## Intel - EP4CE55F29C6 Datasheet





Welcome to <u>E-XFL.COM</u>

#### Understanding <u>Embedded - FPGAs (Field</u> <u>Programmable Gate Array)</u>

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

#### **Applications of Embedded - FPGAs**

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

#### Details

Product Status	Active
Number of LABs/CLBs	3491
Number of Logic Elements/Cells	55856
Total RAM Bits	2396160
Number of I/O	374
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	780-BGA
Supplier Device Package	780-FBGA (29x29)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce55f29c6

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

- Cyclone IV GX devices offer up to eight high-speed transceivers that provide:
  - Data rates up to 3.125 Gbps
  - 8B/10B encoder/decoder
  - 8-bit or 10-bit physical media attachment (PMA) to physical coding sublayer (PCS) interface
  - Byte serializer / deserializer (SERDES)
  - Word aligner
  - Rate matching FIFO
  - TX bit slipper for Common Public Radio Interface (CPRI)
  - Electrical idle
  - Dynamic channel reconfiguration allowing you to change data rates and protocols on-the-fly
  - Static equalization and pre-emphasis for superior signal integrity
  - 150 mW per channel power consumption
  - Flexible clocking structure to support multiple protocols in a single transceiver block
- Cyclone IV GX devices offer dedicated hard IP for PCI Express (PIPE) (PCIe) Gen 1:
  - ×1, ×2, and ×4 lane configurations
  - End-point and root-port configurations
  - Up to 256-byte payload
  - One virtual channel
  - 2 KB retry buffer
  - 4 KB receiver (Rx) buffer
- Cyclone IV GX devices offer a wide range of protocol support:
  - PCIe (PIPE) Gen 1 ×1, ×2, and ×4 (2.5 Gbps)
  - Gigabit Ethernet (1.25 Gbps)
  - CPRI (up to 3.072 Gbps)
  - XAUI (3.125 Gbps)
  - Triple rate serial digital interface (SDI) (up to 2.97 Gbps)
  - Serial RapidIO (3.125 Gbps)
  - Basic mode (up to 3.125 Gbps)
  - V-by-One (up to 3.0 Gbps)
  - DisplayPort (2.7 Gbps)
  - Serial Advanced Technology Attachment (SATA) (up to 3.0 Gbps)
  - OBSAI (up to 3.072 Gbps)

# **Cyclone IV Device Family Architecture**

This section describes Cyclone IV device architecture and contains the following topics:

- "FPGA Core Fabric"
- "I/O Features"
- "Clock Management"
- "External Memory Interfaces"
- "Configuration"
- "High-Speed Transceivers (Cyclone IV GX Devices Only)"
- "Hard IP for PCI Express (Cyclone IV GX Devices Only)"

# **FPGA Core Fabric**

Cyclone IV devices leverage the same core fabric as the very successful Cyclone series devices. The fabric consists of LEs, made of 4-input look up tables (LUTs), memory blocks, and multipliers.

Each Cyclone IV device M9K memory block provides 9 Kbits of embedded SRAM memory. You can configure the M9K blocks as single port, simple dual port, or true dual port RAM, as well as FIFO buffers or ROM. They can also be configured to implement any of the data widths in Table 1–7.

## Table 1–7. M9K Block Data Widths for Cyclone IV Device Family

Mode	Data Width Configurations
Single port or simple dual port	×1, ×2, ×4, ×8/9, ×16/18, and ×32/36
True dual port	×1, ×2, ×4, ×8/9, and ×16/18

The multiplier architecture in Cyclone IV devices is the same as in the existing Cyclone series devices. The embedded multiplier blocks can implement an 18 × 18 or two 9 × 9 multipliers in a single block. Altera offers a complete suite of DSP IP including finite impulse response (FIR), fast Fourier transform (FFT), and numerically controlled oscillator (NCO) functions for use with the multiplier blocks. The Quartus<sup>®</sup> II design software's DSP Builder tool integrates MathWorks Simulink and MATLAB design environments for a streamlined DSP design flow.



