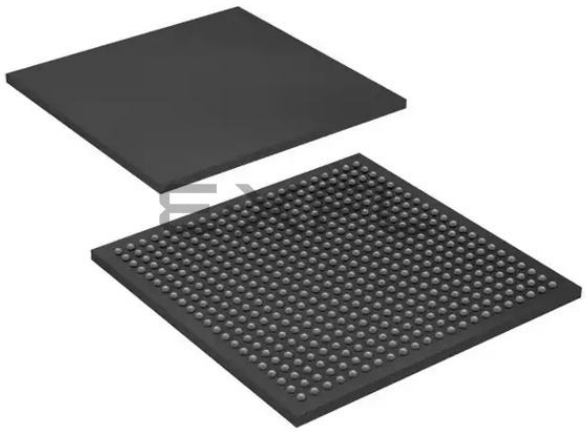


Welcome to E-XFL.COM

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	7155
Number of Logic Elements/Cells	114480
Total RAM Bits	3981312
Number of I/O	280
Number of Gates	-
Voltage - Supply	1.15V ~ 1.25V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/intel/ep4ce115f23c7n

Chapter 11. Power Requirements for Cyclone IV Devices

Chapter Revision Dates

The chapters in this document, Cyclone IV Device Handbook,, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Cyclone IV FPGA Device Family Overview
Revised: *March 2016*
Part Number: *CYIV-51001-2.0*

Chapter 2. Logic Elements and Logic Array Blocks in Cyclone IV Devices
Revised: *November 2009*
Part Number: *CYIV-51002-1.0*

Chapter 3. Memory Blocks in Cyclone IV Devices
Revised: *November 2011*
Part Number: *CYIV-51003-1.1*

Chapter 4. Embedded Multipliers in Cyclone IV Devices
Revised: *February 2010*
Part Number: *CYIV-51004-1.1*

Chapter 5. Clock Networks and PLLs in Cyclone IV Devices
Revised: *October 2012*
Part Number: *CYIV-51005-2.4*

Chapter 6. I/O Features in Cyclone IV Devices
Revised: *March 2016*
Part Number: *CYIV-51006-2.7*

Chapter 7. External Memory Interfaces in Cyclone IV Devices
Revised: *March 2016*
Part Number: *CYIV-51007-2.6*

Chapter 8. Configuration and Remote System Upgrades in Cyclone IV Devices
Revised: *May 2013*
Part Number: *CYIV-51008-1.7*

Chapter 9. SEU Mitigation in Cyclone IV Devices
Revised: *May 2013*
Part Number: *CYIV-51009-1.3*

Chapter 10. JTAG Boundary-Scan Testing for Cyclone IV Devices
Revised: *December 2013*
Part Number: *CYIV-51010-1.3*

Chapter 11. Power Requirements for Cyclone IV Devices
Revised: *May 2013*
Part Number: *CYIV-51011-1.3*

Section I. Device Core

This section provides a complete overview of all features relating to the Cyclone® IV device family, which is the most architecturally advanced, high-performance, low-power FPGA in the marketplace. This section includes the following chapters:

- Chapter 1, Cyclone IV FPGA Device Family Overview
- Chapter 2, Logic Elements and Logic Array Blocks in Cyclone IV Devices
- Chapter 3, Memory Blocks in Cyclone IV Devices
- Chapter 4, Embedded Multipliers in Cyclone IV Devices
- Chapter 5, Clock Networks and PLLs in Cyclone IV Devices

Revision History

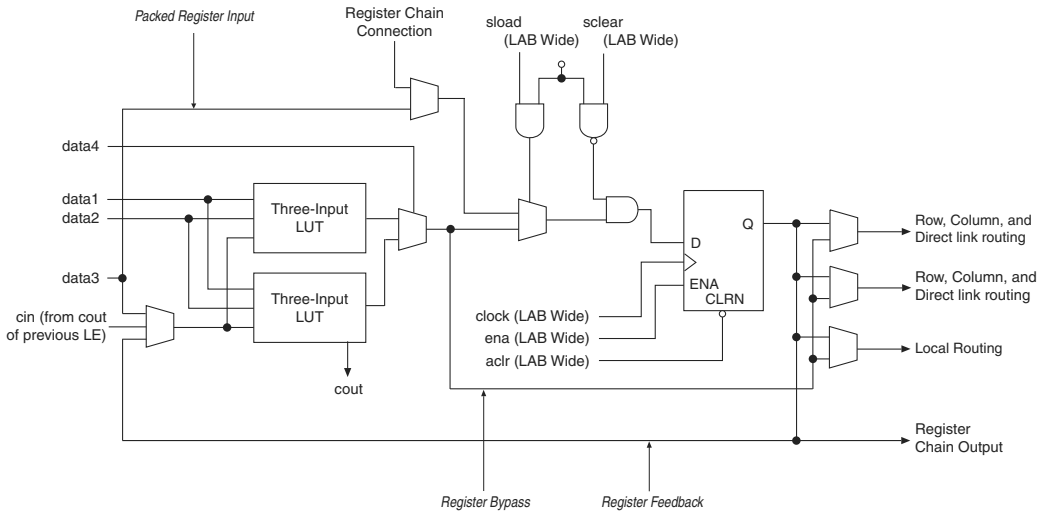
Refer to each chapter for its own specific revision history. For information about when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the complete handbook.

