate device data sheet for register lengths.

IEEE Std. 1149.1 boundary-scan testing is controlled by a test access port (TAP) controller. For more information on the TAP controller, refer to "IEEE Std. 1149.1 BST Operation Control" on page 15–7. The TMS and TCK pins operate the TAP controller, and the TDI and TDO pins provide the serial path for the data registers. The TDI pin also provides data to the instruction register, which then generates control logic for the data registers.

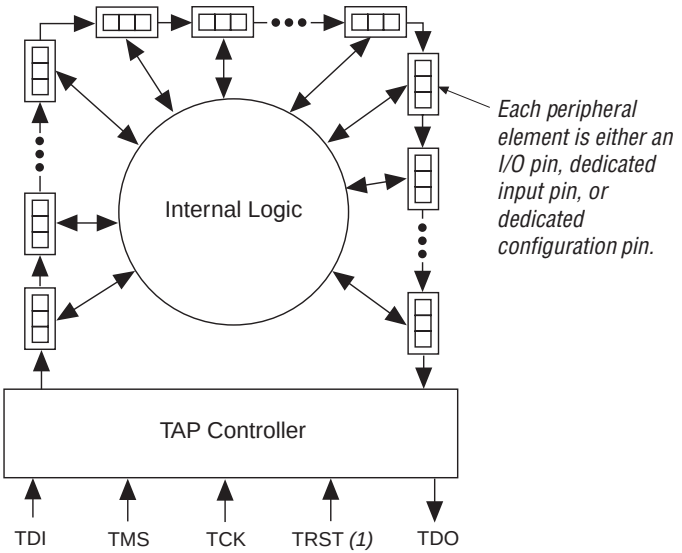
IEEE Std. 1149.1 Boundary-Scan Register

The boundary-scan register is a large serial shift register that uses the TDI pin as an input and the TDO pin as an output. The boundary-scan register consists of 3-bit peripheral elements that are associated with Stratix II or Stratix II GX I/O pins. You can use the boundary-scan register to test external pin connections or to capture internal data.

Refer to the *Configuration & Testing* chapter in volume 1 of the *Stratix II Device Handbook*, or the *Configuration & Testing* chapter in volume 1 of the *Stratix II GX Device Handbook* for the Stratix II family device boundary-scan register lengths.

Figure 15–3 shows how test data is serially shifted around the periphery of the IEEE Std. 1149.1 device.

Figure 15–3. Boundary-Scan Register



Boundary-Scan Cells of a Stratix II or Stratix II GX Device I/O Pin

The Stratix II or the Stratix II GX device 3-bit boundary-scan cell (BSC) consists of a set of capture registers and a set of update registers. The capture registers can connect to internal device data via the OUTJ, OEJ, and PIN_IN signals, while the update registers connect to external data through the PIN_OUT, and PIN_OE signals. The global control signals for the IEEE Std. 1149.1 BST registers (such as shift, clock, and update) are generated internally by the TAP controller. The MODE signal is generated by a decode of the instruction register. The data signal path for the boundary-scan register runs from the serial data in (SDI) signal to the serial data out (SDO) signal. The scan register begins at the TDI pin and ends at the TDO pin of the device.

Figure 15–4 shows the Stratix II and Stratix II GX device's user I/O boundary-scan cell.

Figure 15–4. Stratix II and Stratix II GX Device's User I/O BSC with IEEE Std. 1149.1 BST Circuitry

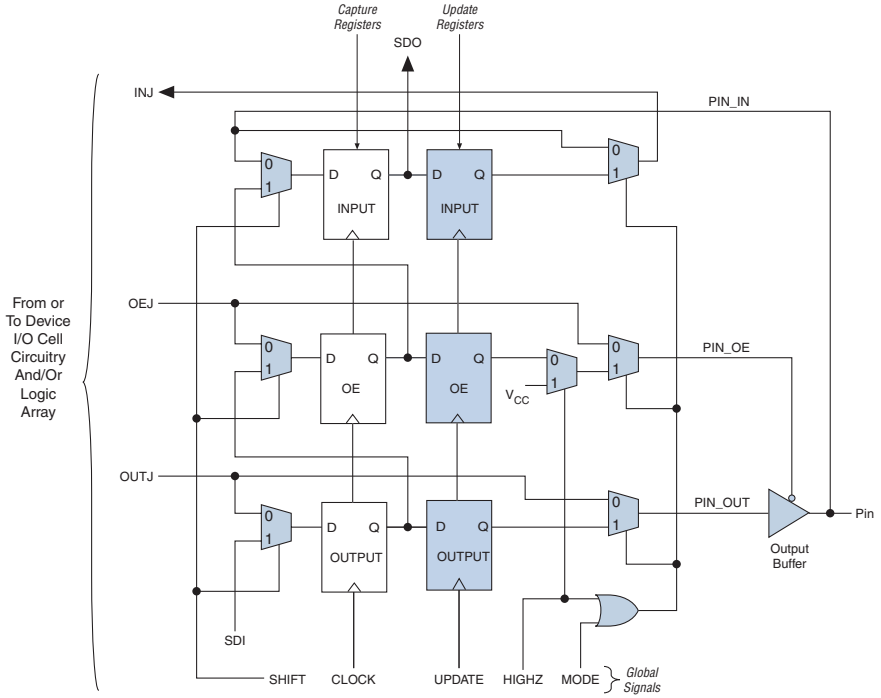


Table 15–2 describes the capture and update register capabilities of all boundary-scan cells within Stratix II and Stratix II GX devices.

Table 15–2. Stratix II and Stratix II GX Device Boundary Scan Cell Descriptions (Part 1 of 2) *Note (1)*

Pin Type	Captures			Drives			Comments
	Output Capture Register	OE Capture Register	Input Capture Register	Output Update Register	OE Update Register	Input Update Register	
User I/O pins	OUTJ	OEJ	PIN_IN	PIN_OUT	PIN_OE	INJ	NA
Dedicated clock input	0	1	PIN_IN	N.C. *(2)*	N.C. *(2)*	N.C. *(2)*	PIN_IN drives to clock network or logic array

Table 15–2. Stratix II and Stratix II GX Device Boundary Scan Cell Descriptions (Part 2 of 2) Note (1)

Pin Type	Captures			Drives			Comments
	Output Capture Register	OE Capture Register	Input Capture Register	Output Update Register	OE Update Register	Input Update Register	
Dedicated input *(3)*	0	1	PIN_IN	N.C. *(2)*	N.C. *(2)*	N.C. *(2)*	PIN_IN drives to control logic
Dedicated bidirectional (open drain) *(4)*	0	OEJ	PIN_IN	N.C. *(2)*	N.C. *(2)*	N.C. *(2)*	PIN_IN drives to configuration control
Dedicated bidirectional *(5)*	OUTJ	OEJ	PIN_IN	N.C. *(2)*	N.C. *(2)*	N.C. *(2)*	PIN_IN drives to configuration control and OUTJ drives to output buffer
Dedicated output *(6)*	OUTJ	0	0	N.C. *(2)*	N.C. *(2)*	N.C. *(2)*	OUTJ drives to output buffer

Notes to Table 15–2:

(1) TDI, TDO, TMS, TCK, all V_{CC} and GND pin types, VREF, and TEMP_DIODE pins do not have BSCs.
(2) No Connect (N.C.).
(3) This includes pins PLL_ENA, nCONFIG, MSEL0, MSEL1, MSEL2, MSEL3, nCE, VCCSEL, PORSEL, and nIO_PULLUP.
(4) This includes pins CONF_DONE and nSTATUS.
(5) This includes pin DCLK.
(6) This includes pin nCEO.

IEEE Std. 1149.1 BST Operation Control

Stratix II and Stratix II GX devices implement the following IEEE Std. 1149.1 BST instructions:

- SAMPLE/PRELOAD instruction mode is used to take snapshot of the device data without interrupting normal device operations
- EXTEST instruction mode is used to check external pin connections between devices
- BYPASS instruction mode is used when an instruction code consisting of all ones is loaded into the instruction register
- IDCODE instruction mode is used to identify the devices in an IEEE Std. 1149.1 chain
- USERCODE instruction mode is used to examine the user electronic signature within the device along an IEEE Std. 1149.1 chain.

