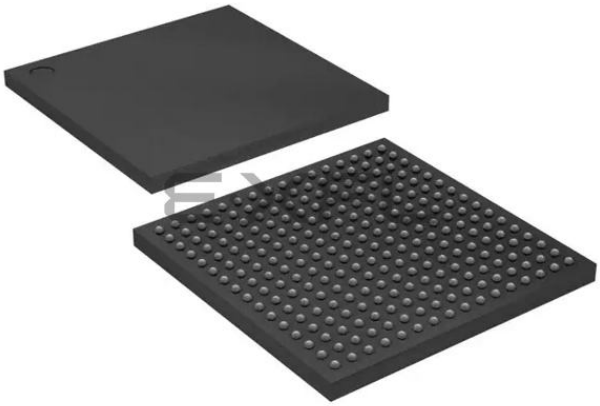


Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

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	392
Number of Logic Elements/Cells	6272
Total RAM Bits	276480
Number of I/O	179
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	256-LBGA
Supplier Device Package	256-FBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce6f17c8

Chapter 9. SEU Mitigation in Cyclone IV Devices

Chapter 10. JTAG Boundary-Scan Testing for Cyclone IV Devices

Visual Cue	Meaning
Courier type	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix `n` denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	The question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	The multimedia icon directs you to a related multimedia presentation.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.

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

Read Port	Write Port								
	8192 × 1	4096 × 2	2048 × 4	1024 × 8	512 × 16	256 × 32	1024 × 9	512 × 18	256 × 36
512 × 16	✓	✓	✓	✓	✓	✓	—	—	—
256 × 32	✓	✓	✓	✓	✓	✓	—	—	—
1024 × 9	—	—	—	—	—	—	✓	✓	✓
512 × 18	—	—	—	—	—	—	✓	✓	✓
256 × 36	—	—	—	—	—	—	✓	✓	✓

In simple dual-port mode, M9K memory blocks support separate `wren` and `rden` signals. You can save power by keeping the `rden` signal low (inactive) when not reading. Read-during-write operations to the same address can either output "Don't Care" data at that location or output "Old Data". To choose the desired behavior, set the **Read-During-Write** option to either **Don't Care** or **Old Data** in the RAM MegaWizard Plug-In Manager in the Quartus II software. For more information about this behavior, refer to "Read-During-Write Operations" on page 3–15.

Figure 3–9 shows the timing waveform for read and write operations in simple dual-port mode with unregistered outputs. Registering the outputs of the RAM simply delays the `q` output by one clock cycle.

Figure 3–9. Cyclone IV Devices Simple Dual-Port Timing Waveform

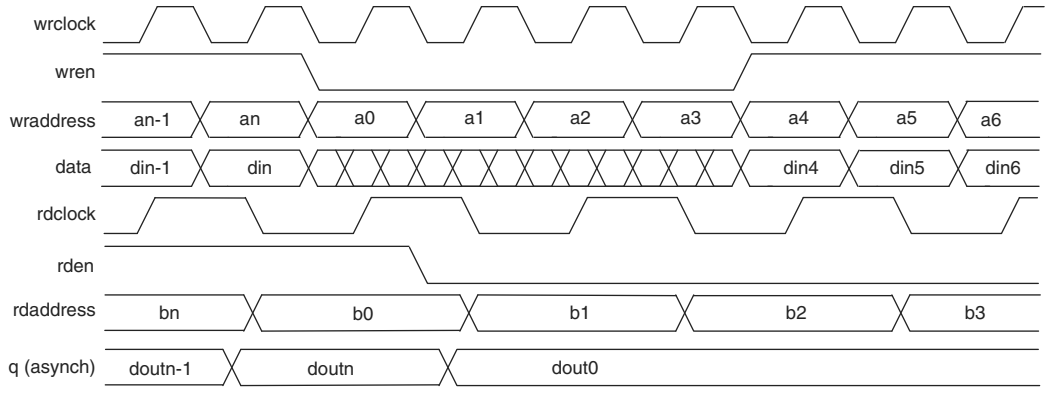


Table 5–2. GCLK Network Connections for EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 Devices (1), (2) (Part 2 of 4)

