Cyclone IV Device Handbook,

Volume 1

ISO
9001:2008
Registered

Contents

Section I. Device Core

Chapter 4. Embedded Multipliers in Cyclone IV Devices

Chapter 5. Clock Networks and PLLs in Cyclone IV Devices

Section II. I/O Interfaces

Chapter 6. I/O Features in Cyclone IV Devices

Chapter 7. External Memory Interfaces in Cyclone IV Devices

Section III. System Integration

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

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*

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."

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

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:

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.

Altera's new Cyclone® IV FPGA device family extends the Cyclone FPGA series leadership in providing the market's lowest-cost, lowest-power FPGAs, now with a transceiver variant. Cyclone IV devices are targeted to high-volume, cost-sensitive applications, enabling system designers to meet increasing bandwidth requirements while lowering costs.

Built on an optimized low-power process, the Cyclone IV device family offers the following two variants:

- Cyclone IV E—lowest power, high functionality with the lowest cost

- Cyclone IV GX—lowest power and lowest cost FPGAs with 3.125 Gbps transceivers

    ☞ Cyclone IV E devices are offered in core voltage of 1.0 V and 1.2 V.

    ✎ For more information, refer to the *Power Requirements for Cyclone IV Devices* chapter.

Providing power and cost savings without sacrificing performance, along with a low-cost integrated transceiver option, Cyclone IV devices are ideal for low-cost, small-form-factor applications in the wireless, wireline, broadcast, industrial, consumer, and communications industries.

Cyclone IV Device Family Features

The Cyclone IV device family offers the following features:

- Low-cost, low-power FPGA fabric:

    - 6K to 150K logic elements

    - Up to 6.3 Mb of embedded memory

    - Up to 360 18 × 18 multipliers for DSP processing intensive applications

    - Protocol bridging applications for under 1.5 W total power

Feedback Subscribe

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

- Up to 532 user I/Os
    - LVDS interfaces up to 840 Mbps transmitter (Tx), 875 Mbps Rx
    - Support for DDR2 SDRAM interfaces up to 200 MHz
    - Support for QDRII SRAM and DDR SDRAM up to 167 MHz
- Up to eight phase-locked loops (PLLs) per device
- Offered in commercial and industrial temperature grades

Device Resources

Table 1–1 lists Cyclone IV E device resources.

Table 1–1. Resources for the Cyclone IV E Device Family

Resources	EP4CE6	EP4CE10	EP4CE15	EP4CE22	EP4CE30	EP4CE40	EP4CE55	EP4CE75	EP4CE115
Logic elements (LEs)	6,272	10,320	15,408	22,320	28,848	39,600	55,856	75,408	114,480
Embedded memory (Kbits)	270	414	504	594	594	1,134	2,340	2,745	3,888
Embedded 18 × 18 multipliers	15	23	56	66	66	116	154	200	266
General-purpose PLLs	2	2	4	4	4	4	4	4	4
Global Clock Networks	10	10	20	20	20	20	20	20	20
User I/O Banks	8	8	8	8	8	8	8	8	8
Maximum user I/O [1]	179	179	343	153	532	532	374	426	528

Note to Table 1–1:

(1) 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.

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.

Package Matrix

Table 1–3 lists Cyclone IV E device package offerings.

Table 1–3. Package Offerings for the Cyclone IV E Device Family [(1), (2)]

Package	E144		M164		M256		U256		F256		F324		U484		F484		F780	
Size (mm)	22 × 22		8 × 8		9 x 9		14 × 14		17 × 17		19 x 19		19 × 19		23 × 23		29 × 29	
Pitch (mm)	0.5		0.5		0.5		0.8		1.0		1.0		0.8		1.0		1.0	
Device	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]	User I/O	LVDS [(3)]
EP4CE6	91	21	—	—	—	—	179	66	179	66	—	—	—	—	—	—	—	—
EP4CE10	91	21	—	—	—	—	179	66	179	66	—	—	—	—	—	—	—	—
EP4CE15	81	18	89	21	165	53	165	53	165	53	—	—	—	—	343	137	—	—
EP4CE22	79	17	—	—	—	—	153	52	153	52	—	—	—	—	—	—	—	—
EP4CE30	—	—	—	—	—	—	—	—	—	—	193	68	—	—	328	124	532	224
EP4CE40	—	—	—	—	—	—	—	—	—	—	193	68	328	124	328	124	532	224
EP4CE55	—	—	—	—	—	—	—	—	—	—	—	—	324	132	324	132	374	160
EP4CE75	—	—	—	—	—	—	—	—	—	—	—	—	292	110	292	110	426	178
EP4CE115	—	—	—	—	—	—	—	—	—	—	—	—	—	—	280	103	528	230

Notes to Table 1–3:

(1) The E144 package has an exposed pad at the bottom of the package. This exposed pad is a ground pad that must be connected to the ground plane of your PCB. Use this exposed pad for electrical connectivity and not for thermal purposes.

(2) Use the Pin Migration View window in Pin Planner of the Quartus II software to verify the pin migration compatibility when you perform device migration. For more information, refer to the *I/O Management* chapter in volume 2 of the *Quartus II Handbook*.

(3) This includes both dedicated and emulated LVDS pairs. For more information, refer to the *I/O Features in Cyclone IV Devices* chapter.

Table 1–4 lists Cyclone IV GX device package offerings, including I/O and transceiver counts.

Table 1–4. Package Offerings for the Cyclone IV GX Device Family [1]

Package	F169			F324			F484			F672			F896		
Size (mm)	14 × 14			19 × 19			23 × 23			27 × 27			31 × 31		
Pitch (mm)	1.0			1.0			1.0			1.0			1.0		
Device	User I/O	LVDS [2]	XCVRs	User I/O	LVDS [2]	XCVRs	User I/O	LVDS [2]	XCVRs	User I/O	LVDS [2]	XCVRs	User I/O	LVDS [2]	XCVR
EP4CGX15	▲72	25	2	—	—	—	—	—	—	—	—	—	—	—	—
EP4CGX22	72	25	2	▲150	64	4	—	—	—	—	—	—	—	—	—
EP4CGX30	▼72	25	2	▼150	64	4	▲290	130	4	—	—	—	—	—	—
EP4CGX50	—	—	—	—	—	—	290	130	4	▲310	140	8	—	—	—
EP4CGX75	—	—	—	—	—	—	290	130	4	310	140	8	—	—	—
EP4CGX110	—	—	—	—	—	—	270	120	4	393	181	8	▲475	220	8
EP4CGX150	—	—	—	—	—	—	▼270	120	4	▼393	181	8	▼475	220	8

Note to Table 1–4:

(1) Use the Pin Migration View window in Pin Planner of the Quartus II software to verify the pin migration compatibility when you perform device migration. For more information, refer to the *I/O Management* chapter in volume 2 of the *Quartus II Handbook*.

(2) This includes both dedicated and emulated LVDS pairs. For more information, refer to the *I/O Features in Cyclone IV Devices* chapter.

Cyclone IV Device Family Speed Grades

Table 1–5 lists the Cyclone IV GX devices speed grades.

Table 1–5. Speed Grades for the Cyclone IV GX Device Family

Device	F169	F324	F484	F672	F896
EP4CGX15	C6, C7, C8, I7	—	—	—	—
EP4CGX22	C6, C7, C8, I7	C6, C7, C8, I7	—	—	—
EP4CGX30	C6, C7, C8, I7	C6, C7, C8, I7	C6, C7, C8, I7	—	—
EP4CGX50	—	—	C6, C7, C8, I7	C6, C7, C8, I7	—
EP4CGX75	—	—	C6, C7, C8, I7	C6, C7, C8, I7	—
EP4CGX110	—	—	C7, C8, I7	C7, C8, I7	C7, C8, I7
EP4CGX150	—	—	C7, C8, I7	C7, C8, I7	C7, C8, I7

Table 1–6 lists the Cyclone IV E devices speed grades.

Table 1–6. Speed Grades for the Cyclone IV E Device Family [1], [2]

Device	E144	M164	M256	U256	F256	F324	U484	F484	F780
EP4CE6	C8L, C9L, I8L C6, C7, C8, I7, A7	—	—	I7N	C8L, C9L, I8L C6, C7, C8, I7, A7	—	—	—	—
EP4CE10	C8L, C9L, I8L C6, C7, C8, I7, A7	—	—	I7N	C8L, C9L, I8L C6, C7, C8, I7, A7	—	—	—	—
EP4CE15	C8L, C9L, I8L C6, C7, C8, I7	I7N	C7N, I7N	I7N	C8L, C9L, I8L C6, C7, C8, I7, A7	—	—	C8L, C9L, I8L C6, C7, C8, I7, A7	—
EP4CE22	C8L, C9L, I8L C6, C7, C8, I7, A7	—	—	I7N	C8L, C9L, I8L C6, C7, C8, I7, A7	—	—	—	—
EP4CE30	—	—	—	—	—	A7N	—	C8L, C9L, I8L C6, C7, C8, I7, A7	C8L, C9L, I8L C6, C7, C8, I7
EP4CE40	—	—	—	—	—	A7N	I7N	C8L, C9L, I8L C6, C7, C8, I7, A7	C8L, C9L, I8L C6, C7, C8, I7
EP4CE55	—	—	—	—	—	—	I7N	C8L, C9L, I8L C6, C7, C8, I7	C8L, C9L, I8L C6, C7, C8, I7
EP4CE75	—	—	—	—	—	—	I7N	C8L, C9L, I8L C6, C7, C8, I7	C8L, C9L, I8L C6, C7, C8, I7
EP4CE115	—	—	—	—	—	—	—	C8L, C9L, I8L C7, C8, I7	C8L, C9L, I8L C7, C8, I7

Notes to Table 1–6:

(1) C8L, C9L, and I8L speed grades are applicable for the 1.0-V core voltage.

(2) C6, C7, C8, I7, and A7 speed grades are applicable for the 1.2-V core voltage.

Cyclone IV Device Family Architecture

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

- "FPGA Core Fabric"

- "I/O Features"

- "Clock Management"

- "External Memory Interfaces"

- "Configuration"

- "High-Speed Transceivers (Cyclone IV GX Devices Only)"

- "Hard IP for PCI Express (Cyclone IV GX Devices Only)"

FPGA Core Fabric

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

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

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

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

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

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

I/O Features

Cyclone IV device I/O supports programmable bus hold, programmable pull-up resistors, programmable delay, programmable drive strength, programmable slew-rate control to optimize signal integrity, and hot socketing. Cyclone IV devices support calibrated on-chip series termination (Rs OCT) or driver impedance matching (Rs) for single-ended I/O standards. In Cyclone IV GX devices, the high-speed transceiver I/Os are located on the left side of the device. The top, bottom, and right sides can implement general-purpose user I/Os.

Table 1–8 lists the I/O standards that Cyclone IV devices support.

Table 1–8. I/O Standards Support for the Cyclone IV Device Family

Type	I/O Standard
Single-Ended I/O	LVTTL, LVCMOS, SSTL, HSTL, PCI, and PCI-X
Differential I/O	SSTL, HSTL, LVPECL, BLVDS, LVDS, mini-LVDS, RSDS, and PPDS

The LVDS SERDES is implemented in the core of the device using logic elements.

For more information, refer to the *I/O Features in Cyclone IV Devices* chapter.

Clock Management

Cyclone IV devices include up to 30 global clock (GCLK) networks and up to eight PLLs with five outputs per PLL to provide robust clock management and synthesis. You can dynamically reconfigure Cyclone IV device PLLs in user mode to change the clock frequency or phase.

Cyclone IV GX devices support two types of PLLs: multipurpose PLLs and general-purpose PLLs:

■ Use multipurpose PLLs for clocking the transceiver blocks. You can also use them for general-purpose clocking when they are not used for transceiver clocking.

■ Use general purpose PLLs for general-purpose applications in the fabric and periphery, such as external memory interfaces. Some of the general purpose PLLs can support transceiver clocking.

For more information, refer to the *Clock Networks and PLLs in Cyclone IV Devices* chapter.

External Memory Interfaces

Cyclone IV devices support SDR, DDR, DDR2 SDRAM, and QDRII SRAM interfaces on the top, bottom, and right sides of the device. Cyclone IV E devices also support these interfaces on the left side of the device. Interfaces may span two or more sides of the device to allow more flexible board design. The Altera® DDR SDRAM memory interface solution consists of a PHY interface and a memory controller. Altera supplies the PHY IP and you can use it in conjunction with your own custom memory controller or an Altera-provided memory controller. Cyclone IV devices support the use of error correction coding (ECC) bits on DDR and DDR2 SDRAM interfaces.

For more information, refer to the *External Memory Interfaces in Cyclone IV Devices* chapter.

Configuration

Cyclone IV devices use SRAM cells to store configuration data. Configuration data is downloaded to the Cyclone IV device each time the device powers up. Low-cost configuration options include the Altera EPCS family serial flash devices and commodity parallel flash configuration options. These options provide the flexibility for general-purpose applications and the ability to meet specific configuration and wake-up time requirements of the applications.

Table 1–9 lists which configuration schemes are supported by Cyclone IV devices.

Table 1–9. Configuration Schemes for Cyclone IV Device Family

Devices	Supported Configuration Scheme
Cyclone IV GX	AS, PS, JTAG, and FPP [1]
Cyclone IV E	AS, AP, PS, FPP, and JTAG

Note to Table 1–9:

(1) The FPP configuration scheme is only supported by the EP4CGX30F484 and EP4CGX50/75/110/150 devices.

IEEE 1149.6 (AC JTAG) is supported on all transceiver I/O pins. All other pins support IEEE 1149.1 (JTAG) for boundary scan testing.

For more information, refer to the *JTAG Boundary-Scan Testing for Cyclone IV Devices* chapter.

For Cyclone IV GX devices to meet the PCIe 100 ms wake-up time requirement, you must use passive serial (PS) configuration mode for the EP4CGX15/22/30 devices and use fast passive parallel (FPP) configuration mode for the EP4CGX30F484 and EP4CGX50/75/110/150 devices.

For more information, refer to the *Configuration and Remote System Upgrades in Cyclone IV Devices* chapter.

The cyclical redundancy check (CRC) error detection feature during user mode is supported in all Cyclone IV GX devices. For Cyclone IV E devices, this feature is only supported for the devices with the core voltage of 1.2 V.

For more information about CRC error detection, refer to the *SEU Mitigation in Cyclone IV Devices* chapter.

High-Speed Transceivers (Cyclone IV GX Devices Only)

Cyclone IV GX devices contain up to eight full duplex high-speed transceivers that can operate independently. These blocks support multiple industry-standard communication protocols, as well as Basic mode, which you can use to implement your own proprietary protocols. Each transceiver channel has its own pre-emphasis and equalization circuitry, which you can set at compile time to optimize signal integrity and reduce bit error rates. Transceiver blocks also support dynamic reconfiguration, allowing you to change data rates and protocols on-the-fly.

Figure 1–1 shows the structure of the Cyclone IV GX transceiver.

Figure 1–1. Transceiver Channel for the Cyclone IV GX Device
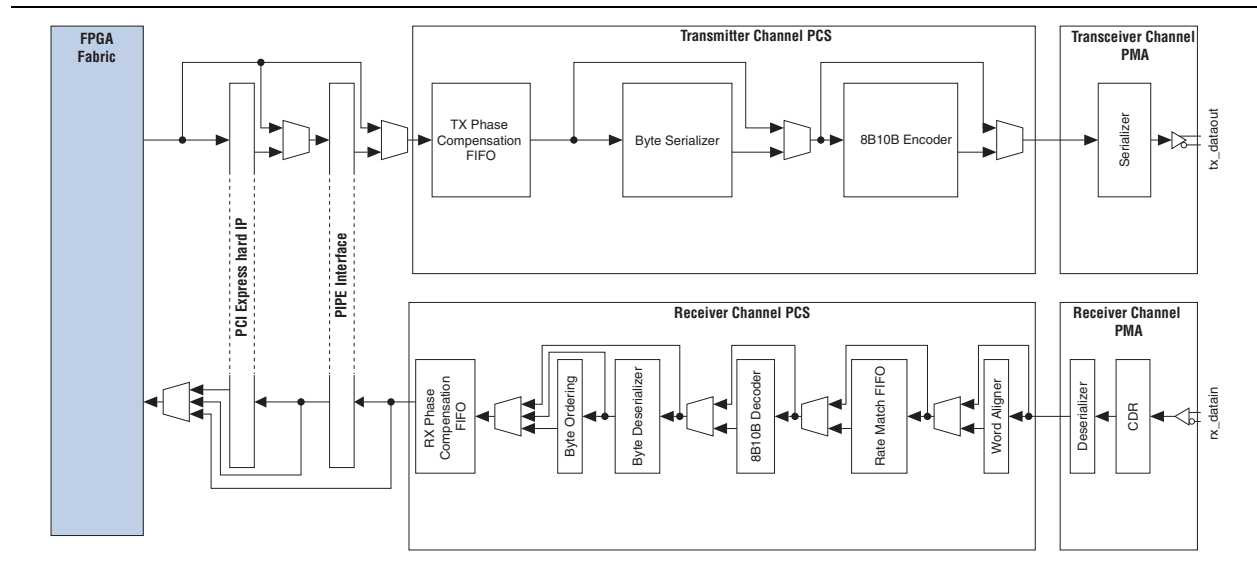
> For more information, refer to the *Cyclone IV Transceivers Architecture* chapter.

Hard IP for PCI Express (Cyclone IV GX Devices Only)

Cyclone IV GX devices incorporate a single hard IP block for ×1, ×2, or ×4 PCIe (PIPE) in each device. This hard IP block is a complete PCIe (PIPE) protocol solution that implements the PHY-MAC layer, Data Link Layer, and Transaction Layer functionality. The hard IP for the PCIe (PIPE) block supports root-port and end-point configurations. This pre-verified hard IP block reduces risk, design time, timing closure, and verification. You can configure the block with the Quartus II software's PCI Express Compiler, which guides you through the process step by step.

> For more information, refer to the *PCI Express Compiler User Guide*.

Reference and Ordering Information

Figure 1–2 shows the ordering codes for Cyclone IV GX devices.

Figure 1–2. Packaging Ordering Information for the Cyclone IV GX Device

Member Code
- 15 : 14,400 logic elements
- 22 : 21,280 logic elements
- 30 : 29,440 logic elements
- 50 : 49,888 logic elements
- 75 : 73,920 logic elements
- 110 : 109,424 logic elements
- 150 : 149,760 logic elements

Package Type
- F : FineLine BGA (FBGA)
- N : Quad Flat Pack No Lead (QFN)

Operating Temperature
- C : Commercial temperature (T_J = 0° C to 85° C)
- I : Industrial temperature (T_J = -40° C to 100° C)

EP4C GX 30 C F 19 C 7 N

Family Signature
EP4C : Cyclone IV

Family Variant
GX : 3-Gbps transceivers

Transceiver Count
- B : 2
- C : 4
- D : 8

Package Code
FBGA Package Type
- 14 : 169 pins
- 19 : 324 pins
- 23 : 484 pins
- 27 : 672 pins
- 31 : 896 pins

Optional Suffix
Indicates specific device options or shipment method
- N : Lead-free packaging
- ES : Engineering sample

Speed Grade
- 6 (fastest)
- 7
- 8

Figure 1–3 shows the ordering codes for Cyclone IV E devices.

Figure 1–3. Packaging Ordering Information for the Cyclone IV E Device

Package Type
- F : FineLine BGA (FBGA)
- E : Enhanced Thin Quad Flat Pack (EQFP)
- U : Ultra FineLine BGA (UBGA)
- M : Micro FineLine BGA (MBGA)

Operating Temperature
- C : Commercial temperature (T_J = 0° C to 85° C)
- I : Industrial temperature (T_J = -40° C to 100° C)
  Extended industrial temperature (T_J = -40° C to 125° C)
- A : Automotive temperature (T_J = -40° C to 125° C)

Family Variant
E : Enhanced logic/memory

EP4C E 40 F 29 C 8 N

Family Signature
EP4C : Cyclone IV

Member Code
- 6 : 6,272 logic elements
- 10 : 10,320 logic elements
- 15 : 15,408 logic elements
- 22 : 22,320 logic elements
- 30 : 28,848 logic elements
- 40 : 39,600 logic elements
- 55 : 55,856 logic elements
- 75 : 75,408 logic elements
- 115 : 114,480 logic elements

Package Code
FBGA Package Type
- 17 : 256 pins
- 19 : 324 pins
- 23 : 484 pins
- 29 : 780 pins

EQFP Package Type
- 22 : 144 pins

UBGA Package Type
- 14 : 256 pins
- 19 : 484 pins

MBGA Package Type
- 8 : 164 pins
- 9 : 256 pins

Optional Suffix
Indicates specific device options or shipment method
- N : Lead-free packaging
- ES : Engineering sample
- L : Low-voltage device

Speed Grade
- 6 (fastest)
- 7
- 8
- 9

Document Revision History

Table 1–10 lists the revision history for this chapter.

Table 1–10. Document Revision History

Date	Version	Changes
March 2016	2.0	■ Updated Table 1–4 and Table 1–5 to remove support for the N148 package. ■ Updated Figure 1–2 to remove support for the N148 package.
April 2014	1.9	Updated "Packaging Ordering Information for the Cyclone IV E Device".
May 2013	1.8	Updated Table 1–3, Table 1–6 and Figure 1–3 to add new device options and packages.
February 2013	1.7	Updated Table 1–3, Table 1–6 and Figure 1–3 to add new device options and packages.
October 2012	1.6	Updated Table 1–3 and Table 1–4.
November 2011	1.5	■ Updated "Cyclone IV Device Family Features" section. ■ Updated Figure 1–2 and Figure 1–3.
December 2010	1.4	■ Updated for the Quartus II software version 10.1 release. ■ Added Cyclone IV E new device package information. ■ Updated Table 1–1, Table 1–2, Table 1–3, Table 1–5, and Table 1–6. ■ Updated Figure 1–3. ■ Minor text edits.
July 2010	1.3	Updated Table 1–2 to include F484 package information.
March 2010	1.2	■ Updated Table 1–3 and Table 1–6. ■ Updated Figure 1–3. ■ Minor text edits.
February 2010	1.1	■ Added Cyclone IV E devices in Table 1–1, Table 1–3, and Table 1–6 for the Quartus II software version 9.1 SP1 release. ■ Added the "Cyclone IV Device Family Speed Grades" and "Configuration" sections. ■ Added Figure 1–3 to include Cyclone IV E Device Packaging Ordering Information. ■ Updated Table 1–2, Table 1–4, and Table 1–5 for Cyclone IV GX devices. ■ Minor text edits.
November 2009	1.0	Initial release.

This chapter contains feature definitions for logic elements (LEs) and logic array blocks (LABs). Details are provided on how LEs work, how LABs contain groups of LEs, and how LABs interface with the other blocks in Cyclone® IV devices.

Logic Elements

Logic elements (LEs) are the smallest units of logic in the Cyclone IV device architecture. LEs are compact and provide advanced features with efficient logic usage. Each LE has the following features:

■ A four-input look-up table (LUT), which can implement any function of four variables

■ A programmable register

■ A carry chain connection

■ A register chain connection

■ The ability to drive the following interconnects:

  ■ Local

  ■ Row

  ■ Column

  ■ Register chain

  ■ Direct link

■ Register packing support

■ Register feedback support

Feedback Subscribe

Figure 2–1 shows the LEs for Cyclone IV devices.

Figure 2–1. Cyclone IV Device LEs
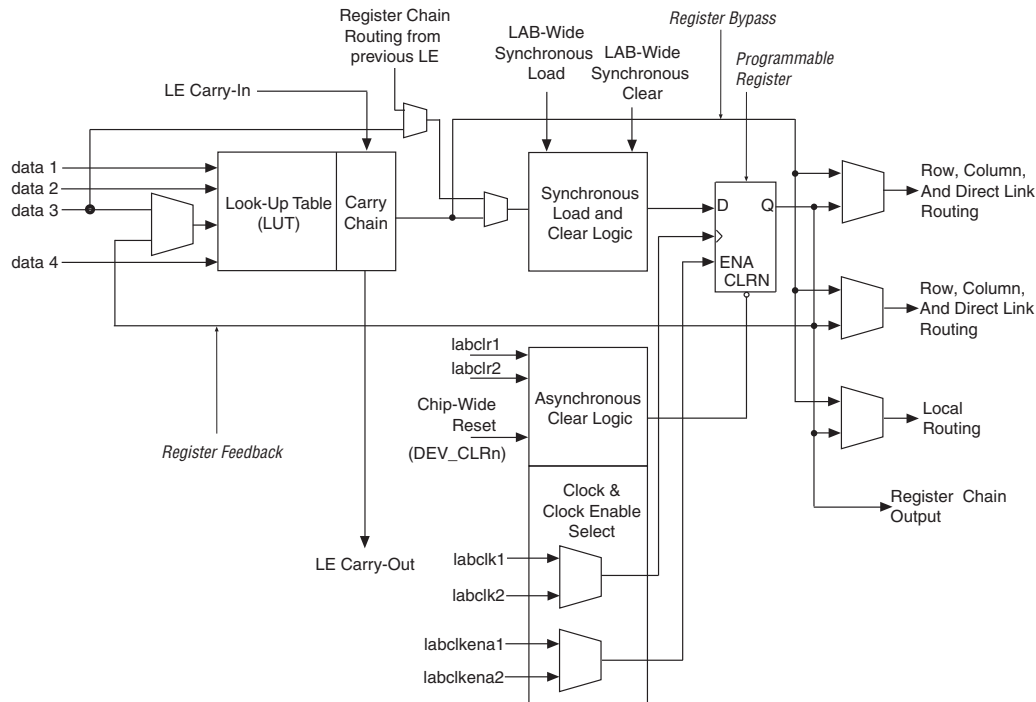
LE Features

You can configure the programmable register of each LE for D, T, JK, or SR flipflop operation. Each register has data, clock, clock enable, and clear inputs. Signals that use the global clock network, general-purpose I/O pins, or any internal logic can drive the clock and clear control signals of the register. Either general-purpose I/O pins or the internal logic can drive the clock enable. For combinational functions, the LUT output bypasses the register and drives directly to the LE outputs.

Each LE has three outputs that drive the local, row, and column routing resources. The LUT or register output independently drives these three outputs. Two LE outputs drive the column or row and direct link routing connections, while one LE drives the local interconnect resources. This allows the LUT to drive one output while the register drives another output. This feature, called register packing, improves device utilization because the device can use the register and the LUT for unrelated functions. The LAB-wide synchronous load control signal is not available when using register packing. For more information about the synchronous load control signal, refer to "LAB Control Signals" on page 2–6.

The register feedback mode allows the register output to feed back into the LUT of the same LE to ensure that the register is packed with its own fan-out LUT, providing another mechanism for improved fitting. The LE can also drive out registered and unregistered versions of the LUT output.

In addition to the three general routing outputs, LEs in an LAB have register chain outputs, which allows registers in the same LAB to cascade together. The register chain output allows the LUTs to be used for combinational functions and the registers to be used for an unrelated shift register implementation. These resources speed up connections between LABs while saving local interconnect resources.

LE Operating Modes

Cyclone IV LEs operate in the following modes:

■ Normal mode

■ Arithmetic mode

The Quartus® II software automatically chooses the appropriate mode for common functions, such as counters, adders, subtractors, and arithmetic functions, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions. You can also create special-purpose functions that specify which LE operating mode to use for optimal performance, if required.

Normal Mode

Normal mode is suitable for general logic applications and combinational functions. In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT (Figure 2–2). The Quartus II Compiler automatically selects the carry-in (cin) or the data3 signal as one of the inputs to the LUT. LEs in normal mode support packed registers and register feedback.

Figure 2–2 shows LEs in normal mode.

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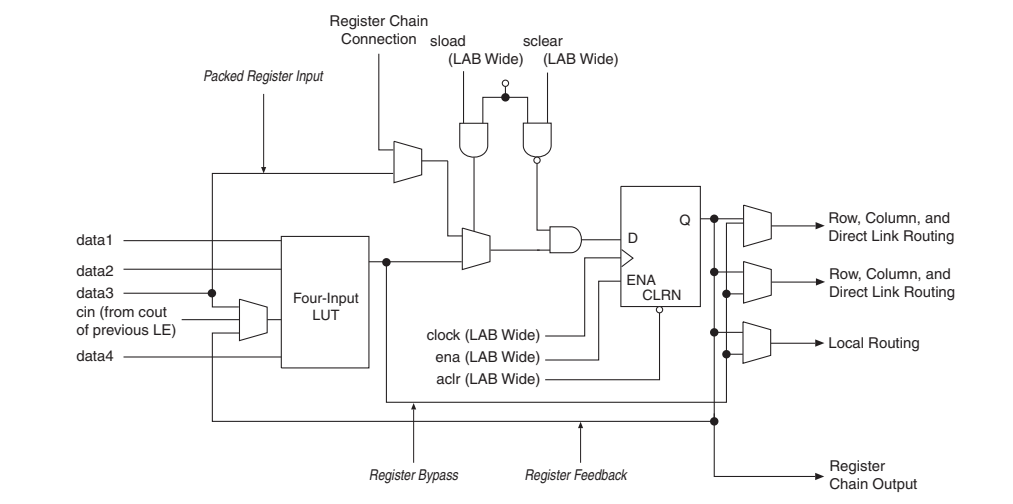
Arithmetic Mode

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

Figure 2–3 shows LEs in arithmetic mode.

Figure 2–3. Cyclone IV Device LEs in Arithmetic Mode
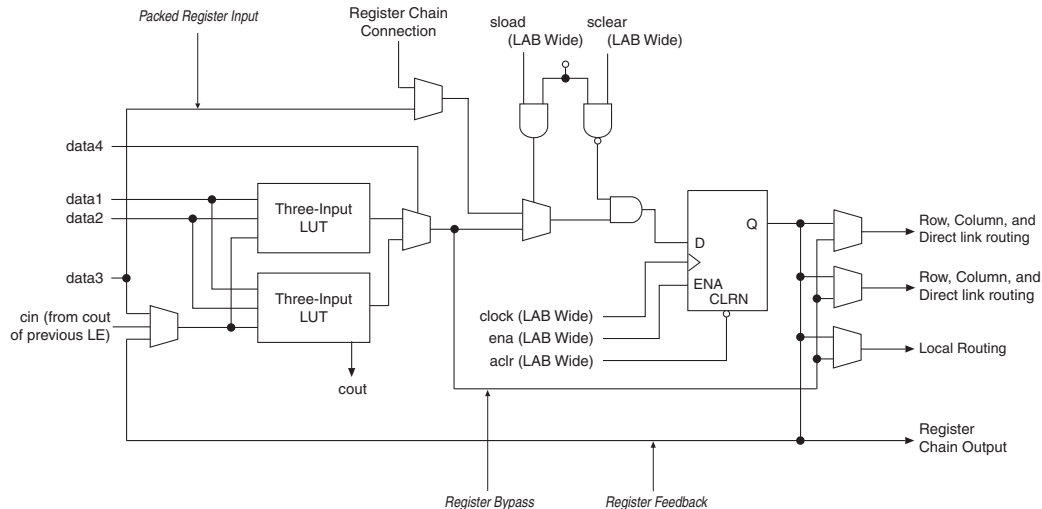
The Quartus II Compiler automatically creates carry chain logic during design processing. You can also manually create the carry chain logic during design entry. Parameterized functions, such as LPM functions, automatically take advantage of carry chains for the appropriate functions.

The Quartus II Compiler creates carry chains longer than 16 LEs by automatically linking LABs in the same column. For enhanced fitting, a long carry chain runs vertically, which allows fast horizontal connections to M9K memory blocks or embedded multipliers through direct link interconnects. For example, if a design has a long carry chain in an LAB column next to a column of M9K memory blocks, any LE output can feed an adjacent M9K memory block through the direct link interconnect. If the carry chains run horizontally, any LAB which is not next to the column of M9K memory blocks uses other row or column interconnects to drive a M9K memory block. A carry chain continues as far as a full column.

Logic Array Blocks

Logic array blocks (LABs) contain groups of LEs.

Topology

Each LAB consists of the following features:

■ 16 LEs

■ LAB control signals

■ LE carry chains

■ Register chains

■ Local interconnect

The local interconnect transfers signals between LEs in the same LAB. Register chain connections transfer the output of one LE register to the adjacent LE register in an LAB. The Quartus II Compiler places associated logic in an LAB or adjacent LABs, allowing the use of local and register chain connections for performance and area efficiency.

Figure 2–4 shows the LAB structure for Cyclone IV devices.

Figure 2–4. Cyclone IV Device LAB Structure
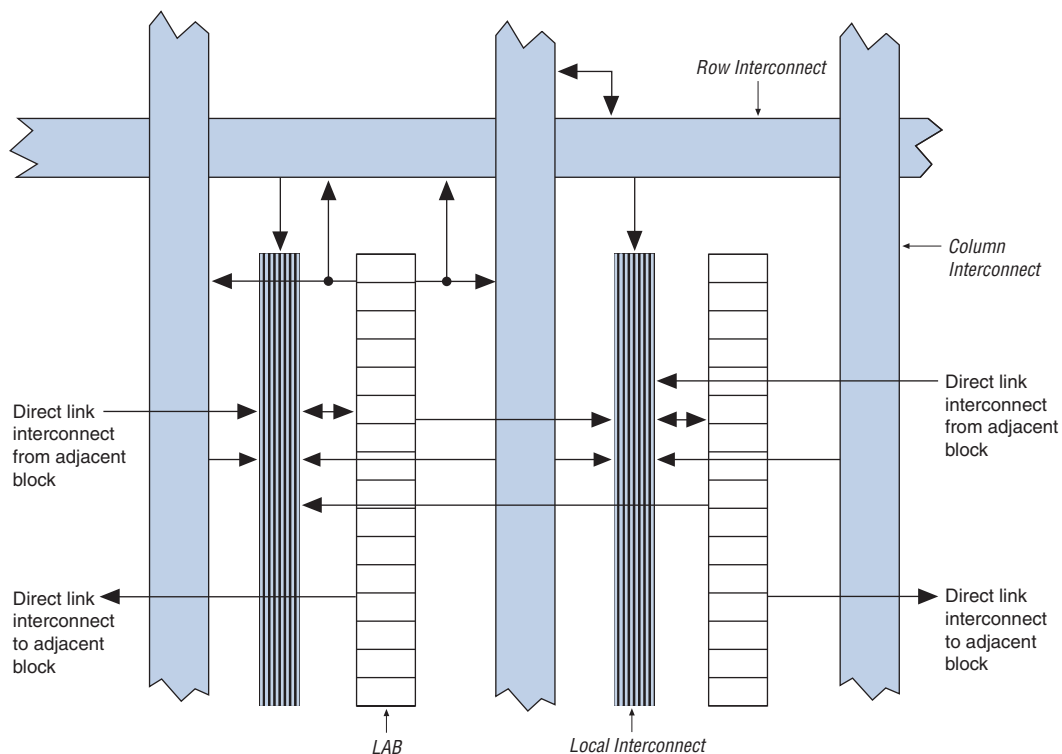
LAB Interconnects

The LAB local interconnect is driven by column and row interconnects and LE outputs in the same LAB. Neighboring LABs, phase-locked loops (PLLs), M9K RAM blocks, and embedded multipliers from the left and right can also drive the local interconnect of a LAB through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each LE can drive up to 48 LEs through fast local and direct link interconnects.

Figure 2–5 shows the direct link connection.

Figure 2–5. Cyclone IV Device Direct Link Connection
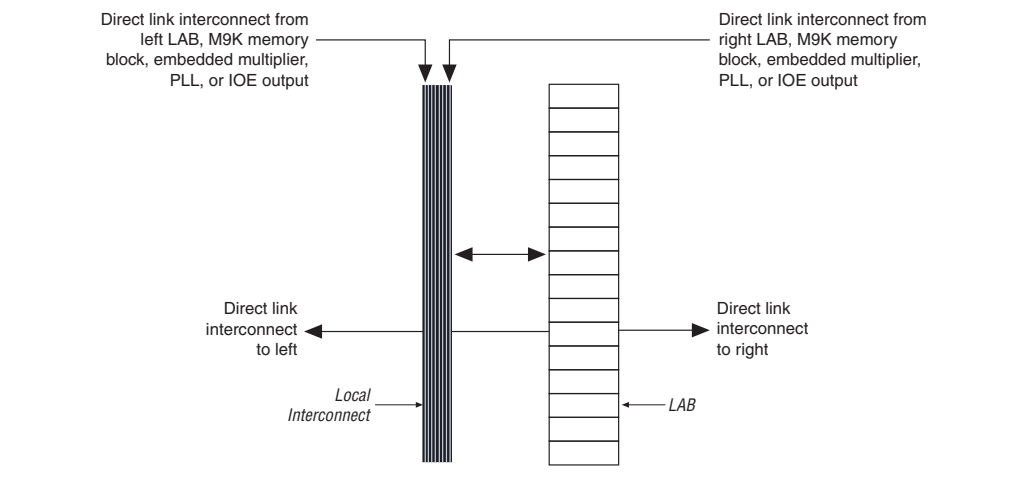
LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its LEs. The control signals include:

- Two clocks

- Two clock enables

- Two asynchronous clears

- One synchronous clear

- One synchronous load

You can use up to eight control signals at a time. Register packing and synchronous load cannot be used simultaneously.

Each LAB can have up to four non-global control signals. You can use additional LAB control signals as long as they are global signals.

Synchronous clear and load signals are useful for implementing counters and other functions. The synchronous clear and synchronous load signals are LAB-wide signals that affect all registers in the LAB.

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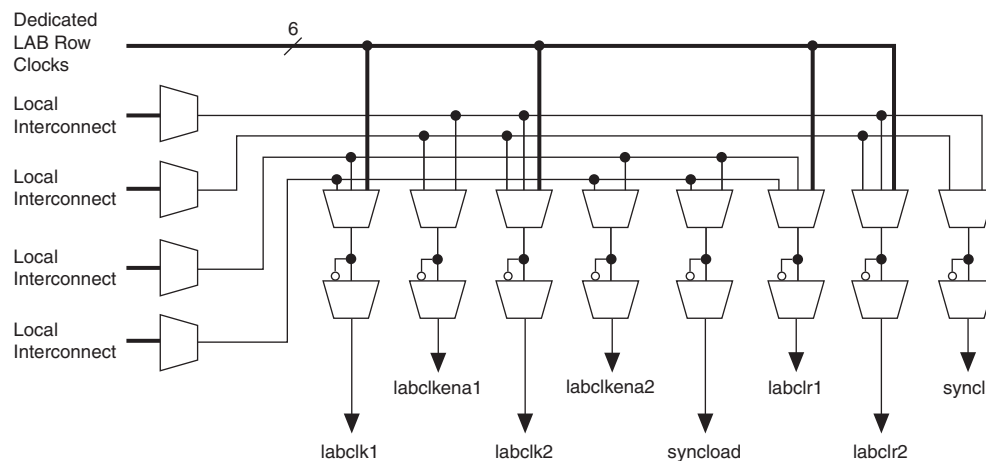
LAB-wide signals control the logic for the clear signal of the register. The LE directly supports an asynchronous clear function. Each LAB supports up to two asynchronous clear signals (`labclr1` and `labclr2`).

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.

Cyclone® IV devices feature embedded memory structures to address the on-chip memory needs of Altera® Cyclone IV device designs. The embedded memory structure consists of columns of M9K memory blocks that you can configure to provide various memory functions, such as RAM, shift registers, ROM, and FIFO buffers.

This chapter contains the following sections:

- "Memory Modes" on page 3–7
- "Clocking Modes" on page 3–14
- "Design Considerations" on page 3–15

Overview

M9K blocks support the following features:

- 8,192 memory bits per block (9,216 bits per block including parity)
- Independent read-enable (rden) and write-enable (wren) signals for each port
- Packed mode in which the M9K memory block is split into two 4.5 K single-port RAMs
- Variable port configurations
- Single-port and simple dual-port modes support for all port widths
- True dual-port (one read and one write, two reads, or two writes) operation
- Byte enables for data input masking during writes
- Two clock-enable control signals for each port (port A and port B)
- Initialization file to pre-load memory content in RAM and ROM modes

Feedback   Subscribe

Table 3–1 lists the features supported by the M9K memory.

Table 3–1. Summary of M9K Memory Features

Feature	M9K Blocks
Configurations (depth × width)	8192 × 1 4096 × 2 2048 × 4 1024 × 8 1024 × 9 512 × 16 512 × 18 256 × 32 256 × 36
Parity bits	✓
Byte enable	✓
Packed mode	✓
Address clock enable	✓
Single-port mode	✓
Simple dual-port mode	✓
True dual-port mode	✓
Embedded shift register mode [1]	✓
ROM mode	✓
FIFO buffer [1]	✓
Simple dual-port mixed width support	✓
True dual-port mixed width support [2]	✓
Memory initialization file (**.mif**)	✓
Mixed-clock mode	✓
Power-up condition	Outputs cleared
Register asynchronous clears	Read address registers and output registers only
Latch asynchronous clears	Output latches only
Write or read operation triggering	Write and read: Rising clock edges
Same-port read-during-write	Outputs set to **Old Data** or **New Data**
Mixed-port read-during-write	Outputs set to **Old Data** or **Don't Care**

Notes to Table 3–1:

(1) FIFO buffers and embedded shift registers that require external logic elements (LEs) for implementing control logic.

(2) Width modes of ×32 and ×36 are not available.

For information about the number of M9K memory blocks for Cyclone IV devices, refer to the *Cyclone IV Device Family Overview* chapter in volume 1 of the *Cyclone IV Device Handbook*.

Control Signals

The clock-enable control signal controls the clock entering the input and output registers and the entire M9K memory block. This signal disables the clock so that the M9K memory block does not see any clock edges and does not perform any operations.

The rden and wren control signals control the read and write operations for each port of M9K memory blocks. You can disable the rden or wren signals independently to save power whenever the operation is not required.

Parity Bit Support

Parity checking for error detection is possible with the parity bit along with internal logic resources. Cyclone IV devices M9K memory blocks support a parity bit for each storage byte. You can use this bit as either a parity bit or as an additional data bit. No parity function is actually performed on this bit.

Byte Enable Support

Cyclone IV devices M9K memory blocks support byte enables that mask the input data so that only specific bytes of data are written. The unwritten bytes retain the previous written value. The wren signals, along with the byte-enable (byteena) signals, control the write operations of the RAM block. The default value of the byteena signals is high (enabled), in which case writing is controlled only by the wren signals. There is no clear port to the byteena registers. M9K blocks support byte enables when the write port has a data width of ×16, ×18, ×32, or ×36 bits.

Byte enables operate in one-hot manner, with the LSB of the byteena signal corresponding to the least significant byte of the data bus. For example, if byteena = 01 and you are using a RAM block in ×18 mode, data[8..0] is enabled and data[17..9] is disabled. Similarly, if byteena = 11, both data[8..0] and data[17..9] are enabled. Byte enables are active high.

Table 3–2 lists the byte selection.

Table 3–2. byteena for Cyclone IV Devices M9K Blocks [1]

byteena[3..0]	Affected Bytes			
	datain ×16	datain ×18	datain ×32	datain ×36
[0] = 1	[7..0]	[8..0]	[7..0]	[8..0]
[1] = 1	[15..8]	[17..9]	[15..8]	[17..9]
[2] = 1	—	—	[23..16]	[26..18]
[3] = 1	—	—	[31..24]	[35..27]

Note to Table 3–2:

(1) Any combination of byte enables is possible.

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

Figure 3–1. Cyclone IV Devices byteena Functional Waveform [1]
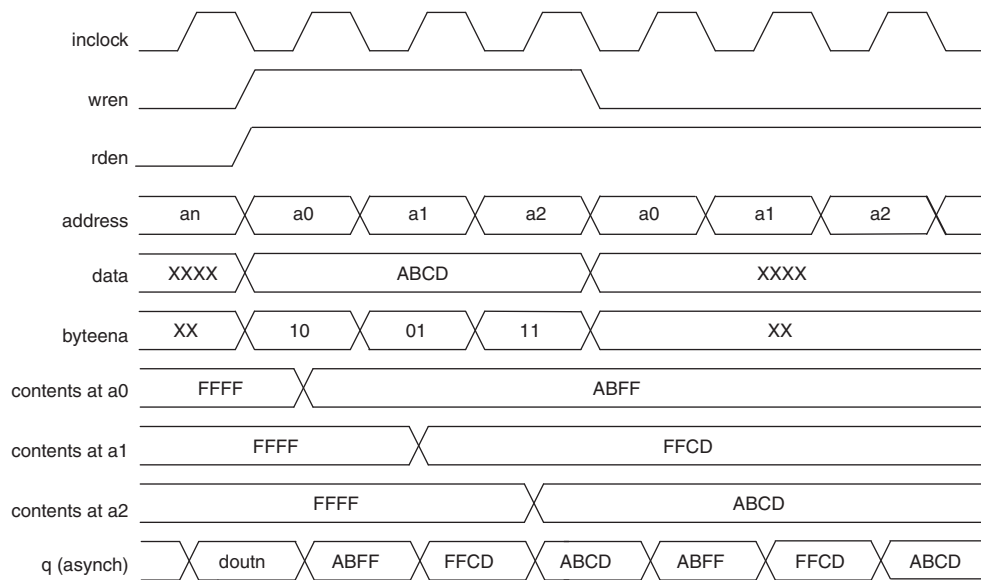
Note to Figure 3–1:

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

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

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

Packed Mode Support

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

■ Each of the two independent block sizes is less than or equal to half of the M9K block size. The maximum data width for each independent block is 18 bits wide.

■ Each of the single-port memory blocks is configured in single-clock mode. For more information about packed mode support, refer to "Single-Port Mode" on page 3–8 and "Single-Clock Mode" on page 3–15.

Address Clock Enable Support

Cyclone IV devices M9K memory blocks support an active-low address clock enable, which holds the previous address value for as long as the `addressstall` signal is high (`addressstall = '1'`). When you configure M9K memory blocks in dual-port mode, each port has its own independent address clock enable.

Figure 3–2 shows an address clock enable block diagram. The address register output feeds back to its input using a multiplexer. The multiplexer output is selected by the address clock enable (`addressstall`) signal.

Figure 3–2. Cyclone IV Devices Address Clock Enable Block Diagram
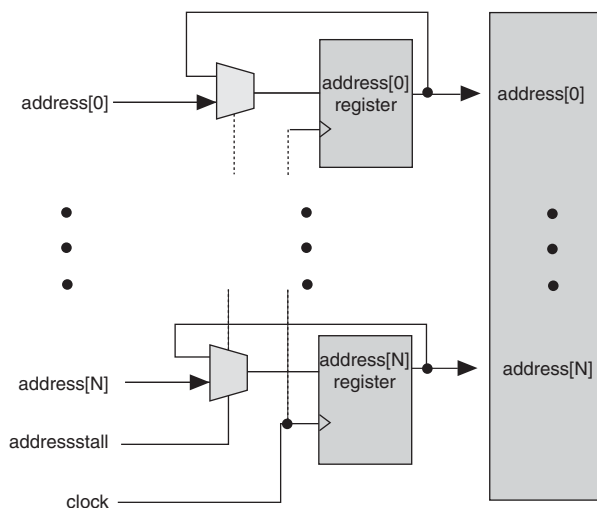
The address clock enable is typically used to improve the effectiveness of cache memory applications during a cache-miss. The default value for the address clock enable signals is low.

Figure 3–3 and Figure 3–4 show the address clock enable waveform during read and write cycles, respectively.

Figure 3–3. Cyclone IV Devices Address Clock Enable During Read Cycle Waveform
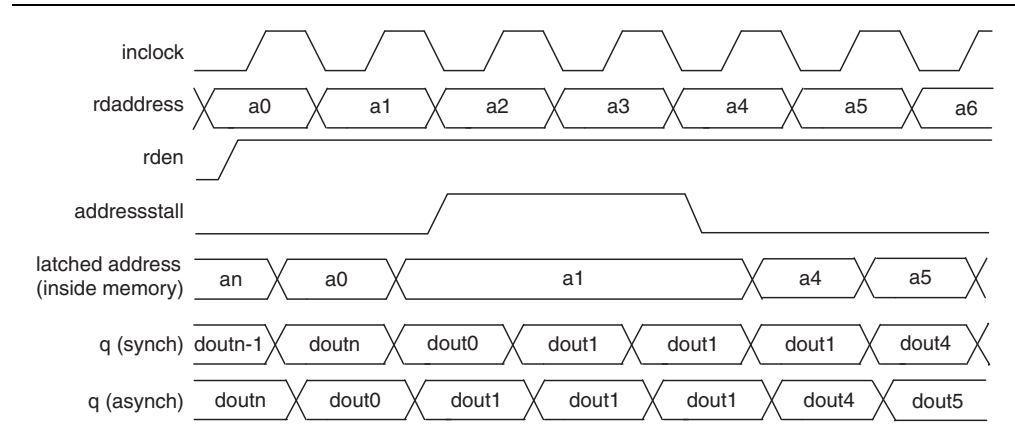
Figure 3–4. Cyclone IV Devices Address Clock Enable During Write Cycle Waveform
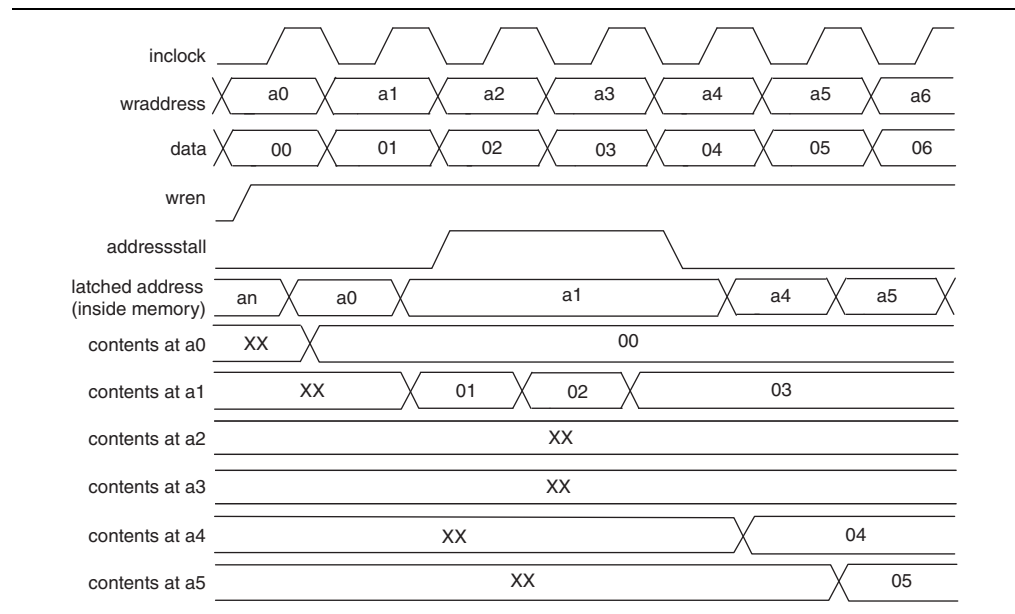
Mixed-Width Support

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

Asynchronous Clear

Cyclone IV devices support asynchronous clears for read address registers, output registers, and output latches only. Input registers other than read address registers are not supported. When applied to output registers, the asynchronous clear signal clears the output registers and the effects are immediately seen. If your RAM does not use output registers, you can still clear the RAM outputs using the output latch asynchronous clear feature.

☞ Asserting asynchronous clear to the read address register during a read operation may corrupt the memory content.

Figure 3–5 shows the functional waveform for the asynchronous clear feature.

Figure 3–5. Output Latch Asynchronous Clear Waveform
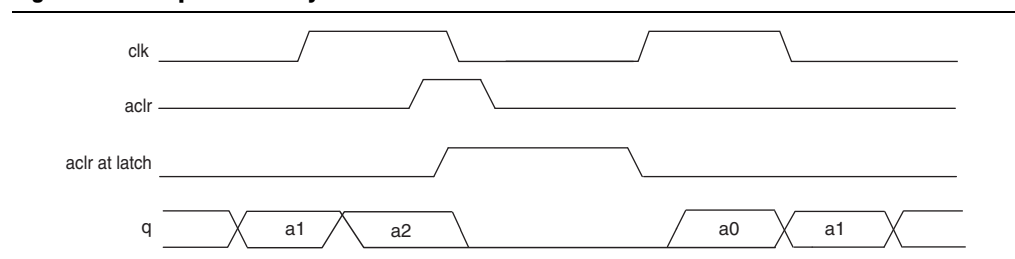
☞ You can selectively enable asynchronous clears per logical memory using the Quartus II RAM MegaWizard™ Plug-In Manager.

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

There are three ways to reset registers in the M9K blocks:

■ Power up the device

■ Use the `aclr` signal for output register only

■ Assert the device-wide reset signal using the **DEV_CLRn** option

Memory Modes

Cyclone IV devices M9K memory blocks allow you to implement fully-synchronous SRAM memory in multiple modes of operation. Cyclone IV devices M9K memory blocks do not support asynchronous (unregistered) memory inputs.

M9K memory blocks support the following modes:

■ Single-port

■ Simple dual-port

■ True dual-port

■ Shift-register

■ ROM

■ FIFO

☞ Violating the setup or hold time on the M9K memory block input registers may corrupt memory contents. This applies to both read and write operations.

Single-Port Mode

Single-port mode supports non-simultaneous read and write operations from a single address. Figure 3–6 shows the single-port memory configuration for Cyclone IV devices M9K memory blocks.

Figure 3–6. Single-Port Memory [1], [2]
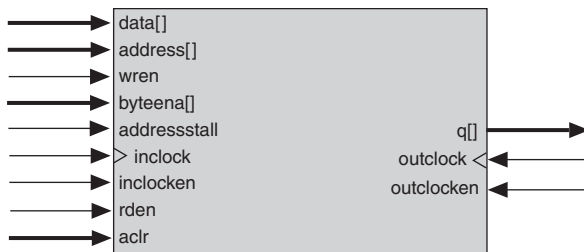
Notes to Figure 3–6:

(1)   You can implement two single-port memory blocks in a single M9K block.

(2)   For more information, refer to "Packed Mode Support" on page 3–4.

During a write operation, the behavior of the RAM outputs is configurable. If you activate rden during a write operation, the RAM outputs show either the new data being written or the old data at that address. If you perform a write operation with rden deactivated, the RAM outputs retain the values they held during the most recent active rden signal.

To choose the desired behavior, set the **Read-During-Write** option to either **New Data** or **Old Data** in the RAM MegaWizard Plug-In Manager in the Quartus II software. For more information about read-during-write mode, refer to "Read-During-Write Operations" on page 3–15.

The port width configurations for M9K blocks in single-port mode are as follow:

■   8192 × 1

■   4096 × 2

■   2048 × 4

■   1024 × 8

■   1024 × 9

■   512 × 16

■   512 × 18

■   256 × 32

■   256 × 36

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

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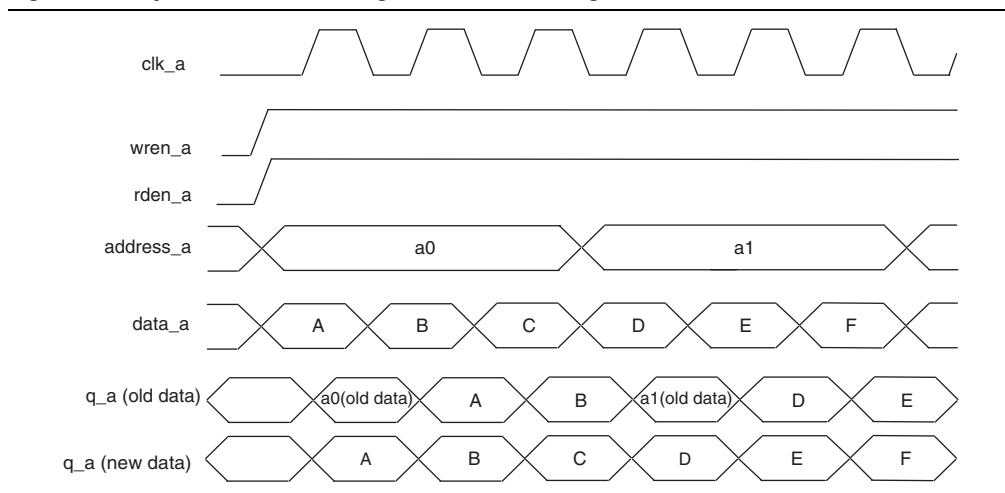
Simple Dual-Port Mode

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

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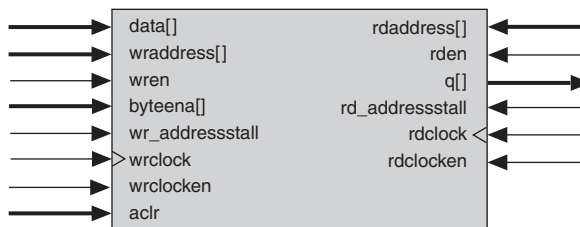
Note to Figure 3–8:

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

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

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

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

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

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

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

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

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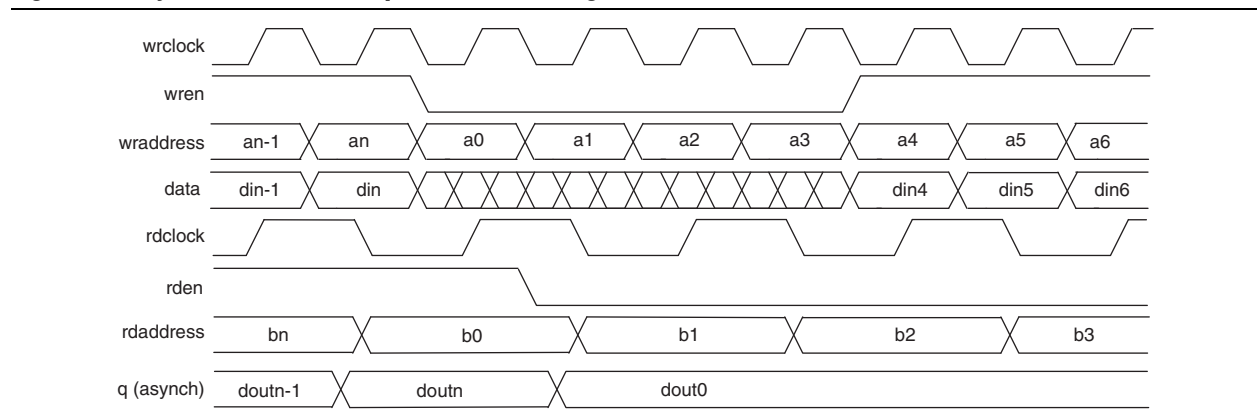
True Dual-Port Mode

True dual-port mode supports any combination of two-port operations: two reads, two writes, or one read and one write, at two different clock frequencies. Figure 3–10 shows Cyclone IV devices true dual-port memory configuration.