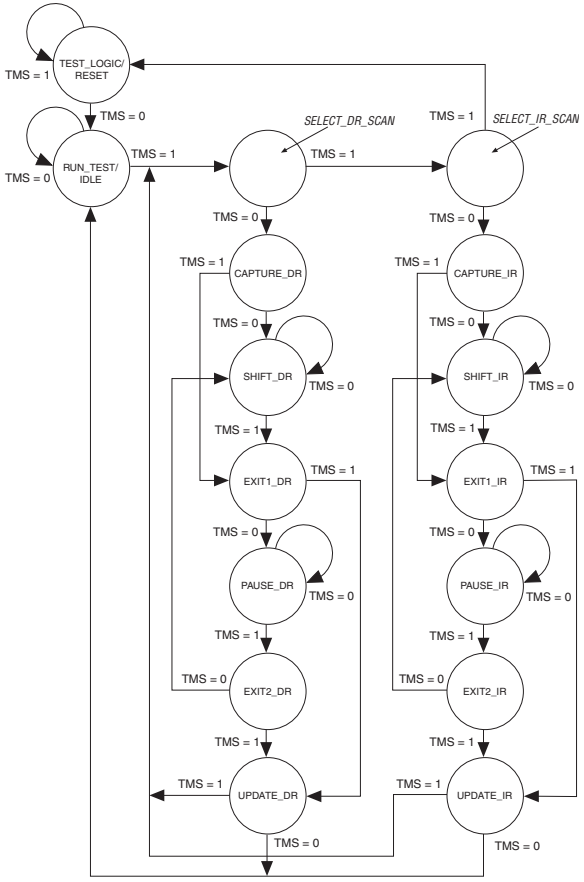
- CLAMP instruction mode is used to allow the state of the signals driven from the pins to be determined from the boundary-scan register while the bypass register is selected as the serial path between the TDI and TDO ports
- HIGHZ instruction mode sets all of the user I/O pins to an inactive drive state

The BST instruction length is 10 bits. These instructions are described later in this chapter.

For summaries of the BST instructions and their instruction codes, refer to the *Configuration & Testing* chapter in volume 1 of the *Stratix II Device Handbook*, or the *Configuration & Testing* chapter in volume 1 of the *Stratix II GX Device Handbook*.

The IEEE Std. 1149.1 TAP controller, a 16-state state machine clocked on the rising edge of TCK, uses the TMS pin to control IEEE Std. 1149.1 operation in the device. Figure 15–5 shows the TAP controller state machine.

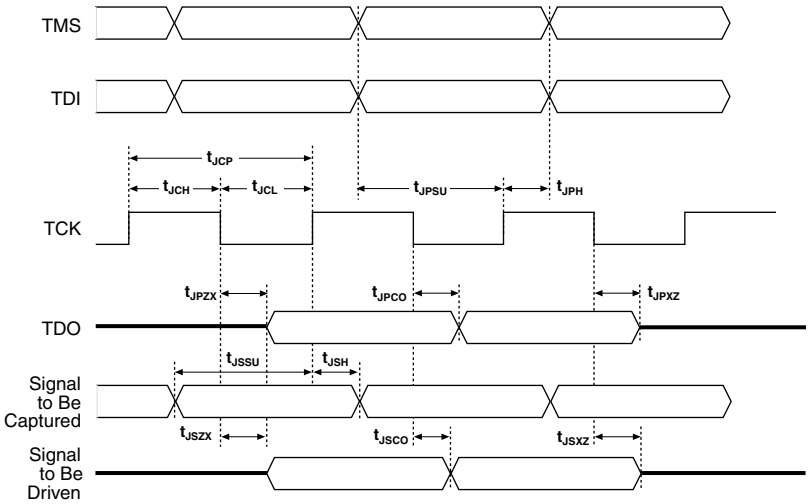
Figure 15–5. IEEE Std. 1149.1 TAP Controller State Machine



When the TAP controller is in the TEST_LOGIC/RESET state, the BST circuitry is disabled, the device is in normal operation, and the instruction register is initialized with IDCODE as the initial instruction. At device power-up, the TAP controller starts in this TEST_LOGIC/RESET state. In addition, forcing the TAP controller to the TEST_LOGIC/RESET state is done by holding TMS high for five TCK clock cycles or by holding the TRST pin low. Once in the TEST_LOGIC/RESET state, the TAP controller remains in this state as long as TMS is held high (while TCK is clocked) or TRST is held low.

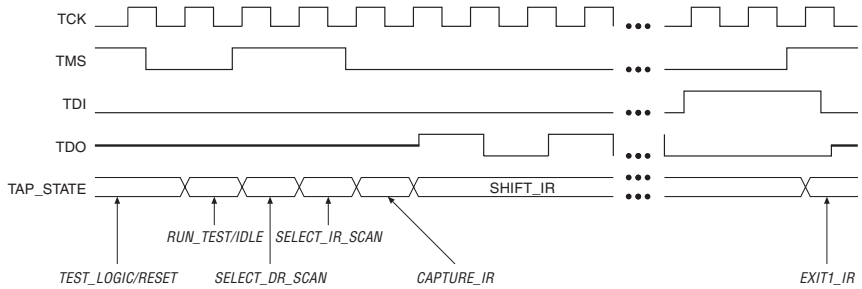
Figure 15–6 shows the timing requirements for the IEEE Std. 1149.1 signals.

Figure 15–6. IEEE Std. 1149.1 Timing Waveforms



To start IEEE Std. 1149.1 operation, select an instruction mode by advancing the TAP controller to the shift instruction register (SHIFT_IR) state and shift in the appropriate instruction code on the TDI pin. The waveform diagram in Figure 15–7 represents the entry of the instruction code into the instruction register. Figure 15–7 shows the values of TCK, TMS, TDI, TDO, and the states of the TAP controller. From the RESET state, TMS is clocked with the pattern 01100 to advance the TAP controller to SHIFT_IR.

Figure 15–7. Selecting the Instruction Mode



The TDO pin is tri-stated in all states except in the SHIFT_IR and SHIFT_DR states. The TDO pin is activated at the first falling edge of TCK after entering either of the shift states and is tri-stated at the first falling edge of TCK after leaving either of the shift states.

When the SHIFT_IR state is activated, TDO is no longer tri-stated, and the initial state of the instruction register is shifted out on the falling edge of TCK. TDO continues to shift out the contents of the instruction register as long as the SHIFT_IR state is active. The TAP controller remains in the SHIFT_IR state as long as TMS remains low.

During the SHIFT_IR state, an instruction code is entered by shifting data on the TDI pin on the rising edge of TCK. The last bit of the instruction code is clocked at the same time that the next state, EXIT1_IR, is activated. Set TMS high to activate the EXIT1_IR state. Once in the EXIT1_IR state, TDO becomes tri-stated again. TDO is always tri-stated except in the SHIFT_IR and SHIFT_DR states. After an instruction code is entered correctly, the TAP controller advances to serially shift test data in one of three modes. The three serially shift test data instruction modes are discussed on the following pages:

- "SAMPLE/PRELOAD Instruction Mode" on page 15–11
- "EXTEST Instruction Mode" on page 15–13
- "BYPASS Instruction Mode" on page 15–15

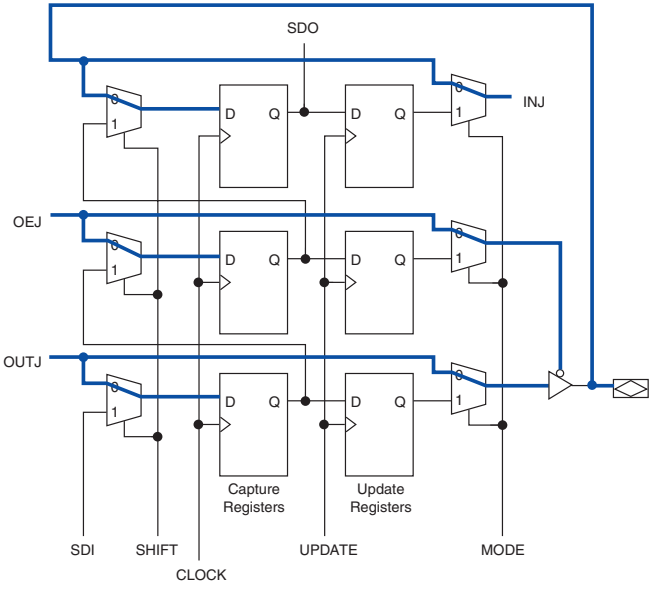
SAMPLE/PRELOAD Instruction Mode

The SAMPLE/PRELOAD instruction mode allows you to take a snapshot of device data without interrupting normal device operation. However, this instruction is most often used to preload the test data into the update registers prior to loading the EXTEST instruction. Figure 15–8 shows the capture, shift, and update phases of the SAMPLE/PRELOAD mode.