GCLK Network Clock Sources	GCLK Networks																													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
PLL_3_C0	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	—	—	—	—	✓	—	—	✓	—	✓
PLL_3_C1	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	✓	—	—	✓	—
PLL_3_C2	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—
PLL_3_C3	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—
PLL_3_C4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	✓	—	✓	✓
PLL_4_C0	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	✓	✓	—	—	✓	—	✓	—	—	—	—	—	—
PLL_4_C1	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	✓	—	—	✓	—	—	—	—	—	—	—
PLL_4_C2	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—
PLL_4_C3	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—
PLL_4_C4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	✓	—	✓	✓	—	—	—	—	—	—
PLL_5_C0	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_5_C1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_5_C2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_5_C3	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_5_C4	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_6_C0	✓	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_6_C1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_6_C2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_6_C3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_6_C4	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C0 (3)	—	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C1 (3)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C2 (3)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C3 (3)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL_7_C4 (3)	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Programmable Bandwidth

The PLL bandwidth is the measure of the PLL's ability to track the input clock and its associated jitter. PLLs of Cyclone IV devices provide advanced control of the PLL bandwidth using the programmable characteristics of the PLL loop, including loop filter and charge pump. 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.

Phase Shift Implementation

Phase shift is used to implement a robust solution for clock delays in Cyclone IV devices. Phase shift is implemented with a combination of the VCO phase output and the counter starting time. The VCO phase output and counter starting time are the most accurate methods of inserting delays, because they are based only on counter settings that are independent of process, voltage, and temperature.

You can phase shift the output clocks from the PLLs of Cyclone IV devices in one of two ways:

- Fine resolution using VCO phase taps
- Coarse resolution using counter starting time

Fine resolution phase shifts are implemented by allowing any of the output counters (`C[4..0]`) or the M counter 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.

Equation 5–1 shows the minimum delay time that you can insert using this method.

Equation 5–1. Fine Resolution Phase Shift

$$\Phi_{fine} = \frac{T_{VCO}}{8} = \frac{1}{8f_{VCO}} = \frac{N}{8Mf_{REF}}$$

in which f_{REF} is the input reference clock frequency.

For example, if f_{REF} is 100 MHz, N = 1, and M = 8, then f_{VCO} = 800 MHz, and Φ_{fine} = 156.25 ps. The PLL operating frequency defines this phase shift, a value that depends on reference clock frequency and counter settings.

Coarse resolution phase shifts are implemented by delaying the start of the counters for a predetermined number of counter clocks. Equation 5–2 shows the coarse phase shift.

Equation 5–2. Coarse Resolution Phase Shift

$$\Phi_{coarse} = \frac{C-1}{f_{VCO}} = \frac{(C-1)N}{Mf_{REF}}$$

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.

Document Revision History

Table 5–14 lists the revision history for this chapter.