For more information, refer to the *Logic Elements and Logic Array Blocks in Cyclone IV Devices*, *Memory Blocks in Cyclone IV Devices*, and *Embedded Multipliers in Cyclone IV Devices* chapters.

# **Arithmetic Mode**

Arithmetic mode is ideal for implementing adders, counters, accumulators, and comparators. An LE in arithmetic mode implements a 2-bit full adder and basic carry chain (Figure 2–3). LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output. Register feedback and register packing are supported when LEs are used in arithmetic mode.

Figure 2–3 shows LEs in arithmetic mode.





The Quartus II Compiler automatically creates carry chain logic during design processing. You can also manually create the carry chain logic 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 LEs by automatically linking LABs in the same column. For enhanced fitting, a long carry chain runs vertically, which allows fast horizontal connections to M9K memory blocks or embedded multipliers through direct link interconnects. For example, if a design has a long carry chain in an LAB column next to a column of M9K memory blocks, any LE output can feed an adjacent M9K memory block through the direct link interconnect. If the carry chains run horizontally, any LAB which is not next to the column of M9K memory blocks uses other row or column interconnects to drive a M9K memory block. A carry chain continues as far as a full column. Figure 3–3 and Figure 3–4 show the address clock enable waveform during read and write cycles, respectively.



Figure 3–3. Cyclone IV Devices Address Clock Enable During Read Cycle Waveform

Figure 3-4. Cyclone IV Devices Address Clock Enable During Write Cycle Waveform



# **Mixed-Width Support**

M9K memory blocks support mixed data widths. When using simple dual-port, true dual-port, or FIFO modes, mixed width support allows you to read and write different data widths to an M9K memory block. For more information about the different widths supported per memory mode, refer to "Memory Modes" on page 3–7.

Figure 3–7 shows a timing waveform for read and write operations in single-port mode with unregistered outputs. Registering the outputs of the RAM simply delays the q output by one clock cycle.



Figure 3–7. Cyclone IV Devices Single-Port Mode Timing Waveform

## **Simple Dual-Port Mode**

Simple dual-port mode supports simultaneous read and write operations to different locations. Figure 3–8 shows the simple dual-port memory configuration.

Figure 3–8. Cyclone IV Devices Simple Dual-Port Memory (1)



### Note to Figure 3-8:

(1) Simple dual-port RAM supports input or output clock mode in addition to the read or write clock mode shown.

Cyclone IV devices M9K memory blocks support mixed-width configurations, allowing different read and write port widths. Table 3–3 lists mixed-width configurations.

 Table 3–3.
 Cyclone IV Devices M9K Block Mixed-Width Configurations (Simple Dual-Port Mode)
 (Part 1 of 2)

Bood Bort					Write Port				
neau ruit	8192 × 1	4096 × 2	2048 × 4	1024 × 8	512 × 16	256 × 32	1024 × 9	512 × 18	256 × 36
8192 × 1	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	—	—	_
4096 × 2	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	—	—	—
2048 × 4	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	—	—	—
1024 × 8	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	—	—	—

Figure 6–14 shows a typical BLVDS topology with multiple transmitter and receiver pairs.



Figure 6-14. BLVDS Topology with Cyclone IV Devices Transmitters and Receivers

The BLVDS I/O standard is supported on the top, bottom, and right I/O banks of Cyclone IV devices. The BLVDS transmitter uses two single-ended output buffers with the second output buffer programmed as inverted, while the BLVDS receiver uses a true LVDS input buffer. The transmitter and receiver share the same pins. An output-enabled (OE) signal is required to tristate the output buffers when the LVDS input buffer receives a signal.

• For more information, refer to the *Cyclone IV Device Datasheet* chapter.