Figure 15–8. IEEE Std. 1149.1 BST SAMPLE/PRELOAD Mode

Capture Phase

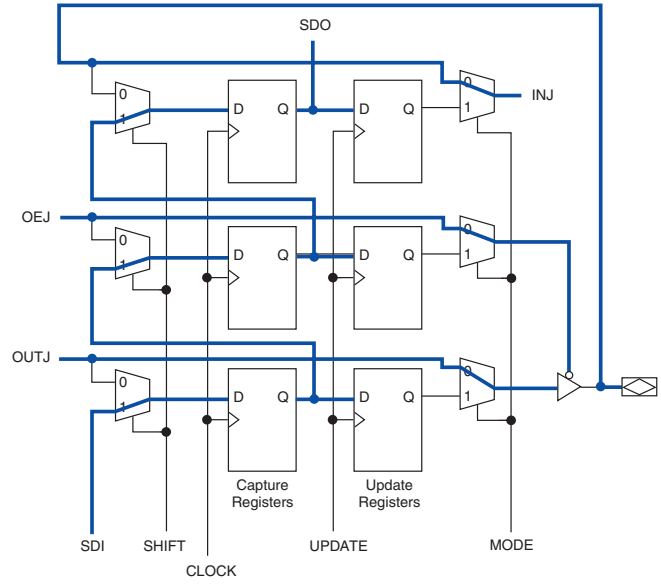
In the capture phase, the signals at the pin, OEJ and OUTJ, are loaded into the capture registers. The CLOCK signals is supplied by the TAP controller's CLOCKDR output. The data retained in these registers consists of signals from normal device operation.



Shift and Update Phases

In the shift phase, the previously captured signals at the pin, OEJ and OUTJ, are shifted out of the boundary-scan register via the TDO pin using CLOCK. As data is shifted out, the patterns for the next test can be shifted in via the TDI pin.

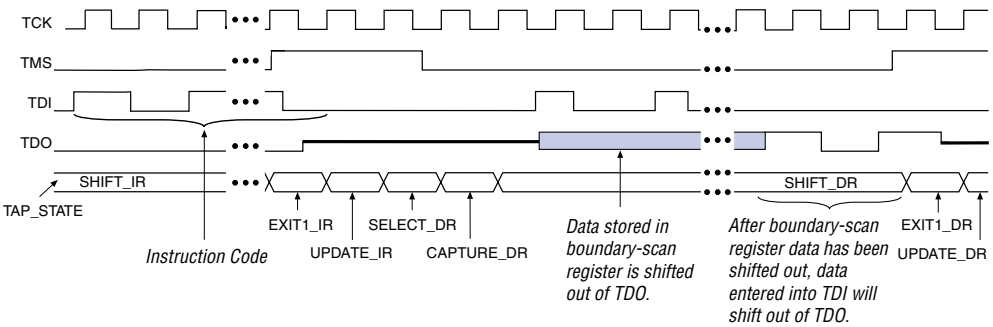
In the update phase, data is transferred from the capture to the UPDATE registers using the UPDATE clock. The data stored in the UPDATE registers can be used for the EXTEST instruction.



During the capture phase, multiplexers preceding the capture registers select the active device data signals. This data is then clocked into the capture registers. The multiplexers at the outputs of the update registers also select active device data to prevent functional interruptions to the device. During the shift phase, the boundary-scan shift register is formed by clocking data through capture registers around the device periphery and then out of the TDO pin. The device can simultaneously shift new test data into TDI and replace the contents of the capture registers. During the update phase, data in the capture registers is transferred to the update registers. This data can then be used in the EXTEST instruction mode. Refer to "EXTEST Instruction Mode" on page 15–13 for more information.

Figure 15–9 shows the SAMPLE/PRELOAD waveforms. The SAMPLE/PRELOAD instruction code is shifted in through the TDI pin. The TAP controller advances to the CAPTURE_DR state and then to the SHIFT_DR state, where it remains if TMS is held low. The data that was present in the capture registers after the capture phase is shifted out of the TDO pin. New test data shifted into the TDI pin appears at the TDO pin after being clocked through the entire boundary-scan register. Figure 15–9 shows that the instruction code at TDI does not appear at the TDO pin until after the capture register data is shifted out. If TMS is held high on two consecutive TCK clock cycles, the TAP controller advances to the UPDATE_DR state for the update phase.

Figure 15–9. SAMPLE/PRELOAD Shift Data Register Waveforms



EXTEST Instruction Mode

The EXTEST instruction mode is used primarily to check external pin connections between devices. Unlike the SAMPLE/PRELOAD mode, EXTEST allows test data to be forced onto the pin signals. By forcing known logic high and low levels on output pins, opens and shorts can be detected at pins of any device in the scan chain.

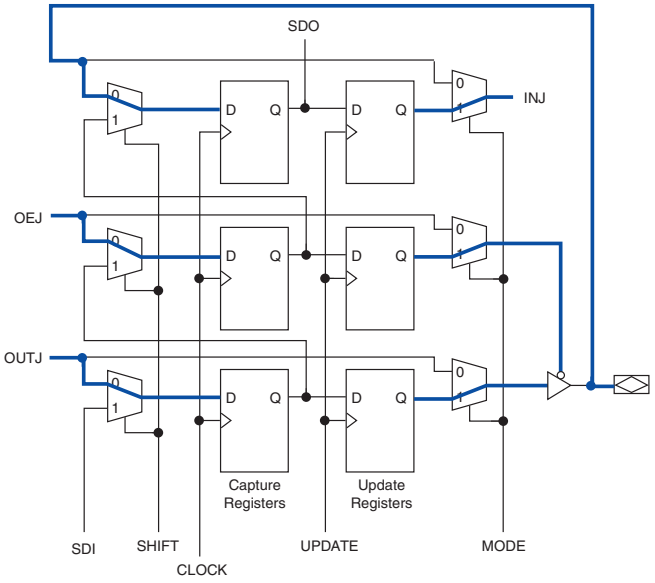
Figure 15–10 shows the capture, shift, and update phases of the EXTEST mode.

Figure 15–10. IEEE Std. 1149.1 BST EXTEST Mode

Capture Phase

In the capture phase, the signals at the pin, OEJ and OUTJ, are loaded into the capture registers. The CLOCK signals is supplied by the TAP controller's CLOCKDR output. Previously retained data in the update registers drive the PIN_IN, INJ, and allows the I/O pin to tri-state or drive a signal out.

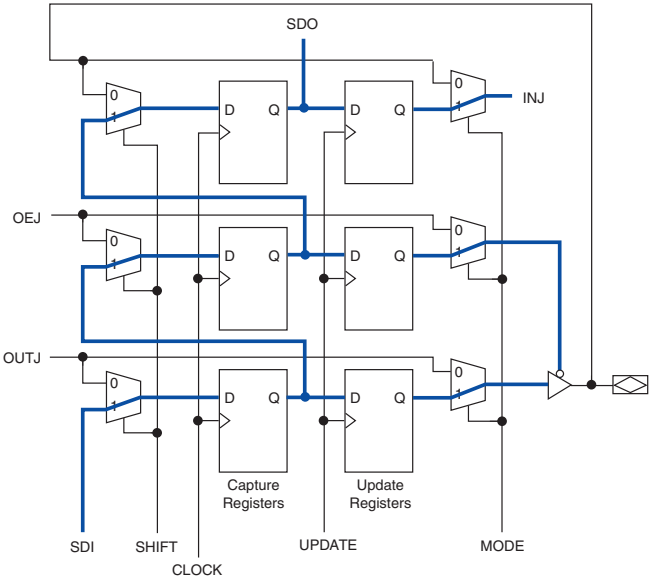
A "1" in the OEJ update register tri-states the output buffer.



Shift and Update Phases

In the shift phase, the previously captured signals at the pin, OEJ and OUTJ, are shifted out of the boundary-scan register via the TDO pin using CLOCK. As data is shifted out, the patterns for the next test can be shifted in via the TDI pin.

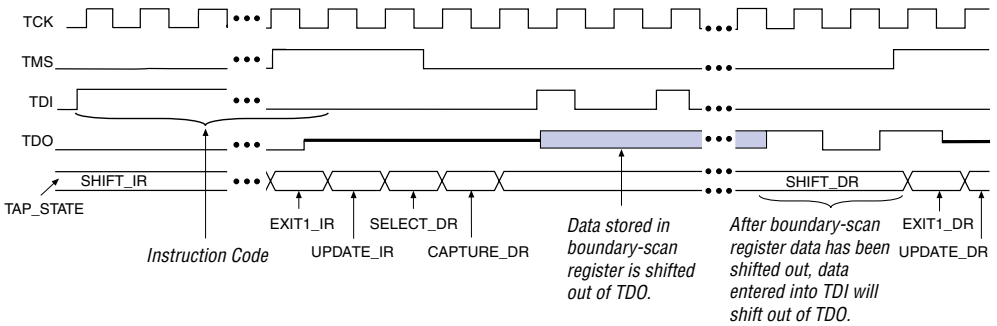
In the update phase, data is transferred from the capture registers to the update registers using the UPDATE clock. The update registers then drive the PIN_IN, INJ, and allow the I/O pin to tri-state or drive a signal out.