Table 5–14. Document Revision History

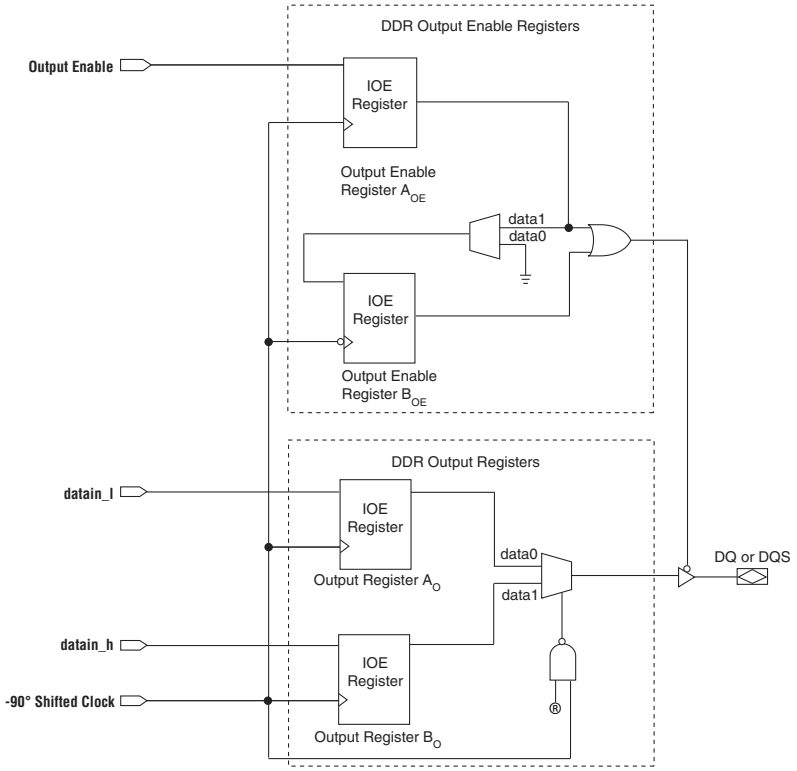
Date	Version	Changes
October 2012	2.4	■ Updated “Manual Override” and “PLL Cascading” sections. ■ Updated Figure 5–9.
November 2011	2.3	■ Updated the “Dynamic Phase Shifting” section. ■ Updated Figure 5–26.
December 2010	2.2	■ Updated for the Quartus II software version 10.1 release. ■ Updated Figure 5–3 and Figure 5–10. ■ Updated “GCLK Network Clock Source Generation”, “PLLs in Cyclone IV Devices”, and “Manual Override” sections. ■ Minor text edits.
July 2010	2.1	■ Updated Figure 5–2, Figure 5–3, Figure 5–4, and Figure 5–10. ■ Updated Table 5–1, Table 5–2, and Table 5–5. ■ Updated “Clock Feedback Modes” section.
February 2010	2.0	■ Added Cyclone IV E devices information for the Quartus II software version 9.1 SP1 release. ■ Updated “Clock Networks” section. ■ Updated Table 5–1 and Table 5–2. ■ Added Table 5–3. ■ Updated Figure 5–2, Figure 5–3, and Figure 5–9. ■ Added Figure 5–4 and Figure 5–10.
November 2009	1.0	Initial release.

DDR Output Registers

A dedicated write DDIO block is implemented in the DDR output and output enable paths.

Figure 7–8 shows how a Cyclone IV dedicated write DDIO block is implemented in the I/O element (IOE) registers.

Figure 7–8. Cyclone IV Dedicated Write DDIO



The two DDR output registers are located in the I/O element (IOE) block. Two serial data streams routed through `datain_l` and `datain_h`, are fed into two registers, `output register Ao` and `output register Bo`, respectively, on the same clock edge. The output from `output register Ao` is captured on the falling edge of the clock, while the output from `output register Bo` is captured on the rising edge of the clock. The registered outputs are multiplexed by the common clock to drive the DDR output pin at twice the data rate.

The DDR output enable path has a similar structure to the DDR output path in the IOE block. The second output enable register provides the write preamble for the `DQS` strobe in DDR external memory interfaces. This active-low output enable register extends the high-impedance state of the pin by half a clock cycle to provide the external memory's `DQS` write preamble time specification.

For more information about Cyclone IV IOE registers, refer to the *Cyclone IV Device I/O Features* chapter.

Configuration Scheme

A configuration scheme with different configuration voltage standards is selected by driving the MSEL pins either high or low, as shown in Table 8–3, Table 8–4, and Table 8–5.

☞ Hardwire the MSEL pins to V_{CCA} or GND without pull-up or pull-down resistors to avoid problems detecting an incorrect configuration scheme. Do not drive the MSEL pins with a microprocessor or another device.

Table 8–3. Configuration Schemes for Cyclone IV GX Devices (EP4CGX15, EP4CGX22, and EP4CGX30 [except for F484 Package])

Configuration Scheme	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) (1)
AS	1	0	1	Fast	3.3
	0	1	1	Fast	3.0, 2.5
	0	0	1	Standard	3.3
	0	1	0	Standard	3.0, 2.5
PS	1	0	0	Fast	3.3, 3.0, 2.5
	1	1	0	Fast	1.8, 1.5
	0	0	0	Standard	3.3, 3.0, 2.5
JTAG-based configuration (2)	(3)	(3)	(3)	—	—

Notes to Table 8–3:

(1) Configuration voltage standard applied to the V_{CCIO} supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to V_{CCA} or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

Table 8–4. Configuration Schemes for Cyclone IV GX Devices (EP4CGX30 [only for F484 package], EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150) (Part 1 of 2)