## **Designing with BLVDS**

The BLVDS bidirectional communication requires termination at both ends of the bus in BLVDS. The termination resistor ( $R_T$ ) must match the bus differential impedance, which in turn depends on the loading on the bus. Increasing the load decreases the bus differential impedance. With termination at both ends of the bus, termination is not required between the two signals at the input buffer. A single series resistor ( $R_S$ ) is required at the output buffer to match the output buffer impedance to the transmission line impedance. However, this series resistor affects the voltage swing at the input buffer. The maximum data rate achievable depends on many factors.

- Altera recommends that you perform simulation using the IBIS model while considering factors such as bus loading, termination values, and output and input buffer location on the bus to ensure that the required performance is achieved.
- **\*** For more information about BLVDS interface support in Altera devices, refer to *AN 522: Implementing Bus LVDS Interface in Supported Altera Device Families.*

JTAG instructions have precedence over any other configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration in Cyclone IV devices during PS configuration, PS configuration terminates and JTAG configuration begins. If the MSEL pins are set to AS mode, the Cyclone IV device does not output a DCLK signal when JTAG configuration takes place.

The four required pins for a device operating in JTAG mode are TDI, TDO, TMS, and TCK. All the JTAG input pins are powered by the  $V_{CCIO}$  pin and support the LVTTL I/O standard only. All user I/O pins are tri-stated during JTAG configuration. Table 8-14 explains the function of each JTAG pin.

**Pin Name Pin Type** Description Serial input pin for instructions as well as test and programming data. Data shifts in on the Test data rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry is TDI disabled by connecting this pin to  $V_{CC}$ . TDI pin has weak internal pull-up resistors (typically 25 input kΩ). Serial data output pin for instructions as well as test and programming data. Data shifts out on Test data the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the TDO output JTAG interface is not required on the board, the JTAG circuitry is disabled by leaving this pin unconnected. Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions in the state machine occur on the rising edge of TCK. Therefore, Test mode TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. TMS select If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to  $V_{CC}$ . TMS pin has weak internal pull-up resistors (typically 25 k $\Omega$ ). The clock input to the BST circuitry. Some operations occur at the rising edge, while others Test clock occur at the falling edge. If the JTAG interface is not required on the board, the JTAG circuitry TCK input

Table 8–14. Dedicated JTAG Pins

You can download data to the device through the USB-Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV download cable, or the EthernetBlaster communications cable during JTAG configuration. Configuring devices with a cable is similar to programming devices in-system. Figure 8-23 and Figure 8-24 show the JTAG configuration of a single Cyclone IV device.

is disabled by connecting this pin to GND. The TCK pin has an internal weak pull-down resistor.

synchronization state machine mode. In bit-slip mode, you can dynamically enable the receiver bit reversal using the rx\_revbitorderwa port. When enabled, the 8-bit or 10-bit data D[7..0] or D[9..0] at the output of the word aligner is rewired to D[0..7] or D[0..9] respectively. Figure 1–20 shows the receiver bit reversal feature.





#### Note to Figure 1-20:

(1) The rx\_revbitordwa port is dynamic and is only available when the word aligner is configured in bit-slip mode.

- When using the receiver bit reversal feature to receive MSB-to-LSB transmission, reversal of the word alignment pattern is required.
- Receiver bit-slip indicator—provides the number of bits slipped in the word aligner for synchronization with rx\_bitslipboundaryselectout signal. For usage details, refer to "Receive Bit-Slip Indication" on page 1–76.

## **Deskew FIF0**

This module is only available when used for the XAUI protocol and is used to align all four channels to meet the maximum skew requirement of 40 UI (12.8 ns) as seen at the receiver of the four lanes. The deskew operation is compliant to the PCS deskew state machine diagram specified in clause 48 of the IEEE P802.3ae specification.

The deskew circuitry consists of a 16-word deep deskew FIFO in each of the four channels, and control logics in the central control unit of the transceiver block that controls the deskew FIFO write and read operations in each channel.

For details about the deskew FIFO operations for channel deskewing, refer to "XAUI Mode" on page 1–67.