EXTEST selects data differently than SAMPLE/PRELOAD. EXTEST chooses data from the update registers as the source of the output and output enable signals. Once the EXTEST instruction code is entered, the multiplexers select the update register data. Thus, data stored in these registers from a previous EXTEST or SAMPLE/PRELOAD test cycle can be forced onto the pin signals. In the capture phase, the results of this test data are stored in the capture registers and then shifted out of TDO during the shift phase. New test data can then be stored in the update registers during the update phase.

The EXTEST waveform diagram in Figure 15–11 resembles the SAMPLE/PRELOAD waveform diagram, except for the instruction code. The data shifted out of TDO consists of the data that was present in the capture registers after the capture phase. New test data shifted into the TDI pin appears at the TDO pin after being clocked through the entire boundary-scan register.

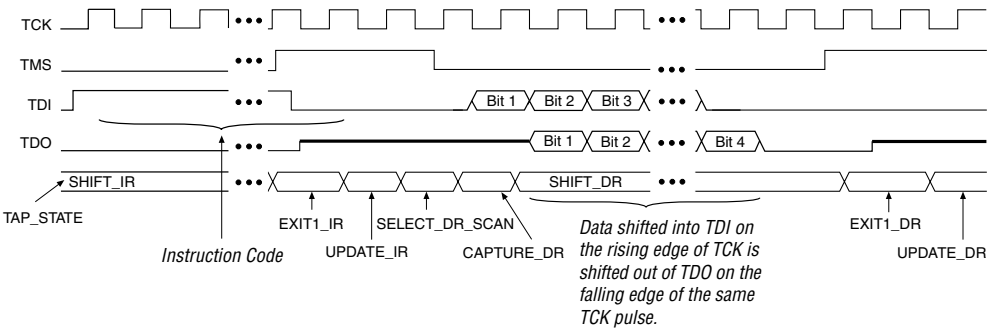
Figure 15–11. EXTEST Shift Data Register Waveforms



BYPASS Instruction Mode

The BYPASS mode is activated when an instruction code of all ones is loaded in the instruction register. The waveforms in Figure 15–12 show how scan data passes through a device once the TAP controller is in the SHIFT_DR state. In this state, data signals are clocked into the bypass register from TDI on the rising edge of TCK and out of TDO on the falling edge of the same clock pulse.

Figure 15–12. BYPASS Shift Data Register Waveforms



IDCODE Instruction Mode

The IDCODE instruction mode is used to identify the devices in an IEEE Std. 1149.1 chain. When IDCODE is selected, the device identification register is loaded with the 32-bit vendor-defined identification code. The device ID register is connected between the TDI and TDO ports, and the device IDCODE is shifted out.

For more information on the IDCODE for Stratix II and Stratix II GX devices refer to the *Configuration & Testing* chapter in volume 1 of the *Stratix II Device Handbook*, or the *Configuration & Testing* chapter in volume 1 of the *Stratix II GX Device Handbook*.

USERCODE Instruction Mode

The USERCODE instruction mode is used to examine the user electronic signature (UES) within the devices along an IEEE Std. 1149.1 chain. When this instruction is selected, the device identification register is connected between the TDI and TDO ports. The user-defined UES is shifted into the device ID register in parallel from the 32-bit USERCODE register. The UES is then shifted out through the device ID register.

The UES value is not user defined until after the device is configured. Before configuration, the UES value is set to the default value.

CLAMP Instruction Mode

The CLAMP instruction mode is used to allow the state of the signals driven from the pins to be determined from the boundary-scan register while the bypass register is selected as the serial path between the TDI and TDO ports. The state of all signals driven from the pins are completely defined by the data held in the boundary-scan register.

If you are testing after configuring the device, the programmable weak pull-up resister or the bus hold feature overrides the CLAMP value (the value stored in the update register of the boundary-scan cell) at the pin.

HIGHZ Instruction Mode

The HIGHZ instruction mode sets all of the user I/O pins to an inactive drive state. These pins are tri-stated until a new JTAG instruction is executed. When this instruction is loaded into the instruction register, the bypass register is connected between the TDI and TDO ports.

If you are testing after configuring the device, the programmable weak pull-up resistor or the bus hold feature overrides the HIGHZ value at the pin.

I/O Voltage Support in JTAG Chain

The JTAG chain supports several devices. However, you should use caution if the chain contains devices that have different V_{CCIO} levels. The output voltage level of the TDO pin must meet the specifications of the TDI pin it drives. The TDI pin is powered by V_{CCPD} (3.3 V). For Stratix II and Stratix II GX devices, the V_{CCIO} power supply of bank 4 powers the TDO pin. Table 15–3 shows board design recommendations to ensure proper JTAG chain operation.

You can interface the TDI and TDO lines of the devices that have different V_{CCIO} levels by inserting a level shifter between the devices. If possible, you should build the JTAG chain in such a way that a device with a higher V_{CCIO} level drives to a device with an equal or lower V_{CCIO} level. This way, a level shifter is used only to shift the TDO level to a level acceptable to the JTAG tester. Figure 15–13 shows the JTAG chain of mixed voltages and how a level shifter is inserted in the chain.

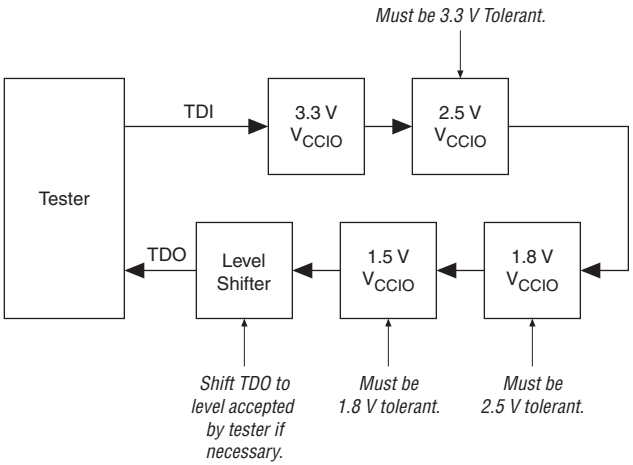
Table 15–3. Supported TDO/TDI Voltage Combinations

Device	TDI Input Buffer Power	Stratix II and Stratix II GX TDO V_{CCIO} Voltage Level in I/O Bank 4			
		V_{CCIO} = 3.3 V	V_{CCIO} = 2.5 V	V_{CCIO} = 1.8 V	V_{CCIO} = 1.5 V
Stratix II and Stratix II GX	Always V_{CCPD} (3.3V)	✓ (1)	✓ (2)	✓ (3)	level shifter required
Non-Stratix II	VCC = 3.3 V	✓ (1)	✓ (2)	✓ (3)	level shifter required
	VCC = 2.5 V	✓ (1), (4)	✓ (2)	✓ (3)	level shifter required
	VCC = 1.8 V	✓ (1), (4)	✓ (2), (5)	✓	level shifter required
	VCC = 1.5 V	✓ (1), (4)	✓ (2), (5)	✓ (6)	✓

Notes to Table 15–3:

(1) The TDO output buffer meets V_{OH} (MIN) = 2.4 V.
(2) The TDO output buffer meets V_{OH} (MIN) = 2.0 V.
(3) An external 250-Ω pull-up resistor is not required, but recommended if signal levels on the board are not optimal.
(4) Input buffer must be 3.3-V tolerant.
(5) Input buffer must be 2.5-V tolerant.
(6) Input buffer must be 1.8-V tolerant.

Figure 15–13. JTAG Chain of Mixed Voltages



Using IEEE Std. 1149.1 BST Circuitry

Stratix II and Stratix II GX devices have dedicated JTAG pins and the IEEE Std. 1149.1 BST circuitry is enabled upon device power-up. Not only can you perform BST on Stratix II and Stratix II GX FPGAs before and after, but also during configuration. Stratix II and Stratix II GX FPGAs support the BYPASS, IDCODE and SAMPLE instructions during configuration without interrupting configuration. To send all other JTAG instructions, you must interrupt configuration using the CONFIG_IO instruction.

The CONFIG_IO instruction allows you to configure I/O buffers via the JTAG port, and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix II or the Stratix II GX FPGA or you can wait for the configuration device to complete configuration. Once configuration is interrupted and JTAG BST is complete, you must reconfigure the part via JTAG (PULSE_CONFIG instruction) or by pulsing nCONFIG low.

When you perform JTAG boundary-scan testing before configuration, the nCONFIG pin must be held low.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Stratix II and Stratix II GX devices do not affect JTAG boundary-scan or configuration operations. Toggling these pins does not disrupt BST operation (other than the expected BST behavior).

When you design a board for JTAG configuration of Stratix II or Stratix II GX devices, you need to consider the connections for the dedicated configuration pins.