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) (1)
AS	1	1	0	1	Fast	3.3
	1	0	1	1	Fast	3.0, 2.5
	1	0	0	1	Standard	3.3
	1	0	1	0	Standard	3.0, 2.5
PS	1	1	0	0	Fast	3.3, 3.0, 2.5
	1	1	1	0	Fast	1.8, 1.5
	1	0	0	0	Standard	3.3, 3.0, 2.5
	0	0	0	0	Standard	1.8, 1.5
FPP	0	0	1	1	Fast	3.3, 3.0, 2.5
	0	1	0	0	Fast	1.8, 1.5
	0	0	0	1	Standard	3.3, 3.0, 2.5
	0	0	1	0	Standard	1.8, 1.5

For Cyclone IV E devices, the Quartus II software prohibits you from using the LVDS I/O standard in I/O Bank 1 when the configuration device I/O voltage is not 2.5 V. If you need to assign LVDS I/O standard in I/O Bank 1, navigate to **Assignments**>**Device**>**Settings**>**Device and Pin Option**>**Configuration** to change the Configuration Device I/O voltage to **2.5 V** or **Auto**.

AS Configuration (Serial Configuration Devices)

In the AS configuration scheme, Cyclone IV devices are configured with a serial configuration device. These configuration devices are low-cost devices with non-volatile memories that feature a simple four-pin interface and a small form factor. These features make serial configuration devices the ideal low-cost configuration solution.

For more information about serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Datasheet* in volume 2 of the *Configuration Handbook*.

Serial configuration devices provide a serial interface to access the configuration data. During device configuration, Cyclone IV devices read the configuration data through the serial interface, decompress the 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.

If you want to gain control of the EPCS pins, hold the nCONFIG pin low and pull the nCE pin high to cause the device to reset and tri-state the AS configuration pins.

Single-Device AS Configuration

The four-pin interface of serial configuration devices consists of the following pins:

- Serial clock input (DCLK)
- Serial data output (DATA)
- Active-low chip select (nCS)
- AS data input (ASDI)

FPP Configuration

The FPP configuration in Cyclone IV devices is designed to meet the increasing demand for faster configuration time. Cyclone IV devices are designed with the capability of receiving byte-wide configuration data per clock cycle.

You can perform FPP configuration of Cyclone IV devices with an intelligent host, such as a MAX II device or microprocessor with flash memory. If your system already contains a CFI flash memory, you can use it for the Cyclone IV device configuration storage as well. The MAX II PFL feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control configuration from the flash memory device to the Cyclone IV device.

For more information about the PFL, refer to *AN 386: Using the Parallel Flash Loader with the Quartus II Software*.

FPP configuration is supported in EP4CGX30 (only for F484 package), EP4CGX50, EP4CGX75, EP4CGX110, EP4CGX150, and all Cyclone IV E devices.

The FPP configuration is not supported in E144 package of Cyclone IV E devices.

Cyclone IV devices do not support enhanced configuration devices for FPP configuration.

FPP Configuration Using an External Host

FPP configuration using an external host provides a fast method to configure Cyclone IV devices. In the FPP configuration scheme, you can use an external host device to control the transfer of configuration data from a storage device, such as flash memory, to the target Cyclone IV device. You can store configuration data in an **.rbf**, **.hex**, or **.ttf** format. When using the external host, a design that controls the configuration process, such as fetching the data from flash memory and sending it to

Remote System Upgrade

Cyclone IV devices support remote system upgrade in AS and AP configuration schemes. You can also implement remote system upgrade with advanced Cyclone IV features such as real-time decompression of configuration data in the AS configuration scheme.

☞ Remote system upgrade is not supported in a multi-device configuration chain for any configuration scheme.

Functional Description

The dedicated remote system upgrade circuitry in Cyclone IV devices manages remote configuration and provides error detection, recovery, and status information. A Nios® II processor or a user logic implemented in the Cyclone IV device logic array provides access to the remote configuration data source and an interface to the configuration memory.

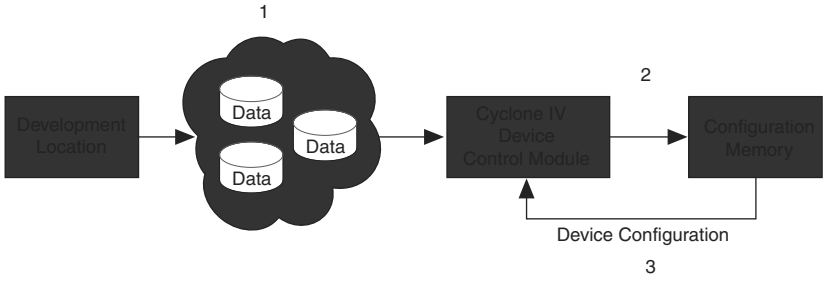
☞ Configuration memory refers to serial configuration devices (EPCS) or supported parallel flash memory, depending on the configuration scheme that is used.