Figure 3–10. Cyclone IV Devices True Dual-Port Memory [(1)]
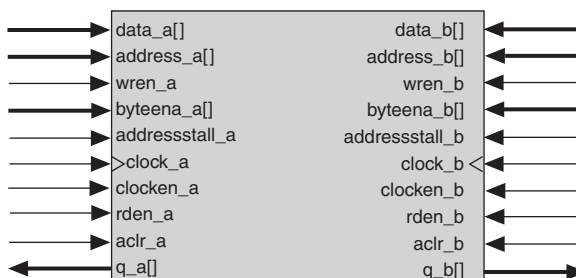
Note to Figure 3–10:

(1) True dual-port memory supports input or output clock mode in addition to the independent clock mode shown.

☞ The widest bit configuration of the M9K blocks in true dual-port mode is 512×16-bit (18-bit with parity).

Table 3–4 lists the possible M9K block mixed-port width configurations.

Table 3–4. Cyclone IV Devices M9K Block Mixed-Width Configurations (True Dual-Port Mode)

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

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

In true dual-port mode, you can access any memory location at any time from either port A or port B. However, when accessing the same memory location from both ports, you must avoid possible write conflicts. When you attempt to write to the same address location from both ports at the same time, a write conflict happens. This results in unknown data being stored to that address location. There is no conflict resolution circuitry built into the Cyclone IV devices M9K memory blocks. You must handle address conflicts external to the RAM block.

Figure 3–11 shows true dual-port timing waveforms for the write operation at port A and read operation at port B. Registering the outputs of the RAM simply delays the q outputs by one clock cycle.

Figure 3–11. Cyclone IV Devices True Dual-Port Timing Waveform
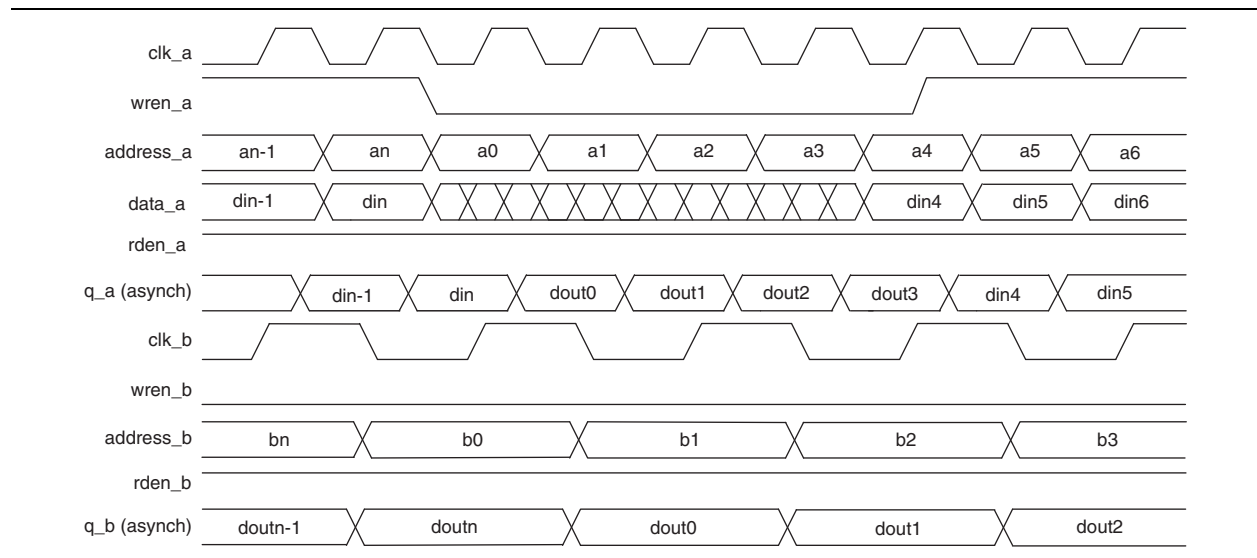
Shift Register Mode

Cyclone IV devices M9K memory blocks can implement shift registers for digital signal processing (DSP) applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto-correlation and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flipflops that quickly exhaust many logic cells for large shift registers. A more efficient alternative is to use embedded memory as a shift register block, which saves logic cell and routing resources.

The size of a ($w \times m \times n$) shift register is determined by the input data width (w), the length of the taps (m), and the number of taps (n), and must be less than or equal to the maximum number of memory bits, which is 9,216 bits. In addition, the size of ($w \times n$) must be less than or equal to the maximum width of the block, which is 36 bits. If you need a larger shift register, you can cascade the M9K memory blocks.

Figure 3–12 shows the Cyclone IV devices M9K memory block in shift register mode.

Figure 3–12. Cyclone IV Devices Shift Register Mode Configuration
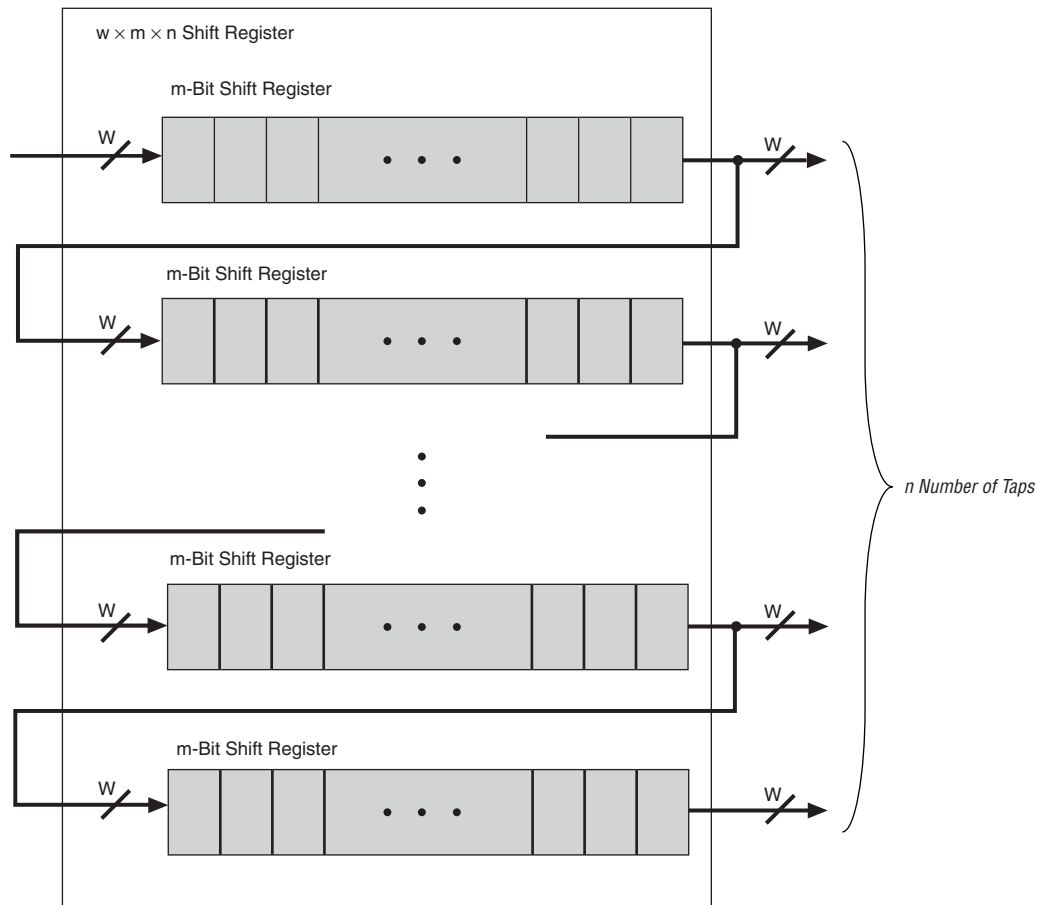
ROM Mode

Cyclone IV devices M9K memory blocks support ROM mode. A **.mif** initializes the ROM contents of these blocks. The address lines of the ROM are registered. The outputs can be registered or unregistered. The ROM read operation is identical to the read operation in the single-port RAM configuration.

FIFO Buffer Mode

Cyclone IV devices M9K memory blocks support single-clock or dual-clock FIFO buffers. Dual clock FIFO buffers are useful when transferring data from one clock domain to another clock domain. Cyclone IV devices M9K memory blocks do not support simultaneous read and write from an empty FIFO buffer.

For more information about FIFO buffers, refer to the *Single- and Dual-Clock FIFO Megafunction User Guide*.

Clocking Modes

Cyclone IV devices M9K memory blocks support the following clocking modes:

- Independent

- Input or output

- Read or write

- Single-clock

When using read or write clock mode, if you perform a simultaneous read or write to the same address location, the output read data is unknown. If you require the output data to be a known value, use either single-clock mode or I/O clock mode and choose the appropriate read-during-write behavior in the MegaWizard Plug-In Manager.

☞ Violating the setup or hold time on the memory block input registers might corrupt the memory contents. This applies to both read and write operations.

☞ Asynchronous clears are available on read address registers, output registers, and output latches only.

Table 3–5 lists the clocking mode versus memory mode support matrix.

Table 3–5. Cyclone IV Devices Memory Clock Modes

Clocking Mode	True Dual-Port Mode	Simple Dual-Port Mode	Single-Port Mode	ROM Mode	FIFO Mode
Independent	✓	—	—	✓	—
Input or output	✓	✓	✓	✓	—
Read or write	—	✓	—	—	✓
Single-clock	✓	✓	✓	✓	✓

Independent Clock Mode

Cyclone IV devices M9K memory blocks can implement independent clock mode for true dual-port memories. In this mode, a separate clock is available for each port (port A and port B). `clock A` controls all registers on the port A side, while `clock B` controls all registers on the port B side. Each port also supports independent clock enables for port A and B registers.

Input or Output Clock Mode

Cyclone IV devices M9K memory blocks can implement input or output clock mode for FIFO, single-port, true, and simple dual-port memories. In this mode, an input clock controls all input registers to the memory block including data, address, `byteena`, `wren`, and `rden` registers. An output clock controls the data-output registers. Each memory block port also supports independent clock enables for input and output registers.

Read or Write Clock Mode

Cyclone IV devices M9K memory blocks can implement read or write clock mode for FIFO and simple dual-port memories. In this mode, a write clock controls the data inputs, write address, and `wren` registers. Similarly, a read clock controls the data outputs, read address, and `rden` registers. M9K memory blocks support independent clock enables for both the read and write clocks.

When using read or write mode, if you perform a simultaneous read or write to the same address location, the output read data is unknown. If you require the output data to be a known value, use either single-clock mode, input clock mode, or output clock mode and choose the appropriate read-during-write behavior in the MegaWizard Plug-In Manager.

Single-Clock Mode

Cyclone IV devices M9K memory blocks can implement single-clock mode for FIFO, ROM, true dual-port, simple dual-port, and single-port memories. In this mode, you can control all registers of the M9K memory block with a single clock together with clock enable.

Design Considerations

This section describes designing with M9K memory blocks.

Read-During-Write Operations

"Same-Port Read-During-Write Mode" on page 3–16 and "Mixed-Port Read-During-Write Mode" on page 3–16 describe the functionality of the various RAM configurations when reading from an address during a write operation at that same address.

There are two read-during-write data flows: same-port and mixed-port. Figure 3–13 shows the difference between these flows.

Figure 3–13. Cyclone IV Devices Read-During-Write Data Flow
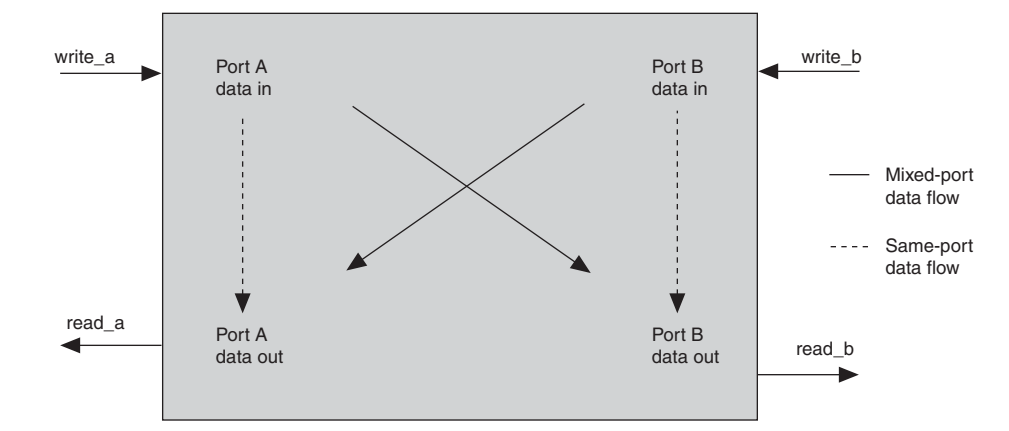
Same-Port Read-During-Write Mode

This mode applies to a single-port RAM or the same port of a true dual-port RAM. In the same port read-during-write mode, there are two output choices: **New Data** mode (or flow-through) and **Old Data** mode. In **New Data** mode, new data is available on the rising edge of the same clock cycle on which it was written. In **Old Data** mode, the RAM outputs reflect the old data at that address before the write operation proceeds.

When using **New Data** mode together with byteena, you can control the output of the RAM. When byteena is high, the data written into the memory passes to the output (flow-through). When byteena is low, the masked-off data is not written into the memory and the old data in the memory appears on the outputs. Therefore, the output can be a combination of new and old data determined by byteena.

Figure 3–14 and Figure 3–15 show sample functional waveforms of same port read-during-write behavior with both **New Data** and **Old Data** modes, respectively.

Figure 3–14. Same Port Read-During Write: New Data Mode
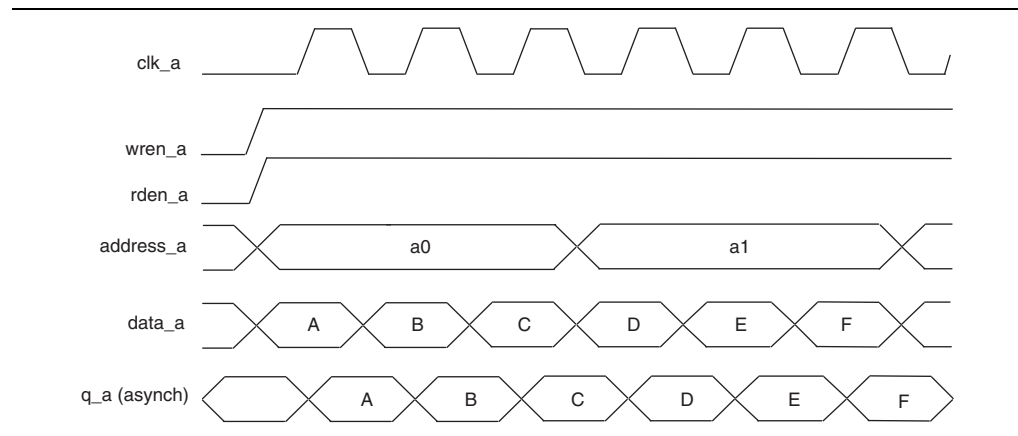
Figure 3–15. Same Port Read-During-Write: Old Data Mode
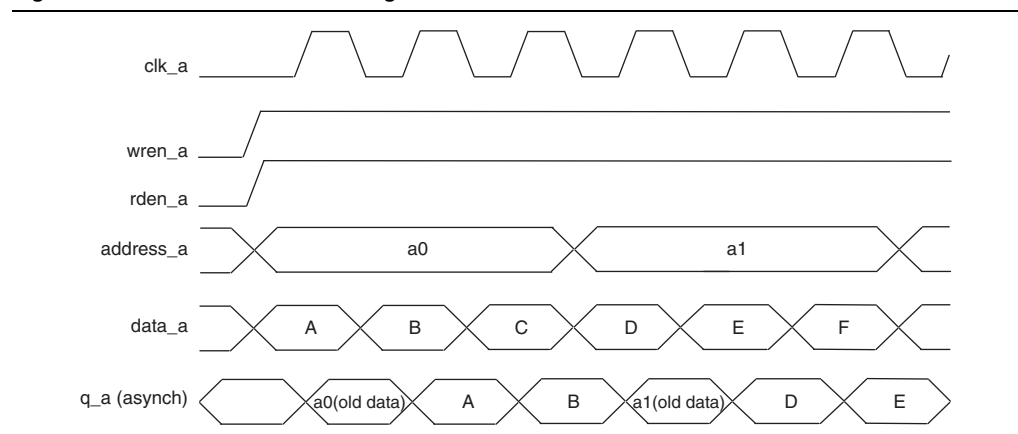
Mixed-Port Read-During-Write Mode

This mode applies to a RAM in simple or true dual-port mode, which has one port reading and the other port writing to the same address location with the same clock.

In this mode, you also have two output choices: **Old Data** mode or **Don't Care** mode. In **Old Data** mode, a read-during-write operation to different ports causes the RAM outputs to reflect the old data at that address location. In **Don't Care** mode, the same operation results in a "Don't Care" or unknown value on the RAM outputs.

For more information about how to implement the desired behavior, refer to the *RAM Megafunction User Guide*.

Figure 3–16 shows a sample functional waveform of mixed port read-during-write behavior for **Old Data** mode. In **Don't Care** mode, the old data is replaced with "Don't Care".

Figure 3–16. Mixed Port Read-During-Write: Old Data Mode
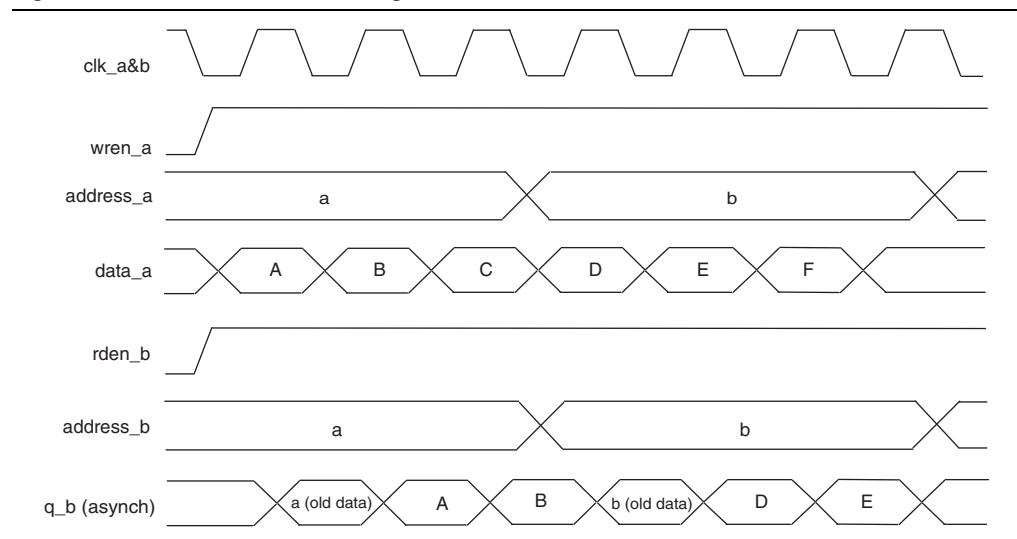
For mixed-port read-during-write operation with dual clocks, the relationship between the clocks determines the output behavior of the memory. If you use the same clock for the two clocks, the output is the old data from the address location. However, if you use different clocks, the output is unknown during the mixed-port read-during-write operation. This unknown value may be the old or new data at the address location, depending on whether the read happens before or after the write.

Conflict Resolution

When you are using M9K memory blocks in true dual-port mode, it is possible to attempt two write operations to the same memory location (address). Because there is no conflict resolution circuitry built into M9K memory blocks, this results in unknown data being written to that location. Therefore, you must implement conflict-resolution logic external to the M9K memory block.

Power-Up Conditions and Memory Initialization

The M9K memory block outputs of Cyclone IV devices power up to zero (cleared) regardless of whether the output registers are used or bypassed. All M9K memory blocks support initialization using a **.mif**. You can create **.mif**s in the Quartus II software and specify their use using the RAM MegaWizard Plug-In Manager when instantiating memory in your design. Even if memory is pre-initialized (for example, using a **.mif**), it still powers up with its outputs cleared. Only the subsequent read after power up outputs the pre-initialized values.

For more information about **.mif**s, refer to the *RAM Megafunction User Guide* and the *Quartus II Handbook*.

Power Management

The M9K memory block clock enables of Cyclone IV devices allow you to control clocking of each M9K memory block to reduce AC power consumption. Use the rden signal to ensure that read operations only occur when necessary. If your design does not require read-during-write, reduce power consumption by deasserting the rden signal during write operations or any period when there are no memory operations. The Quartus II software automatically powers down any unused M9K memory blocks to save static power.

Document Revision History

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

Table 3–6. Document Revision History

Date	Version	Changes
November 2011	1.1	Updated the "Byte Enable Support" section.
November 2009	1.0	Initial release.

Cyclone® IV devices include a combination of on-chip resources and external interfaces that help increase performance, reduce system cost, and lower the power consumption of digital signal processing (DSP) systems. Cyclone IV devices, either alone or as DSP device co-processors, are used to improve price-to-performance ratios of DSP systems. Particular focus is placed on optimizing Cyclone IV devices for applications that benefit from an abundance of parallel processing resources, which include video and image processing, intermediate frequency (IF) modems used in wireless communications systems, and multi-channel communications and video systems.

This chapter contains the following sections:

- "Embedded Multiplier Block Overview" on page 4–1
- "Architecture" on page 4–2
- "Operational Modes" on page 4–4

Embedded Multiplier Block Overview

Figure 4–1 shows one of the embedded multiplier columns with the surrounding logic array blocks (LABs). The embedded multiplier is configured as either one 18 × 18 multiplier or two 9 × 9 multipliers. For multiplications greater than 18 × 18, the Quartus® II software cascades multiple embedded multiplier blocks together. There are no restrictions on the data width of the multiplier, but the greater the data width, the slower the multiplication process.

Figure 4–1. Embedded Multipliers Arranged in Columns with Adjacent LABs

Feedback   Subscribe

Table 4–1 lists the number of embedded multipliers and the multiplier modes that can be implemented in each Cyclone IV device.

Table 4–1. Number of Embedded Multipliers in Cyclone IV Devices

Device Family	Device	Embedded Multipliers	9 × 9 Multipliers [1]	18 × 18 Multipliers [1]
Cyclone IV GX	EP4CGX15	0	0	0
	EP4CGX22	40	80	40
	EP4CGX30	80	160	80
	EP4CGX50	140	280	140
	EP4CGX75	198	396	198
	EP4CGX110	280	560	280
	EP4CGX150	360	720	360
Cyclone IV E	EP4CE6	15	30	15
	EP4CE10	23	46	23
	EP4CE15	56	112	56
	EP4CE22	66	132	66
	EP4CE30	66	132	66
	EP4CE40	116	232	116
	EP4CE55	154	308	154
	EP4CE75	200	400	200
	EP4CE115	266	532	266

Note to Table 4–1:

(1) These columns show the number of 9 × 9 or 18 × 18 multipliers for each device.

In addition to the embedded multipliers in Cyclone IV devices, you can implement soft multipliers by using the M9K memory blocks as look-up tables (LUTs). The LUTs contain partial results from the multiplication of input data with coefficients that implement variable depth and width high-performance soft multipliers for low-cost, high-volume DSP applications. The availability of soft multipliers increases the number of available multipliers in the device.

For more information about M9K memory blocks, refer to the *Memory Blocks in Cyclone IV Devices* chapter.

For more information about soft multipliers, refer to *AN 306: Implementing Multipliers in FPGA Devices*.

Architecture

Each embedded multiplier consists of the following elements:

- Multiplier stage
- Input and output registers
- Input and output interfaces

Figure 4–2 shows the multiplier block architecture.

Figure 4–2. Multiplier Block Architecture
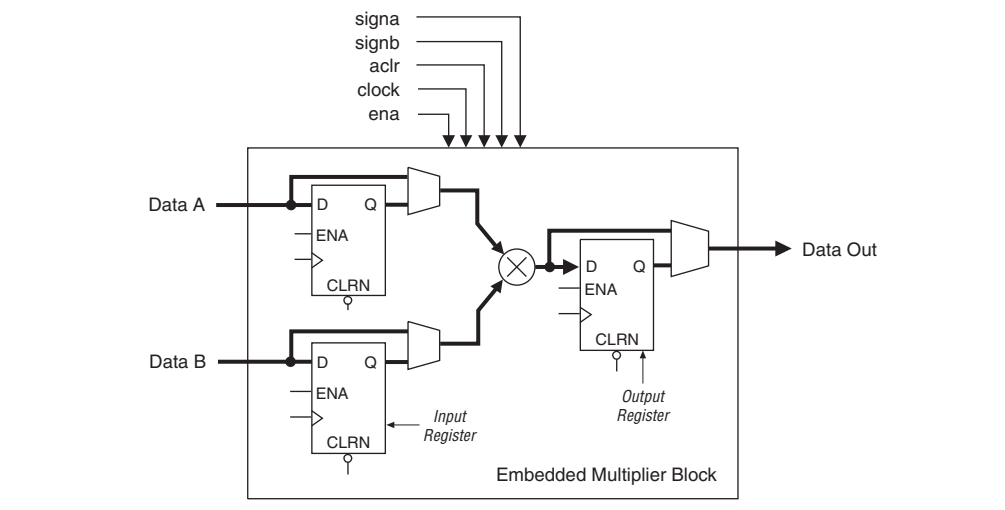
Input Registers

You can send each multiplier input signal into an input register or directly into the multiplier in 9- or 18-bit sections, depending on the operational mode of the multiplier. You can send each multiplier input signal through a register independently of other input signals. For example, you can send the multiplier Data A signal through a register and send the Data B signal directly to the multiplier.

The following control signals are available for each input register in the embedded multiplier:

- clock

- clock enable

- asynchronous clear

All input and output registers in a single embedded multiplier are fed by the same clock, clock enable, and asynchronous clear signals.

Multiplier Stage

The multiplier stage of an embedded multiplier block supports 9 × 9 or 18 × 18 multipliers, as well as other multipliers between these configurations. Depending on the data width or operational mode of the multiplier, a single embedded multiplier can perform one or two multiplications in parallel. For multiplier information, refer to "Operational Modes" on page 4–4.

Each multiplier operand is a unique signed or unsigned number. The signa and signb signals control an input of a multiplier and determine if the value is signed or unsigned. If the signa signal is high, the Data A operand is a signed number. If the signa signal is low, the Data A operand is an unsigned number.

Table 4–2 lists the sign of the multiplication results for the various operand sign representations. The results of the multiplication are signed if any one of the operands is a signed value.

Table 4–2. Multiplier Sign Representation

Data A		Data B		Result
signa Value	**Logic Level**	**signb Value**	**Logic Level**	
Unsigned	Low	Unsigned	Low	Unsigned
Unsigned	Low	Signed	High	Signed
Signed	High	Unsigned	Low	Signed
Signed	High	Signed	High	Signed

Each embedded multiplier block has only one `signa` and one `signb` signal to control the sign representation of the input data to the block. If the embedded multiplier block has two 9 × 9 multipliers, the `Data A` input of both multipliers share the same `signa` signal, and the `Data B` input of both multipliers share the same `signb` signal. You can dynamically change the `signa` and `signb` signals to modify the sign representation of the input operands at run time. You can send the `signa` and `signb` signals through a dedicated input register. The multiplier offers full precision, regardless of the sign representation.

☞ When the `signa` and `signb` signals are unused, the Quartus II software sets the multiplier to perform unsigned multiplication by default.

Output Registers

You can register the embedded multiplier output with output registers in either 18- or 36-bit sections, depending on the operational mode of the multiplier. The following control signals are available for each output register in the embedded multiplier:

■ clock

■ clock enable

■ asynchronous clear

All input and output registers in a single embedded multiplier are fed by the same clock, clock enable, and asynchronous clear signals.

Operational Modes

You can use an embedded multiplier block in one of two operational modes, depending on the application needs:

■ One 18 × 18 multiplier

■ Up to two 9 × 9 independent multipliers

☞ You can also use embedded multipliers of Cyclone IV devices to implement multiplier adder and multiplier accumulator functions, in which the multiplier portion of the function is implemented with embedded multipliers, and the adder or accumulator function is implemented in logic elements (LEs).

18-Bit Multipliers

You can configure each embedded multiplier to support a single 18 × 18 multiplier for input widths of 10 to 18 bits.

Figure 4–3 shows the embedded multiplier configured to support an 18-bit multiplier.

Figure 4–3. 18-Bit Multiplier Mode
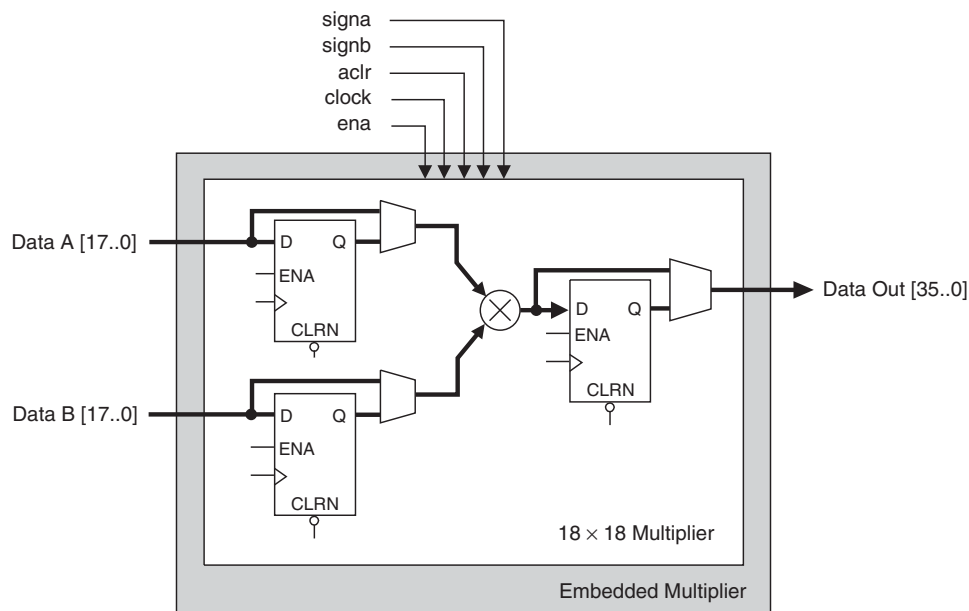
All 18-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Also, you can dynamically change the `signa` and `signb` signals and send these signals through dedicated input registers.

9-Bit Multipliers

You can configure each embedded multiplier to support two 9 × 9 independent multipliers for input widths of up to 9 bits.

Figure 4–4 shows the embedded multiplier configured to support two 9-bit multipliers.

Figure 4–4. 9-Bit Multiplier Mode
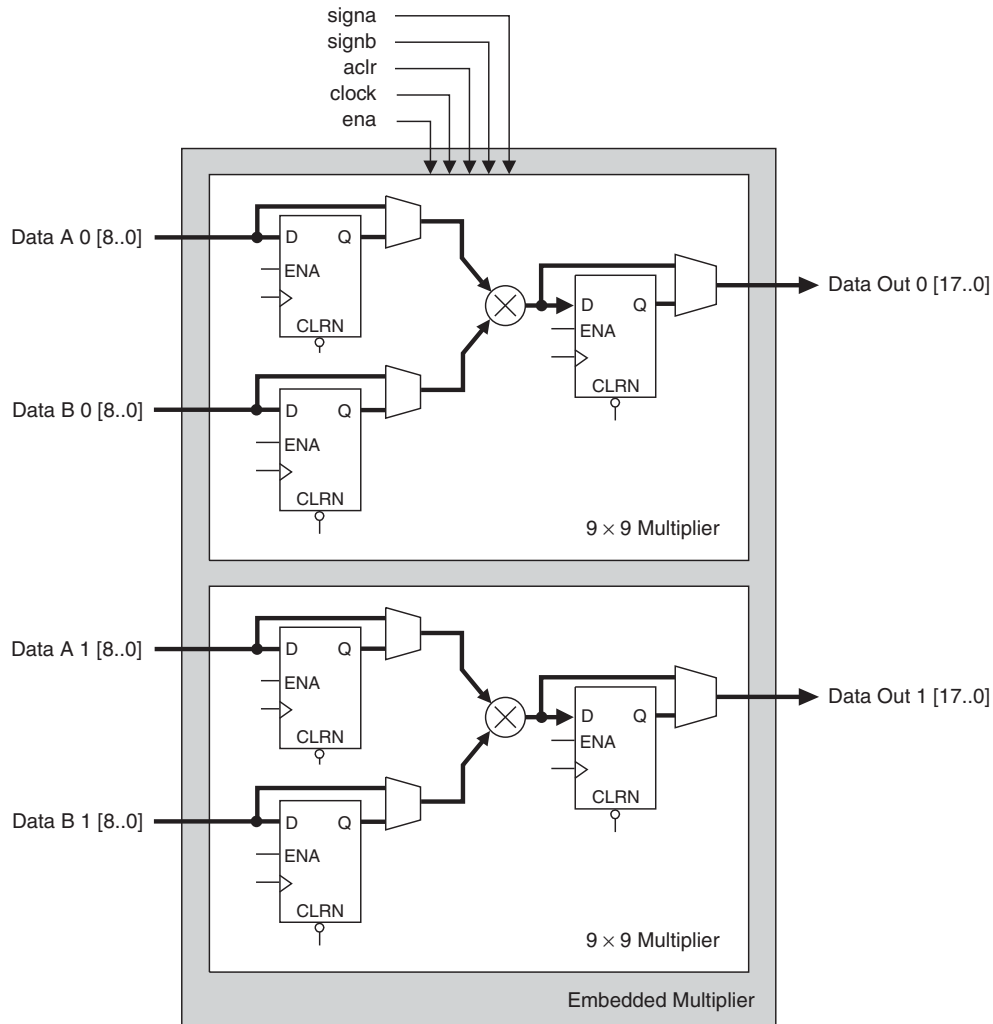
All 9-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Two 9 × 9 multipliers in the same embedded multiplier block share the same signa and signb signal. Therefore, all the Data A inputs feeding the same embedded multiplier must have the same sign representation. Similarly, all the Data B inputs feeding the same embedded multiplier must have the same sign representation.

Document Revision History

Table 4–3 lists the revision history for this chapter.

Table 4–3. Document Revision History

Date	Version	Changes
February 2010	1.1	Added Cyclone IV E devices in Table 4–1 for the Quartus II software version 9.1 SP1 release.
November 2009	1.0	Initial release.

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

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.

ISO
9001:2008
Registered

Feedback   Subscribe

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	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓	—

Table 5–1. GCLK Network Connections for EP4CGX15, EP4CGX22, and EP4CGX30 [1], [2] **(Part 2 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
PLL_3_C1	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	✓	—	—	✓
PLL_3_C2	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	—	—
PLL_3_C3	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	✓	—	✓	✓	—
PLL_3_C4	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	✓	—	✓
PLL_4_C0 [3]	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	✓	—	—	—	—	—	—
PLL_4_C1 [3]	—	—	—	—	—	—	✓	—	—	✓	—	✓	—	—	✓	—	—	—	—	—
PLL_4_C2 [3]	—	—	—	—	—	✓	—	✓	—	—	✓	—	✓	—	—	—	—	—	—	—
PLL_4_C3 [3]	—	—	—	—	—	—	✓	—	✓	—	—	✓	—	✓	—	—	—	—	—	—
PLL_4_C4 [3]	—	—	—	—	—	—	✓	—	✓	—	—	—	✓	—	✓	—	—	—	—	—
DPCLK2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—
DPCLK3 [4]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—
DPCLK4 [4]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
DPCLK5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
DPCLK6 [4]	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK7	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK8	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
DPCLK9 [4]	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—
DPCLK11 [4]	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
DPCLK12 [4]	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
DPCLK13	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—

Notes to Table 5–1:

(1) EP4CGX30 information in this table refers to all EP4CGX30 packages except F484 package.

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

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

(4) This pin applies to EP4CGX22 and EP4CGX30 devices.

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	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—	—	—	✓	—	✓	✓	—	—	—	—	—	—

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

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

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 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 5–3. 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 5–3. 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.

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.
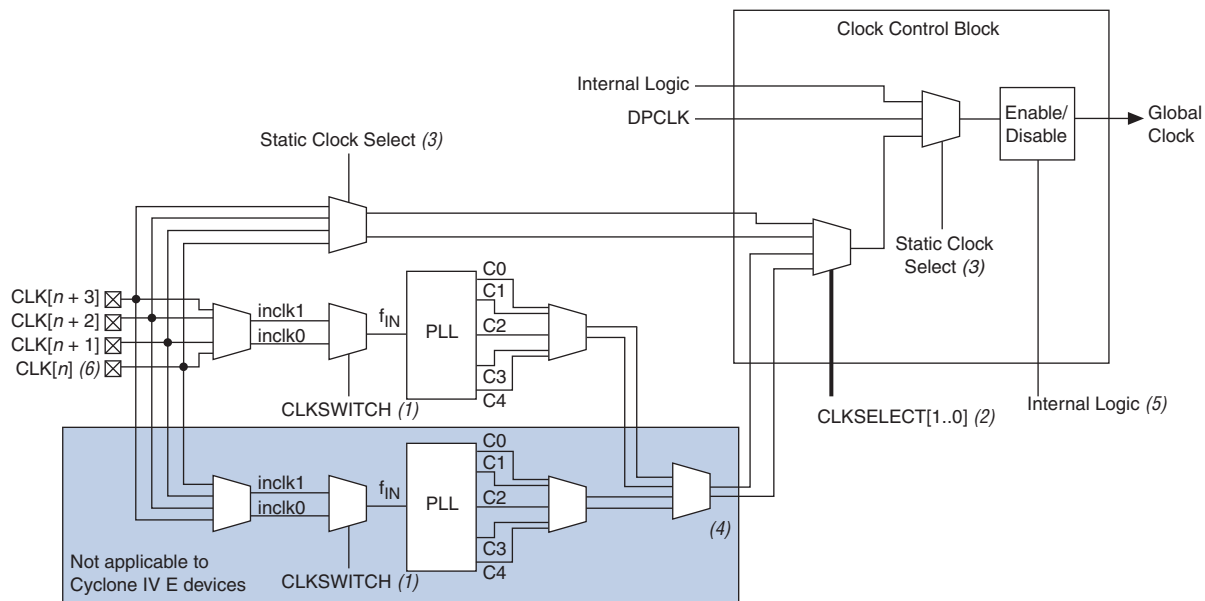
The control block has two functions:

■ Dynamic GCLK clock source selection (not applicable for DPCLK, CDPCLK, and internal logic input)

■ GCLK network power down (dynamic enable and disable)

Figure 5–1 shows the clock control block.

Figure 5–1. Clock Control Block

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*.

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.
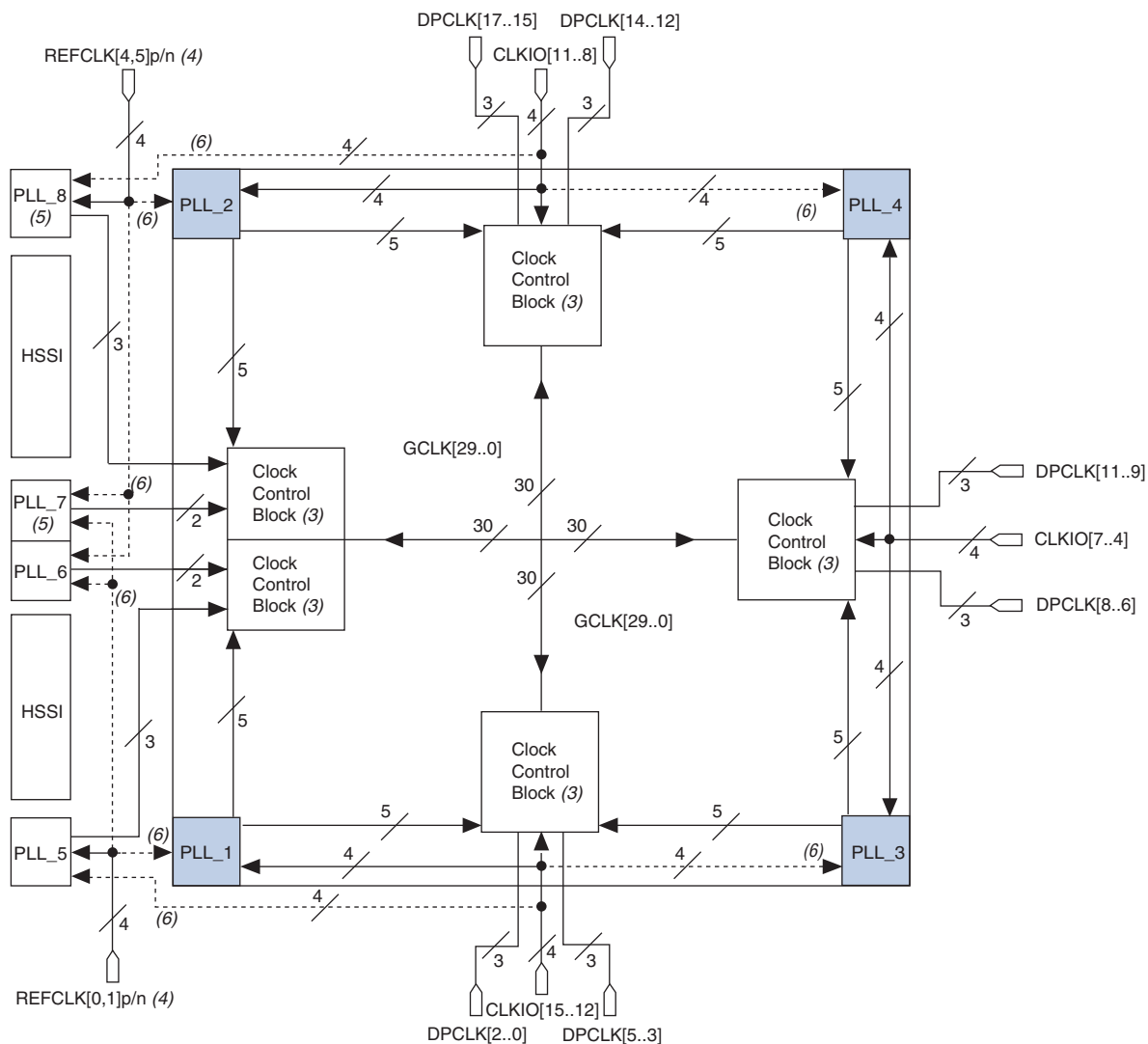
Figure 5–2. Clock Networks and Clock Control Block Locations in EP4CGX15, EP4CGX22, and EP4CGX30 Devices [1], [2]
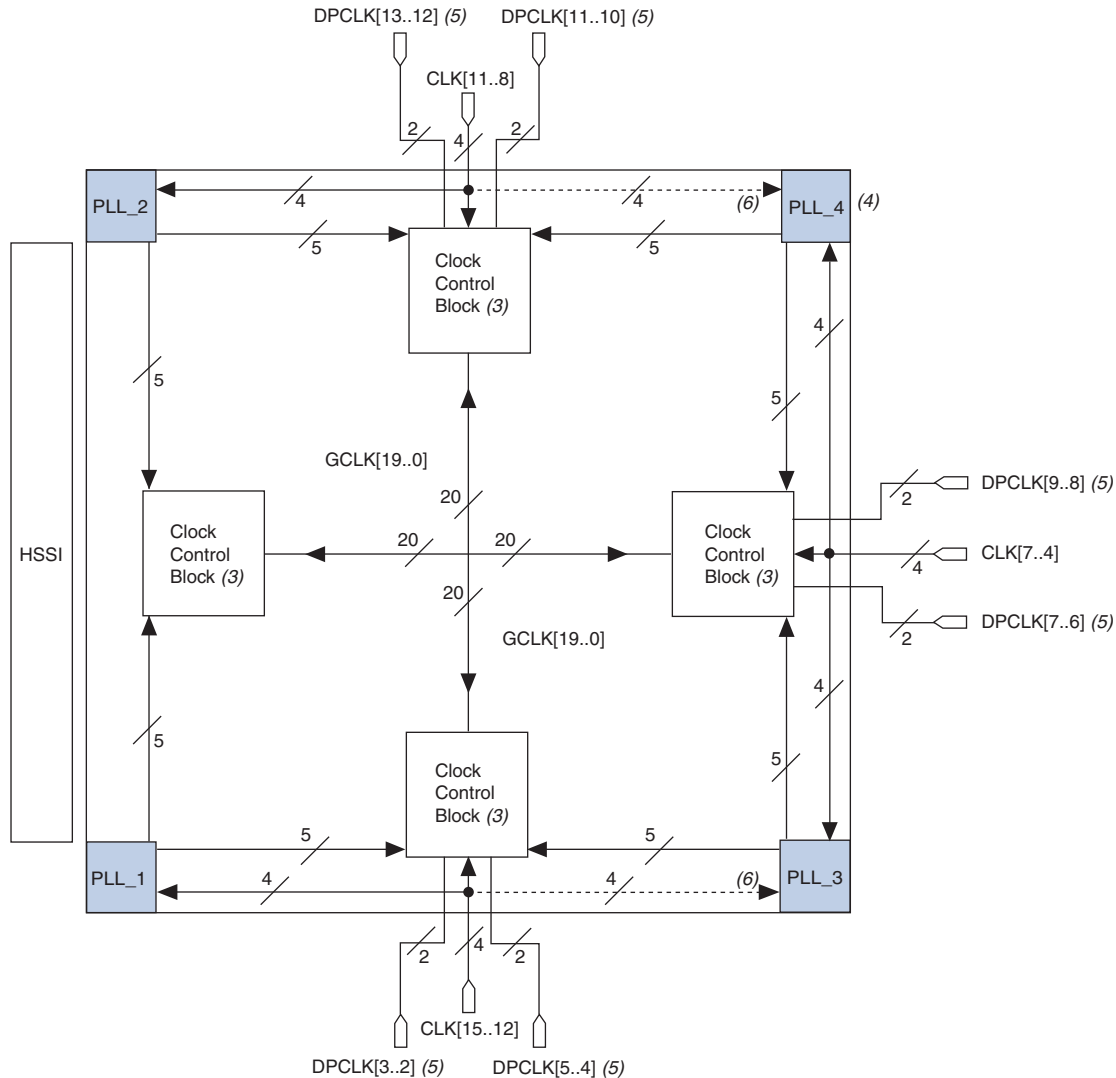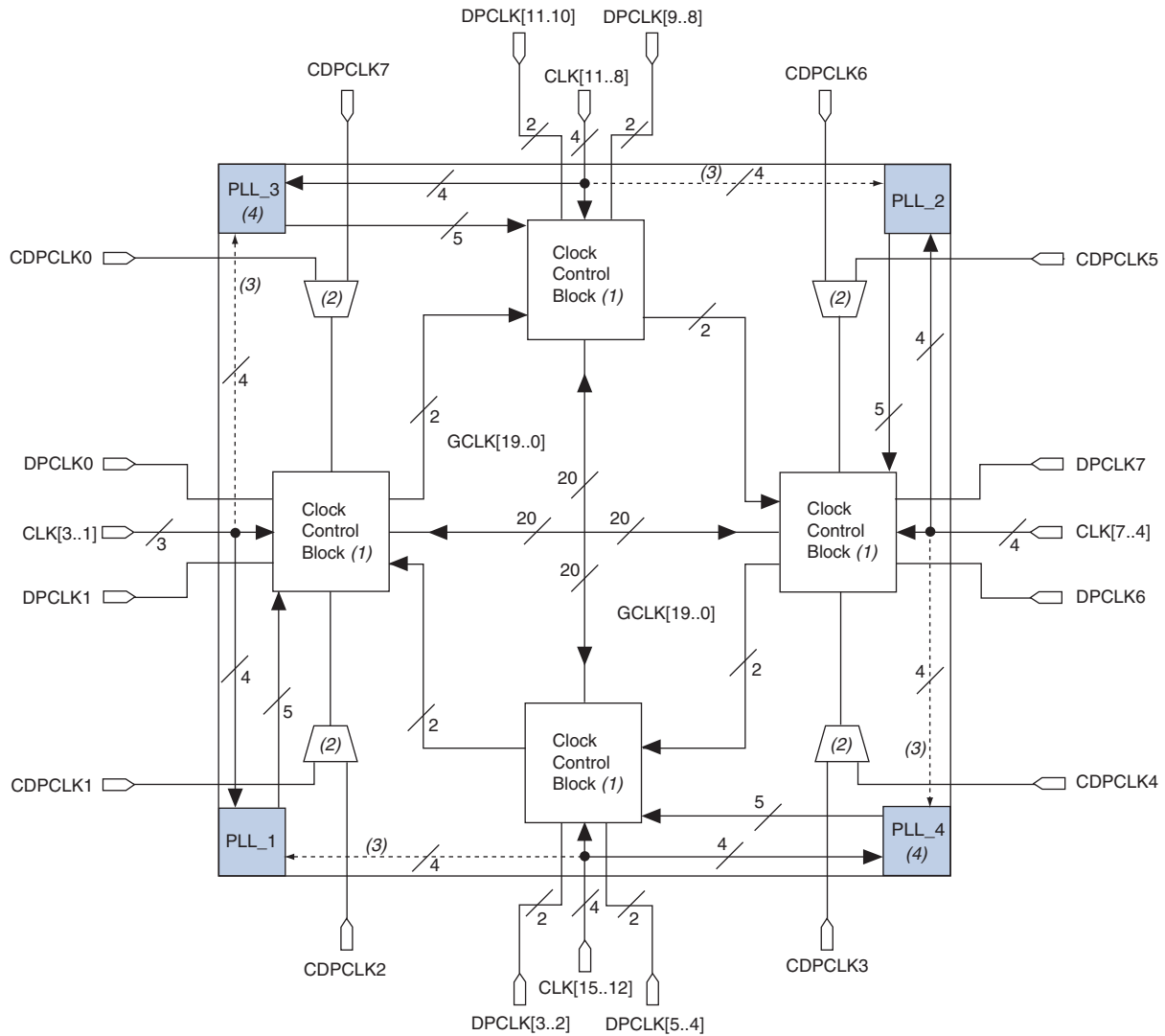

Notes to Figure 5–2:

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

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.

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

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

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
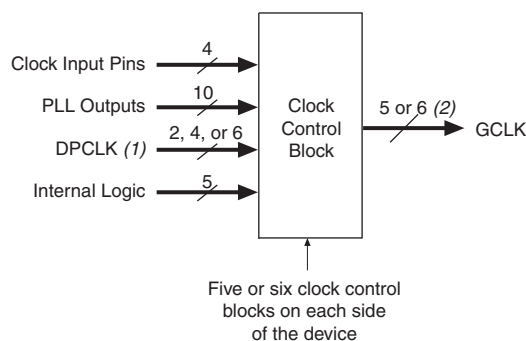
Notes to Figure 5–5:

(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 5–6 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)]
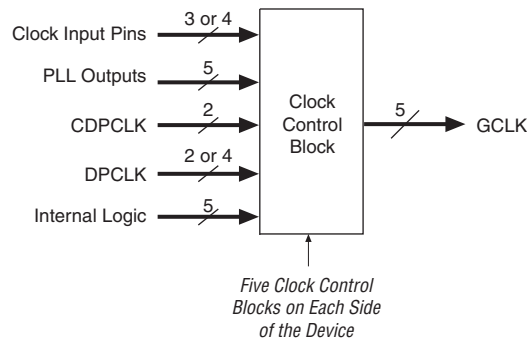
Note to Figure 5–6:

(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.

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

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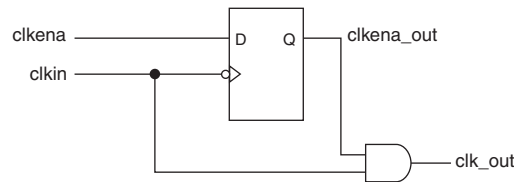
☞ The `clkena` circuitry controlling the output C0 of the PLL to an output pin is implemented with two registers instead of a single register, as shown in Figure 5–7.

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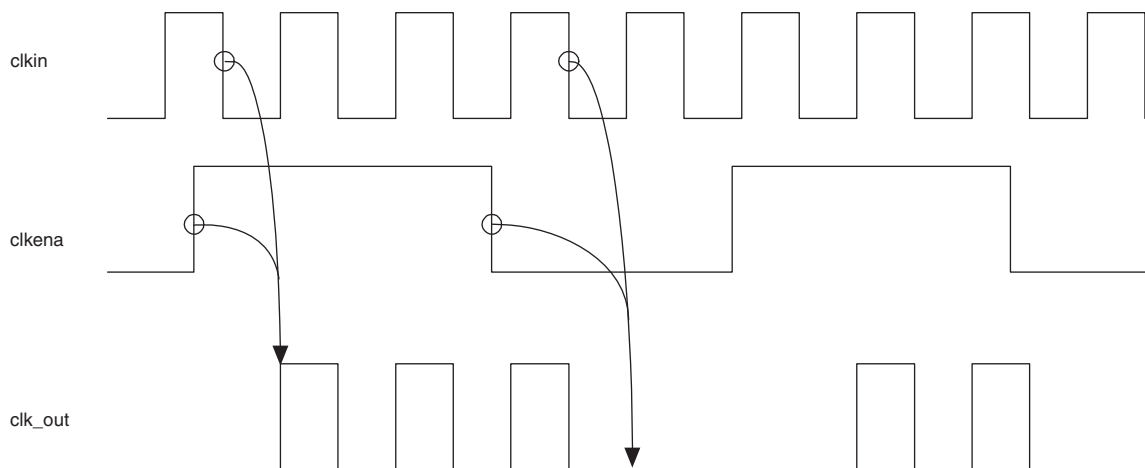
The `clkena` signal can also disable clock outputs if the system is not tolerant to frequency overshoot during PLL resynchronization.

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.

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	✓									

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	✓

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)]
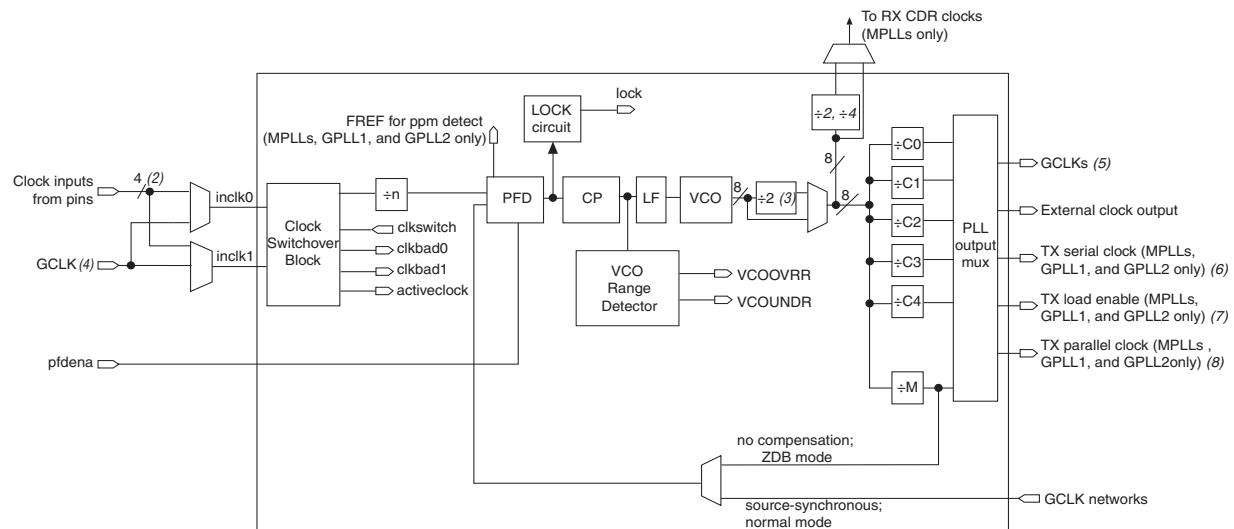
Notes to Figure 5–9:

(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.

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]
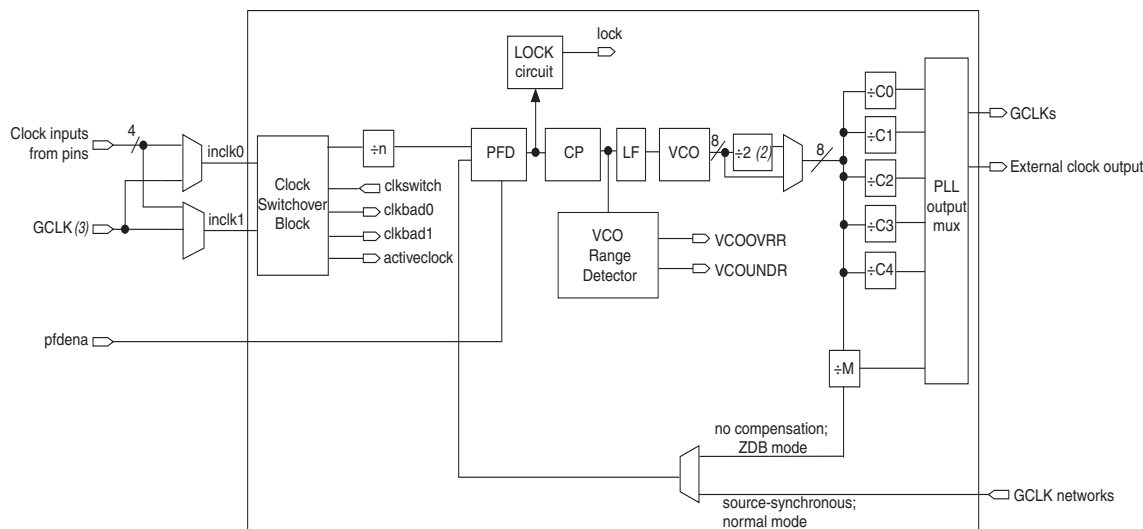
Notes to Figure 5–10:

(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.