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.

Figure 2–3. Cyclone IV Device LEs in Arithmetic Mode

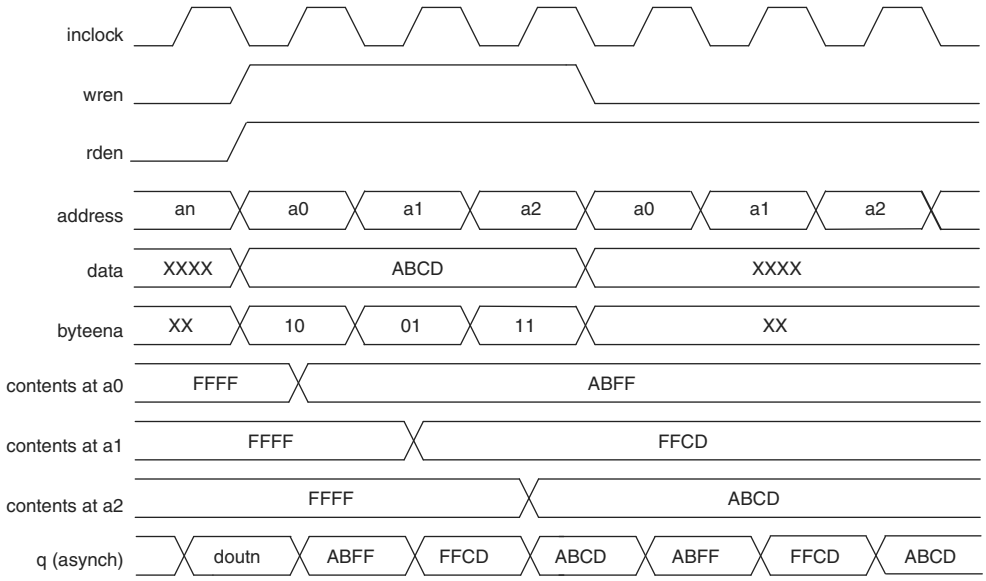


The Quartus II Compiler automatically creates carry chain logic during design processing. You can also manually create the carry chain logic during design entry. Parameterized functions, such as LPM functions, automatically take advantage of carry chains for the appropriate functions.

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 3–1 shows how the wren and byteena signals control the RAM operations.

Figure 3–1. Cyclone IV Devices byteena Functional Waveform *(1)*



Note to Figure 3–1:

(1) For this functional waveform, **New Data** mode is selected.

When a byteena bit is deasserted during a write cycle, the old data in the memory appears in the corresponding data-byte output. When a byteena bit is asserted during a write cycle, the corresponding data-byte output depends on the setting chosen in the Quartus® II software. The setting can either be the newly written data or the old data at that location.

Byte enables are only supported for True Dual-Port memory configurations when both the PortA and PortB data widths of the individual M9K memory blocks are multiples of 8 or 9 bits.

Packed Mode Support

Cyclone IV devices M9K memory blocks support packed mode. You can implement two single-port memory blocks in a single block under the following conditions:

- Each of the two independent block sizes is less than or equal to half of the M9K block size. The maximum data width for each independent block is 18 bits wide.
- Each of the single-port memory blocks is configured in single-clock mode. For more information about packed mode support, refer to “Single-Port Mode” on page 3–8 and “Single-Clock Mode” on page 3–15.

Table 5–5. Cyclone IV GX PLL Features (Part 2 of 2)

Features	Availability									
	General Purpose PLLs				Multipurpose PLLs					
	PLL_1 (1), (10)	PLL_2 (1), (10)	PLL_3 (2)	PLL_4 (3)	PLL_1 (4)	PLL_2 (4)	PLL_5 (1), (10)	PLL_6 (1), (10)	PLL_7 (1)	PLL_8 (1)
Input clock switchover	✓									
User mode reconfiguration	✓									
Loss of lock detection	✓									
PLL drives TX Serial Clock, TX Load Enable, and TX Parallel Clock	✓	✓	—	—	✓					
VCO output drives RX clock data recovery (CDR) clock	—				✓					
PLL drives FREF for ppm detect	✓	✓	—	—	✓					

Notes to Table 5–5:

(1) This is only applicable to EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices in F672 and F896 package.

(2) This is applicable to all Cyclone IV devices.

(3) This is applicable to all Cyclone IV devices except EP4CGX15 devices in all packages, EP4CGX22, and EP4CGX30 devices in F169 package.

(4) This is only applicable to EP4CGX15, EP4CGX22, and all EP4CGX30 devices except EP4CGX30 in the F484 package..

(5) C 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.

(6) These clock pins can access the GCLK networks.

(7) These clock pins are only available in EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices and cannot access the GCLK networks. CLK[17,19,20,21]p can be used as single-ended clock input pins.

(8) Only applicable if the input clock jitter is in the input jitter tolerance specifications.

(9) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by eight. For degree increments, Cyclone IV 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.

(10) This is applicable to the EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices in F484 package.