The remote system upgrade process of the Cyclone IV device consists of the following steps:

1. A Nios II processor (or user logic) implemented in the Cyclone IV device 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 II processor (or user logic) writes this new configuration data into a configuration memory.
3. The Nios II processor (or user logic) starts a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any error that might occur during or after the reconfiguration cycle and provides error status information to the user design.

Figure 8–30 shows the steps required for performing remote configuration updates (the numbers in Figure 8–30 coincide with steps 1–3).

Figure 8–30. Functional Diagram of Cyclone IV Device Remote System Upgrade



11. Power Requirements for Cyclone IV Devices

CYIV-51011-1.3

This chapter describes information about external power supply requirements, hot-socketing specifications, power-on reset (POR) requirements, and their implementation in Cyclone IV devices.

This chapter includes the following sections:

External Power Supply Requirements

This section describes the different external power supplies required to power Cyclone IV devices. Table 11–1 and Table 11–2 list the descriptions of external power supply pins for Cyclone IV GX and Cyclone IV E devices, respectively.

For each Altera recommended power supply's operating conditions, refer to the *Cyclone IV Device Datasheet* chapter.

For power supply pin connection guidelines and power regulator sharing, refer to the *Cyclone IV Device Family Pin Connection Guidelines*.

Table 11–1. Power Supply Descriptions for the Cyclone IV GX Devices (Part 1 of 2)

Power Supply Pin	Nominal Voltage Level (V)	Description
VCCINT	1.2	Core voltage, PCI Express (PCIe) hard IP block, and transceiver physical coding sublayer (PCS) power supply
VCCA (1)	2.5	PLL analog power supply
VCCD_PLL	1.2	PLL digital power supply
VCCIO (2)	1.2, 1.5, 1.8, 2.5, 3.0, 3.3	I/O banks power supply
VCC_CLKIN (3), (4)	1.2, 1.5, 1.8, 2.5, 3.0, 3.3	Differential clock input pins power supply
VCCH_GXB	2.5	Transceiver output (TX) buffer power supply
VCCA_GXB	2.5	Transceiver physical medium attachment (PMA) and auxiliary power supply

This chapter provides additional information about the document and Altera.

About this Handbook

This handbook provides comprehensive information about the Altera® Cyclone® 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 (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
Nontechnical support (general)	Email	nacomp@altera.com
(software licensing)	Email	authorization@altera.com

Note to Table:

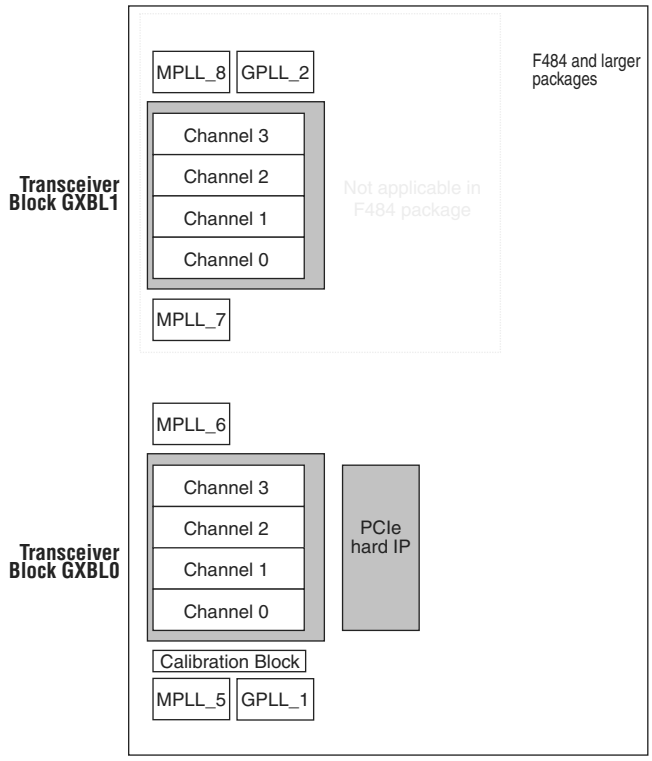
(1) You can also contact your local Altera sales office or sales representative.