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

Figure 5–11. External Clock Outputs for PLLs
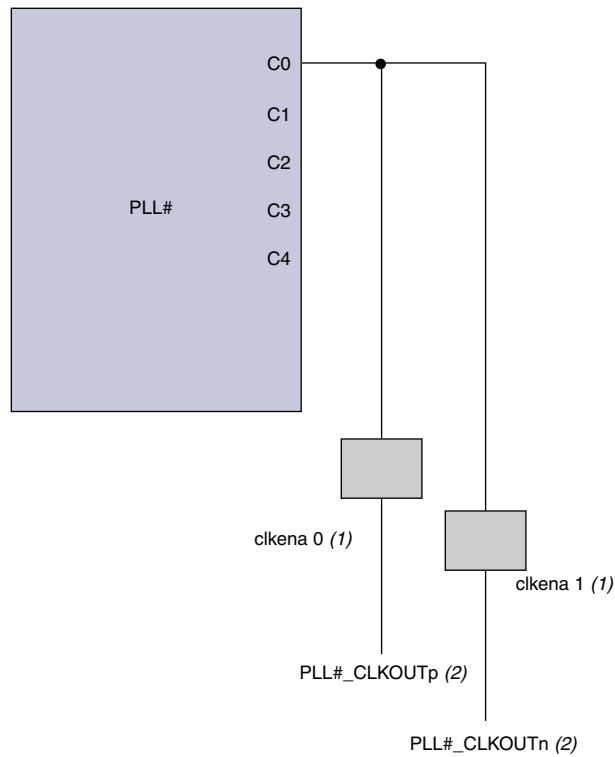
Notes to Figure 5–11:

(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.

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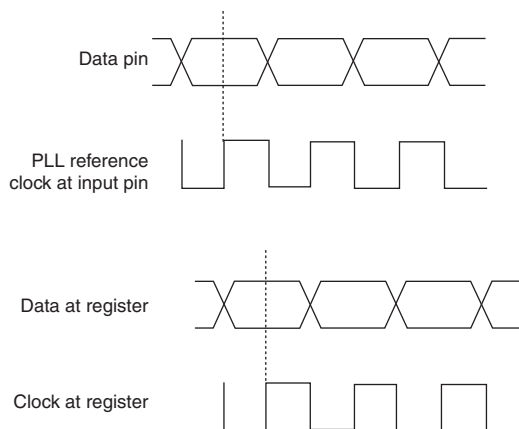
Source-synchronous mode compensates for delay of the clock network used, including any difference in the delay between the following two paths:

■ 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.

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**
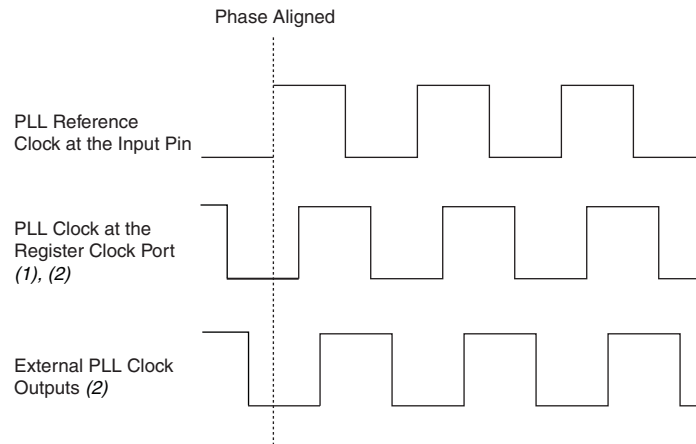
Notes to Figure 5–13:

(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.

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
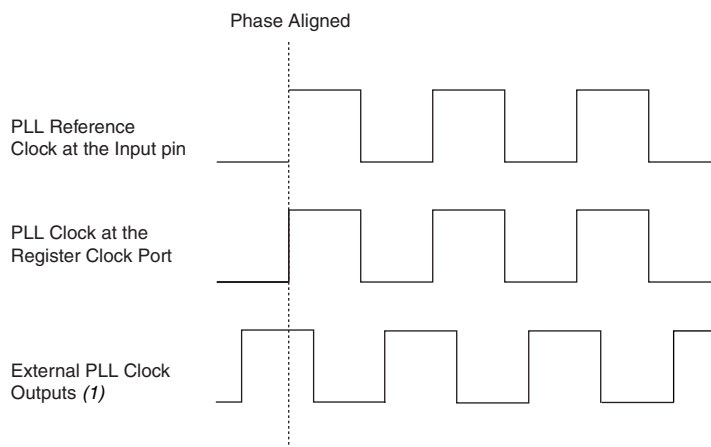

Note to Figure 5–14:

(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
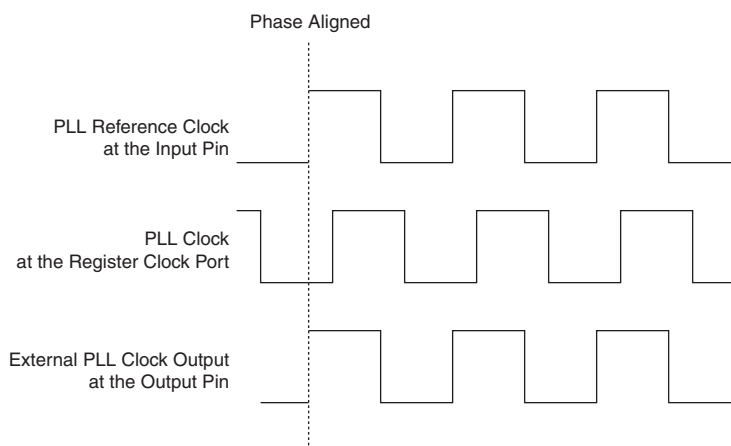
Deterministic Latency Compensation Mode

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.

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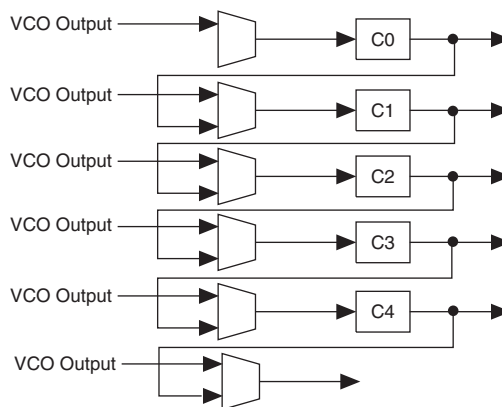
When cascading counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings.

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*.

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**
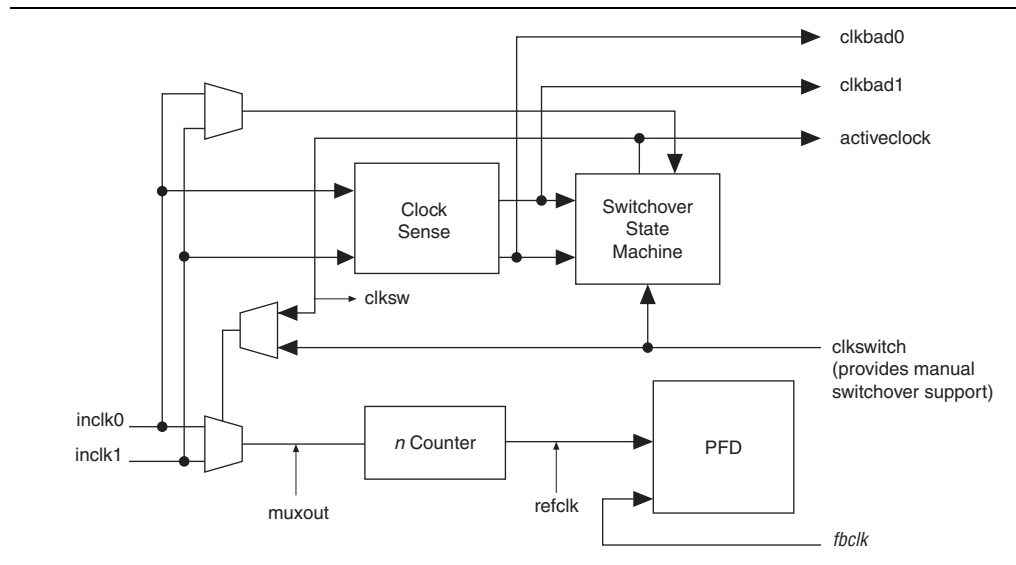
There are two ways to use the clock switchover feature:

■ 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

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]
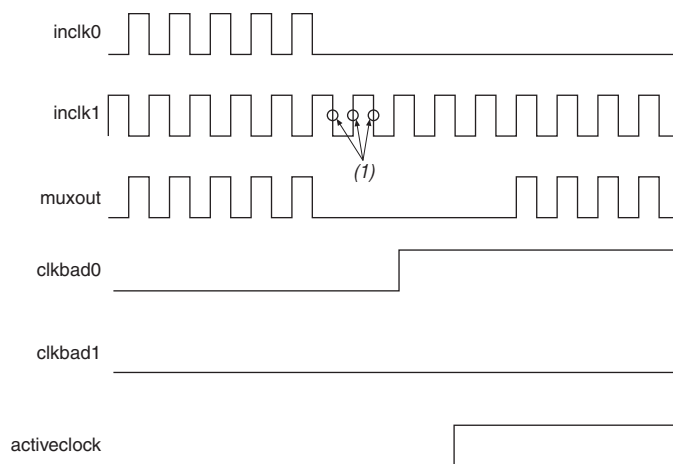
Note to Figure 5–18:

(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.

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]
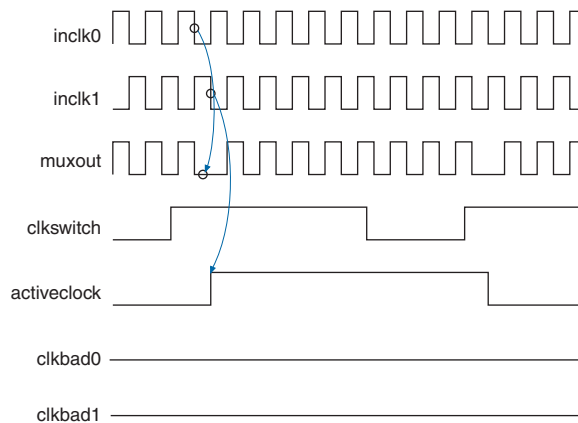
Note to Figure 5–19:

(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.

- 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
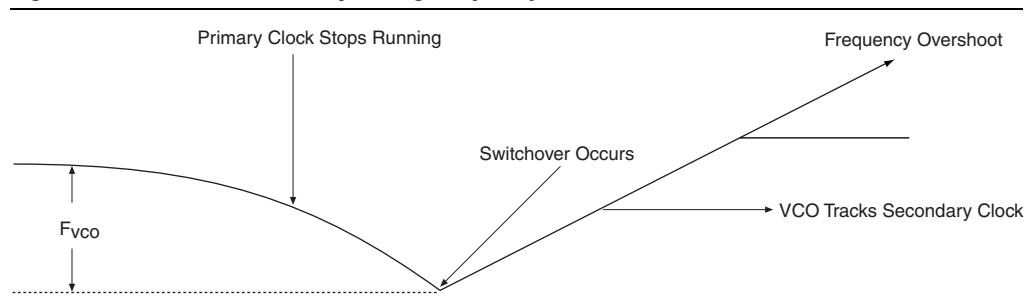
- Disable the system during switchover if the system is not tolerant to frequency variations during the PLL resynchronization period. You can use the `clkbad0` and `clkbad1` status signals to turn off the PFD (`pfdena = 0`) so the VCO maintains its last frequency. You can also use the switchover state 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.

Programmable Bandwidth

The PLL bandwidth is the measure of the PLL's ability to track the input clock and its associated jitter. PLLs of Cyclone IV devices provide advanced control of the PLL bandwidth using the programmable characteristics of the PLL loop, including loop filter and charge pump. The closed-loop gain 3-dB frequency in the PLL determines the PLL bandwidth. The bandwidth is approximately the unity gain point for open loop PLL response.

Phase Shift Implementation

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

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

■ Fine resolution using VCO phase taps

■ Coarse resolution using counter starting time

Fine resolution phase shifts are implemented by allowing any of the output counters (`C[4..0]`) or the M counter to use any of the eight phases of the VCO as the reference clock. This allows you to adjust the delay time with a fine resolution.

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

Equation 5–1. Fine Resolution Phase Shift

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

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

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

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

Equation 5–2. Coarse Resolution Phase Shift

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

C is the count value set for the counter delay time (this is the initial setting in the PLL usage section of the compilation report in the Quartus II software). If the initial value is 1, C − 1 = 0° phase shift.

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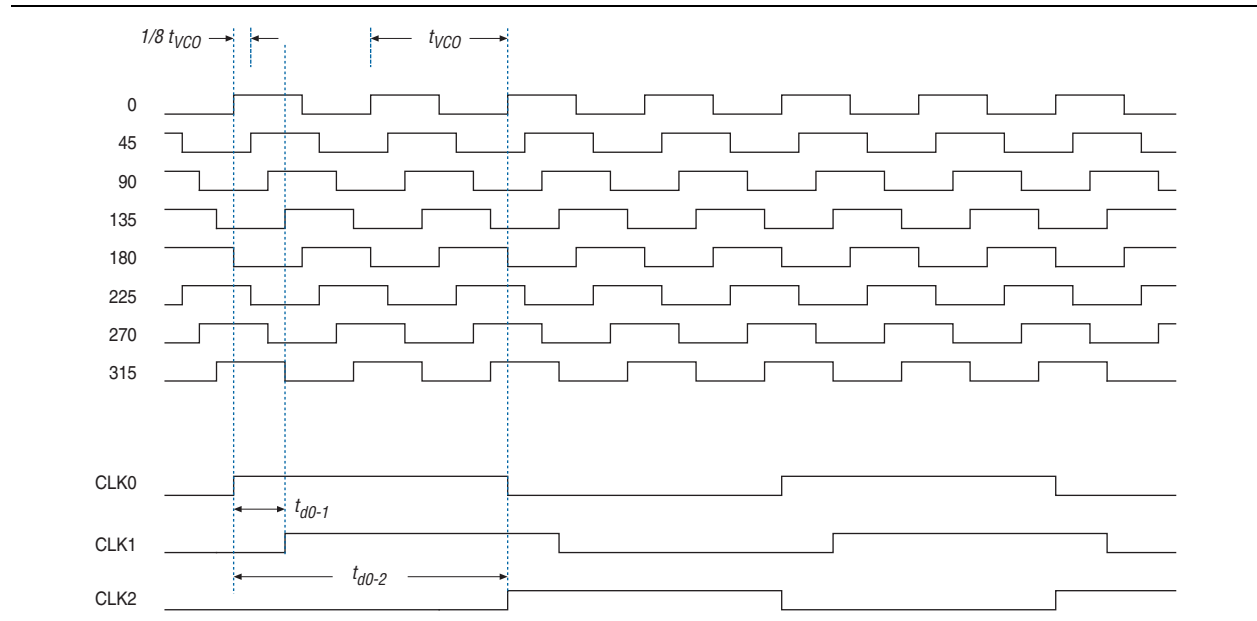
You can use the coarse and fine phase shifts to implement clock delays in Cyclone IV devices.

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.

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 5–22 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**
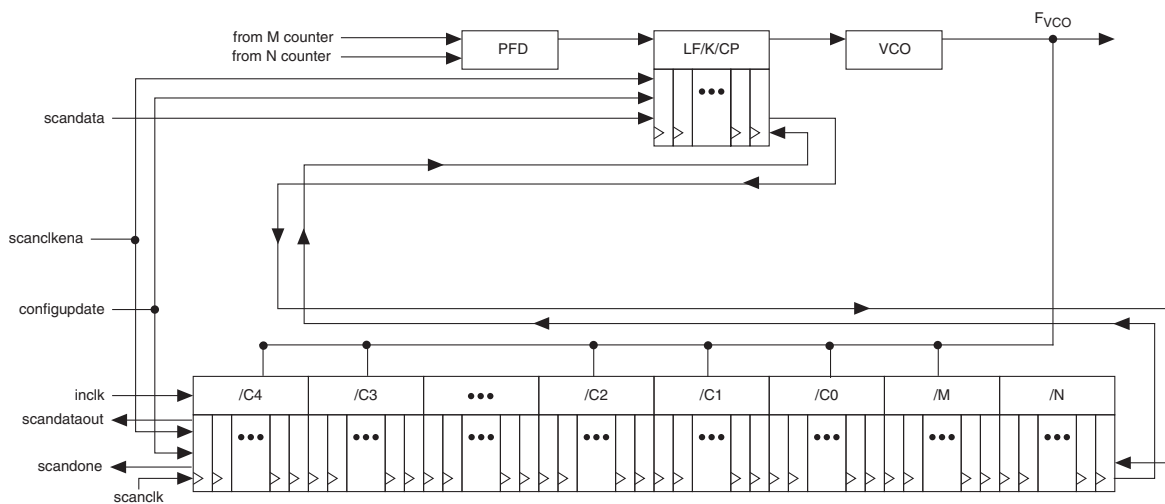
☞ The counter settings are updated synchronously to the clock frequency of the individual counters. Therefore, not all counters update simultaneously.

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.

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

Figure 5–23. PLL Reconfiguration Scan Chain
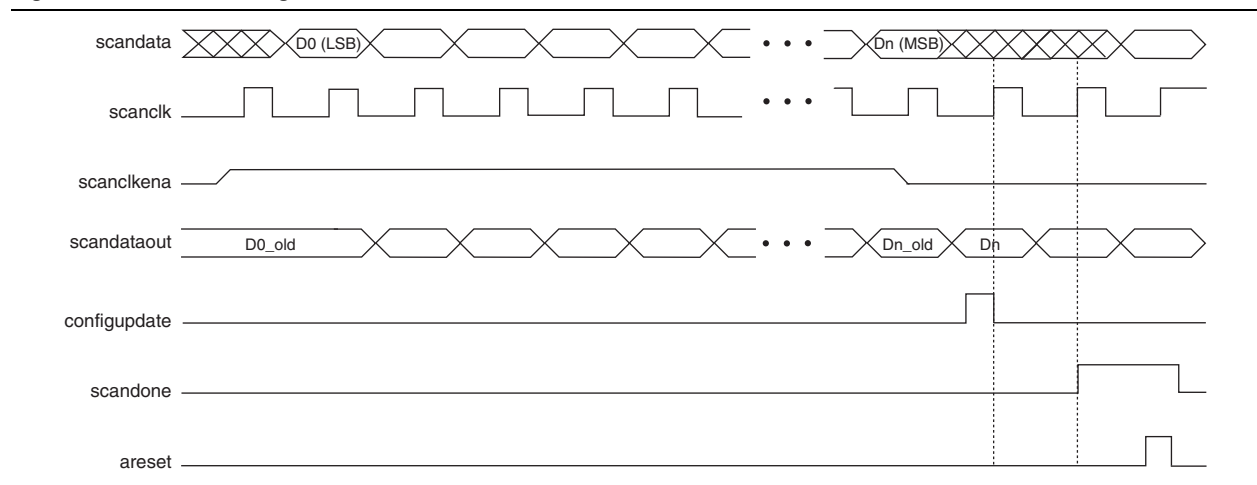
☞ When reconfiguring the counter clock frequency, the corresponding counter phase shift settings cannot be reconfigured using the same interface. You can reconfigure phase shifts in real time using the dynamic phase shift reconfiguration interface. If you reconfigure the counter frequency, but wish to 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

■ 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.

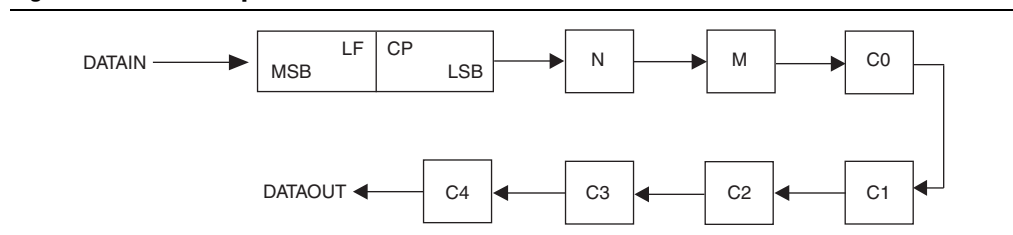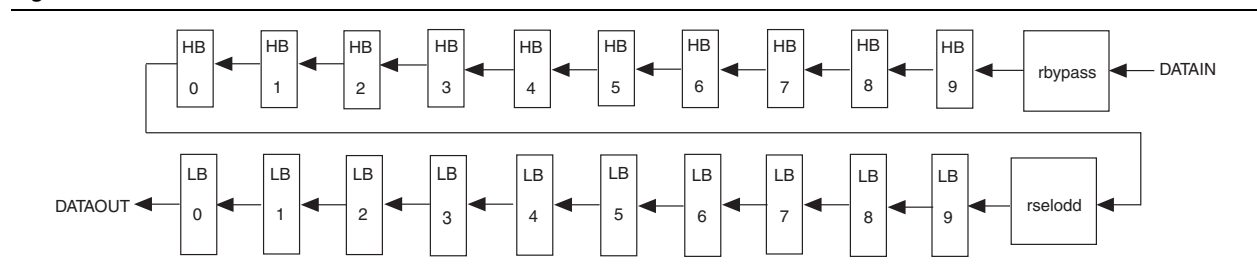**Figure 5–24. PLL Component Scan Chain Order**

Figure 5–25 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

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

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

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.

Figure 5–26 shows the dynamic phase shifting waveform.

Figure 5–26. PLL Dynamic Phase Shift
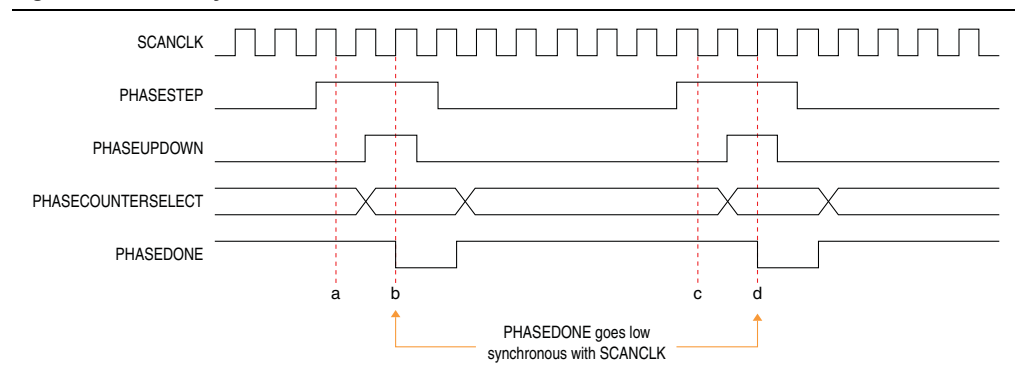
The PHASESTEP signal is latched on the negative edge of SCANCLK (a,c) and must remain asserted for at least two SCANCLK cycles. Deassert PHASESTEP after PHASEDONE goes low. On the second SCANCLK rising edge (b,d) after PHASESTEP is latched, the values of PHASEUPDOWN and PHASECOUNTERSELECT are latched 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.

Document Revision History

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

Table 5–14. Document Revision History

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

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:

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.

This chapter describes the I/O and high speed I/O capabilities and features offered in Cyclone® IV devices.

The I/O capabilities of Cyclone IV devices are driven by the diversification of I/O standards in many low-cost applications, and the significant increase in required I/O performance. Altera's objective is to create a device that accommodates your key board design needs with ease and flexibility.

The I/O flexibility of Cyclone IV devices is increased from the previous generation low-cost FPGAs by allowing all I/O standards to be selected on all I/O banks. Improvements to on-chip termination (OCT) support and the addition of true differential buffers have eliminated the need for external resistors in many applications, such as display system interfaces.

High-speed differential I/O standards have become popular in high-speed interfaces because of their significant advantages over single-ended I/O standards. The Cyclone IV devices support LVDS, BLVDS, RSDS, mini-LVDS, and PPDS. The transceiver reference clocks and the existing general-purpose I/O (GPIO) clock input features also support the LVDS I/O standards.

The Quartus® II software completes the solution with powerful pin planning features that allow you to plan and optimize I/O system designs even before the design files are available.

This chapter includes the following sections:

Feedback   Subscribe

Cyclone IV I/O Elements

Cyclone IV I/O elements (IOEs) contain a bidirectional I/O buffer and five registers for registering input, output, output-enable signals, and complete embedded bidirectional single-data rate transfer. I/O pins support various single-ended and differential I/O standards.

The IOE contains one input register, two output registers, and two output-enable (OE) registers. The two output registers and two OE registers are used for DDR applications. You can use input registers for fast setup times and output registers for fast clock-to-output times. Additionally, you can use OE registers for fast clock-to-output enable timing. You can use IOEs for input, output, or bidirectional data paths.

Figure 6–1 shows the Cyclone IV devices IOE structure for single data rate (SDR) operation.

Figure 6–1. Cyclone IV IOEs in a Bidirectional I/O Configuration for SDR Mode
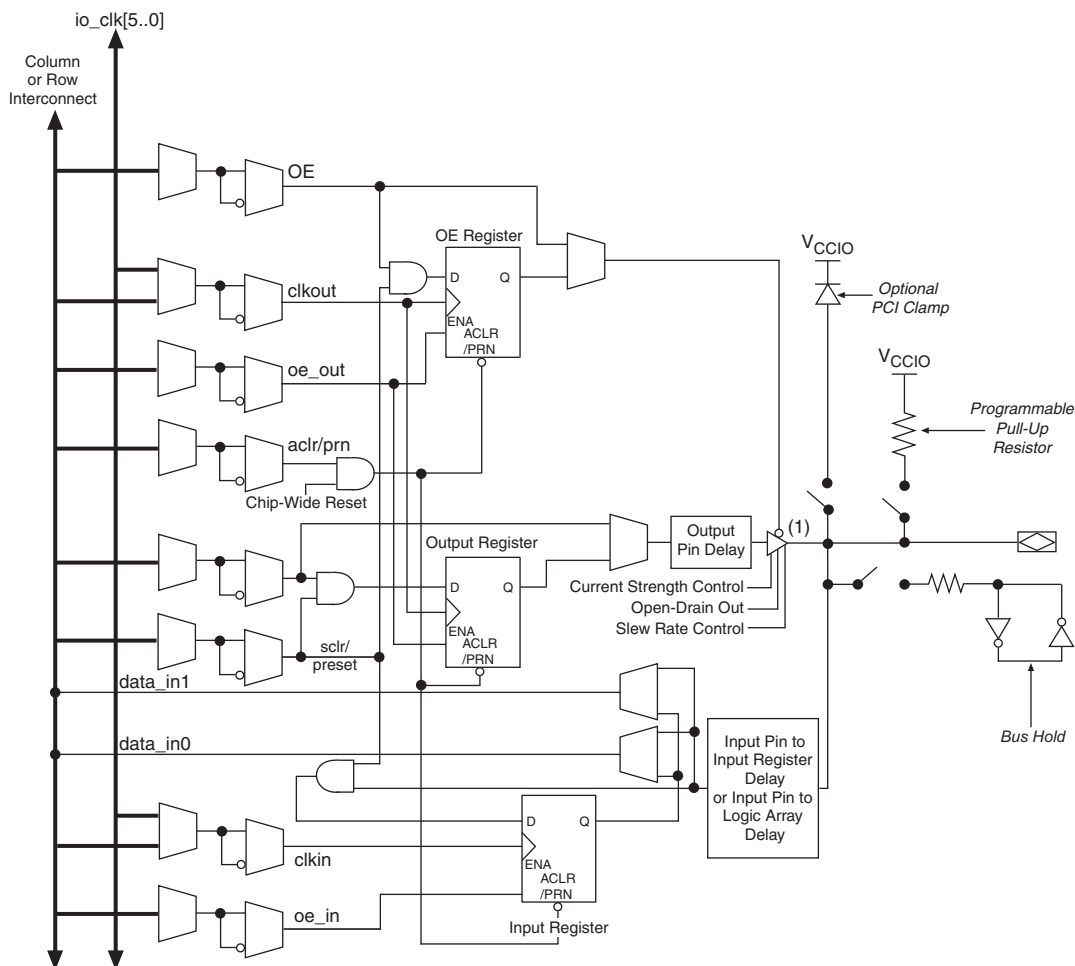
Note to Figure 6–1:

(1)   Tri-state control is not available for outputs configured with true differential I/O standards.

I/O Element Features

The Cyclone IV IOE offers a range of programmable features for an I/O pin. These features increase the flexibility of I/O utilization and provide a way to reduce the usage of external discrete components, such as pull-up resistors and diodes.

Programmable Current Strength

The output buffer for each Cyclone IV I/O pin has a programmable current strength control for certain I/O standards.

The LVTTL, LVCMOS, SSTL-2 Class I and II, SSTL-18 Class I and II, HSTL-18 Class I and II, HSTL-15 Class I and II, and HSTL-12 Class I and II I/O standards have several levels of current strength that you can control.

Table 6–2 on page 6–7 shows the possible settings for I/O standards with current strength control. These programmable current strength settings are a valuable tool in helping decrease the effects of simultaneously switching outputs (SSO) in conjunction with reducing system noise. The supported settings ensure that the device driver meets the specifications for IOH and IOL of the corresponding I/O standard.

☞ When you use programmable current strength, on-chip series termination (R_S OCT) is not available.

Slew Rate Control

The output buffer for each Cyclone IV I/O pin provides optional programmable output slew-rate control. Table 6–2 on page 6–7 shows the possible slew rate option and the Quartus II default slew rate setting. However, these fast transitions may introduce noise transients in the system. A slower slew rate reduces system noise, but adds a nominal delay to rising and falling edges. Because each I/O pin has an individual slew-rate control, you can specify the slew rate on a pin-by-pin basis. The slew-rate control affects both the rising and falling edges. Slew rate control is available for single-ended I/O standards with current strength of 8 mA or higher.

☞ You cannot use the programmable slew rate feature when using OCT with calibration.

☞ You cannot use the programmable slew rate feature when using the 3.0-V PCI, 3.0-V PCI-X, 3.3-V LVTTL, or 3.3-V LVCMOS I/O standards. Only the fast slew rate (default) setting is available.

Open-Drain Output

Cyclone IV devices provide an optional open-drain (equivalent to an open-collector) output for each I/O pin. This open-drain output enables the device to provide system-level control signals (for example, interrupt and write enable signals) that are asserted by multiple devices in your system.

Bus Hold

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

The bus-hold circuitry also pulls undriven pins away from the input threshold voltage in which noise can cause unintended high-frequency switching. You can select this feature individually for each I/O pin. The bus-hold output drives no higher than V_{CCIO} to prevent overdriving signals.

☞ If you enable the bus-hold feature, the device cannot use the programmable pull-up option. Disable the bus-hold feature when the I/O pin is configured for differential signals. Bus-hold circuitry is not available on dedicated clock pins.

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

For the specific sustaining current for each V_{CCIO} voltage level driven through the resistor and for the overdrive current used to identify the next driven input level, refer to the *Cyclone IV Device Datasheet* chapter.

Programmable Pull-Up Resistor

Each Cyclone IV device I/O pin provides an optional programmable pull-up resistor while in user mode. If you enable this feature for an I/O pin, the pull-up resistor holds the output to the V_{CCIO} level of the output pin's bank.

☞ If you enable the programmable pull-up resistor, the device cannot use the bus-hold feature. Programmable pull-up resistors are not supported on the dedicated configuration, JTAG, and dedicated clock pins.

☞ When the optional DEV_OE signal drives low, all I/O pins remains tri-stated even with the programmable pull-up option enabled.

Programmable Delay

The Cyclone IV IOE includes programmable delays to ensure zero hold times, minimize setup times, increase clock-to-output times, and delay the clock input signal.

A path in which a pin directly drives a register may require a programmable delay to ensure zero hold time, whereas a path in which a pin drives a register through combinational logic may not require the delay. Programmable delays minimize setup time. The Quartus II Compiler can program these delays to automatically minimize setup time while providing a zero hold time. Programmable delays can increase the register-to-pin delays for output registers. Each dual-purpose clock input pin provides a programmable delay to the global clock networks.

Table 6–1 shows the programmable delays for Cyclone IV devices.

Table 6–1. Cyclone IV Devices Programmable Delay Chain

Programmable Delay	Quartus II Logic Option
Input pin-to-logic array delay	Input delay from pin to internal cells
Input pin-to-input register delay	Input delay from pin to input register
Output pin delay	Delay from output register to output pin
Dual-purpose clock input pin delay	Input delay from dual-purpose clock pin to fan-out destinations

There are two paths in the IOE for an input to reach the logic array. Each of the two paths can have a different delay. This allows you to adjust delays from the pin to the internal logic element (LE) registers that reside in two different areas of the device. You must set the two combinational input delays with the input delay from pin to internal cells logic option in the Quartus II software for each path. If the pin uses the input register, one of the delays is disregarded and the delay is set with the input delay from pin to input register logic option in the Quartus II software.

The IOE registers in each I/O block share the same source for the preset or clear features. You can program preset or clear for each individual IOE, but you cannot use both features simultaneously. You can also program the registers to power-up high or low after configuration is complete. If programmed to power-up low, an asynchronous clear can control the registers. If programmed to power-up high, an asynchronous preset can control the registers. This feature prevents the inadvertent activation of the active-low input of another device upon power-up. If one register in an IOE uses a preset or clear signal, all registers in the IOE must use that same signal if they require preset or clear. Additionally, a synchronous reset signal is available for the IOE registers.

For more information about the input and output pin delay settings, refer to the *Area and Timing Optimization* chapter in volume 2 of the *Quartus II Handbook*.

PCI-Clamp Diode

Cyclone IV devices provide an optional PCI-clamp diode enabled input and output for each I/O pin. Dual-purpose configuration pins support the diode in user mode if the specific pins are not used as configuration pins for the selected configuration scheme. For example, if you are using the active serial (AS) configuration scheme, you cannot use the clamp diode on the ASDO and nCSO pins in user mode. Dedicated configuration pins do not support the on-chip diode.

The PCI-clamp diode is available for the following I/O standards:

- 3.3-V LVTTL
- 3.3-V LVCMOS
- 3.0-V LVTTL
- 3.0-V LVCMOS
- 2.5-V LVTTL/LVCMOS
- PCI
- PCI-X

If the input I/O standard is one of the listed standards, the PCI-clamp diode is enabled by default in the Quartus II software.

OCT Support

Cyclone IV devices feature OCT to provide I/O impedance matching and termination capabilities. OCT helps prevent reflections and maintain signal integrity while minimizing the need for external resistors in high pin-count ball grid array (BGA) packages. Cyclone IV devices provide I/O driver on-chip impedance matching and R_S OCT for single-ended outputs and bidirectional pins.

When using R_S OCT, programmable current strength is not available.

There are two ways to implement OCT in Cyclone IV devices:

- OCT with calibration
- OCT without calibration

Table 6–2 lists the I/O standards that support impedance matching and series termination.

Table 6–2. Cyclone IV Device I/O Features Support   (Part 1 of 2)

I/O Standard	IOH/IOL Current Strength Setting (mA) [1], [9]		R_S OCT with Calibration Setting, Ohm (Ω)		R_S OCT Without Calibration Setting, Ohm (Ω)		Cyclone IV E I/O Banks Support	Cyclone IV GX I/O Banks Support	Slew Rate Option [6]	PCI-clamp Diode Support
	Column I/O	Row I/O	Column I/O	Row I/O [8]	Column I/O	Row I/O [8]				
3.3-V LVTTL	4,8	4,8	—	—	—	—			—	✓
3.3-V LVCMOS	2	2	—	—	—	—			—	✓
3.0-V LVTTL	4,8,12,16	4,8,12,16	50,25	50,25	50,25	50,25			0,1,**2**	✓
3.0-V LVCMOS	4,8,12,16	4,8,12,16	50,25	50,25	50,25	50,25		3,4,5,6, 7,8,9		✓
3.0-V PCI/PCI-X	—	—	—	—	—	—			—	✓
2.5-V LVTTL/LVCMOS	4,8,12,16	4,8,12,16	50,25	50,25	50,25	50,25				✓
1.8-V LVTTL/LVCMOS	2,4,6,8,10,12,16	2,4,6,8,10,12,16	50,25	50,25	50,25	50,25				—
1.5-V LVCMOS	2,4,6,8,10,12,16	2,4,6,8,10,12,16	50,25	50,25	50,25	50,25				—
1.2-V LVCMOS	2,4,6,8,10,12	2,4,6,8,10	50,25	50	50,25	50	1,2,3,4, 5,6,7,8	4,5,6,7, 8		—
SSTL-2 Class I	8,12	8,12	50	50	50	50				—
SSTL-2 Class II	16	16	25	25	25	25			0,1,**2**	—
SSTL-18 Class I	8,10,12	8,10,12	50	50	50	50				—
SSTL-18 Class II	12,16	12,16	25	25	25	25		3,4,5,6, 7,8,9		—
HSTL-18 Class I	8,10,12	8,10,12	50	50	50	50				—
HSTL-18 Class II	16	16	25	25	25	25				—
HSTL-15 Class I	8,10,12	8,10,12	50	50	50	50				—
HSTL-15 Class II	16	16	25	25	25	25				—
HSTL-12 Class I	8,10,12	8,10	50	50	50	50		4,5,6,7, 8		—
HSTL-12 Class II	14	—	25	—	25	—	3,4,7,8	4,7,8		—
Differential SSTL-2 Class I [2], [7]	8,12	8,12	50	50	50	50				—
Differential SSTL-2 Class II [2], [7]	16	16	25	25	25	25				—
Differential SSTL-18 [2], [7]	8,10,12	—	50	—	50	—	1,2,3,4, 5,6,7,8	3,4,5,6, 7,8	0,1,**2**	—
Differential HSTL-18 [2], [7]	8,10,12	—	50	—	50	—				—
Differential HSTL-15 [2], [7]	8,10,12	—	50	—	50	—				—
Differential HSTL-12 [2], [7]	8,10,12	—	50	—	50	—	3,4,7,8	4,7,8		—

Table 6–2. Cyclone IV Device I/O Features Support (Part 2 of 2)

I/O Standard	IOH/IOL Current Strength Setting (mA) [1], [9]		R_S OCT with Calibration Setting, Ohm (Ω)		R_S OCT Without Calibration Setting, Ohm (Ω)		Cyclone IV E I/O Banks Support	Cyclone IV GX I/O Banks Support	Slew Rate Option [6]	PCI-clamp Diode Support
	Column I/O	Row I/O	Column I/O	Row I/O [8]	Column I/O	Row I/O [8]				
BLVDS	8,12,16	8,12,16	—	—	—	—	1,2,3,4, 5,6,7,8	3,4,5,6, 7,8	0,1,**2**	—
LVDS [3]	—	—	—	—	—	—		5,6	—	—
PPDS [3], [4]	—	—	—	—	—	—			—	—
RSDS and mini-LVDS [3], [4]	—	—	—	—	—	—			—	—
Differential LVPECL [5]	—	—	—	—	—	—		3,4,5,6, 7,8	—	—

Notes to Table 6–2:

(1) The default current strength setting in the Quartus II software is 50-Ω OCT without calibration for all non-voltage reference and HSTL/SSTL Class I I/O standards. The default setting is 25-Ω OCT without calibration for HSTL/SSTL Class II I/O standards.

(2) The differential SSTL-18 and SSTL-2, differential HSTL-18, HSTL-15, and HSTL-12 I/O standards are supported only on clock input pins and PLL output clock pins.

(3) True differential (PPDS, LVDS, mini-LVDS, and RSDS I/O standards) outputs are supported in row I/O banks 1, 2, 5, and 6 only for Cyclone IV E devices and right I/O banks 5 and 6 only for Cyclone IV GX devices. Differential outputs in column I/O banks require an external resistor network.

(4) This I/O standard is supported for outputs only.

(5) This I/O standard is supported for clock inputs only

(6) The default Quartus II slew rate setting is in bold; **2** for all I/O standards that supports slew rate option.

(7) Differential SSTL-18, differential HSTL-18, HSTL-15, and HSTL-12 I/O standards do not support Class II output.

(8) Cyclone IV GX devices only support right I/O pins.

(9) Altera not only offers current strength that meets the industrial standard specification but also other additional current strengths.

☞ For more details about the differential I/O standards supported in Cyclone IV I/O banks, refer to "High-Speed I/O Interface" on page 6–24.

On-Chip Series Termination with Calibration

Cyclone IV devices support R_S OCT with calibration in the top, bottom, and right I/O banks. The R_S OCT calibration circuit compares the total impedance of the I/O buffer to the external 25-Ω ±1% or 50-Ω ±1% resistors connected to the RUP and RDN pins, and dynamically adjusts the I/O buffer impedance until they match (as shown in Figure 6–2).

The R_S shown in Figure 6–2 is the intrinsic impedance of the transistors that make up the I/O buffer.

Figure 6–2. Cyclone IV Devices R_S OCT with Calibration
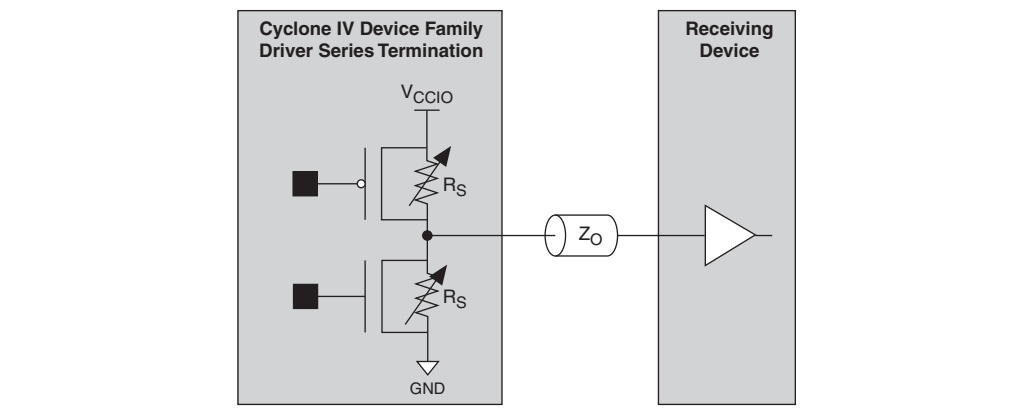
OCT with calibration is achieved using the OCT calibration block circuitry. There is one OCT calibration block in each of I/O banks 2, 4, 5, and 7 for Cyclone IV E devices and I/O banks 4, 5, and 7 for Cyclone IV GX devices. Each calibration block supports each side of the I/O banks. Because there are two I/O banks sharing the same calibration block, both banks must have the same V_{CCIO} if both banks enable OCT calibration. If two related banks have different V_{CCIO}, only the bank in which the calibration block resides can enable OCT calibration.

Figure 6–10 on page 6–18 shows the top-level view of the OCT calibration blocks placement.

Each calibration block comes with a pair of RUP and RDN pins. When used for calibration, the RUP pin is connected to V_{CCIO} through an external 25-Ω ±1% or 50-Ω ±1% resistor for an R_S OCT value of 25 Ω or 50 Ω, respectively. The RDN pin is connected to GND through an external 25-Ω ±1% or 50-Ω ±1% resistor for an R_S OCT value of 25 Ω or 50 Ω, respectively. The external resistors are compared with the internal resistance using comparators. The resultant outputs of the comparators are used by the OCT calibration block to dynamically adjust buffer impedance.

☞ During calibration, the resistance of the RUP and RDN pins varies.

Figure 6–3 shows the external calibration resistors setup on the RUP and RDN pins and the associated OCT calibration circuitry.

Figure 6–3. Cyclone IV Devices R$_S$ OCT with Calibration Setup
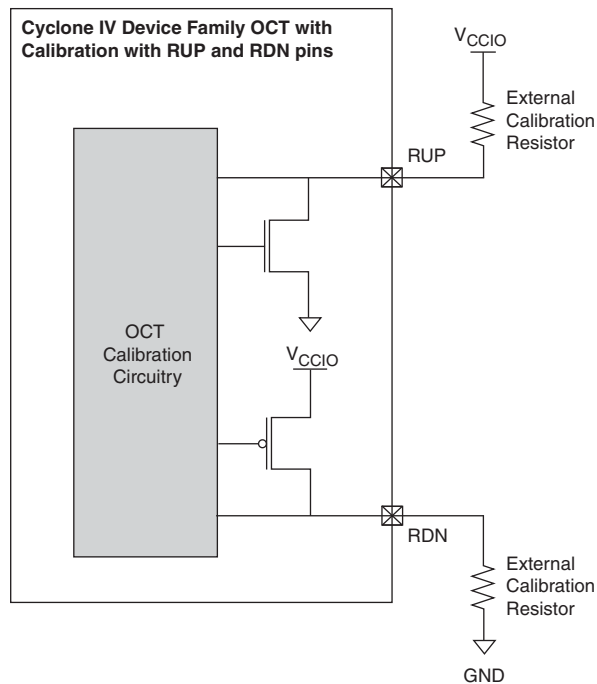
RUP and RDN pins go to a tri-state condition when calibration is completed or not running. These two pins are dual-purpose I/Os and function as regular I/Os if you do not use the calibration circuit.

On-Chip Series Termination Without Calibration

Cyclone IV devices support driver impedance matching to match the impedance of the transmission line, which is typically 25 or 50 Ω. When used with the output drivers, OCT sets the output driver impedance to 25 or 50 Ω. Cyclone IV devices also support I/O driver series termination (R$_S$ = 50 Ω) for SSTL-2 and SSTL-18.

Figure 6–4 shows the single-ended I/O standards for OCT without calibration. The R$_S$ shown is the intrinsic transistor impedance.

Figure 6–4. Cyclone IV Devices R$_S$ OCT Without Calibration
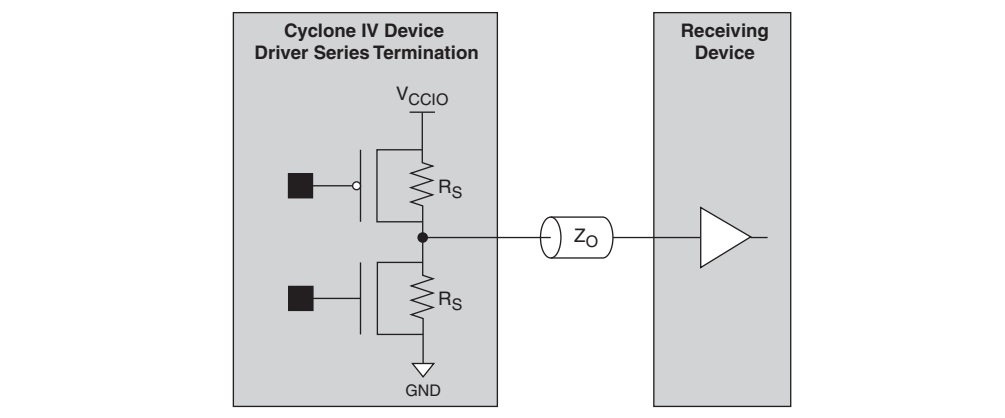
All I/O banks and I/O pins support impedance matching and series termination. Dedicated configuration pins and JTAG pins do not support impedance matching or series termination.

R$_S$ OCT is supported on any I/O bank. V$_{CCIO}$ and V$_{REF}$ must be compatible for all I/O pins to enable R$_S$ OCT in a given I/O bank. I/O standards that support different R$_S$ values can reside in the same I/O bank as long as their V$_{CCIO}$ and V$_{REF}$ do not conflict.

Impedance matching is implemented using the capabilities of the output driver and is subject to a certain degree of variation, depending on the process, voltage, and temperature.

For more information about tolerance specification, refer to the *Cyclone IV Device Datasheet* chapter.

I/O Standards

Cyclone IV devices support multiple single-ended and differential I/O standards. Cyclone IV devices support 3.3-, 3.0-, 2.5-, 1.8-, 1.5-, and 1.2-V I/O standards.

Table 6–3 summarizes I/O standards supported by Cyclone IV devices and which I/O pins support them.

Table 6–3. Cyclone IV Devices Supported I/O Standards and Constraints (Part 1 of 3)

I/O Standard	Type	Standard Support	V$_{CCIO}$ Level (in V)		Column I/O Pins			Row I/O Pins [1]	
			Input	Output	CLK, DQS	PLL_OUT	User I/O Pins	CLK, DQS	User I/O Pins
3.3-V LVTTL, 3.3-V LVCMOS [2]	Single-ended	JESD8-B	3.3/3.0/2.5 [3]	3.3	✓	✓	✓	✓	✓
3.0-V LVTTL, 3.0-V LVCMOS [2]	Single-ended	JESD8-B	3.3/3.0/2.5 [3]	3.0	✓	✓	✓	✓	✓

Table 6–3. Cyclone IV Devices Supported I/O Standards and Constraints (Part 2 of 3)

I/O Standard	Type	Standard Support	V_{CCIO} Level (in V)		Column I/O Pins			Row I/O Pins [1]	
			Input	Output	CLK, DQS	PLL_OUT	User I/O Pins	CLK, DQS	User I/O Pins
2.5-V LVTTL / LVCMOS	Single-ended	JESD8-5	3.3/3.0/2.5 [3]	2.5	✓	✓	✓	✓	✓
1.8-V LVTTL / LVCMOS	Single-ended	JESD8-7	1.8/1.5 [3]	1.8	✓	✓	✓	✓	✓
1.5-V LVCMOS	Single-ended	JESD8-11	1.8/1.5 [3]	1.5	✓	✓	✓	✓	✓
1.2-V LVCMOS [4]	Single-ended	JESD8-12A	1.2	1.2	✓	✓	✓	✓	✓
SSTL-2 Class I, SSTL-2 Class II	voltage-referenced	JESD8-9A	2.5	2.5	✓	✓	✓	✓	✓
SSTL-18 Class I, SSTL-18 Class II	voltage-referenced	JESD815	1.8	1.8	✓	✓	✓	✓	✓
HSTL-18 Class I, HSTL-18 Class II	voltage-referenced	JESD8-6	1.8	1.8	✓	✓	✓	✓	✓
HSTL-15 Class I, HSTL-15 Class II	voltage-referenced	JESD8-6	1.5	1.5	✓	✓	✓	✓	✓
HSTL-12 Class I	voltage-referenced	JESD8-16A	1.2	1.2	✓	✓	✓	✓	✓
HSTL-12 Class II [9]	voltage-referenced	JESD8-16A	1.2	1.2	✓	✓	✓	—	—
PCI and PCI-X	Single-ended	—	3.0	3.0	✓	✓	✓	✓	✓
Differential SSTL-2 Class I or Class II [5]	Differential	JESD8-9A	—	2.5	—	✓	—	—	—
			2.5	—	—	✓	—	✓	—
Differential SSTL-18 Class I or Class II [5]	Differential	JESD815	—	1.8	—	✓	—	—	—
			1.8	—	—	✓	—	✓	—
Differential HSTL-18 Class I or Class II [5]	Differential	JESD8-6	—	1.8	—	✓	—	—	—
			1.8	—	—	✓	—	✓	—
Differential HSTL-15 Class I or Class II [5]	Differential	JESD8-6	—	1.5	—	✓	—	—	—
			1.5	—	—	✓	—	✓	—
Differential HSTL-12 Class I or Class II [5]	Differential	JESD8-16A	—	1.2	—	✓	—	—	—
			1.2	—	—	✓	—	✓	—
PPDS [6]	Differential	—	—	2.5	—	✓	✓	—	✓
LVDS [10]	Differential	ANSI/TIA/ EIA-644	2.5	2.5	✓	✓	✓	✓	✓
RSDS and mini-LVDS [6]	Differential	—	—	2.5	—	✓	✓	—	✓
BLVDS [8]	Differential	—	2.5	2.5	—	—	✓	—	✓

Table 6–3. Cyclone IV Devices Supported I/O Standards and Constraints (Part 3 of 3)

| I/O Standard | Type | Standard Support | V_{CCIO} Level (in V) | | Column I/O Pins | | | Row I/O Pins [1] | |
			Input	Output	CLK, DQS	PLL_OUT	User I/O Pins	CLK, DQS	User I/O Pins
LVPECL [7]	Differential	—	2.5	—	✔	—	—	✔	—

Notes to Table 6–3:

(1) Cyclone IV GX devices only support right I/O pins.

(2) The PCI-clamp diode must be enabled for 3.3-V/3.0-V LVTTL/LVCMOS.

(3) The Cyclone IV architecture supports the MultiVolt I/O interface feature that allows Cyclone IV devices in all packages to interface with I/O systems that have different supply voltages.

(4) Cyclone IV GX devices do not support 1.2-V V_{CCIO} in banks 3 and 9. I/O pins in bank 9 are dual-purpose I/O pins that are used as configuration or GPIO pins. Configuration scheme is not support at 1.2 V, therefore bank 9 can not be powered up at 1.2-V V_{CCIO}.

(5) Differential HSTL and SSTL outputs use two single-ended outputs with the second output programmed as inverted. Differential HSTL and SSTL inputs treat differential inputs as two single-ended HSTL and SSTL inputs and only decode one of them. Differential HSTL and SSTL are only supported on CLK pins.

(6) PPDS, mini-LVDS, and RSDS are only supported on output pins.

(7) LVPECL is only supported on clock inputs.

(8) Bus LVDS (BLVDS) output uses two single-ended outputs with the second output programmed as inverted. BLVDS input uses LVDS input buffer.

(9) 1.2-V HSTL input is supported at both column and row I/Os regardless of Class I or Class II.

(10) True LVDS, RSDS, and mini-LVDS I/O standards are supported in left and right I/O pins, while emulated LVDS, RSDS, and mini-LVDS I/O standards are supported in the top, bottom, and right I/O pins.

Cyclone IV devices support PCI and PCI-X I/O standards at 3.0-V V_{CCIO}. The 3.0-V PCI and PCI-X I/O are fully compatible for direct interfacing with 3.3-V PCI systems without requiring any additional components. The 3.0-V PCI and PCI-X outputs meet the V_{IH} and V_{IL} requirements of 3.3-V PCI and PCI-X inputs with sufficient noise margin.

For more information about the 3.3/3.0/2.5-V LVTTL & LVCMOS multivolt I/O support, refer to *AN 447: Interfacing Cyclone III and Cyclone IV Devices with 3.3/3.0/2.5-V LVTTL/LVCMOS I/O Systems*.

Termination Scheme for I/O Standards

This section describes recommended termination schemes for voltage-referenced and differential I/O standards.

The 3.3-V LVTTL, 3.0-V LVTTL and LVCMOS, 2.5-V LVTTL and LVCMOS, 1.8-V LVTTL and LVCMOS, 1.5-V LVCMOS, 1.2-V LVCMOS, 3.0-V PCI, and PCI-X I/O standards do not specify a recommended termination scheme per the JEDEC standard

Voltage-Referenced I/O Standard Termination

Voltage-referenced I/O standards require an input reference voltage (V_{REF}) and a termination voltage (V_{TT}). The reference voltage of the receiving device tracks the termination voltage of the transmitting device, as shown in Figure 6–5 and Figure 6–6.

Figure 6–5. Cyclone IV Devices HSTL I/O Standard Termination
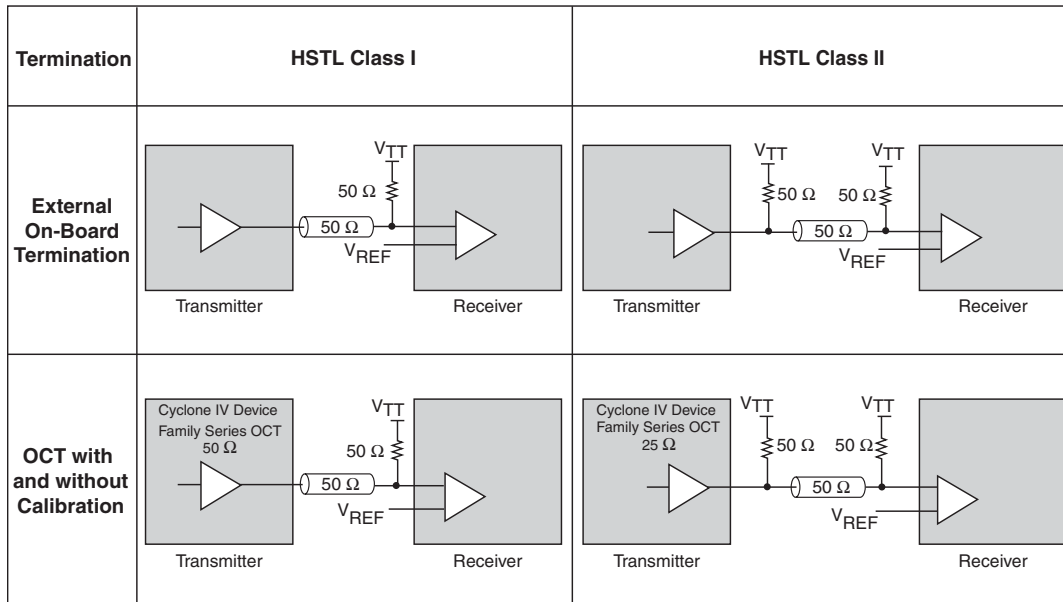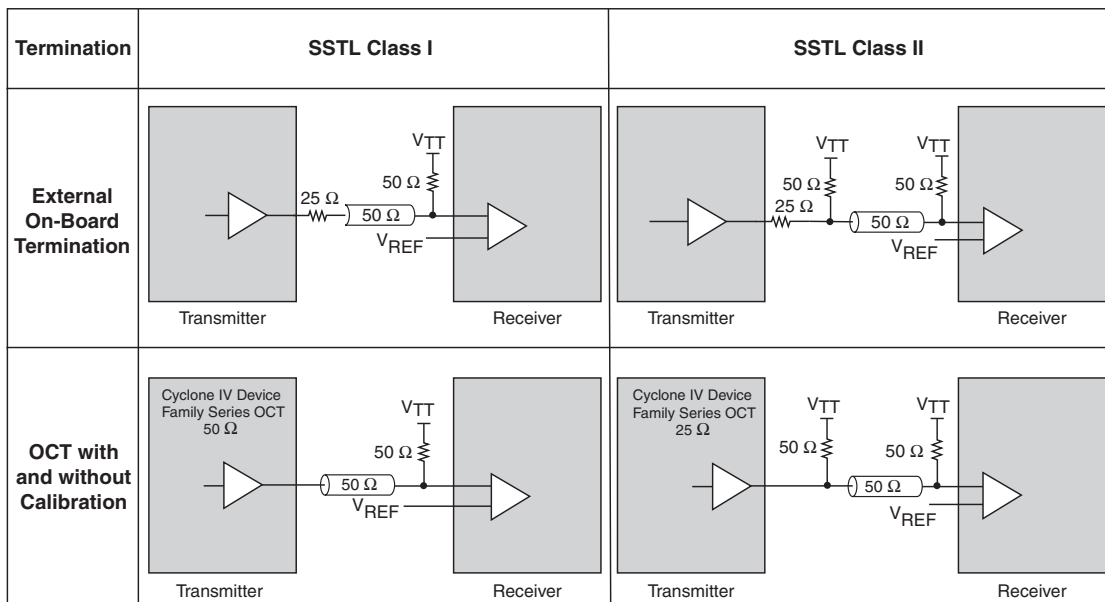

Figure 6–6. Cyclone IV Devices SSTL I/O Standard Termination

Differential I/O Standard Termination

Differential I/O standards typically require a termination resistor between the two signals at the receiver. The termination resistor must match the differential load impedance of the bus (refer to Figure 6–7 and Figure 6–8).