## **Bonded Channel Configuration**

In bonded channel configuration, the low-speed clock for the bonded channels share a common bonded clock path that reduces clock skew between the bonded channels. The phase compensation FIFOs in bonded channels share a set of pointers and control logic that results in equal FIFO latency between the bonded channels. These features collectively result in lower channel-to-channel skew when implementing multi-channel serial interface in bonded channel configuration.

In a transceiver block, the high-speed clock for each bonded channels is distributed independently from one of the two multipurpose PLLs directly adjacent to the block. The low-speed clock for bonded channels is distributed from a common bonded clock path that selects from one of the two multipurpose PLLs directly adjacent to the block. Transceiver channels for devices in F484 and larger packages support additional clocking flexibility for ×2 bonded channels. In these packages, the ×2 bonded channels support high-speed and low-speed bonded clock distribution from PLLs beyond the two multipurpose PLLs directly adjacent to the block. Table 1–10 lists the high- and low-speed clock sources for the bonded channels.

Dookogo	Transceiver	Dandad Obannala	High- and Low-Speed Clocks Source		
гаскауе	Block	Bonueu Channeis	Option 1	Option 2	
F324 and smaller	GXBL0	×2 in channels 0, 1 ×4 in all channels	MPLL_1	MPLL_2	
F484 and larger	GXBL0	×2 in channels 0, 1	MPLL_5/ GPLL_1	MPLL_6	
		×4 in all channels	MPLL_5	MPLL_6	
	GXBL1 (1)	×2 in channels 0, 1	MPLL_7/ MPLL_6	MPLL_8	
		×4 in all channels	MPLL_7	MPLL_8	

Table 1–10. High- and Low-Speed Clock Sources for Bonded Channels in Bonded Channel Configuration

### Note to Table 1-10:

(1) GXBL1 is not available for transceivers in F484 package.

When implementing ×2 bonded channel configuration in a transceiver block, remaining channels 2 and 3 are available to implement other non-bonded channel configuration.

Figure 1–55 shows the transceiver channel datapath and clocking when configured in GIGE mode.





### Notes to Figure 1-55:

- (1) Low-speed recovered clock.
- (2) High-speed recovered clock.

(3) Optional rx\_recovclkout port from CDR low-speed recovered clock is available for applications such as Synchronous Ethernet.

Figure 1–57 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 /K28.5/ code group received at an odd code group boundary in cycle n + 3 takes the receiver synchronization state machine in Loss-of-Sync state. The first synchronization ordered-set /K28.5/Dx.y/ in cycles n + 3 and n + 4 is discounted and three additional ordered sets are required for successful synchronization.





## **Running Disparity Preservation with Idle Ordered Set**

During idle ordered sets transmission in GIGE mode, the transmitter ensures a negative running disparity at the end of an idle ordered set. Any /Dx.y/, except for /D21.5/ (part of /C1/ ordered set) or /D2.2/ (part of /C2/ ordered set) following a /K28.5/ is automatically replaced with either of the following:

- A /D5.6/ (/I1/ ordered set) if the running disparity before /K28.5/ is positive
- A /D16.2/ (/I2/ ordered set) if the running disparity before /K28.5/ is negative

## **Lane Synchronization**

In GIGE mode, the word aligner is configured in automatic synchronization state machine mode that complies with the IEEE P802.3ae standard. A synchronization ordered set is a /K28.5/ code group followed by an odd number of valid /Dx.y/ code groups. Table 1–19 lists the synchronization state machine parameters that implements the GbE-compliant synchronization.

Parameter	Value
Number of valid synchronization ordered sets received to achieve synchronization	3
Number of erroneous code groups received to lose synchronization	4
Number of continuous good code groups received to reduce the error count by one	4

#### Note to Table 1-19:

(1) The word aligner supports 7-bit and 10-bit pattern lengths in GIGE mode.

### Table 1–21. XGMII Character to PCS Code Groups Mapping (Part 2 of 2)

XGMII TXC <sup>(1)</sup>	XGMII TXD <sup>(2), (3)</sup>	PCS Code Group	Description	
1	Any other value	K30.7	Invalid XGMII character	

#### Notes to Table 1-21:

(2) Equivalent to 8-bit input data to 8B/10B encoder.