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, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>***.pof** 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."

Figure 1–2. F484 and Larger Packages with Transceiver Channels for Cyclone IV GX Devices



For more information about the transceiver architecture, refer to the following sections:

- "Architectural Overview" on page 1–4
- "Transmitter Channel Datapath" on page 1–5
- "Receiver Channel Datapath" on page 1–11
- "Transceiver Clocking Architecture" on page 1–26
- "Transceiver Channel Datapath Clocking" on page 1–29
- "FPGA Fabric-Transceiver Interface Clocking" on page 1–43
- "Calibration Block" on page 1–45
- "PCI-Express Hard IP Block" on page 1–46

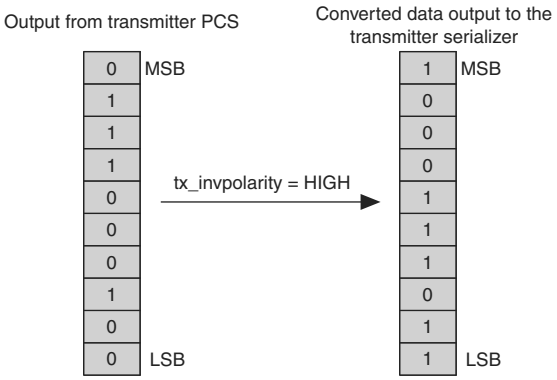
at time n + 2 is encoded as a positive disparity code group. In the same example, the current running disparity at time n + 5 indicates that the K28.5 in time n + 6 should be encoded with a positive disparity. Because tx_forcedisp is high at time n + 6, and tx_dispval is high, the K28.5 at time n + 6 is encoded as a negative disparity code group.

Miscellaneous Transmitter PCS Features

The transmitter PCS supports the following additional features:

- Polarity inversion—corrects accidentally swapped positive and negative signals from the serial differential link during board layout by inverting the polarity of each bit. An optional tx_invpolarity port is available to dynamically invert the polarity of every bit of the 8-bit or 10-bit input data to the serializer in the transmitter datapath. Figure 1–9 shows the transmitter polarity inversion feature.

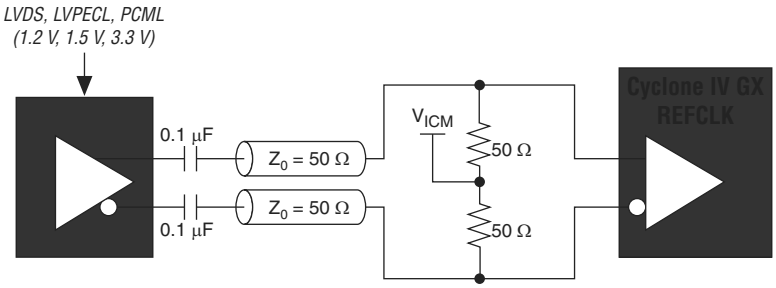
Figure 1–9. Transmitter Polarity Inversion



tx_invpolarity is a dynamic signal and might 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 1–27 shows an example of the termination scheme for AC-coupled connections for REFCLK pins.

Figure 1–27. AC-Coupled Termination Scheme for a Reference Clock

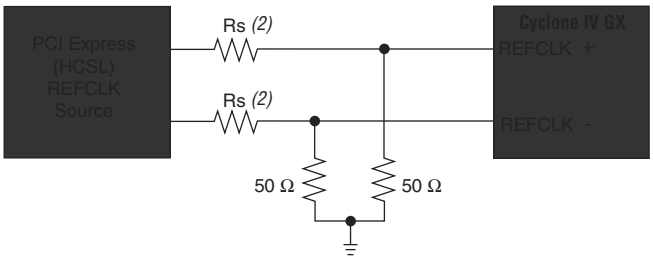


Note to Figure 1–27:

(1) For more information about the V_{ICM} value, refer to the *Cyclone IV Device Datasheet* chapter.

Figure 1–28 shows an example termination scheme for the REFCLK pin when configured as a **HCSL** input.

