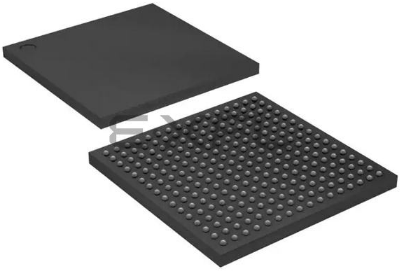
Intel - EP4CE15F17C8L Datasheet



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	963
Number of Logic Elements/Cells	15408
Total RAM Bits	516096
Number of I/O	165
Number of Gates	-
Voltage - Supply	0.97V ~ 1.03V
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/ep4ce15f17c8l

Word-Wide Multi-Device AP Configuration 8–26
Guidelines for Connecting Parallel Flash to Cyclone IV E Devices for an AP Interface 8–28
Configuring With Multiple Bus Masters 8–28
Estimating AP Configuration Time 8–30
Programming Parallel Flash Memories 8–31
PS Configuration 8–32
PS Configuration Using an External Host 8–33
PS Configuration Timing 8–36
PS Configuration Using a Download Cable 8–37
FPP Configuration 8–40
FPP Configuration Using an External Host 8–40
FPP Configuration Timing 8–44
JTAG Configuration 8–45
Configuring Cyclone IV Devices with Jam STAPL 8–52
Configuring Cyclone IV Devices with the JRunner Software Driver 8–52
Combining JTAG and AS Configuration Schemes 8–53
Programming Serial Configuration Devices In-System with the JTAG Interface 8–55
JTAG Instructions 8–57
Device Configuration Pins 8–62
Remote System Upgrade 8–69
Functional Description 8–69
Enabling Remote Update 8–70
Configuration Image Types 8–70
Remote System Upgrade Mode 8–71
Remote Update Mode 8–71
Dedicated Remote System Upgrade Circuitry 8–74
Remote System Upgrade Registers 8–75
Remote System Upgrade State Machine 8–78
User Watchdog Timer 8–79
Quartus II Software Support 8–80
Document Revision History 8–80

Chapter 9. SEU Mitigation in Cyclone IV Devices

Configuration Error Detection 9–1
User Mode Error Detection 9–2
Automated SEU Detection 9–3
CRC_ERROR Pin 9–3
Error Detection Block 9–4
Error Detection Registers 9–4
Error Detection Timing 9–5
Software Support 9–6
Accessing Error Detection Block Through User Logic 9–7
Recovering from CRC Errors 9–9
Document Revision History 9–10

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

IEEE Std. 1149.6 Boundary-Scan Register 10–2
BST Operation Control 10–3
EXTEST_PULSE 10–5
EXTEST_TRAIN 10–5
I/O Voltage Support in a JTAG Chain 10–5
Boundary-Scan Description Language Support 10–6
Document Revision History 10–7

Differential HSTL I/O Standard Support in Cyclone IV Devices 6–35
True Differential Output Buffer Feature 6–35
Programmable Pre-Emphasis 6–35
High-Speed I/O Timing 6–36
Design Guidelines 6–37
Differential Pad Placement Guidelines 6–37
Board Design Considerations 6–38
Software Overview 6–38
Document Revision History 6–39

Chapter 7. External Memory Interfaces in Cyclone IV Devices

Cyclone IV Devices Memory Interfaces Pin Support 7–2
Data and Data Clock/Strobe Pins 7–2
Optional Parity, DM, and Error Correction Coding Pins 7–11
Address and Control/Command Pins 7–12
Memory Clock Pins 7–12
Cyclone IV Devices Memory Interfaces Features 7–12
DDR Input Registers 7–12
DDR Output Registers 7–14
OCT with Calibration 7–15
PLL 7–15
Document Revision History 7–16

Section III. System Integration

Chapter 8. Configuration and Remote System Upgrades in Cyclone IV Devices

Configuration 8–1
Configuration Features 8–2
Configuration Data Decompression 8–2
Configuration Requirement 8–3
Power-On Reset (POR) Circuit 8–4
Configuration File Size 8–4
Configuration and JTAG Pin I/O Requirements 8–5
Configuration Process 8–6
Power Up 8–6
Reset 8–6
Configuration 8–6
Configuration Error 8–7
Initialization 8–7
User Mode 8–7
Configuration Scheme 8–8
AS Configuration (Serial Configuration Devices) 8–10
Single-Device AS Configuration 8–10
Multi-Device AS Configuration 8–13
Configuring Multiple Cyclone IV Devices with the Same Design 8–14
Guidelines for Connecting a Serial Configuration Device to Cyclone IV Devices for an AS Interface 8–18
Programming Serial Configuration Devices 8–19
AP Configuration (Supported Flash Memories) 8–21
AP Configuration Supported Flash Memories 8–22
Single-Device AP Configuration 8–23
Multi-Device AP Configuration 8–25
Byte-Wide Multi-Device AP Configuration 8–26

        Manual Clock Switchover . . . 5–30
        Guidelines . . . 5–30
  Programmable Bandwidth . . . 5–32
  Phase Shift Implementation . . . 5–32
  PLL Cascading . . . 5–33
  PLL Reconfiguration . . . 5–34
    PLL Reconfiguration Hardware Implementation . . . 5–34
        Post-Scale Counters (C0 to C4) . . . 5–36
        Scan Chain Description . . . 5–37
        Charge Pump and Loop Filter . . . 5–38
        Bypassing a PLL Counter . . . 5–39
        Dynamic Phase Shifting . . . 5–39
  Spread-Spectrum Clocking . . . 5–41
  PLL Specifications . . . 5–41
  Document Revision History . . . 5–42

Section II. I/O Interfaces

Chapter 6. I/O Features in Cyclone IV Devices

  Cyclone IV I/O Elements . . . 6–2
  I/O Element Features . . . 6–3
    Programmable Current Strength . . . 6–3
    Slew Rate Control . . . 6–4
    Open-Drain Output . . . 6–4
    Bus Hold . . . 6–4
    Programmable Pull-Up Resistor . . . 6–5
    Programmable Delay . . . 6–5
    PCI-Clamp Diode . . . 6–6
  OCT Support . . . 6–6
    On-Chip Series Termination with Calibration . . . 6–8
    On-Chip Series Termination Without Calibration . . . 6–10
  I/O Standards . . . 6–11
  Termination Scheme for I/O Standards . . . 6–13
    Voltage-Referenced I/O Standard Termination . . . 6–14
    Differential I/O Standard Termination . . . 6–15
  I/O Banks . . . 6–16
    High-Speed Differential Interfaces . . . 6–22
    External Memory Interfacing . . . 6–23
  Pad Placement and DC Guidelines . . . 6–23
    Pad Placement . . . 6–23
    DC Guidelines . . . 6–23
  Clock Pins Functionality . . . 6–23
  High-Speed I/O Interface . . . 6–24
  High-Speed I/O Standards Support . . . 6–28
    High Speed Serial Interface (HSSI) Input Reference Clock Support . . . 6–28
    LVDS I/O Standard Support in Cyclone IV Devices . . . 6–29
        Designing with LVDS . . . 6–30
    BLVDS I/O Standard Support in Cyclone IV Devices . . . 6–30
        Designing with BLVDS . . . 6–31
    RSDS, Mini-LVDS, and PPDS I/O Standard Support in Cyclone IV Devices . . . 6–32
        Designing with RSDS, Mini-LVDS, and PPDS . . . 6–32
    LVPECL I/O Support in Cyclone IV Devices . . . 6–34
    Differential SSTL I/O Standard Support in Cyclone IV Devices . . . 6–35

- True Dual-Port Mode . . . 3–11
- Shift Register Mode . . . 3–12
- ROM Mode . . . 3–13
- FIFO Buffer Mode . . . 3–13

Clocking Modes . . . 3–14
- Independent Clock Mode . . . 3–14
- Input or Output Clock Mode . . . 3–14
- Read or Write Clock Mode . . . 3–15
- Single-Clock Mode . . . 3–15

Design Considerations . . . 3–15
- Read-During-Write Operations . . . 3–15
  - Same-Port Read-During-Write Mode . . . 3–16
  - Mixed-Port Read-During-Write Mode . . . 3–16
- Conflict Resolution . . . 3–17
- Power-Up Conditions and Memory Initialization . . . 3–18
- Power Management . . . 3–18

Document Revision History . . . 3–18

Chapter 4. Embedded Multipliers in Cyclone IV Devices

Embedded Multiplier Block Overview . . . 4–1

Architecture . . . 4–2
- Input Registers . . . 4–3
- Multiplier Stage . . . 4–3
- Output Registers . . . 4–4

Operational Modes . . . 4–4
- 18-Bit Multipliers . . . 4–5
- 9-Bit Multipliers . . . 4–6

Document Revision History . . . 4–7

Chapter 5. Clock Networks and PLLs in Cyclone IV Devices

Clock Networks . . . 5–1
- GCLK Network . . . 5–2
- Clock Control Block . . . 5–10
- GCLK Network Clock Source Generation . . . 5–12
- GCLK Network Power Down . . . 5–16
- clkena Signals . . . 5–17

PLLs in Cyclone IV Devices . . . 5–18

Cyclone IV PLL Hardware Overview . . . 5–20
- External Clock Outputs . . . 5–22

Clock Feedback Modes . . . 5–23
- Source-Synchronous Mode . . . 5–23
- No Compensation Mode . . . 5–24
- Normal Mode . . . 5–24
- Zero Delay Buffer Mode . . . 5–25
- Deterministic Latency Compensation Mode . . . 5–26

Hardware Features . . . 5–26
- Clock Multiplication and Division . . . 5–26
- Post-Scale Counter Cascading . . . 5–27
- Programmable Duty Cycle . . . 5–27
- PLL Control Signals . . . 5–27
- Clock Switchover . . . 5–28
  - Automatic Clock Switchover . . . 5–28
  - Manual Override . . . 5–29

Table 9–7 lists the input and output ports that you must include in the atom.

Table 9–7. CRC Block Input and Output Ports

Port	Input/Output	Definition
<crcblock_name>	Input	Unique identifier for the CRC block, and represents any identifier name that is legal for the given description language (for example, Verilog HDL, VHDL, and AHDL). This field is required.
.clk(*<clock source>*	Input	This signal designates the clock input of this cell. All operations of this cell are with respect to the rising edge of the clock. Whether it is the loading of the data into the cell or data out of the cell, it always occurs on the rising edge. This port is required.
`.shiftnld` (*<shiftnld source>*)	Input	This signal is an input into the error detection block. If `shiftnld=1`, the data is shifted from the internal shift register to the `regout` at each rising edge of `clk`. If `shiftnld=0`, the shift register parallel loads either the pre-calculated CRC value or the update register contents, depending on the `ldsrc` port input. To do this, the `shiftnld` must be driven low for at least two clock cycles. This port is required.
`.ldsrc` (*<ldsrc source>*)	Input	This signal is an input into the error detection block. If `ldsrc=0`, the pre-computed CRC register is selected for loading into the 32-bit shift register at the rising edge of `clk` when `shiftnld=0`. If `ldsrc=1`, the signature register (result of the CRC calculation) is selected for loading into the shift register at the rising edge of `clk` when `shiftnld=0`. This port is ignored when `shiftnld=1`. This port is required.
`.crcerror` (*<crcerror indicator output>*)	Output	This signal is the output of the cell that is synchronized to the internal oscillator of the device (80-MHz internal oscillator) and not to the `clk` port. It asserts high if the error block detects that a SRAM bit has flipped and the internal CRC computation has shown a difference with respect to the pre-computed value. You must connect this signal either to an output pin or a bidirectional pin. If it is connected to an output pin, you can only monitor the `CRC_ERROR` pin (the core cannot access this output). If the `CRC_ERROR` signal is used by core logic to read error detection logic, you must connect this signal to a `BIDIR` pin. The signal is fed to the core indirectly by feeding a `BIDIR` pin that has its output enable port connected to V_{CC} (see Figure 9–3 on page 9–8).
`.regout` (*<registered output>*)	Output	This signal is the output of the error detection shift register synchronized to the `clk` port to be read by core logic. It shifts one bit at each cycle, so you should clock the `clk` signal 31 cycles to read out the 32 bits of the shift register.

Recovering from CRC Errors

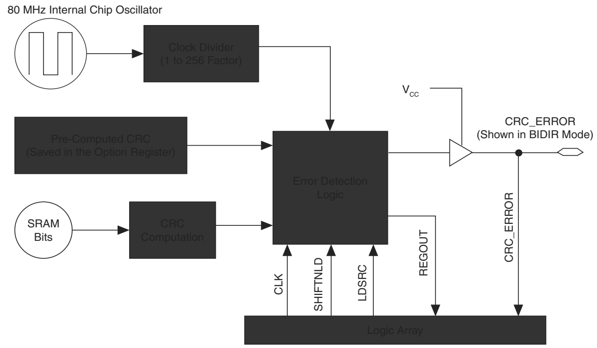
The system that the Altera FPGA resides in must control device reconfiguration. After detecting an error on the `CRC_ERROR` pin, strobing the `nCONFIG` low directs the system to perform the reconfiguration at a time when it is safe for the system to reconfigure the FPGA.

When the data bit is rewritten with the correct value by reconfiguring the device, the device functions correctly.

While soft errors are uncommon in Altera devices, certain high-reliability applications might require a design to account for these errors.

Figure 9–3 shows the error detection block diagram in FPGA devices and shows the interface that the WYSIWYG atom enables in your design.

Figure 9–3. Error Detection Block Diagram


80 MHz Internal Chip Oscillator
Clock Divider (1 to 256 Factor)
V_{CC}
CRC_ERROR (Shown in BIDIR Mode)
Pre-Computed CRC (Saved in the Option Register)
Error Detection Logic
SRAM Bits
CRC Computation
CRC_ERROR
REGOUT
CLK
SHIFTNLD
LDSRC
Logic Array

The user logic is affected by the soft error failure, so reading out the 32-bit CRC signature through the `regout` should not be relied upon to detect a soft error. You should rely on the `CRC_ERROR` output signal itself, because this `CRC_ERROR` output signal cannot be affected by a soft error.

To enable the `cycloneiv_crcblock` WYSIWYG atom, you must name the atom for each Cyclone IV device accordingly.

Example 9–1 shows an example of how to define the input and output ports of a WYSIWYG atom in a Cyclone IV device.