Table 5–6 lists the features available in Cyclone IV E PLLs.

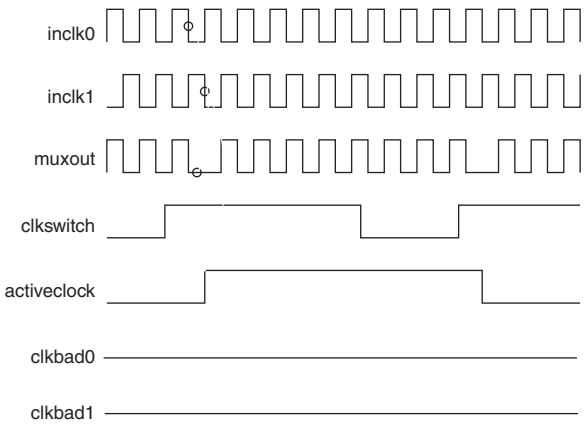
Table 5–6. Cyclone IV E PLL Features (Part 1 of 2)

Hardware Features	Availability
C (output counters)	5
M, N, C counter sizes	1 to 512 (1)
Dedicated clock outputs	1 single-ended or 1 differential pair
Clock input pins	4 single-ended or 2 differential pairs
Spread-spectrum input clock tracking	✓ (2)
PLL cascading	Through GCLK
Compensation modes	Source-Synchronous Mode, No Compensation Mode, Normal Mode, and Zero Delay Buffer Mode
Phase shift resolution	Down to 96-ps increments (3)
Programmable duty cycle	✓
Output counter cascading	✓
Input clock switchover	✓
User mode reconfiguration	✓

In this mode, the activeclock signal mirrors the clkswitch signal. As both blocks are still functional during the manual switch, neither clkbad signals go high. Because the switchover circuit is positive 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. The clkswitch signal and the automatic switch only works depending on the availability of the clock that is switched to. If the clock is unavailable, the state machine waits until the clock is available.

When CLKSWITCH = 1, it overrides the automatic switch-over function. As long as clkswitch signal is high, further switch-over action is blocked.

Figure 5–19. Clock Switchover Using the clkswitch Control *(1)*



Note to Figure 5–19:

(1) Both inclk0 and inclk1 must be running when the clkswitch signal goes high to start a manual clock switchover event.

Manual Clock Switchover

PLLs of Cyclone IV devices support manual switchover, in which the clkswitch signal controls whether inclk0 or inclk1 is the input clock to the PLL. The characteristics of a manual switchover are similar to the manual override feature in an automatic clock switchover, in which the switchover circuit is edge-sensitive. When the clkswitch signal goes high, the switchover sequence starts. The falling edge of the clkswitch signal does not cause the circuit to switch back to the previous input clock.

For more information about PLL software support in the Quartus II software, refer to the *ALTPLL Megafunction User Guide*.

Guidelines

Use the following guidelines to design with clock switchover in PLLs:

- Clock loss detection and automatic clock switchover require the inclk0 and inclk1 frequencies be within 20% of each other. Failing to meet this requirement causes the clkbad0 and clkbad1 signals to function improperly.

8. Configuration and Remote System Upgrades in Cyclone IV Devices

CYIV-51008-1.7

This chapter describes the configuration and remote system upgrades in Cyclone® IV devices. Cyclone IV (Cyclone IV GX and Cyclone IV E) devices use SRAM cells to store configuration data. You must download the configuration data to Cyclone IV devices each time the device powers up because SRAM memory is volatile.

Cyclone IV devices are configured using one of the following configuration schemes:

- Active serial (AS)
- Active parallel (AP) (supported in Cyclone IV E devices only)
- Passive serial (PS)
- Fast passive parallel (FPP) (not supported in EP4CGX15, EP4CGX22, and EP4CGX30 [except for the F484 package] devices)
- JTAG

Cyclone IV devices offer the following configuration features:

- Configuration data decompression (“Configuration Data Decompression” on page 8–2)
- Remote system upgrade (“Remote System Upgrade” on page 8–69)

System designers face difficult challenges, such as shortened design cycles, evolving standards, and system deployments in remote locations. Cyclone IV devices help overcome these challenges with inherent re-programmability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduced time-to-market, and extended product life.

Configuration

This section describes Cyclone IV device configuration and includes the following topics:

Configuration Process

This section describes Cyclone IV device configuration requirements and includes the following topics:

For more information about the Altera® FPGA configuration cycle state machine, refer to the *Configuring Altera FPGAs* chapter in volume 1 of the *Configuration Handbook*.

Power Up

If the device is powered up from the power-down state, V_{CCINT}, V_{CCA}, and V_{CCIO} (for the I/O banks in which the configuration and JTAG pins reside) must be powered up to the appropriate level for the device to exit from POR.

Reset

After power up, Cyclone IV devices go through POR. POR delay depends on the MSEL pin settings, which correspond to your configuration scheme. During POR, the device resets, holds nSTATUS and CONF_DONE low, and tri-states all user I/O pins (for PS and FPP configuration schemes only).

To tri-state the configuration bus for AS and AP configuration schemes, you must tie nCE high and nCONFIG low.