(3) The values in XGMII TXD column are in hexadecimal.

8B/10B decoder in the receiver datapath maps received PCS code groups into specific 8-bit XGMII codes as listed in Table 1–22.

XGMII RXC <sup>(1)</sup>	XGMII RXD <sup>(2), (3)</sup>	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	ldle 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

#### Notes to Table 1-22:

(1) Equivalent to rx\_ctrlenable port.

(2) Equivalent to 8-bit input data to 8B/10B encoder.

(3) The values in XGMII RXD column are in hexadecimal.

## **Channel Deskewing**

The deskew FIFO in each of the four lanes expects to receive /A/ code group simultaneously on all four channels during the inter-packet gap, as required by XAUI protocol. The skew introduced in the physical medium and the receiver channels might cause the /A/ code group to be received misaligned with respect to each other.

The deskew FIFO works to align the /A/ code group across the four channels, which operation is compliant to the PCS deskew state machine diagram specified in clause 48 of the IEEE P802.3ae specification. The deskew operation begins after link synchronization is achieved on all four channels as indicated by the word aligner in each channel. The following are the deskew FIFO operations:

- Until the first /A/ code group is received, the deskew FIFO read and write pointers in each channel are not incremented.
- After the first /A/ code group is received, the write pointer starts incrementing for each word received but the read pointer is frozen.
- When all the four channels received the /A/ code group within 10 recovered clock cycles of each other, the read pointer of all four deskew FIFOs is released simultaneously, aligning the /A/ code group of all four channels in a column.

<sup>(1)</sup> Equivalent to tx\_ctrlenable port.

- 4. For the receiver operation, after deassertion of busy signal, wait for two parallel clock cycles to deassert the rx analogreset signal.
- 5. Wait for the rx\_freqlocked signal from each channel to go high. The rx\_freqlocked signal of each channel may go high at different times (indicated by the slashed pattern at marker 7).
- 6. In a bonded channel group, when the rx\_freqlocked signals of all the channels has gone high, from that point onwards, wait for at least t<sub>LTD\_Auto</sub> time for the receiver parallel clock to be stable, then deassert the rx\_digitalreset signal (marker 8). At this point, all the receivers are ready for data traffic.

### **Receiver and Transmitter Channel—Receiver CDR in Manual Lock Mode**

This configuration contains both a transmitter and receiver channel. When the receiver CDR is in manual lock mode, use the reset sequence shown in Figure 2–5.





#### Notes to Figure 2-5:

- (1) For t<sub>LTD Manual</sub> duration, refer to the Cyclone IV Device Datasheet chapter.
- (2) The number of rx\_locktorefclk [n] and rx\_locktodata [n] signals depend on the number of channels configured. n=number of channels.
- (3) For  $t_{LTR\_LTD\_Manual}$  duration, refer to the *Cyclone IV Device Datasheet* chapter.
- (4) The busy signal is asserted and deasserted only during initial power up when offset cancellation occurs. In subsequent reset sequences, the busy signal is asserted and deasserted only if there is a read or write operation to the ALTGX\_RECONFIG megafunction.

- 4. Wait for at least t<sub>LTR\_LTD\_Manual</sub> (the time between markers 6 and 7), then deassert the rx\_locktorefclk signal. At the same time, assert the rx\_locktodata signal (marker 7). At this point, the receiver CDR enters lock-to-data mode and the receiver CDR starts locking to the received data.
- 5. Deassert rx\_digitalreset at least  $t_{LTD\_Manual}$  (the time between markers 7 and 8) after asserting the rx\_locktodata signal. At this point, the transmitter and receiver are ready for data traffic.

## **Reset Sequence in Loss of Link Conditions**

Loss of link can occur due to loss of local reference clock source or loss of the link due to an unplugged cable. Other adverse conditions like loss of power could also cause the loss of signal from the other device or link partner.