Figure 1–28. Termination Scheme for a Reference Clock When Configured as HCSL (1)



Notes to Figure 1–28:

(1) No biasing is required if the reference clock signals are generated from a clock source that conforms to the PCIe specification.

(2) Select values as recommended by the PCIe clock source vendor.

Transceiver Channel Datapath Clocking

Channel datapath clocking varies with channel configuration options and PCS configurations. This section describes the clock distribution from the left PLLs for transceiver channels and the datapath clocking in various supported configurations.

Table 1–7 lists the clocks generated by the PLLs for transceiver datapath.

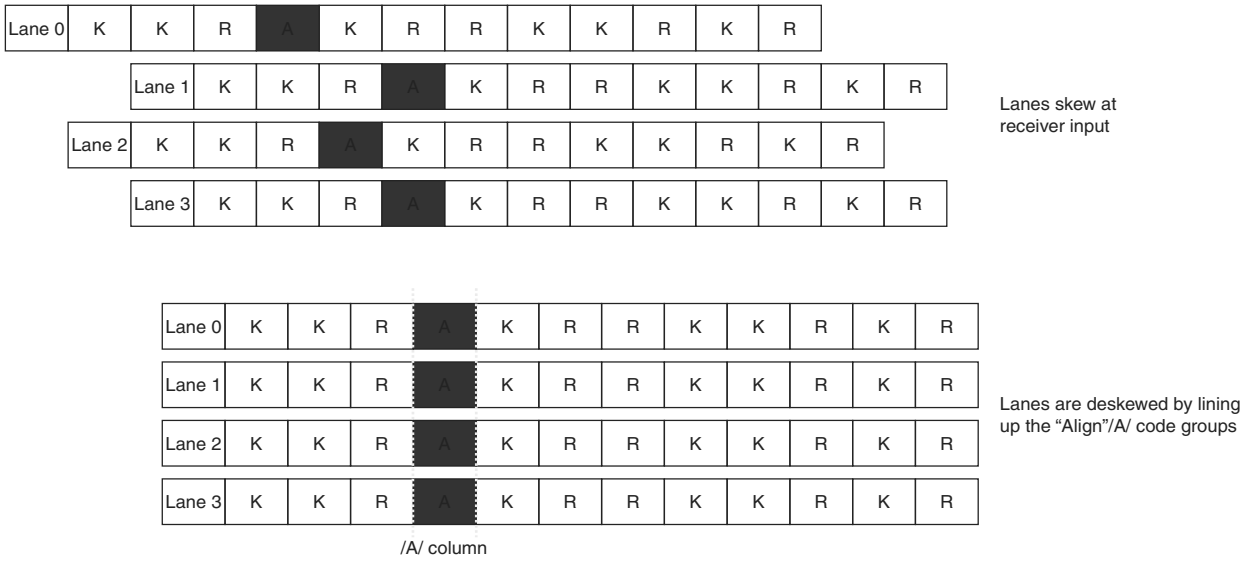
Table 1–7. PLL Clocks for Transceiver Datapath

Clock	Usage
CDR clocks	Receiver CDR unit
High-speed clock	Transmitter serializer block in PMA
Low-speed clock	Transmitter PCS blocks Receiver PCS blocks when rate match FIFO enabled

- Channel alignment is acquired if three additional aligned ||A|| columns are observed at the output of the deskew FIFOs of the four channels after alignment of the first ||A|| column.
- Channel alignment is indicated by the assertion of `rx_channelaligned` signal.
- After acquiring channel alignment, if four misaligned ||A|| columns are seen at the output of the deskew FIFOs in all four channels with no aligned ||A|| columns in between, the `rx_channelaligned` signal is deasserted, indicating loss of channel alignment.

Figure 1–65 shows lane skew at the receiver input and how the deskew FIFO uses the /A/ code group to align the channels.

Figure 1–65. Deskew FIFO–Lane Skew at the Receiver Input



Lane Synchronization

In XAUI mode, the word aligner is configured in automatic synchronization state machine mode that is compliant to the PCS synchronization state diagram specified in clause 48 of the IEEE P802.3ae specification. Table 1–23 lists the synchronization state machine parameters that implements the lane synchronization in XAUI mode.