The user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are always enabled (after POR) before and during configuration. When the device exits POR, all user I/O pins continue to tri-state. While nCONFIG is low, the device is in reset. When nCONFIG goes high, the device exits reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage starts.

For more information about the value of the weak pull-up resistors on the I/O pins that are on before and during configuration, refer to the *Cyclone IV Device Datasheet* chapter.

Configuration

Configuration data is latched into the Cyclone IV device at each DCLK cycle. However, the width of the data bus and the configuration time taken for each scheme are different. After the device receives all the configuration data, the device 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 the CONF_DONE pin indicates that the configuration is complete and initialization of the device can begin.

The first Cyclone IV device in the chain is the configuration master and it controls the configuration of the entire chain. Other Altera devices that support PS configuration can also be part of the chain as configuration slaves.

☞ In the multi-device AS configuration, the board trace length between the serial configuration device and the master device of the Cyclone IV device must follow the recommendations in Table 8–7 on page 8–18.

The nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors, as shown in Figure 8–3 on page 8–13. These pins are open-drain bidirectional pins on the devices. When the first device asserts nCEO (after receiving all its configuration data), it releases its CONF_DONE pin. However, the subsequent devices in the chain keep this shared CONF_DONE line low until they receive their configuration data. When all target devices in the chain receive their configuration data and release CONF_DONE, the pull-up resistor drives a high level on CONF_DONE line and all devices simultaneously enter initialization mode.

☞ Although you can cascade Cyclone IV devices, serial configuration devices cannot be cascaded or chained together.

If the configuration bitstream size exceeds the capacity of a serial configuration device, you must select a larger configuration device, enable the compression feature, or both. When configuring multiple devices, the size of the bitstream is the sum of the individual device's configuration bitstream.

Configuring Multiple Cyclone IV Devices with the Same Design

Certain designs require that you configure multiple Cyclone IV devices with the same design through a configuration bitstream, or a **.sof**. You can do this through the following methods:

- Multiple **.sof**
- Single **.sof**

☞ For both methods, the serial configuration devices cannot be cascaded or chained together.

Multiple SRAM Object Files

Two copies of the **.sof** are stored in the serial configuration device. Use the first copy to configure the master device of the Cyclone IV device and the second copy to configure all remaining slave devices concurrently. All slave devices must have the same density and package. The setup is similar to Figure 8–3 on page 8–13.

To configure four identical Cyclone IV devices with the same **.sof**, you must set up the chain similar to the example shown in Figure 8–4. The first device is the master device and its MSEL pins must be set to select AS configuration. The other three slave devices are set up for concurrent configuration and their MSEL pins must be set to select PS configuration. The nCEO pin from the master device drives the nCE input pins on all three slave devices, as well as the DATA and DCLK pins that connect in parallel to all

For more information about the operation of the Micron P30 Parallel NOR and P33 Parallel NOR flash memories, search for the keyword "P30" or "P33" on the Micron website (www.micron.com) to obtain the P30 or P33 family datasheet.

Single-Device AP Configuration

The following groups of interface pins are supported in Micron P30 and P33 flash memories:

- Control pins
- Address pins
- Data pins

The following control signals are from the supported parallel flash memories:

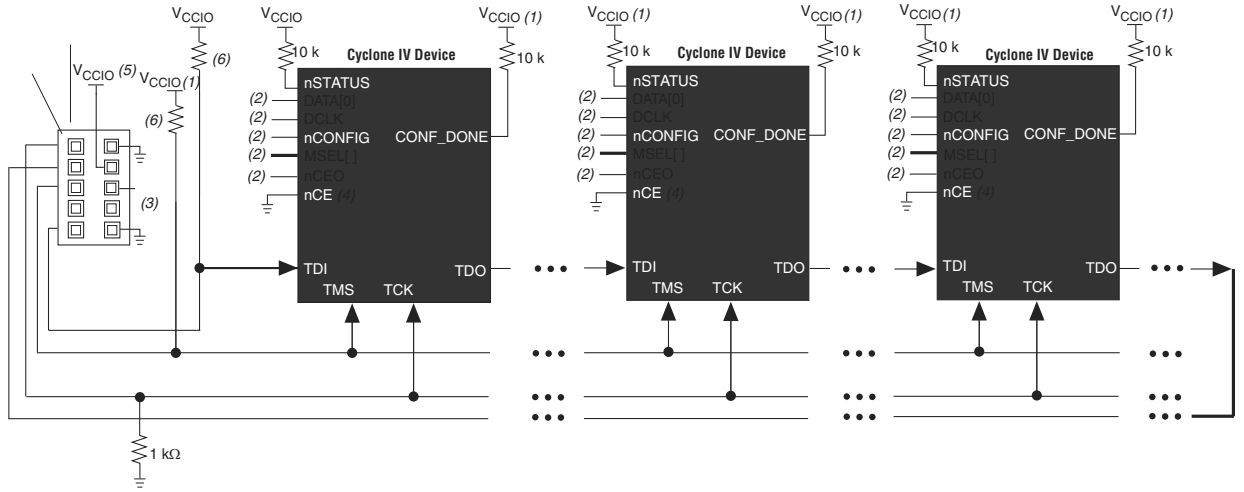
- CLK
- active-low reset (RST#)
- active-low chip enable (CE#)
- active-low output enable (OE#)
- active-low address valid (ADV#)
- active-low write enable (WE#)

The supported parallel flash memories output a control signal (WAIT) to Cyclone IV E devices to indicate when synchronous data is ready on the data bus. Cyclone IV E devices have a 24-bit address bus connecting to the address bus (A[24:1]) of the flash memory. A 16-bit bidirectional data bus (DATA[15..0]) provides data transfer between the Cyclone IV E device and the flash memory.