Example 9–1. Error Detection Block Diagram

```
cycloneiv_crcblock<crcblock_name>

(

.clk(<clock source>),

.shiftnld(<shiftnld source>),

.ldsrc(<ldsrc source>),

.crcerror(<crcerror out destination>),

.regout(<output destination>),

);
```

The divisor value divides the frequency of the configuration oscillator output clock. This output clock is used as the clock source for the error detection process.

8. Click **OK**.

Figure 9–2. Enabling the Error Detection CRC Feature in the Quartus II Software

Accessing Error Detection Block Through User Logic

The error detection circuit stores the computed 32-bit CRC signature in a 32-bit register, which is read out by user logic from the core. The `cycloneiv_crcblock` primitive is a WYSIWYG component used to establish the interface from the user logic to the error detection circuit. The `cycloneiv_crcblock` primitive atom contains the input and output ports that must be included in the atom. To access the logic array, the `cycloneiv_crcblock` WYSIWYG atom must be inserted into your design.

Table 9–6 lists the estimated time for each CRC calculation with minimum and maximum clock frequencies for Cyclone IV devices.

Table 9–6. CRC Calculation Time

Device		Minimum Time (ms) (1)	Maximum Time (s) (2)
Cyclone IV E	EP4CE6 (3)	5	2.29
	EP4CE10 (3)	5	2.29
	EP4CE15 (3)	7	3.17
	EP4CE22 (3)	9	4.51
	EP4CE30 (3)	15	7.48
	EP4CE40 (3)	15	7.48
	EP4CE55 (3)	23	11.77
	EP4CE75 (3)	31	15.81
	EP4CE115 (3)	45	22.67
Cyclone IV GX	EP4CGX15	6	2.93
	EP4CGX22	12	5.95
	EP4CGX30	12	5.95
		34 (4)	17.34 (4)
	EP4CGX50	34	17.34
	EP4CGX75	34	17.34
	EP4CGX110	62	31.27
	EP4CGX150	62	31.27

Notes to Table 9–6:

(1) The minimum time corresponds to the maximum error detection clock frequency and may vary with different processes, voltages, and temperatures (PVT).

(2) The maximum time corresponds to the minimum error detection clock frequency and may vary with different PVT.

(3) Only applicable for device with 1.2-V core voltage

(4) Only applicable for the F484 device package.

Software Support

Enabling the CRC error detection feature in the Quartus II software generates the `CRC_ERROR` output to the optional dual purpose `CRC_ERROR` pin.

To enable the error detection feature using CRC, perform the following steps:

1. Open the Quartus II software and load a project using Cyclone IV devices.
2. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.
3. In the Category list, select **Device**. The **Device** page appears.
4. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears as shown in Figure 9–2.
5. In the **Device and Pin Options** dialog box, click the **Error Detection CRC** tab.
6. Turn on **Enable error detection CRC**.
7. In the **Divide error check frequency by** box, enter a valid divisor as documented in Table 9–5 on page 9–5.

Table 9–4 defines the registers shown in Figure 9–1.

Table 9–4. Error Detection Registers

Register	Function
32-bit signature register	This register contains the CRC signature. The signature register contains the result of the user mode calculated CRC value compared against the pre-calculated CRC value. If no errors are detected, the signature register is all zeros. A non-zero signature register indicates an error in the configuration CRAM contents. The `CRC_ERROR` signal is derived from the contents of this register.
32-bit storage register	This register is loaded with the 32-bit pre-computed CRC signature at the end of the configuration stage. The signature is then loaded into the 32-bit CRC circuit (called the Compute and Compare CRC block, as shown in Figure 9–1) during user mode to calculate the CRC error. This register forms a 32-bit scan chain during execution of the `CHANGE_EDREG` JTAG instruction. The `CHANGE_EDREG` JTAG instruction can change the content of the storage register. Therefore, the functionality of the error detection CRC circuitry is checked in-system by executing the instruction to inject an error during the operation. The operation of the device is not halted when issuing the `CHANGE_EDREG` instruction.

Error Detection Timing

When the error detection CRC feature is enabled through the Quartus II software, the device automatically activates the CRC process upon entering user mode after configuration and initialization is complete.

The `CRC_ERROR` pin is driven low until the error detection circuitry detects a corrupted bit in the previous CRC calculation. After the pin goes high, it remains high during the next CRC calculation. This pin does not log the previous CRC calculation. If the new CRC calculation does not contain any corrupted bits, the `CRC_ERROR` pin is driven low. The error detection runs until the device is reset.

The error detection circuitry runs off an internal configuration oscillator with a divisor that sets the maximum frequency.

Table 9–5 lists the minimum and maximum error detection frequencies.

Table 9–5. Minimum and Maximum Error Detection Frequencies for Cyclone IV Devices

Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (2^n)
80 MHz/2^n	80 MHz	312.5 kHz	0, 1, 2, 3, 4, 5, 6, 7, 8

You can set a lower clock frequency by specifying a division factor in the Quartus II software (for more information, refer to “Software Support”). The divisor is a power of two (2), where *n* is between 0 and 8. The divisor ranges from one through 256. Refer to Equation 9–1.

Equation 9–1.

$$\text{ror detection frequency} = \frac{80\ \text{MH}}{2^n}$$

CRC calculation time depends on the device and the error detection clock frequency.

☞ WYSIWYG is an optimization technique that performs optimization on a VQM (Verilog Quartus Mapping) netlist in the Quartus II software.

Error Detection Block

Table 9–3 lists the types of CRC detection to check the configuration bits.

Table 9–3. Types of CRC Detection to Check the Configuration Bits

First Type of CRC Detection	Second Type of CRC Detection
■ CRAM error checking ability (32-bit CRC) during user mode, for use by the `CRC_ERROR` pin. ■ There is only one 32-bit CRC value. This value covers all the CRAM data.	■ 16-bit CRC embedded in every configuration data frame. ■ During configuration, after a frame of data is loaded into the device, the pre-computed CRC is shifted into the CRC circuitry. ■ Simultaneously, the CRC value for the data frame shifted-in is calculated. If the pre-computed CRC and calculated CRC values do not match, `nSTATUS` is set low. ■ Every data frame has a 16-bit CRC. Therefore, there are many 16-bit CRC values for the whole configuration bit stream. ■ Every device has a different length of configuration data frame.

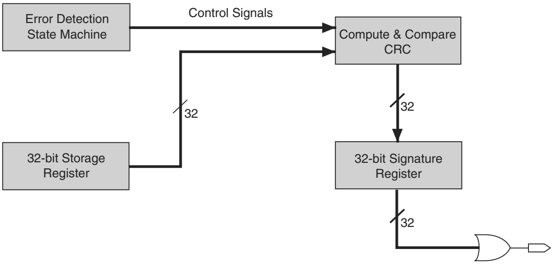
This section focuses on the first type—the 32-bit CRC when the device is in user mode.

Error Detection Registers

There are two sets of 32-bit registers in the error detection circuitry that store the computed CRC signature and pre-calculated CRC value. A non-zero value on the signature register causes the `CRC_ERROR` pin to set high.

Figure 9–1 shows the block diagram of the error detection block and the two related 32-bit registers: the signature register and the storage register.

Figure 9–1. Error Detection Block Diagram



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" on page 9–1
- "User Mode Error Detection" on page 9–2
- "Automated SEU Detection" on page 9–3
- "CRC_ERROR Pin" on page 9–3
- "Error Detection Block" on page 9–4
- "Error Detection Timing" on page 9–5
- "Software Support" on page 9–6
- "Recovering from CRC Errors" on page 9–9

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.

© 2013 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Section III. System Integration

This section includes the following chapters:

- Chapter 8, Configuration and Remote System Upgrades in Cyclone IV Devices
- Chapter 9, SEU Mitigation in Cyclone IV Devices
- Chapter 10, JTAG Boundary-Scan Testing for Cyclone IV Devices
- Chapter 11, Power Requirements for Cyclone IV Devices

Revision History

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

Section II. I/O Interfaces

This section provides information about Cyclone® IV device family I/O features and high-speed differential and external memory interfaces.

This section includes the following chapters:

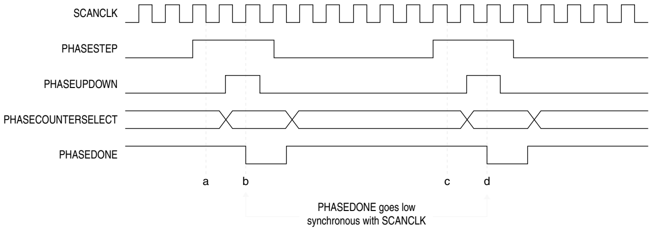
- Chapter 6, I/O Features in Cyclone IV Devices
- Chapter 7, External Memory Interfaces in Cyclone IV Devices

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.

Figure 5–26 shows the dynamic phase shifting waveform.

Figure 5–26. PLL Dynamic Phase Shift



The `PHASESTEP` signal is latched on the negative edge of `SCANCLK` (a,c) and must remain asserted for at least two `SCANCLK` cycles. Deassert `PHASESTEP` after `PHASEDONE` goes low. On the second `SCANCLK` rising edge (b,d) after `PHASESTEP` is latched, the values of `PHASEUPDOWN` and `PHASECOUNTERSELECT` are latched and the PLL starts dynamic phase-shifting for the specified counters, and in the indicated direction. `PHASEDONE` is deasserted synchronous to `SCANCLK` at the second rising edge (b,d) and remains low until the PLL finishes dynamic phase-shifting. Depending on the VCO and `SCANCLK` frequencies, `PHASEDONE` low time may be greater than or less than one `SCANCLK` cycle.

You can perform another dynamic phase-shift after the `PHASEDONE` signal goes from low to high. Each `PHASESTEP` pulse enables one phase shift. `PHASESTEP` pulses must be at least one `SCANCLK` cycle apart.

For information about the ALTPLL_RECONFIG MegaWizard™ Plug-In Manager, refer to the *ALTPLL_RECONFIG Megafunction User Guide*.

Spread-Spectrum Clocking

Cyclone IV devices can accept a spread-spectrum input with typical modulation frequencies. However, the device cannot automatically detect that the input is a spread-spectrum signal. Instead, the input signal looks like deterministic jitter at the input of the PLL. PLLs of Cyclone IV devices can track a spread-spectrum input clock as long as it is in the input jitter tolerance specifications and the modulation frequency of the input clock is below the PLL bandwidth, that is specified in the fitter report. Cyclone IV devices cannot generate spread-spectrum signals internally.

PLL Specifications

For information about PLL specifications, refer to the *Cyclone IV Device Datasheet* chapter.

Table 5–12. Dynamic Phase Shifting Control Signals (Part 2 of 2)

Signal Name	Description	Source	Destination
`scanclk`	Free running clock from core used in combination with `phasestep` to enable or disable dynamic phase shifting. Shared with `scanclk` for dynamic reconfiguration.	GCLK or I/O pins	PLL reconfiguration circuit
`phasedone`	When asserted, it indicates to core logic that the phase adjustment is complete and PLL is ready to act on a possible second adjustment pulse. Asserts based on internal PLL timing. De-asserts on the rising edge of `scanclk`.	PLL reconfiguration circuit	Logic array or I/O pins

Table 5–13 lists the PLL counter selection based on the corresponding `PHASECOUNTERSELECT` setting.

Table 5–13. Phase Counter Select Mapping

phasecounterselect			Selects
[2]	[1]	[0]	
0	0	0	All Output Counters
0	0	1	M Counter
0	1	0	C0 Counter
0	1	1	C1 Counter
1	0	0	C2 Counter
1	0	1	C3 Counter
1	1	0	C4 Counter

To perform one dynamic phase-shift, follow these steps:

1. Set `PHASEUPDOWN` and `PHASECOUNTERSELECT` as required.
2. Assert `PHASESTEP` for at least two `SCANCLK` cycles. Each `PHASESTEP` pulse allows one phase shift.
3. Deassert `PHASESTEP` after `PHASEDONE` goes low.
4. Wait for `PHASEDONE` to go high.
5. Repeat steps 1 through 4 as many times as required to perform multiple phase-shifts.

`PHASEUPDOWN` and `PHASECOUNTERSELECT` signals are synchronous to `SCANCLK` and must meet the t_{su} and t_h requirements with respect to the `SCANCLK` edges.

☞ You can repeat dynamic phase-shifting indefinitely. For example, in a design where the VCO frequency is set to 1,000 MHz and the output clock frequency is set to 100 MHz, performing 40 dynamic phase shifts (each one yields 125 ps phase shift) results in shifting the output clock by 180°, in other words, a phase shift of 5 ns.

Table 5–10. Loop Filter Control of High Frequency Capacitor

LFC[1]	LFC[0]	Setting (Decimal)
0	0	0
0	1	1
1	1	3

Bypassing a PLL Counter

Bypassing a PLL counter results in a divide (N, C0 to C4 counters) factor of one.

Table 5–11 lists the settings for bypassing the counters in PLLs of Cyclone IV devices.

Table 5–11. PLL Counter Settings

PLL Scan Chain Bits [0..8] Settings									Description
LSB								MSB	
X	X	X	X	X	X	X	X	1 (1)	PLL counter bypassed
X	X	X	X	X	X	X	X	0 (1)	PLL counter not bypassed

Note to Table 5–11:

(1) Bypass bit.

To bypass any of the PLL counters, set the bypass bit to 1. The values on the other bits are then ignored.

Dynamic Phase Shifting

The dynamic phase shifting feature allows the output phase of individual PLL outputs to be dynamically adjusted relative to each other and the reference clock without sending serial data through the scan chain of the corresponding PLL. This feature simplifies the interface and allows you to quickly adjust t_{CO} delays by changing output clock phase shift in real time. This is achieved by incrementing or decrementing the VCO phase-tap selection to a given C counter or to the M counter. The phase is shifted by 1/8 the VCO frequency at a time. The output clocks are active during this phase reconfiguration process.

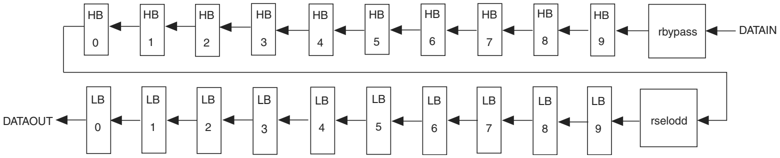
Table 5–12 lists the control signals that are used for dynamic phase shifting.