Cyclone IV devices support differential SSTL-2 and SSTL-18, differential HSTL-18, HSTL-15, and HSTL-12, PPDS, LVDS, RSDS, mini-LVDS, and differential LVPECL.

Figure 6–7. Cyclone IV Devices Differential HSTL I/O Standard Class I and Class II Interface and Termination
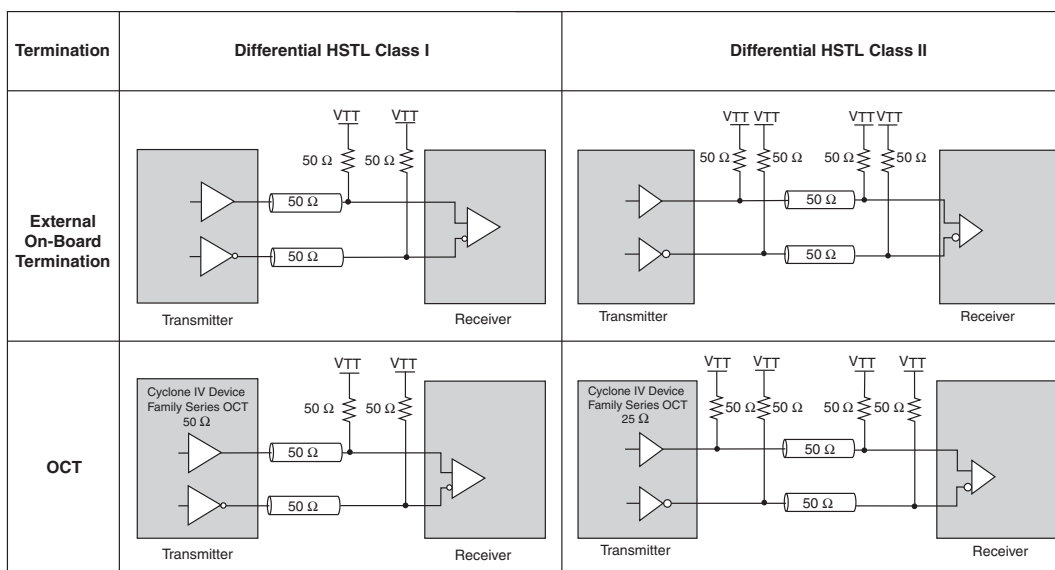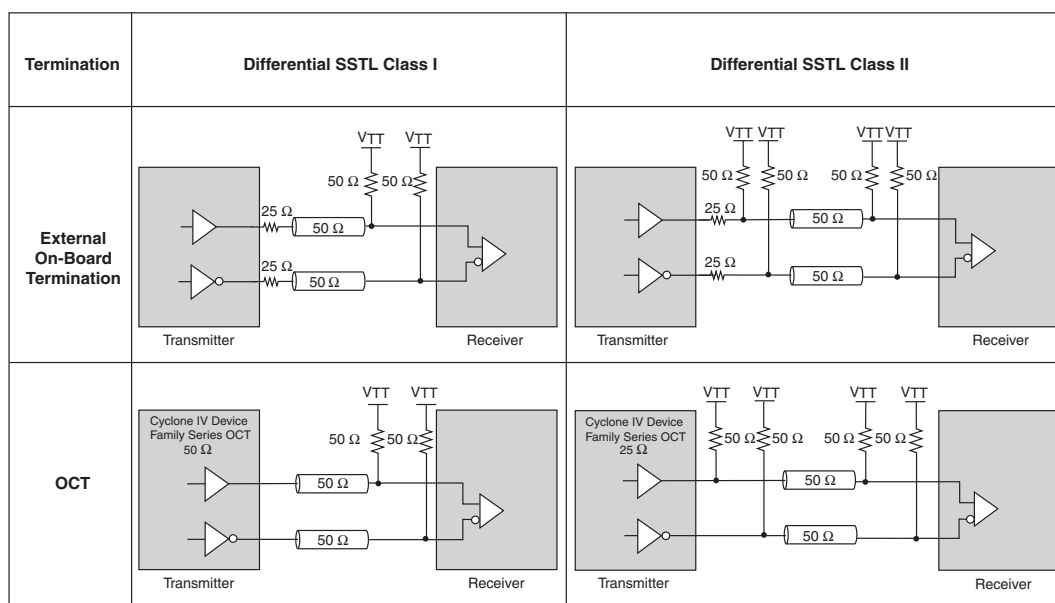

Figure 6–8. Cyclone IV Devices Differential SSTL I/O Standard Class I and Class II Interface and Termination [1]

Note to Figure 6–8:

(1) Only Differential SSTL-2 I/O standard supports Class II output.

I/O Banks

I/O pins on Cyclone IV devices are grouped together into I/O banks. Each bank has a separate power bus.

Cyclone IV E devices have eight I/O banks, as shown in Figure 6–9. Each device I/O pin is associated with one I/O bank. All single-ended I/O standards are supported in all banks except HSTL-12 Class II, which is only supported in column I/O banks. All differential I/O standards are supported in all banks. The only exception is HSTL-12 Class II, which is only supported in column I/O banks.

Cyclone IV GX devices have up to ten I/O banks and two configuration banks, as shown in Figure 6–10 on page 6–18 and Figure 6–11 on page 6–19. The Cyclone IV GX configuration I/O bank contains three user I/O pins that can be used as normal user I/O pins if they are not used in configuration modes. Each device I/O pin is associated with one I/O bank. All single-ended I/O standards are supported except HSTL-12 Class II, which is only supported in column I/O banks. All differential I/O standards are supported in top, bottom, and right I/O banks. The only exception is HSTL-12 Class II, which is only supported in column I/O banks.

The entire left side of the Cyclone IV GX devices contain dedicated high-speed transceiver blocks for high speed serial interface applications. There are a total of 2, 4, and 8 transceiver channels for Cyclone IV GX devices, depending on the density and package of the device. For more information about the transceiver channels supported, refer to Figure 6–10 on page 6–18 and Figure 6–11 on page 6–19.

Figure 6–9 shows the overview of Cyclone IV E I/O banks.

Figure 6–9. Cyclone IV E I/O Banks *(1)*, *(2)*
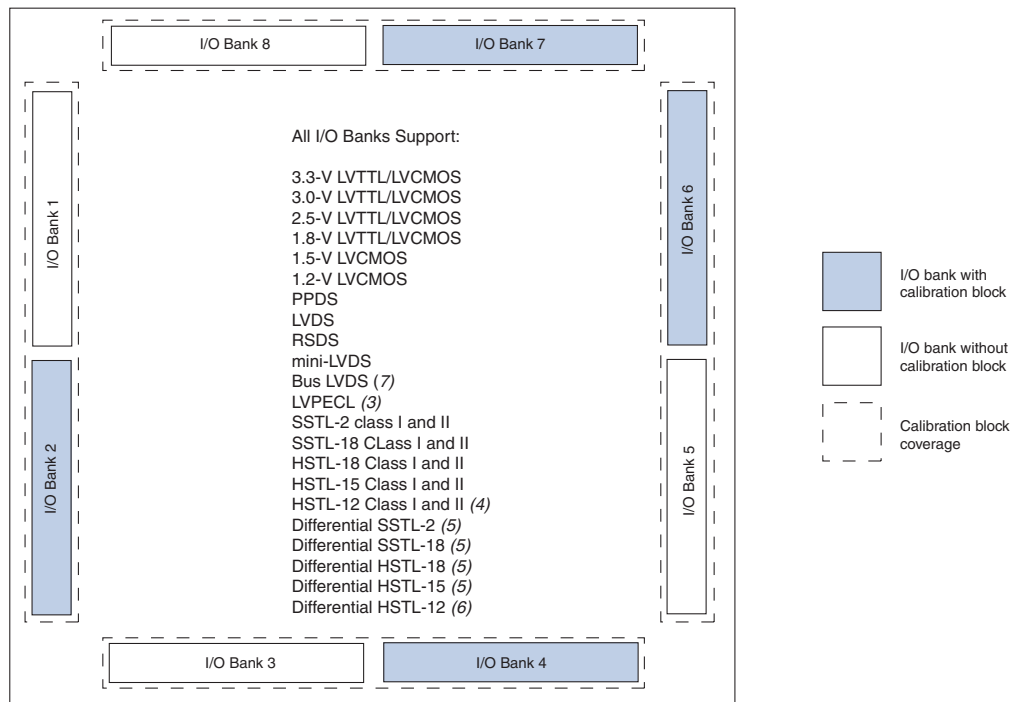
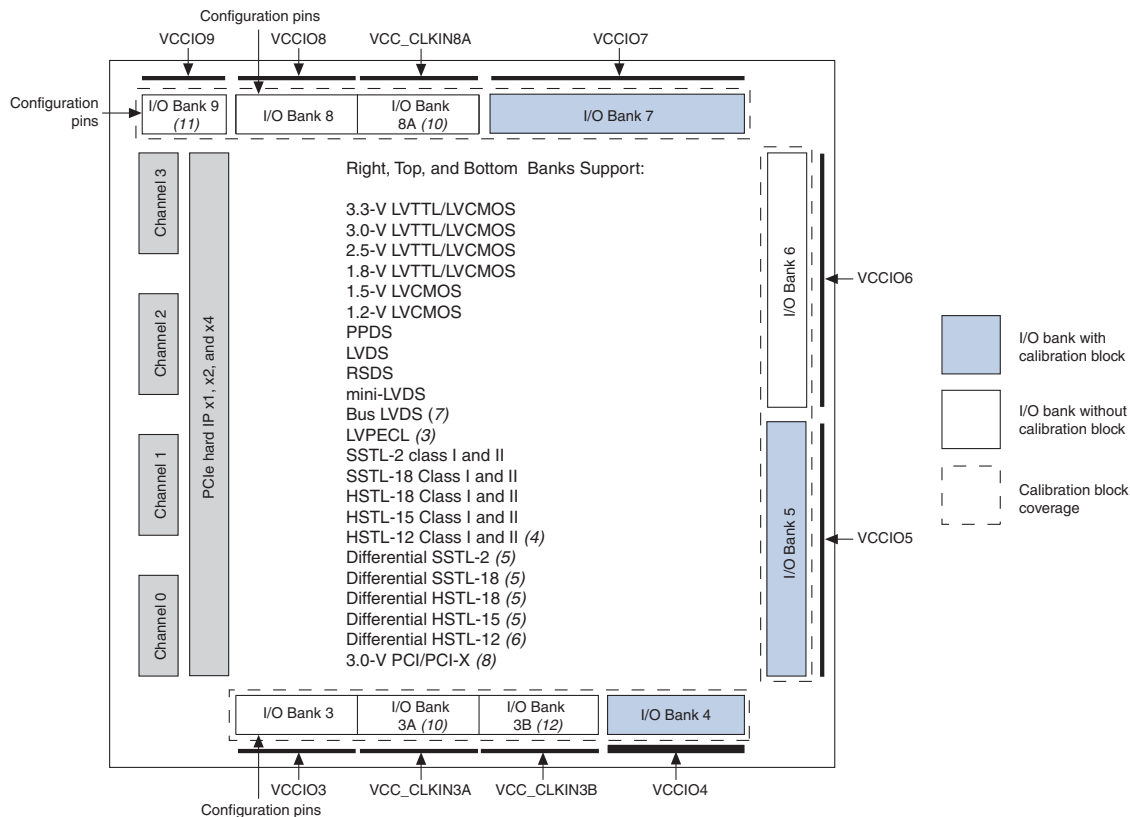

Notes to Figure 6–9:

(1) This is a top view of the silicon die. This is only a graphical representation. For exact pin locations, refer to the pin list and the Quartus II software.

(2) True differential (PPDS, LVDS, mini-LVDS, and RSDS I/O standards) outputs are supported in row I/O banks 1, 2, 5, and 6 only. External resistors are needed for the differential outputs in column I/O banks.

(3) The LVPECL I/O standard is only supported on clock input pins. This I/O standard is not supported on output pins.

(4) The HSTL-12 Class II is supported in column I/O banks 3, 4, 7, and 8 only.

(5) The differential SSTL-18 and SSTL-2, differential HSTL-18, and HSTL-15 I/O standards are supported only on clock input pins and phase-locked loops (PLLs) output clock pins. Differential SSTL-18, differential HSTL-18, and HSTL-15 I/O standards do not support Class II output.

(6) The differential HSTL-12 I/O standard is only supported on clock input pins and PLL output clock pins. Differential HSTL-12 Class II is supported only in column I/O banks 3, 4, 7, and 8.

(7) BLVDS output uses two single-ended outputs with the second output programmed as inverted. BLVDS input uses true LVDS input buffer.

Figure 6–10 and Figure 6–11 show the overview of Cyclone IV GX I/O banks.

Figure 6–10. Cyclone IV GX I/O Banks for EP4CGX15, EP4CGX22, and EP4CGX30 [1], [2], [9]

Notes to Figure 6–10:

(1) This is a top view of the silicon die. For exact pin locations, refer to the pin list and the Quartus II software. Channels 2 and 3 are not available in EP4CGX15 and F169 package type in EP4CGX22 and EP4CGX30 devices.

(2) True differential (PPDS, LVDS, mini-LVDS, and RSDS I/O standards) outputs are supported in row I/O banks 5 and 6 only. External resistors are needed for the differential outputs in column I/O banks.

(3) The LVPECL I/O standard is only supported on clock input pins. This I/O standard is not supported on output pins.

(4) The HSTL-12 Class II is supported in column I/O banks 4, 7, and 8.

(5) The differential SSTL-18 and SSTL-2, differential HSTL-18, and HSTL-15 I/O standards are supported only on clock input pins and phase-locked loops (PLLs) output clock pins. PLL output clock pins do not support Class II interface type of differential SSTL-18, HSTL-18, HSTL-15, and HSTL-12 I/O standards.

(6) The differential HSTL-12 I/O standard is only supported on clock input pins and PLL output clock pins. Differential HSTL-12 Class II is supported only in column I/O banks 4, 7, and 8.

(7) BLVDS output uses two single-ended outputs with the second output programmed as inverted. BLVDS input uses the LVDS input buffer.

(8) The PCI-X I/O standard does not meet the IV curve requirement at the linear region.

(9) The OCT block is located in the shaded banks 4, 5, and 7.

(10) There are two dedicated clock input I/O banks (I/O bank 3A and I/O bank 8A) that can be used for either high-speed serial interface (HSSI) input reference clock pins or clock input pins.

(11) There are dual-purpose I/O pins in bank 9. If input pins with VREF I/O standards are used on these dual-purpose I/O pins during user mode, they share the VREF pin in bank 8. These dual-purpose IO pins in bank 9 when used in user mode also support R_S OCT without calibration and they share the OCT block with bank 8.

(12) There are four dedicated clock input in I/O bank 3B for the EP4CGX30F484 device that can be used for either HSSI input reference clock pins or clock input pins.

Figure 6–11. Cyclone IV GX I/O Banks for EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 [1], [2], [9]
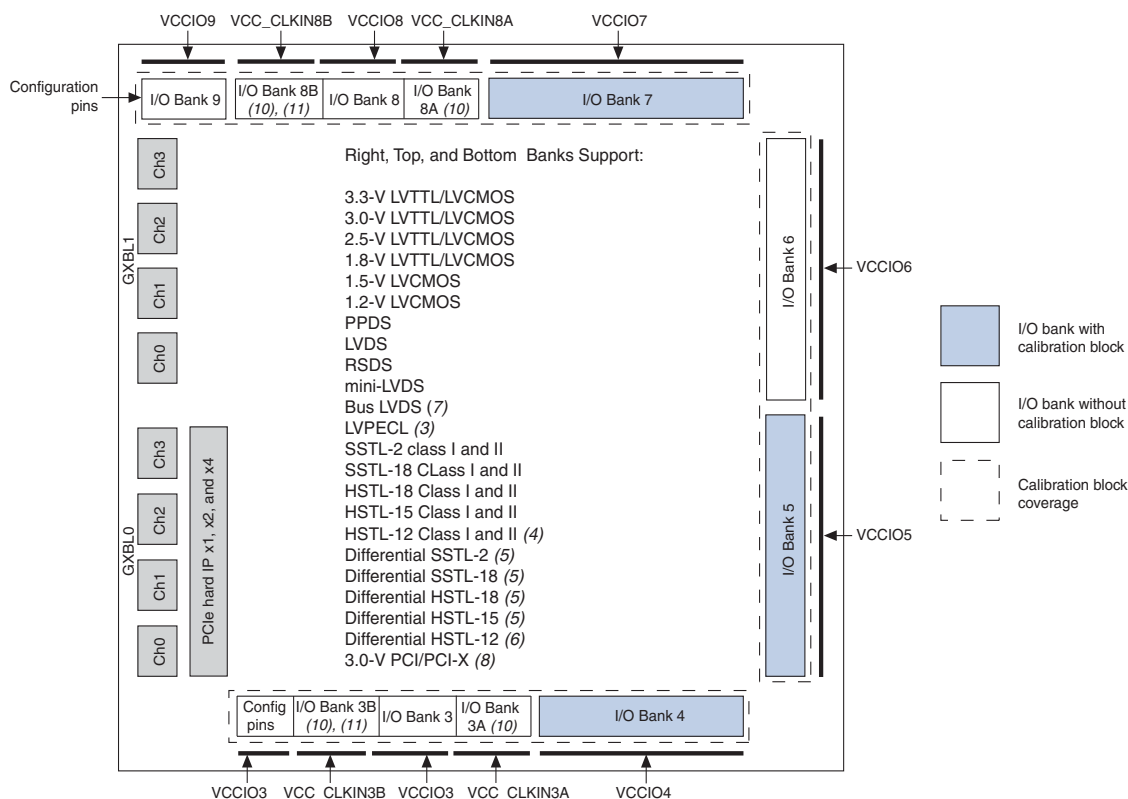
Notes to Figure 6–11:

(1) This is a top view of the silicon die. For exact pin locations, refer to the pin list and the Quartus II software.

(2) True differential (PPDS, LVDS, mini-LVDS, and RSDS I/O standards) outputs are supported in row I/O banks 5 and 6 only. External resistors are needed for the differential outputs in column I/O banks.

(3) The LVPECL I/O standard is only supported on clock input pins. This I/O standard is not supported on output pins.

(4) The HSTL-12 Class II is supported in column I/O banks 4, 7, and 8.

(5) The differential SSTL-18 and SSTL-2, differential HSTL-18, and HSTL-15 I/O standards are supported only on clock input pins and phase-locked loops (PLLs) output clock pins. PLL output clock pins do not support Class II interface type of differential SSTL-18, HSTL-18, HSTL-15, and HSTL-12 I/O standards.

(6) The differential HSTL-12 I/O standard is only supported on clock input pins and PLL output clock pins. Differential HSTL-12 Class II is supported only in column I/O banks 4, 7, and 8.

(7) BLVDS output uses two single-ended outputs with the second output programmed as inverted. BLVDS input uses the LVDS input buffer.

(8) The PCI-X I/O standard does not meet the IV curve requirement at the linear region.

(9) The OCT block is located in the shaded banks 4, 5, and 7.

(10) The dedicated clock input I/O banks 3A, 3B, 8A, and 8B can be used either for HSSI input reference clock pins or clock input pins.

(11) Single-ended clock input support is available for dedicated clock input I/O banks 3B and 8B.

Each Cyclone IV I/O bank has a VREF bus to accommodate voltage-referenced I/O standards. Each VREF pin is the reference source for its V_{REF} group. If you use a V_{REF} group for voltage-referenced I/O standards, connect the VREF pin for that group to the appropriate voltage level. If you do not use all the V_{REF} groups in the I/O bank for voltage-referenced I/O standards, you can use the VREF pin in the unused voltage-referenced groups as regular I/O pins. For example, if you have SSTL-2 Class I input pins in I/O bank 1 and they are all placed in the VREFB1N[0] group, VREFB1N[0] must be powered with 1.25 V, and the remaining VREFB1N[1..3] pins (if available) are used as I/O pins. If multiple V_{REF} groups are used in the same I/O bank, the VREF pins must all be powered by the same voltage level because the VREF pins are shorted together within the same I/O bank.

☞ When VREF pins are used as regular I/Os, they have higher pin capacitance than regular user I/O pins. This has an impact on the timing if the pins are used as inputs and outputs.

👣 For more information about VREF pin capacitance, refer to the pin capacitance section in the *Cyclone IV Device Datasheet* chapter.

👣 For information about how to identify V_{REF} groups, refer to the Cyclone IV **Device Pin-Out** files or the **Quartus II Pin Planner** tool.

Table 6–4 and Table 6–5 summarize the number of VREF pins in each I/O bank for the Cyclone IV device family.

Table 6–4. Number of VREF Pins Per I/O Bank for Cyclone IV E Devices   (Part 1 of 2)

Device	EP4CE6			EP4CE10			EP4CE15						EP4CE22			EP4CE30			EP4CE40				EP4CE55			EP4CE75			EP4CE115	
I/O Bank *(1)*	144-EQPF	256-UBGA	256-FBGA	144-EQPF	256-UBGA	256-FBGA	144-EQPF	164-MBGA	256-MBGA	256-UBGA	256-FBGA	484-FBGA	144-EQPF	256-UBGA	256-FBGA	324-FBGA	484-FBGA	780-FBGA	324-FBGA	484-UBGA	484-FBGA	780-FBGA	484-UBGA	484-FBGA	780-FBGA	484-UBGA	484-FBGA	780-FBGA	484-FBGA	780-FBGA
1	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3
2	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3
3	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3
4	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3
5	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3
6	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3
7	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3

Table 6–4. Number of VREF Pins Per I/O Bank for Cyclone IV E Devices (Part 2 of 2)

Device	EP4CE6			EP4CE10			EP4CE15						EP4CE22			EP4CE30			EP4CE40				EP4CE55			EP4CE75			EP4CE115	
I/O Bank (1)	144-EQPF	256-UBGA	256-FBGA	144-EQPF	256-UBGA	256-FBGA	144-EQPF	164-MBGA	256-MBGA	256-UBGA	256-FBGA	484-FBGA	144-EQPF	256-UBGA	256-FBGA	324-FBGA	484-FBGA	780-FBGA	324-FBGA	484-UBGA	484-FBGA	780-FBGA	484-UBGA	484-FBGA	780-FBGA	484-UBGA	484-FBGA	780-FBGA	484-FBGA	780-FBGA
8	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	4	4	4	4	4	4	4	2	2	2	3	3	3	3	3

Note to Table 6–4:

(1) User I/O pins are used as inputs or outputs; clock input pins are used as inputs only; clock output pins are used as output only.

Table 6–5. Number of VREF Pins Per I/O Bank for Cyclone IV GX Devices

Device	4CGX15	4CGX22		4CGX30			4CGX50		4CGX75		4CGX110			4CGX150		
I/O Bank (1)	169-FBGA	169-FBGA	324-FBGA	169-FBGA	324-FBGA	484-FBGA	484-FBGA	672-FBGA	484-FBGA	672-FBGA	484-FBGA	672-FBGA	896-FBGA	484-FBGA	672-FBGA	896-FBGA
3	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	
4	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	
5	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	
6	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	
7	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	
8 (2)	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	

Notes to Table 6–5:

(1) User I/O pins are used as inputs or outputs; clock input pins are used as inputs only; clock output pins are used as output only.

(2) Bank 9 does not have VREF pin. If input pins with VREF I/O standards are used in bank 9 during user mode, it shares the VREF pin in bank 8.

Each Cyclone IV I/O bank has its own VCCIO pins. Each I/O bank can support only one V_{CCIO} setting from among 1.2, 1.5, 1.8, 2.5, 3.0, or 3.3 V. Any number of supported single-ended or differential standards can be simultaneously supported in a single I/O bank, as long as they use the same V_{CCIO} levels for input and output pins.

When designing LVTTL/LVCMOS inputs with Cyclone IV devices, refer to the following guidelines:

■ All pins accept input voltage (V_I) up to a maximum limit (3.6 V), as stated in the recommended operating conditions provided in the *Cyclone IV Device Datasheet* chapter.

■ Whenever the input level is higher than the bank V_{CCIO}, expect higher leakage current.

■ The LVTTL/LVCMOS I/O standard input pins can only meet the V_{IH} and V_{IL} levels according to bank voltage level.

Voltage-referenced standards are supported in an I/O bank using any number of single-ended or differential standards, as long as they use the same V_{REF} and V_{CCIO} values. For example, if you choose to implement both SSTL-2 and SSTL-18 in your Cyclone IV devices, I/O pins using these standards—because they require different V_{REF} values—must be in different banks from each other. However, the same I/O bank can support SSTL-2 and 2.5-V LVCMOS with the V_{CCIO} set to 2.5 V and the V_{REF} set to 1.25 V.

☞ When using Cyclone IV devices as a receiver in 3.3-, 3.0-, or 2.5-V LVTTL/LVCMOS systems, you are responsible for managing overshoot or undershoot to stay in the absolute maximum ratings and the recommended operating conditions, provided in the *Cyclone IV Device Datasheet* chapter.

☞ The PCI clamping diode is enabled by default in the Quartus II software for input signals with bank V_{CCIO} at 2.5, 3.0, or 3.3 V.

High-Speed Differential Interfaces

Cyclone IV devices can send and receive data through LVDS signals. For the LVDS transmitter and receiver, the input and output pins of Cyclone IV devices support serialization and deserialization through internal logic.

The BLVDS extends the benefits of LVDS to multipoint applications such as bidirectional backplanes. The loading effect and the need to terminate the bus at both ends for multipoint applications require BLVDS to drive out a higher current than LVDS to produce a comparable voltage swing. All the I/O banks of Cyclone IV devices support BLVDS for user I/O pins.

The RSDS and mini-LVDS standards are derivatives of the LVDS standard. The RSDS and mini-LVDS I/O standards are similar in electrical characteristics to LVDS, but have a smaller voltage swing and therefore provide increased power benefits and reduced electromagnetic interference (EMI).

The PPDS standard is the next generation of the RSDS standard introduced by National Semiconductor Corporation. Cyclone IV devices meet the National Semiconductor Corporation PPDS Interface Specification and support the PPDS standard for outputs only. All the I/O banks of Cyclone IV devices support the PPDS standard for output pins only.

The LVDS standard does not require an input reference voltage, but it does require a 100-Ω termination resistor between the two signals at the input buffer. An external resistor network is required on the transmitter side for the top and bottom I/O banks.

External Memory Interfacing

Cyclone IV devices support I/O standards required to interface with a broad range of external memory interfaces, such as DDR SDRAM, DDR2 SDRAM, and QDR II SRAM.

For more information about Cyclone IV devices external memory interface support, refer to the *External Memory Interfaces in Cyclone IV Devices* chapter.

Pad Placement and DC Guidelines

You can use the Quartus II software to validate your pad and pin placement.

Pad Placement

Altera recommends that you create a Quartus II design, enter your device I/O assignments and compile your design to validate your pin placement. The Quartus II software checks your pin connections with respect to the I/O assignment and placement rules to ensure proper device operation. These rules depend on device density, package, I/O assignments, voltage assignments and other factors that are not fully described in this chapter.

For more information about how the Quartus II software checks I/O restrictions, refer to the *I/O Management* chapter in volume 2 of the *Quartus II Handbook*.

DC Guidelines

For the Quartus II software to automatically check for illegally placed pads according to the DC guidelines, set the DC current sink or source value to **Electromigration Current** assignment on each of the output pins that are connected to the external resistive load.

The programmable current strength setting has an impact on the amount of DC current that an output pin can source or sink. Determine if the current strength setting is sufficient for the external resistive load condition on the output pin.

Clock Pins Functionality

Cyclone IV clock pins have multiple purposes, as per listed:

- `CLK` pins—Input support for single-ended and voltage-referenced standards. For I/O standard support, refer to Table 6–3 on page 6–11.

- `DIFFCLK` pins—Input support for differential standards. For I/O standard support, refer to Table 6–3 on page 6–11. When used as `DIFFCLK` pins, DC or AC coupling can be used depending on the interface requirements and external termination is required. For more information, refer to "High-Speed I/O Standards Support" on page 6–28.

- `REFCLK` pins—Input support for high speed differential reference clocks used by the transceivers in Cyclone IV GX devices. For I/O support, coupling, and termination requirements, refer to Table 6–10 on page 6–29.

High-Speed I/O Interface

Cyclone IV E I/Os are separated into eight I/O banks, as shown in Figure 6–9 on page 6–17. Cyclone IV GX I/Os are separated into six user I/O banks with the left side of the device as the transceiver block, as shown in Figure 6–10 on page 6–18. Each bank has an independent power supply. True output drivers for LVDS, RSDS, mini-LVDS, and PPDS are on the right I/O banks. On the Cyclone IV E row I/O banks and the Cyclone IV GX right I/O banks, some of the differential pin pairs (p and n pins) of the true output drivers are not located on adjacent pins. In these cases, a power pin is located between the p and n pins. These I/O standards are also supported on all I/O banks using two single-ended output with the second output programmed as inverted, and an external resistor network. True input buffers for these I/O standards are supported on the top, bottom, and right I/O banks except for I/O bank 9.

Table 6–6 and Table 6–7 summarize which I/O banks support these I/O standards in the Cyclone IV device family.

Table 6–6. Differential I/O Standards Supported in Cyclone IV E I/O Banks

Differential I/O Standards	I/O Bank Location	External Resistor Network at Transmitter	Transmitter (TX)	Receiver (RX)
LVDS	1,2,5,6	Not Required	✓	✓
	All	Three Resistors		
RSDS	1,2,5,6	Not Required	✓	—
	3,4,7,8	Three Resistors		
	All	Single Resistor		
mini-LVDS	1,2,5,6	Not Required	✓	—
	All	Three Resistors		
PPDS	1,2,5,6	Not Required	✓	—
	All	Three Resistors		
BLVDS [1]	All	Single Resistor	✓	✓
LVPECL [2]	All	—	—	✓
Differential SSTL-2 [3]	All	—	✓	✓
Differential SSTL-18 [3]	All	—	✓	✓
Differential HSTL-18 [3]	All	—	✓	✓
Differential HSTL-15 [3]	All	—	✓	✓
Differential HSTL-12 [3], [4]	All	—	✓	✓

Notes to Table 6–6:

(1) Transmitter and Receiver f_{MAX} depend on system topology and performance requirement.

(2) The LVPECL I/O standard is only supported on dedicated clock input pins.

(3) The differential SSTL-2, SSTL-18, HSTL-18, HSTL-15, and HSTL-12 I/O standards are only supported on clock input pins and PLL output clock pins. PLL output clock pins do not support Class II interface type of differential SSTL-18, HSTL-18, HSTL-15, and HSTL-12 I/O standards.

(4) Differential HSTL-12 Class II is supported only in column I/O banks.

Table 6–7. Differential I/O Standards Supported in Cyclone IV GX I/O Banks

Differential I/O Standards	I/O Bank Location	External Resistor Network at Transmitter	Transmitter (TX)	Receiver (RX)
LVDS	5,6	Not Required	✓	✓
	3,4,5,6,7,8	Three Resistors		
RSDS	5,6	Not Required	✓	—
	3,4,7,8	Three Resistors		
	3,4,5,6,7,8	Single Resistor		
mini-LVDS	5,6	Not Required	✓	—
	3,4,5,6,7,8	Three Resistors		
PPDS	5,6	Not Required	✓	—
	3,4,5,6,7,8	Three Resistors		
BLVDS [1]	3,4,5,6,7,8	Single Resistor	✓	✓
LVPECL [2]	3,4,5,6,7,8	—	—	✓
Differential SSTL-2 [3]	3,4,5,6,7,8	—	✓	✓
Differential SSTL-18 [3]	3,4,5,6,7,8	—	✓	✓
Differential HSTL-18 [3]	3,4,5,6,7,8	—	✓	✓
Differential HSTL-15 [3]	3,4,5,6,7,8	—	✓	✓
Differential HSTL-12 [3]	4,5,6,7,8	—	✓	✓

Notes to Table 6–7:

(1) Transmitter and Receiver f_{MAX} depend on system topology and performance requirement.

(2) The LVPECL I/O standard is only supported on dedicated clock input pins.

(3) The differential SSTL-2, SSTL-18, HSTL-18, HSTL-15, and HSTL-12 I/O standards are only supported on clock input pins and PLL output clock pins. PLL output clock pins do not support Class II interface type of differential SSTL-18, HSTL-18, HSTL-15, and HSTL-12 I/O standards.

You can use I/O pins and internal logic to implement a high-speed differential interface in Cyclone IV devices. Cyclone IV devices do not contain dedicated serialization or deserialization circuitry. Therefore, shift registers, internal phase-locked loops (PLLs), and I/O cells are used to perform serial-to-parallel conversions on incoming data and parallel-to-serial conversion on outgoing data. The differential interface data serializers and deserializers (SERDES) are automatically constructed in the core logic elements (LEs) with the Quartus II software ALTLVDS megafunction.

Table 6–8 and Table 6–9 summarize the total number of supported row and column differential channels in the Cyclone IV device family.

Table 6–8. Cyclone IV E I/O and Differential Channel Count

Device	EP4CE6			EP4CE10			EP4CE15						EP4CE22			EP4CE30			EP4CE40				EP4CE55			EP4CE75			EP4CE115	
Numbers of Differential Channels (1), (2)	144-EQPF	256-UBGA	256-FBGA	144-EQPF	256-UBGA	256-FBGA	144-EQPF	164-MBGA	256-MBGA	256-UBGA	256-FBGA	484-FBGA	144-EQPF	256-UBGA	256-FBGA	324-FBGA	484-FBGA	780-FBGA	324-FBGA	484-UBGA	484-FBGA	780-FBGA	484-UBGA	484-FBGA	780-FBGA	484-UBGA	484-FBGA	780-FBGA	484-FBGA	780-FBGA
User I/O (3)	91	179	179	91	179	179	81	89	165	165	165	343	79	153	153	193	328	532	193	328	328	532	324	324	374	292	292	426	280	528
User I/O Banks	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
LVDS (4), (6)	8	23	23	8	23	23	6	8	21	21	21	67	7	20	20	30	60	112	30	60	60	112	62	62	70	54	54	79	50	103
Emulated LVDS (5), (6)	13	43	43	13	43	43	12	13	32	32	32	70	10	32	32	38	64	112	38	64	64	112	70	70	90	56	56	99	53	127

Notes to Table 6–8:

(1) User I/O pins are used as inputs or outputs; clock input pins are used as inputs only; clock output pins are used as output only.

(2) For differential pad placement guidelines, refer to "Pad Placement" on page 6–23.

(3) The I/O pin count includes all GPIOs, dedicated clock pins, and dual-purpose configuration pins. Dedicated configuration pins are not included in the pin count.

(4) The true LVDS count includes all LVDS I/O pairs, differential clock input and clock output pins in row I/O banks 1, 2, 5, and 6.

(5) The emulated LVDS count includes all LVDS I/O pairs, differential clock input and clock output pins in column I/O banks 3, 4, 7, and 8.

(6) LVDS input and output buffers are sharing the same p and n pins. One LVDS I/O channel can only be either transmitter or receiver at a time.

Table 6–9. Cyclone IV GX I/O, Differential, and XCVRs Channel Count

Device	4CGX15	4CGX22		4CGX30			4CGX50		4CGX75		4CGX110			4CGX150		
Numbers of Differential Channels *(1)*, *(2)*	169-FBGA	169-FBGA	324-FBGA	169-FBGA	324-FBGA	484-FBGA	484-FBGA	672-FBGA	484-FBGA	672-FBGA	484-FBGA	672-FBGA	896-FBGA	484-FBGA	672-FBGA	896-FBGA
User I/O *(3)*	72	72	150	72	150	290	290	310	290	310	270	393	475	270	393	475
User I/O banks	9 *(4)*	9 *(4)*	9 *(4)*	9 *(4)*	9 *(4)*	11 *(5)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*	11 *(5)*, *(6)*
LVDS *(7)*, *(9)*	9	9	16	9	16	45	45	51	45	51	38	52	63	38	52	63
Emulated LVDS *(8)*, *(9)*	16	16	48	16	48	85	85	89	85	89	82	129	157	82	129	157
XCVRs	2	2	4	2	4	4	4	8	4	8	4	8	8	4	8	8

Notes to Table 6–9:

(1) User I/O pins are used as inputs or outputs; clock input pins are used as inputs only; clock output pins are used as outputs only.

(2) For differential pad placement guidelines, refer to "Pad Placement" on page 6–23.

(3) The I/O pin count includes all GPIOs, dedicated clock pins, and dual-purpose configuration pins. Transceivers pins and dedicated configuration pins are not included in the pin count.

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

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

(6) Single-ended clock input support is available for dedicated clock input I/O banks 3B (pins CLKIO20 and CLKIO22) and 8B (pins CLKIO17 and CLKIO19).

(7) The true LVDS count includes all LVDS I/O pairs, differential clock input and clock output pins in right I/O banks 5 and 6.

(8) The emulated LVDS count includes all LVDS I/O pairs, differential clock input and clock output pins in column I/O banks 3, 4, 7, and 8.

(9) LVDS input and output buffers are sharing the same p and n pins. One LVDS I/O channel can only be either transmitter or receiver at a time.

High-Speed I/O Standards Support

This section provides information about the high-speed I/O standards and the HSSI input reference clock supported in Cyclone IV devices.

High Speed Serial Interface (HSSI) Input Reference Clock Support

Cyclone IV GX devices support the same I/O features for GPIOs with additional new features where current I/O banks 3A and 8A consist of dual-purpose clock input pins (CLKIN) and 3B and 8B consist of dedicated CLKIN that can be used to support the high-speed transceiver input reference clock (REFCLK) features on top of the general-purpose clock input function.

The EP4CGX15, EP4CGX22, and EP4CGX30 devices contain two pairs of CLKIN/REFCLK pins located in I/O banks 3A and 8A. I/O banks 3B and 8B are not available in EP4CGX15, EP4CGX22, and EP4CGX30 devices. The EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices have a total of four pairs of CLKIN/REFCLK pins located in I/O banks 3A, 3B, 8A, and 8B. I/O banks 3B and 8B can also support single-ended clock inputs. For more information about the CLKIN/REFCLK pin location, refer to Figure 6–10 on page 6–18 and Figure 6–11 on page 6–19.

The `CLKIN/REFCLK` pins are powered by dedicated $V_{CC\_CLKIN3A}$, $V_{CC\_CLKIN3B}$, $V_{CC\_CLKIN8A}$, and $V_{CC\_CLKIN8B}$ power supplies separately in their respective I/O banks to avoid the different power level requirements in the same bank for GPIO.

Table 6–10. Cyclone IV GX HSSI REFCLK I/O Standard Support Using GPIO CLKIN Pins [1], [2]

| I/O Standard | HSSI Protocol | Coupling | Termination | VCC_CLKIN Level | | I/O Pin Type | | |
				Input	Output	Column I/O	Row I/O	Supported I/O Banks
LVDS	All	Differential AC (Need off chip resistor to restore V_{CM})	Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
LVPECL	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
1.2V, 1.5V, 3.3V PCML	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
	All		Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B
HCSL	PCIe	Differential DC	Off chip	2.5V	Not supported	Yes	No	3A, 3B, 8A, 8B

Notes to Table 6–10:

(1) The EP4CGX15, EP4CGX22, and EP4CGX30 devices have two pairs of dedicated clock input pins in banks 3A and 8A for HSSI input reference clock. I/O banks 3B and 8B are not available in EP4CGX15, EP4CGX22, and EP4CGX30 devices.

(2) The EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices have four pairs of dedicated clock input pins in banks 3A, 3B, 8A, and 8B for HSSI input or single-ended clock input.

For more information about the AC-coupled termination scheme for the HSSI reference clock, refer to the *Cyclone IV Transceivers Architecture* chapter.

LVDS I/O Standard Support in Cyclone IV Devices

The LVDS I/O standard is a high-speed, low-voltage swing, low power, and GPIO interface standard. Cyclone IV devices meet the ANSI/TIA/EIA-644 standard with the following exceptions:

- The maximum differential output voltage (V_{OD}) is increased to 600 mV. The maximum V_{OD} for ANSI specification is 450 mV.

- The input voltage range is reduced to the range of 1.0 V to 1.6 V, 0.5 V to 1.85 V, or 0 V to 1.8 V based on different frequency ranges. The ANSI/TIA/EIA-644 specification supports an input voltage range of 0 V to 2.4 V.

For LVDS I/O standard electrical specifications in Cyclone IV devices, refer to the *Cyclone IV Device Datasheet* chapter.

Designing with LVDS

Cyclone IV I/O banks support the LVDS I/O standard. The Cyclone IV GX right I/O banks support true LVDS transmitters while the Cyclone IV E left and right I/O banks support true LVDS transmitters. On the top and bottom I/O banks, the emulated LVDS transmitters are supported using two single-ended output buffers with external resistors. One of the single-ended output buffers is programmed to have opposite polarity. The LVDS receiver requires an external 100-Ω termination resistor between the two signals at the input buffer.

Figure 6–12 shows a point-to-point LVDS interface using Cyclone IV devices true LVDS output and input buffers.

Figure 6–12. Cyclone IV Devices LVDS Interface with True Output Buffer on the Right I/O Banks
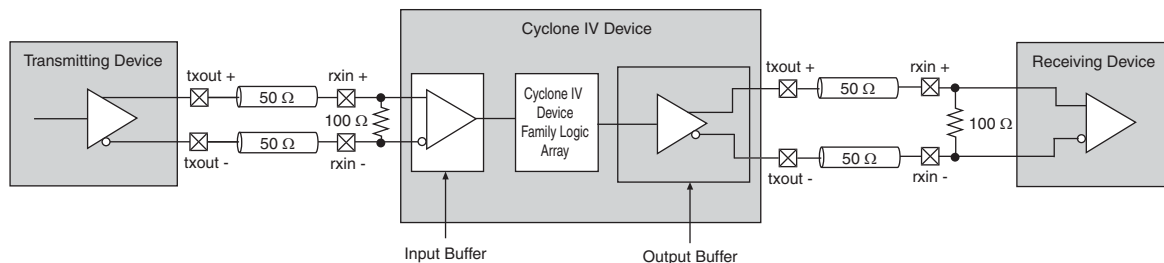
Figure 6–13 shows a point-to-point LVDS interface with Cyclone IV devices LVDS using two single-ended output buffers and external resistors.

Figure 6–13. LVDS Interface with External Resistor Network on the Top and Bottom I/O Banks [1]
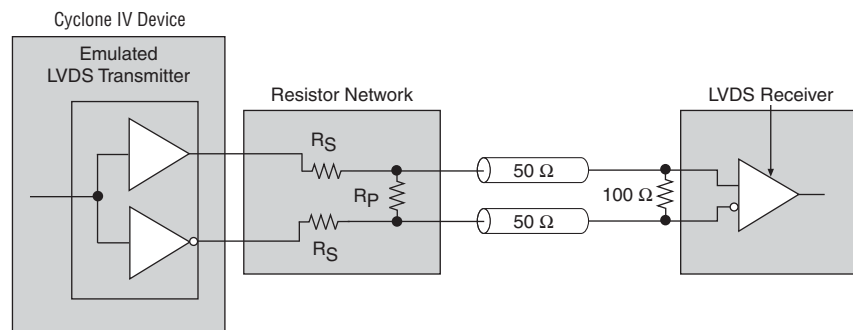
Note to Figure 6–13:

(1) R_S = 120 Ω. R_P = 170 Ω.

BLVDS I/O Standard Support in Cyclone IV Devices

The BLVDS I/O standard is a high-speed differential data transmission technology that extends the benefits of standard point-to-point LVDS to multipoint configuration that supports bidirectional half-duplex communication. BLVDS differs from standard LVDS by providing a higher drive to achieve similar signal swings at the receiver while loaded with two terminations at both ends of the bus.

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

Figure 6–14. BLVDS Topology with Cyclone IV Devices Transmitters and Receivers
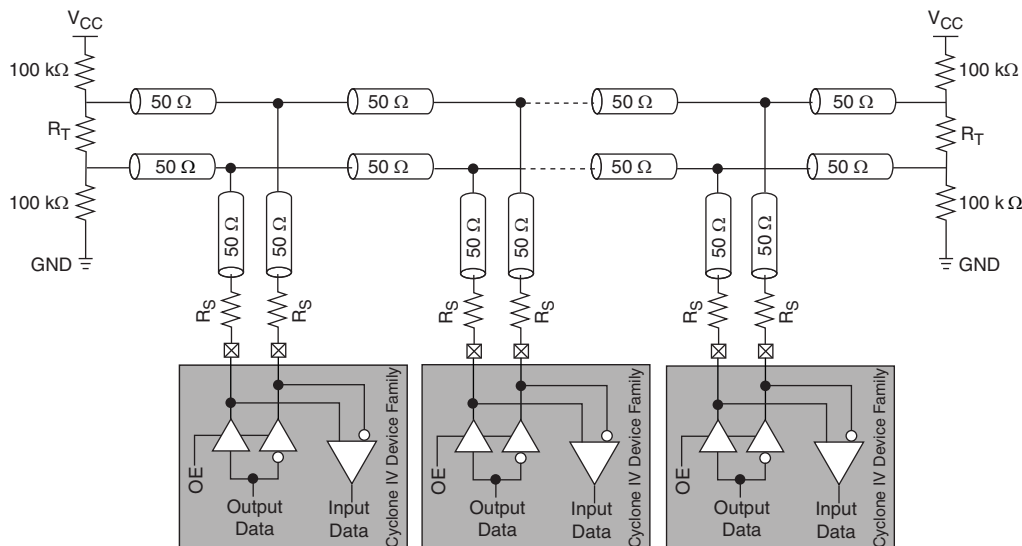
The BLVDS I/O standard is supported on the top, bottom, and right I/O banks of Cyclone IV devices. The BLVDS transmitter uses two single-ended output buffers with the second output buffer programmed as inverted, while the BLVDS receiver uses a true LVDS input buffer. The transmitter and receiver share the same pins. An output-enabled (OE) signal is required to tristate the output buffers when the LVDS input buffer receives a signal.

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

Designing with BLVDS

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

☞ Altera recommends that you perform simulation using the IBIS model while considering factors such as bus loading, termination values, and output and input buffer location on the bus to ensure that the required performance is achieved.

For more information about BLVDS interface support in Altera devices, refer to *AN 522: Implementing Bus LVDS Interface in Supported Altera Device Families*.

RSDS, Mini-LVDS, and PPDS I/O Standard Support in Cyclone IV Devices

The RSDS, mini-LVDS, and PPDS I/O standards are used in chip-to-chip applications between the timing controller and the column drivers on the display panels such as LCD monitor panels and LCD televisions. Cyclone IV devices meet the National Semiconductor Corporation RSDS Interface Specification, Texas Instruments mini-LVDS Interface Specification, and National Semiconductor Corporation PPDS Interface Specification to support RSDS, mini-LVDS and PPDS output standards, respectively.

For Cyclone IV devices RSDS, mini-LVDS, and PPDS output electrical specifications, refer to the *Cyclone IV Device Datasheet* chapter.

For more information about the RSDS I/O standard, refer to the RSDS specification from the National Semiconductor website (www.national.com).

Designing with RSDS, Mini-LVDS, and PPDS

Cyclone IV I/O banks support RSDS, mini-LVDS, and PPDS output standards. The right I/O banks support true RSDS, mini-LVDS, and PPDS transmitters. On the top and bottom I/O banks, RSDS, mini-LVDS, and PPDS transmitters are supported using two single-ended output buffers with external resistors. The two single-ended output buffers are programmed to have opposite polarity.

Figure 6–15 shows an RSDS, mini-LVDS, or PPDS interface with a true output buffer.

Figure 6–15. Cyclone IV Devices RSDS, Mini-LVDS, or PPDS Interface with True Output Buffer on the Right I/O Banks
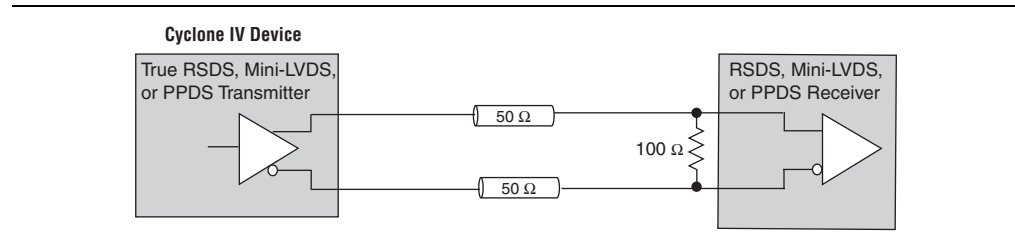
Figure 6–16 shows an RSDS, mini-LVDS, or PPDS interface with two single-ended output buffers and external resistors.

Figure 6–16. RSDS, Mini-LVDS, or PPDS Interface with External Resistor Network on the Top and Bottom I/O Banks [1]
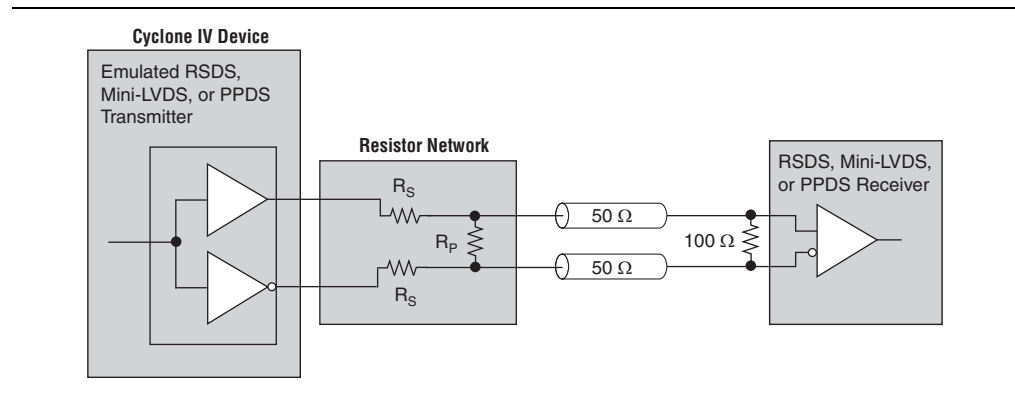
Figure 6–16. RSDS, Mini-LVDS, or PPDS Interface with External Resistor Network on the Top and Bottom I/O Banks [1]

Note to Figure 6–16:

(1) R_S and R_P values are pending characterization.

A resistor network is required to attenuate the output voltage swing to meet RSDS, mini-LVDS, and PPDS specifications when using emulated transmitters. You can modify the resistor network values to reduce power or improve the noise margin.

The resistor values chosen must satisfy Equation 6–1.

Equation 6–1. Resistor Network

$$\frac{R_S \times \dfrac{R_P}{2}}{R_S + \dfrac{R_P}{2}} = 50 \ \Omega$$

☞   Altera recommends that you perform simulations using Cyclone IV devices IBIS models to validate that custom resistor values meet the RSDS, mini-LVDS, or PPDS requirements.

It is possible to use a single external resistor instead of using three resistors in the resistor network for an RSDS interface, as shown in Figure 6–17. The external single-resistor solution reduces the external resistor count while still achieving the required signaling level for RSDS. However, the performance of the single-resistor solution is lower than the performance with the three-resistor network.

Figure 6–17 shows the RSDS interface with a single resistor network on the top and bottom I/O banks.

Figure 6–17. RSDS Interface with Single Resistor Network on the Top and Bottom I/O Banks [1]
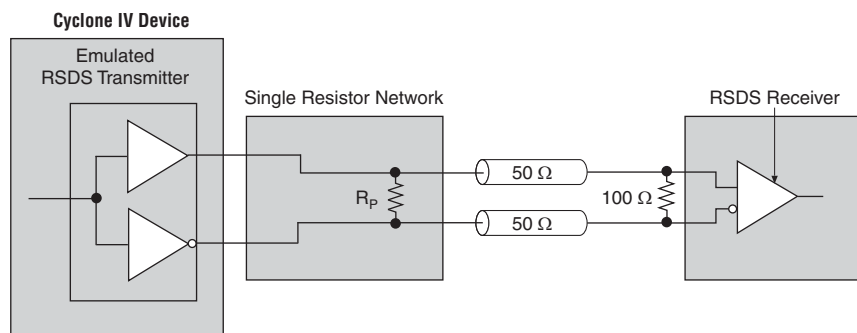
Note to Figure 6–17:

(1) R_P value is pending characterization.

LVPECL I/O Support in Cyclone IV Devices

The LVPECL I/O standard is a differential interface standard that requires a 2.5-V V_{CCIO}. This standard is used in applications involving video graphics, telecommunications, data communications, and clock distribution. Cyclone IV devices support the LVPECL input standard at the dedicated clock input pins only. The LVPECL receiver requires an external 100-Ω termination resistor between the two signals at the input buffer.

For the LVPECL I/O standard electrical specification, refer to the *Cyclone IV Device Datasheet* chapter.

AC coupling is required when the LVPECL common mode voltage of the output buffer is higher than the Cyclone IV devices LVPECL input common mode voltage.

Figure 6–18 shows the AC-coupled termination scheme. The 50-Ω resistors used at the receiver are external to the device. DC-coupled LVPECL is supported if the LVPECL output common mode voltage is in the Cyclone IV devices LVPECL input buffer specification (refer to Figure 6–19).

Figure 6–18. LVPECL AC-Coupled Termination [1]
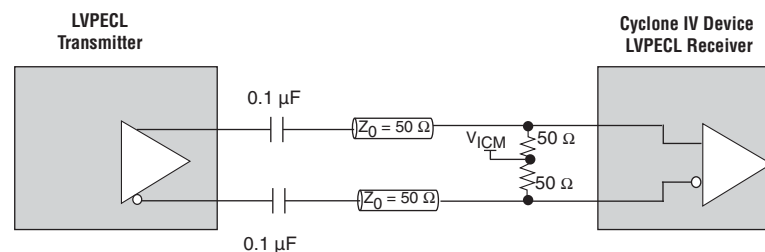
Note to Figure 6–18:

(1) The LVPECL AC-coupled termination is applicable only when an Altera FPGA transmitter is used.

Figure 6–19 shows the LVPECL DC-coupled termination.

Figure 6–19. LVPECL DC-Coupled Termination [1]
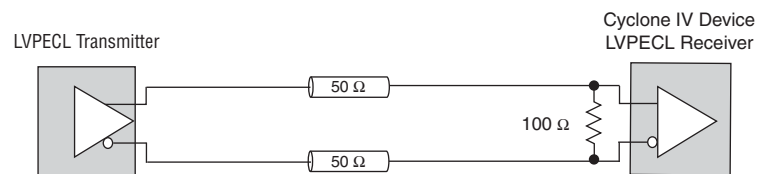
Note to Figure 6–19:

(1) The LVPECL DC-coupled termination is applicable only when an Altera FPGA transmitter is used.

Differential SSTL I/O Standard Support in Cyclone IV Devices

The differential SSTL I/O standard is a memory-bus standard used for applications such as high-speed DDR SDRAM interfaces. Cyclone IV devices support differential SSTL-2 and SSTL-18 I/O standards. The differential SSTL output standard is only supported at PLL#_CLKOUT pins using two single-ended SSTL output buffers (PLL#_CLKOUTp and PLL#_CLKOUTn), with the second output programmed to have opposite polarity. The differential SSTL input standard is supported on the GCLK pins only, treating differential inputs as two single-ended SSTL and only decoding one of them.

The differential SSTL I/O standard requires two differential inputs with an external reference voltage (VREF) as well as an external termination voltage (VTT) of $0.5 \times V_{CCIO}$ to which termination resistors are connected.

For differential SSTL electrical specifications, refer to "Differential I/O Standard Termination" on page 6–15 and the *Cyclone IV Device Datasheet* chapter.

Figure 6–8 on page 6–15 shows the differential SSTL Class I and Class II interface.

Differential HSTL I/O Standard Support in Cyclone IV Devices

The differential HSTL I/O standard is used for the applications designed to operate in 0 V to 1.2 V, 0 V to 1.5 V, or 0 V to 1.8 V HSTL logic switching range. Cyclone IV devices support differential HSTL-18, HSTL-15, and HSTL-12 I/O standards. The differential HSTL input standard is available on GCLK pins only, treating the differential inputs as two single-ended HSTL and only decoding one of them. The differential HSTL output standard is only supported at the PLL#_CLKOUT pins using two single-ended HSTL output buffers (PLL#_CLKOUTp and PLL#_CLKOUTn), with the second output programmed to have opposite polarity.

The differential HSTL I/O standard requires two differential inputs with an external reference voltage (VREF), as well as an external termination voltage (VTT) of $0.5 \times V_{CCIO}$ to which termination resistors are connected.

For differential HSTL signaling characteristics, refer to "Differential I/O Standard Termination" on page 6–15 and the *Cyclone IV Device Datasheet* chapter.

Figure 6–7 on page 6–15 shows the differential HSTL Class I and Class II interface.

True Differential Output Buffer Feature

Cyclone IV devices true differential transmitters offer programmable pre-emphasis—you can turn it on or off. The default setting is on.

Programmable Pre-Emphasis

The programmable pre-emphasis boosts the high frequencies of the output signal to compensate the frequency-dependant attenuation of the transmission line to maximize the data eye opening at the far-end receiver. Without pre-emphasis, the output current is limited by the V_{OD} specification and the output impedance of the transmitter. At high frequency, the slew rate may not be fast enough to reach full V_{OD}

before the next edge; this may lead to pattern-dependent jitter. With pre-emphasis, the output current is momentarily boosted during switching to increase the output slew rate. The overshoot produced by this extra switching current is different from the overshoot caused by signal reflection. This overshoot happens only during switching, and does not produce ringing.

The Quartus II software allows two settings for programmable pre-emphasis control—**0** and **1**, in which **0** is pre-emphasis off and **1** is pre-emphasis on. The default setting is **1**. The amount of pre-emphasis needed depends on the amplification of the high-frequency components along the transmission line. You must adjust the setting to suit your designs, as pre-emphasis decreases the amplitude of the low-frequency component of the output signal.