The following control signals are from the Cyclone IV E device to flash memory:

- DCLK
- active-low hard rest (nRESET)
- active-low chip enable (FLASH_nCE)
- active-low output enable for the DATA[15..0] bus and WAIT pin (nOE)
- active-low address valid signal and is used to write data into the flash (nAVD)
- active-low write enable and is used to write data into the flash (nWE)

Figure 8–26. JTAG Configuration of Multiple Devices Using a Download Cable (1.2, 1.5, and 1.8-V V_{CCIO} Powering the JTAG Pins)



Notes to Figure 8–26:

(1) Connect these pull-up resistors to the V_{CCIO} supply of the bank in which the pin resides.

(2) Connect the nCONFIG and MSEL pins to support a non-JTAG configuration scheme. If you only use a JTAG configuration, connect the nCONFIG pin to logic-high and the MSEL pins to GND. In addition, pull DCLK and DATA[0] to either high or low, whichever is convenient on your board.

(3) In the USB-Blaster and ByteBlaster II cable, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

(4) You must connect the nCE pin to GND or driven low for successful JTAG configuration.

(5) Power up the V_{CC} of the ByteBlaster II or USB-Blaster cable with supply from V_{CCIO}. The ByteBlaster II and USB-Blaster cables do not support a target supply voltage of 1.2 V. For the target supply voltage value, refer to the *ByteBlaster II Download Cable User Guide* and the *USB-Blaster Download Cable User Guide*.

(6) Resistor value can vary from 1 kΩ to 10 kΩ.

If a non-Cyclone IV device is cascaded in the JTAG-chain, TDO of the non-Cyclone IV device driving into TDI of the Cyclone IV device must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

The CONF_DONE and nSTATUS signals are shared in multi-device AS, AP, PS, and FPP configuration chains to ensure that the devices enter user mode at the same time after configuration is complete. When the CONF_DONE and nSTATUS signals are shared among all the devices, you must configure every device when JTAG configuration is performed.

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 8–25 or Figure 8–26, in which 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 nCE pin of the second device, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, ensure that 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 drives the nCE pin of the next device low when it has successfully been JTAG configured. You can place other Altera devices that have JTAG support in the same JTAG chain for device programming and configuration.

Table 8–25 lists the contents of previous state register 1 and previous state register 2 in the status register. The status register bit in Table 8–25 shows the bit positions in a 3-bit register. The previous state register 1 and previous state register 2 have the same bit definitions. The previous state register 1 reflects the current application configuration and the previous state register 2 reflects the previous application configuration.

Table 8–25. Remote System Upgrade Previous State Register 1 and Previous State Register 2 Contents in Status Register

Status Register Bit	Definition	Description
30	nCONFIG source	One-hot, active-high field that describes the reconfiguration source that caused the Cyclone IV device to leave the previous application configuration. If there is a tie, the higher bit order indicates precedence. For example, if nCONFIG and remote system upgrade nCONFIG reach the reconfiguration state machine at the same time, the nCONFIG precedes the remote system upgrade nCONFIG.
29	CRC error source	
28	nSTATUS source	
27	User watchdog timer source	
26	Remote system upgrade nCONFIG source	
25:24	Master state machine current state	The state of the master state machine during reconfiguration causes the Cyclone IV device to leave the previous application configuration.
23:0	Boot address	The address used by the configuration scheme to load the previous application configuration.

If a capture is inappropriately done while capturing a previous state before the system has entered remote update application configuration for the first time, a value outputs from the shift register to indicate that the capture is incorrectly called.

Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (Table 8–22 on page 8–75). While both registers can only be updated when the device 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 option bits (Cd_early and Osc_int), the configuration 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 starts system reconfiguration from the new application page.

☞ To ensure the successful reconfiguration between the pages, assert the RU_nCONFIG signal for a minimum of 250 ns. This is equivalent to strobing the reconfig input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

If there is an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (based on mode and error condition) by setting the control register accordingly.

Table 8–26 lists the contents of the control register after such an event occurs for all possible error or trigger conditions.

9. SEU Mitigation in Cyclone IV Devices

CYIV-51009-1.3

This chapter describes the cyclical redundancy check (CRC) error detection feature in user mode and how to recover from soft errors.

☞ Configuration error detection is supported in all Cyclone® IV devices including Cyclone IV GX devices, Cyclone IV E devices with 1.0-V core voltage, and Cyclone IV E devices with 1.2-V core voltage. However, user mode error detection is only supported in Cyclone IV GX devices and Cyclone IV E devices with 1.2-V core voltage.

Dedicated circuitry built into Cyclone IV devices consists of a CRC error detection feature that can optionally check for a single-event upset (SEU) continuously and automatically.