Table 5–12. Dynamic Phase Shifting Control Signals (Part 1 of 2)

Signal Name	Description	Source	Destination
`phasecounterselect[2..0]`	Counter Select. Three bits decoded to select either the M or one of the C counters for phase adjustment. One address map to select all C counters. This signal is registered in the PLL on the rising edge of `scanclk`.	Logic array or I/O pins	PLL reconfiguration circuit
`phaseupdown`	Selects dynamic phase shift direction; 1= UP, 0 = DOWN. Signal is registered in the PLL on the rising edge of `scanclk`.	Logic array or I/O pins	PLL reconfiguration circuit
`phasestep`	Logic high enables dynamic phase shifting.	Logic array or I/O pins	PLL reconfiguration circuit

Figure 5–25 shows the scan chain bit order sequence for one PLL post-scale counter in PLLs of Cyclone IV devices.

Figure 5–25. Scan Chain Bit Order


HB 0
HB 1
HB 2
HB 3
HB 4
HB 5
HB 6
HB 7
HB 8
HB 9
rbypass
DATAIN
DATAOUT
LB 0
LB 1
LB 2
LB 3
LB 4
LB 5
LB 6
LB 7
LB 8
LB 9
rselodd

Charge Pump and Loop Filter

You can reconfigure the charge pump and loop filter settings to update the PLL bandwidth in real time. Table 5–8 through Table 5–10 list the possible settings for charge pump current (I_{CP}), loop filter resistor (R), and capacitor (C) values for PLLs of Cyclone IV devices.

Table 5–8. Charge Pump Bit Control

CP[2]	CP[1]	CP[0]	Setting (Decimal)
0	0	0	0
1	0	0	1
1	1	0	3
1	1	1	7

Table 5–9. Loop Filter Resistor Value Control

LFR[4]	LFR[3]	LFR[2]	LFR[1]	LFR[0]	Setting (Decimal)
0	0	0	0	0	0
0	0	0	1	1	3
0	0	1	0	0	4
0	1	0	0	0	8
1	0	0	0	0	16
1	0	0	1	1	19
1	0	1	0	0	20
1	1	0	0	0	24
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	1	0	30

- Low time count = 1 cycle
- `rselodd` = 1 effectively equals:
  - High time count = 1.5 cycles
  - Low time count = 1.5 cycles
  - Duty cycle = (1.5/3)% high time count and (1.5/3)% low time count

Scan Chain Description

Cyclone IV PLLs have a 144-bit scan chain.

Table 5–7 lists the number of bits for each component of the PLL.

Table 5–7. Cyclone IV PLL Reprogramming Bits

Block Name	Number of Bits		
	Counter	Other	Total
C4 (1)	16	2 (2)	18
C3	16	2 (2)	18
C2	16	2 (2)	18
C1	16	2 (2)	18
C0	16	2 (2)	18
M	16	2 (2)	18
N	16	2 (2)	18
Charge Pump	9	0	9
Loop Filter (3)	9	0	9
Total number of bits:			144

Notes to Table 5–7:

(1) LSB bit for C4 low-count value is the first bit shifted into the scan chain.

(2) These two control bits include `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

(3) MSB bit for loop filter is the last bit shifted into the scan chain.

Figure 5–24 shows the scan chain order of the PLL components.

Figure 5–24. PLL Component Scan Chain Order

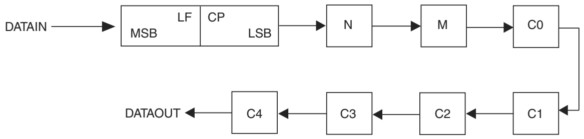
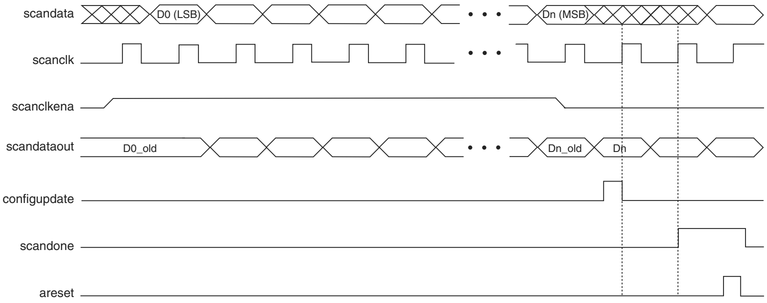


Figure 5–23 shows a functional simulation of the PLL reconfiguration feature.

Figure 5–23. PLL Reconfiguration Scan Chain



☞ When reconfiguring the counter clock frequency, the corresponding counter phase shift settings cannot be reconfigured using the same interface. You can reconfigure phase shifts in real time using the dynamic phase shift reconfiguration interface. If you reconfigure the counter frequency, but wish to keep the same non-zero phase shift setting (for example, 90°) on the clock output, you must reconfigure the phase shift after reconfiguring the counter clock frequency.

Post-Scale Counters (C0 to C4)

You can configure multiply or divide values and duty cycle of post-scale counters in real time. Each counter has an 8-bit high time setting and an 8-bit low time setting. The duty cycle is the ratio of output high or low time to the total cycle time, that is the sum of the two. Additionally, these counters have two control bits, `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

When the `rbypass` bit is set to 1, it bypasses the counter, resulting in a divide by one. When this bit is set to 0, the PLL computes the effective division of the VCO output frequency based on the high and low time counters. For example, if the post-scale divide factor is 10, the high and low count values are set to 5 and 5, to achieve a 50–50% duty cycle. The PLL implements this duty cycle by transitioning the output clock from high-to-low on the rising edge of the VCO output clock. However, a 4 and 6 setting for the high and low count values, respectively, would produce an output clock with a 40–60% duty cycle.

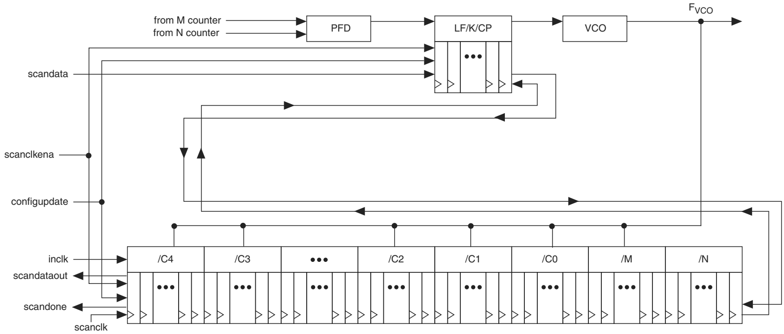
The `rselodd` bit indicates an odd divide factor for the VCO output frequency with a 50% duty cycle. For example, if the post-scale divide factor is three, the high and low time count values are 2 and 1, respectively, to achieve this division. This implies a 67%–33% duty cycle. If you need a 50%–50% duty cycle, you must set the `rselodd` control bit to 1 to achieve this duty cycle despite an odd division factor. The PLL implements this duty cycle by transitioning the output clock from high-to-low on a falling edge of the VCO output clock. When you set `rselodd` = 1, subtract 0.5 cycles from the high time and add 0.5 cycles to the low time.

For example:

- High time count = 2 cycles

Figure 5–22 shows how to adjust PLL counter settings dynamically by shifting their new settings into a serial shift register chain or scan chain. Serial data shifts to the scan chain via the `scandataport`, and shift registers are clocked by `scanclk`. The maximum `scanclk` frequency is 100 MHz. After shifting the last bit of data, asserting the `configupdate` signal for at least one `scanclk` clock cycle synchronously updates the PLL configuration bits with the data in the scan registers.

Figure 5–22. PLL Reconfiguration Scan Chain



The counter settings are updated synchronously to the clock frequency of the individual counters. Therefore, not all counters update simultaneously.

To reconfigure the PLL counters, perform the following steps:

1. The `scanclkena` signal is asserted at least one `scanclk` cycle prior to shifting in the first bit of `scandata` (`D0`).
2. Serial data (`scandata`) is shifted into the scan chain on the second rising edge of `scanclk`.
3. After all 144 bits have been scanned into the scan chain, the `scanclkena` signal is de-asserted to prevent inadvertent shifting of bits in the scan chain.
4. The `configupdate` signal is asserted for one `scanclk` cycle to update the PLL counters with the contents of the scan chain.
5. The `scandone` signal goes high indicating that the PLL is being reconfigured. A falling edge indicates that the PLL counters have been updated with new settings.
6. Reset the PLL using the `areset` signal if you make any changes to the M, N, post-scale output C counters, or the I_{CP}, R, C settings.
7. You can repeat steps 1 through 5 to reconfigure the PLL any number of times.

PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In PLLs of Cyclone IV devices, you can reconfigure both counter settings and phase shift the PLL output clock in real time. You can also change the charge pump and loop filter components, which dynamically affects PLL bandwidth. You can use these PLL components to update the output clock frequency, PLL bandwidth, and phase shift in real time, without reconfiguring the entire FPGA.

The ability to reconfigure the PLL in real time is useful in applications that might operate at multiple frequencies. It is also useful in prototyping environments, allowing you to sweep PLL output frequencies and adjust the output clock phase dynamically. For instance, a system generating test patterns is required to generate and send patterns at 75 or 150 MHz, depending on the requirements of the device under test. Reconfiguring PLL components in real time allows you to switch between two such output frequencies in a few microseconds.

You can also use this feature to adjust clock-to-out (t_{CO}) delays in real time by changing the PLL output clock phase shift. This approach eliminates the need to regenerate a configuration file with the new PLL settings.

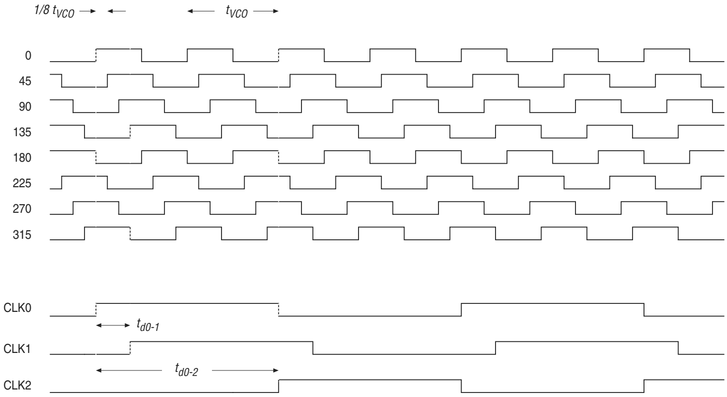
PLL Reconfiguration Hardware Implementation

The following PLL components are configurable in real time:

- Pre-scale counter (N)
- Feedback counter (M)
- Post-scale output counters (`C0–C4`)
- Dynamically adjust the charge pump current (I_{CP}) and loop filter components (R, C) to facilitate on-the-fly reconfiguration of the PLL bandwidth

Figure 5–21 shows an example of phase shift insertion using fine resolution through VCO phase taps method. The eight phases from the VCO are shown and labeled for reference. In this example, `CLK0` is based on 0° phase from the VCO and has the C value for the counter set to one. The `CLK1` signal is divided by four, two VCO clocks for high time and two VCO clocks for low time. `CLK1` is based on the 135° phase tap from the VCO and has the C value for the counter set to one. The `CLK1` signal is also divided by four. In this case, the two clocks are offset by 3 Φ_{fine}. `CLK2` is based on the 0° phase from the VCO but has the *C* value for the counter set to three. This creates a delay of two Φ_{coarse} (two complete VCO periods).

Figure 5–21. Delay Insertion Using VCO Phase Output and Counter Delay Time



You can use the coarse and fine phase shifts to implement clock delays in Cyclone IV devices.

Cyclone IV devices support dynamic phase shifting of VCO phase taps only. The phase shift is configurable for any number of times. Each phase shift takes about one `scanclk` cycle, allowing you to implement large phase shifts quickly.

PLL Cascading

Cyclone IV devices allow cascading between general purpose PLLs and multipurpose PLLs in normal or direct mode through the GCLK network. If your design cascades PLLs, the source (upstream) PLL must have a low-bandwidth setting, while the destination (downstream) PLL must have a high-bandwidth setting.

☞ `PLL_6` and `PLL7` have upstream cascading capability only.

☞ PLL cascading is not supported when used in transceiver applications.

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

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

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.

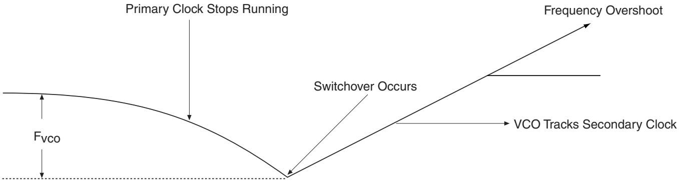
Equation 5–2. Coarse Resolution Phase Shift

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

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.

- When using manual clock switchover, the difference between `inclk0` and `inclk1` can be more than 20%. However, differences between the two clock sources (frequency, phase, or both) can cause the PLL to lose lock. Resetting the PLL ensures that the correct phase relationships are maintained between the input and output clocks.
- Both `inclk0` and `inclk1` must be running when the `clkswitch` signal goes high to start the manual clock switchover event. Failing to meet this requirement causes the clock switchover to malfunction.
- Applications that require a clock switchover feature and a small frequency drift must use a low-bandwidth PLL. When referencing input clock changes, the low-bandwidth PLL reacts slower than a high-bandwidth PLL. When the switchover happens, the low-bandwidth PLL propagates the stopping of the clock to the output slower than the high-bandwidth PLL. The low-bandwidth PLL filters out jitter on the reference clock. However, the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The exact amount of time it takes for the PLL to re-lock is dependent on the PLL configuration.
- If the phase relationship between the input clock to the PLL and output clock from the PLL is important in your design, assert `areset` for 10 ns after performing a clock switchover. Wait for the locked signal (or gated lock) to go high before re-enabling the output clocks from the PLL.
- Figure 5–20 shows how the VCO frequency gradually decreases when the primary clock is lost and then increases as the VCO locks on to the secondary clock. After the VCO locks on to the secondary clock, some overshoot can occur (an over-frequency condition) in the VCO frequency.

Figure 5–20. VCO Switchover Operating Frequency

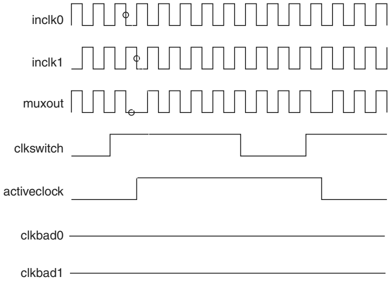


- Disable the system during switchover if the system is not tolerant to frequency variations during the PLL resynchronization period. You can use the `clkbad0` and `clkbad1` status signals to turn off the PFD (`pfdena = 0`) so the VCO maintains its last frequency. You can also use the switchover state machine to switch over to the secondary clock. Upon enabling the PFD, output clock enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. After the lock indication is stable, the system can re-enable the output clock or clocks.