## Loss of Local REFCLK or Other Reference Clock Condition

Should local reference clock input become disabled or unstable, take the following steps:

- 1. Monitor pll\_locked signal. Pll\_locked is de-asserted if local reference clock source becomes unavailable.
- 2. Pll\_locked assertion indicates a stable reference clock because TX PLL locks to the incoming clock. You can follow appropriate reset sequence provided in the device handbook, starting from pll\_locked assertion.

### Loss of Link Due To Unplugged Cable or Far End Shut-off Condition

Use one or more of the following methods to identify whether link partner is alive:

- Signal detect is available in PCIe and Basic modes. You can monitor rx\_signaldetect signal as loss of link indicator. rx\_signaldetect is asserted when the link partner comes back up.
- You can implement a ppm detector in device core for modes that do not have signal detect to monitor the link. Ppm detector helps in identifying whether the link is alive.
- Data corruption or RX phase comp FIFO overflow or underflow condition in user logic may indicate a loss of link condition.

Apply the following reset sequences when loss of link is detected:

- For Automatic CDR lock mode:
  - a. Monitor rx\_freqlocked signal. Loss of link causes rx\_freqlocked to be deasserted when CDR moves back to lock-to-data (LTD) mode.
  - b. Assert rx\_digitalreset.
  - c. rx\_freqlocked toggles over time when CDR switches between lock-to-reference (LTR) and LTD modes.
  - d. If rx\_freqlocked goes low at any point, re-assert rx\_digitalreset.
  - e. If data corruption or RX phase comp FIFO overflow or underflow condition is observed in user logic, assert rx\_digitalreset for 2 parallel clock cycles, then de-assert the signal.

## **PCIe Initialization/Compliance Phase**

After the device is powered up, a PCIe-compliant device goes through the compliance phase during initialization. The rx\_digitalreset signal must be deasserted during this compliance phase to achieve transitions on the pipephydonestatus signal, as expected by the link layer. The rx\_digitalreset signal is deasserted based on the assertion of the rx\_freqlocked signal.

During the initialization/compliance phase, do not use the rx\_freqlocked signal to trigger a deassertion of the rx\_digitalreset signal. Instead, perform the following reset sequence:

- 1. After power up, assert pll\_areset for a minimum period of 1  $\mu$ s (the time between markers 1 and 2). Keep the tx\_digitalreset, rx\_analogreset, and rx\_digitalreset signals asserted during this time period. After you deassert the pll\_areset signal, the multipurpose PLL starts locking to the input reference clock.
- 2. After the multipurpose PLL locks, as indicated by the pll\_locked signal going high (marker 3), deassert tx\_digitalreset. For a receiver operation, after deassertion of busy signal, wait for two parallel clock cycles to deassert the rx\_analogreset signal. After rx\_analogreset is deasserted, the receiver CDR starts locking to the receiver input reference clock.
- 3. Deassert both the rx\_analogreset signal (marker 6) and rx\_digitalreset signal (marker 7) together, as indicated in Figure 2–10. After deasserting rx\_digitalreset, the pipephydonestatus signal transitions from the transceiver channel to indicate the status to the link layer. Depending on its status, pipephydonestatus helps with the continuation of the compliance phase. After successful completion of this phase, the device enters into the normal operation phase.

# **PCIe Normal Phase**

For the normal PCIe phase:

- 1. After completion of the Initialization/Compliance phase, during the normal operation phase at the Gen1 data rate, when the rx\_freqlocked signal is deasserted (marker 9 in Figure 2–10).
- 2. Wait for the rx\_freqlocked signal to go high again. In this phase, the received data is valid (not electrical idle) and the receiver CDR locks to the incoming data. Proceed with the reset sequence after assertion of the rx\_freqlocked signal.
- 3. After the rx\_freqlocked signal goes high, wait for at least  $t_{LTD\_Manual}$  before asserting rx\_digitalreset (marker 12 in Figure 2–10) for two parallel receive clock cycles so that the receiver phase compensation FIFO is initialized. For bonded PCIe Gen 1 mode (×2 and ×4), wait for all the rx\_freqlocked signals to go high, then wait for  $t_{LTD\_Manual}$  before asserting rx\_digitalreset for 2 parallel clock cycles.