In critical applications used in the fields of avionics, telecommunications, system control, medical, and military applications, it is important to be able to:

- Confirm the accuracy of the configuration data stored in an FPGA device
- Alert the system to an occurrence of a configuration error

Using the CRC error detection feature for Cyclone IV devices does not impact fitting or performance.

This chapter contains the following sections:

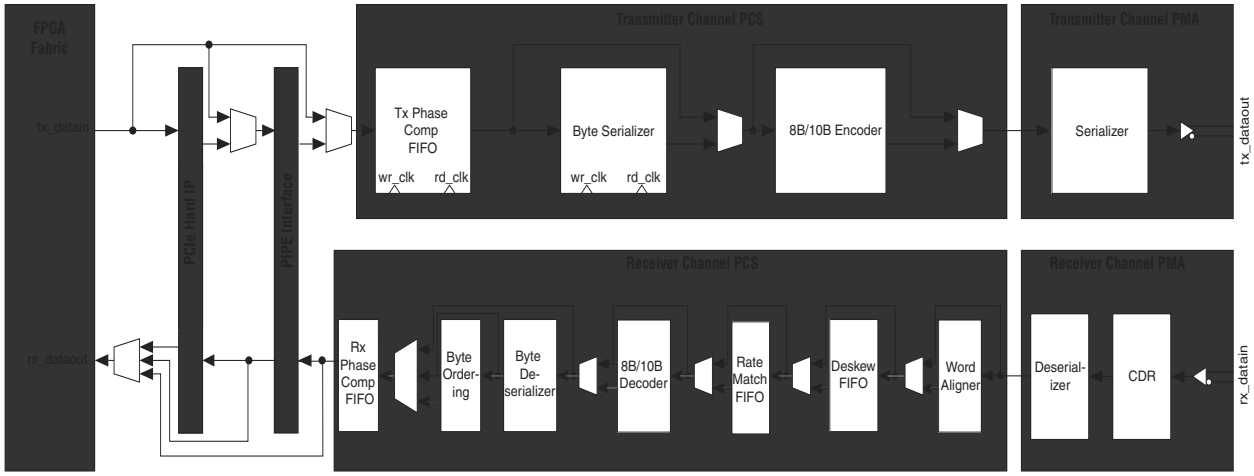
Configuration Error Detection

☞ Configuration error detection is available in all Cyclone IV devices including Cyclone IV GX devices, Cyclone IV E devices with 1.0-V core voltage, and Cyclone IV E devices with 1.2-V core voltage.

Architectural Overview

Figure 1–3 shows the Cyclone IV GX transceiver channel datapath.

Figure 1–3. Transceiver Channel Datapath for Cyclone IV GX Devices



Each transceiver channel consists of a transmitter and a receiver datapath. Each datapath is further structured into the following:

- Physical media attachment (PMA)—includes analog circuitry for I/O buffers, clock data recovery (CDR), serializer/deserializer (SERDES), and programmable pre-emphasis and equalization to optimize serial data channel performance.
- Physical coding sublayer (PCS)—includes hard logic implementation of digital functionality within the transceiver that is compliant with supported protocols.

Outbound parallel data from the FPGA fabric flows through the transmitter PCS and PMA, is transmitted as serial data. Received inbound serial data flows through the receiver PMA and PCS into the FPGA fabric. The transceiver supports the following interface widths:

- FPGA fabric-transceiver PCS—8, 10, 16, or 20 bits
- PMA-PCS—8 or 10 bits

The transceiver channel interfaces through the PIPE when configured for PCIe protocol implementation. The PIPE is compliant with version 2.00 of the *PHY Interface for the PCI Express Architecture* specification.

Transmitter Channel Datapath

The following sections describe the Cyclone IV GX transmitter channel datapath architecture as shown in Figure 1–3:

- TX Phase Compensation FIFO
- Byte Serializer
- 8B/10B Encoder
- Serializer
- Transmitter Output Buffer

TX Phase Compensation FIFO

The TX phase compensation FIFO compensates for the phase difference between the low-speed parallel clock and the FPGA fabric interface clock, when interfacing the transmitter channel to the FPGA fabric (directly or through the PIPE and PCIe hard IP). The FIFO is four words deep, with latency between two to three parallel clock cycles. Figure 1–4 shows the TX phase compensation FIFO block diagram.

Figure 1–4. TX Phase Compensation FIFO Block Diagram

Note to Figure 1–4:

(1) The x refers to the supported 8-, 10-, 16-, or 20-bits transceiver channel width.

The FIFO can operate in registered mode, contributing to only one parallel clock cycle of latency in Deterministic Latency functional mode. For more information, refer to "Deterministic Latency Mode" on page 1–73.

For more information about FIFO clocking, refer to "FPGA Fabric-Transceiver Interface Clocking" on page 1–43.

Byte Serializer

The byte serializer divides the input datapath width by two to allow transmitter channel operation at higher data rates while meeting the maximum FPGA fabric frequency limit. This module is required in configurations that exceed the maximum FPGA fabric-transceiver interface clock frequency limit and optional in configurations that do not.