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

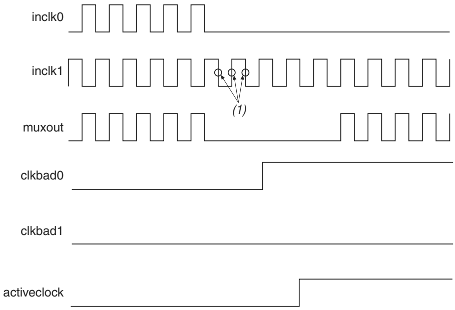
Use the following guidelines to design with clock switchover in PLLs:

- 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.

20%. This feature is useful when clock sources can originate from multiple cards on the backplane, requiring a system-controlled switchover between frequencies of operation. Choose the secondary clock frequency so the VCO operates in the recommended frequency range. Also, set the M, N, and C counters accordingly to keep the VCO operating frequency in the recommended range.

Figure 5–18 shows a waveform example of the switchover feature when using automatic loss of clock detection. Here, the `inclk0` signal remains low. After the `inclk0` signal remains low for approximately two clock cycles, the clock-sense circuitry drives the `clkbad0` signal high. Also, because the reference clock signal is not toggling, the switchover state machine controls the multiplexer through the `clksw` signal to switch to `inclk1`.

Figure 5–18. Automatic Switchover Upon Clock Loss Detection *(1)*



Note to Figure 5–18:

(1) Switchover is enabled on the falling edge of `inclk0` or `inclk1`, depending on which clock is available. In this figure, switchover is enabled on the falling edge of `inclk1`.

Manual Override

If you are using the automatic switchover, you must switch input clocks with the manual override feature with the `clkswitch` input.

Figure 5–19 shows an example of a waveform illustrating the switchover feature when controlled by `clkswitch`. In this case, both clock sources are functional and `inclk0` is selected as the reference clock. A low-to-high transition of the `clkswitch` signal starts the switchover sequence. The `clkswitch` signal must be high for at least three clock cycles (at least three of the longer clock period if `inclk0` and `inclk1` have different frequencies). On the falling edge of `inclk0`, the reference clock of the counter, `muxout`, is gated off to prevent any clock glitching. On the falling edge of `inclk1`, the reference clock multiplexer switches from `inclk0` to `inclk1` as the PLL reference, and the `activeclock` signal changes to indicate which clock is currently feeding the PLL.

Clock Switchover

The clock switchover feature allows the PLL to switch between two reference input clocks. Use this feature for clock redundancy or for a dual-clock domain application, such as a system that turns on the redundant clock if the previous clock stops running. Your design can automatically perform clock switchover when the clock is no longer toggling, or based on the user control signal, `clkswitch`.

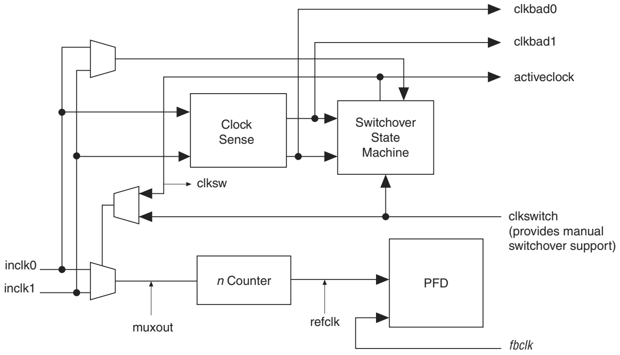
Automatic Clock Switchover

PLLs of Cyclone IV devices support a fully configurable clock switchover capability.

When the current reference clock is not present, the clock-sense block automatically switches to the backup clock for PLL reference. The clock switchover circuit also sends out three status signals—`clkbad0`, `clkbad1`, and `activeclock`—from the PLL to implement a custom switchover circuit. You can select a clock source at the backup clock by connecting it to the `inclk1` port of the PLL in your design.

Figure 5–17 shows the block diagram of the switchover circuit built into the PLL.

Figure 5–17. Automatic Clock Switchover Circuit



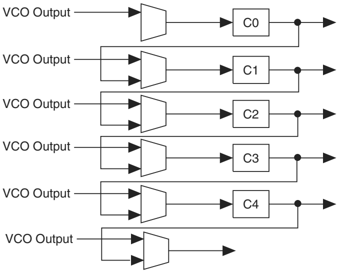
There are two ways to use the clock switchover feature:

- Use the switchover circuitry for switching from `inclk0` to `inclk1` running at the same frequency. For example, in applications that require a redundant clock with the same frequency as the reference clock, the switchover state machine generates a signal that controls the multiplexer select input shown in Figure 5–17. In this case, `inclk1` becomes the reference clock for the PLL. This automatic switchover can switch back and forth between the `inclk0` and `inclk1` clocks any number of times, when one of the two clocks fails and the other clock is available.
- Use the `clkswitch` input for user- or system-controlled switch conditions. This is possible for same-frequency switchover or to switch between inputs of different frequencies. For example, if `inclk0` is 66 MHz and `inclk1` is 200 MHz, you must control the switchover because the automatic clock-sense circuitry cannot monitor primary and secondary clock frequencies with a frequency difference of more than

Post-Scale Counter Cascading

PLLs of Cyclone IV devices support post-scale counter cascading to create counters larger than 512. This is implemented by feeding the output of one C counter into the input of the next C counter, as shown in Figure 5–16.

Figure 5–16. Counter Cascading



When cascading counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings.

For example, if C0 = 4 and C1 = 2, the cascaded value is C0 × C1 = 8.

☞ Post-scale counter cascading is automatically set by the Quartus II software in the configuration file. Post-scale counter cascading cannot be performed using the PLL reconfiguration.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on the PLL post-scale counters. You can achieve the duty cycle setting by a low and high time count setting for the post-scale counters. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices. The post-scale counter value determines the precision of the duty cycle. The precision is defined by 50% divided by the post-scale counter value. For example, if the C0 counter is 10, steps of 5% are possible for duty cycle choices between 5 to 90%.

Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

PLL Control Signals

You can use the `pfdena`, `areset`, and `locked` signals to observe and control the PLL operation and resynchronization.

For more information about the PLL control signals, refer to the *ALTPLL Megafunction User Guide*.

Deterministic Latency Compensation Mode

The deterministic latency mode compensates for the delay of the multipurpose PLLs through the clock network and serializer in Common Public Radio Interface (CPRI) applications. In this mode, the PLL PFD feedback path compensates the latency uncertainty in `Tx dataout` and `Tx clkout` paths relative to the reference clock.

Hardware Features

Cyclone IV PLLs support several features for general-purpose clock management. This section discusses clock multiplication and division implementation, phase shifting implementations, and programmable duty cycles.

Clock Multiplication and Division

Each Cyclone IV PLL provides clock synthesis for PLL output ports using M/(N*post-scale counter) scaling factors. The input clock is divided by a pre-scale factor, N, and is then multiplied by the M feedback factor. The control loop drives the VCO to match f_{IN} (M/N). Each output port has a unique post-scale counter that divides down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO value is the least common multiple of the output frequencies that meets its frequency specifications. For example, if output frequencies required from one PLL are 33 and 66 MHz, the Quartus II software sets the VCO to 660 MHz (the least common multiple of 33 and 66 MHz in the VCO range). Then, the post-scale counters scale down the VCO frequency for each output port.

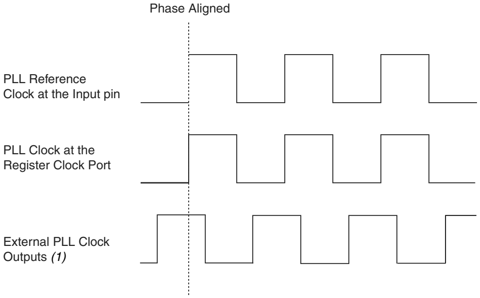
There is one pre-scale counter, N, and one multiply counter, M, per PLL, with a range of 1 to 512 for both M and N. The N counter does not use duty cycle control because the purpose of this counter is only to calculate frequency division. There are five generic post-scale counters per PLL that can feed GCLKs or external clock outputs. These post-scale counters range from 1 to 512 with a 50% duty cycle setting. The post-scale counters range from 1 to 256 with any non-50% duty cycle setting. The sum of the high/low count values chosen for a design selects the divide value for a given counter.

The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTPLL megafunction.

☞ Phase alignment between output counters is determined using the t_{PLL_PSERR} specification.

Figure 5–14 shows a waveform example of the phase relationship of the PLL clocks in this mode.

Figure 5–14. Phase Relationship Between PLL Clocks in Normal Mode


Phase Aligned
PLL Reference Clock at the Input pin
PLL Clock at the Register Clock Port
External PLL Clock Outputs *(1)*

Note to Figure 5–14:

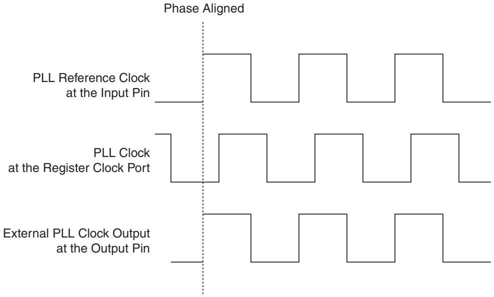
(1) The external clock output can lead or lag the PLL internal clock signals.

Zero Delay Buffer Mode

In zero delay buffer (ZDB) mode, the external clock output pin is phase-aligned with the clock input pin for zero delay through the device. When using this mode, use the same I/O standard on the input clock and output clocks to guarantee clock alignment at the input and output pins.

Figure 5–15 shows an example waveform of the phase relationship of the PLL clocks in ZDB mode.

Figure 5–15. Phase Relationship Between PLL Clocks in ZDB Mode

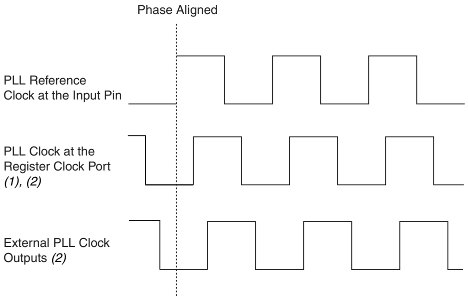

Phase Aligned
PLL Reference Clock at the Input Pin
PLL Clock at the Register Clock Port
External PLL Clock Output at the Output Pin

No Compensation Mode

In no compensation mode, the PLL does not compensate for any clock networks. This provides better jitter performance because clock feedback into the PFD does not pass through as much circuitry. Both the PLL internal and external clock outputs are phase shifted with respect to the PLL clock input.

Figure 5–13 shows a waveform example of the phase relationship of the PLL clock in this mode.

Figure 5–13. Phase Relationship Between PLL Clocks in No Compensation Mode


Phase Aligned
PLL Reference Clock at the Input Pin
PLL Clock at the Register Clock Port *(1), (2)*
External PLL Clock Outputs *(2)*

Notes to Figure 5–13:

(1) Internal clocks fed by the PLL are phase-aligned to each other.

(2) The PLL clock outputs can lead or lag the PLL input clocks.

Normal Mode

An internal clock in normal mode is phase-aligned to the input clock pin. The external clock output pin has a phase delay relative to the clock input pin if connected in this mode. The Quartus II software timing analyzer reports any phase difference between the two. In normal mode, the PLL fully compensates the delay introduced by the GCLK network.

Clock Feedback Modes

Cyclone IV PLLs support up to five different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle. For the supported feedback modes, refer to Table 5–5 on page 5–18 for Cyclone IV GX PLLs and Table 5–6 on page 5–19 for Cyclone IV E PLLs.

☞ Input and output delays are fully compensated by the PLL only if you are using the dedicated clock input pins associated with a given PLL as the clock sources.

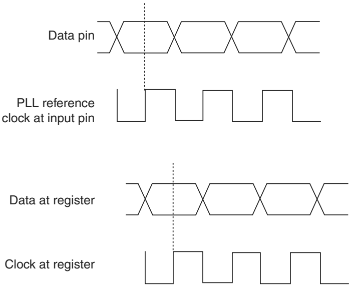
When driving the PLL using the GCLK network, the input and output delays may not be fully compensated in the Quartus II software.

Source-Synchronous Mode

If the data and clock arrive at the same time at the input pins, the phase relationship between the data and clock remains the same at the data and clock ports of any I/O element input register.

Figure 5–12 shows an example waveform of the data and clock in this mode. Use this mode for source-synchronous data transfers. Data and clock signals at the I/O element experience similar buffer delays as long as the same I/O standard is used.

Figure 5–12. Phase Relationship Between Data and Clock in Source-Synchronous Mode



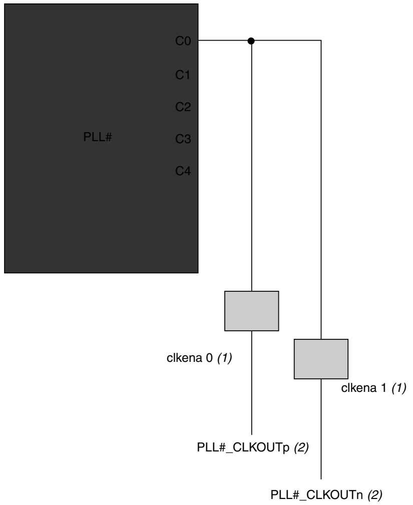
Source-synchronous mode compensates for delay of the clock network used, including any difference in the delay between the following two paths:

- Data pin to I/O element register input
- Clock input pin to the PLL phase frequency detector (PFD) input

☞ Set the input pin to the register delay chain in the I/O element to zero in the Quartus II software for all data pins clocked by a source-synchronous mode PLL. Also, all data pins must use the **PLL COMPENSATED logic** option in the Quartus II software.

Figure 5–11 shows the external clock outputs for PLLs.

Figure 5–11. External Clock Outputs for PLLs



Notes to Figure 5–11:

(1) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

(2) `PLL#_CLKOUTp` and `PLL#_CLKOUTn` pins are dual-purpose I/O pins that you can use as one single-ended clock output or one differential clock output. When using both pins as single-ended I/Os, one of them can be the clock output while the other pin is configured as a regular user I/O.