For more information on using the IEEE Std.1149.1 circuitry for device configuration, refer to the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II Device Handbook* or the *Configuring Stratix II & Stratix II GX Devices* chapter in volume 2 of the *Stratix II GX Device Handbook*.

BST for Configured Devices

For a configured device, the input buffers are turned off by default for I/O pins that are set as output only in the design file. You cannot sample on the configured device output pins with the default BSDL file when the input buffers are turned off. You can set the Quartus II software to always enable the input buffers on a configured device so it behaves the same as an unconfigured device for boundary-scan testing, allowing sample function on output pins in the design. This aspect can cause slight increase in standby current because the unused input buffer is always on. In the Quartus II software, do the following:

1. Choose **Settings** (Assignments menu).

2. Click **Assembler**.

3. Turn on **Always Enable Input Buffers**.

Disabling IEEE Std. 1149.1 BST Circuitry

The IEEE Std. 1149.1 BST circuitry for Stratix II and Stratix II GX devices is enabled upon device power-up. Because the IEEE Std. 1149.1 BST circuitry is used for BST or in-circuit reconfiguration, you must enable the circuitry only at specific times as mentioned in, "Using IEEE Std. 1149.1 BST Circuitry" on page 15–19.

If you are not using the IEEE Std. 1149.1 circuitry in Stratix II or Stratix II GX, then you should permanently disable the circuitry to ensure that you do not inadvertently enable when it is not required.

Table 15–4 shows the pin connections necessary for disabling the IEEE Std. 1149.1 circuitry in Stratix II and Stratix II GX devices.

Table 15–4. Disabling IEEE Std. 1149.1 Circuitry

JTAG Pins *(1)*	Connection for Disabling
TMS	V_{CC}
TCK	GND
TDI	V_{CC}
TDO	Leave open
TRST	GND

Note to Table 15–4:
(1) There is no software option to disable JTAG in Stratix II or Stratix II GX devices. The JTAG pins are dedicated.

Guidelines for IEEE Std. 1149.1 Boundary-Scan Testing

Use the following guidelines when performing boundary-scan testing with IEEE Std. 1149.1 devices:

- If the "10..." pattern does not shift out of the instruction register via the TDO pin during the first clock cycle of the SHIFT_IR state, the TAP controller did not reach the proper state. To solve this problem, try one of the following procedures:
  - Verify that the TAP controller has reached the SHIFT_IR state correctly. To advance the TAP controller to the SHIFT_IR state, return to the RESET state and send the code 01100 to the TMS pin.

- Check the connections to the V_{CC}, GND, JTAG, and dedicated configuration pins on the device.

- Perform a SAMPLE/PRELOAD test cycle prior to the first EXTEST test cycle to ensure that known data is present at the device pins when you enter the EXTEST mode. If the OEJ update register contains a 0, the data in the OUTJ update register is driven out. The state must be known and correct to avoid contention with other devices in the system.
- Do not perform EXTEST testing during ICR. This instruction is supported before or after ICR, but not during ICR. Use the CONFIG_IO instruction to interrupt configuration and then perform testing, or wait for configuration to complete.
- If performing testing before configuration, hold nCONFIG pin low.
- After configuration, any pins in a differential pin pair cannot be tested. Therefore, performing BST after configuration requires editing of BSC group definitions that correspond to these differential pin pairs. The BSC group should be redefined as an internal cell.

Refer to the Boundary-Scan Description Language (BSDL) file for more information on editing.

For more information on boundary scan testing, contact Altera Applications Group.

Boundary-Scan Description Language (BSDL) Support

The Boundary-Scan Description Language (BSDL), a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.1 BST-capable device that can be tested. Test software development systems then use the BSDL files for test generation, analysis, and failure diagnostics.

For more information, or to receive BSDL files for IEEE Std. 1149.1-compliant Stratix II and Stratix II GX devices, visit the Altera web site at www.altera.com.

Conclusion

The IEEE Std. 1149.1 BST circuitry available in Stratix II and Stratix II GX devices provides a cost-effective and efficient way to test systems that contain devices with tight lead spacing. Circuit boards with Altera and other IEEE Std. 1149.1-compliant devices can use the EXTEST, SAMPLE/PRELOAD, and BYPASS modes to create serial patterns that internally test the pin connections between devices and check device operation.

References

Bleeker, H., P. van den Eijnden, and F. de Jong. *Boundary-Scan Test: A Practical Approach*. Eindhoven, The Netherlands: Kluwer Academic Publishers, 1993.

Institute of Electrical and Electronics Engineers, Inc. *IEEE Standard Test Access Port and Boundary-Scan Architecture* (IEEE Std 1149.1-2001). New York: Institute of Electrical and Electronics Engineers, Inc., 2001.

Maunder, C. M., and R. E. Tulloss. *The Test Access Port and Boundary-Scan Architecture*. Los Alamitos: IEEE Computer Society Press, 1990.

Referenced Documents

This chapter references the following documents:

- *Configuration & Testing* chapter in volume 1 of the *Stratix II Device Handbook*
- *Configuration & Testing* chapter in volume 1 of the *Stratix II GX Device Handbook*
- *Configuring Stratix II & Stratix II GX Devices chapter in volume 2 of the Stratix II Device Handbook*
- *Configuring Stratix II & Stratix II GX Devices chapter in volume 2 of the Stratix II GX Device Handbook*

Document Revision History

Table 15–5 shows the revision history for this chapter.

Table 15–5. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2007, v3.3	Added "Referenced Documents" section.	—
	Minor text edits.	—
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 14. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
February 2007 v3.2	Added the "Document Revision History" section to this chapter.	—
April 2006, v3.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—

Table 15–5. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
No change	Formerly chapter 13. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
October 2005 v3.0	Added chapter to the *Stratix II GX Device Handbook*.	—

Section VII. PCB Layout Guidelines

This section provides information for board layout designers to successfully layout their boards for Stratix® II GX devices. These chapters contain the required PCB layout guidelines and package specifications.

This section contains the following chapters:

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

16. Package Information for Stratix II & Stratix II GX Devices

SII52010-4.3

Introduction

This chapter provides package information for Altera® Stratix® II and Stratix II GX devices, including:

- Device and package cross reference
- Thermal resistance values
- Package outlines

Tables 16–1 and 16–2 show which Altera Stratix II and Stratix II GX devices, respectively, are available in FineLine BGA® (FBGA) packages.

Table 16–1. Stratix II Devices in FBGA Packages

Device	Package	Pins
EP2S15	Flip-chip FBGA	484
	Flip-chip FBGA	672
EP2S30	Flip-chip FBGA	484
	Flip-chip FBGA	672
EP2S60	Flip-chip FBGA	484
	Flip-chip FBGA	672
	Flip-chip FBGA	1,020
EP2S90	Flip-chip FBGA	484
	Flip-chip FBGA	780
	Flip-chip FBGA	1,020
	Flip-chip FBGA	1,508
EP2S130	Flip-chip FBGA	780
	Flip-chip FBGA	1,020
	Flip-chip FBGA	1,508
EP2S180	Flip-chip FBGA	1,020
	Flip-chip FBGA	1,508

Table 16–2. Stratix II GX Devices in FBGA Packages		
Device	**Package**	**Pins**
EP2SGX30	Flip-chip FBGA	780
EP2SGX60	Flip-chip FBGA	780
	Flip-chip FBGA	1,152
EP2SGX90	Flip-chip FBGA	1,152
	Flip-chip FBGA	1,508
EP2SGX130	Flip-chip FBGA	1,508

Thermal Resistance

Thermal resistance values for Stratix II devices are provided for a board that meets JDEC specifications and for a typical board. The following values are provided:

- θ_{JA} (°C/W) still air—Junction-to-ambient thermal resistance with no air flow when a heat sink is not used.
- θ_{JA} (°C/W) 100 ft./min.—Junction-to-ambient thermal resistance with 100 ft./min. airflow when a heat sink is not used.
- θ_{JA} (°C/W) 200 ft./min.—Junction-to-ambient thermal resistance with 200 ft./min. airflow when a heat sink is not used.
- θ_{JA} (°C/W) 400 ft./min.—Junction-to-ambient thermal resistance with 400 ft./min. airflow when a heat sink is not used.
- θ_{JC}—Junction-to-case thermal resistance for device.
- θ_{JB}—Junction-to-board thermal resistance for device.

Tables 16–3 provides θ_{JA} (junction-to-ambient thermal resistance), θ_{JC} (junction-to-case thermal resistance), and θ_{JB} (junction-to-board thermal resistance) values for Stratix II devices on a board meeting JEDEC specifications for thermal resistance calculation. The JEDEC board specifications require two signal and two power/ground planes and are available at **www.jedec.org**.