Figure 6–20 shows the differential output signal with pre-emphasis.

Figure 6–20. The Output Signal with Pre-Emphasis
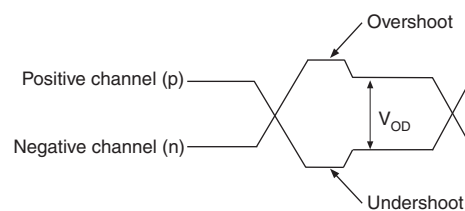
High-Speed I/O Timing

This section discusses the timing budget, waveforms, and specifications for source-synchronous signaling in Cyclone IV devices. Timing for source-synchronous signaling is based on skew between the data and clock signals.

High-speed differential data transmission requires timing parameters provided by IC vendors and requires you to consider the board skew, cable skew, and clock jitter. This section provides information about high-speed I/O standards timing parameters in Cyclone IV devices.

Table 6–11 defines the parameters of the timing diagram shown in Figure 6–21.

Table 6–11. High-Speed I/O Timing Definitions   (Part 1 of 2)

Parameter	Symbol	Description
Transmitter channel-to-channel skew [1]	TCCS	The timing difference between the fastest and slowest output edges, including t_{CO} variation and clock skew. The clock is included in the TCCS measurement.
Sampling window	SW	The period of time during which the data must be valid in order for you to capture it correctly. The setup and hold times determine the ideal strobe position in the sampling window. $T_{SW} = T_{SU} + T_{hd} + $ PLL jitter.
Time unit interval	TUI	The TUI is the data-bit timing budget allowed for skew, propagation delays, and data sampling window.
Receiver input skew margin	RSKM	RSKM is defined by the total margin left after accounting for the sampling window and TCCS. The RSKM equation is: $RSKM = \dfrac{(TUI - SW - TCCS)}{2}$

Table 6–11. High-Speed I/O Timing Definitions (Part 2 of 2)

Parameter	Symbol	Description
Input jitter tolerance (peak-to-peak)	—	Allowed input jitter on the input clock to the PLL that is tolerable while maintaining PLL lock.
Output jitter (peak-to-peak)	—	Peak-to-peak output jitter from the PLL.

Note to Table 6–11:

(1) The TCCS specification applies to the entire bank of differential I/O as long as the SERDES logic is placed in the logic array block (LAB) adjacent to the output pins.
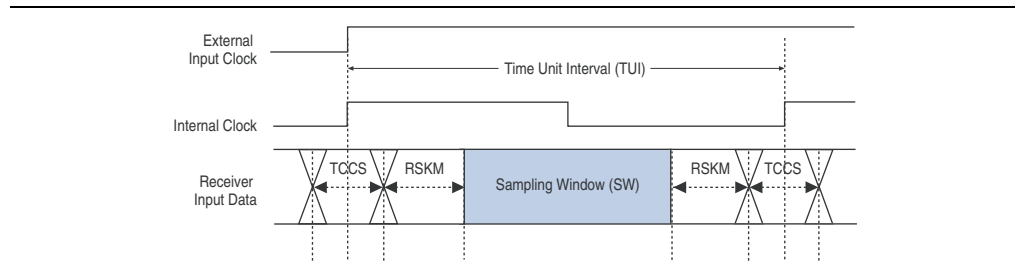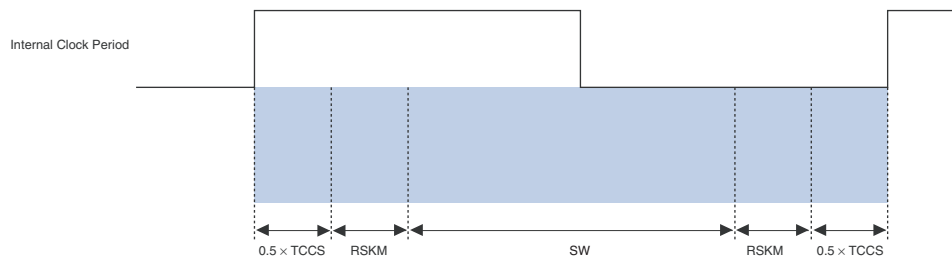
Figure 6–21. High-Speed I/O Timing Diagram

Figure 6–22 shows the Cyclone IV devices high-speed I/O timing budget.

Figure 6–22. Cyclone IV Devices High-Speed I/O Timing Budget   [1]

Note to Figure 6–22:

(1) The equation for the high-speed I/O timing budget is:
$$\text{period} = 0.5 \times \text{TCCS} + \text{RSKM} + \text{SW} + \text{RSKM} + 0.5 \times \text{TCCS}$$

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

Design Guidelines

This section provides guidelines for designing with Cyclone IV devices.

Differential Pad Placement Guidelines

To maintain an acceptable noise level on the V_{CCIO} supply, you must observe some restrictions on the placement of single-ended I/O pins in relation to differential pads.

☞ For guidelines on placing single-ended pads with respect to differential pads in Cyclone IV devices, refer to "Pad Placement and DC Guidelines" on page 6–23.

Board Design Considerations

This section explains how to achieve the optimal performance from a Cyclone IV I/O interface and ensure first-time success in implementing a functional design with optimal signal quality. You must consider the critical issues of controlled impedance of traces and connectors, differential routing, and termination techniques to get the best performance from Cyclone IV devices.

Use the following general guidelines to improve signal quality:

■ Base board designs on controlled differential impedance. Calculate and compare all parameters, such as trace width, trace thickness, and the distance between two differential traces.

■ Maintain equal distance between traces in differential I/O standard pairs as much as possible. Routing the pair of traces close to each other maximizes the common-mode rejection ratio (CMRR).

■ Longer traces have more inductance and capacitance. These traces must be as short as possible to limit signal integrity issues.

■ Place termination resistors as close to receiver input pins as possible.

■ Use surface mount components.

■ Avoid 90° corners on board traces.

■ Use high-performance connectors.

■ Design backplane and card traces so that trace impedance matches the impedance of the connector and termination.

■ Keep an equal number of vias for both signal traces.

■ Create equal trace lengths to avoid skew between signals. Unequal trace lengths result in misplaced crossing points and decrease system margins as the TCCS value increases.

■ Limit vias because they cause discontinuities.

■ Keep switching transistor-to-transistor logic (TTL) signals away from differential signals to avoid possible noise coupling.

■ Do not route TTL clock signals to areas under or above the differential signals.

■ Analyze system-level signals.

For PCB layout guidelines, refer to *AN 224: High-Speed Board Layout Guidelines* and *AN 315: Guidelines for Designing High-Speed FPGA PCBs*.

Software Overview

Cyclone IV devices high-speed I/O system interfaces are created in core logic by a Quartus II software megafunction because they do not have a dedicated circuit for the SERDES. Cyclone IV devices use the I/O registers and LE registers to improve the timing performance and support the SERDES. The Quartus II software allows you to design your high-speed interfaces using ALTLVDS megafunction. This megafunction

implements either a high-speed deserializer receiver or a high-speed serializer transmitter. There is a list of parameters in the ALTLVDS megafunction that you can set to customize your SERDES based on your design requirements. The megafunction is optimized to use Cyclone IV devices resources to create high-speed I/O interfaces in the most effective manner.

☞ When you use Cyclone IV devices with the ALTLVDS megafunction, the interface always sends the MSB of your parallel data first.

👣 For more details about designing your high-speed I/O systems interfaces using the ALTLVDS megafunction, refer to the *ALTLVDS Megafunction User Guide* and the *Quartus II Handbook*.

Document Revision History

Table 6–12 lists the revision history for this chapter.

Table 6–12. Document Revision History **(Part 1 of 2)**

Date	Version	Changes
March 2016	2.7	■ Updated Table 6–5 and Table 6–9 to remove support for the N148 package.
May 2013	2.6	■ Updated Table 6–2 by adding Note (9). ■ Updated Table 6–4 and Table 6–8 to add new device options and packages.
February 2013	2.5	Updated Table 6–4 and Table 6–8 to add new device options and packages.
October 2012	2.4	■ Updated "I/O Banks" and "High Speed Serial Interface (HSSI) Input Reference Clock Support " sections. ■ Updated Table 6–3 and Table 6–5. ■ Updated Figure 6–10.
November 2011	2.3	■ Updated "Differential SSTL I/O Standard Support in Cyclone IV Devices" and "Differential HSTL I/O Standard Support in Cyclone IV Devices" sections. ■ Updated Table 6–1, Table 6–8, and Table 6–9. ■ Updated Figure 6–1.
December 2010	2.2	■ Updated for the Quartus II software version 10.1 release. ■ Added Cyclone IV E new device package information. ■ Added "Clock Pins Functionality" section. ■ Updated Table 6–4 and Table 6–8. ■ Minor text edits.
July 2010	2.1	■ Updated "Cyclone IV I/O Elements", "Programmable Pull-Up Resistor", "I/O Banks", "High-Speed I/O Interface", and "Designing with BLVDS" sections. ■ Updated Table 6–6 and Table 6–7. ■ Updated Figure 6–19.

Table 6–12. Document Revision History   (Part 2 of 2)

Date	Version	Changes
February 2010	2.0	■ Added Cyclone IV E devices information for the Quartus II software version 9.1 SP1 release. ■ Updated Table 6–2, Table 6–3, and Table 6–10. ■ Updated "I/O Banks" section. ■ Added Figure 6–9. ■ Updated Figure 6–10 and Figure 6–11. ■ Added Table 6–4, Table 6–6, and Table 6–8.
November 2009	1.0	Initial release.

This chapter describes the memory interface pin support and the external memory interface features of Cyclone® IV devices.

In addition to an abundant supply of on-chip memory, Cyclone IV devices can easily interface with a broad range of external memory devices, including DDR2 SDRAM, DDR SDRAM, and QDR II SRAM. External memory devices are an important system component of a wide range of image processing, storage, communications, and general embedded applications.

☞ Altera recommends that you construct all DDR2 or DDR SDRAM external memory interfaces using the Altera® ALTMEMPHY megafunction. You can implement the controller function using the Altera DDR2 or DDR SDRAM memory controllers, third-party controllers, or a custom controller for unique application needs. Cyclone IV devices support QDR II interfaces electrically, but Altera does not supply controller or physical layer (PHY) megafunctions for QDR II interfaces.

This chapter includes the following sections:

- "Cyclone IV Devices Memory Interfaces Pin Support" on page 7–2
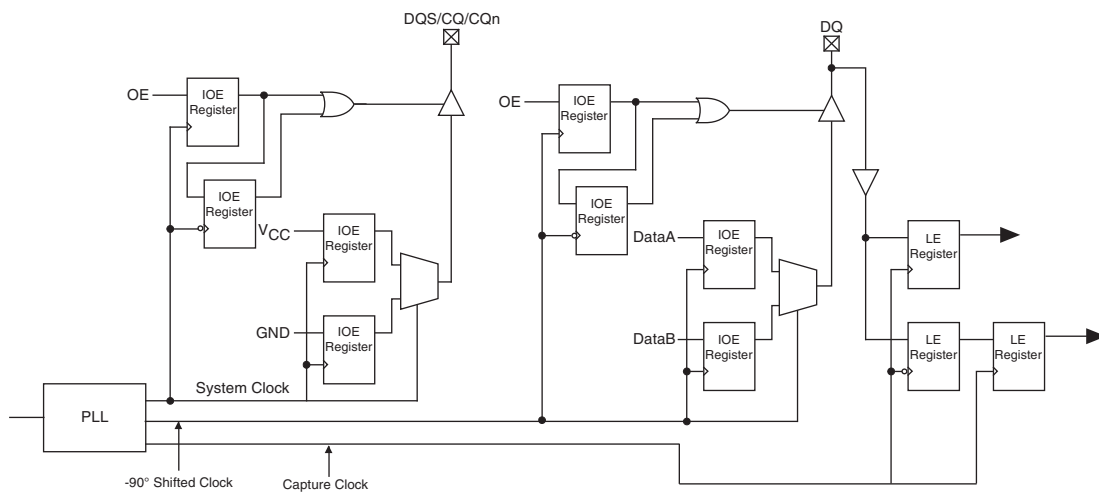- "Cyclone IV Devices Memory Interfaces Features" on page 7–12

For more information about supported maximum clock rate, device and pin planning, IP implementation, and device termination, refer to the *External Memory Interface Handbook*.

Feedback Subscribe

Figure 7–1 shows the block diagram of a typical external memory interface data path in Cyclone IV devices.

Figure 7–1. Cyclone IV Devices External Memory Data Path [1]

Note to Figure 7–1:

(1) All clocks shown here are global clocks.

> For more information about implementing complete external memory interfaces, refer to the *External Memory Interface Handbook*.

Cyclone IV Devices Memory Interfaces Pin Support

Cyclone IV devices use data (DQ), data strobe (DQS), clock, command, and address pins to interface with external memory. Some memory interfaces use the data mask (DM) or byte write select (BWS#) pins to enable data masking. This section describes how Cyclone IV devices support all these different pins.

> For more information about pin utilization, refer to *Volume 2: Device, Pin, and Board Layout Guidelines* of the *External Memory Interface Handbook.*

Data and Data Clock/Strobe Pins

Cyclone IV data pins for external memory interfaces are called D for write data, Q for read data, or DQ for shared read and write data pins. The read-data strobes or read clocks are called DQS pins. Cyclone IV devices support both bidirectional data strobes and unidirectional read clocks. Depending on the external memory standard, the DQ and DQS are bidirectional signals (in DDR2 and DDR SDRAM) or unidirectional signals (in QDR II SRAM). Connect the bidirectional DQ data signals to the same Cyclone IV devices DQ pins. For unidirectional D or Q signals, connect the read-data signals to a group of DQ pins and the write-data signals to a different group of DQ pins.

☞ In QDR II SRAM, the Q read-data group must be placed at a different V_{REF} bank location from the D write-data group, command, or address pins.

In Cyclone IV devices, DQS is used only during write mode in DDR2 and DDR SDRAM interfaces. Cyclone IV devices ignore DQS as the read-data strobe because the PHY internally generates the read capture clock for read mode. However, you must connect the DQS pin to the DQS signal in DDR2 and DDR SDRAM interfaces, or to the CQ signal in QDR II SRAM interfaces.

☞ Cyclone IV devices do not support differential strobe pins, which is an optional feature in the DDR2 SDRAM device.

👣 When you use the Altera Memory Controller MegaCore® function, the PHY is instantiated for you. For more information about the memory interface data path, refer to the *External Memory Interface Handbook*.

☞ ALTMEMPHY is a self-calibrating megafunction, enhanced to simplify the implementation of the read-data path in different memory interfaces. The auto-calibration feature of ALTMEMPHY provides ease-of-use by optimizing clock phases and frequencies across process, voltage, and temperature (PVT) variations. You can save on the global clock resources in Cyclone IV devices through the ALTMEMPHY megafunction because you are not required to route the DQS signals on the global clock buses (because DQS is ignored for read capture). Resynchronization issues do not arise because no transfer occurs from the memory domain clock (DQS) to the system domain for capturing data DQ.

All I/O banks in Cyclone IV devices can support DQ and DQS signals with DQ-bus modes of ×8, ×9, ×16, ×18, ×32, and ×36 except Cyclone IV GX devices that do not support left I/O bank interface. DDR2 and DDR SDRAM interfaces use ×8 mode DQS group regardless of the interface width. For a wider interface, you can use multiple ×8 DQ groups to achieve the desired width requirement.

In the ×9, ×18, and ×36 modes, a pair of complementary DQS pins (CQ and CQ#) drives up to 9, 18, or 36 DQ pins, respectively, in the group, to support one, two, or four parity bits and the corresponding data bits. The ×9, ×18, and ×36 modes support the QDR II memory interface. CQ# is the inverted read-clock signal that is connected to the complementary data strobe (DQS or CQ#) pin. You can use any unused DQ pins as regular user I/O pins if they are not used as memory interface signals.

👣 For more information about unsupported DQS and DQ groups of the Cyclone IV transceivers that run at ≥2.97 Gbps data rate, refer to the *Cyclone IV Device Family Pin Connection Guidelines*.

Table 7–1 lists the number of DQS or DQ groups supported on each side of the Cyclone IV GX device.

Table 7–1. Cyclone IV GX Device DQS and DQ Bus Mode Support for Each Side of the Device

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP4CGX15	169-pin FBGA	Right	1	0	0	0	—	—
		Top [1]	1	0	0	0	—	—
		Bottom [2]	1	0	0	0	—	—
EP4CGX22 EP4CGX30	169-pin FBGA	Right	1	0	0	0	—	—
		Top [1]	1	0	0	0	—	—
		Bottom [2]	1	0	0	0	—	—
	324-pin FBGA	Right	2	2	1	1	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	484-pin FBGA [3]	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
EP4CGX50 EP4CGX75	484-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	672-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
EP4CGX110 EP4CGX150	484-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	672-pin FBGA	Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	896-pin FBGA	Right	6	3	2	2	1	1
		Top	6	3	3	3	1	1
		Bottom	6	3	3	3	1	1

Notes to Table 7–1:

(1) Some of the DQ pins can be used as RUP and RDN pins. You cannot use these groups if you are using these pins as RUP and RDN pins for OCT calibration.

(2) Some of the DQ pins can be used as RUP pins while the DM pins can be used as RDN pins. You cannot use these groups if you are using the RUP and RDN pins for OCT calibration.

(3) Only available for EP4CGX30 device.

Table 7–2 lists the number of DQS or DQ groups supported on each side of the Cyclone IV E device.

Table 7–2. Cyclone IV E Device DQS and DQ Bus Mode Support for Each Side of the Device   (Part 1 of 3)

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP4CE6 EP4CE10	144-pin EQFP	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Bottom [1], [3]	1	0	0	0	—	—
		Top [1], [4]	1	0	0	0	—	—
	256-pin UBGA	Left [1]	1	1	0	0	—	—
		Right [2]	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
	256-pin FBGA	Left [1]	1	1	0	0	—	—
		Right [2]	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
EP4CE15	144-pin EQFP	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Bottom [1], [3]	1	0	0	0	—	—
		Top [1], [4]	1	0	0	0	—	—
	164-pin MBGA	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Bottom [1], [3]	1	0	0	0	—	—
		Top [1], [4]	1	0	0	0	—	—
	256-pin MBGA	Left	1	1	0	0	—	—
		Right	1	1	0	0	—	—
		Bottom [1], [3]	2	2	1	1	—	—
		Top [1], [4]	2	2	1	1	—	—
	256-pin UBGA	Left [1]	1	1	0	0	—	—
		Right [2]	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
	256-pin FBGA	Left [1]	1	1	0	0	—	—
		Right [2]	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
	484-pin FBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	4	4	2	2	1	1
		Top	4	4	2	2	1	1

Table 7–2. Cyclone IV E Device DQS and DQ Bus Mode Support for Each Side of the Device   (Part 2 of 3)

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP4CE22	144-pin EQFP	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Bottom [1], [3]	1	0	0	0	—	—
		Top [1], [4]	1	0	0	0	—	—
	256-pin UBGA	Left [1]	1	1	0	0	—	—
		Right [2]	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
	256-pin FBGA	Left [1]	1	1	0	0	—	—
		Right [2]	1	1	0	0	—	—
		Bottom	2	2	1	1	—	—
		Top	2	2	1	1	—	—
EP4CE30	324-pin FBGA	Left [1]	2	2	1	1	0	0
		Right [2]	2	2	1	1	0	0
		Bottom	2	2	1	1	0	0
		Top	2	2	1	1	0	0
EP4CE30 EP4CE115	484-pin FBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	4	4	2	2	1	1
		Top	4	4	2	2	1	1
	780-pin FBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	6	6	2	2	1	1
		Top	6	6	2	2	1	1
EP4CE40	324-pin FBGA	Left	2	2	1	1	0	0
		Right	2	2	1	1	0	0
		Bottom	2	2	1	1	0	0
		Top	2	2	1	1	0	0

Table 7–2. Cyclone IV E Device DQS and DQ Bus Mode Support for Each Side of the Device (Part 3 of 3)

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP4CE40 EP4CE55 EP4CE75	484-pin UBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	4	4	2	2	1	1
		Top	4	4	2	2	1	1
	484-pin FBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	4	4	2	2	1	1
		Top	4	4	2	2	1	1
	780-pin FBGA	Left	4	4	2	2	1	1
		Right	4	4	2	2	1	1
		Bottom	6	6	2	2	1	1
		Top	6	6	2	2	1	1

Notes to Table 7–2:

(1) Some of the DQ pins can be used as RUP and RDN pins. You cannot use these groups if you are using these pins as RUP and RDN pins for OCT calibration.

(2) Some of the DQ pins can be used as RUP pins while the DM pins can be used as RDN pins. You cannot use these groups if you are using the RUP and RDN pins for OCT calibration.

(3) There is no DM pin support for these groups.

(4) PLLCLKOUT3n and PLLCLKOUT3p pins are shared with the DQ or DM pins to gain ×8 DQ group. You cannot use these groups if you are using PLLCLKOUT3n and PLLCLKOUT3p.

👣 For more information about device package outline, refer to the Device Packaging Specifications webpage.

DQS pins are listed in the Cyclone IV pin tables as DQSXY, in which X indicates the DQS grouping number and Y indicates whether the group is located on the top (T), bottom (B), or right (R) side of the device. Similarly, the corresponding DQ pins are marked as DQXY, in which the X denotes the DQ grouping number and Y denotes whether the group is located on the top (T), bottom (B), or right (R) side of the device. For example, DQS2T indicates a DQS pin belonging to group 2, located on the top side of the device. Similarly, the DQ pins belonging to that group is shown as DQ2T.

☞ Each DQ group is associated with its corresponding DQS pins, as defined in the Cyclone IV pin tables. For example:

■ For DDR2 or DDR SDRAM, ×8 DQ group DQ3B[7..0] pins are associated with the DQS3B pin (same 3B group index)

■ For QDR II SRAM, ×9 Q read-data group DQ3T[8..0] pins are associated with DQS0T/CQ0T and DQS1T/CQ0T# pins (same 0T group index)

The Quartus® II software issues an error message if a DQ group is not placed properly with its associated DQS.
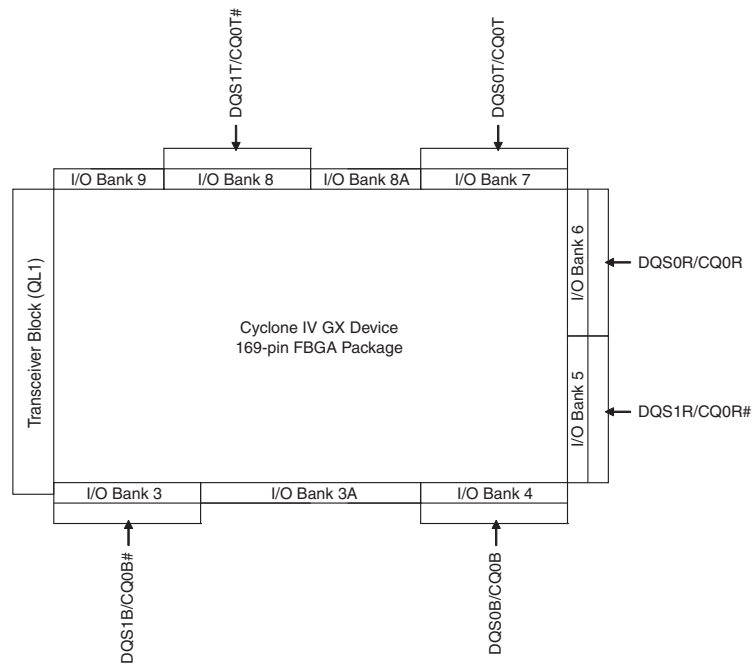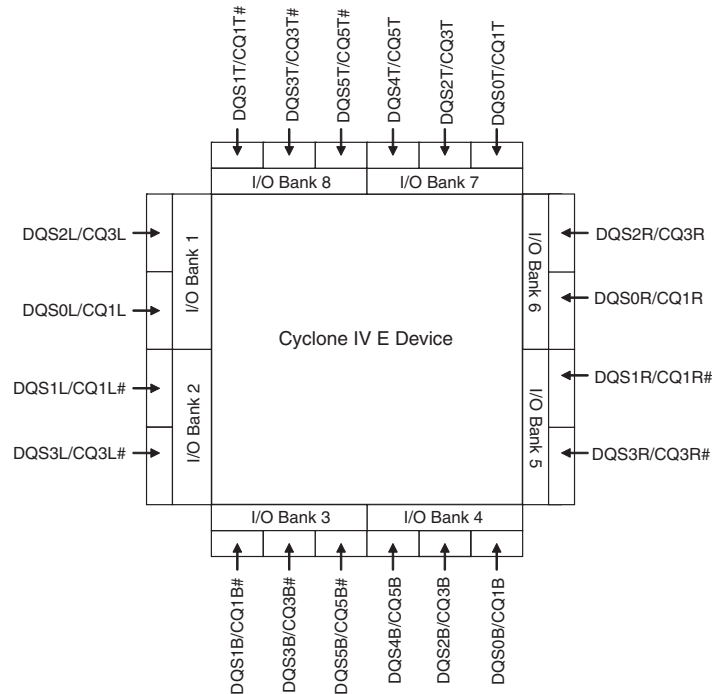
Figure 7–2 shows the location and numbering of the DQS, DQ, or CQ# pins in the Cyclone IV GX I/O banks.

**Figure 7–2. DQS, CQ, or CQ# Pins in Cyclone IV GX I/O Banks** [1]
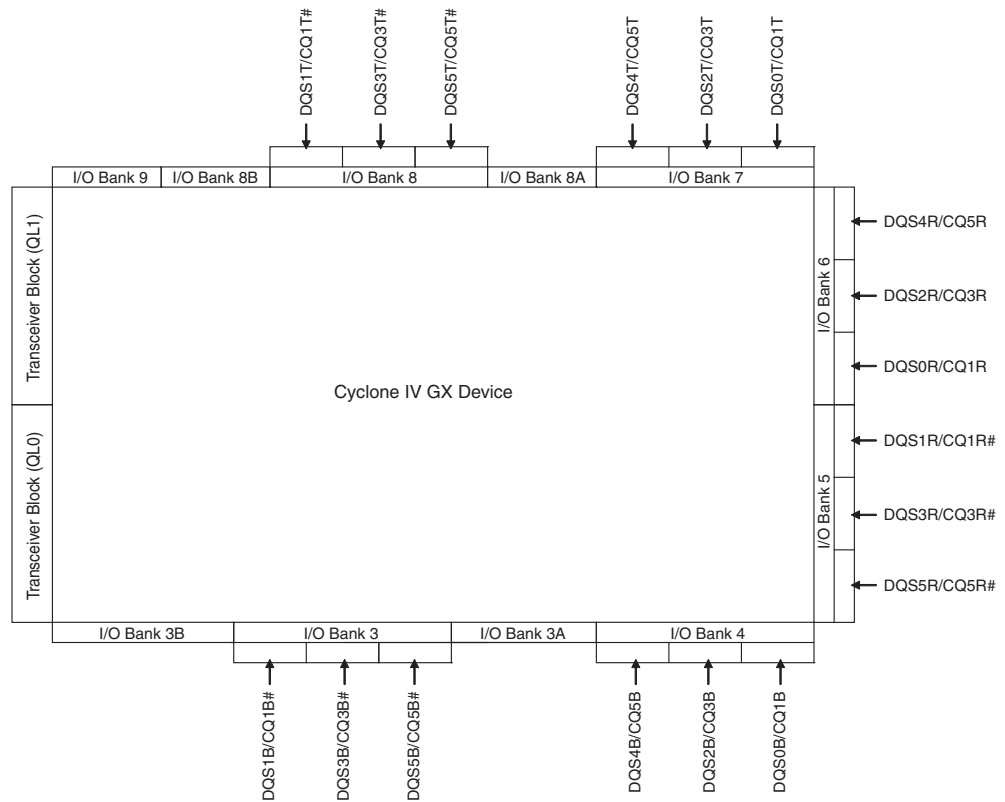
Note to Figure 7–2:

(1) The DQS, CQ, or CQ# pin locations in this diagram apply to all packages in Cyclone IV GX devices except devices in 169-pin FBGA and 324-pin FBGA.

Figure 7–3 shows the location and numbering of the DQS, DQ, or CQ# pins in I/O banks of the Cyclone IV GX device in the 324-pin FBGA package only.

Figure 7–3. DQS, CQ, or CQ# Pins for Cyclone IV GX Devices in the 324-Pin FBGA Package
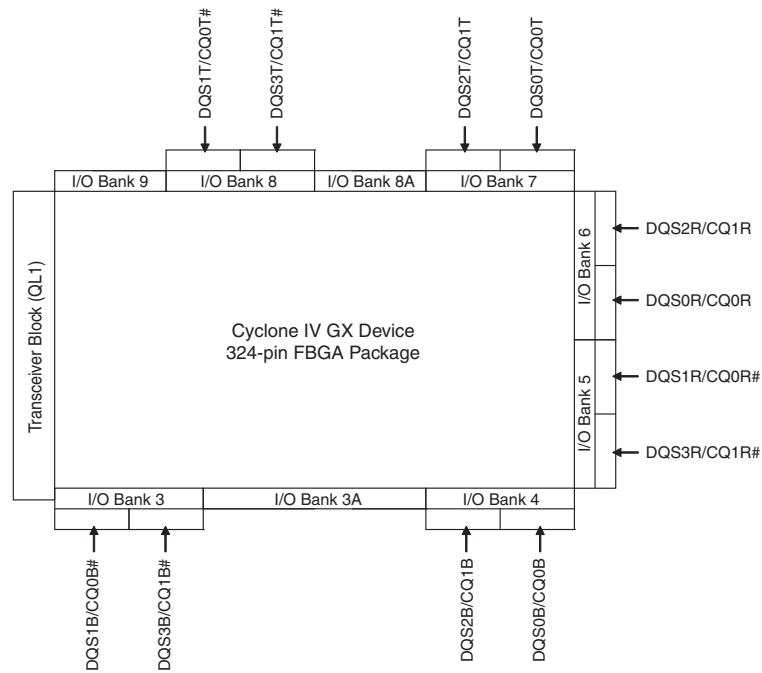
Figure 7–4 shows the location and numbering of the DQS, DQ, or CQ# pins in I/O banks of the Cyclone IV GX device in the 169-pin FBGA package.

Figure 7–4. DQS, CQ, or CQ# Pins for Cyclone IV GX Devices in the 169-Pin FBGA Package

Figure 7–5 shows the location and numbering of the DQS, DQ, or CQ# pins in the Cyclone IV E device I/O banks.

Figure 7–5. DQS, CQ, or CQ# Pins in Cyclone IV E I/O Banks [(1)]

Note to Figure 7–5:

(1) The DQS, CQ, or CQ# pin locations in this diagram apply to all packages in Cyclone IV E devices except devices in 144-pin EQFP.

Figure 7–6 shows the location and numbering of the DQS, DQ, or CQ# pins in I/O banks of the Cyclone IV E device in the 144-pin EQFP and 164-pin MBGA packages.

Figure 7–6. DQS, CQ, or CQ# Pins for Cyclone IV E Devices in the 144-Pin EQFP and 164-pin MBGA Packages
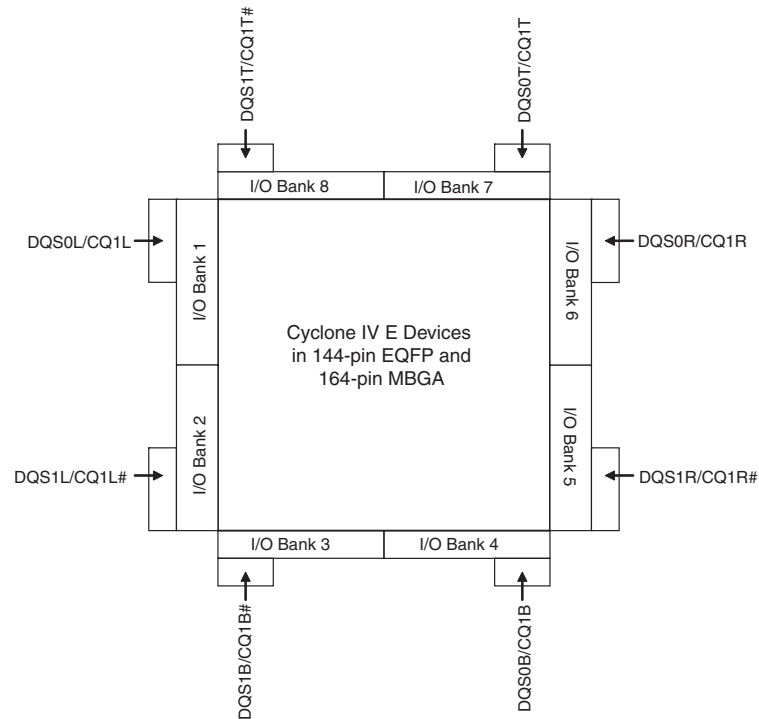
In Cyclone IV devices, the ×9 mode uses the same DQ and DQS pins as the ×8 mode, and one additional DQ pin that serves as a regular I/O pin in the ×8 mode. The ×18 mode uses the same DQ and DQS pins as ×16 mode, with two additional DQ pins that serve as regular I/O pins in the ×16 mode. Similarly, the ×36 mode uses the same DQ and DQS pins as the ×32 mode, with four additional DQ pins that serve as regular I/O pins in the ×32 mode. When not used as DQ or DQS pins, the memory interface pins are available as regular I/O pins.

Optional Parity, DM, and Error Correction Coding Pins

Cyclone IV devices support parity in ×9, ×18, and ×36 modes. One parity bit is available per eight bits of data pins. You can use any of the DQ pins for parity in Cyclone IV devices because the parity pins are treated and configured similarly to DQ pins.

DM pins are only required when writing to DDR2 and DDR SDRAM devices. QDR II SRAM devices use the BWS# signal to select the byte to be written into memory. A low signal on the DM or BWS# pin indicates the write is valid. Driving the DM or BWS# pin high causes the memory to mask the DQ signals. Each group of DQS and DQ signals has one DM pin. Similar to the DQ output signals, the DM signals are clocked by the -90° shifted clock.

In Cyclone IV devices, the DM pins are preassigned in the device pinouts. The Quartus II Fitter treats the DQ and DM pins in a DQS group equally for placement purposes. The preassigned DQ and DM pins are the preferred pins to use.

Some DDR2 SDRAM and DDR SDRAM devices support error correction coding (ECC), a method of detecting and automatically correcting errors in data transmission. In 72-bit DDR2 or DDR SDRAM, there are eight ECC pins and 64 data pins. Connect the DDR2 and DDR SDRAM ECC pins to a separate DQS or DQ group in Cyclone IV devices. The memory controller needs additional logic to encode and decode the ECC data.

Address and Control/Command Pins

The address signals and the control or command signals are typically sent at a single data rate. You can use any of the user I/O pins on all I/O banks of Cyclone IV devices to generate the address and control or command signals to the memory device.

☞ Cyclone IV devices do not support QDR II SRAM in the burst length of two.

Memory Clock Pins

In DDR2 and DDR SDRAM memory interfaces, the memory clock signals (CK and CK#) are used to capture the address signals and the control or command signals. Similarly, QDR II SRAM devices use the write clocks (K and K#) to capture the address and command signals. The CK/CK# and K/K# signals are generated to resemble the write-data strobe using the DDIO registers in Cyclone IV devices.

☞ CK/CK# pins must be placed on differential I/O pins (DIFFIO in Pin Planner) and in the same bank or on the same side as the data pins. You can use either side of the device for wraparound interfaces. As seen in the Pin Planner Pad View, CK0 cannot be located in the same row and column pad group as any of the interfacing DQ pins.

👣 For more information about memory clock pin placement, refer to *Volume 2: Device, Pin, and Board Layout Guidelines* of the *External Memory Interface Handbook.*

# Cyclone IV Devices Memory Interfaces Features

This section discusses Cyclone IV memory interfaces, including DDR input registers, DDR output registers, OCT, and phase-lock loops (PLLs).

DDR Input Registers

The DDR input registers are implemented with three internal logic element (LE) registers for every DQ pin. These LE registers are located in the logic array block (LAB) adjacent to the DDR input pin.

Figure 7–7 illustrates Cyclone IV DDR input registers.

Figure 7–7. Cyclone IV DDR Input Registers
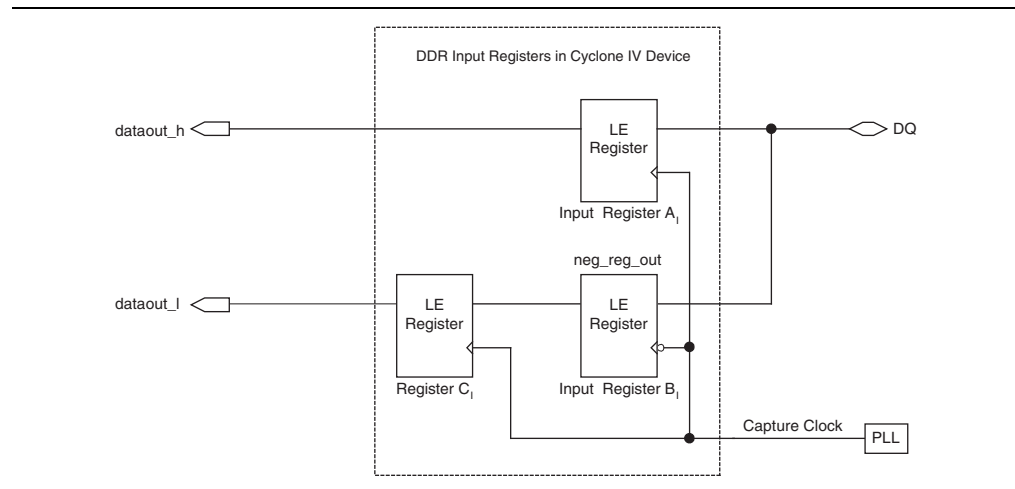
These DDR input registers are implemented in the core of devices. The DDR data is first fed to two registers, input register A_I and input register B_I.

- Input register A_I captures the DDR data present during the rising edge of the clock
- Input register B_I captures the DDR data present during the falling edge of the clock
- Register C_I aligns the data before it is synchronized with the system clock

The data from the DDR input register is fed to two registers, sync_reg_h and sync_reg_l, then the data is typically transferred to a FIFO block to synchronize the two data streams to the rising edge of the system clock. Because the read-capture clock is generated by the PLL, the read-data strobe signal (DQS or CQ) is not used during read operation in Cyclone IV devices; hence, postamble is not a concern in this case.

DDR Output Registers

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

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

Figure 7–8. Cyclone IV Dedicated Write DDIO
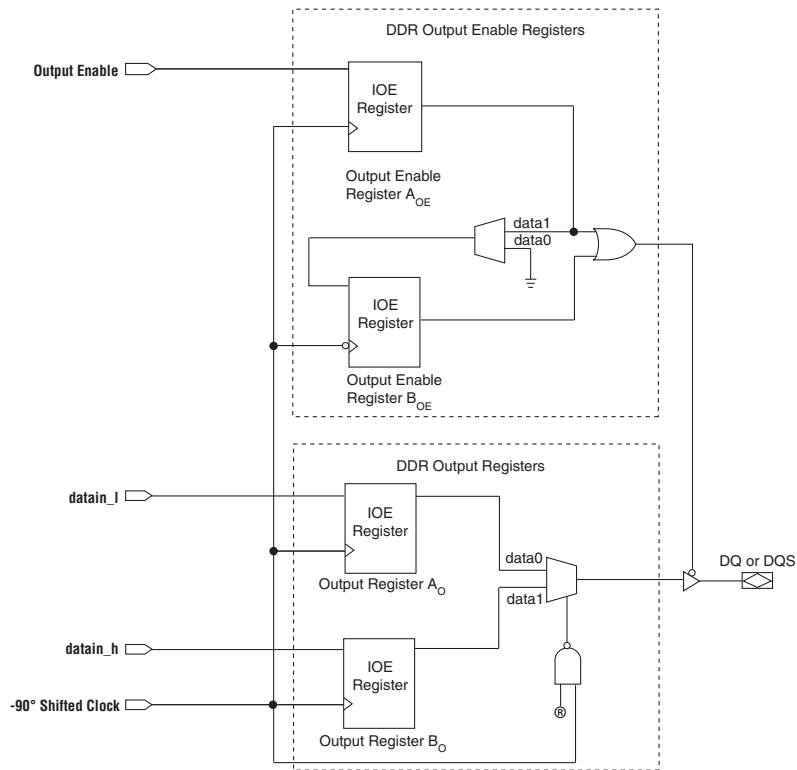
The two DDR output registers are located in the I/O element (IOE) block. Two serial data streams routed through `datain_l` and `datain_h`, are fed into two registers, `output register Ao` and `output register Bo`, respectively, on the same clock edge. The output from `output register Ao` is captured on the falling edge of the clock, while the output from `output register Bo` is captured on the rising edge of the clock. The registered outputs are multiplexed by the common clock to drive the DDR output pin at twice the data rate.

The DDR output enable path has a similar structure to the DDR output path in the IOE block. The second output enable register provides the write preamble for the `DQS` strobe in DDR external memory interfaces. This active-low output enable register extends the high-impedance state of the pin by half a clock cycle to provide the external memory's `DQS` write preamble time specification.
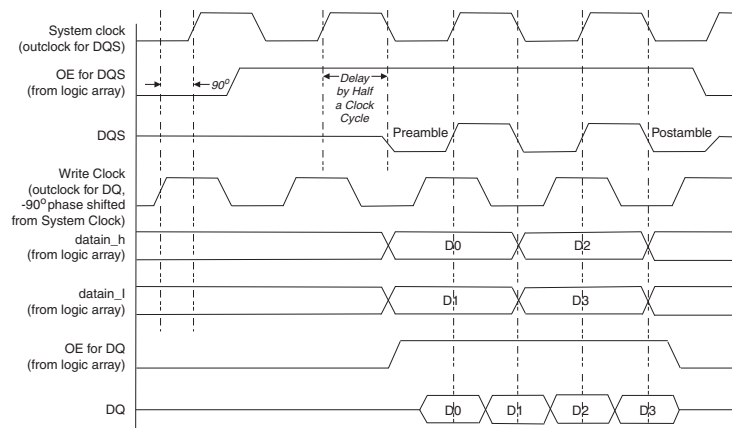
For more information about Cyclone IV IOE registers, refer to the *Cyclone IV Device I/O Features* chapter.

Figure 7–9 illustrates how the second output enable register extends the DQS high-impedance state by half a clock cycle during a write operation.

Figure 7–9. Extending the OE Disable by Half a Clock Cycle for a Write Transaction *(1)*

Note to Figure 7–9:

(1) The waveform reflects the software simulation result. The OE signal is an active low on the device. However, the Quartus II software implements the signal as an active high and automatically adds an inverter before the A_{OE} register D input.

OCT with Calibration

Cyclone IV devices support calibrated on-chip series termination (R_S OCT) in both vertical and horizontal I/O banks. To use the calibrated OCT, you must use the RUP and RDN pins for each R_S OCT control block (one for each side). You can use each OCT calibration block to calibrate one type of termination with the same V_{CCIO} for that given side.

For more information about the Cyclone IV devices OCT calibration block, refer to the *Cyclone IV Device I/O Features* chapter.

PLL

When interfacing with external memory, the PLL is used to generate the memory system clock, the write clock, the capture clock and the logic-core clock. The system clock generates the DQS write signals, commands, and addresses. The write-clock is shifted by -90° from the system clock and generates the DQ signals during writes. You can use the PLL reconfiguration feature to calibrate the read-capture phase shift to balance the setup and hold margins.

The PLL is instantiated in the ALTMEMPHY megafunction. All outputs of the PLL are used when the ALTMEMPHY megafunction is instantiated to interface with external memories. PLL reconfiguration is used in the ALTMEMPHY megafunction to calibrate and track the read-capture phase to maintain the optimum margin.

For more information about usage of PLL outputs by the ALTMEMPHY megafunction, refer to the *External Memory Interface Handbook*.

For more information about Cyclone IV PLL, refer to the *Clock Networks and PLLs in Cyclone IV Devices* chapter.

Document Revision History

Table 7–3 lists the revision history for this chapter.

Table 7–3. Document Revision History

Date	Version	Changes
March 2016	2.6	■ Updated Table 7–1 to remove support for the N148 package. ■ Updated note (1) in Figure 7–2 to remove support for the N148 package. ■ Updated Figure 7–4 to remove support for the N148 package.
May 2013	2.5	Updated Table 7–2 to add new device options and packages.
February 2013	2.4	Updated Table 7–2 to add new device options and packages.
October 2012	2.3	Updated Table 7–1 and Table 7–2.
December 2010	2.2	■ Updated for the Quartus II software version 10.1 release. ■ Added Cyclone IV E new device package information. ■ Updated Table 7–2. ■ Minor text edits.
November 2010	2.1	Updated "Data and Data Clock/Strobe Pins" section.
February 2010	2.0	■ Added Cyclone IV E devices information for the Quartus II software version 9.1 SP1 release. ■ Updated Table 7–1. ■ Added Table 7–2. ■ Added Figure 7–5 and Figure 7–6.
November 2009	1.0	Initial release.

This section includes the following chapters:

Revision History

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

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

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

■ Active serial (AS)

■ Active parallel (AP) (supported in Cyclone IV E devices only)

■ Passive serial (PS)

■ Fast passive parallel (FPP) (not supported in EP4CGX15, EP4CGX22, and EP4CGX30 [except for the F484 package] devices)

■ JTAG

Cyclone IV devices offer the following configuration features:

■ Configuration data decompression ("Configuration Data Decompression" on page 8–2)

■ Remote system upgrade ("Remote System Upgrade" on page 8–69)

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

Configuration

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

Feedback   Subscribe

Configuration Features

Table 8–1 lists the configuration methods you can use in each configuration scheme.

Table 8–1. Configuration Features in Cyclone IV Devices

Configuration Scheme	Configuration Method	Decompression	Remote System Upgrade [1]
AS	Serial Configuration Device	✔	✔
AP	Supported Flash Memory [2]	—	✔
PS	External Host with Flash Memory	✔	✔ [3]
PS	Download Cable	✔	—
FPP	External Host with Flash Memory	—	✔ [3]
JTAG based configuration	External Host with Flash Memory	—	—
JTAG based configuration	Download Cable	—	—

Notes to Table 8–1:

(1) Remote update mode is supported when you use the Remote System Upgrade feature. You can enable or disable remote update mode with an option setting in the Quartus® II software.

(2) For more information about the supported device families for the Micron commodity parallel flash, refer to Table 8–10 on page 8–22.

(3) Remote update mode is supported externally using the Parallel Flash Loader (PFL) with the Quartus II software.

Configuration Data Decompression

Cyclone IV devices support configuration data decompression, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other memory and send the compressed bitstream to Cyclone IV devices. During configuration, Cyclone IV devices decompress the bitstream in real time and program the SRAM cells.

☞ Compression may reduce the configuration bitstream size by 35 to 55%.

When you enable compression, the Quartus II software generates configuration files with compressed configuration data. This compressed file reduces the storage requirements in the configuration device or flash memory and decreases the time required to send the bitstream to the Cyclone IV device. The time required by a Cyclone IV device to decompress a configuration file is less than the time required to send the configuration data to the device. There are two methods for enabling compression for the Cyclone IV device bitstreams in the Quartus II software:

- Before design compilation (through the Compiler Settings menu)
- After design compilation (through the **Convert Programming Files** dialog box)

To enable compression in the compiler settings of the project in the Quartus II software, perform the following steps:

1. On the Assignments menu, click **Device**. The **Settings** dialog box appears.

2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.

3. Click the **Configuration** tab.

4. Turn on **Generate compressed bitstreams**.

5. Click **OK**.

6. In the **Settings** dialog box, click **OK**.

You can enable compression when creating programming files from the **Convert Programming Files** dialog box. To enable compression, perform the following steps:

1. On the File menu, click **Convert Programming Files**.

2. Under **Output programming file**, select your desired file type from the **Programming file type** list.

3. If you select **Programmer Object File (.pof)**, you must specify the configuration device in the **Configuration device** list.

4. Under **Input files to convert**, select **SOF Data**.

5. Click **Add File** to browse to the Cyclone IV device SRAM object files (**.sof**).

6. In the **Convert Programming Files** dialog box, select the **.pof** you added to **SOF Data** and click **Properties**.

7. In the **SOF File Properties** dialog box, turn on the **Compression** option.

When multiple Cyclone IV devices are cascaded, you can selectively enable the compression feature for each device in the chain. Figure 8–1 shows a chain of two Cyclone IV devices. The first device has compression enabled and receives compressed bitstream from the configuration device. The second device has the compression feature disabled and receives uncompressed data. You can generate programming files for this setup in the **Convert Programming Files** dialog box.

Figure 8–1. Compressed and Uncompressed Configuration Data in the Same Configuration File
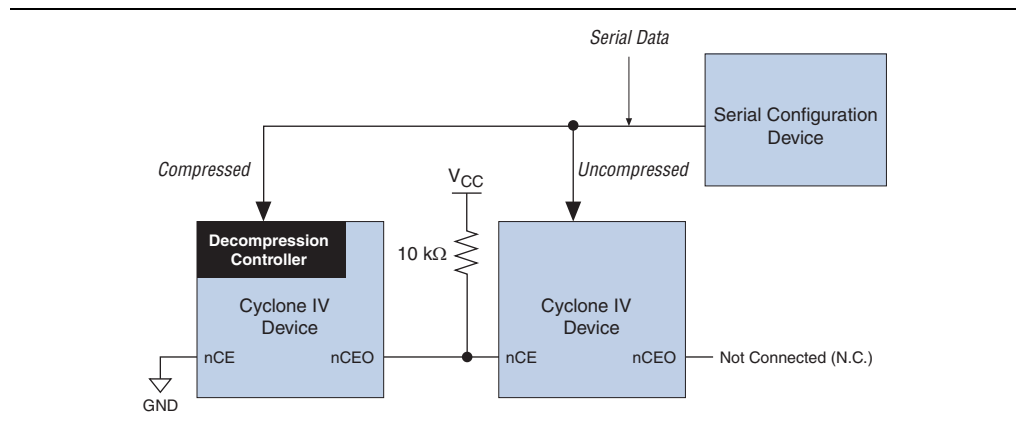
Configuration Requirement

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

■ "Power-On Reset (POR) Circuit" on page 8–4

■ "Configuration File Size" on page 8–4

■ "Power Up" on page 8–6

Power-On Reset (POR) Circuit

The POR circuit keeps the device in reset state until the power supply voltage levels have stabilized during device power up. After device power up, the device does not release nSTATUS until V_{CCINT}, V_{CCA}, and V_{CCIO} (for I/O banks in which the configuration and JTAG pins reside) are above the POR trip point of the device. V_{CCINT} and V_{CCA} are monitored for brown-out conditions after device power up.

☞ V_{CCA} is the analog power to the phase-locked loop (PLL).

In some applications, it is necessary for a device to wake up very quickly to begin operation. Cyclone IV devices offer the fast POR time option to support fast wake-up time applications. The fast POR time option has stricter power-up requirements when compared with the standard POR time option. You can select either the fast option or the standard POR option with the MSEL pin settings.

☞ If your system exceeds the fast or standard POR time, you must hold nCONFIG low until all the power supplies are stable.

👣 For more information about the POR specifications, refer to the *Cyclone IV Device Datasheet*.

👣 For more information about the wake-up time and POR circuit, refer to the *Power Requirements for Cyclone IV Devices* chapter.

Configuration File Size

Table 8–2 lists the approximate uncompressed configuration file sizes for Cyclone IV devices. To calculate the amount of storage space required for multiple device configurations, add the file size of each device together.

Table 8–2. Uncompressed Raw Binary File (.rbf) Sizes for Cyclone IV Devices (Part 1 of 2)

Device		Data Size (bits)
Cyclone IV E	EP4CE6	2,944,088
	EP4CE10	2,944,088
	EP4CE15	4,086,848
	EP4CE22	5,748,552
	EP4CE30	9,534,304
	EP4CE40	9,534,304
	EP4CE55	14,889,560
	EP4CE75	19,965,752
	EP4CE115	28,571,696

Table 8–2. Uncompressed Raw Binary File (.rbf) Sizes for Cyclone IV Devices (Part 2 of 2)

Device		Data Size (bits)
Cyclone IV GX	EP4CGX15	3,805,568
	EP4CGX22	7,600,040
	EP4CGX30	7,600,040
		22,010,888 [1]
	EP4CGX50	22,010,888
	EP4CGX75	22,010,888
	EP4CGX110	39,425,016
	EP4CGX150	39,425,016

Note to Table 8–2:

(1) Only for the F484 package.

Use the data in Table 8–2 to estimate the file size before design compilation. Different configuration file formats, such as Hexadecimal (**.hex**) or Tabular Text File (**.ttf**) formats, have different file sizes. However, for any specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you use compression, the file size varies after each compilation, because the compression ratio depends on the design.

For more information about setting device configuration options or creating configuration files, refer to the *Software Settings* section in volume 2 of the *Configuration Handbook*.

Configuration and JTAG Pin I/O Requirements

Cyclone IV devices are manufactured using the TSMC 60-nm low-k dielectric process. Although Cyclone IV devices use TSMC 2.5-V transistor technology in the I/O buffers, the devices are compatible and able to interface with 2.5, 3.0, and 3.3-V configuration voltage standards by following specific requirements.

All I/O inputs must maintain a maximum AC voltage of 4.1 V. When using a serial configuration device in an AS configuration scheme, you must connect a 25-Ω series resistor for the DATA[0] pin. When cascading the Cyclone IV device family in a multi-device configuration for AS, AP, FPP, and PS configuration schemes, you must connect the repeater buffers between the master and slave devices for the DATA and DCLK pins. When using the JTAG configuration scheme in a multi-device configuration, connect 25-Ω resistors on both ends of the TDO-TDI path if the TDO output driver is a non-Cyclone IV device.

The output resistance of the repeater buffers and the TDO path for all cases must fit the maximum overshoot equation shown in Equation 8–1.

Equation 8–1. [1]

$$0.8Z_O \le R_E \le 1.8Z_O$$

Note to Equation 8–1:

(1) Z_O is the transmission line impedance and R_E is the equivalent resistance of the output buffer.

Configuration Process

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

■ "Power Up" on page 8–6

■ "Reset" on page 8–6

■ "Configuration" on page 8–6

■ "Configuration Error" on page 8–7

■ "Initialization" on page 8–7

■ "User Mode" on page 8–7

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

Power Up

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

Reset

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

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

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

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

Configuration

Configuration data is latched into the Cyclone IV device at each `DCLK` cycle. However, the width of the data bus and the configuration time taken for each scheme are different. After the device receives all the configuration data, the device releases the open-drain `CONF_DONE` pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on the `CONF_DONE` pin indicates that the configuration is complete and initialization of the device can begin.

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.

Configuration Scheme

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

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

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

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

Notes to Table 8–3:

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

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

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

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

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

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

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) [1]
JTAG-based configuration [2]	[3]	[3]	[3]	[3]	—	—

Notes to Table 8–4:

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

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

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

☞ Smaller Cyclone IV E devices or package options (E144 and F256 packages) do not have the MSEL[3] pin. The AS Fast POR configuration scheme at 3.0- or 2.5-V configuration voltage standard and the AP configuration scheme are not supported in Cyclone IV E devices without the MSEL[3] pin. To configure these devices with other supported configuration schemes, select MSEL[2..0] pins according to the MSEL settings in Table 8–5.

Table 8–5. Configuration Schemes for Cyclone IV E Devices

Configuration Scheme	MSEL3	MSEL2	MSEL1	MSEL0	POR Delay	Configuration Voltage Standard (V) [1]
AS	1	1	0	1	Fast	3.3
	0	1	0	0	Fast	3.0, 2.5
	0	0	1	0	Standard	3.3
	0	0	1	1	Standard	3.0, 2.5
AP	0	1	0	1	Fast	3.3
	0	1	1	0	Fast	1.8
	0	1	1	1	Standard	3.3
	1	0	1	1	Standard	3.0, 2.5
	1	0	0	0	Standard	1.8
PS	1	1	0	0	Fast	3.3, 3.0, 2.5
	0	0	0	0	Standard	3.3, 3.0, 2.5
FPP	1	1	1	0	Fast	3.3, 3.0, 2.5
	1	1	1	1	Fast	1.8, 1.5
JTAG-based configuration [2]	[3]	[3]	[3]	[3]	—	—

Notes to Table 8–5:

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

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

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

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

AS Configuration (Serial Configuration Devices)

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

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

Serial configuration devices provide a serial interface to access the configuration data. During device configuration, Cyclone IV devices read the configuration data through the serial interface, decompress the data if necessary, and configure their SRAM cells. This scheme is referred to as the AS configuration scheme because the device controls the configuration interface.

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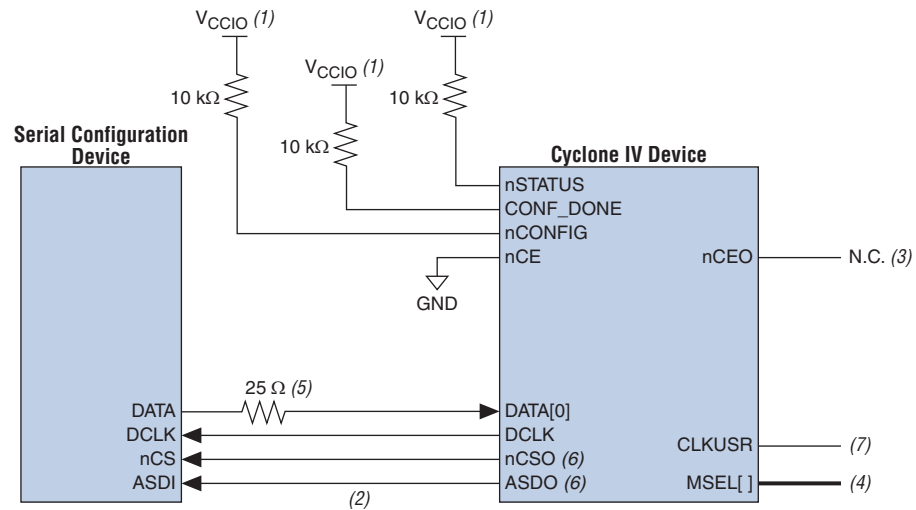
Single-Device AS Configuration

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

■ Serial clock input (`DCLK`)

■ Serial data output (`DATA`)

■ Active-low chip select (`nCS`)

■ AS data input (`ASDI`)

This four-pin interface connects to Cyclone IV device pins, as shown in Figure 8–2.