Figure 3–9 shows the connection for PMA reconfiguration mode.



(1) This block can be reconfigured in PMA reconfiguration mode.

# **Transceiver Channel Reconfiguration Mode**

You can dynamically reconfigure the transceiver channel from an existing functional mode to a different functional mode by selecting the **Channel Reconfiguration** option in ALTGX and ALTGX\_RECONFIG MegaWizards. The blocks that are reconfigured by channel reconfiguration mode are the PCS and RX PMA blocks of a transceiver channel.

For more information about reconfiguring the RX PMA blocks of the transceiver channel using channel reconfiguration mode, you can refer to "Data Rate Reconfiguration Mode Using RX Local Divider" on page 3–26.

In channel reconfiguration, only a write transaction can occur; no read transactions are allowed. You can optionally choose to trigger write\_all once by selecting the continuous write operation in the ALTGX\_RECONFIG MegaWizard Plug-In Manager. The Quartus II software then continuously writes all the words required for reconfiguration.

For channel reconfiguration, **.mif** files are required to dynamically reconfigure the transceivers channels in channel reconfiguration modes. The **.mif** carries the reconfiguration information that will be used to reconfigure the transceivers channel dynamically on-the-fly. The **.mif** contents is generated automatically when you select the **Generate GXB Reconfig MIF** option in the Quartus II software setting. For different **.mif** settings, you need to later reconfigure and recompile the ALTGX MegaWizard to generate the **.mif** based on the required reconfiguration settings.

The dynamic reconfiguration controller can optionally perform a continuos write operation or a regular write operation of the **.mif** contents in terms of word size (16-bit data) to the transceivers channel that is selected for reconfiguration.

## Figure 3–9. ALTGX and ALTGX\_RECONFIG Connection for PMA Reconfiguration Mode

## **Option 1: Share a Single Transmitter Core Clock Between Receivers**

- Enable this option if you want tx\_clkout of the first channel (channel 0) of the transceiver block to provide the read clock to the Receive Phase Compensation FIFOs of the remaining receiver channels in the transceiver block.
- This option is typically enabled when all the channels of a transceiver block are in a Basic or Protocol configuration with rate matching enabled and are reconfigured to another Basic or Protocol configuration with rate matching enabled.

Figure 3–13 shows the sharing of channel 0's tx\_clkout between all four channels of a transceiver block.





This chapter provides additional information about the document and Altera.

# **About this Handbook**

This handbook provides comprehensive information about the Altera<sup>®</sup> Cyclone<sup>®</sup> IV family of devices.

# **How to Contact Altera**

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

Contact <sup>(1)</sup>	<b>Contact Method</b>	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
recinical training	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Nontechnical 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**

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, <b>Save As</b> dialog box. For GUI elements, capitalization matches the GUI.
bold typeIndicates directory names, project names, disk drive names, file names, fiextensions, software utility names, and GUI labels. For example, \qdesign directory, D: drive, and chiptrip.gdf file.	
Italic Type with Initial Capital Letters	Indicate document titles. For example, Stratix IV Design Guidelines.
	Indicates variables. For example, $n + 1$ .
italic type	Variable names are enclosed in angle brackets (< >). For example, <i><file name=""></file></i> and <i><project name="">.pof</project></i> file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, "Typographic Conventions."

# I/O Timing

Use the following methods to determine I/O timing:

- the Excel-based I/O Timing
- the Quartus II timing analyzer

The Excel-based I/O timing provides pin timing performance for each device density and speed grade. The data is typically used prior to designing the FPGA to get a timing budget estimation as part of the link timing analysis. The Quartus II timing analyzer provides a more accurate and precise I/O timing data based on the specifics of the design after place-and-route is complete.