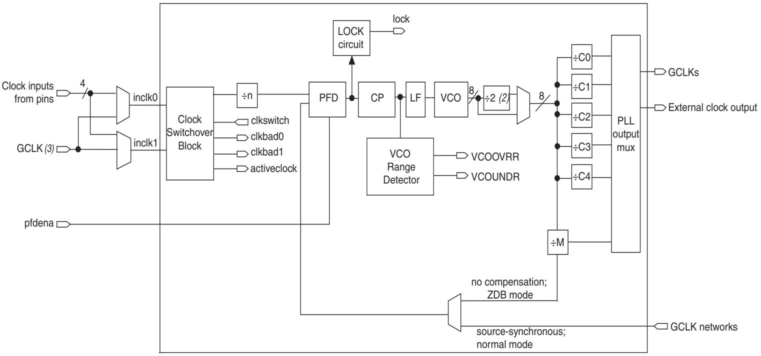
Each pin of a differential output pair is 180° out of phase. The Quartus II software places the NOT gate in your design into the I/O element to implement 180° phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins.

To determine which I/O standards are supported by the PLL clock input and output pins, refer to the *Cyclone IV Device I/O Features* chapter.

Cyclone IV PLLs can drive out to any regular I/O pin through the GCLK. You can also use the external clock output pins as GPIO pins if external PLL clocking is not required.

Figure 5–10 shows a simplified block diagram of the major components of the PLL of Cyclone IV E devices.

Figure 5–10. Cyclone IV E PLL Block Diagram *(1)*


lock
LOCK circuit
Clock inputs from pins
4
inclk0
inclk1
GCLK (3)
Clock Switchover Block
÷n
clkswitch
clkbad0
clkbad1
activeclock
PFD
CP
LF
VCO
8
÷2 (2)
8
VCO Range Detector
VCOOVRR
VCOUNDR
÷C0
÷C1
÷C2
÷C3
÷C4
PLL output mux
GCLKs
External clock output
pfdena
÷M
no compensation; ZDB mode
source-synchronous; normal mode
GCLK networks

Notes to Figure 5–10:

(1) Each clock source can come from any of the four clock pins located on the same side of the device as the PLL.

(2) This is the VCO post-scale counter K.

(3) This input port is fed by a pin-driven dedicated GCLK, or through a clock control block if the clock control block is fed by an output from another PLL or a pin-driven dedicated GCLK. An internally generated global signal cannot drive the PLL.

The VCO post-scale counter K is used to divide the supported VCO range by two. The VCO frequency reported by the Quartus II software in the PLL summary section of the compilation report takes into consideration the VCO post-scale counter value. Therefore, if the VCO post-scale counter has a value of 2, the frequency reported is lower than the f_{VCO} specification specified in the *Cyclone IV Device Datasheet* chapter.

External Clock Outputs

Each PLL of Cyclone IV devices supports one single-ended clock output or one differential clock output. Only the C0 output counter can feed the dedicated external clock outputs, as shown in Figure 5–11, without going through the GCLK. Other output counters can feed other I/O pins through the GCLK.

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

Hardware Features	Availability
Loss of lock detection	✓

Notes to Table 5–6:

(1) 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.

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

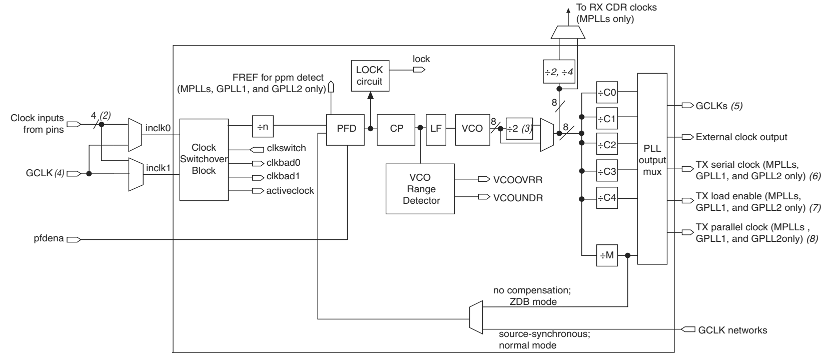
(3) The smallest phase shift is determined by the VCO period divided by eight. For degree increments, Cyclone IV E devices can shift all output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the frequency and divide parameters.

Cyclone IV PLL Hardware Overview

This section gives a hardware overview of the Cyclone IV PLL.

Figure 5–9 shows a simplified block diagram of the major components of the PLL of Cyclone IV GX devices.

Figure 5–9. Cyclone IV GX PLL Block Diagram *(1)*


To RX CDR clocks
(MPLLs only)
lock
LOCK circuit
FREF for ppm detect
(MPLLs, GPLL1, and GPLL2 only)
÷2, ÷4
8
Clock inputs from pins
4 (2)
inclk0
Clock Switchover Block
÷n
PFD
CP
LF
VCO
8
÷2 (3)
8
÷C0
÷C1
÷C2
÷C3
÷C4
÷M
PLL output mux
GCLKs (5)
External clock output
TX serial clock (MPLLs, GPLL1, and GPLL2 only) (6)
TX load enable (MPLLs, GPLL1, and GPLL2 only) (7)
TX parallel clock (MPLLs , GPLL1, and GPLL2only) (8)
GCLK (4)
inclk1
clkswitch
clkbad0
clkbad1
activeclock
VCO Range Detector
VCOOVRR
VCOUNDR
pfdena
no compensation;
ZDB mode
source-synchronous;
normal mode
GCLK networks

Notes to Figure 5–9:

(1) Each clock source can come from any of the four clock pins located on the same side of the device as the PLL.

(2) There are additional 4 pairs of dedicated differential clock inputs in EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices that can only drive general purpose PLLs and multipurpose PLLs on the left side of the device. `CLK[19..16]` can access `PLL_2`, `PLL_6`, `PLL_7`, and `PLL_8` while `CLK[23..20]` can access `PLL_1`, `PLL_5`, `PLL_6`, and `PLL_7`. For the location of these clock input pins, refer to Figure 5–3 on page 5–13.

(3) This is the VCO post-scale counter K.

(4) This input port is fed by a pin-driven dedicated GCLK, or through a clock control block if the clock control block is fed by an output from another PLL or a pin-driven dedicated GCLK. An internally generated global signal cannot drive the PLL.

(5) For the general purpose PLL and multipurpose PLL counter outputs connectivity to the GCLKs, refer to Table 5–1 on page 5–2 and Table 5–2 on page 5–4.

(6) Only the CI output counter can drive the TX serial clock.

(7) Only the C2 output counter can drive the TX load enable.

(8) Only the C3 output counter can drive the TX parallel clock.

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	✓

PLLs in Cyclone IV Devices

Cyclone IV GX devices offer two variations of PLLs: general purpose PLLs and multipurpose PLLs. Cyclone IV E devices only have the general purpose PLLs.

The general purpose PLLs are used for general-purpose applications in the FPGA fabric and periphery such as external memory interfaces. The multipurpose PLLs are used for clocking the transceiver blocks. When the multipurpose PLLs are not used for transceiver clocking, they can be used for general-purpose clocking.

For more details about the multipurpose PLLs used for transceiver clocking, refer to the *Cyclone IV Transceivers* chapter.

Cyclone IV GX devices contain up to eight general purpose PLLs and multipurpose PLLs while Cyclone IV E devices have up to four general purpose PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces.

For more information about the number of general purpose PLLs and multipurpose PLLs in each device density, refer to the *Cyclone IV Device Family Overview* chapter.

The general I/O pins cannot drive the PLL clock input pins.

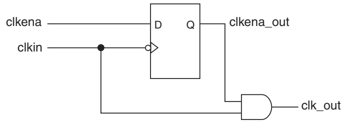
Table 5–5 lists the features available in Cyclone IV GX PLLs.

Table 5–5. Cyclone IV GX PLL Features (Part 1 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)
C (output counters)	5									
M, N, C counter sizes	1 to 512 (5)									
Dedicated clock outputs	1 single-ended or 1 differential pair									
Clock input pins	12 single-ended or 6 differential pairs (6) and 4 differential pairs (7)									
Spread-spectrum input clock tracking	✓ (8)									
PLL cascading	Through GCLK									
Source-Synchronous Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
No Compensation Mode	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Normal Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
Zero Delay Buffer Mode	✓	✓	✓	✓	✓	✓	✓	—	—	✓
Deterministic Latency Compensation Mode	✓	✓	—	—	✓	✓	✓	✓	✓	✓
Phase shift resolution (9)	Down to 96 ps increments									
Programmable duty cycle	✓									
Output counter cascading	✓									

Figure 5–7 shows how to implement the `clkena` signal with a single register.

Figure 5–7. clkena Implementation

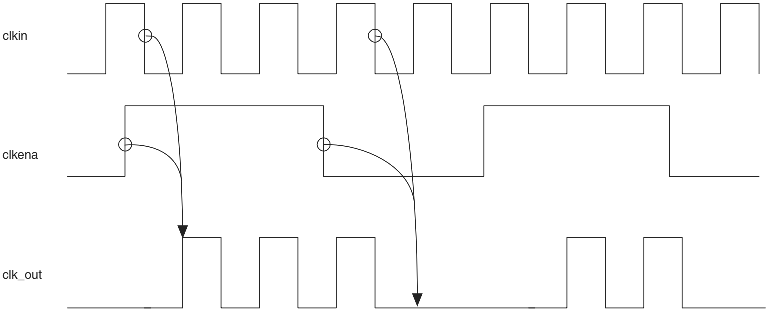


☞ The `clkena` circuitry controlling the output C0 of the PLL to an output pin is implemented with two registers instead of a single register, as shown in Figure 5–7.

Figure 5–8 shows the waveform example for a clock output enable. The `clkena` signal is sampled on the falling edge of the clock (`clkin`).

☞ This feature is useful for applications that require low power or sleep mode.

Figure 5–8. clkena Implementation: Output Enable



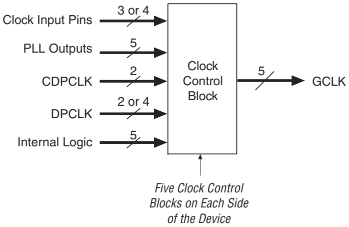
The `clkena` signal can also disable clock outputs if the system is not tolerant to frequency overshoot during PLL resynchronization.

Altera recommends using the `clkena` signals when switching the clock source to the PLLs or the GCLK. The recommended sequence is:

1. Disable the primary output clock by de-asserting the `clkena` signal.
2. Switch to the secondary clock using the dynamic select signals of the clock control block.
3. Allow some clock cycles of the secondary clock to pass before reasserting the `clkena` signal. The exact number of clock cycles you must wait before enabling the secondary clock is design-dependent. You can build custom logic to ensure glitch-free transition when switching between different clock sources.

Figure 5–6 shows a simplified version of the five clock control blocks on each side of the Cyclone IV E device periphery.

Figure 5–6. Clock Control Blocks on Each Side of Cyclone IV E Device *(1)*



Note to Figure 5–6:

(1) The left and right sides of the device have two `DPCLK` pins; the top and bottom of the device have four `DPCLK` pins.

GCLK Network Power Down

You can disable a Cyclone IV device's GCLK (power down) using both static and dynamic approaches. In the static approach, configuration bits are set in the configuration file generated by the Quartus II software, which automatically disables unused GCLKs. The dynamic clock enable or disable feature allows internal logic to control clock enable or disable the GCLKs in Cyclone IV devices.

When a clock network is disabled, all the logic fed by the clock network is in an off-state, thereby reducing the overall power consumption of the device. This function is independent of the PLL and is applied directly on the clock network, as shown in Figure 5–1 on page 5–11.

You can set the input clock sources and the `clkena` signals for the GCLK multiplexers through the Quartus II software using the ALTCLKCTRL megafunction.

For more information, refer to the *ALTCLKCTRL Megafunction User Guide*.

clkena Signals

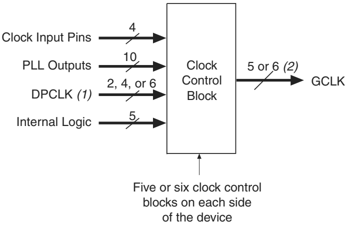
Cyclone IV devices support `clkena` signals at the GCLK network level. This allows you to gate-off the clock even when a PLL is used. Upon re-enabling the output clock, the PLL does not need a resynchronization or re-lock period because the circuit gates off the clock at the clock network level. In addition, the PLL can remain locked independent of the `clkena` signals because the loop-related counters are not affected.

From the clock sources listed above, only two clock input pins, two out of four PLL clock outputs (two clock outputs from either adjacent PLLs), one `DPCLK` pin, and one source from internal logic can drive into any given clock control block, as shown in Figure 5–1 on page 5–11.

Out of these six inputs to any clock control block, the two clock input pins and two PLL outputs are dynamically selected to feed a GCLK. The clock control block supports static selection of the signal from internal logic.

Figure 5–5 shows a simplified version of the clock control blocks on each side of the Cyclone IV GX device periphery.

Figure 5–5. Clock Control Blocks on Each Side of Cyclone IV GX Device



Notes to Figure 5–5:

(1) The EP4CGX15 device has two `DPCLK` pins; the EP4CGX22 and EP4CGX30 devices have four `DPCLK` pins; the EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices have six `DPCLK` pins.

(2) Each clock control block in the EP4CGX15, EP4CGX22, and EP4CGX30 devices can drive five GCLK networks. Each clock control block in the EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices can drive six GCLK networks.