Figure 8–2. Single-Device AS Configuration

Notes to Figure 8–2:

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

(2) Cyclone IV devices use the ASDO-to-ASDI path to control the configuration device.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) Connect the series resistor at the near end of the serial configuration device.

(6) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as the DATA[1] pin in AP and FPP modes.

(7) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

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

☞ The 25-Ω resistor at the near end of the serial configuration device for DATA[0] works to minimize the driver impedance mismatch with the board trace and reduce the overshoot seen at the Cyclone IV device DATA[0] input pin.

In the single-device AS configuration, the maximum board loading and board trace length between the supported serial configuration device and the Cyclone IV device must follow the recommendations in Table 8–7 on page 8–18.

The DCLK generated by the Cyclone IV device controls the entire configuration cycle and provides timing for the serial interface. Cyclone IV devices use an internal oscillator or an external clock source to generate the DCLK. For Cyclone IV E devices, you can use a 40-MHz internal oscillator to generate the DCLK and for Cyclone IV GX devices you can use a slow clock (20 MHz maximum) or a fast clock (40 MHz maximum) from the internal oscillator or an external clock from CLKUSR to generate the DCLK. There are some variations in the internal oscillator frequency because of the process, voltage, and temperature (PVT) conditions in Cyclone IV

devices. The internal oscillator is designed to ensure that its maximum frequency is guaranteed to meet EPCS device specifications. Cyclone IV devices offer the option to select CLKUSR as the external clock source for DCLK. You can change the clock source option in the Quartus II software in the **Configuration** tab of the **Device and Pin Options** dialog box.

☞ EPCS1 does not support Cyclone IV devices because of its insufficient memory capacity.

Table 8–6. AS DCLK Output Frequency

Oscillator	Minimum	Typical	Maximum	Unit
40 MHz	20	30	40	MHz

In configuration mode, the Cyclone IV device enables the serial configuration device by driving the nCSO output pin low, which connects to the nCS pin of the configuration device. The Cyclone IV device uses the DCLK and DATA[1] pins to send operation commands and read address signals to the serial configuration device. The configuration device provides data on its DATA pin, which connects to the DATA[0] input of the Cyclone IV device.

All AS configuration pins (DATA[0], DCLK, nCSO, and DATA[1]) have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and are driven high by the weak internal pull-up resistors.

The timing parameters for AS mode are not listed here because the t_{CF2CD}, t_{CF2ST0}, t_{CFG}, t_{STATUS}, t_{CF2ST1}, and t_{CD2UM} timing parameters are identical to the timing parameters for PS mode shown in Table 8–12 on page 8–36.

Multi-Device AS Configuration

You can configure multiple Cyclone IV devices with a single serial configuration device. When the first device captures all its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. If the last device in the chain is a Cyclone IV device, you can leave the nCEO pin of the last device unconnected or use it as a user I/O pin after configuration. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA[0] pins of each device in the chain are connected together (Figure 8–3).

Figure 8–3. Multi-Device AS Configuration
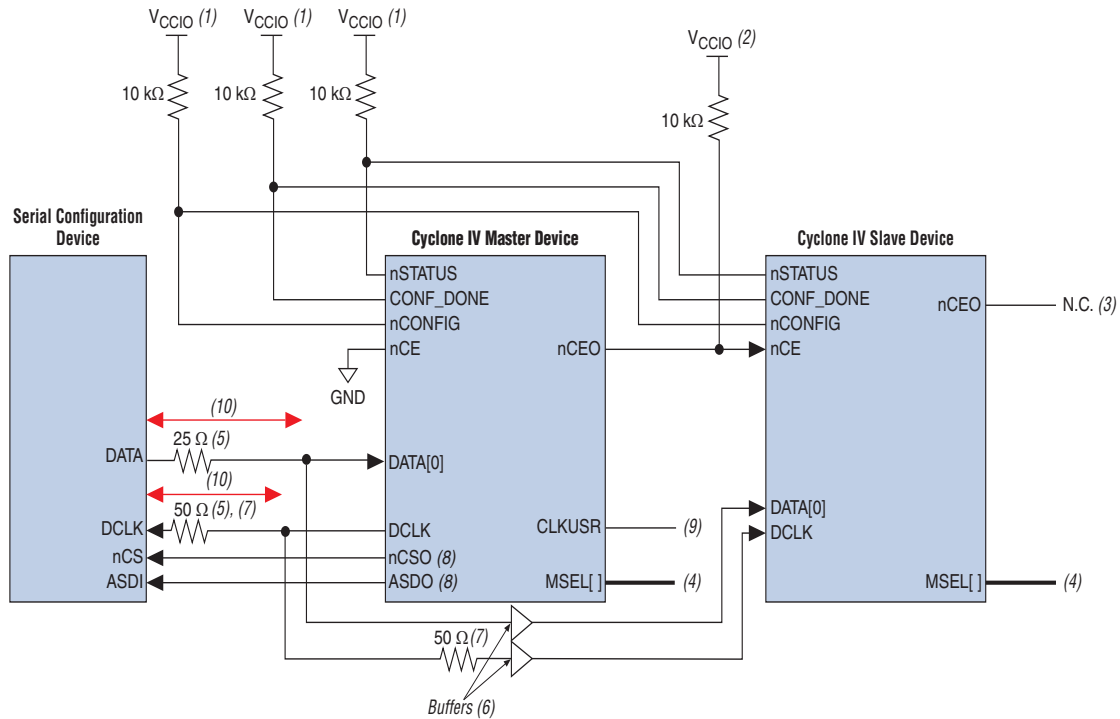
Notes to Figure 8–3:

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

(2) Connect the pull-up resistor to the V$_{CCIO}$ supply voltage of I/O bank in which the nCE pin resides.

(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device of the Cyclone IV device in AS mode and the slave devices in PS mode. To connect the MSEL pins for the master device in AS mode and slave devices in PS mode, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V$_{CCA}$ or GND.

(5) Connect the series resistor at the near end of the serial configuration device.

(6) Connect the repeater buffers between the master and slave devices of the Cyclone IV device for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) The 50-Ω series resistors are optional if the 3.3-V configuration voltage standard is applied. For optimal signal integrity, connect these 50-Ω series resistors if the 2.5- or 3.0-V configuration voltage standard is applied.

(8) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(9) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

(10) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

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

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

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

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

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

Configuring Multiple Cyclone IV Devices with the Same Design

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

■ Multiple **.sof**

■ Single **.sof**

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

Multiple SRAM Object Files

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

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

four devices. During the first configuration cycle, the master device reads its configuration data from the serial configuration device while holding nCEO high. After completing its configuration cycle, the master device drives nCE low and sends the second copy of the configuration data to all three slave devices, configuring them simultaneously.

The advantage of the setup in Figure 8–4 is that you can have a different **.sof** for the master device. However, all the slave devices must be configured with the same **.sof**. You can either compress or uncompress the **.sof** in this configuration method.

☞ You can still use this method if the master and slave devices use the same **.sof**.

Figure 8–4. Multi-Device AS Configuration in Which Devices Receive the Same Data with Multiple .sof
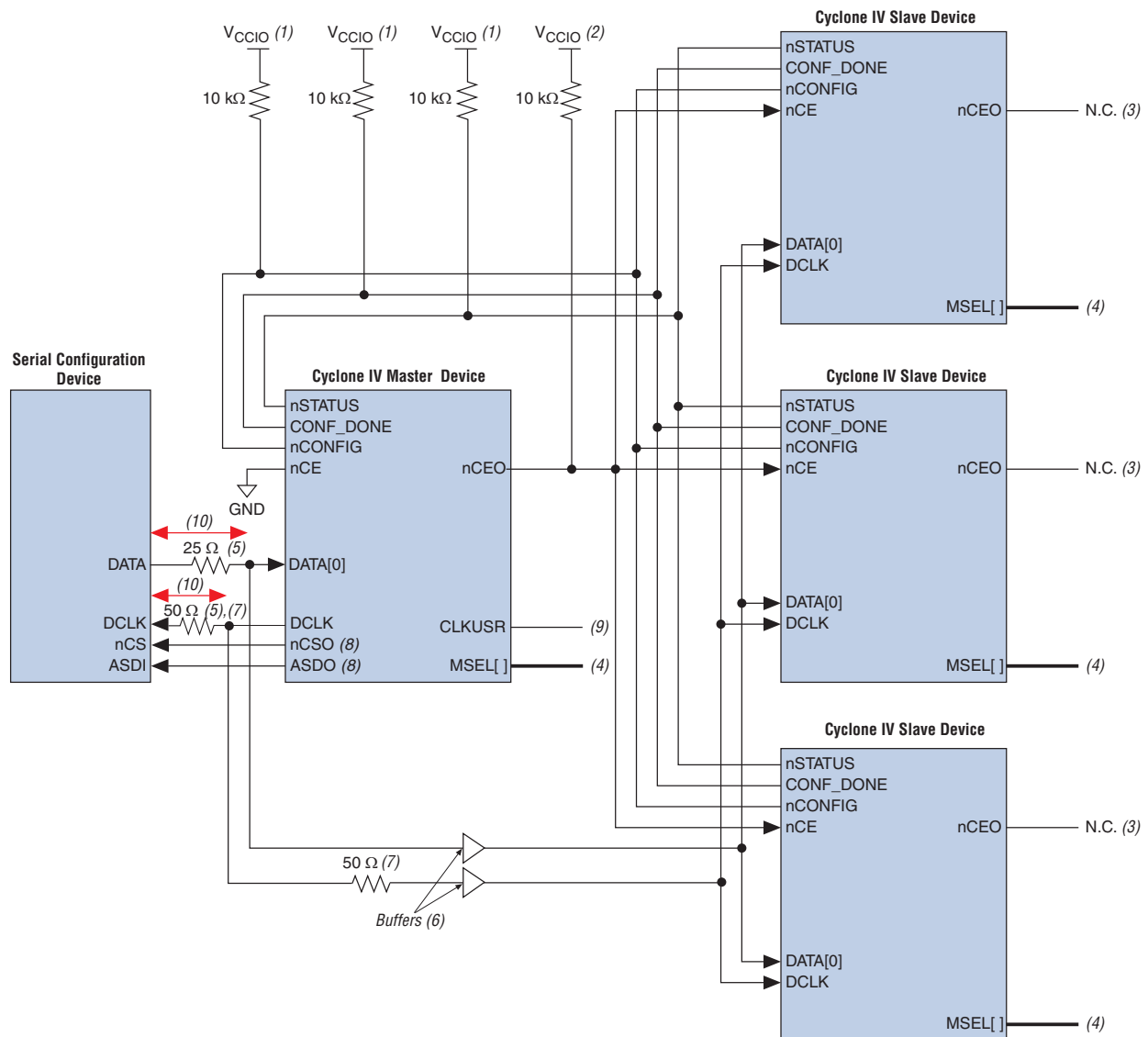
Notes to Figure 8–4:

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

(2) Connect the pull-up resistor to the V_{CCIO} supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device in AS mode and the slave devices in PS mode. To connect the MSEL pins for the master device in AS mode and the slave devices in PS mode, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) Connect the series resistor at the near end of the serial configuration device.

(6) Connect the repeater buffers between the master and slave devices for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) The 50-Ω series resistors are optional if the 3.3-V configuration voltage standard is applied. For optimal signal integrity, connect these 50-Ω series resistors if the 2.5- or 3.0-V configuration voltage standard is applied.

(8) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(9) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

(10) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

Single SRAM Object File

The second method configures both the master device and slave devices with the same **.sof**. The serial configuration device stores one copy of the **.sof**. You must set up one or more slave devices in the chain. All the slave devices must be set up in the same way (Figure 8–5).

Figure 8–5. Multi-Device AS Configuration in Which Devices Receive the Same Data with a Single .sof
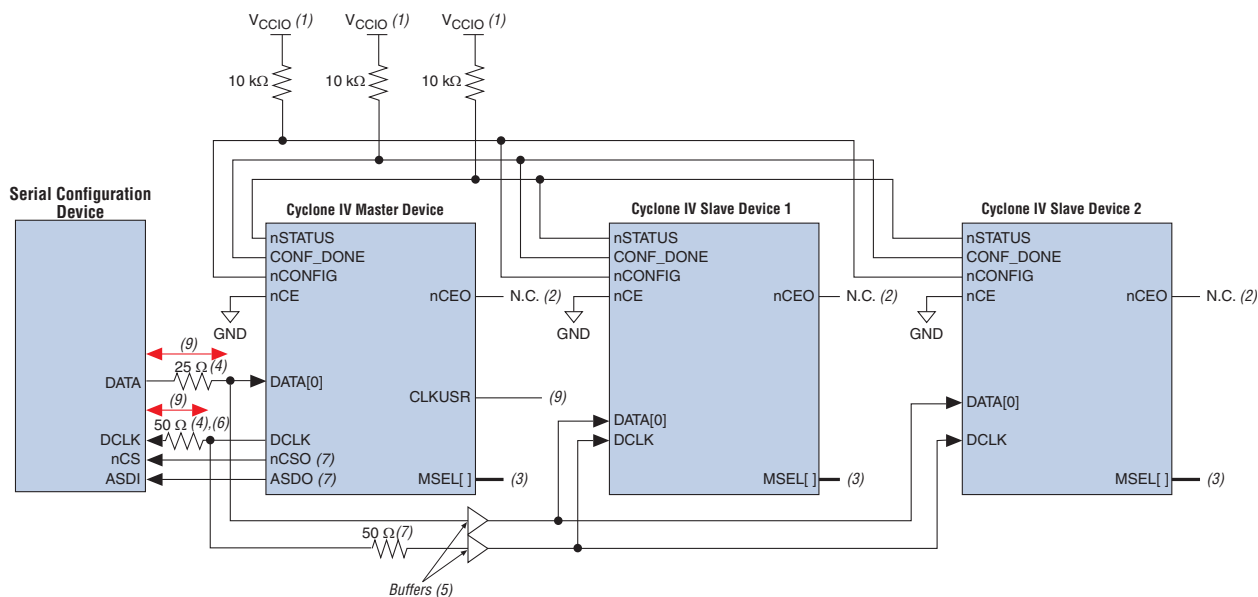
Notes to Figure 8–5:

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

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device of the Cyclone IV device in AS mode and the slave devices in PS mode. To connect the MSEL pins for the master device in AS mode and slave devices in PS mode, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(4) Connect the series resistor at the near end of the serial configuration device.

(5) Connect the repeater buffers between the master and slave devices for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(6) The 50-Ω series resistors are optional if the 3.3-V configuration voltage standard is applied. For optimal signal integrity, connect these 50-Ω series resistors if the 2.5- or 3.0-V configuration voltage standard is applied.

(7) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(8) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

(9) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

In this setup, all the Cyclone IV devices in the chain are connected for concurrent configuration. This reduces the AS configuration time because all the Cyclone IV devices are configured in one configuration cycle. Connect the nCE input pins of all the Cyclone IV devices to GND. You can either leave the nCEO output pins on all the Cyclone IV devices unconnected or use the nCEO output pins as normal user I/O pins. The DATA and DCLK pins are connected in parallel to all the Cyclone IV devices.

Altera recommends putting a buffer before the DATA and DCLK output from the master device to avoid signal strength and signal integrity issues. The buffer must not significantly change the DATA-to-DCLK relationships or delay them with respect to other AS signals (ASDI and nCS). Also, the buffer must only drive the slave devices to ensure that the timing between the master device and the serial configuration device is unaffected.

This configuration method supports both compressed and uncompressed **.sof**. Therefore, if the configuration bitstream size exceeds the capacity of a serial configuration device, you can enable the compression feature in the **.sof** or you can select a larger serial configuration device.

Guidelines for Connecting a Serial Configuration Device to Cyclone IV Devices for an AS Interface

For single- and multi-device AS configurations, the board trace length and loading between the supported serial configuration device and Cyclone IV device must follow the recommendations listed in Table 8–7.

Table 8–7. Maximum Trace Length and Loading for AS Configuration

Cyclone IV Device AS Pins	Maximum Board Trace Length from a Cyclone IV Device to a Serial Configuration Device (Inches)		Maximum Board Load (pF)
	Cyclone IV E	Cyclone IV GX	
DCLK	10	6	15
DATA[0]	10	6	30
nCSO	10	6	30
ASDO	10	6	30

Note to Table 8–7:

(1) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

Estimating AS Configuration Time

AS configuration time is dominated by the time it takes to transfer data from the serial configuration device to the Cyclone IV device. This serial interface is clocked by the Cyclone IV device DCLK output (generated from a 40-MHz internal oscillator for Cyclone IV E devices, a 20- or 40-MHz internal oscillator, or an external CLKUSR of up to 40 MHz for Cyclone IV GX devices).

Equation 8–2 and Equation 8–3 show the configuration time calculations.

Equation 8–2.

$$\text{Size} \times \left(\frac{\text{maximum DCLK period}}{1 \text{ bit}} \right) = \text{estimated maximum configuration ti}$$

Equation 8–3.

$$9{,}600{,}000 \text{ bits} \times \left(\frac{50 \text{ ns}}{1 \text{ bit}} \right) = 480 \text{ ms}$$

Table 8–8 provides the configuration time for AS configuration.

Table 8–8. AS Configuration Time for Cyclone IV Devices [1]

Symbol	Parameter	Cyclone IV E	Cyclone IV GX	Unit
t_{SU}	Setup time	10	8	ns
t_H	Hold time	0	0	ns
t_{CO}	Clock-to-output time	4	4	ns

Note to Table 8–8:

(1) For the AS configuration timing diagram, refer to the *Serial Configuration (EPCS) Devices Datasheet*.

Enabling compression reduces the amount of configuration data that is sent to the Cyclone IV device, which also reduces configuration time. On average, compression reduces configuration time by 50%.

Programming Serial Configuration Devices

Serial configuration devices are non-volatile, flash memory-based devices. You can program these devices in-system with the USB-Blaster™ or ByteBlaster™ II download cables. Alternatively, you can program them with the Altera Programming Unit (APU), supported third-party programmers, or a microprocessor with the SRunner software driver.

You can perform in-system programming of serial configuration devices through the AS programming interface. During in-system programming, the download cable disables device access to the AS interface by driving the nCE pin high. Cyclone IV devices are also held in reset by a low level on nCONFIG. After programming is complete, the download cable releases nCE and nCONFIG, allowing the pull-down and pull-up resistors to drive V_{CC} and GND, respectively.

To perform in-system programming of a serial configuration device through the AS programming interface, you must place the diodes and capacitors as close as possible to the Cyclone IV device. You must ensure that the diodes and capacitors maintain a maximum AC voltage of 4.1 V (Figure 8–6).

☞ If you want to use the setup shown in Figure 8–6 to perform in-system programming of a serial configuration device and single- or multi-device AS configuration, you do not require a series resistor on the DATA line at the near end of the serial configuration device. The existing diodes and capacitors are sufficient.

Altera has developed the Serial FlashLoader (SFL), a JTAG-based in-system programming solution for Altera serial configuration devices. The SFL is a bridge design for the Cyclone IV device that uses its JTAG interface to access the EPCS JIC (JTAG Indirect Configuration Device Programming) file and then uses the AS interface to program the EPCS device. Both the JTAG interface and AS interface are bridged together inside the SFL design.

👣 For more information about implementing the SFL with Cyclone IV devices, refer to *AN 370: Using the Serial FlashLoader with the Quartus II Software*.

> For more information about the USB-Blaster download cable, refer to the *USB-Blaster Download Cable User Guide*. For more information about the ByteBlaster II download cable, refer to the *ByteBlaster II Download Cable User Guide*.

Figure 8–6 shows the download cable connections to the serial configuration device.

Figure 8–6. In-System Programming of Serial Configuration Devices
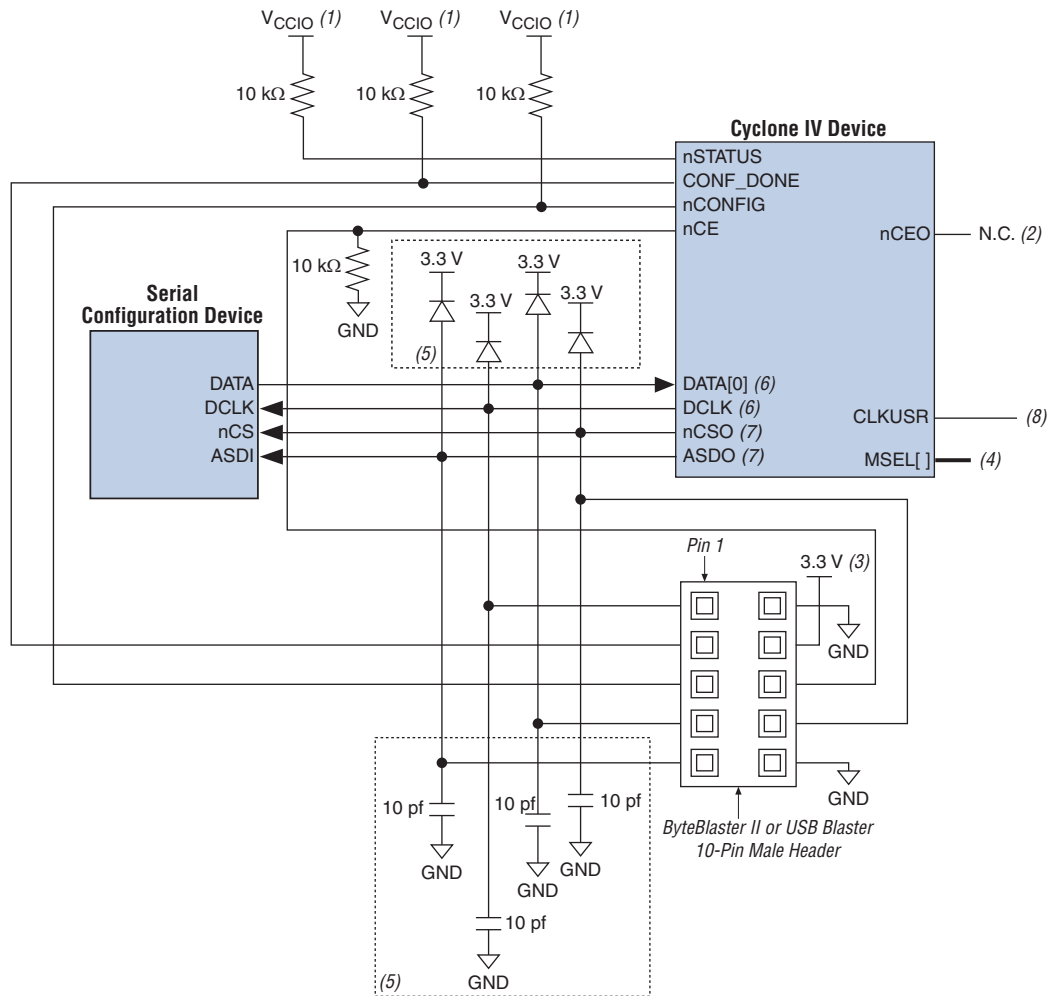
Notes to Figure 8–6:

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

(2) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(3) Power up the V$_{CC}$ of the ByteBlaster II or USB-Blaster download cable with the 3.3-V supply.

(4) The `MSEL` pin settings vary for different configuration voltage standards and POR time. To connect the `MSEL` pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the `MSEL` pins directly to V$_{CCA}$ or GND.

(5) The diodes and capacitors must be placed as close as possible to the Cyclone IV device. You must ensure that the diodes and capacitors maintain a maximum AC voltage of 4.1 V. The external diodes and capacitors are required to prevent damage to the Cyclone IV device AS configuration input pins due to possible overshoot when programming the serial configuration device with a download cable. Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes, for effective voltage clamping.

(6) When cascading Cyclone IV devices in a multi-device AS configuration, connect the repeater buffers between the master and slave devices for `DATA[0]` and `DCLK`. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) These pins are dual-purpose I/O pins. The `nCSO` pin functions as `FLASH_nCE` pin in AP mode. The `ASDO` pin functions as `DATA[1]` pin in AP and FPP modes.

(8) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

You can use the Quartus II software with the APU and the appropriate configuration device programming adapter to program serial configuration devices. All serial configuration devices are offered in an 8- or 16-pin small outline integrated circuit (SOIC) package.

In production environments, serial configuration devices are programmed using multiple methods. Altera programming hardware or other third-party programming hardware is used to program blank serial configuration devices before they are mounted onto PCBs. Alternatively, you can use an on-board microprocessor to program the serial configuration device in-system by porting the reference C-based SRunner software driver provided by Altera.

A serial configuration device is programmed in-system by an external microprocessor with the SRunner software driver. The SRunner software driver is a software driver developed for embedded serial configuration device programming, which is easily customized to fit in different embedded systems. The SRunner software driver is able to read a Raw Programming Data (**.rpd**) file and write to serial configuration devices. The serial configuration device programming time, using the SRunner software driver, is comparable to the programming time with the Quartus II software.

For more information about the SRunner software driver, refer to *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming* and the source code at the Altera website.

AP Configuration (Supported Flash Memories)

The AP configuration scheme is only supported in Cyclone IV E devices. In the AP configuration scheme, Cyclone IV E devices are configured using commodity 16-bit parallel flash memory. These external non-volatile configuration devices are industry standard microprocessor flash memories. The flash memories provide a fast interface to access configuration data. The speed up in configuration time is mainly due to the 16-bit wide parallel data bus, which is used to retrieve data from the flash memory.

Some of the smaller Cyclone IV E devices or package options do not support the AP configuration scheme. Table 8–9 lists the supported AP configuration scheme for each Cyclone IV E devices.

Table 8–9. Supported AP Configuration Scheme for Cyclone IV E Devices

Device	Package Options								
	E144	M164	M256	U256	F256	F324	U484	F484	F780
EP4CE6	—	—	—	—	—	—	—	—	—
EP4CE10	—	—	—	—	—	—	—	—	—
EP4CE15	—	—	—	—	—	—	—	✓	—
EP4CE22	—	—	—	—	—	—	—	—	—
EP4CE30	—	—	—	—	—	✓	—	✓	✓
EP4CE40	—	—	—	—	—	✓	✓	✓	✓
EP4CE55	—	—	—	—	—	—	✓	✓	✓
EP4CE75	—	—	—	—	—	—	✓	✓	✓
EP4CE115	—	—	—	—	—	—	—	✓	✓

During device configuration, Cyclone IV E devices read configuration data using the parallel interface and configure their SRAM cells. This scheme is referred to as the AP configuration scheme because the device controls the configuration interface. This scheme contrasts with the FPP configuration scheme, where an external host controls the interface.

AP Configuration Supported Flash Memories

The AP configuration controller in Cyclone IV E devices is designed to interface with two industry-standard flash families—the Micron P30 Parallel NOR flash family and the Micron P33 Parallel NOR flash family. Unlike serial configuration devices, both of the flash families supported in AP configuration scheme are designed to interface with microprocessors. By configuring from an industry standard microprocessor flash which allows access to the flash after entering user mode, the AP configuration scheme allows you to combine configuration data and user data (microprocessor boot code) on the same flash memory.

The Micron P30 flash family and the P33 flash family support a continuous synchronous burst read mode at 40 MHz DCLK frequency for reading data from the flash. Additionally, the Micron P30 and P33 flash families have identical pin-out and adopt similar protocols for data access.

☞ Cyclone IV E devices use a 40-MHz oscillator for the AP configuration scheme. The oscillator is the same oscillator used in the Cyclone IV E AS configuration scheme.

Table 8–10 lists the supported families of the commodity parallel flash for the AP configuration scheme.

Table 8–10. Supported Commodity Flash for AP Configuration Scheme for Cyclone IV E Devices [1]

Flash Memory Density	Micron P30 Flash Family [2]	Micron P33 Flash Family [3]
64 Mbit	✔	✔
128 Mbit	✔	✔
256 Mbit	✔	✔

Notes to Table 8–10:

(1) The AP configuration scheme only supports flash memory speed grades of 40 MHz and above.

(2) 3.3- , 3.0-, 2.5-, and 1.8-V I/O options are supported for the Micron P30 flash family.

(3) 3.3-, 3.0- and 2.5-V I/O options are supported for the Micron P33 flash family.

Configuring Cyclone IV E devices from the Micron P30 and P33 family 512-Mbit flash memory is possible, but you must properly drive the extra address and FLASH_nCE pins as required by these flash memories.

👣 To check for supported speed grades and package options, refer to the respective flash datasheets.

The AP configuration scheme in Cyclone IV E devices supports flash speed grades of 40 MHz and above. However, AP configuration for all these speed grades must be capped at 40 MHz. The advantage of faster speed grades is realized when your design in the Cyclone IV E devices accesses flash memory in user mode.

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

Single-Device AP Configuration

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

- Control pins
- Address pins
- Data pins

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

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

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

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

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

Figure 8–7 shows the interface for the Micron P30 flash memory and P33 flash memory to the Cyclone IV E device pins.

Figure 8–7. Single-Device AP Configuration Using Micron P30 and P33 Flash Memory
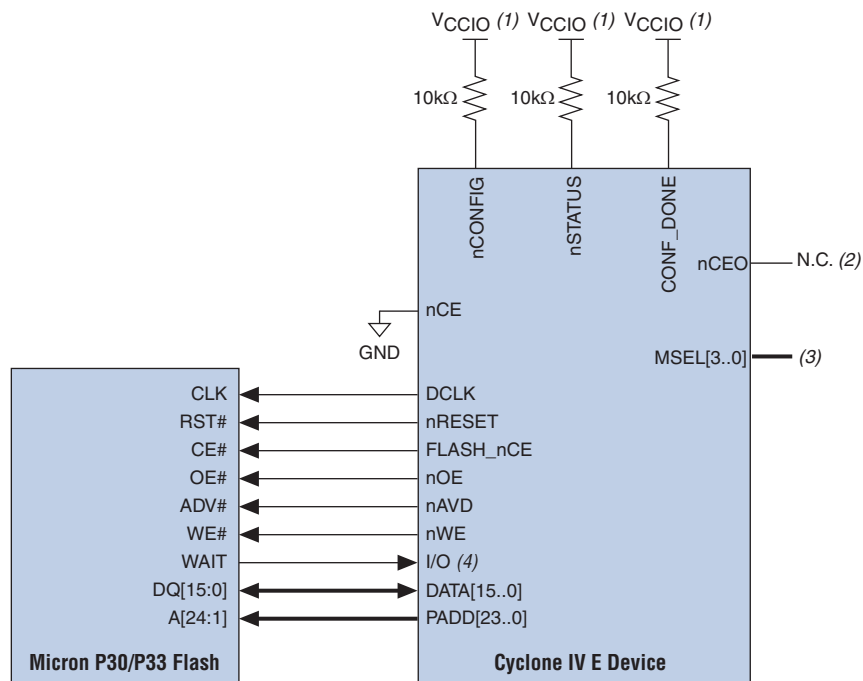
Notes to Figure 8–7:

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

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

(2) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(3) The `MSEL` pin settings vary for different configuration voltage standards and POR time. To connect `MSEL[3..0]`, refer to Table 8–5 on page 8–9. Connect the `MSEL` pins directly to V_{CCA} or GND.

(4) AP configuration ignores the `WAIT` signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use normal I/O to monitor the `WAIT` signal from the Micron P30 or P33 flash.

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

☞ In a single-device AP configuration, the maximum board loading and board trace length between supported parallel flash and Cyclone IV E devices must follow the recommendations listed in Table 8–11 on page 8–28.

☞ If you use the AP configuration scheme for Cyclone IV E devices, the V_{CCIO} of I/O banks 1, 6, 7, and 8 must be 3.3, 3.0, 2.5, or 1.8 V. Altera does not recommend using the level shifter between the Micron P30 or P33 flash and the Cyclone IV E device in the AP configuration scheme.

☞ There are no series resistors required in AP configuration mode for Cyclone IV E devices when using the Micron flash at 2.5-, 3.0-, and 3.3-V I/O standard. The output buffer of the Micron P30 IBIS model does not overshoot above 4.1 V. Thus, series resistors are not required for the 2.5-, 3.0-, and 3.3-V AP configuration option. However, if there are any other devices sharing the same flash I/Os with Cyclone IV E devices, all shared pins are still subject to the 4.1-V limit and may require series resistors.

Default read mode of the supported parallel flash memory and all writes to the parallel flash memory are asynchronous. Both the parallel flash families support a synchronous read mode, with data supplied on the positive edge of DCLK.

The serial clock (DCLK) generated by Cyclone IV E devices controls the entire configuration cycle and provides timing for the parallel interface.

Multi-Device AP Configuration

You can configure multiple Cyclone IV E devices using a single parallel flash. You can cascade multiple Cyclone IV E devices using the chip-enable (nCE) and chip-enable-out (nCEO) pins. The first device in the chain must have its nCE pin connected to GND. You must connect its nCEO pin to the nCE pin of the next device in the chain. Use an external 10-kΩ pull-up resistor to pull the nCEO signal high to its V$_{CCIO}$ level to help the internal weak pull-up resistor. When the first device captures all its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. You can leave the nCEO pin of the last device unconnected or use it as a user I/O pin after configuration if the last device in the chain is a Cyclone IV E device. The nCONFIG, nSTATUS, CONF_DONE, DCLK, DATA[15..8], and DATA[7..0] pins of each device in the chain are connected (Figure 8–8 on page 8–26 and Figure 8–9 on page 8–27).

The first Cyclone IV E device in the chain, as shown in Figure 8–8 on page 8–26 and Figure 8–9 on page 8–27, is the configuration master device and controls the configuration of the entire chain. You must connect its MSEL pins to select the AP configuration scheme. The remaining Cyclone IV E devices are used as configuration slaves. You must connect their MSEL pins to select the FPP configuration scheme. Any other Altera device that supports FPP configuration can also be part of the chain as a configuration slave.

The following are the configurations for the DATA[15..0] bus in a multi-device AP configuration:

■ Byte-wide multi-device AP configuration

■ Word-wide multi-device AP configuration

Byte-Wide Multi-Device AP Configuration

The simpler method for multi-device AP configuration is the byte-wide multi-device AP configuration. In the byte-wide multi-device AP configuration, the LSB of the DATA[7..0] pin from the flash and master device (set to the AP configuration scheme) is connected to the slave devices set to the FPP configuration scheme, as shown in Figure 8–8.

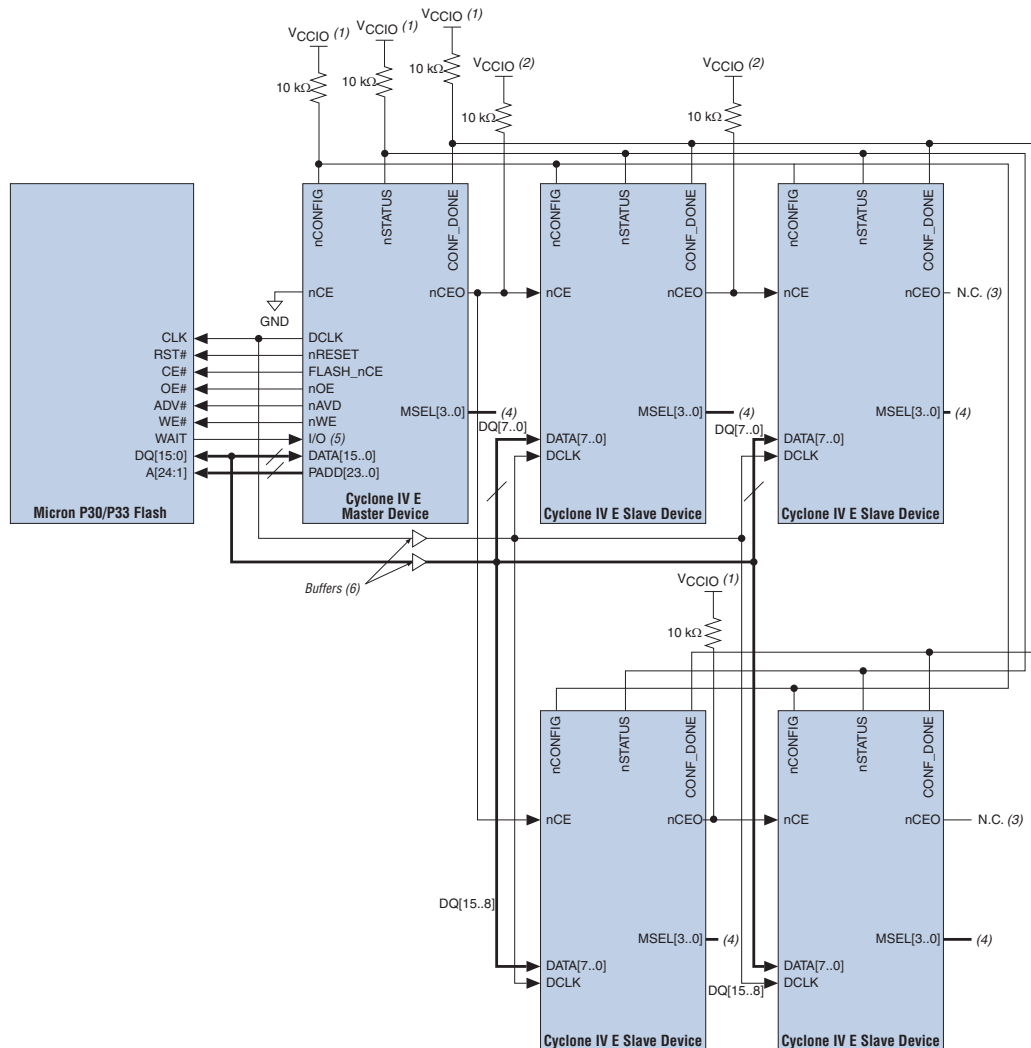**Figure 8–8. Byte-Wide Multi-Device AP Configuration**
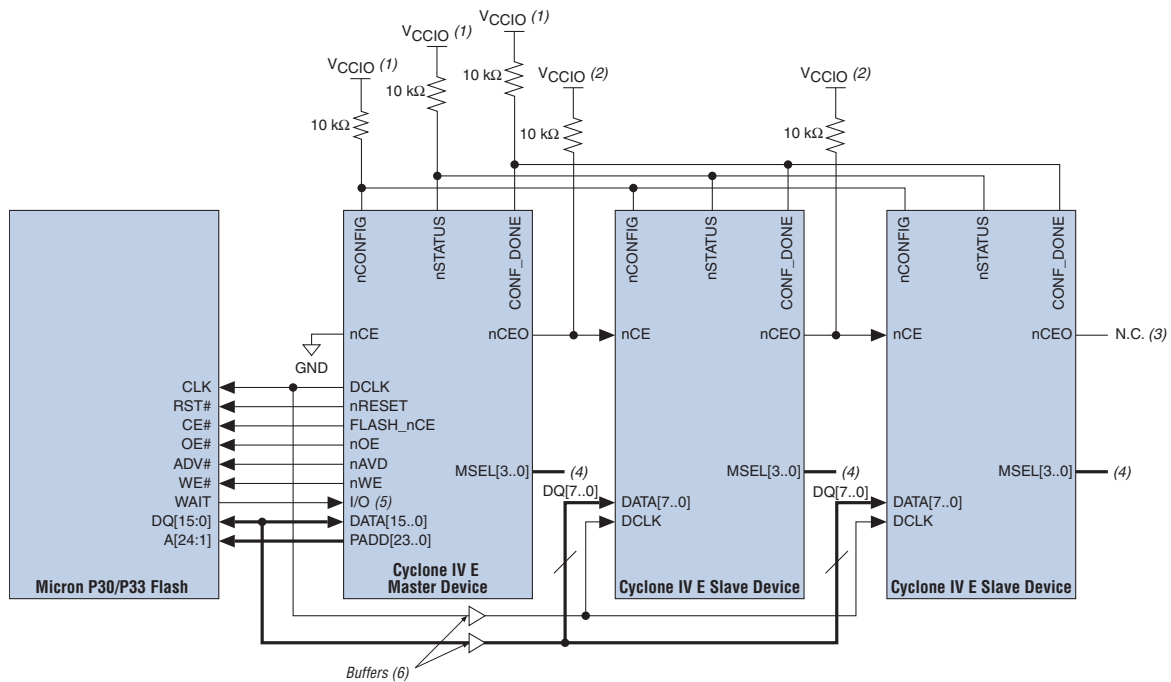
Notes to Figure 8–8:

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

(2) Connect the pull-up resistor to the V_{CCIO} supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device in AP mode and the slave devices in FPP mode. To connect MSEL[3..0] for the master device in AP mode and the slave devices in FPP mode, refer to Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) The AP configuration ignores the WAIT signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

(6) Connect the repeater buffers between the Cyclone IV E master device and slave devices for DATA[15..0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

Word-Wide Multi-Device AP Configuration

The more efficient setup is one in which some of the slave devices are connected to the LSB of the DATA[7..0] and the remaining slave devices are connected to the MSB of the DATA[15..8]. In the word-wide multi-device AP configuration, the nCEO pin of the master device enables two separate daisy chains of slave devices, allowing both chains to be programmed concurrently, as shown in Figure 8–9.

Figure 8–9. Word-Wide Multi-Device AP Configuration

Notes to Figure 8–9:

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

(2) Connect the pull-up resistor to the V_{CCIO} supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device in AP mode and the slave devices in FPP mode. To connect MSEL[3..0] for the master device in AP mode and the slave devices in FPP mode, refer to Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) The AP configuration ignores the WAIT signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O pin to monitor the WAIT signal from the Micron P30 or P33 flash.

(6) Connect the repeater buffers between the Cyclone IV E master device and slave devices for DATA[15..0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

☞ In a multi-device AP configuration, the board trace length between the parallel flash and the master device must follow the recommendations listed in Table 8–11.

The nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors, as shown in Figure 8–8 on page 8–26 and Figure 8–9 on page 8–27. These pins are open-drain bidirectional pins on the devices. When the first device asserts nCEO (after receiving all its configuration data), it releases its CONF_DONE pin. However, the subsequent devices in the chain keep this shared CONF_DONE line low until they receive their configuration data. When all target devices in the chain receive their configuration data and release CONF_DONE, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode.

Guidelines for Connecting Parallel Flash to Cyclone IV E Devices for an AP Interface

For single- and multi-device AP configuration, the board trace length and loading between the supported parallel flash and Cyclone IV E devices must follow the recommendations listed in Table 8–11. These recommendations also apply to an AP configuration with multiple bus masters.

Table 8–11. Maximum Trace Length and Loading for AP Configuration

Cyclone IV E AP Pins	Maximum Board Trace Length from Cyclone IV E Device to Flash Device (inches)	Maximum Board Load (pF)
DCLK	6	15
DATA[15..0]	6	30
PADD[23..0]	6	30
nRESET	6	30
Flash_nCE	6	30
nOE	6	30
nAVD	6	30
nWE	6	30
I/O [1]	6	30

Note to Table 8–11:

(1) The AP configuration ignores the WAIT signal from the flash during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

Configuring With Multiple Bus Masters

Similar to the AS configuration scheme, the AP configuration scheme supports multiple bus masters for the parallel flash. For another master to take control of the AP configuration bus, the master must assert nCONFIG low for at least 500 ns to reset the master Cyclone IV E device and override the weak 10-kΩ pull-down resistor on the nCE pin. This resets the master Cyclone IV E device and causes it to tri-state its AP configuration bus. The other master device then takes control of the AP configuration bus. After the other master device is done, it releases the AP configuration bus, then releases the nCE pin, and finally pulses nCONFIG low to restart the configuration.

In the AP configuration scheme, multiple masters share the parallel flash. Similar to the AS configuration scheme, the bus control is negotiated by the nCE pin.

Figure 8–10 shows the AP configuration with multiple bus masters.

**Figure 8–10. AP Configuration with Multiple Bus Masters**
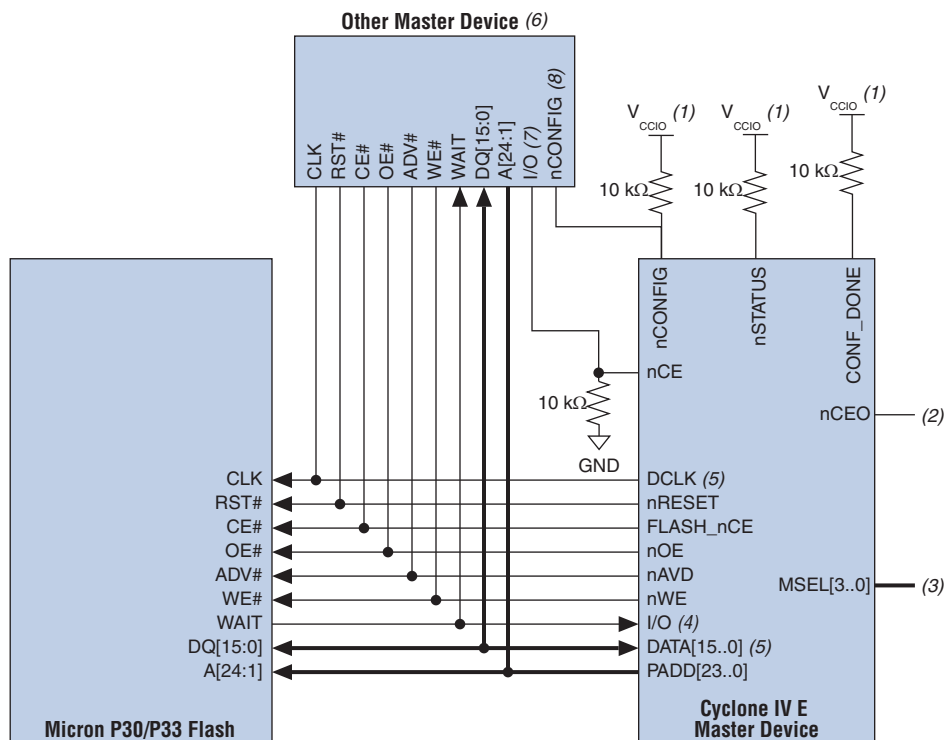
Notes to Figure 8–10:

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

(2)   The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3)   The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL[3..0], refer to Table 8–5 on page 8–9. Connect the MSEL pins directly to V$_{CCA}$ or GND.

(4)   The AP configuration ignores the WAIT signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

(5)   When cascading Cyclone IV E devices in a multi-device AP configuration, connect the repeater buffers between the master device and slave devices for DATA[15..0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(6)   The other master device must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7)   The other master device can control the AP configuration bus by driving the nCE to high with an output high on the I/O pin.

(8)   The other master device can pulse nCONFIG if it is under system control and not tied to V$_{CCIO}$.

Figure 8–11 shows the recommended balanced star routing for multiple bus master interfaces to minimize signal integrity issues.

Figure 8–11. Balanced Star Routing
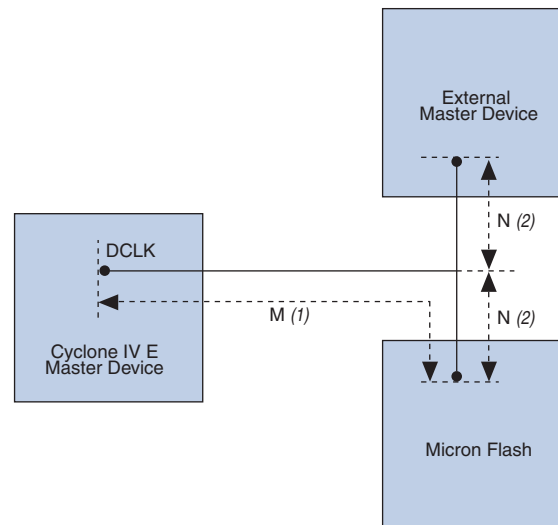
Notes to Figure 8–11:

(1) Altera recommends that *M* does not exceed 6 inches, as listed in Table 8–11 on page 8–28.

(2) Altera recommends using a balanced star routing. Keep the *N* length equal and as short as possible to minimize reflection noise from the transmission line. The *M* length is applicable for this setup.

Estimating AP Configuration Time

AP configuration time is dominated by the time it takes to transfer data from the parallel flash to Cyclone IV E devices. This parallel interface is clocked by the Cyclone IV E DCLK output (generated from an internal oscillator). The DCLK minimum frequency when using the 40-MHz oscillator is 20 MHz (50 ns). In word-wide cascade programming, the DATA[15..0] bus transfers a 16-bit word and essentially cuts configuration time to approximately 1/16 of the AS configuration time. Equation 8–4 and Equation 8–5 show the configuration time calculations.

Equation 8–4.

$$\text{Size} \times \left(\frac{\text{maximum DCLK period}}{16 \text{ bits per DCLK cycle}}\right) = \text{estimated maximum configuration time}$$

Equation 8–5.

$$9{,}600{,}000 \text{ bits} \times \left(\frac{50 \text{ ns}}{16 \text{ bit}}\right) = 30 \text{ ms}$$

Programming Parallel Flash Memories

Supported parallel flash memories are external non-volatile configuration devices. They are industry standard microprocessor flash memories. For more information about the supported families for the commodity parallel flash, refer to Table 8–10 on page 8–22.

Cyclone IV E devices in a single- or multiple-device chain support in-system programming of a parallel flash using the JTAG interface with the flash loader megafunction. The board intelligent host or download cable uses the four JTAG pins on Cyclone IV E devices to program the parallel flash in system, even if the host or download cable cannot access the configuration pins of the parallel flash.

For more information about using the JTAG pins on Cyclone IV E devices to program the parallel flash in-system, refer to *AN 478: Using FPGA-Based Parallel Flash Loader (PFL) with the Quartus II Software*.

In the AP configuration scheme, the default configuration boot address is 0×010000 when represented in 16-bit word addressing in the supported parallel flash memory (Figure 8–12). In the Quartus II software, the default configuration boot address is 0×020000 because it is represented in 8-bit byte addressing. Cyclone IV E devices configure from word address 0×010000, which is equivalent to byte address 0×020000.

The Quartus II software uses byte addressing for the default configuration boot address. You must set the start address field to **0×020000**.

The default configuration boot address allows the system to use special parameter blocks in the flash memory map. Parameter blocks are at the top or bottom of the memory map. Figure 8–12 shows the configuration boot address in the AP configuration scheme. You can change the default configuration default boot address 0×010000 to any desired address using the APFC_BOOT_ADDR JTAG instruction. For more information about the APFC_BOOT_ADDR JTAG instruction, refer to "JTAG Instructions" on page 8–57.

Figure 8–12. Configuration Boot Address in AP Flash Memory Map
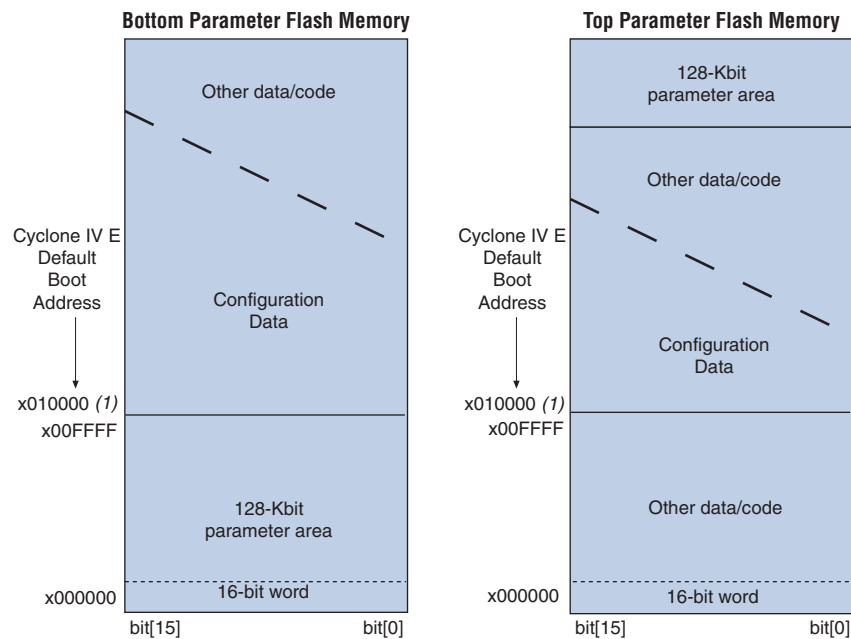
Note to Figure 8–12:

(1)   The default configuration boot address is x010000 when represented in 16-bit word addressing.

PS Configuration

You can perform PS configuration on Cyclone IV devices with an external intelligent host, such as a MAX® II device, microprocessor with flash memory, or a download cable. In the PS scheme, an external host controls the configuration. Configuration data is clocked into the target Cyclone IV device through DATA[0] at each rising edge of DCLK.

If your system already contains a common flash interface (CFI) flash memory, you can use it for Cyclone IV device configuration storage as well. The MAX II PFL feature provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control the configuration from the flash memory device to the Cyclone IV device.

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

☞ Cyclone IV devices do not support enhanced configuration devices for PS configuration.

PS Configuration Using an External Host

In the PS configuration scheme, you can use an intelligent host such as a MAX II device or microprocessor that controls the transfer of configuration data from a storage device, such as flash memory, to the target Cyclone IV device. You can store the configuration data in **.rbf**, **.hex**, or **.ttf** format.

Figure 8–13 shows the configuration interface connections between a Cyclone IV device and an external host device for single-device configuration.

Figure 8–13. Single-Device PS Configuration Using an External Host
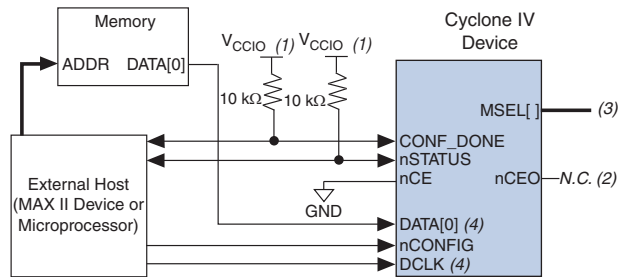
Notes to Figure 8–13:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

To begin the configuration, the external host device must generate a low-to-high transition on the nCONFIG pin. When nSTATUS is pulled high, the external host device must place the configuration data one bit at a time on DATA[0]. If you use configuration data in **.rbf**, **.ttf**, or **.hex**, you must first send the LSB of each data byte. For example, if the **.rbf** contains the byte sequence 02 1B EE 01 FA, the serial bitstream you must send to the device is:

0100-0000 1101-1000 0111-0111 1000-0000 0101-1111

Cyclone IV devices receive configuration data on DATA[0] and the clock is received on DCLK. Data is latched into the device on the rising edge of DCLK. Data is continuously clocked into the target device until CONF_DONE goes high and the device enters initialization state.

☞ Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device.

INIT_DONE is released and pulled high when initialization is complete. The external host device must be able to detect this low-to-high transition which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

To ensure DCLK and DATA[0] are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. In the PS scheme, the DATA[0] pin is tri-stated by default in user mode and must be driven by the external host device. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified system frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

The external host device can also monitor CONF_DONE and INIT_DONE to ensure successful configuration. The CONF_DONE pin must be monitored by the external device to detect errors and to determine when programming is complete. If all configuration data is sent, but CONF_DONE or INIT_DONE has not gone high, the external device must reconfigure the target device.

Figure 8–14 shows how to configure multiple devices using an external host device. This circuit is similar to the PS configuration circuit for a single device, except that Cyclone IV devices are cascaded for multi-device configuration.

Figure 8–14. Multi-Device PS Configuration Using an External Host
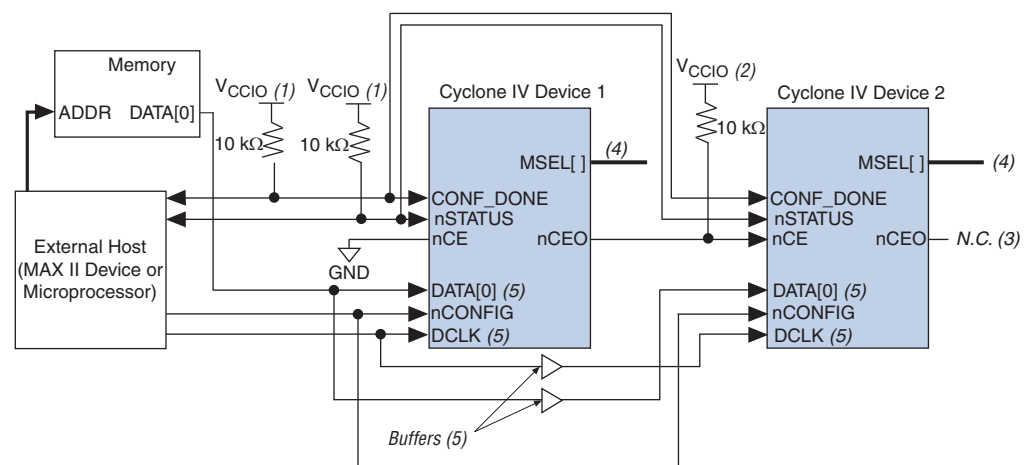
Notes to Figure 8–14:

(1) The pull-up resistor must be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) Connect the pull-up resistor to the V_{CCIO} supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle. Therefore, the transfer of data destinations is transparent to the external host device. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain. To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that DCLK and DATA lines are buffered. All devices initialize and enter user mode at the same time because all CONF_DONE pins are tied together.

If any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain because all nSTATUS and CONF_DONE pins are tied together. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

You can have multiple devices that contain the same configuration data in your system. To support this configuration scheme, all device nCE inputs are tied to GND, while the nCEO pins are left floating. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain. To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that the DCLK and DATA lines are buffered. Devices must be of the same density and package. All devices start and complete configuration at the same time.

Figure 8–15 shows a multi-device PS configuration when both Cyclone IV devices are receiving the same configuration data.

Figure 8–15. Multi-Device PS Configuration When Both Devices Receive the Same Data
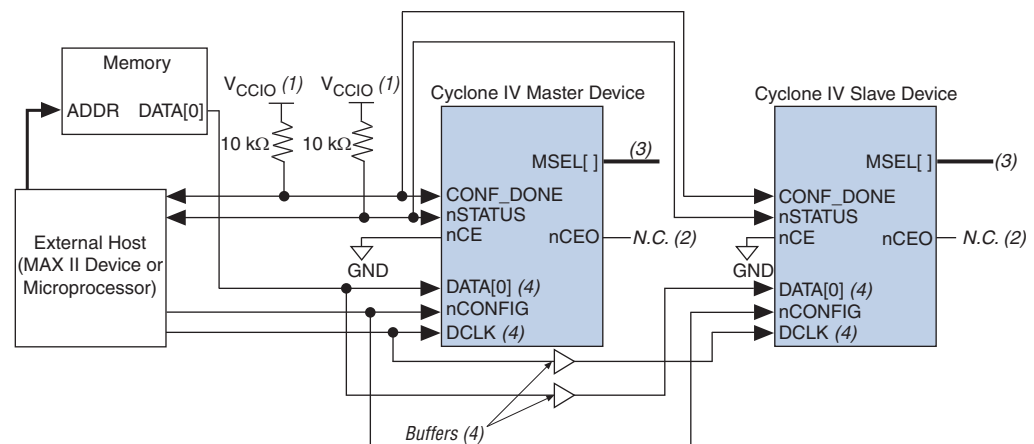
Notes to Figure 8–15:

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) The nCEO pins of both devices are left unconnected or used as user I/O pins when configuring the same configuration data into multiple devices.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

PS Configuration Timing

A PS configuration must meet the setup and hold timing parameters and the maximum clock frequency. When using a microprocessor or another intelligent host to control the PS interface, ensure that you meet these timing requirements. Figure 8–16 shows the timing waveform for PS configuration when using an external host device.