Table 1–23. Synchronization State Machine Parameters (1)

Parameter	Value
Number of valid synchronization (/K28.5/) code groups received to achieve synchronization	4
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 1–23:

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

Table 3–5 describes the rx_dataoutfull[31..0] FPGA fabric-Transceiver channel interface signals.

Table 3–5. rx_dataoutfull[31..0] FPGA Fabric-Transceiver Channel Interface Signal Descriptions (Part 1 of 3)

FPGA Fabric-Transceiver Channel Interface Description	Receive Signal Description (Based on Cyclone IV GX Supported FPGA Fabric-Transceiver Channel Interface Widths)
8-bit FPGA fabric-Transceiver Channel 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 (rx_errdetect)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: Disparity error status signal (rx_disperr)
	rx_dataoutfull[12]: Pattern detect status signal (rx_patterndetect)
	rx_dataoutfull[13]: Rate Match FIFO deletion status indicator (rx_rmfifodatadeleted) in non-PCI Express (PIPE) functional modes.
	rx_dataoutfull[14]: Rate Match FIFO insertion status indicator (rx_rmfifodatainserted) in non-PCI Express (PIPE) functional modes.
	rx_dataoutfull[14:13]: PCI Express (PIPE) functional mode (rx_pipestatus)
	rx_dataoutfull[15]: 8B/10B running disparity indicator (rx_runningdisp)
10-bit FPGA fabric-Transceiver Channel Interface	rx_dataoutfull[9:0]: 10-bit un-encoded data (rx_dataout)
	rx_dataoutfull[10]: rx_syncstatus
	rx_dataoutfull[11]: 8B/10B disparity error indicator (rx_disperr)
	rx_dataoutfull[12]: rx_patterndetect
	rx_dataoutfull[13]: Rate Match FIFO deletion status indicator (rx_rmfifodatadeleted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[14]: Rate Match FIFO insertion status indicator (rx_rmfifodatainserted) in non-PCI Express (PIPE) functional modes
	rx_dataoutfull[15]: 8B/10B running disparity indicator (rx_runningdisp)

Table 1–44 and Table 1–45 list the IOE programmable delay for Cyclone IV GX devices.

Table 1–44. IOE Programmable Delay on Column Pins for Cyclone IV GX Devices *(1), (2)*

Parameter	Paths Affected	Number of Settings	Min Offset	Max Offset						Unit
				Fast Corner		Slow Corner				
				C6	I7	C6	C7	C8	I7	
Input delay from pin to internal cells	Pad to I/O dataout to core	7	0	1.313	1.209	2.184	2.336	2.451	2.387	ns
Input delay from pin to input register	Pad to I/O input register	8	0	1.312	1.208	2.200	2.399	2.554	2.446	ns
Delay from output register to output pin	I/O output register to pad	2	0	0.438	0.404	0.751	0.825	0.886	0.839	ns
Input delay from dual-purpose clock pin to fan-out destinations	Pad to global clock network	12	0	0.713	0.682	1.228	1.41	1.566	1.424	ns

Notes to Table 1–44:

(1) The incremental values for the settings are generally linear. For exact values of each setting, use the latest version of the Quartus II software.

(2) The minimum and maximum offset timing numbers are in reference to setting **0** as available in the Quartus II software.

Table 1–45. IOE Programmable Delay on Row Pins for Cyclone IV GX Devices *(1), (2)*

Parameter	Paths Affected	Number of Settings	Min Offset	Max Offset						Unit
				Fast Corner		Slow Corner				
				C6	I7	C6	C7	C8	I7	
Input delay from pin to internal cells	Pad to I/O dataout to core	7	0	1.314	1.210	2.209	2.398	2.526	2.443	ns
Input delay from pin to input register	Pad to I/O input register	8	0	1.313	1.208	2.205	2.406	2.563	2.450	ns
Delay from output register to output pin	I/O output register to pad	2	0	0.461	0.421	0.789	0.869	0.933	0.884	ns
Input delay from dual-purpose clock pin to fan-out destinations	Pad to global clock network	12	0	0.712	0.682	1.225	1.407	1.562	1.421	ns

Notes to Table 1–45:

(1) The incremental values for the settings are generally linear. For exact values of each setting, use the latest version of Quartus II software.

(2) The minimum and maximum offset timing numbers are in reference to setting **0** as available in the Quartus II software