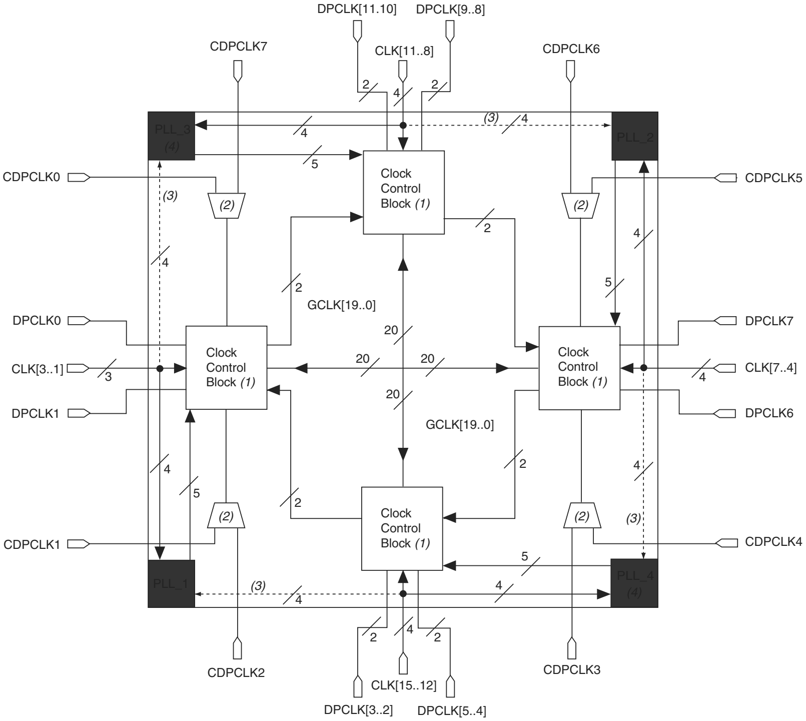
The inputs to the five clock control blocks on each side of the Cyclone IV E device must be chosen from among the following clock sources:

- Three or four clock input pins, depending on the specific device
- Five PLL counter outputs
- Two `DPCLK` pins and two `CDPCLK` pins from both the left and right sides and four `DPCLK` pins from both the top and bottom
- Five signals from internal logic

From the clock sources listed above, only two clock input pins, two PLL clock outputs, one `DPCLK` or `CDPCLK` pin, and one source from internal logic can drive into any given clock control block, as shown in Figure 5–1 on page 5–11.

Out of these six inputs to any clock control block, the two clock input pins and two PLL outputs are dynamically selected to feed a GCLK. The clock control block supports static selection of the signal from internal logic.

Figure 5–4. Clock Networks and Clock Control Block Locations in Cyclone IV E Devices


DPCLK[11.10]
DPCLK[9..8]
CDPCLK7
CLK[11..8]
CDPCLK6
2
4
2
PLL_3
(4)
(3)
4
4
5
PLL_2
CDPCLK0
CDPCLK5
Clock Control Block (1)
(3)
(2)
(2)
2
4
4
2
5
GCLK[19..0]
DPCLK0
DPCLK7
20
CLK[3..1]
3
Clock Control Block (1)
20
20
Clock Control Block (1)
CLK[7..4]
4
20
DPCLK1
DPCLK6
GCLK[19..0]
4
2
4
5
2
(2)
(2)
(3)
CDPCLK1
CDPCLK4
Clock Control Block (1)
5
PLL_1
(3)
4
4
PLL_4
(4)
2
4
2
CDPCLK2
CDPCLK3
CLK[15..12]
DPCLK[3..2]
DPCLK[5..4]

Notes to Figure 5–4:

(1) There are five clock control blocks on each side.

(2) Only one of the corner `CDPCLK` pins in each corner can feed the clock control block at a time. You can use the other `CDPCLK` pins as general-purpose I/O (GPIO) pins.

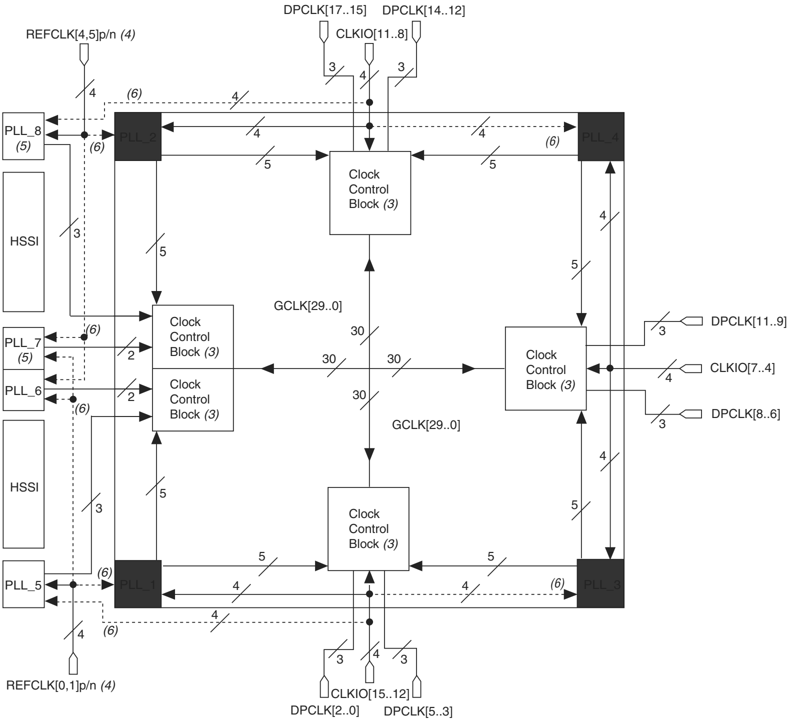
(3) Dedicated clock pins can feed into this PLL. However, these paths are not fully compensated.

(4) `PLL_3` and `PLL_4` are not available in EP4CE6 and EP4CE10 devices.

The inputs to the clock control blocks on each side of the Cyclone IV GX device must be chosen from among the following clock sources:

- Four clock input pins
- Ten PLL counter outputs (five from each adjacent PLLs)
- Two, four, or six `DPCLK` pins from the top, bottom, and right sides of the device
- Five signals from internal logic

Figure 5–3. Clock Networks and Clock Control Block Locations in EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 Devices *(1), (2)*


DPCLK[17..15]
DPCLK[14..12]
CLKIO[11..8]
REFCLK[4,5]p/n (4)
PLL_8 (5)
PLL_2
PLL_4
Clock Control Block (3)
HSSI
PLL_7 (5)
PLL_6
HSSI
PLL_5
PLL_1
PLL_3
GCLK[29..0]
30
DPCLK[11..9]
CLKIO[7..4]
DPCLK[8..6]
(6)
REFCLK[0,1]p/n (4)
CLKIO[15..12]
DPCLK[2..0]
DPCLK[5..3]

Notes to Figure 5–3:

(1) The clock networks and clock control block locations in this figure apply to only the EP4CGX30 device in F484 package and all EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices.

(2) `PLL_1`, `PLL_2`, `PLL_3`, and `PLL_4` are general purpose PLLs while `PLL_5`, `PLL_6`, `PLL_7`, and `PLL_8` are multipurpose PLLs.

(3) There are 6 clock control blocks on the top, right and bottom sides of the device and 12 clock control blocks on the left side of the device.

(4) `REFCLK[0,1]p/n` and `REFCLK[4,5]p/n` can only drive the general purpose PLLs and multipurpose PLLs on the left side of the device. These clock pins do not have access to the clock control blocks and GCLK networks. The `REFCLK[4,5]p/n` pins are not available in devices in F484 package.

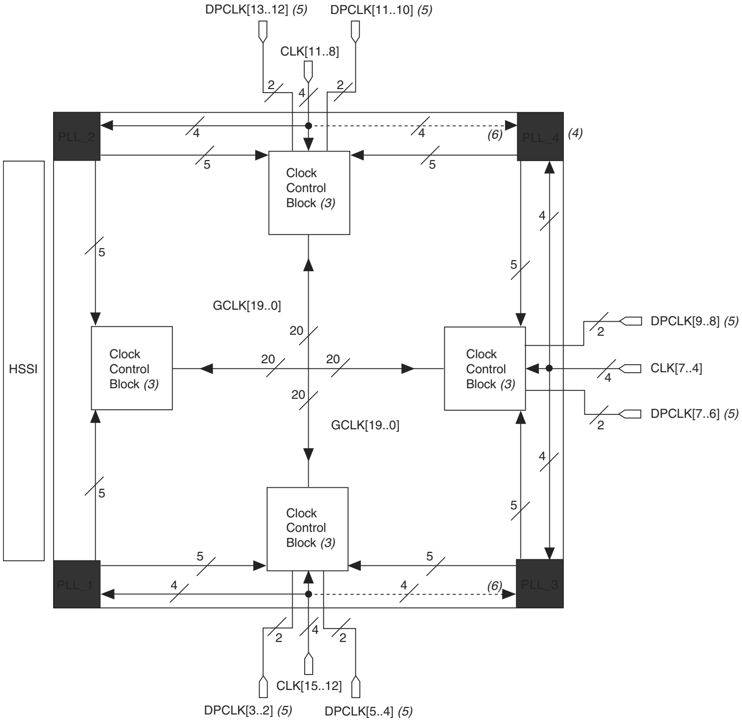
(5) Not available for EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices in F484 package.

(6) Dedicated clock pins can feed into this PLL. However, these paths are not fully compensated.

GCLK Network Clock Source Generation

Figure 5–2, Figure 5–3, and Figure 5–4 on page 5–14 show the Cyclone IV PLLs, clock inputs, and clock control block location for different Cyclone IV device densities.

Figure 5–2. Clock Networks and Clock Control Block Locations in EP4CGX15, EP4CGX22, and EP4CGX30 Devices *(1), (2)*


DPCLK[13..12] (5)
DPCLK[11..10] (5)
CLK[11..8]
2
4
2
PLL_2
4
4
(6)
PLL_4
(4)
5
5
Clock Control Block (3)
4
5
5
HSSI
GCLK[19..0]
20
Clock Control Block (3)
20
20
20
Clock Control Block (3)
2
DPCLK[9..8] (5)
CLK[7..4]
4
DPCLK[7..6] (5)
2
GCLK[19..0]
4
5
5
Clock Control Block (3)
5
5
PLL_1
4
4
(6)
PLL_3
2
4
2
CLK[15..12]
DPCLK[3..2] (5)
DPCLK[5..4] (5)

Notes to Figure 5–2:

(1) The clock networks and clock control block locations apply to all EP4CGX15, EP4CGX22, and EP4CGX30 devices except EP4CGX30 device in F484 package.

(2) `PLL_1` and `PLL_2` are multipurpose PLLs while `PLL_3` and `PLL_4` are general purpose PLLs.

(3) There are five clock control blocks on each side.

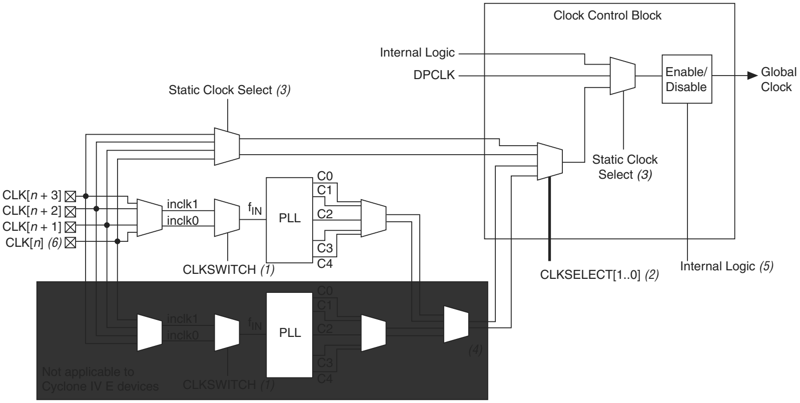
(4) `PLL_4` is only available in EP4CGX22 and EP4CGX30 devices in F324 package.

(5) The EP4CGX15 device has two `DPCLK` pins on three sides of the device: `DPCLK2` and `DPCLK5` on bottom side, `DPCLK7` and `DPCLK8` on the right side, `DPCLK10` and `DPCLK13` on the top side of device.

(6) Dedicated clock pins can feed into this PLL. However, these paths are not fully compensated.

Figure 5–1 shows the clock control block.

Figure 5–1. Clock Control Block


Clock Control Block
Internal Logic
DPCLK
Enable/ Disable
Global Clock
Static Clock Select (3)
Static Clock Select (3)
CLK[n + 3]
CLK[n + 2]
CLK[n + 1]
CLK[n] (6)
inclk1
inclk0
fIN
PLL
C0
C1
C2
C3
C4
CLKSWITCH (1)
CLKSELECT[1..0] (2)
Internal Logic (5)
inclk1
inclk0
fIN
PLL
C0
C1
C2
C3
C4
(4)
Not applicable to Cyclone IV E devices
CLKSWITCH (1)

Notes to Figure 5–1:

(1) The `clkswitch` signal can either be set through the configuration file or dynamically set when using the manual PLL switchover feature. The output of the multiplexer is the input clock (f_{IN}) for the PLL.

(2) The `clkselect[1..0]` signals are fed by internal logic and are used to dynamically select the clock source for the GCLK when the device is in user mode.

(3) The static clock select signals are set in the configuration file. Therefore, dynamic control when the device is in user mode is not feasible.

(4) Two out of four PLL clock outputs are selected from adjacent PLLs to drive into the clock control block.

(5) You can use internal logic to enable or disable the GCLK in user mode.

(6) `CLK[n]` is not available on the left side of Cyclone IV E devices.

Each PLL generates five clock outputs through the `c[4..0]` counters. Two of these clocks can drive the GCLK through a clock control block, as shown in Figure 5–1.

For more information about how to use the clock control block in the Quartus II software, refer to the *ALTCLKCTRL Megafunction User Guide*.

Table 5–3. GCLK Network Connections for Cyclone IV E Devices (1) (Part 3 of 3)

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
`DPCLK2` (4) `CDPCLK1`, or `CDPCLK2` (2), (5)	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK5` (4) `DPCLK7` (2)	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK4` (4) `DPCLK6` (2)	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK6` (4) `CDPCLK5`, or `CDPCLK6` (2), (5)	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK3` (4) `CDPCLK4`, or `CDPCLK3` (2), (5)	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—
`DPCLK8`	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
`DPCLK11`	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
`DPCLK9`	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
`DPCLK10`	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—
`DPCLK5`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
`DPCLK2`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
`DPCLK4`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
`DPCLK3`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓

Notes to Table 5–3:

(1) EP4CE6 and EP4CE10 devices only have GCLK networks 0 to 9.

(2) These pins apply to all Cyclone IV E devices except EP4CE6 and EP4CE10 devices.

(3) EP4CE6 and EP4CE10 devices only have `PLL_1` and `PLL_2`.

(4) This pin applies only to EP4CE6 and EP4CE10 devices.

(5) Only one of the two `CDPCLK` pins can feed the clock control block. You can use the other pin as a regular I/O pin.