The Excel-based I/O Timing spreadsheet is downloadable from Cyclone IV Devices Literature website.

# Glossary

Table 1–46 lists the glossary for this chapter.

Letter	Term	Definitions		
Α	—	—		
В	—	_		
C	—	_		
D	—	_		
E	—	_		
F	f <sub>HSCLK</sub>	High-speed I/O block: High-speed receiver/transmitter input and output clock frequency.		
C	GCLK	Input pin directly to Global Clock network.		
u	GCLK PLL	Input pin to Global Clock network through the PLL.		
Н	HSIODR	High-speed I/O block: Maximum/minimum LVDS data transfer rate (HSIODR = 1/TUI).		
I	Input Waveforms for the SSTL Differential I/O Standard	Vswing Vswing V <sub>REF</sub> V <sub>IL</sub>		

## Table 1-46. Glossary (Part 1 of 5)

## Table 1-46. Glossary (Part 5 of 5)

Letter	Term	Definitions
V	V <sub>CM(DC)</sub>	DC common mode input voltage.
	V <sub>DIF(AC)</sub>	AC differential input voltage: The minimum AC input differential voltage required for switching.
	V <sub>DIF(DC)</sub>	DC differential input voltage: The minimum DC input differential voltage required for switching.
	V <sub>ICM</sub>	Input common mode voltage: The common mode of the differential signal at the receiver.
	V <sub>ID</sub>	Input differential voltage swing: The difference in voltage between the positive and complementary conductors of a differential transmission at the receiver.
	V <sub>IH</sub>	Voltage input high: The minimum positive voltage applied to the input that is accepted by the device as a logic high.
	V <sub>IH(AC)</sub>	High-level AC input voltage.
	V <sub>IH(DC)</sub>	High-level DC input voltage.
	V <sub>IL</sub>	Voltage input low: The maximum positive voltage applied to the input that is accepted by the device as a logic low.
	V <sub>IL (AC)</sub>	Low-level AC input voltage.
	V <sub>IL (DC)</sub>	Low-level DC input voltage.
	V <sub>IN</sub>	DC input voltage.
	V <sub>OCM</sub>	Output common mode voltage: The common mode of the differential signal at the transmitter.
	V <sub>OD</sub>	Output differential voltage swing: The difference in voltage between the positive and complementary conductors of a differential transmission at the transmitter. $V_{OD} = V_{OH} - V_{OL}$ .
	V <sub>OH</sub>	Voltage output high: The maximum positive voltage from an output that the device considers is accepted as the minimum positive high level.
	V <sub>OL</sub>	Voltage output low: The maximum positive voltage from an output that the device considers is accepted as the maximum positive low level.
	V <sub>OS</sub>	Output offset voltage: $V_{OS} = (V_{OH} + V_{OL}) / 2$ .
	V <sub>OX (AC)</sub>	AC differential output cross point voltage: the voltage at which the differential output signals must cross.
	V <sub>REF</sub>	Reference voltage for the SSTL and HSTL I/O standards.
	V <sub>REF (AC)</sub>	AC input reference voltage for the SSTL and HSTL I/O standards. $V_{REF(AC)} = V_{REF(DC)} + noise$ . The peak-to-peak AC noise on $V_{REF}$ must not exceed 2% of $V_{REF(DC)}$ .
	V <sub>REF (DC)</sub>	DC input reference voltage for the SSTL and HSTL I/O standards.
	V <sub>SWING (AC)</sub>	AC differential input voltage: AC input differential voltage required for switching. For the SSTL differential I/O standard, refer to Input Waveforms.
	V <sub>SWING (DC)</sub>	DC differential input voltage: DC input differential voltage required for switching. For the SSTL differential I/O standard, refer to Input Waveforms.
	V <sub>TT</sub>	Termination voltage for the SSTL and HSTL I/O standards.
	V <sub>X (AC)</sub>	AC differential input cross point voltage: The voltage at which the differential input signals must cross.
W	_	—
X	—	—
Y	—	—
Z		—