Table 16–3. Stratix II Device Thermal Resistance for Boards Meeting JEDEC Specifications (Part 1 of 2)								
Device	**Pin Count**	**Package**	**θ_{JA} (° C/W) Still Air**	**θ_{JA} (° C/W) 100 ft./min.**	**θ_{JA} (° C/W) 200 ft./min.**	**θ_{JA} (° C/W) 400 ft./min.**	**θ_{JC} (° C/W)**	**θ_{JB} (° C/W)**
EP2S15	484	FBGA	13.1	11.1	9.6	8.3	0.36	4.19
	672	FBGA	12.2	10.2	8.8	7.6	0.36	4.09
EP2S30	484	FBGA	12.6	10.6	9.1	7.9	0.21	3.72
	672	FBGA	11.7	9.7	8.3	7.1	0.21	3.35

Table 16–3. Stratix II Device Thermal Resistance for Boards Meeting JEDEC Specifications (Part 2 of 2)

Device	Pin Count	Package	θ_{JA} (° C/W) Still Air	θ_{JA} (° C/W) 100 ft./min.	θ_{JA} (° C/W) 200 ft./min.	θ_{JA} (° C/W) 400 ft./min.	θ_{JC} (° C/W)	θ_{JB} (° C/W)
EP2S60	484	FBGA	12.3	10.3	8.8	7.5	0.13	3.38
	672	FBGA	11.4	9.4	7.8	6.7	0.13	2.95
	1,020	FBGA	10.4	8.4	7.0	5.9	0.13	2.67
EP2S90	484	Hybrid FBGA	12.0	9.9	8.3	7.1	0.07	3.73
	780	FBGA	10.8	8.8	7.3	6.1	0.09	2.59
	1,020	FBGA	9.2	8.2	6.8	5.7	0.10	2.41
	1,508	FBGA	9.3	7.4	6.1	5.0	0.10	2.24
EP2S130	780	FBGA	10.1	8.7	7.2	6.0	0.07	2.44
	1,020	FBGA	9.5	8.1	6.7	5.5	0.07	2.24
	1,508	FBGA	8.6	7.3	6.0	4.8	0.07	2.08
EP2S180	1,020	FBGA	9.0	7.9	6.5	5.4	0.05	2.10
	1,508	FBGA	8.1	7.1	5.8	4.7	0.05	1.94

Table 16–4 provides θ_{JA} (junction-to-ambient thermal resistance), θ_{JC} (junction-to-case thermal resistance), and θ_{JB} (junction-to-board thermal resistance) values for Stratix II devices on a board with the information shown in Table 16–5.

Table 16–4. Stratix II Device Thermal Resistance for Typical Board (Part 1 of 2)

Device	Pin Count	Package	θ_{JA} (° C/W) Still Air	θ_{JA} (° C/W) 100 ft./min.	θ_{JA} (° C/W) 200 ft./min.	θ_{JA} (° C/W) 400 ft./min.	θ_{JC} (° C/W)	θ_{JB} (° C/W)
EP2S15	484	FBGA	12.6	9.9	8.1	6.7	0.36	2.48
	672	FBGA	11.4	8.8	7.2	5.9	0.36	2.41
EP2S30	484	FBGA	12.3	9.6	7.8	6.4	0.21	2.02
	672	FBGA	11.1	8.5	6.9	5.6	0.21	1.95
EP2S60	484	FBGA	12.1	9.4	7.6	6.3	0.13	1.74
	672	FBGA	10.9	8.3	6.6	5.4	0.13	1.56
	1,020	FBGA	9.6	7.1	5.6	4.5	0.13	1.33
EP2S90	484	Hybrid FBGA	11.2	8.9	7.2	5.9	0.07	2.48
	780	FBGA	10.0	7.6	6.1	4.9	0.09	1.22
	1,020	FBGA	9.2	6.9	5.5	4.4	0.10	1.16
	1,508	FBGA	8.2	6.0	4.7	3.7	0.10	1.15

Table 16–4. Stratix II Device Thermal Resistance for Typical Board (Part 2 of 2)

Device	Pin Count	Package	θ_{JA} (° C/W) Still Air	θ_{JA} (° C/W) 100 ft./min.	θ_{JA} (° C/W) 200 ft./min.	θ_{JA} (° C/W) 400 ft./min.	θ_{JC} (° C/W)	θ_{JB} (° C/W)
EP2S130	780	FBGA	9.3	7.5	6.0	4.8	0.07	1.12
	1,020	FBGA	8.5	6.8	5.3	4.2	0.07	1.03
	1,508	FBGA	7.5	5.8	4.6	3.6	0.07	1.02
EP2S180	1,020	FBGA	8.0	6.7	5.3	4.2	0.05	0.93
	1,508	FBGA	7.1	5.7	4.5	3.5	0.05	0.91

Table 16–5. Board Specifications *Notes (1), (2)*

Pin Count	Package	Signal Layers	Power/Ground Layers	Size (mm)
1,508	FBGA	12	12	100 × 100
1,020	FBGA	10	10	93 × 93
780	FBGA	9	9	89 × 89
672	FBGA	8	8	87 × 87
484	FBGA	7	7	83 × 83

Notes to Table 16–5:
(1) Power layer Cu thickness 35 um, Cu 90%.
(2) Signal layer Cu thickness 17 um, Cu 15%.

Table 16–6 provides θ_{JA} (junction-to-ambient thermal resistance) and θ_{JC} (junction-to-case thermal resistance) values for Stratix II devices.

Table 16–6. Stratix II GX Device Thermal Resistance

Device	Pin Count	Package	θ_{JA} (° C/W) Still Air	θ_{JA} (° C/W) 100 ft./min.	θ_{JA} (° C/W) 200 ft./min.	θ_{JA} (° C/W) 400 ft./min.	θ_{JC} (° C/W)
EP2SGX30	780	FBGA	11.1	8.6	7.2	6.0	0.24
EP2SGX60	780	FBGA	10.9	8.4	6.9	5.8	0.15
	1,152	FBGA	9.9	7.5	6.1	5.0	0.15
EP2SGX90	1,152	FBGA	9.6	7.3	5.9	4.9	0.11
	1,508	FBGA	9.0	6.7	5.4	4.4	0.11
EP2SGX130	1,508	FBGA	8.3	6.6	5.3	4.3	0.10

Package Outlines

The package outlines are listed in order of ascending pin count. Altera package outlines meet the requirements of *JEDEC Publication No. 95.*

484-Pin FBGA - Flip Chip

- All dimensions and tolerances conform to ASME Y14.5M – 1994.
- Controlling dimension is in millimeters.
- Pin A1 may be indicated by an ID dot, or a special feature, in its proximity on the package surface.

Tables 16–7 and 16–8 show the package information and package outline figure references, respectively, for the 484-pin FBGA packaging.

Table 16–7. 484-Pin FBGA Package Information

Description	Specification
Ordering code reference	F
Package acronym	FBGA
Substrate material	BT
Solder ball composition	Regular: 63Sn:37Pb (Typ.) Pb-free: Sn:3Ag:0.5Cu (Typ.)
JEDEC outline reference	MS-034 variation: AAJ-1
Maximum lead coplanarity	0.008 inches (0.20 mm)
Weight	5.8 g
Moisture sensitivity level	Printed on moisture barrier bag

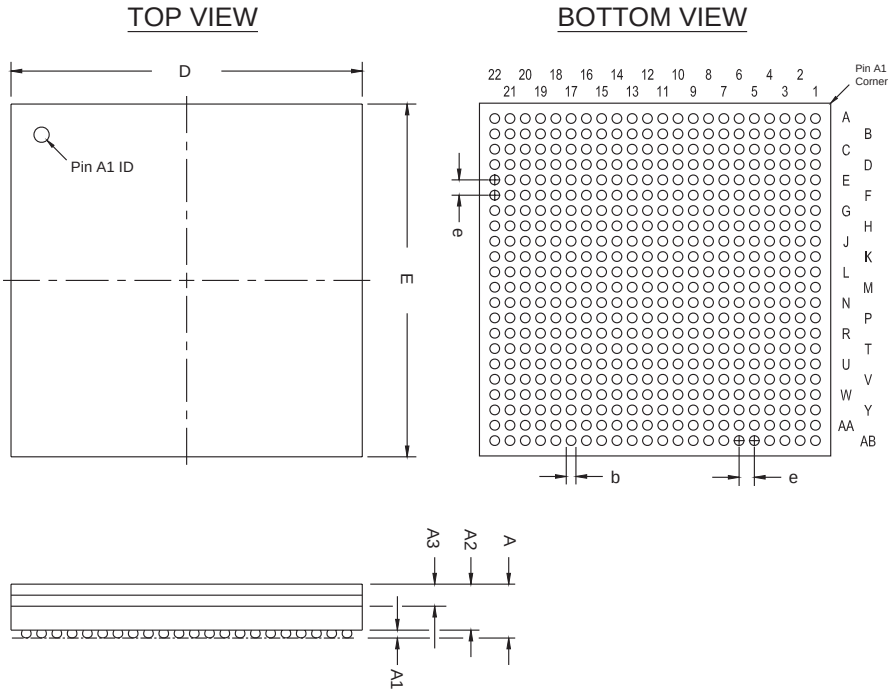
Table 16–8. 484-Pin FBGA Package Outline Dimensions (Part 1 of 2)

Symbol	Millimeter		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	23.00 BSC		
E	23.00 BSC		

Table 16–8. 484-Pin FBGA Package Outline Dimensions (Part 2 of 2)			
Symbol	Millimeter		
	Min.	Nom.	Max.
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 16–1 shows a package outline for the 484-pin FineLine BGA packaging.