Table 5–3. GCLK Network Connections for Cyclone IV E Devices (1) (Part 2 of 3)

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
`CLK15/DIFFCLK_6p` *(2)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
`PLL_1_C0` *(3)*	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_1_C1` *(3)*	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_1_C2` *(3)*	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_1_C3` *(3)*	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_1_C4` *(3)*	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_2_C0` *(3)*	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
`PLL_2_C1` *(3)*	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—
`PLL_2_C2` *(3)*	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_2_C3` *(3)*	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—
`PLL_2_C4` *(3)*	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—
`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`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓
`DPCLK0`	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK1`	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK7` *(4)* `CDPCLK0`, or `CDPCLK7` *(2)*, *(5)*	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 5–2. GCLK Network Connections for EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 Devices (1), (2) (Part 4 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
`DPCLK17`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—

Notes to Table 5–2:

(1) EP4CGX30 information in this table refers to only EP4CGX30 device in F484 package.

(2) `PLL_1`, `PLL_2`, `PLL_3`, and `PLL_4` are general purpose PLLs while `PLL_5`, `PLL_6`, `PLL_7`, and `PLL_8` are multipurpose PLLs.

(3) `PLL_7` and `PLL_8` are not available in EP4CXGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices in F484 package.

Table 5–3. GCLK Network Connections for Cyclone IV E Devices (1) (Part 1 of 3)

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
`CLK1`	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`CLK2/DIFFCLK_1p`	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`CLK3/DIFFCLK_1n`	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`CLK4/DIFFCLK_2p`	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—
`CLK5/DIFFCLK_2n`	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
`CLK6/DIFFCLK_3p`	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—
`CLK7/DIFFCLK_3n`	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
`CLK8/DIFFCLK_5n` *(2)*	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—
`CLK9/DIFFCLK_5p` *(2)*	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—
`CLK10/DIFFCLK_4n` *(2)*	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—
`CLK11/DIFFCLK_4p` *(2)*	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
`CLK12/DIFFCLK_7n` *(2)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓
`CLK13/DIFFCLK_7p` *(2)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—
`CLK14/DIFFCLK_6n` *(2)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓

Table 5–2. GCLK Network Connections for EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 Devices (1), (2) (Part 3 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_8_C0` (3)	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_8_C1` (3)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_8_C2` (3)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_8_C3` (3)	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`PLL_8_C4` (3)	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK0`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
`DPCLK1`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
`DPCLK2`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
`DPCLK3`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—
`DPCLK4`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
`DPCLK5`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—
`DPCLK6`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK7`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK8`	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK9`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK10`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK11`	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`DPCLK12`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
`DPCLK13`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
`DPCLK14`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
`DPCLK15`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
`DPCLK16`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—

Table 5–2. GCLK Network Connections for EP4CGX30, EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 Devices (1), (2) (Part 1 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
CLKIO4/DIFFCLK_2n	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
CLKIO5/DIFFCLK_2p	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
CLKIO6/DIFFCLK_3n	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
CLKIO7/DIFFCLK_3p	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
CLKIO8/DIFFCLK_5n	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—
CLKIO9/DIFFCLK_5p	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	✓	—	—	—	—	—	—
CLKIO10/DIFFCLK_4n/REFCLK3n	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓		—	—	—	—	—	—
CLKIO11/DIFFCLK_4p/REFCLK3p	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	—	—	—	—
CLKIO12/DIFFCLK_7p/REFCLK2p	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓	—
CLKIO13/DIFFCLK_7n/REFCLK2n	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	✓
CLKIO14/DIFFCLK_6p	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—
CLKIO15/DIFFCLK_6n	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	✓
PLL_1_C0	✓	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	✓
PLL_1_C1	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
PLL_1_C2	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—
PLL_1_C3	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—
PLL_1_C4	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓
PLL_2_C0	—	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	—	—	—	—
PLL_2_C1	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—
PLL_2_C2	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—
PLL_2_C3	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—
PLL_2_C4	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—

For more information about the number of GCLK networks in each device density, refer to the *Cyclone IV FPGA Device Family Overview* chapter.

GCLK Network

GCLKs drive throughout the entire device, feeding all device quadrants. All resources in the device (I/O elements, logic array blocks (LABs), dedicated multiplier blocks, and M9K memory blocks) can use GCLKs as clock sources. Use these clock network resources for control signals, such as clock enables and clears fed by an external pin. Internal logic can also drive GCLKs for internally generated GCLKs and asynchronous clears, clock enables, or other control signals with high fan-out.

Table 5–1, Table 5–2 on page 5–4, and Table 5–3 on page 5–7 list the connectivity of the clock sources to the GCLK networks.

Table 5–1. GCLK Network Connections for EP4CGX15, EP4CGX22, and EP4CGX30 (1), (2) (Part 1 of 2)

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
`CLK4/DIFFCLK_2n`	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—
`CLK5/DIFFCLK_2p`	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
`CLK6/DIFFCLK_3n`	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	—	—
`CLK7/DIFFCLK_3p`	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
`CLK8/DIFFCLK_5n`	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓	—	—	—	—	—
`CLK9/DIFFCLK_5p`	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—
`CLK10/DIFFCLK_4n/REFCLK1n`	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—
`CLK11/DIFFCLK_4p/REFCLK1p`	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
`CLK12/DIFFCLK_7p/REFCLK0p`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	✓
`CLK13/DIFFCLK_7n/REFCLK0n`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	—	—
`CLK14/DIFFCLK_6p`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓
`CLK15/DIFFCLK_6n`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
`PLL_1_C0`	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
`PLL_1_C1`	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓
`PLL_1_C2`	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—
`PLL_1_C3`	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—
`PLL_1_C4`	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓
`PLL_2_C0`	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
`PLL_2_C1`	—	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—
`PLL_2_C2`	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—
`PLL_2_C3`	—	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—
`PLL_2_C4`	—	—	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—
`PLL_3_C0`	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓	—

5. Clock Networks and PLLs in Cyclone IV Devices

CYIV-51005-2.4

This chapter describes the hierarchical clock networks and phase-locked loops (PLLs) with advanced features in the Cyclone® IV device family. It includes details about the ability to reconfigure the PLL counter clock frequency and phase shift in real time, allowing you to sweep PLL output frequencies and dynamically adjust the output clock phase shift.

☞ The Quartus® II software enables the PLLs and their features without external devices.

This chapter contains the following sections:

- "Clock Networks" on page 5–1
- "PLLs in Cyclone IV Devices" on page 5–18
- "Cyclone IV PLL Hardware Overview" on page 5–20
- "Clock Feedback Modes" on page 5–23
- "Hardware Features" on page 5–26
- "Programmable Bandwidth" on page 5–32
- "Phase Shift Implementation" on page 5–32
- "PLL Cascading" on page 5–33
- "PLL Reconfiguration" on page 5–34
- "Spread-Spectrum Clocking" on page 5–41
- "PLL Specifications" on page 5–41

Clock Networks

The Cyclone IV GX device provides up to 12 dedicated clock pins (`CLK[15..4]`) that can drive the global clocks (GCLKs). Cyclone IV GX devices support four dedicated clock pins on each side of the device except the left side. These clock pins can drive up to 30 GCLKs.

The Cyclone IV E device provides up to 15 dedicated clock pins (`CLK[15..1]`) that can drive up to 20 GCLKs. Cyclone IV E devices support three dedicated clock pins on the left side and four dedicated clock pins on the top, right, and bottom sides of the device except EP4CE6 and EP4CE10 devices. EP4CE6 and EP4CE10 devices only support three dedicated clock pins on the left side and four dedicated clock pins on the right side of the device.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

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.

Additional Information

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.”

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*

Chapter 11. Power Requirements for Cyclone IV Devices

External Power Supply Requirements 11–1
Hot-Socketing Specifications 11–2
    Devices Driven Before Power-Up 11–2
    I/O Pins Remain Tri-stated During Power-Up 11–2
Hot-socketing Feature Implementation 11–3
Power-On Reset Circuitry 11–3
Document Revision History 11–4

Table 1–2 lists Cyclone IV GX device resources.

Table 1–2. Resources for the Cyclone IV GX Device Family

Resources	**EP4CGX15**	**EP4CGX22**	**EP4CGX30** *(1)*	**EP4CGX30** *(2)*	**EP4CGX50** *(3)*	**EP4CGX75** *(3)*	**EP4CGX110** *(3)*	**EP4CGX150** *(3)*
Logic elements (LEs)	14,400	21,280	29,440	29,440	49,888	73,920	109,424	149,760
Embedded memory (Kbits)	540	756	1,080	1,080	2,502	4,158	5,490	6,480
Embedded 18 × 18 multipliers	0	40	80	80	140	198	280	360
General purpose PLLs	1	2	2	4 *(4)*	4 *(4)*	4 *(4)*	4 *(4)*	4 *(4)*
Multipurpose PLLs	2 *(5)*	2 *(5)*	2 *(5)*	2 *(5)*	4 *(5)*	4 *(5)*	4 *(5)*	4 *(5)*
Global clock networks	20	20	20	30	30	30	30	30
High-speed transceivers *(6)*	2	4	4	4	8	8	8	8
Transceiver maximum data rate (Gbps)	2.5	2.5	2.5	3.125	3.125	3.125	3.125	3.125
PCIe (PIPE) hard IP blocks	1	1	1	1	1	1	1	1
User I/O banks	9 *(7)*	9 *(7)*	9 *(7)*	11 *(8)*	11 *(8)*	11 *(8)*	11 *(8)*	11 *(8)*
Maximum user I/O *(9)*	72	150	150	290	310	310	475	475

Notes to Table 1–2:

(1) Applicable for the F169 and F324 packages.

(2) Applicable for the F484 package.

(3) Only two multipurpose PLLs for F484 package.

(4) Two of the general purpose PLLs are able to support transceiver clocking. For more information, refer to the *Clock Networks and PLLs in Cyclone IV Devices* chapter.

(5) You can use the multipurpose PLLs for general purpose clocking when they are not used to clock the transceivers. For more information, refer to the *Clock Networks and PLLs in Cyclone IV Devices* chapter.

(6) If PCIe ×1, you can use the remaining transceivers in a quad for other protocols at the same or different data rates.

(7) Including one configuration I/O bank and two dedicated clock input I/O banks for HSSI reference clock input.

(8) Including one configuration I/O bank and four dedicated clock input I/O banks for HSSI reference clock input.

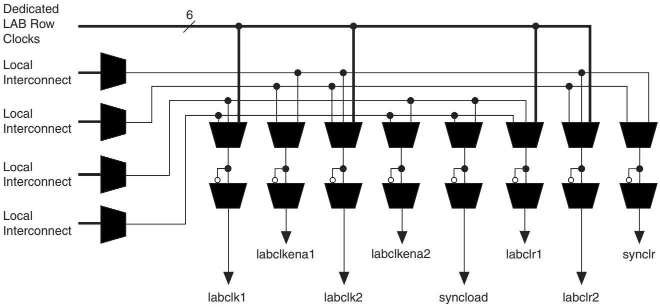
(9) The user I/Os count from pin-out files includes all general purpose I/O, dedicated clock pins, and dual purpose configuration pins. Transceiver pins and dedicated configuration pins are not included in the pin count.

Each LAB can use two clocks and two clock enable signals. The clock and clock enable signals of each LAB are linked. For example, any LE in a particular LAB using the `labclk1` signal also uses the `labclkena1`. If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock signals. Deasserting the clock enable signal turns off the LAB-wide clock.

The LAB row clocks `[5..0]` and LAB local interconnect generate the LAB-wide control signals. The MultiTrack interconnect inherent low skew allows clock and control signal distribution in addition to data distribution.

Figure 2–6 shows the LAB control signal generation circuit.

Figure 2–6. Cyclone IV Device LAB-Wide Control Signals



LAB-wide signals control the logic for the clear signal of the register. The LE directly supports an asynchronous clear function. Each LAB supports up to two asynchronous clear signals (`labclr1` and `labclr2`).

A LAB-wide asynchronous load signal to control the logic for the preset signal of the register is not available. The register preset is achieved with a NOT gate push-back technique. Cyclone IV devices only support either a preset or asynchronous clear signal**.**

In addition to the clear port, Cyclone IV devices provide a chip-wide reset pin (`DEV_CLRn`) that resets all registers in the device. An option set before compilation in the Quartus II software controls this pin. This chip-wide reset overrides all other control signals.

Document Revision History

Table 2–1 shows the revision history for this chapter.

Table 2–1. Document Revision History

Date	Version	Changes
November 2009	1.0	Initial release.

If you do not use dedicated clock pins to feed the GCLKs, you can use them as general-purpose input pins to feed the logic array. However, when using them as general-purpose input pins, they do not have support for an I/O register and must use LE-based registers in place of an I/O register.

For more information about how to connect the clock and PLL pins, refer to the *Cyclone IV Device Family Pin Connection Guidelines*.

Clock Control Block

The clock control block drives the GCLKs. Clock control blocks are located on each side of the device, close to the dedicated clock input pins. GCLKs are optimized for minimum clock skew and delay.

Table 5–4 lists the sources that can feed the clock control block, which in turn feeds the GCLKs.

Table 5–4. Clock Control Block Inputs

Input	Description
Dedicated clock inputs	Dedicated clock input pins can drive clocks or global signals, such as synchronous and asynchronous clears, presets, or clock enables onto given GCLKs.
Dual-purpose clock (`DPCLK` and `CDPCLK`) I/O input	`DPCLK` and `CDPCLK` I/O pins are bidirectional dual function pins that are used for high fan-out control signals, such as protocol signals, `TRDY` and `IRDY` signals for PCI, via the GCLK. Clock control blocks that have inputs driven by dual-purpose clock I/O pins are not able to drive PLL inputs.
PLL outputs	PLL counter outputs can drive the GCLK.
Internal logic	You can drive the GCLK through logic array routing to enable internal logic elements (LEs) to drive a high fan-out, low-skew signal path. Clock control blocks that have inputs driven by internal logic are not able to drive PLL inputs.

In Cyclone IV devices, dedicated clock input pins, PLL counter outputs, dual-purpose clock I/O inputs, and internal logic can all feed the clock control block for each GCLK. The output from the clock control block in turn feeds the corresponding GCLK. The GCLK can drive the PLL input if the clock control block inputs are outputs of another PLL or dedicated clock input pins. There are five or six clock control blocks on each side of the device periphery—depending on device density; providing up to 30 clock control blocks in each Cyclone IV GX device. The maximum number of clock control blocks per Cyclone IV E device is 20. For the clock control block locations, refer to Figure 5–2 on page 5–12, Figure 5–3 on page 5–13, and Figure 5–4 on page 5–14.