Figure 8–16. PS Configuration Timing Waveform [1]
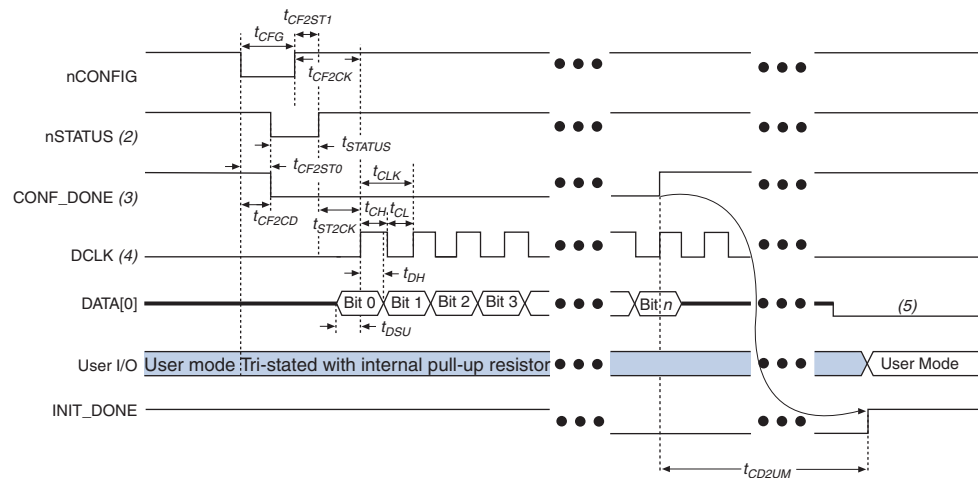
Notes to Figure 8–16:

(1) The beginning of this waveform shows the device in user mode. In user mode, `nCONFIG`, `nSTATUS`, and `CONF_DONE` are at logic-high levels. When `nCONFIG` is pulled low, a reconfiguration cycle begins.

(2) After power up, the Cyclone IV device holds `nSTATUS` low during POR delay.

(3) After power up, before and during configuration, `CONF_DONE` is low.

(4) In user mode, drive `DCLK` either high or low when using the PS configuration scheme, whichever is more convenient. When using the AS configuration scheme, `DCLK` is a Cyclone IV device output pin and must not be driven externally.

(5) Do not leave the `DATA[0]` pin floating after configuration. Drive the `DATA[0]` pin high or low, whichever is more convenient.

Table 8–12 lists the PS configuration timing parameters for Cyclone IV devices.

Table 8–12. PS Configuration Timing Parameters For Cyclone IV Devices (Part 1 of 2)

Symbol	Parameter	Minimum		Maximum		Unit
		Cyclone IV [1]	Cyclone IV E [2]	Cyclone IV [1]	Cyclone IV E [2]	
t_{CF2CD}	`nCONFIG` low to `CONF_DONE` low	—		500		ns
t_{CF2ST0}	`nCONFIG` low to `nSTATUS` low	—		500		ns
t_{CFG}	`nCONFIG` low pulse width	500		—		ns
t_{STATUS}	`nSTATUS` low pulse width	45		230 [3]		μs

Table 8–12. PS Configuration Timing Parameters For Cyclone IV Devices (Part 2 of 2)

Symbol	Parameter	Minimum		Maximum		Unit
		Cyclone IV [1]	Cyclone IV E [2]	Cyclone IV [1]	Cyclone IV E [2]	
t_{CF2ST1}	nCONFIG high to nSTATUS high	—		230 [4]		μs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	230 [3]		—		μs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2		—		μs
t_{DH}	Data hold time after rising edge on DCLK	0		—		ns
t_{CD2UM}	CONF_DONE high to user mode [5]	300		650		μs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period		—		—
t_{CD2UMC}	CONF_DONE high to user mode with **CLKUSR** option on	t_{CD2CU} + (3,192 × CLKUSR period)		—		—
t_{DSU}	Data setup time before rising edge on DCLK	5	8	—	—	ns
t_{CH}	DCLK high time	3.2	6.4	—	—	ns
t_{CL}	DCLK low time	3.2	6.4	—	—	ns
t_{CLK}	DCLK period	7.5	15	—	—	ns
f_{MAX}	DCLK frequency [6]	—	—	133	66	MHz

Notes to Table 8–12:

(1) Applicable for Cyclone IV GX and Cyclone IV E devices with 1.2-V core voltage.

(2) Applicable for Cyclone IV E devices with 1.0-V core voltage.

(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.

(5) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.

(6) Cyclone IV E devices with 1.0-V core voltage have slower F_{MAX} when compared with Cyclone IV GX devices with 1.2-V core voltage.

PS Configuration Using a Download Cable

In this section, the generic term "download cable" includes the Altera USB-Blaster USB port download cable, MasterBlaster™ serial and USB communications cable, ByteBlaster II parallel port download cable, the ByteBlasterMV™ parallel port download cable, and the EthernetBlaster communications cable.

In the PS configuration with a download cable, an intelligent host (such as a PC) transfers data from a storage device to the Cyclone IV device through the download cable.

The programming hardware or download cable then places the configuration data one bit at a time on the DATA[0] pin of the device. The configuration data is clocked into the target device until CONF_DONE goes high. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

When you use a download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software if an error occurs. Additionally, the **Enable user-supplied start-up clock (CLKUSR)** option has no effect on device initialization, because this option is disabled in the **.sof** when programming the device with the Quartus II Programmer and download cable. Therefore, if you turn on the **CLKUSR** option, you do not have to provide a clock on CLKUSR when you configure the device with the Quartus II Programmer and a download cable.

Figure 8–17 shows PS configuration for Cyclone IV devices with a download cable.

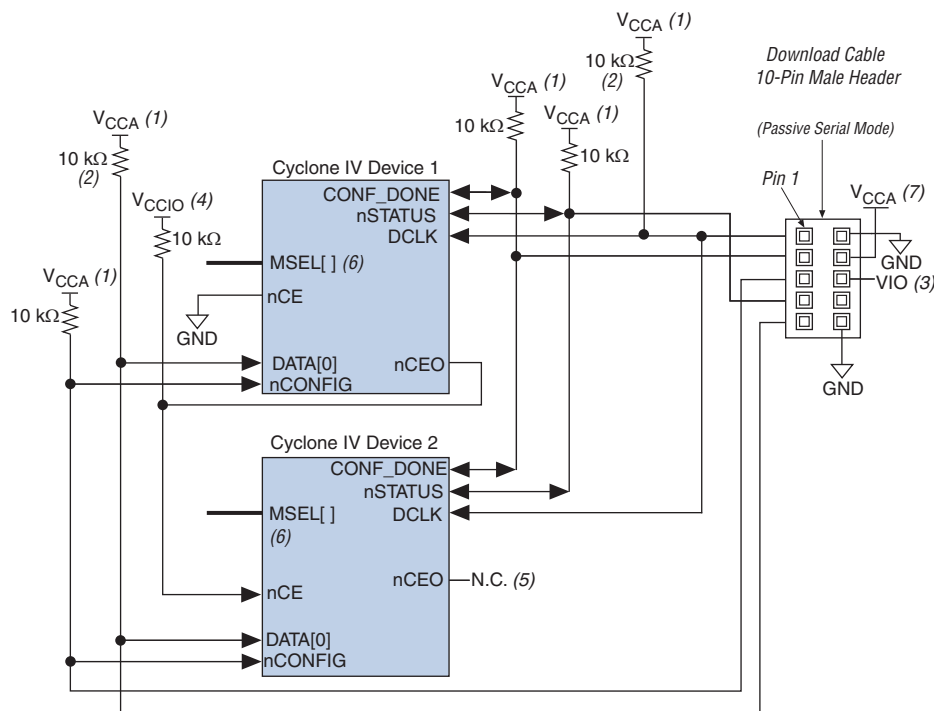**Figure 8–17. PS Configuration Using a Download Cable**
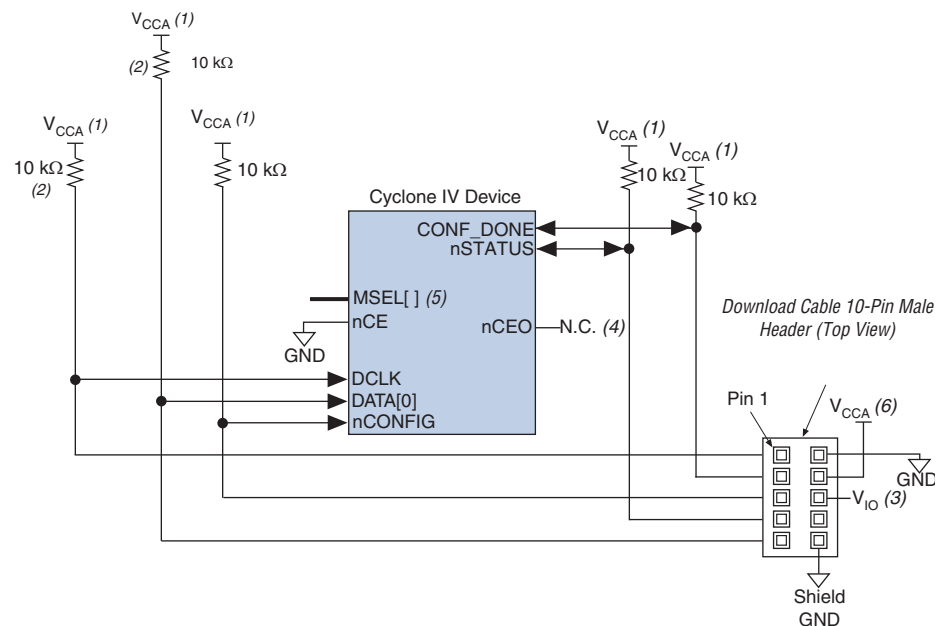
Notes to Figure 8–17:

(1) You must connect the pull-up resistor to the same supply voltage as the V_{CCA} supply.

(2) The pull-up resistors on DATA[0] and DCLK are only required if the download cable is the only configuration scheme used on your board. This is to ensure that DATA[0] and DCLK are not left floating after configuration. For example, if you also use a configuration device, the pull-up resistors on DATA[0] and DCLK are not required.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. With the USB-Blaster, ByteBlaster II, ByteBlaster MV, and EthernetBlaster, this pin is a no connect.

(4) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(5) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9 for PS configuration schemes. Connect the MSEL pins directly to V_{CCA} or GND.

(6) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from V_{CCA}. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

You can use a download cable to configure multiple Cyclone IV device configuration pins. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE are connected to every device in the chain. All devices in the chain utilize and enter user mode at the same time because all CONF_DONE pins are tied together.

In addition, the entire chain halts configuration if any device detects an error because the nSTATUS pins are tied together. Figure 8–18 shows the PS configuration for multiple Cyclone IV devices using a MasterBlaster, USB-Blaster, ByteBlaster II, or ByteBlasterMV cable.

Figure 8–18. Multi-Device PS Configuration Using a Download Cable

Notes to Figure 8–18:

(1) You must connect the pull-up resistor to the same supply voltage as the V_CCA supply.

(2) The pull-up resistors on DATA[0] and DCLK are only required if the download cable is the only configuration scheme used on your board. This ensures that DATA[0] and DCLK are not left floating after configuration. For example, if you also use a configuration device, the pull-up resistors on DATA[0] and DCLK are not required.

(3) Pin 6 of the header is a V_IO reference voltage for the MasterBlaster output driver. V_IO must match the V_CCA of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using USB-Blaster, ByteBlaster II, and EthernetBlaster cables, this pin is connected to nCE when it is used for AS programming. Otherwise, it is a no connect.

(4) Connect the pull-up resistor to the V_CCIO supply voltage of the I/O bank in which the nCE pin resides.

(5) The nCEO pin of the last device in the chain is left unconnected or used as a user I/O pin.

(6) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for PS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_CCA or GND.

(7) Power up the V_CC of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5 V supply from V_CCA. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_CC power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

FPP Configuration

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

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

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

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

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

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

FPP Configuration Using an External Host

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

the device, must be stored in the external host device. Figure 8–19 shows the configuration interface connections between the Cyclone IV devices and an external device for single-device configuration.

Figure 8–19. Single-Device FPP Configuration Using an External Host
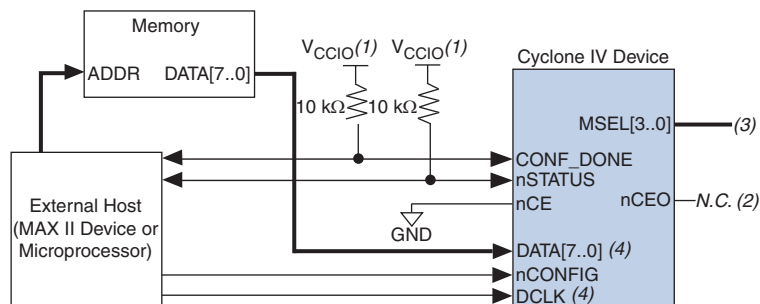
Notes to Figure 8–19:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–4 on page 8–8 and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[7..0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the external host device places the configuration data one byte at a time on the DATA[7..0] pins.

Cyclone IV devices receive configuration data on the DATA[7..0] pins and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. Data is continuously clocked into the target device until CONF_DONE goes high. The CONF_DONE pin goes high one byte early in FPP configuration mode. The last byte is required for serial configuration (AS and PS) modes.

☞ Two DCLK falling edges are required after CONF_DONE goes high to begin initialization of the device.

Supplying a clock on CLKUSR does not affect the configuration process. After the CONF_DONE pin goes high, CLKUSR is enabled after the time specified as t_{CD2CU}. After this time period elapses, Cyclone IV devices require 3,192 clock cycles to initialize properly and enter user mode. For more information about the supported CLKUSR f_{MAX} value for Cyclone IV devices, refer to Table 8–13 on page 8–44.

The INIT_DONE pin is released and pulled high when initialization is complete. The external host device must be able to detect this low-to-high transition, which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

To ensure that DCLK and DATA[0] are not left floating at the end of the configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When you choose the FPP scheme in the Quartus II software, the DATA[0] pin is tri-stated by default in user mode and must be driven by the external host device. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The DCLK speed must be below the specified system frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

The external host device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The CONF_DONE pin must be monitored by the external device to detect errors and to determine when programming is complete. If all configuration data is sent, but CONF_DONE or INIT_DONE has not gone high, the external device must reconfigure the target device.

Figure 8–20 shows how to configure multiple devices with a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the Cyclone IV devices are cascaded for multi-device configuration.

Figure 8–20. Multi-Device FPP Configuration Using an External Host
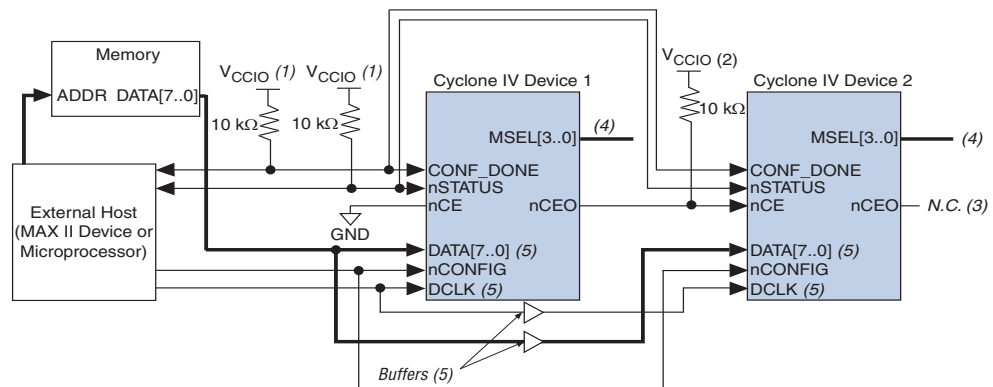
Notes to Figure 8–20:

(1) The pull-up resistor must be connected to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) Connect the pull-up resistor to the V_{CCIO} supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–4 on page 8–8 and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[7..0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS,

DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered. All devices initialize and enter user mode at the same time, because all device CONF_DONE pins are tied together.

All nSTATUS and CONF_DONE pins are tied together and if any device detects an error, configuration stops for the entire chain and the entire chain must be reconfigured. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

Figure 8–21 shows multi-device FPP configuration when both Cyclone IV devices are receiving the same configuration data. Configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered. Devices must be of the same density and package. All devices start and complete configuration at the same time.

Figure 8–21. Multi-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data
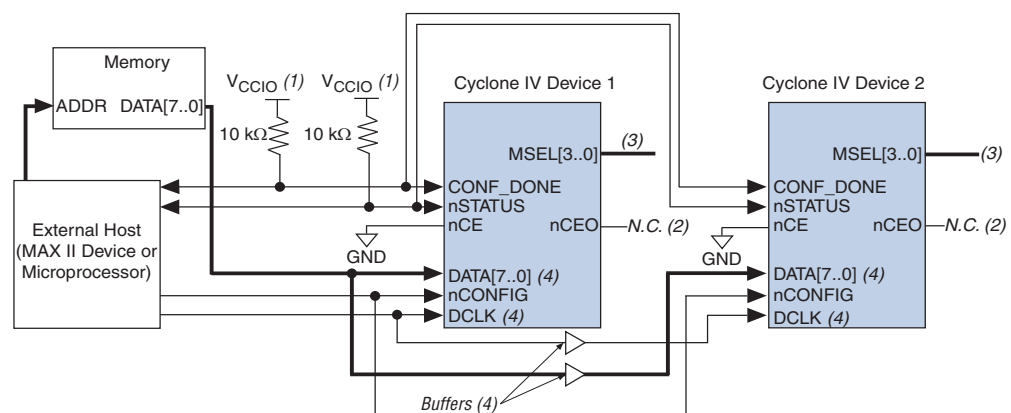
Notes to Figure 8–21:

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. V_{CC} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host.

(2) The nCEO pins of both devices are left unconnected or used as user I/O pins when configuring the same configuration data into multiple devices.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–4 on page 8–8 and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[7..0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

You can use a single configuration chain to configure Cyclone IV devices with other Altera devices that support FPP configuration. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device starts reconfiguration in all devices, tie all the CONF_DONE and nSTATUS pins together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to *Configuring Mixed Altera FPGA Chains* in volume 2 of the *Configuration Handbook*.

FPP Configuration Timing

Figure 8–22 shows the timing waveform for the FPP configuration when using an external host.

Figure 8–22. FPP Configuration Timing Waveform *(1)*
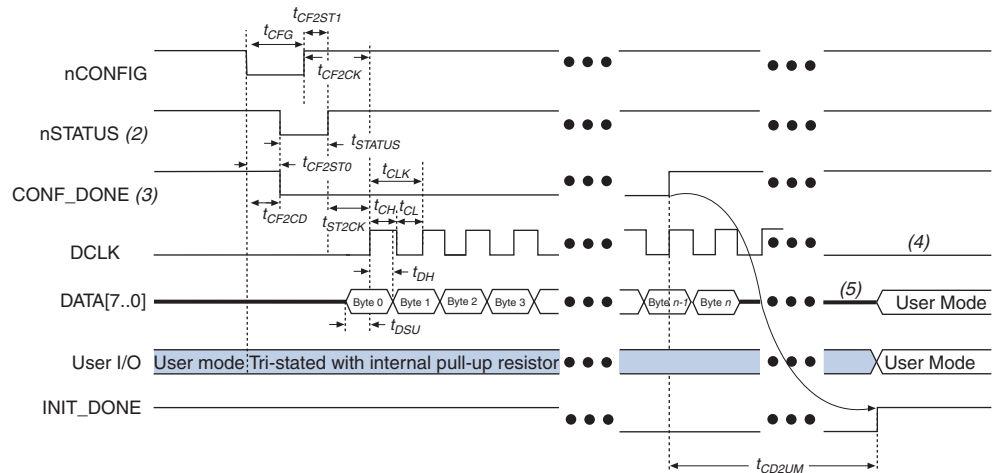
Notes to Figure 8–22:

(1) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic-high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(2) After power up, the Cyclone IV device holds nSTATUS low during POR delay.

(3) After power up, before and during configuration, CONF_DONE is low.

(4) Do not leave DCLK floating after configuration. It must be driven high or low, whichever is more convenient.

(5) DATA[7..0] is available as a user I/O pin after configuration; the state of the pin depends on the dual-purpose pin settings.

Table 8–13 lists the FPP configuration timing parameters for Cyclone IV devices.

Table 8–13. FPP Timing Parameters for Cyclone IV Devices   (Part 1 of 2)

Symbol	Parameter	Minimum		Maximum		Unit
		Cyclone IV *(1)*	Cyclone IV E *(2)*	Cyclone IV *(1)*	Cyclone IV E *(2)*	
t_{CF2CD}	nCONFIG low to CONF_DONE low	—		500		ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—		500		ns
t_{CFG}	nCONFIG low pulse width	500		—		ns
t_{STATUS}	nSTATUS low pulse width	45		230 *(3)*		μs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—		230 *(4)*		μs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	230 *(3)*		—		μs

Table 8–13. FPP Timing Parameters for Cyclone IV Devices   (Part 2 of 2)

Symbol	Parameter	Minimum		Maximum		Unit
		Cyclone IV [1]	Cyclone IV E [2]	Cyclone IV [1]	Cyclone IV E [2]	
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2		—		μs
t_{DH}	Data hold time after rising edge on DCLK	0		—		ns
t_{CD2UM}	CONF_DONE high to user mode [5]	300		650		μs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period		—		—
t_{CD2UMC}	CONF_DONE high to user mode with **CLKUSR** option on	t_{CD2CU} + (3,192 × CLKUSR period)		—		—
t_{DSU}	Data setup time before rising edge on DCLK	5	8	—	—	ns
t_{CH}	DCLK high time	3.2	6.4	—	—	ns
t_{CL}	DCLK low time	3.2	6.4	—	—	ns
t_{CLK}	DCLK period	7.5	15	—	—	ns
f_{MAX}	DCLK frequency [6]	—	—	133	66	MHz

Notes to Table 8–13:

(1) Applicable for Cyclone IV GX and Cyclone IV E with 1.2-V core voltage.

(2) Applicable for Cyclone IV E with 1.0-V core voltage.

(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.

(5) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.

(6) Cyclone IV E devices with 1.0-V core voltage have slower F_{MAX} when compared with Cyclone IV GX devices with 1.2-V core voltage.

JTAG Configuration

JTAG has developed a specification for boundary-scan testing (BST). The BST architecture offers the capability to efficiently test components on PCBs with tight lead spacing. The BST architecture can test pin connections without using physical test probes and capture functional data while a device is normally operating. You can also use the JTAG circuitry to shift configuration data into the device. The Quartus II software automatically generates **.sof** for JTAG configuration with a download cable in the Quartus II software Programmer.

For more information about the JTAG boundary-scan testing, refer to the *JTAG Boundary-Scan Testing for Cyclone IV Devices* chapter.

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

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

Table 8–14. Dedicated JTAG Pins

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

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

For device using V_{CCIO} of 2.5, 3.0, and 3.3 V, refer to Figure 8–23. All I/O inputs must maintain a maximum AC voltage of 4.1 V because JTAG pins do not have the internal PCI clamping diodes to prevent voltage overshoot when using V_{CCIO} of 2.5, 3.0, and 3.3 V. You must power up the V_{CC} of the download cable with a 2.5-V supply from V_{CCA}. For device using V_{CCIO} of 1.2, 1.5, and 1.8 V, refer to Figure 8–24. You can power up the V_{CC} of the download cable with the supply from V_{CCIO}.

Figure 8–23. JTAG Configuration of a Single Device Using a Download Cable (2.5, 3.0, and 3.3-V V_{CCIO} Powering the JTAG Pins)
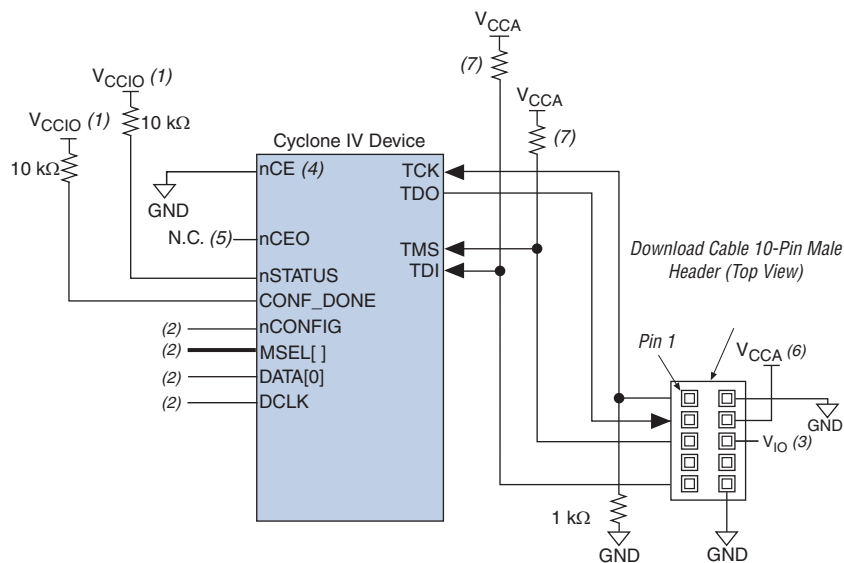
Notes to Figure 8–23:

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

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

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCA}. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the USB-Blaster, ByteBlaster II, ByteBlasterMV, and EthernetBlaster cables, this pin is a no connect.

(4) The `nCE` pin must be connected to GND or driven low for successful JTAG configuration.

(5) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(6) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from V_{CCA}. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

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

Figure 8–24. JTAG Configuration of a Single Device Using a Download Cable (1.5-V or 1.8-V V_CCIO Powering the JTAG Pins)
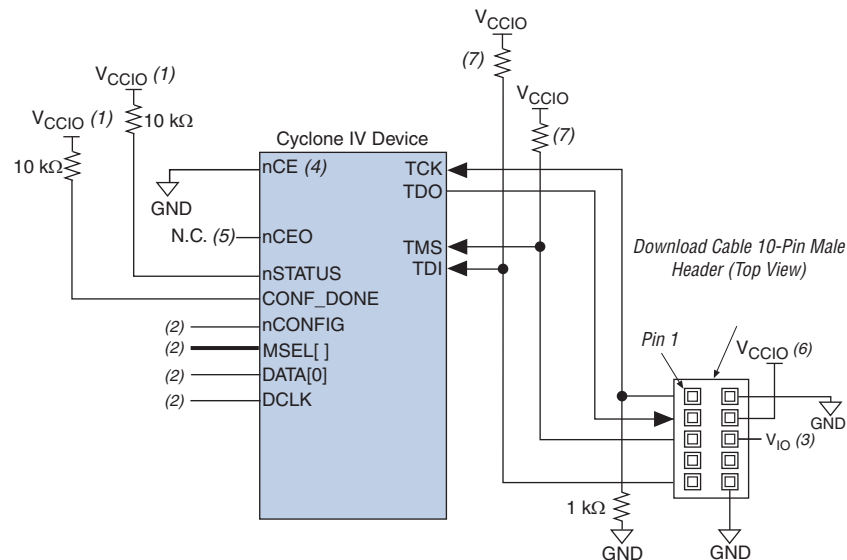
Notes to Figure 8–24:

(1) Connect these pull-up resistors to the V_CCIO supply of the bank in which the pin resides.

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

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

(4) The nCE must be connected to GND or driven low for successful JTAG configuration.

(5) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(6) Power up the V_CC of the EthernetBlaster, ByteBlaster II or USB-Blaster cable with supply from V_CCIO. The Ethernet-Blaster, ByteBlaster II, and USB-Blaster cables do not support a target supply voltage of 1.2 V. For the target supply voltage value, refer to the *ByteBlaster II Download Cable User Guide*, the *USB-Blaster Download Cable User Guide*, and the *EthernetBlaster Communications Cable User Guide*.

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

To configure a single device in a JTAG chain, the programming software places all other devices in bypass mode. In bypass mode, devices pass programming data from the TDI pin to the TDO pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the TDO pin one clock cycle later.

The Quartus II software verifies successful JTAG configuration after completion. At the end of configuration, the software checks the state of CONF_DONE through the JTAG port. When Quartus II generates a **.jam** for a multi-device chain, it contains instructions so that all the devices in the chain are initialized at the same time. If CONF_DONE is not high, the Quartus II software indicates that configuration has failed. If CONF_DONE is high, the software indicates that configuration was successful. After the configuration bitstream is serially sent using the JTAG TDI port, the TCK port clocks an additional clock cycles to perform device initialization.

You can perform JTAG testing on Cyclone IV devices before, during, and after configuration. Cyclone IV devices support the BYPASS, IDCODE, and SAMPLE instructions during configuration without interrupting configuration. All other JTAG instructions can only be issued by first interrupting configuration and reprogramming I/O pins with the ACTIVE_DISENGAGE and CONFIG_IO instructions.

The CONFIG_IO instruction allows you to configure the I/O buffers through the JTAG port and interrupts configuration when issued after the ACTIVE_DISENGAGE instruction. This instruction allows you to perform board-level testing prior to configuring the Cyclone IV device or waiting for a configuration device to complete configuration. Prior to issuing the CONFIG_IO instruction, you must issue the ACTIVE_DISENGAGE instruction. This is because in Cyclone IV devices, the CONFIG_IO instruction does not hold nSTATUS low until reconfiguration, so you must disengage the active configuration mode controller when active configuration is interrupted. The ACTIVE_DISENGAGE instruction places the active configuration mode controllers in an idle state prior to JTAG programming. Additionally, the ACTIVE_ENGAGE instruction allows you to re-engage a disengaged active configuration mode controller.

☞ You must follow a specific flow when executing the ACTIVE_DISENGAGE, CONFIG_IO, and ACTIVE_ENGAGE JTAG instructions in Cyclone IV devices.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins in Cyclone IV devices do not affect JTAG boundary-scan or programming operations. Toggling these pins do not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration of Cyclone IV devices, consider the dedicated configuration pins. Table 8–15 describes how you must connect these pins during JTAG configuration.

Table 8–15. Dedicated Configuration Pin Connections During JTAG Configuration

Signal	Description
nCE	On all Cyclone IV devices in the chain, nCE must be driven low by connecting it to GND, pulling it low through a resistor, or driving it by some control circuitry. For devices that are also in multi-device AS, AP, PS, or FPP configuration chains, you must connect the nCE pins to GND during JTAG configuration or JTAG configured in the same order as the configuration chain.
nCEO	On all Cyclone IV devices in the chain, nCEO is left floating or connected to the nCE of the next device.
MSEL	These pins must not be left floating. These pins support whichever non-JTAG configuration that you used in production. If you only use JTAG configuration, tie these pins to GND.
nCONFIG	Driven high by connecting to the V_{CCIO} supply of the bank in which the pin resides and pulling up through a resistor or driven high by some control circuitry.
nSTATUS	Pull to the V_{CCIO} supply of the bank in which the pin resides through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin must be pulled up to the V_{CCIO} individually.
CONF_DONE	Pull to the V_{CCIO} supply of the bank in which the pin resides through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin must be pulled up to V_{CCIO} supply of the bank in which the pin resides individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration.
DCLK	Must not be left floating. Drive low or high, whichever is more convenient on your board.

When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer.

JTAG-chain device programming is ideal when the system contains multiple devices, or when testing your system with JTAG BST circuitry. Figure 8–25 and Figure 8–26 show multi-device JTAG configuration.

For devices using 2.5-, 3.0-, and 3.3-V V_{CCIO} supply, you must refer to Figure 8–25. All I/O inputs must maintain a maximum AC voltage of 4.1 V because JTAG pins do not have the internal PCI clamping diodes to prevent voltage overshoot when using 2.5-, 3.0-, and 3.3- V V_{CCIO} supply. You must power up the V_{CC} of the download cable with a 2.5-V V_{CCA} supply. For device using V_{CCIO} of 1.2, 1.5 V, and 1.8 V, refer to Figure 8–26. You can power up the V_{CC} of the download cable with the supply from V_{CCIO}.

Figure 8–25. JTAG Configuration of Multiple Devices Using a Download Cable (2.5, 3.0, and 3.3-V V_{CCIO} Powering the JTAG Pins)
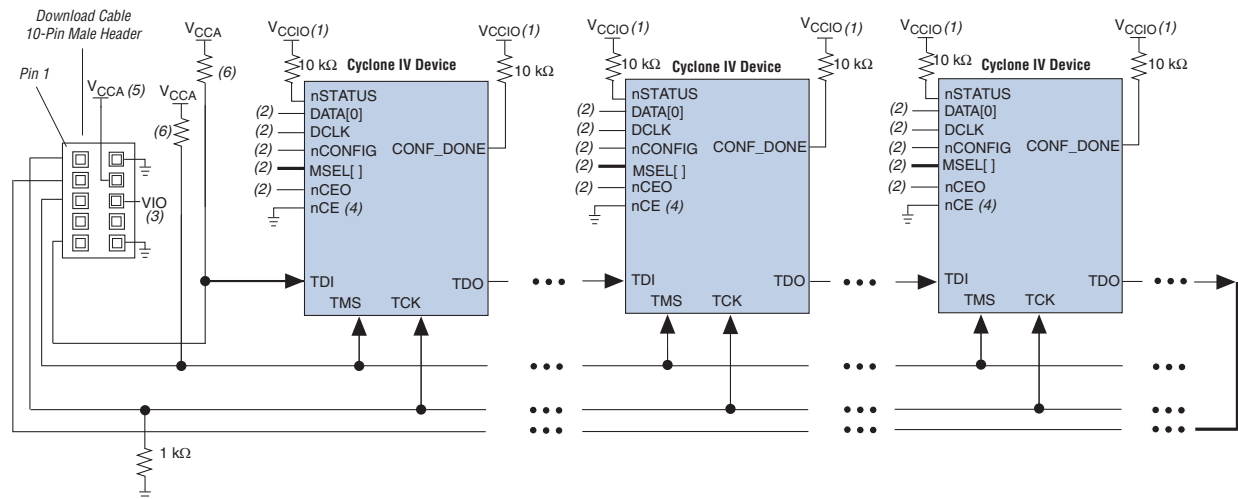
Notes to Figure 8–25:

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

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

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the ByteBlasterMV cable, this pin is a no connect. In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

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

(5) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from V_{CCA}. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.
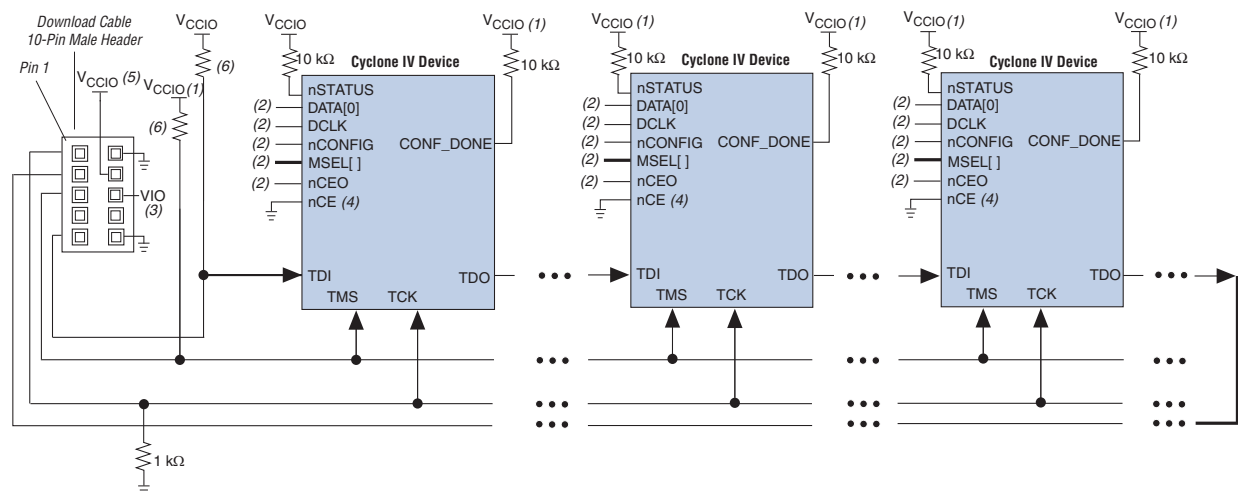
(6) Resistor value can vary from 1 kΩ to 10 kΩ.

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

Notes to Figure 8–26:

(1) Connect these pull-up resistors to the V_CCIO supply of the bank in which the pin resides.

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

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

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

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

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

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

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

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 8–25 or Figure 8–26, in which each of the CONF_DONE and nSTATUS signals are isolated so that each device can enter user mode individually.

After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, ensure that the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the nCEO of the previous device drives the nCE pin of the next device low when it has successfully been JTAG configured. You can place other Altera devices that have JTAG support in the same JTAG chain for device programming and configuration.

☞ JTAG configuration allows an unlimited number of Cyclone IV devices to be cascaded in a JTAG chain.

📍 For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

Figure 8–27 shows JTAG configuration with a Cyclone IV device and a microprocessor.

Figure 8–27. JTAG Configuration of a Single Device Using a Microprocessor
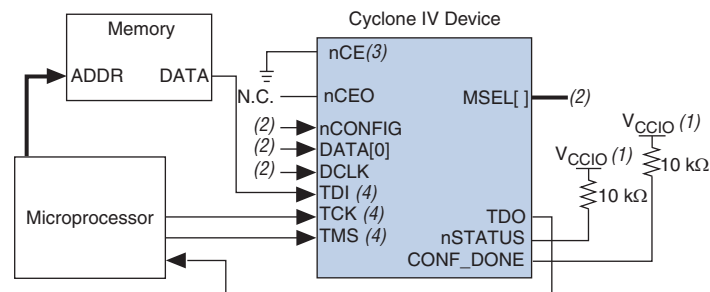
Notes to Figure 8–27:

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain.

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

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

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. Signals driving into TDI, TMS, and TCK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

Configuring Cyclone IV Devices with Jam STAPL

Jam™ STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface. Jam STAPL is a freely licensed open standard. The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

📍 For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the Jam Player, visit the Altera website (www.altera.com).

Configuring Cyclone IV Devices with the JRunner Software Driver

The JRunner software driver allows you to configure Cyclone IV devices through the ByteBlaster II or ByteBlasterMV cables in JTAG mode. The supported programming input file is in **.rbf** format. The JRunner software driver also requires a Chain Description File (**.cdf**) generated by the Quartus II software. The JRunner software driver is targeted for embedded JTAG configuration. The source code is developed for the Windows NT operating system (OS). You can customize the code to make it run on your embedded platform.

☞ The **.rbf** used by the JRunner software driver cannot be a compressed **.rbf** because the JRunner software driver uses JTAG-based configuration. During JTAG-based configuration, the real-time decompression feature is not available.

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

Combining JTAG and AS Configuration Schemes

You can combine the AS configuration scheme with the JTAG-based configuration (Figure 8–28). This setup uses two 10-pin download cable headers on the board. One download cable is used in JTAG mode to configure the Cyclone IV device directly through the JTAG interface. The other download cable is used in AS mode to program the serial configuration device in-system through the AS programming interface. If you try configuring the device using both schemes simultaneously, JTAG configuration takes precedence and AS configuration terminates.

Figure 8–28. Combining JTAG and AS Configuration Schemes
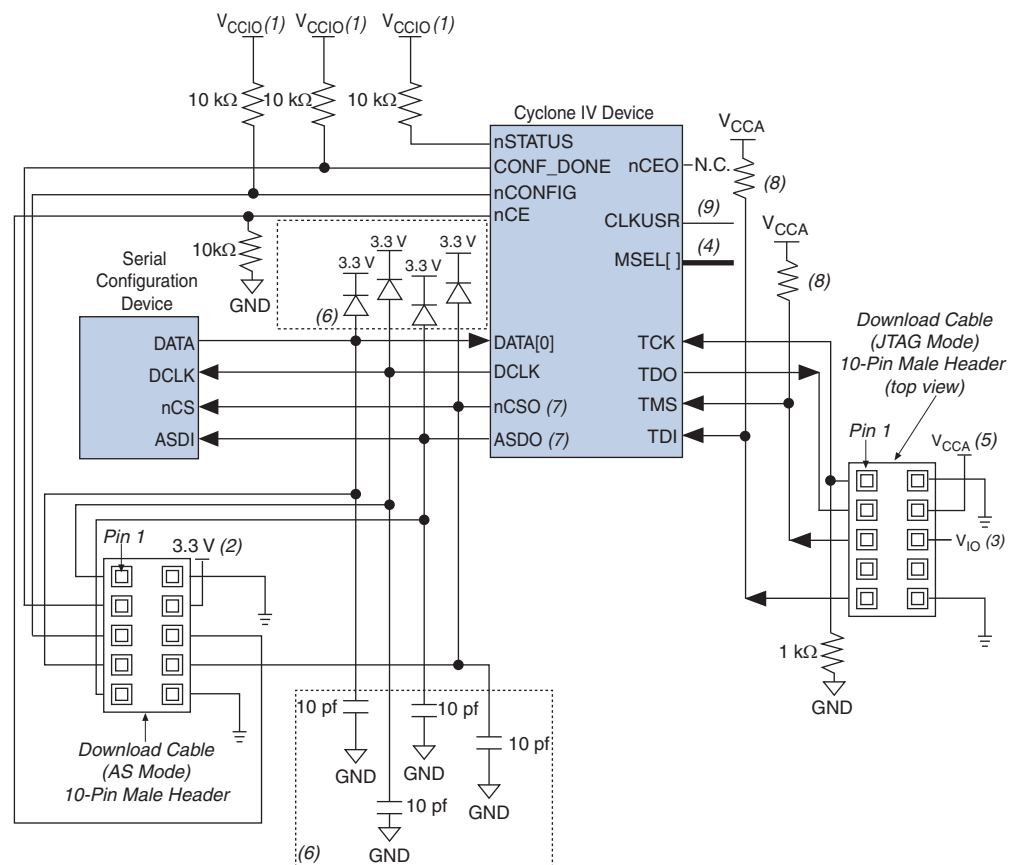
Notes to Figure 8–28:

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

(2) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, or USB-Blaster cable with the 3.3-V supply.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. The V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for AS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(5) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V V_{CCA} supply. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(6) You must place the diodes and capacitors as close as possible to the Cyclone IV device. Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes, for effective voltage clamping.

(7) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

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

(9) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

Programming Serial Configuration Devices In-System with the JTAG Interface

Cyclone IV devices in a single- or multiple-device chain support in-system programming of a serial configuration device with the JTAG interface through the SFL design. The intelligent host or download cable of the board can use the four JTAG pins on the Cyclone IV device to program the serial configuration device in system, even if the host or download cable cannot access the configuration pins (DCLK, DATA, ASDI, and nCS pins).

The SFL design is a JTAG-based in-system programming solution for Altera serial configuration devices. The SFL is a bridge design for the Cyclone IV device that uses their JTAG interface to access the EPCS JTAG Indirect Configuration Device Programming (**.jic**) file and then uses the AS interface to program the EPCS device. Both the JTAG interface and AS interface are bridged together inside the SFL design.

In a multiple device chain, you must only configure the master device that controls the serial configuration device. Slave devices in the multiple device chain that are configured by the serial configuration device do not have to be configured when using this feature. To successfully use this feature, set the MSEL pins of the master device to select the AS configuration scheme (Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9). The serial configuration device in-system programming through the Cyclone IV device JTAG interface has three stages, which are described in the following sections:

■ "Loading the SFL Design"

■ "ISP of the Configuration Device" on page 8–56

■ "Reconfiguration" on page 8–57

Loading the SFL Design

The SFL design is a design inside the Cyclone IV device that bridges the JTAG interface and AS interface with glue logic.

The intelligent host uses the JTAG interface to configure the master device with a SFL design. The SFL design allows the master device to control the access of four serial configuration device pins, also known as the Active Serial Memory Interface (ASMI) pins, through the JTAG interface. The ASMI pins are serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and active-low chip select (nCS) pins.

If you configure a master device with an SFL design, the master device enters user mode even though the slave devices in the multiple device chain are not being configured. The master device enters user mode with a SFL design even though the CONF_DONE signal is externally held low by the other slave devices in chain. Figure 8–29 shows the JTAG configuration of a single Cyclone IV device with a SFL design.

Figure 8–29. Programming Serial Configuration Devices In-System Using the JTAG Interface
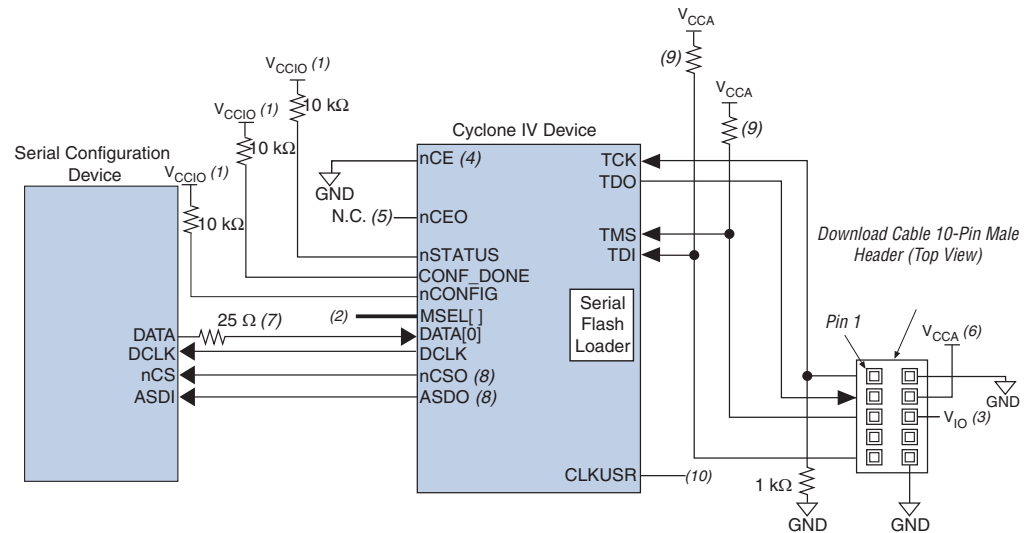
Notes to Figure 8–29:

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

(2) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for AS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to V_{CCA} or GND.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. The V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using USB-Blaster, ByteBlaster II, and EthernetBlaster cables, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

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

(5) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(6) Power up the V_{CC} of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5- V V_{CCA} supply. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(7) Connect the series resistor at the near end of the serial configuration device.

(8) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

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

(10) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

ISP of the Configuration Device

In the second stage, the SFL design in the master device allows you to write the configuration data for the device chain into the serial configuration device with the Cyclone IV device JTAG interface. The JTAG interface sends the programming data for the serial configuration device to the Cyclone IV device first. The Cyclone IV device then uses the ASMI pins to send the data to the serial configuration device.

Reconfiguration

After the configuration data is successfully written into the serial configuration device, the Cyclone IV device does not automatically start reconfiguration. The intelligent host issues the PULSE_NCONFIG JTAG instruction to initialize the reconfiguration process. During reconfiguration, the master device is reset and the SFL design no longer exists in the Cyclone IV device and the serial configuration device configures all the devices in the chain with the user design.

👣 For more information about the SFL, refer to *AN 370: Using the Serial FlashLoader with Quartus II Software.*

JTAG Instructions

👣 For more information about the JTAG binary instruction code, refer to the *JTAG Boundary-Scan Testing for Cyclone IV Devices* chapter.

I/O Reconfiguration

Use the CONFIG_IO instruction to reconfigure the I/O configuration shift register (IOCSR) chain. This instruction allows you to perform board-level testing prior to configuring the Cyclone IV device or waiting for a configuration device to complete configuration. After the configuration is interrupted and JTAG testing is complete, you must reconfigure the part through the PULSE_NCONFIG JTAG instruction or by pulsing the nCONFIG pin low.

You can issue the CONFIG_IO instruction any time during user mode.

You must meet the following timing restrictions when using the CONFIG_IO instruction:

■ The CONFIG_IO instruction cannot be issued when the nCONFIG pin is low

■ You must observe a 230 μs minimum wait time after any of the following conditions:

   ■ nCONFIG pin goes high

   ■ Issuing the PULSE_NCONFIG instruction

   ■ Issuing the ACTIVE_ENGAGE instruction, before issuing the CONFIG_IO instruction

■ You must wait 230 μs after power up, with the nCONFIG pin high before issuing the CONFIG_IO instruction (or wait for the nSTATUS pin to go high)

Use the ACTIVE_DISENGAGE instruction with the CONFIG_IO instruction to interrupt configuration. Table 8–16 lists the sequence of instructions to use for various CONFIG_IO usage scenarios.

Table 8–16. JTAG CONFIG_IO (without JTAG_PROGRAM) Instruction Flows *(1)*

JTAG Instruction	Configuration Scheme and Current State of the Cyclone IV Device											
	Prior to User Mode (Interrupting Configuration)				User Mode				Power Up			
	PS	FPP	AS	AP	PS	FPP	AS	AP	PS	FPP	AS	AP
ACTIVE_DISENGAGE	O	O	O	O	O	O	O	O	—	—	—	—
CONFIG_IO	R	R	R	R	R	R	R	R	NA	NA	NA	NA
JTAG Boundary Scan Instructions (no JTAG_PROGRAM)	O	O	O	O	O	O	O	O	—	—	—	—
ACTIVE_ENGAGE	A	A	R *(2)*	R *(2)*	A	A	R *(2)*	R *(2)*	—	—	—	—
PULSE_NCONFIG			A *(3)*	A *(3)*			O	O	—	—	—	—
Pulse nCONFIG pin			A *(3)*	A *(3)*			O	O	—	—	—	—
JTAG TAP Reset	R	R	R	R	R	R	R	R	—	—	—	—

Notes to Table 8–16:

(1) You must execute "R" indicates that the instruction before the next instruction, "O" indicates the optional instruction, "A" indicates that the instruction must be executed, and "NA" indicates that the instruction is not allowed in this mode.

(2) Required if you use ACTIVE_DISENGAGE.

(3) Neither of the instruction is required if you use ACTIVE_ENGAGE.

The CONFIG_IO instruction does not hold nSTATUS low until reconfiguration. You must disengage the AS or AP configuration controller by issuing the ACTIVE_DISENGAGE and ACTIVE_ENGAGE instructions when active configuration is interrupted. You must issue the ACTIVE_DISENGAGE instruction alone or prior to the CONFIG_IO instruction if the JTAG_PROGRAM instruction is to be issued later (Table 8–17). This puts the active configuration controllers into the idle state. The active configuration controller is re-engaged after user mode is reached through JTAG programming (Table 8–17).

☞   While executing the CONFIG_IO instruction, all user I/Os are tri-stated.

If reconfiguration after interruption is performed using configuration modes (rather than using JTAG_PROGRAM), it is not necessary to issue the ACTIVE_DISENGAGE instruction prior to CONFIG_IO. You can start reconfiguration by either pulling nCONFIG low for at least 500 ns or issuing the PULSE_NCONFIG instruction. If the ACTIVE_DISENGAGE instruction was issued and the JTAG_PROGRAM instruction fails to enter user mode, you must issue the ACTIVE_ENGAGE instruction to reactivate the active configuration controller. Issuing the ACTIVE_ENGAGE instruction also triggers reconfiguration in configuration modes; therefore, it is not necessary to pull nCONFIG low or issue the PULSE_NCONFIG instruction.

ACTIVE_DISENGAGE

The ACTIVE_DISENGAGE instruction places the active configuration controller (AS and AP) into an idle state prior to JTAG programming. The two purposes of placing the active controller in an idle state are:

■ To ensure that it is not trying to configure the device during JTAG programming

■ To allow the controllers to properly recognize a successful JTAG programming that results in the device reaching user mode

The ACTIVE_DISENGAGE instruction is required before JTAG programming regardless of the current state of the Cyclone IV device if the MSEL pins are set to an AS or AP configuration scheme. If the ACTIVE_DISENGAGE instruction is issued during a passive configuration scheme (PS or FPP), it has no effect on the Cyclone IV device. Similarly, the CONFIG_IO instruction is issued after an ACTIVE_DISENGAGE instruction, but is no longer required to properly halt configuration. Table 8–17 lists the required, recommended, and optional instructions for each configuration mode. The ordering of the required instructions is a hard requirement and must be met to ensure functionality.

Table 8–17. JTAG Programming Instruction Flows [1]

| JTAG Instruction | Configuration Scheme and Current State of the Cyclone IV Device | | | | | | | | | | | |
| | Prior to User Mode (Interrupting Configuration) | | | | User Mode | | | | Power Up | | | |
	PS	FPP	AS	AP	PS	FPP	AS	AP	PS	FPP	AS	AP
ACTIVE_DISENGAGE	O	O	R	R	O	O	O	R	O	O	R	R
CONFIG_IO	Rc	Rc	O	O	O	O	O	O	NA	NA	NA	NA
Other JTAG instructions	O	O	O	O	O	O	O	O	O	O	O	O
JTAG_PROGRAM	R	R	R	R	R	R	R	R	R	R	R	R
CHECK_STATUS	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc
JTAG_STARTUP	R	R	R	R	R	R	R	R	R	R	R	R
JTAG TAP Reset/other instruction	R	R	R	R	R	R	R	R	R	R	R	R

Note to Table 8–17:

(1) "R" indicates that the instruction must be executed before the next instruction, "O" indicates the optional instruction, "Rc" indicates the recommended instruction, and "NA" indicates that the instruction is not allowed in this mode.

In the AS or AP configuration scheme, the ACTIVE_DISENGAGE instruction puts the active configuration controller into idle state. If a successful JTAG programming is executed, the active controller is automatically re-engaged after user mode is reached through JTAG programming. This causes the active controller to transition to their respective user mode states.

If JTAG programming fails to get the Cyclone IV device to enter user mode and re-engage active programming, there are available methods to achieve this:

■ In AS configuration scheme, you can re-engage the AS controller by moving the JTAG TAP controller to the reset state or by issuing the ACTIVE_ENGAGE instruction.

■ In AP configuration scheme, the only way to re-engage the AP controller is to issue the ACTIVE_ENGAGE instruction. In this case, asserting the nCONFIG pin does not re-engage either active controller.

ACTIVE_ENGAGE

The ACTIVE_ENGAGE instruction allows you to re-engage a disengaged active controller. You can issue this instruction any time during configuration or user mode to re-engage an already disengaged active controller, as well as trigger reconfiguration of the Cyclone IV device in the active configuration scheme.

The ACTIVE_ENGAGE instruction functions as the PULSE_NCONFIG instruction when the device is in the PS or FPP configuration schemes. The nCONFIG pin is disabled when the ACTIVE_ENGAGE instruction is issued.

☞ Altera does not recommend using the ACTIVE_ENGAGE instruction, but it is provided as a fail-safe instruction for re-engaging the active configuration controller (AS and AP).

Overriding the Internal Oscillator

This feature allows you to override the internal oscillator during the active configuration scheme. The AS and AP configuration controllers use the internal oscillator as the clock source. You can change the clock source to CLKUSR through the JTAG instruction.

The EN_ACTIVE_CLK and DIS_ACTIVE_CLK JTAG instructions toggle on or off whether or not the active clock is sourced from the CLKUSR pin or the internal configuration oscillator. To source the active clock from the CLKUSR pin, issue the EN_ACTIVE_CLK instruction. This causes the CLKUSR pin to become the active clock source. When using the EN_ACTIVE_CLK instruction, you must enable the internal oscillator for the clock change to occur. By default, the configuration oscillator is disabled after configuration and initialization is complete as well as the device has entered user mode.

However, the internal oscillator is enabled in user mode by any of the following conditions:

■ A reconfiguration event (for example, driving the nCONFIG pin to go low)

■ Remote update is enabled

■ Error detection is enabled

☞ When using the EN_ACTIVE_CLK and DIS_ACTIVE_CLK JTAG instructions to override the internal oscillator, you must clock the CLKUSR pin at two times the expected DCLK frequency. The CLKUSR pin allows a maximum frequency of 40 MHz (40 MHz DCLK).

Normally, a test instrument uses the CLKUSR pin when it wants to drive its own clock to control the AS state machine.

To revert the clock source back to the configuration oscillator, issue the DIS_ACTIVE_CLK instruction. After you issue the DIS_ACTIVE_CLK instruction, you must continue to clock the CLKUSR pin for 10 clock cycles. Otherwise, even toggling the nCONFIG pin does not revert the clock source and reconfiguration does not occur. A POR reverts the clock source back to the configuration oscillator. Toggling the nCONFIG pin or driving the JTAG state machine to reset state does not revert the clock source.

EN_ACTIVE_CLK

The `EN_ACTIVE_CLK` instruction causes the `CLKUSR` pin signal to replace the internal oscillator as the clock source. When using the `EN_ACTIVE_CLK` instruction, you must enable the internal oscillator for the clock change to occur. After this instruction is issued, other JTAG instructions can be issued while the `CLKUSR` pin signal remains as the clock source. The clock source is only reverted back to the internal oscillator by issuing the `DIS_ACTIVE_CLK` instruction or a POR.

DIS_ACTIVE_CLK

The `DIS_ACTIVE_CLK` instruction breaks the `CLKUSR` enable latch set by the `EN_ACTIVE_CLK` instruction and causes the clock source to revert back to the internal oscillator. After the `DIS_ACTIVE_CLK` instruction is issued, you must continue to clock the `CLKUSR` pin for 10 clock cycles.

Changing the Start Boot Address of the AP Flash

In the AP configuration scheme (for Cyclone IV E devices only), you can change the default configuration boot address of the parallel flash memory to any desired address using the `APFC_BOOT_ADDR` JTAG instruction.

APFC_BOOT_ADDR

The `APFC_BOOT_ADDR` instruction is for Cyclone IV E devices only and allows you to define a start boot address for the parallel flash memory in the AP configuration scheme.