For the FPGA fabric-transceiver interface frequency specifications, refer to the *Cyclone IV Device Data Sheet*.

For example, when operating an EP4CGX150 transmitter channel at 3.125 Gbps without byte serializer, the FPGA fabric frequency is 312.5 MHz (3.125 Gbps/10). This implementation violates the frequency limit and is not supported. Channel operation at 3.125 Gbps is supported when byte serializer is used, where the FPGA fabric frequency is 156.25 MHz (3.125 Gbps/20).

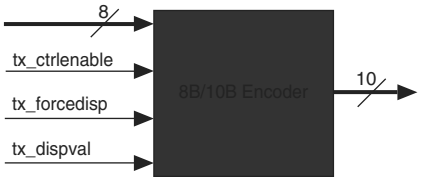
The byte serializer forwards the least significant byte first, followed by the most significant byte.

8B/10B Encoder

The optional 8B/10B encoder generates 10-bit code groups with proper disparity from the 8-bit data and 1-bit control identifier as shown in Figure 1–5.

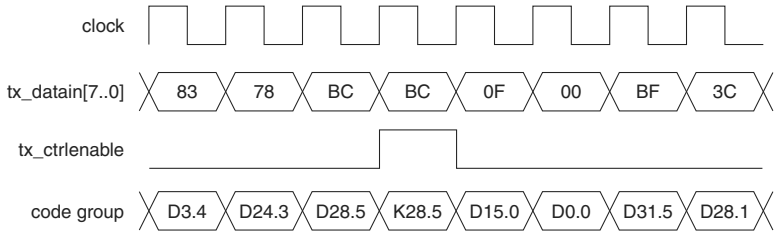
The encoder is compliant with Clause 36 of the *IEEE 802.3 Specification*.

Figure 1–5. 8B/10B Encoder Block Diagram



The 1-bit control identifier (`tx_ctrlenable`) port controls the 8-bit translation to either a 10-bit data word (Dx.y) or a 10-bit control word (Kx.y). Figure 1–6 shows the 8B/10B encoding operation with the `tx_ctrlenable` port, where the second 8'hBC data is encoded as a control word when `tx_ctrlenable` port is asserted, while the rest of the data is encoded as a data word.

Figure 1–6. Control and Data Word Encoding with the 8B/10B Encoder



The IEEE 802.3 8B/10B encoder specification identifies only a set of 8-bit characters for which the `tx_ctrlenable` port should be asserted. If you assert `tx_ctrlenable` port for any other set of characters, the 8B/10B encoder might encode the output 10-bit code as an invalid code (it does not map to a valid Dx.y or Kx.y code), or an unintended valid Dx.y code, depending on the value entered. It is possible for a downstream 8B/10B decoder to decode an invalid control word into a valid Dx.y code without asserting any code error flags. Altera recommends not to assert `tx_ctrlenable` port for unsupported 8-bit characters.

Table 1–4 lists the synchronization state machine parameters for the word aligner in this mode.

Table 1–4. Synchronization State Machine Parameters

Parameter	Allowed Values
Number of erroneous code groups received to lose synchronization	1–64
Number of continuous good code groups received to reduce the error count by one	1–256

After deassertion of the rx_digitalreset signal in automatic synchronization state machine mode, the word aligner starts looking for the synchronization code groups, word alignment pattern or its complement in the received data stream. When the programmed number of valid synchronization code groups or ordered sets are received, the rx_syncstatus signal is driven high to indicate that synchronization is acquired. The rx_syncstatus signal is constantly driven high until the programmed number of erroneous code groups are received without receiving intermediate good groups; after which the rx_syncstatus signal is driven low. The word aligner indicates loss of synchronization (rx_syncstatus signal remains low) until the programmed number of valid synchronization code groups are received again.

In addition to restoring word boundaries, the word aligner supports the following features:

- Programmable run length violation detection—detects consecutive 1s or 0s in the data stream, and asserts run length violation signal (rx_rlv) when a preset run length threshold (maximum number of consecutive 1s or 0s) is detected. The rx_rlv signal in each channel is clocked by its parallel recovered clock and is asserted for a minimum of two recovered clock cycles to ensure that the FPGA fabric clock can latch the rx_rlv signal reliably because the FPGA fabric clock might have phase differences, ppm differences (in asynchronous systems), or both, with the recovered clock. Table 1–5 lists the run length violation circuit detection capabilities.