☞ The clock control blocks on the left side of the Cyclone IV GX device do not support any clock inputs.

The control block has two functions:

- Dynamic GCLK clock source selection (not applicable for `DPCLK`, `CDPCLK`, and internal logic input)
- GCLK network power down (dynamic enable and disable)

You can begin reconfiguration by pulling the `nCONFIG` pin low. The `nCONFIG` pin must be low for at least 500 ns. When `nCONFIG` is pulled low, the Cyclone IV device is reset. The Cyclone IV device also pulls `nSTATUS` and `CONF_DONE` low and all I/O pins are tri-stated. When `nCONFIG` returns to a logic-high level and `nSTATUS` is released by the Cyclone IV device, reconfiguration begins.

Configuration Error

If an error occurs during configuration, Cyclone IV devices assert the `nSTATUS` signal low, indicating a data frame error and the `CONF_DONE` signal stays low. If the **Auto-restart configuration after error** option (available in the Quartus II software in the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Cyclone IV device releases `nSTATUS` after a reset time-out period (a maximum of 230 µs), and retries configuration. If this option is turned off, the system must monitor `nSTATUS` for errors and then pulse `nCONFIG` low for at least 500 ns to restart configuration.

Initialization

In Cyclone IV devices, the initialization clock source is either the internal oscillator or the optional `CLKUSR` pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the device provides itself with enough clock cycles for proper initialization. When using the internal oscillator, you do not have to send additional clock cycles from an external source to the `CLKUSR` pin during the initialization stage. Additionally, you can use the `CLKUSR` pin as a user I/O pin.

You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the **CLKUSR** option. The `CLKUSR` pin allows you to control when your device enters user mode for an indefinite amount of time. You can turn on the **Enable user-supplied start-up clock (CLKUSR)** option in the Quartus II software in the **General** tab of the **Device and Pin Options** dialog box. When you turn on the **Enable user supplied start-up clock option (CLKUSR)** option, the `CLKUSR` pin is the initialization clock source. Supplying a clock on the `CLKUSR` pin does not affect the configuration process. After the configuration data is accepted and `CONF_DONE` goes high, Cyclone IV devices require 3,192 clock cycles to initialize properly and enter user mode.

If you use the optional `CLKUSR` pin and the `nCONFIG` pin is pulled low to restart configuration during device initialization, ensure that the `CLKUSR` pin continues to toggle when `nSTATUS` is low (a maximum of 230 µs).

User Mode

An optional `INIT_DONE` pin is available, which signals the end of initialization and the start of user mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software in the **General** tab of the **Device and Pin Options** dialog box. If you use the `INIT_DONE` pin, it is high due to an external 10-kΩ pull-up resistor when `nCONFIG` is low and during the beginning of configuration. After the option bit to enable `INIT_DONE` is programmed into the device (during the first frame of configuration data), the `INIT_DONE` pin goes low. When initialization is complete, the `INIT_DONE` pin is released and pulled high. This low-to-high transition signals that the device has entered user mode. In user mode, the user I/O pins function as assigned in your design and no longer have weak pull-up resistors.

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`)

Table 10–1. Boundary-Scan Register Length for Cyclone IV Devices (Part 2 of 2)

Device	Boundary-Scan Register Length
EP4CGX75	1006
EP4CGX110	1495
EP4CGX150	1495

Note to Table 10–1:

(1) For the F484 package of the EP4CGX30 device, the boundary-scan register length is 1006.

Table 10–2 lists the `IDCODE` information for Cyclone IV devices.

Table 10–2. IDCODE Information for 32-Bit Cyclone IV Devices

Device	IDCODE (32 Bits) (1)			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) (2)
EP4CE6	0000	0010 0000 1111 0001	000 0110 1110	1
EP4CE10	0000	0010 0000 1111 0001	000 0110 1110	1
EP4CE15	0000	0010 0000 1111 0010	000 0110 1110	1
EP4CE22	0000	0010 0000 1111 0011	000 0110 1110	1
EP4CE30	0000	0010 0000 1111 0100	000 0110 1110	1
EP4CE40	0000	0010 0000 1111 0100	000 0110 1110	1
EP4CE55	0000	0010 0000 1111 0101	000 0110 1110	1
EP4CE75	0000	0010 0000 1111 0110	000 0110 1110	1
EP4CE115	0000	0010 0000 1111 0111	000 0110 1110	1
EP4CGX15	0000	0010 1000 0000 0001	000 0110 1110	1
EP4CGX22	0000	0010 1000 0001 0010	000 0110 1110	1
EP4CGX30 (3)	0000	0010 1000 0000 0010	000 0110 1110	1
EP4CGX30 (4)	0000	0010 1000 0010 0011	000 0110 1110	1
EP4CGX50	0000	0010 1000 0001 0011	000 0110 1110	1
EP4CGX75	0000	0010 1000 0000 0011	000 0110 1110	1
EP4CGX110	0000	0010 1000 0001 0100	000 0110 1110	1
EP4CGX150	0000	0010 1000 0000 0100	000 0110 1110	1

Notes to Table 10–2:

(1) The MSB is on the left.

(2) The `IDCODE` LSB is always 1.

(3) The `IDCODE` is applicable for all packages except for the F484 package.

(4) The `IDCODE` is applicable for the F484 package only.

IEEE Std.1149.6 mandates the addition of two new instructions: `EXTEST_PULSE` and `EXTEST_TRAIN`. These two instructions enable edge-detecting behavior on the signal path containing the AC pins.

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.

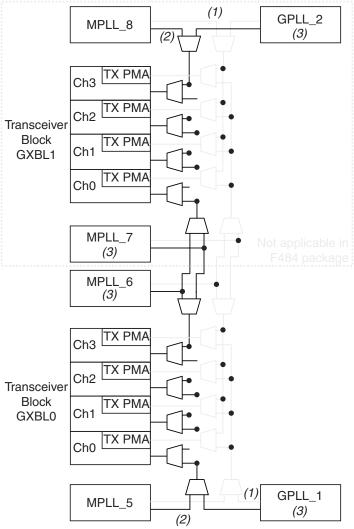
In addition to restoring word boundaries, the word aligner supports the following features:

- 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

Figure 1–32. Clock Distribution in Non-Bonded Channel Configuration for Transceivers in F484 and Larger Packages


MPLL_8
(1)
(2)
GPLL_2
(3)
Transceiver Block GXBL1
Ch3
TX PMA
Ch2
TX PMA
Ch1
TX PMA
Ch0
TX PMA
MPLL_7
(3)
Not applicable in F484 package
MPLL_6
(3)
Transceiver Block GXBL0
Ch3
TX PMA
Ch2
TX PMA
Ch1
TX PMA
Ch0
TX PMA
MPLL_5
(1)
(2)
GPLL_1
(3)

Notes to Figure 1–32:

(1) High-speed clock.

(2) Low-speed clock.

(3) These PLLs have restricted clock driving capability and may not reach all connected channels. For details, refer to Table 1–9.

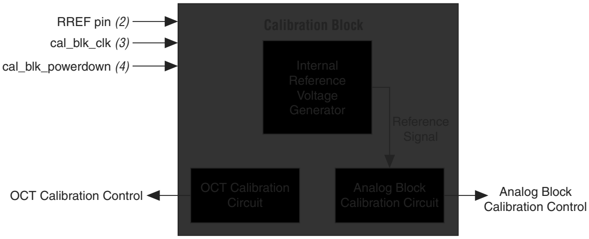
The transceiver datapath clocking varies in non-bonded channel configuration depending on the PCS configuration.

Figure 1–33 shows the datapath clocking in transmitter only operation. In this mode, each channel selects the high- and low-speed clock from one of the supported PLLs. The high-speed clock feeds to the serializer for parallel to serial operation. The low-speed clock feeds to the following blocks in the transmitter PCS:

- 8B/10B encoder
- read clock of the byte serializer
- read clock of the TX phase compensation FIFO

The calibration block internally generates a constant internal reference voltage, independent of PVT variations and uses this voltage and the external reference resistor on the `RREF` pin to generate constant reference currents. The OCT calibration circuit calibrates the OCT resistors present in the transceiver channels. Figure 1–41 shows the calibration block diagram.

Figure 1–41. Input Signals to the Calibration Blocks (1)


RREF pin (2)
cal_blk_clk (3)
cal_blk_powerdown (4)
Calibration Block
Internal Reference Voltage Generator
Reference Signal
OCT Calibration Circuit
Analog Block Calibration Circuit
OCT Calibration Control
Analog Block Calibration Control

Notes to Figure 1–41:

(1) All transceiver channels use the same calibration block clock and power down signals.

(2) Connect a 2 kΩ (tolerance max ± 1%) external resistor to the `RREF` pin to ground. The `RREF` resistor connection in the board must be free from any external noise.

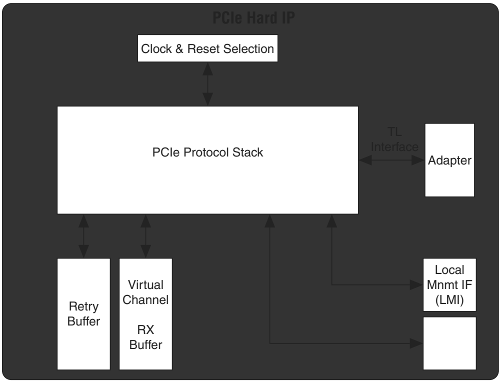
(3) Supports up to 125 MHz clock frequency. Use either dedicated global clock or divide-down logic from the FPGA fabric to generate a slow clock on the local clock routing.

(4) The calibration block restarts the calibration process following deassertion of the `cal_blk_powerdown` signal.

PCI-Express Hard IP Block

Figure 1–42 shows the block diagram of the PCIe hard IP block implementing the PHY MAC, Data Link Layer, and Transaction Layer for PCIe interfaces. The PIPE interface is used as the interface between the transceiver and the hard IP block.

Figure 1–42. PCI Express Hard IP High-Level Block Diagram

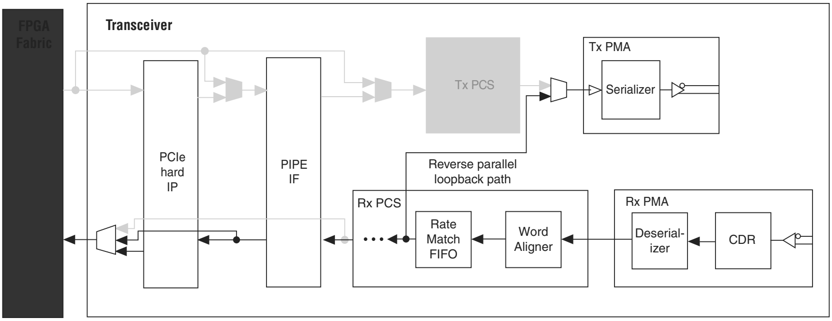

PCIe Hard IP
Clock & Reset Selection
PCIe Protocol Stack
TL Interface
Adapter
Retry Buffer
Virtual Channel
RX Buffer
Local Mnmt IF (LMI)

Reverse Parallel Loopback

The reverse parallel loopback option is only available for PIPE mode. In this mode, the received serial data passes through the receiver CDR, deserializer, word aligner, and rate match FIFO before looping back to the transmitter serializer and transmitted out through the TX buffer, as shown in Figure 1–70. The received data is also available to the FPGA fabric. This loopback mode is compliant with version 2.00 of the *PHY Interface for the PCI Express Architecture* specification.

To enable the reverse parallel loopback mode, assert the `tx_detectrxloopback` port in P0 power state.

Figure 1–70. PIPE Reverse Parallel Loopback Path (1)



Note to Figure 1–70:

(1) Grayed-Out Blocks are Not Active in this mode.

Serial Loopback

The serial loopback option is available for all functional modes except PIPE mode. In this mode, the data from the FPGA fabric passes through the transmitter channel and looped back to the receiver channel, bypassing the receiver buffer, as shown in Figure 1–71. The received data is available to the FPGA logic for verification. The receiver input buffer is not active in this mode. With this option, you can check the operation of all enabled PCS and PMA functional blocks in the transmitter and receiver channels.

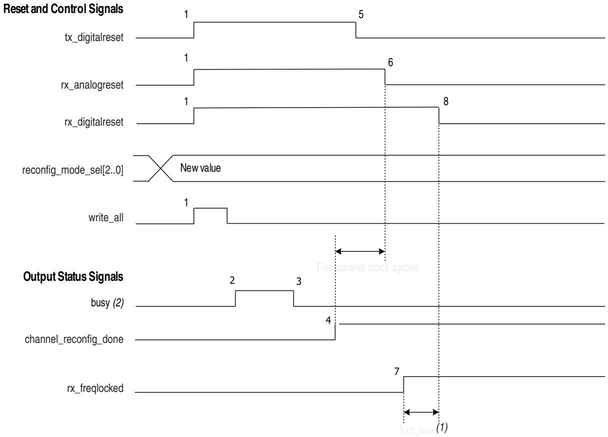
The transmitter channel sends the data to both the serial output port and the receiver channel. The differential output voltage on the serial ports is based on the selected V_{OD} settings. The data is looped back to the receiver CDR and is retimed through different clock domains. You must provide an alignment pattern for the word aligner to enable the receiver channel to retrieve the byte boundary.

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


Reset and Control Signals
tx_digitalreset
1
5
rx_analogreset
1
6
rx_digitalreset
1
8
reconfig_mode_sel[2..0]
New value
write_all
1
Five parallel clock cycles
Output Status Signals
busy (2)
2
3
channel_reconfig_done
4
rx_freqlocked
7
t_{LTD_Auto} (1)

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.

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`).
r	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.
1	The hand points to information that requires special attention.
h	The question mark directs you to a software help system with related information.
f	The feet direct you to another document or website with related information.
m	The multimedia icon directs you to a related multimedia presentation.
c	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
w	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 1–47. Document Revision History

Date	Version	Changes
February 2010	1.1	■ Updated Table 1–3 through Table 1–44 to include information for Cyclone IV E devices and Cyclone IV GX devices for Quartus II software version 9.1 SP1 release. ■ Minor text edits.
November 2009	1.0	Initial release.

Section I. Device Datasheet

This section provides the Cyclone® IV device datasheet. It includes the following chapter:

- Chapter 1, Cyclone IV Device Datasheet

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.