Figure 16–1. 484-Pin FBGA Package Outline



672-Pin FBGA - Flip Chip

- All dimensions and tolerances conform to ASME Y14.5M - 1994.
- Controlling dimension is in millimeters.
- Pin A1 may be indicated by an ID dot, or a special feature, in its proximity on package surface.

Tables 16–9 and 16–10 show the package information and package outline figure references, respectively, for the 672-pin FBGA packaging.

Table 16–9. 672-Pin FBGA Package Information

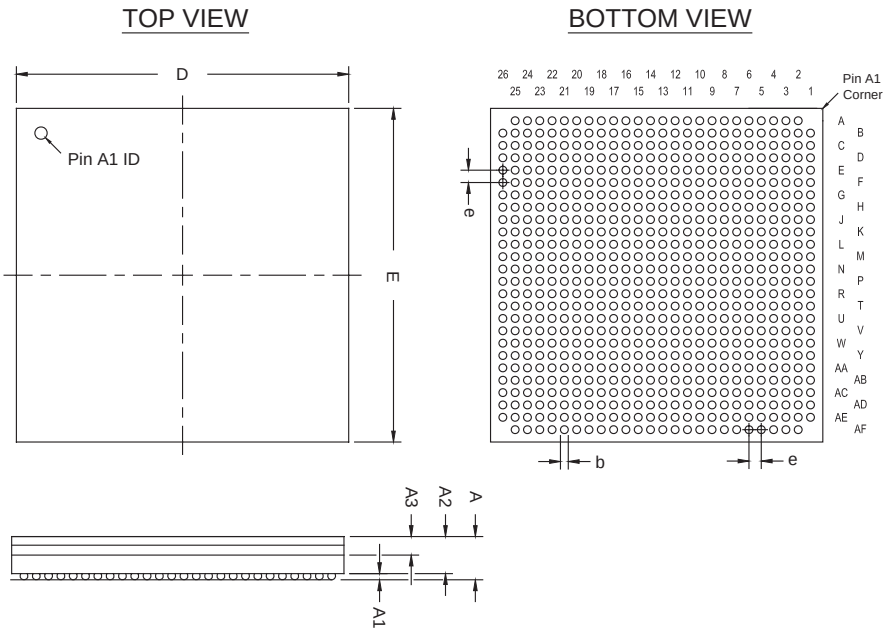
Description	Specification
Ordering code reference	F
Package acronym	FBGA
Substrate material	BT
Solder ball composition	Regular: 63Sn:37Pb (Typ.) Pb-free: Sn:3Ag:0.5Cu (Typ.)
JEDEC Outline Reference	MS-034 Variation: AAL-1
Maximum Lead coplanarity	0.008 inches (0.20 mm)
Weight	7.7 g
Moisture Sensitivity level	Printed on moisture barrier bag

Table 16–10. 672-Pin FBGA Package Outline Dimensions

Symbol	Millimeters		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	27.00 BSC		
E	27.00 BSC		
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 16–2 shows a package outline for the 672-pin FineLine BGA packaging.

Figure 16–2. 672-Pin FBGA Package Outline



780-Pin FBGA - Flip Chip

- All dimensions and tolerances conform to ASME Y14.5M - 1994.
- Controlling dimension is in millimeters.
- Pin A1 may be indicated by an ID dot, or a special feature, in its proximity on package surface.

Tables 16–11 and 16–12 show the package information and package outline figure references, respectively, for the 780-pin FBGA packaging.

Table 16–11. 780-Pin FBGA Package Information

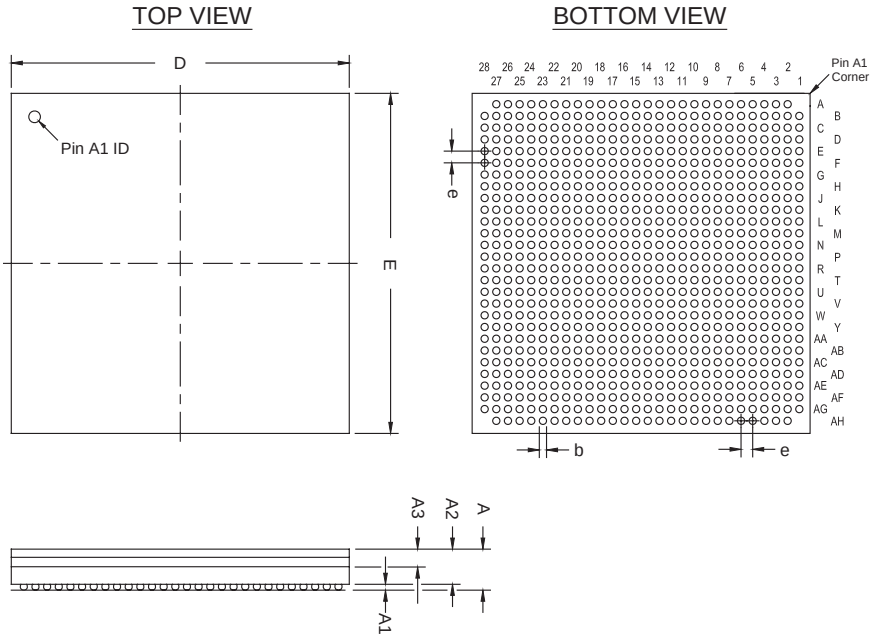
Description	Specification
Ordering code reference	F
Package acronym	FBGA
Substrate material	BT
Solder ball composition	Regular: 63Sn:37Pb (Typ.) Pb-free: Sn:3Ag:0.5Cu (Typ.)
JEDEC outline reference	MS-034 variation: AAM-1
Maximum lead coplanarity	0.008 inches (0.20 mm)
Weight	8.9 g
Moisture Sensitivity Level	Printed on moisture barrier bag

Table 16–12. 780-Pin FBGA Package Outline Dimensions

Symbol	Millimeters		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	29.00 BSC		
E	29.00 BSC		
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 16–3 shows a package outline for the 780-pin FineLine BGA packaging.

Figure 16–3. 780-Pin FBGA Package Outline



1,020-Pin FBGA - Flip Chip

- All dimensions and tolerances conform to ASME Y14.5M - 1994.
- Controlling dimension is in millimeters.
- Pin A1 may be indicated by an ID dot, or a special feature, in its proximity on package surface.

Tables 16–13 and 16–14 show the package information and package outline figure references, respectively, for the 1,020-pin FBGA packaging.

Table 16–13. 1,020-Pin FBGA Package Information

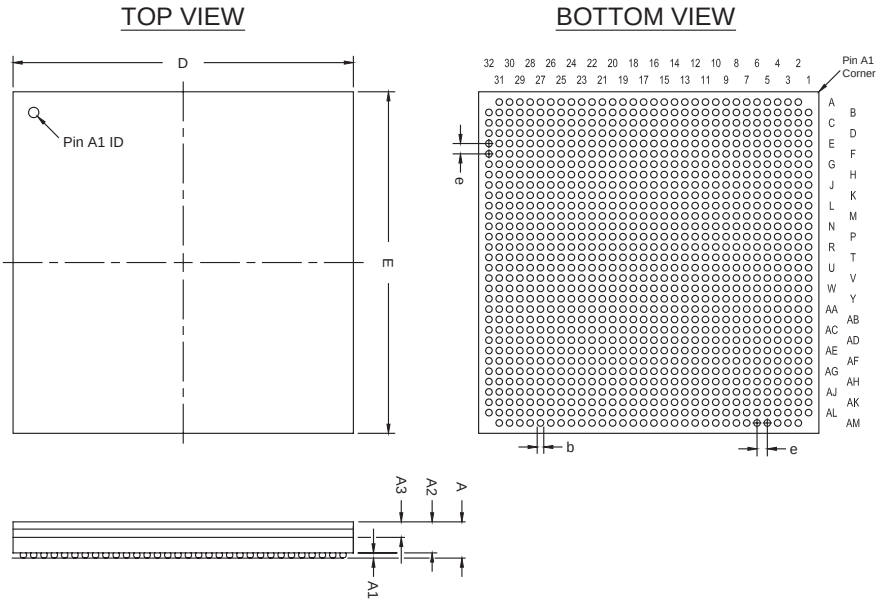
Description	Specification
Ordering code reference	F
Package acronym	FBGA
Substrate material	BT
Solder ball composition	Regular: 63Sn:37Pb (Typ.) Pb-free: Sn:3Ag:0.5Cu (Typ.)
JEDEC outline reference	MS-034 variation: AAP-1
Maximum lead coplanarity	0.008 inches (0.20 mm)
Weight	11.5 g
Moisture sensitivity level	Printed on moisture barrier bag

Table 16–14. 1,020-Pin FBGA Package Outline Dimensions

Symbol	Millimeters		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	33.00 BSC		
E	33.00 BSC		
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 16–4 shows a package outline for the 1,020-pin FineLine BGA packaging.