Table 1–5. Run Length Violation Circuit Detection Capabilities

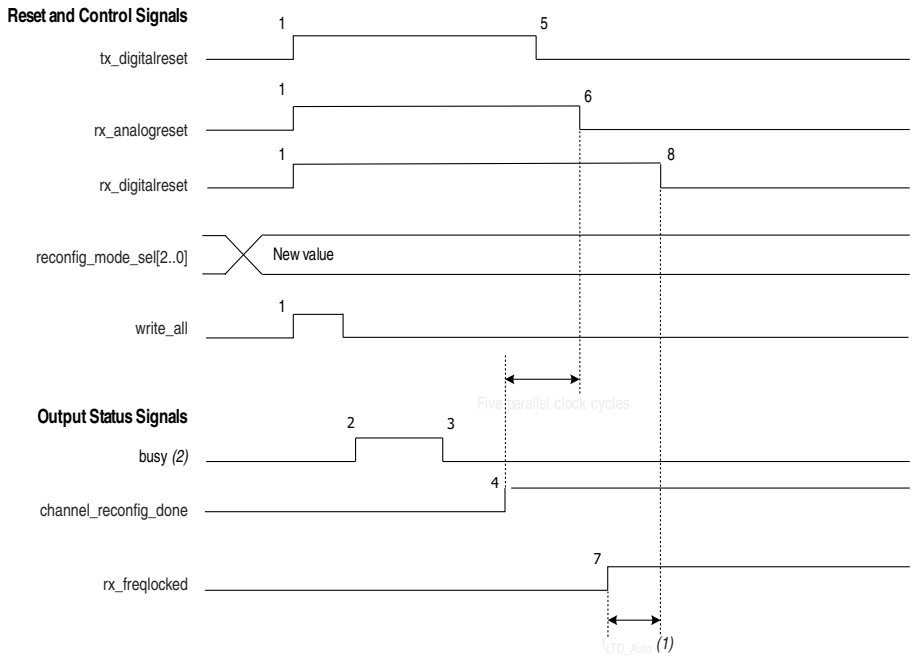
Supported Data Width	Detector Range		Increment Step Settings
	Minimum	Maximum	
8-bit	4	128	4
10-bit	5	160	5

2. After the PLL is reset, wait for the `pll_locked` signal to go high (marker 4) indicating that the PLL is locked to the input reference clock. After the assertion of the `pll_locked` signal, deassert the `tx_digitalreset` signal (marker 5).
3. Wait at least five parallel clock cycles after the `pll_locked` signal is asserted to deassert the `rx_analogreset` signal (marker 6).
4. When the `rx_freqlocked` signal goes high (marker 7), from that point onwards, wait for at least t_{LTD_Auto} time, then deassert the `rx_digitalreset` signal (marker 8). At this point, the receiver is ready for data traffic.

Reset Sequence in Channel Reconfiguration Mode

Use the example reset sequence shown in Figure 2–12 when you are using the dynamic reconfiguration controller to change the PCS settings of the transceiver channel. In this example, the dynamic reconfiguration is used to dynamically reconfigure the transceiver channel configured in Basic ×1 mode with receiver CDR in automatic lock mode.

Figure 2–12. Reset Sequence When Using the Dynamic Reconfiguration Controller to Change the PCS Settings of the Transceiver Channel



Notes to Figure 2–12:

(1) For t_{LTD_Auto} duration, refer to the *Cyclone IV Device Datasheet* chapter.

(2) 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.

Control and Status Signals for Channel Reconfiguration

The various control and status signals involved in the Channel Reconfiguration mode are as follows. Refer to "Dynamic Reconfiguration Controller Port List" on page 3–4 for the descriptions of the control and status signals.

The following are the input control signals:

- `logical_channel_address[n..0]`
- `reset_reconfig_address`
- `reconfig_reset`
- `reconfig_mode_sel[2..0]`
- `write_all`

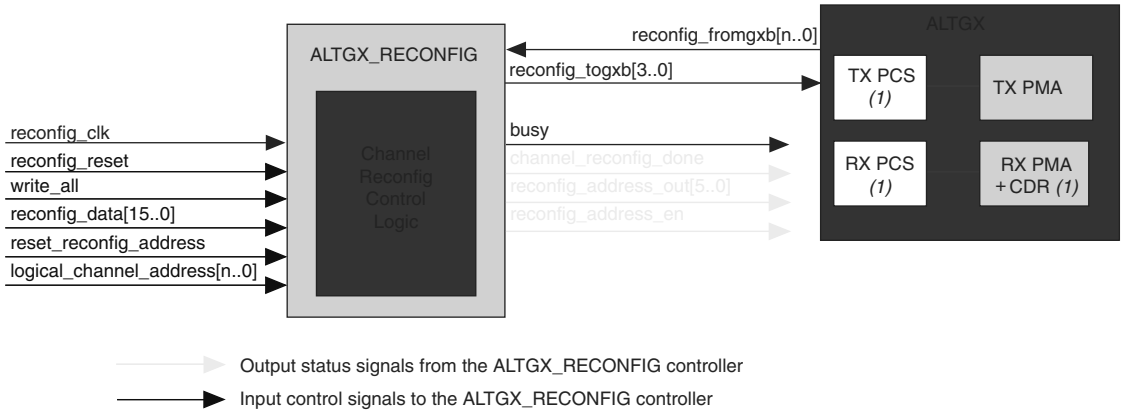
The following are output status signals:

- `reconfig_address_en`
- `reconfig_address_out[5..0]`
- `channel_reconfig_done`
- `busy`

The ALTGX_RECONFIG connection to the ALTGX instances when set in channel reconfiguration mode are as follows. For the port information, refer to "Dynamic Reconfiguration Controller Port List" on page 3–4.

Figure 3–10 shows the connection for channel reconfiguration mode.

Figure 3–10. ALTGX and ALTGX_RECONFIG Connection for Channel Reconfiguration Mode



Note to Figure 3–10:

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