Figure 16–4. 1,020-Pin FBGA Package Outline



1,152-Pin FBGA - Flip Chip

- All dimensions and tolerances conform to ASME Y14.5M - 1994.
- Controlling dimension is in millimeters.
- Pin A1 may be indicated by an ID dot, or a special feature, in its proximity on package surface.

Tables 16–15 and 16–16 show the package information and package outline figure references, respectively, for the 1,152-pin FBGA packaging.

Table 16–15. 1,152-Pin FBGA Package Information

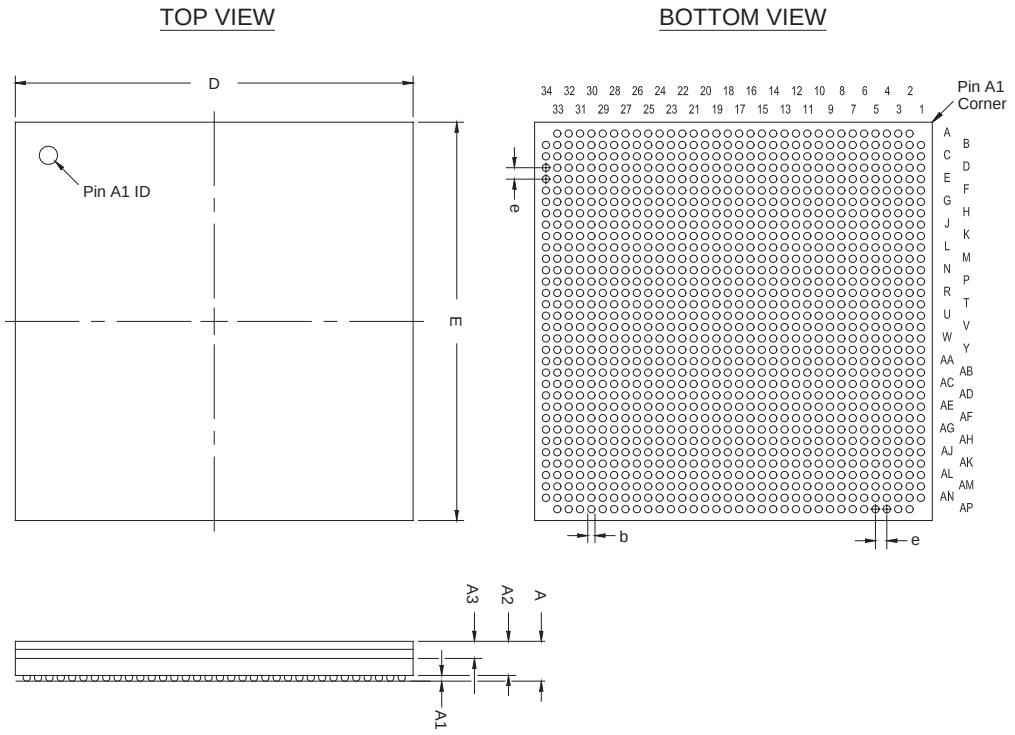
Description	Specification
Ordering code reference	F
Package acronym	FBGA
Substrate material	BT
Solder ball composition	Regular: 63Sn:37Pb (Typ.) Pb-free: Sn:3Ag:0.5Cu (Typ.)
JEDEC outline reference	MS-034 variation: AAR-1
Maximum lead coplanarity	0.008 inches (0.20 mm)
Weight	12.0 g
Moisture sensitivity level	Printed on moisture barrier bag

Table 16–16. 1,152-Pin FBGA Package Outline Dimensions

Symbol	Millimeters		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	35.00 BSC		
E	35.00 BSC		
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 16–5 shows a package outline for the 1,152-pin FineLine BGA packaging.

Figure 16–5. 1,152-Pin FBGA Package Outline



1,508-Pin FBGA - Flip Chip

- All dimensions and tolerances conform to ASME Y14.5M - 1994.
- Controlling dimension is in millimeters.
- Pin A1 may be indicated by an ID dot, or a special feature, in its proximity on package surface.

Tables 16–17 and 16–18 show the package information and package outline figure references, respectively, for the 1,508-pin FBGA packaging.

Table 16–17. 1,508-Pin FBGA Package Information

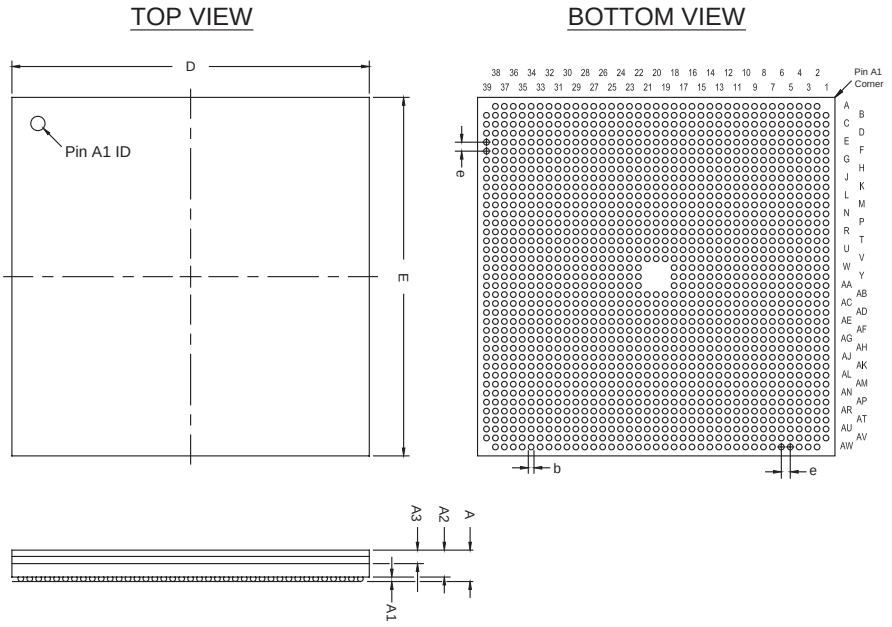
Description	Specification
Ordering code reference	F
Package acronym	FBGA
Substrate material	BT
Solder ball composition	Regular: 63Sn:37Pb (Typ.) Pb-free: Sn:3Ag:0.5Cu (Typ.)
JEDEC outline reference	MS-034 Variation: AAU-1
Maximum lead coplanarity	0.008 inches (0.20 mm)
Weight	14.6 g
Moisture sensitivity level	Printed on moisture barrier bag

Table 16–18. 1,508-Pin FBGA Package Outline Dimensions

Symbol	Millimeters		
	Min.	Nom.	Max.
A	–	–	3.50
A1	0.30	–	–
A2	0.25	–	3.00
A3	–	–	2.50
D	40.00 BSC		
E	40.00 BSC		
b	0.50	0.60	0.70
e	1.00 BSC		

Figure 16–6 shows a package outline for the 1,508-pin FineLine BGA packaging.

Figure 16–6. 1,508-Pin FBGA Package Outline



Document Revision History

Table 16–19 shows the revision history for this chapter.

Table 16–19. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
No change	For the *Stratix II GX Device Handbook* only: Formerly chapter 15. The chapter number changed due to the addition of the *Stratix II GX Dynamic Reconfiguration* chapter. No content change.	—
May 2007, v4.3	Minor change to Table 16–3.	—
February 2007 v4.2	Added the "Document Revision History" section to this chapter.	—
No change	Formerly chapter 14. Chapter number change only due to chapter addition to Section I in February 2006; no content change.	—
December 2005, v4.1	Chapter updated as part of the *Stratix II Device Handbook* update.	—
October 2005 v4.0	Added chapter to the *Stratix II GX Device Handbook*.	—