This instruction shifts in a start boot address for the AP flash. When this instruction becomes the active instruction, the `TDI` and `TDO` pins are connected through a 22-bit active boot address shift register. The shifted-in boot address bits get loaded into the 22-bit AP boot address update register, which feeds into the AP controller. The content of the AP boot address update register can be captured and shifted-out of the active boot address shift register from `TDO`.

The boot address in the boot address shift register and update register are shifted to the right (in the LSB direction) by two bits versus the intended boot address. The reason for this is that the two LSB of the address are not accessible. When this boot address is fed into the AP controller, two 0s are attached in the end as LSB, thereby pushing the shifted-in boot address to the left by two bits, which become the actual AP boot address the AP controller gets.

If you have enabled the remote update feature, the `APFC_BOOT_ADDR` instruction sets the boot address for the factory configuration only.

☞ The `APFC_BOOT_ADDR` instruction is retained after reconfiguration while the system board is still powered on. However, you must reprogram the instruction whenever you restart the system board.

Device Configuration Pins

Table 8–18 through Table 8–21 describe the connections and functionality of all the configuration related pins on Cyclone IV devices. Table 8–18 and Table 8–19 list the device pin configuration for the Cyclone IV GX and Cyclone IV E, respectively.

Table 8–18. Configuration Pin Summary for Cyclone IV GX Devices

Bank	Description	Input/Output	Dedicated	Powered By	Configuration Mode
8	Data[4:2]	Input	—	V_{CCIO}	FPP
3	Data[7:5]	Input	—	V_{CCIO}	FPP
9	nCSO [2]	Output	—	V_{CCIO}	AS
3	CRC_ERROR	Output	—	V_{CCIO}/Pull-up [1]	Optional, all modes
9	DATA[0] [2]	Input	Yes	V_{CCIO}	PS, FPP, AS
9	DATA[1]/ASDO [2]	Input	—	V_{CCIO}	FPP
		Output		V_{CCIO}	AS
3	INIT_DONE	Output	—	Pull-up	Optional, all modes
3	nSTATUS	Bidirectional	Yes	Pull-up	All modes
9	nCE	Input	Yes	V_{CCIO}	All modes
9	DCLK [2]	Input	Yes	V_{CCIO}	PS, FPP
		Output	Yes	V_{CCIO}	AS
3	CONF_DONE	Bidirectional	Yes	Pull-up	All modes
9	TDI	Input	Yes	V_{CCIO}	JTAG
9	TMS	Input	Yes	V_{CCIO}	JTAG
9	TCK	Input	Yes	V_{CCIO}	JTAG
9	nCONFIG	Input	Yes	V_{CCIO}	All modes
8	CLKUSR	Input	—	V_{CCIO}	Optional
3	nCEO	Output	—	V_{CCIO}	Optional, all modes
3	MSEL	Input	Yes	V_{CCINT}	All modes
9	TDO	Output	Yes	V_{CCIO}	JTAG
6	DEV_OE	Input	—	V_{CCIO}	Optional
6	DEV_CLRn	Input	—	V_{CCIO}	Optional

Notes to Table 8–18:

(1) The CRC_ERROR pin is a dedicated open-drain output or an optional user I/O pin. Active high signal indicates that the error detection circuit has detected errors in the configuration SRAM bits. This pin is optional and is used when the CRC error detection circuit is enabled in the Quartus II software from the **Error Detection CRC** tab of the **Device and Pin Options** dialog box. When using this pin, connect it to an external 10-kΩ pull-up resistor to an acceptable voltage that satisfies the input voltage of the receiving device.

(2) To tri-state AS configuration pins in the AS configuration scheme, turn on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, nCSO, Data[0], and Data[1]/ASDO pins. Dual-purpose pins settings for these pins are ignored. To set these pins to different settings, turn off the **Enable input tri-state on active configuration pins in user mode** option and set the desired setting from the Dual-purpose Pins Setting menu.

Table 8–19. Configuration Pin Summary for Cyclone IV E Devices (Part 1 of 3)

Bank	Description	Input/Output	Dedicated	Powered By	Configuration Mode
1	nCSO [1] FLASH_nCE [2]	Output	—	V_{CCIO}	AS, AP
6	CRC_ERROR [3]	Output	—	V_{CCIO}/Pull-up [4]	Optional, all modes

Table 8–19. Configuration Pin Summary for Cyclone IV E Devices (Part 2 of 3)

Bank	Description	Input/Output	Dedicated	Powered By	Configuration Mode
1	DATA[0] (1), (2)	Input	—	V_{CCIO}	PS, FPP, AS
		Bidirectional		V_{CCIO}	AP
1	DATA[1] (2)/ASDO (1)	Input	—	V_{CCIO}	FPP
		Output		V_{CCIO}	AS
		Bidirectional		V_{CCIO}	AP
8	DATA[7..2] (2)	Input	—	V_{CCIO}	FPP
		Bidirectional		V_{CCIO}	AP
8	DATA[15..8] (2)	Bidirectional	—	V_{CCIO}	AP
6	INIT_DONE	Output	—	Pull-up	Optional, all modes
1	nSTATUS	Bidirectional	Yes	Pull-up	All modes
1	nCE	Input	Yes	V_{CCIO}	All modes
1	DCLK (1), (2)	Input	Yes	V_{CCIO}	PS, FPP
		Output	—	V_{CCIO}	AS, AP
6	CONF_DONE	Bidirectional	Yes	Pull-up	All modes
1	TDI	Input	Yes	V_{CCIO}	JTAG
1	TMS	Input	Yes	V_{CCIO}	JTAG
1	TCK	Input	Yes	V_{CCIO}	JTAG
1	nCONFIG	Input	Yes	V_{CCIO}	All modes
6	CLKUSR	Input	—	V_{CCIO}	Optional
6	nCEO	Output	—	V_{CCIO}	Optional, all modes
6	MSEL[]	Input	Yes	V_{CCINT}	All modes
1	TDO	Output	Yes	V_{CCIO}	JTAG
7	PADD[14..0]	Output	—	V_{CCIO}	AP
8	PADD[19..15]	Output	—	V_{CCIO}	AP
6	PADD[23..20]	Output	—	V_{CCIO}	AP
1	nRESET	Output	—	V_{CCIO}	AP
6	nAVD	Output	—	V_{CCIO}	AP
6	nOE	Output	—	V_{CCIO}	AP
6	nWE	Output	—	V_{CCIO}	AP
5	DEV_OE	Input	—	V_{CCIO}	Optional, AP

Table 8–19. Configuration Pin Summary for Cyclone IV E Devices (Part 3 of 3)

Bank	Description	Input/Output	Dedicated	Powered By	Configuration Mode
5	DEV_CLRn	Input	—	V$_{CCIO}$	Optional, AP

Notes to **Table 8–19**:

(1) To tri-state AS configuration pins in the AS configuration scheme, turn-on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, nCSO, Data[0], and Data[1]/ASDO pins. Dual-purpose pins settings for these pins are ignored. To set these pins to different settings, turn off the **Enable input tri-state on active configuration pins in user mode** option and set the desired setting from the Dual-purpose Pins Setting menu.

(2) To tri-state AP configuration pins in the AP configuration scheme, turn-on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, Data[0..15], FLASH_nCE, and other AP pins. Dual-purpose pins settings for these pins are ignored. To set these pins to different settings, turn off the **Enable input tri-state on active configuration pins in user mode** option and set the desired setting from the Dual-purpose Pins Setting menu.

(3) The CRC_ERROR pin is not available in Cyclone IV E devices with 1.0-V core voltage.

(4) The CRC_ERROR pin is a dedicated open-drain output or an optional user I/O pin. Active high signal indicates that the error detection circuit has detected errors in the configuration SRAM bits. This pin is optional and is used when the CRC error detection circuit is enabled in the Quartus II software from the **Error Detection CRC** tab of the **Device and Pin Options** dialog box. When using this pin, connect it to an external 10-kΩ pull-up resistor to an acceptable voltage that satisfies the input voltage of the receiving device.

Table 8–20 describes the dedicated configuration pins. You must properly connect these pins on your board for successful configuration. You may not need some of these pins for your configuration schemes.

Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 1 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
MSEL	N/A	All	Input	Configuration input that sets the Cyclone IV device configuration scheme. You must hardwire these pins to V$_{CCA}$ or GND. The MSEL pins have internal 9-kΩ pull-down resistors that are always active.
nCONFIG	N/A	All	Input	Configuration control input. Pulling this pin low with external circuitry during user mode causes the Cyclone IV device to lose its configuration data, enter a reset state, and tri-state all I/O pins. Returning this pin to a logic-high level starts a reconfiguration.
nSTATUS	N/A	All	Bidirectional open-drain	The Cyclone IV device drives nSTATUS low immediately after power-up and releases it after the POR time. ■ Status output—if an error occurs during configuration, nSTATUS is pulled low by the target device. ■ Status input—if an external source (for example, another Cyclone IV device) drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. Driving nSTATUS low after configuration and initialization does not affect the configured device. If you use a configuration device, driving nSTATUS low causes the configuration device to attempt to configure the device, but because the device ignores transitions on nSTATUS in user mode, the device does not reconfigure. To start a reconfiguration, you must pull nCONFIG low.

Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 2 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
CONF_DONE	N/A	All	Bidirectional open-drain	■ Status output—the target Cyclone IV device drives the CONF_DONE pin low before and during configuration. After all the configuration data is received without error and the initialization cycle starts, the target device releases CONF_DONE. ■ Status input—after all the data is received and CONF_DONE goes high, the target device initializes and enters user mode. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize. Driving CONF_DONE low after configuration and initialization does not affect the configured device. Do not connect bus holds or ADC to CONF_DONE pin.
nCE	N/A	All	Input	Active-low chip enable. The nCE pin activates the Cyclone IV device with a low signal to allow configuration. You must hold nCE pin low during configuration, initialization, and user-mode. In a single-device configuration, you must tie the nCE pin low. In a multi-device configuration, nCE of the first device is tied low while its nCEO pin is connected to nCE of the next device in the chain. You must hold the nCE pin low for successful JTAG programming of the device.
nCEO	N/A if option is on. I/O if option is off.	All	Output open-drain	Output that drives low when configuration is complete. In a single-device configuration, you can leave this pin floating or use it as a user I/O pin after configuration. In a multi-device configuration, this pin feeds the nCE pin of the next device. The nCEO of the last device in the chain is left floating or used as a user I/O pin after configuration. If you use the nCEO pin to feed the nCE pin of the next device, use an external 10-kΩ pull-up resistor to pull the nCEO pin high to the V_CCIO voltage of its I/O bank to help the internal weak pull-up resistor. If you use the nCEO pin as a user I/O pin after configuration, set the state of the pin on the **Dual-Purpose Pin** settings.
nCSO, FLASH_nCE [1]	I/O	AS, AP [2]	Output	Output control signal from the Cyclone IV device to the serial configuration device in AS mode that enables the configuration device. This pin functions as nCSO in AS mode and FLASH_nCE in AP mode. Output control signal from the Cyclone IV device to the parallel flash in AP mode that enables the flash. Connects to the CE# pin on the Micron P30 or P33 flash. [2] This pin has an internal pull-up resistor that is always active.

Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 3 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DCLK [1]	N/A	PS, FPP, AS, AP [2]	Input (PS, FPP) [2]	In PS and FPP configuration, DCLK is the clock input used to clock data from an external source into the target Cyclone IV device. Data is latched into the device on the rising edge of DCLK.
	I/O		Output (AS, AP)	In AS mode, DCLK is an output from the Cyclone IV device that provides timing for the configuration interface. It has an internal pull-up resistor (typically 25 kΩ) that is always active.
				In AP mode, DCLK is an output from the Cyclone IV E device that provides timing for the configuration interface. [2]
				In AS or AP configuration schemes, this pin is driven into an inactive state after configuration completes. Alternatively, in active schemes, you can use this pin as a user I/O during user mode. In PS or FPP schemes that use a control host, you must drive DCLK either high or low, whichever is more convenient. In passive schemes, you cannot use DCLK as a user I/O in user mode. Toggling this pin after configuration does not affect the configured device.
DATA[0] [1]	I/O	PS, FPP, AS, AP [2]	Input (PS, FPP, AS). Bidirectional (AP) [2]	Data input. In serial configuration modes, bit-wide configuration data is presented to the target Cyclone IV device on the DATA[0] pin.
				In AS mode, DATA[0] has an internal pull-up resistor that is always active. After AS configuration, DATA[0] is a dedicated input pin with optional user control.
				After PS or FPP configuration, DATA[0] is available as a user I/O pin. The state of this pin depends on the **Dual-Purpose Pin** settings.
				After AP configuration, DATA[0] is a dedicated bidirectional pin with optional user control. [2]
DATA[1]/ ASDO [1]	I/O	FPP, AS, AP [2]	Input (FPP). Output (AS). Bidirectional (AP) [2]	The DATA[1] pin functions as the ASDO pin in AS mode. Data input in non-AS mode. Control signal from the Cyclone IV device to the serial configuration device in AS mode used to read out configuration data.
				In AS mode, DATA[1] has an internal pull-up resistor that is always active. After AS configuration, DATA[1] is a dedicated output pin with optional user control.
				In a PS configuration scheme, DATA[1] functions as a user I/O pin during configuration, which means it is tri-stated.
				After FPP configuration, DATA[1] is available as a user I/O pin and the state of this pin depends on the **Dual-Purpose Pin** settings.
				In an AP configuration scheme, for Cyclone IV E devices only, the byte-wide or word-wide configuration data is presented to the target Cyclone IV E device on DATA[7..0] or DATA[15..0], respectively. After AP configuration, DATA[1] is a dedicated bidirectional pin with optional user control. [2]

Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 4 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DATA[7..2]	I/O	FPP, AP [2]	Inputs (FPP). Bidirectional (AP) [2]	In an AS or PS configuration scheme, DATA[7..2] function as user I/O pins during configuration, which means they are tri-stated. After FPP configuration, DATA[7..2] are available as user I/O pins and the state of these pin depends on the **Dual-Purpose Pin** settings. In an AP configuration scheme, for Cyclone IV E devices only, the byte-wide or word-wide configuration data is presented to the target Cyclone IV E device on DATA[7..0] or DATA[15..0], respectively. After AP configuration, DATA[7..2] are dedicated bidirectional pins with optional user control. [2]
DATA[15..8]	I/O	AP [2]	Bidirectional	Data inputs. Word-wide configuration data is presented to the target Cyclone IV E device on DATA[15..0]. In a PS, FPP, or AS configuration scheme, DATA[15:8] function as user I/O pins during configuration, which means they are tri stated. After AP configuration, DATA[15:8] are dedicated bidirectional pins with optional user control.
PADD[23..0]	I/O	AP [2]	Output	In AP mode, it is a 24-bit address bus from the Cyclone IV E device to the parallel flash. Connects to the A[24:1] bus on the Micron P30 or P33 flash.
nRESET	I/O	AP [2]	Output	Active-low reset output. Driving the nRESET pin low resets the parallel flash. Connects to the RST# pin on the Micron P30 or P33 flash.
nAVD	I/O	AP [2]	Output	Active-low address valid output. Driving the nAVD pin low during read or write operation indicates to the parallel flash that a valid address is present on the PADD[23..0] address bus. Connects to the ADV# pin on the Micron P30 or P33 flash.
nOE	I/O	AP [2]	Output	Active-low output enable to the parallel flash. During the read operation, driving the nOE pin low enables the parallel flash outputs (DATA[15..0]). Connects to the OE# pin on the Micron P30 or P33 flash.
nWE	I/O	AP [2]	Output	Active-low write enable to the parallel flash. During the write operation, driving the nWE pin low indicates to the parallel flash that data on the DATA[15..0] bus is valid. Connects to the WE# pin on the Micron P30 or P33 flash.

Note to Table 8–20:

(1) If you are accessing the EPCS device with the ALTASMI_PARALLEL megafunction or your own user logic in user mode, in the **Device and Pin Options** window of the Quartus II software, in the **Dual-Purpose Pins** category, select **Use as regular I/O** for this pin.

(2) The AP configuration scheme is for Cyclone IV E devices only.

Table 8–21 lists the optional configuration pins. If you do not enable these optional configuration pins in the Quartus II software, they are available as general-purpose user I/O pins. Therefore, during configuration, these pins function as user I/O pins and are tri-stated with weak pull-up resistors.

Table 8–21. Optional Configuration Pins

Pin Name	User Mode	Pin Type	Description
CLKUSR	N/A if option is on. I/O if option is off.	Input	Optional user-supplied clock input synchronizes the initialization of one or more devices. This pin is enabled by turning on the **Enable user-supplied start-up clock (CLKUSR)** option in the Quartus II software. In AS configuration for Cyclone IV GX devices, you can use this pin as an external clock source to generate the DCLK by changing the clock source option in the Quartus II software in the **Configuration** tab of the **Device and Pin Options** dialog box.
INIT_DONE	N/A if option is on. I/O if option is off.	Output open-drain	Status pin is used to indicate when the device has initialized and is in user-mode. When nCONFIG is low, the INIT_DONE pin is tri-stated and pulled high due to an external 10-kΩ pull-up resistor 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 and the device enters user mode. Thus, the monitoring circuitry must be able to detect a low-to-high transition. This pin is enabled by turning on the **Enable INIT_DONE output** option in the Quartus II software. The functionality of this pin changes if the **Enable OCT_DONE** option is enabled in the Quartus II software. This option controls whether the INIT_DONE signal is gated by the OCT_DONE signal, which indicates the power-up on-chip termination (OCT) calibration is complete. If this option is turned off, the INIT_DONE signal is not gated by the OCT_DONE signal.
DEV_OE	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all tri-states on the device. When this pin is driven low, all I/O pins are tri-stated; when this pin is driven high, all I/O pins behave as programmed. This pin is enabled by turning on the **Enable device-wide output enable (DEV_OE)** option in the Quartus II software.
DEV_CLRn	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all clears on all device registers. When this pin is driven low, all registers are cleared; when this pin is driven high, all registers behave as programmed. You can enable this pin by turning on the **Enable device-wide reset (DEV_CLRn)** option in the Quartus II software.

Remote System Upgrade

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

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

Functional Description

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

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

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

1. A Nios II processor (or user logic) implemented in the Cyclone IV device logic array receives new configuration data from a remote location. The connection to the remote source is a communication protocol, such as the transmission control protocol/Internet protocol (TCP/IP), peripheral component interconnect (PCI), user datagram protocol (UDP), universal asynchronous receiver/transmitter (UART), or a proprietary interface.

2. The Nios II processor (or user logic) writes this new configuration data into a configuration memory.

3. The Nios II processor (or user logic) starts a reconfiguration cycle with the new or updated configuration data.

4. The dedicated remote system upgrade circuitry detects and recovers from any error that might occur during or after the reconfiguration cycle and provides error status information to the user design.

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

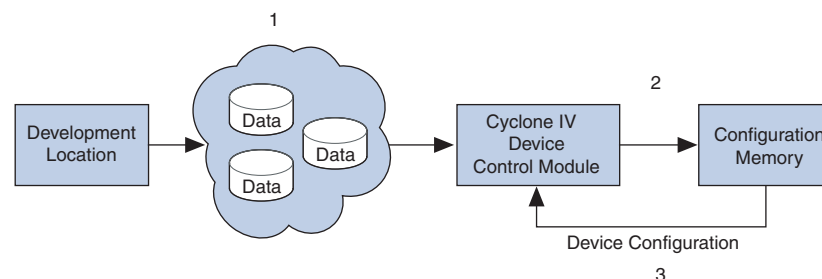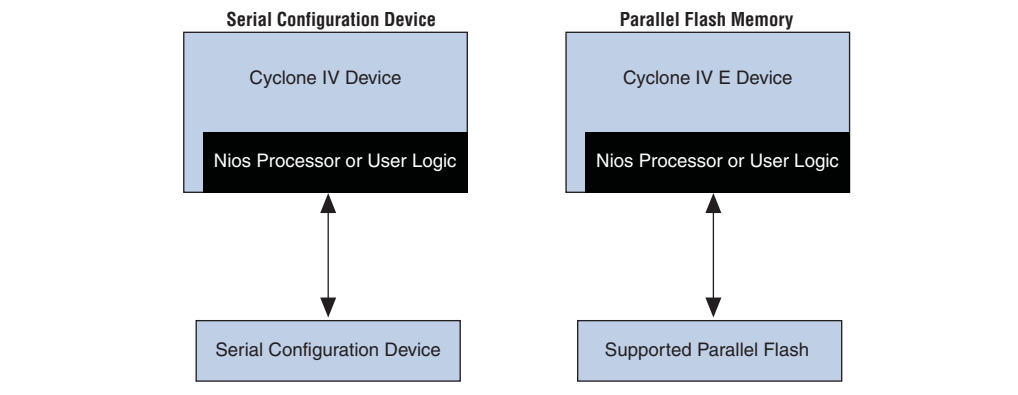**Figure 8–30. Functional Diagram of Cyclone IV Device Remote System Upgrade**

Figure 8–31 shows the block diagrams to implement remote system upgrade in Cyclone IV devices.

Figure 8–31. Remote System Upgrade Block Diagrams for AS and AP Configuration Schemes

The `MSEL` pin setting in the remote system upgrade mode is the same as the standard configuration mode. Standard configuration mode refers to normal Cyclone IV device configuration mode with no support for remote system upgrades (the remote system upgrade circuitry is disabled). When using remote system upgrade in Cyclone IV devices, you must enable the remote update mode option setting in the Quartus II software.

Enabling Remote Update

You can enable or disable remote update for Cyclone IV devices in the Quartus II software before design compilation (in the Compiler Settings menu). To enable remote update in the compiler settings of the project, perform the following steps:

1. On the Assignments menu, click **Device**. The **Settings** dialog box appears.

2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.

3. Click the **Configuration** tab.

4. From the **Configuration Mode** list, select **Remote**.

5. Click **OK**.

6. In the **Settings** dialog box, click **OK**.

Configuration Image Types

When using remote system upgrade, Cyclone IV device configuration bitstreams are classified as factory configuration images or application configuration images. An image, also referred to as a configuration, is a design loaded into the device that performs certain user-defined functions. Each device in your system requires one factory image or with addition of one or more application images. The factory image is a user-defined fall-back or safe configuration and is responsible for administering remote updates with the dedicated circuitry. Application images implement user-defined functionality in the target Cyclone IV device. You can include the default application image functionality in the factory image.

Remote System Upgrade Mode

In remote update mode, Cyclone IV devices load the factory configuration image after power up. The user-defined factory configuration determines the application configuration to be loaded and triggers a reconfiguration cycle. The factory configuration can also contain application logic.

When used with configuration memory, the remote update mode allows an application configuration to start at any flash sector boundary. Additionally, the remote update mode features a user watchdog timer that can detect functional errors in an application configuration.

Remote Update Mode

In AS configuration scheme, when a Cyclone IV device is first powered up in remote update, it loads the factory configuration located at address `boot_address[23:0] = 24b'0`. Altera recommends storing the factory configuration image for your system at boot address `24b'0`, which corresponds to the start address location `0x000000` in the serial configuration device. A factory configuration image is a bitstream for the Cyclone IV device in your system that is programmed during production and is the fall-back image when an error occurs. This image is stored in non-volatile memory and is never updated or modified using remote access.

When you use the AP configuration in Cyclone IV E devices, the Cyclone IV E device loads the default factory configuration located at the following address after device power-up in remote update mode:
`boot_address[23:0] = 24'h010000 = 24'b1 0000 0000 0000 0000`.

You can change the default factory configuration address to any desired address using the `APFC_BOOT_ADDR` JTAG instruction. The factory configuration image is stored in non-volatile memory and is never updated or modified using remote access. This corresponds to the default start address location `0x010000` represented in 16-bit word addressing (or the updated address if the default address is changed) in the supported parallel flash memory. For more information about the application of the `APFC_BOOT_ADDR` JTAG instruction in AP configuration scheme, refer to the "JTAG Instructions" on page 8–57.

The factory configuration image is user-designed and contains soft logic (Nios II processor or state machine and the remote communication interface) to:

■ Process any errors based on status information from the dedicated remote system upgrade circuitry

■ Communicate with the remote host and receive new application configurations and store the new configuration data in the local non-volatile memory device

■ Determine the application configuration to be loaded into the Cyclone IV device

■ Enable or disable the user watchdog timer and load its time-out value (optional)

■ Instruct the dedicated remote system upgrade circuitry to start a reconfiguration cycle

Figure 8–32 shows the transitions between the factory configuration and application configuration in remote update mode.

Figure 8–32. Transitions Between Configurations in Remote Update Mode
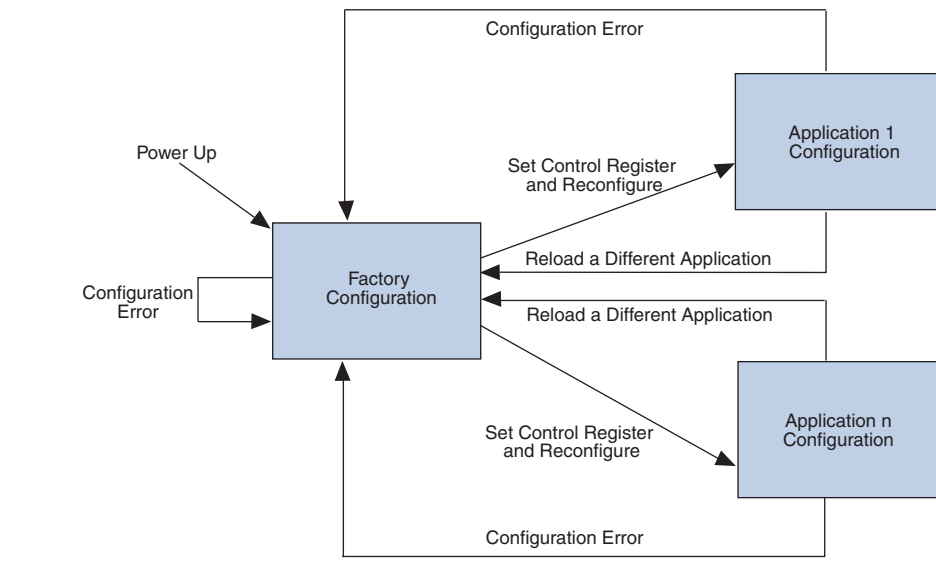
After power up or a configuration error, the factory configuration logic writes the remote system upgrade control register to specify the address of the application configuration to be loaded. The factory configuration also specifies whether or not to enable the user watchdog timer for the application configuration and, if enabled, specifies the timer setting.

☞ Only valid application configurations designed for remote update mode include the logic to reset the timer in user mode. For more information about the user watchdog timer, refer to the "User Watchdog Timer" on page 8–79.

If there is an error while loading the application configuration, the remote system upgrade status register is written by the dedicated remote system upgrade circuitry of the Cyclone IV device to specify the cause of the reconfiguration.

The following actions cause the remote system upgrade status register to be written:

■ nSTATUS driven low externally

■ Internal cyclical redundancy check (CRC) error

■ User watchdog timer time-out

■ A configuration reset (logic array nCONFIG signal or external nCONFIG pin assertion)

The Cyclone IV device automatically loads the factory configuration when an error occurs. This user-designed factory configuration reads the remote system upgrade status register to determine the reason for reconfiguration. Then the factory configuration takes the appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

When Cyclone IV devices successfully load the application configuration, they enter user mode. In user mode, the soft logic (the Nios II processor or state machine and the remote communication interface) assists the Cyclone IV device in determining when a remote system update is arriving. When a remote system update arrives, the soft logic receives the incoming data, writes it to the configuration memory device and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and starts system reconfiguration.

Dedicated Remote System Upgrade Circuitry

This section describes the implementation of the Cyclone IV device remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces with the user-defined factory application configurations implemented in the Cyclone IV device logic array to provide the complete remote configuration solution. The remote system upgrade circuitry contains the remote system upgrade registers, a watchdog timer, and state machines that control those components. Figure 8–33 shows the data path of the remote system upgrade block.

Figure 8–33. Remote System Upgrade Circuit Data Path *(1)*

Notes to Figure 8–33:

(1) The `RU_DOUT`, `RU_SHIFTnLD`, `RU_CAPTnUPDT`, `RU_CLK`, `RU_DIN`, `RU_nCONFIG`, and `RU_nRSTIMER` signals are internally controlled by the ALTREMOTE_UPDATE megafunction.

(2) The `RU_CLK` refers to the ALTREMOTE_UPDATE megafunction block "clock" input. For more information, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide.*

Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that stores the configuration addresses, watchdog timer settings, and status information. Table 8–22 lists these registers.

Table 8–22. Remote System Upgrade Registers

Register	Description
Shift register	This register is accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic. Write access is enabled in remote update mode for factory configurations to allow writing to the update register. Write access is disabled for all application configurations in remote update mode.
Control register	This register contains the current configuration address, the user watchdog timer settings, one option bit for checking early CONF_DONE, and one option bit for selecting the internal oscillator as the startup state machine clock. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is started, the contents of the update register are written into the control register.
Update register	This register contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a read in a factory configuration, this register is read into the shift register.
Status register	This register is written by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

The control and status registers of the remote system upgrade are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer) or the CLKUSR. However, the shift and update registers of the remote system upgrade are clocked by the maximum frequency of 40-MHz user clock input (RU_CLK). There is no minimum frequency for RU_CLK.

Remote System Upgrade Control Register

The remote system upgrade control register stores the application configuration address, the user watchdog timer settings, and option bits for a application configuration. In remote update mode for the AS configuration scheme, the control register address bits are set to all zeros (24'b0) at power up to load the AS factory configuration. In remote update mode for the AP configuration scheme, the control register address bits are set to 24'h010000 (24'b1 0000 0000 0000 0000) at power up to load the AP default factory configuration. However, for the AP configuration scheme, you can change the default factory configuration address to any desired address using the APFC_BOOT_ADDR JTAG instruction. Additionally, a factory configuration in remote update mode has write access to this register.

Figure 8–34 shows the control register bit positions. Table 8–23 defines the control register bit contents. The numbers in Figure 8–34 show the bit position of a setting in a register. For example, bit number 35 is the enable bit for the watchdog timer.

Figure 8–34. Remote System Upgrade Control Register

38	37	36	35	34	33	12	11	0
Rsv2	Cd_early	Osc_int	Wd_en	Rsv1	Ru_address[21..0]		Wd_timer[11..0]	

Table 8–23. Remote System Upgrade Control Register Contents

Control Register Bit	Value	Definition
Wd_timer[11..0]	12'b000000000000	User watchdog time-out value (most significant 12 bits of 29-bit count value: $\{Wd\_timer[11..0],17'b1000\}$)
Ru_address[21..0]	22'b0000000000000000000000	Configuration address (most significant 22 bits of 24-bit boot address value: boot_address[23:0] = $\{Ru\_address[21..0],2'b0\}$)
Rsv1	1'b0	Reserved bit
Wd_en	1'b1	User watchdog timer enable bit
Osc_int [1]	1'b1	Internal oscillator as startup state machine clock enable bit
Cd_early [1]	1'b1	Early CONF_DONE check
Rsv2	1'b1	Reserved bit

Note to Table 8–23:

(1) Option bit for the application configuration.

When enabled, the early CONF_DONE check (Cd_early) option bit ensures that there is a valid configuration at the boot address specified by the factory configuration and that it is of the proper size. If an invalid configuration is detected or the CONF_DONE pin is asserted too early, the device resets and then reconfigures the factory configuration image. The internal oscillator (as the startup state machine clock [Osc_int] option bit) ensures a functional startup clock to eliminate the hanging of startup. When all option bits are turned on, they provide complete coverage for the programming and startup portions of the application configuration. Altera recommends turning on both the Cd_early and Osc_int option bits.

☞ The Cd_early and Osc_int option bits for the application configuration must be turned on by the factory configuration.

Remote System Upgrade Status Register

The remote system upgrade status register specifies the reconfiguration trigger condition. The various trigger and error conditions include:

■ Cyclical redundancy check (CRC) error during application configuration

■ nSTATUS assertion by an external device due to an error

■ Cyclone IV device logic array triggers a reconfiguration cycle, possibly after downloading a new application configuration image

■ External configuration reset (nCONFIG) assertion

■ User watchdog timer time out

Table 8–24 lists the contents of the current state logic in the status register, when the remote system upgrade master state machine is in factory configuration or application configuration accessing the factory information or application information, respectively. The status register bit in Table 8–24 lists the bit positions in a 32-bit logic.

Table 8–24. Remote System Upgrade Current State Logic Contents In Status Register

Remote System Upgrade Master State Machine	Status Register Bit	Definition	Description
Factory information [1]	31:30	Master state machine current state	The current state of the remote system upgrade master state machine
	29:24	Reserved bits	Padding bits that are set to all 0's
	23:0	Boot address	The current 24-bit boot address that was used by the configuration scheme as the start address to load the current configuration.
Application information 1 [2]	31:30	Master state machine current state	The current state of the remote system upgrade master state machine
	29	User watchdog timer enable bit	The current state of the user watchdog enable, which is active high
	28:0	User watchdog timer time-out value	The current entire 29-bit watchdog time-out value.
Application information 2 [2]	31:30	Master state machine current state	The current state of the remote system upgrade master state machine
	29:24	Reserved bits	Padding bits that are set to all 0's
	23:0	Boot address	The current 24-bit boot address that was used as the start address to load the current configuration

Notes to Table 8–24:

(1) The remote system upgrade master state machine is in factory configuration.

(2) The remote system upgrade master state machine is in application configuration.

The previous two application configurations are available in the previous state registers (previous state register 1 and previous state register 2), but only for debugging purposes.

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

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

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

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

Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (Table 8–22 on page 8–75). While both registers can only be updated when the device is loaded with a factory configuration image, the update register writes are controlled by the user logic, and the control register writes are controlled by the remote system upgrade state machine.

In factory configurations, the user logic should send the option bits (Cd_early and Osc_int), the configuration address, and watchdog timer settings for the next application configuration bit to the update register. When the logic array configuration reset (RU_nCONFIG) goes high, the remote system upgrade state machine updates the control register with the contents of the update register and starts system reconfiguration from the new application page.

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

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

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

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition, but before the factory configuration is loaded.

Table 8–26. Control Register Contents After an Error or Reconfiguration Trigger Condition

Reconfiguration Error/Trigger	Control Register Setting In Remote Update
nCONFIG reset	All bits are 0
nSTATUS error	All bits are 0
CORE triggered reconfiguration	Update register
CRC error	All bits are 0
Wd time out	All bits are 0

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from indefinitely stalling the device. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Cyclone IV device.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29 bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The remote system upgrade circuitry appends 17'b1000 to form the 29-bits value for the watchdog timer. The granularity of the timer setting is 2^{17} cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator or CLKUSR (maximum frequency of 40 MHz).

Table 8–27 lists the operating range of the 10-MHz internal oscillator.

Table 8–27. 10-MHz Internal Oscillator Specifications

Minimum	Typical	Maximum	Unit
5	6.5	10	MHz

The user watchdog timer begins counting after the application configuration enters device user mode. This timer must be periodically reloaded or reset by the application configuration before the timer expires by asserting RU_nRSTIMER. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (Wd) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

☞ To allow the remote system upgrade dedicated circuitry to reset the watchdog timer, you must assert the RU_nRSTIMER signal active for a minimum of 250 ns. This is equivalent to strobing the reset_timer input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

Errors during configuration are detected by the CRC engine. Functional errors must not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

☞ The user watchdog timer is disabled in factory configurations and during the configuration cycle of the application configuration. It is enabled after the application configuration enters user mode.

Quartus II Software Support

Implementation in your design requires a remote system upgrade interface between the Cyclone IV device logic array and remote system upgrade circuitry. You must also generate configuration files for production and remote programming of the system configuration memory. The Quartus II software provides these features.

The two implementation options, ALTREMOTE_UPDATE megafunction and remote system upgrade atom, are for the interface between the remote system upgrade circuitry and the device logic array interface. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

For more information about the ALTREMOTE_UPDATE megafunction, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide*.

Document Revision History

Table 8–28 lists the revision history for this chapter.

Table 8–28. Document Revision History (Part 1 of 2)

Date	Version	Changes
May 2013	1.7	■ Added Table 8–6. ■ Updated Table 8–9 to add new device options and packages. ■ Updated Figure 8–16 and Figure 8–22 to include user mode. ■ Updated the "Dedicated" column for DATA[0] and DCLK in Table 8–19. ■ Updated the "User Mode" and "Pin Type" columns for DCLK in Table 8–20.
February 2013	1.6	Updated Table 8–9 to add new device options and packages.
October 2012	1.5	■ Updated "AP Configuration Supported Flash Memories", "Configuration Data Decompression", and "Overriding the Internal Oscillator" sections. ■ Updated Figure 8–3, Figure 8–4, Figure 8–5, Figure 8–7, Figure 8–8, Figure 8–9, Figure 8–10, and Figure 8–11. ■ Updated Table 8–2, Table 8–8, Table 8–12, Table 8–13, Table 8–18, and Table 8–19.
November 2011	1.4	■ Added information about how to gain control of EPCS pins. ■ Updated "Reset", "Single-Device AS Configuration", "Single-Device AP Configuration", and "Overriding the Internal Oscillator" sections. ■ Added Table 8–7. ■ Updated Table 8–6 and Table 8–19. ■ Updated Figure 8–3, Figure 8–4, and Figure 8–5.
December 2010	1.3	■ Updated for the Quartus II software version 10.1 release. ■ Added Cyclone IV E new device package information. ■ Updated Table 8–7, Table 8–10, and Table 8–11. ■ Minor text edits.

Table 8–28. Document Revision History (Part 2 of 2)

Date	Version	Changes
July 2010	1.2	Updated for the Quartus II software 10.0 release: ■ Updated "Power-On Reset (POR) Circuit", "Configuration and JTAG Pin I/O Requirements", and "Reset" sections. ■ Updated Figure 8–10. ■ Updated Table 8–16 and Table 8–17.
February 2010	1.1	Updated for the Quartus II software 9.1 SP1 release: ■ Added "Overriding the Internal Oscillator" and "AP Configuration (Supported Flash Memories)" sections. ■ Updated "JTAG Instructions" section. ■ Added Table 8–6. ■ Updated Table 8–2, Table 8–3, Table 8–4, Table 8–6, Table 8–11, Table 8–13, Table 8–14, Table 8–15, and Table 8–18. ■ Updated Figure 8–4, Figure 8–5, Figure 8–6, Figure 8–13, Figure 8–14, Figure 8–15, Figure 8–17, Figure 8–18, Figure 8–23, Figure 8–24, Figure 8–25, Figure 8–26, Figure 8–27, Figure 8–28, and Figure 8–29.
November 2009	1.0	Initial release.

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

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

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

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

■ Confirm the accuracy of the configuration data stored in an FPGA device

■ Alert the system to an occurrence of a configuration error

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

This chapter contains the following sections:

Configuration Error Detection

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

Feedback   Subscribe

Configuration error detection determines if the configuration data received through an external memory device is corrupted during configuration. To validate the configuration data, the Quartus® II software uses a function to calculate the CRC value for each configuration data frame and stores the frame-based CRC value in the configuration data as part of the configuration bit stream.

During configuration, Cyclone IV devices use the same methodology to calculate the CRC value based on the frame of data that is received and compares it against the frame CRC value in the data stream. Configuration continues until either the device detects an error or all the values are calculated.

In addition to the frame-based CRC value, the Quartus II software generates a 32-bit CRC value for the whole configuration bit stream. This 32-bit CRC value is stored in the 32-bit storage register at the end of the configuration and is used for user mode error detection that is discussed in "User Mode Error Detection".

User Mode Error Detection

☞ User mode error detection is available in Cyclone IV GX and Cyclone IV E devices with 1.2-V core voltage. Cyclone IV E devices with 1.0-V core voltage do not support user mode error detection.

Soft errors are changes in a configuration random-access memory (CRAM) bit state due to an ionizing particle. Cyclone IV devices have built-in error detection circuitry to detect data corruption by soft errors in the CRAM cells.

This error detection capability continuously computes the CRC of the configured CRAM bits based on the contents of the device and compares it with the pre-calculated CRC value obtained at the end of the configuration. If the CRCs match, there is no error in the current configuration CRAM bits. The process of error detection continues until the device is reset (by setting nCONFIG to low).

The Cyclone IV device error detection feature does not check memory blocks and I/O buffers. These device memory blocks support parity bits that are used to check the contents of memory blocks for any error. The I/O buffers are not verified during error detection because the configuration data uses flip-flops as storage elements that are more resistant to soft errors. Similar flip-flops are used to store the pre-calculated CRC and other error detection circuitry option bits.

The error detection circuitry in Cyclone IV devices uses a 32-bit CRC IEEE 802 standard and a 32-bit polynomial as the CRC generator. Therefore, a single 32-bit CRC calculation is performed by the device. If a soft error does not occur, the resulting 32-bit signature value is 0x00000000, that results in a 0 on the CRC_ERROR output signal. If a soft error occurs in the device, the resulting signature value is non-zero and the CRC_ERROR output signal is 1.

You can inject a soft error by changing the 32-bit CRC storage register in the CRC circuitry. After verifying the induced failure, you can restore the 32-bit CRC value to the correct CRC value with the same instruction and inserting the correct value.

☞ Before updating it with a known bad value, Altera recommends reading out the correct value.

In user mode, Cyclone IV devices support the CHANGE_EDREG JTAG instruction, that allows you to write to the 32-bit storage register. You can use Jam™ STAPL files (**.jam**) to automate the testing and verification process. You can only execute this instruction when the device is in user mode, and it is a powerful design feature that enables you to dynamically verify the CRC functionality in-system without having to reconfigure the device. You can then use the CRC circuit to check for real errors induced by an SEU.

Table 9–1 describes the CHANGE_EDREG JTAG instructions.

Table 9–1. CHANGE_EDREG JTAG Instruction

JTAG Instruction	Instruction Code	Description
CHANGE_EDREG	00 0001 0101	This instruction connects the 32-bit CRC storage register between TDI and TDO. Any precomputed CRC is loaded into the CRC storage register to test the operation of the error detection CRC circuitry at the CRC_ERROR pin.

☞ After the test completes, Altera recommends that you power cycle the device.

Automated SEU Detection

Cyclone IV devices offer on-chip circuitry for automated checking of SEU detection. Applications that require the device to operate error-free at high elevations or in close proximity to earth's north or south pole require periodic checks to ensure continued data integrity. The error detection cyclic redundancy code feature controlled by the **Device and Pin Options** dialog box in the Quartus II software uses a 32-bit CRC circuit to ensure data reliability and is one of the best options for mitigating SEU.

You can implement the error detection CRC feature with existing circuitry in Cyclone IV devices, eliminating the need for external logic. The CRC is computed by the device during configuration and checked against an automatically computed CRC during normal operation. The CRC_ERROR pin reports a soft error when configuration CRAM data is corrupted. You must decide whether to reconfigure the FPGA by strobing the nCONFIG pin low or ignore the error.

CRC_ERROR Pin

A specific CRC_ERROR error detection pin is required to monitor the results of the error detection circuitry during user mode. Table 9–2 describes the CRC_ERROR pin.

Table 9–2. Cyclone IV Device CRC_ERROR Pin Description

CRC_ERROR Pin Type	Description
I/O, Output (open-drain)	Active high signal indicates that the error detection circuit has detected errors in the configuration SRAM bits. This pin is optional and is used when the CRC error detection circuit is enabled in the Quartus II software from the **Error Detection CRC** tab of the **Device and Pin Options** dialog box.
	When using this pin, connect it to an external 10-kΩ pull-up resistor to an acceptable voltage that satisfies the input voltage of the receiving device.

📝 The CRC_ERROR pin information for Cyclone IV devices is reported in the Cyclone IV Devices Pin-Outs on the Altera® website.

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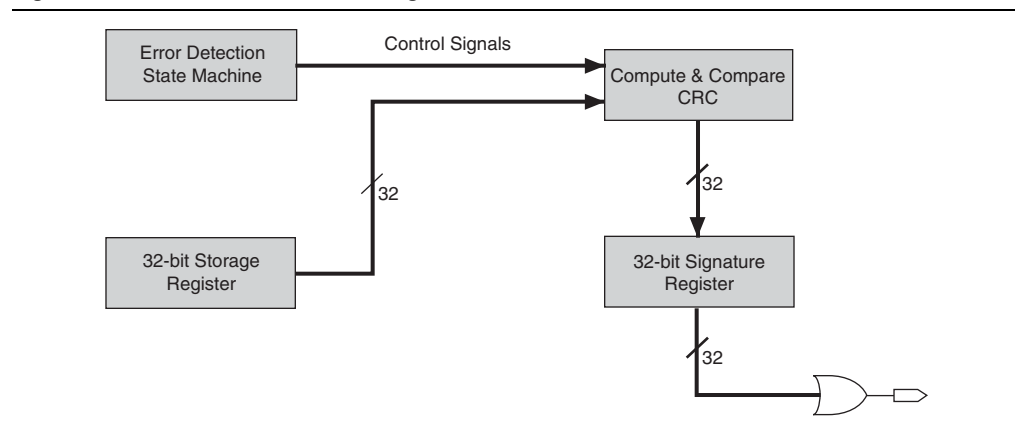
Table 9–4 defines the registers shown in Figure 9–1.

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.

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

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

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.

☞ 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
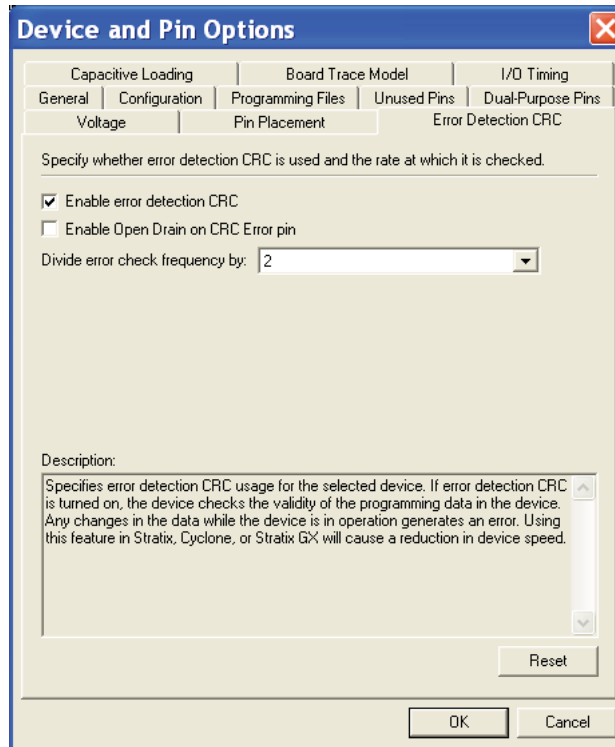

Accessing Error Detection Block Through User Logic

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.

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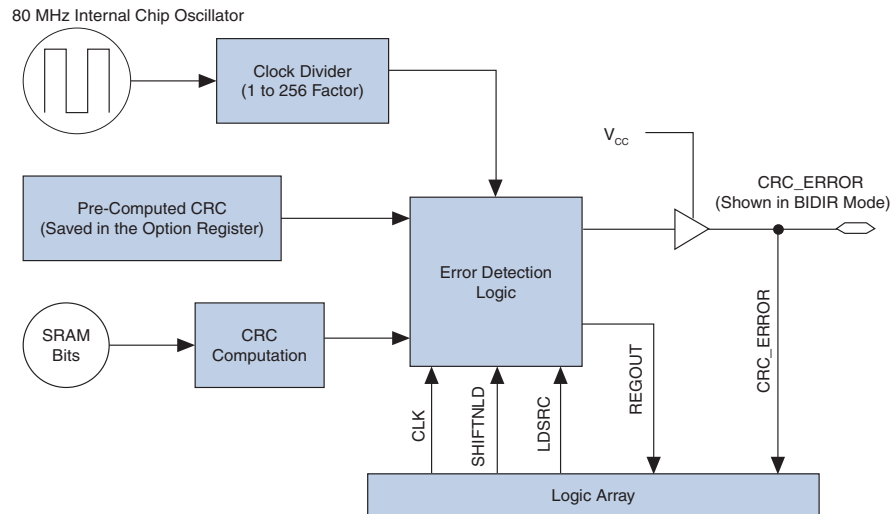
☞ The user logic is affected by the soft error failure, so reading out the 32-bit CRC signature through the regout should not be relied upon to detect a soft error. You should rely on the CRC_ERROR output signal itself, because this CRC_ERROR output signal cannot be affected by a soft error.

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>),

);
```

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.

Document Revision History

Table 9–8 lists the revision history for this chapter.

Table 9–8. Document Revision History

Date	Version	Changes
May 2013	1.3	Updated "CRC_ERROR Pin Type" in Table 9–2.
October 2012	1.2	Updated Table 9–2.
February 2010	1.1	Updated for the Quartus II software version 9.1 SP1 release: ■ Updated "Configuration Error Detection" section. ■ Updated Table 9–6. ■ Added Cyclone IV E devices in Table 9–6.
November 2009	1.0	Initial release.

This chapter describes the boundary-scan test (BST) features that are supported in Cyclone® IV devices. The features are similar to Cyclone III devices, unless stated in this chapter.

Cyclone IV devices (Cyclone IV E devices and Cyclone IV GX devices) support IEEE Std. 1149.1. Cyclone IV GX devices also support IEEE Std. 1149.6. The IEEE Std. 1149.6 (AC JTAG) is only supported on the high-speed serial interface (HSSI) transceivers in Cyclone IV GX devices. The purpose of IEEE Std. 1149.6 is to enable board-level connectivity checking between transmitters and receivers that are AC coupled.

This chapter includes the following sections:

- "IEEE Std. 1149.6 Boundary-Scan Register" on page 10–2
- "BST Operation Control" on page 10–3
- "I/O Voltage Support in a JTAG Chain" on page 10–5
- "Boundary-Scan Description Language Support" on page 10–6

For more information about the JTAG instructions code with descriptions and IEEE Std.1149.1 BST guidelines, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices* chapter.

For more information about the following topics, refer to *AN 39: IEEE 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*:

- IEEE Std. 1149.1 BST architecture and circuitry
- TAP controller state-machine
- Instruction mode

Feedback   Subscribe

IEEE Std. 1149.6 Boundary-Scan Register

The boundary-scan cell (BSC) for HSSI transmitters (GXB_TX[p,n]) and receivers (GXB_RX[p,n]) in Cyclone IV GX devices are different from the BSCs for I/O pins.

Figure 10–1 shows the Cyclone IV GX HSSI transmitter boundary-scan cell.

Figure 10–1. HSSI Transmitter BSC with IEEE Std. 1149.6 BST Circuitry for Cyclone IV GX Devices
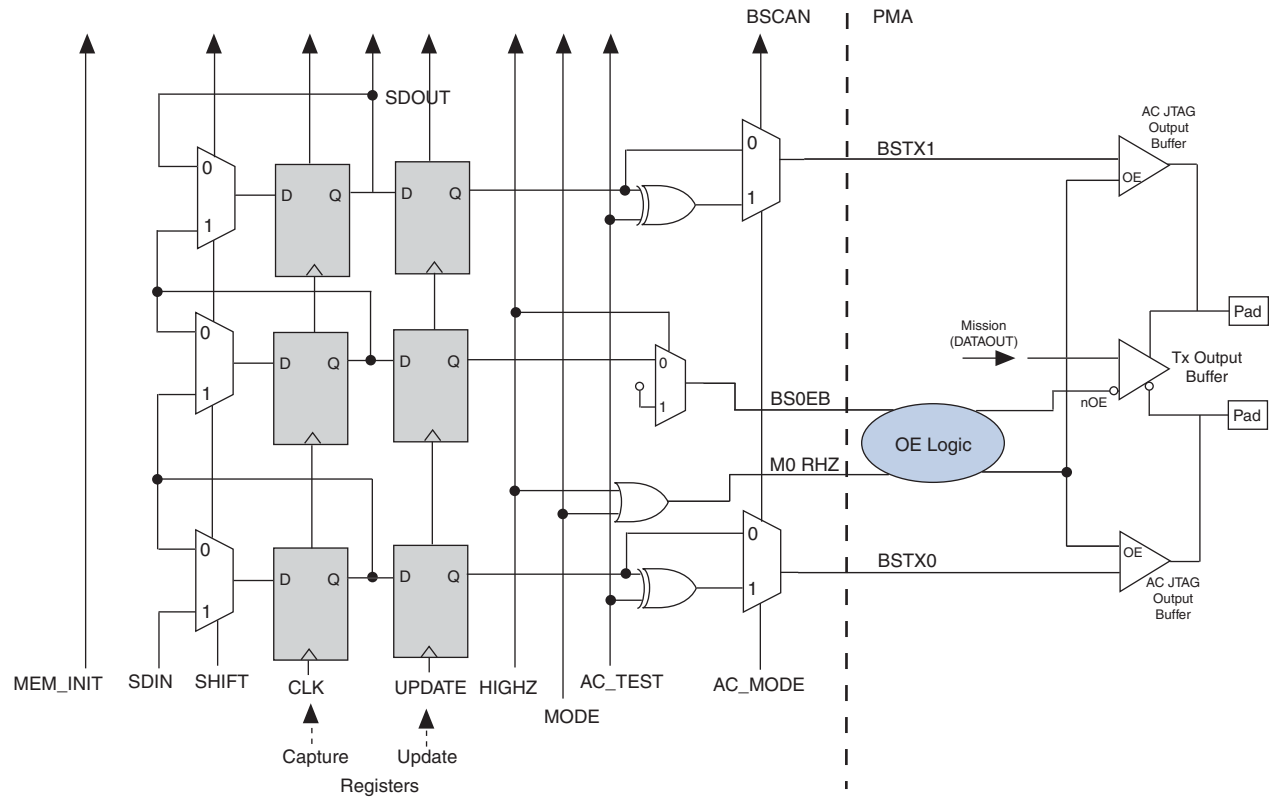
Figure 10–2 shows the Cyclone IV GX HSSI receiver BSC.

Figure 10–2. HSSI Receiver BSC with IEEE Std. 1149.6 BST Circuitry for the Cyclone IV GX Devices
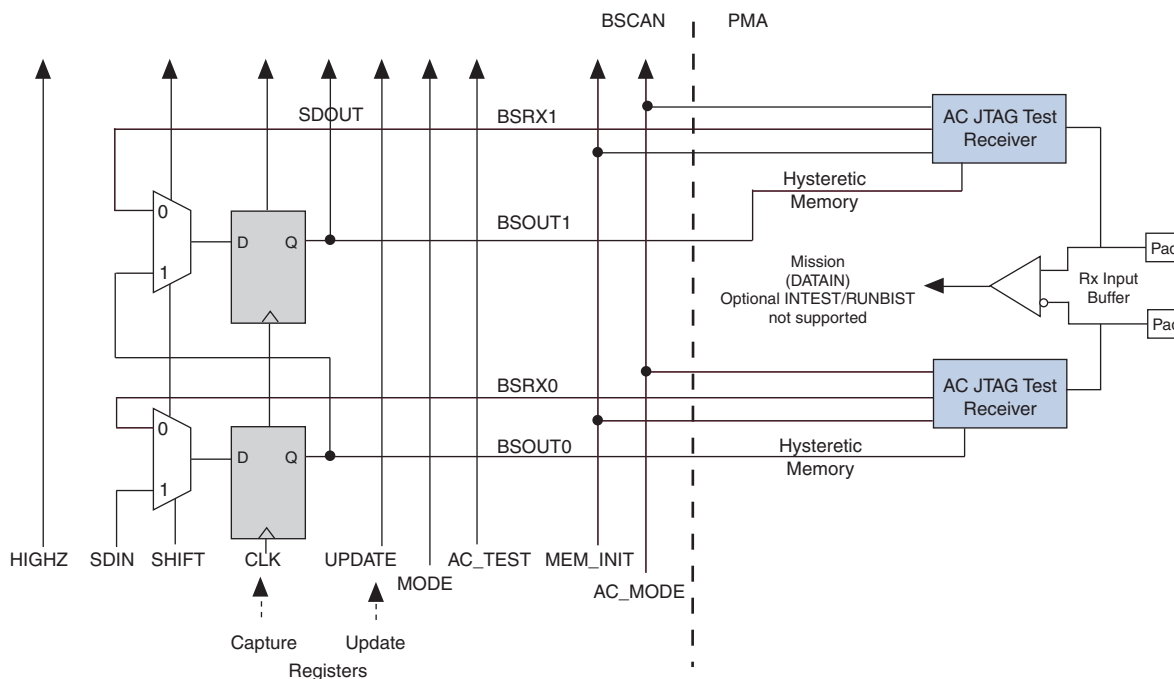
For more information about Cyclone IV devices user I/O boundary-scan cells, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices* chapter.

BST Operation Control

Table 10–1 lists the boundary-scan register length for Cyclone IV devices.

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

Device	Boundary-Scan Register Length
EP4CE6	603
EP4CE10	603
EP4CE15	1080
EP4CE22	732
EP4CE30	1632
EP4CE40	1632
EP4CE55	1164
EP4CE75	1314
EP4CE115	1620
EP4CGX15	260
EP4CGX22	494
EP4CGX30 [1]	494
EP4CGX50	1006

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.

EXTEST_PULSE

The instruction code for `EXTEST_PULSE` is 0010001111. The `EXTEST_PULSE` instruction generates three output transitions:

- Driver drives data on the falling edge of `TCK` in UPDATE_IR/DR.

- Driver drives inverted data on the falling edge of `TCK` after entering the RUN_TEST/IDLE state.

- Driver drives data on the falling edge of `TCK` after leaving the RUN_TEST/IDLE state.

☞ If you use DC-coupling on HSSI signals, you must execute the `EXTEST` instruction. If you use AC-coupling on HSSI signals, you must execute the `EXTEST_PULSE` instruction. AC-coupled and DC-coupled HSSI are only supported in post-configuration mode.

EXTEST_TRAIN

The instruction code for `EXTEST_TRAIN` is 0001001111. The `EXTEST_TRAIN` instruction behaves the same as the `EXTEST_PULSE` instruction with one exception. The output continues to toggle on the `TCK` falling edge as long as the test access port (TAP) controller is in the RUN_TEST/IDLE state.

☞ These two instruction codes are only supported in post-configuration mode for Cyclone IV GX devices.

☞ When you perform JTAG boundary-scan testing before configuration, the `nCONFIG` pin must be held low.

I/O Voltage Support in a JTAG Chain

A Cyclone IV device operating in BST mode uses four required pins: `TDI`, `TDO`, `TMS`, and `TCK`. The `TDO` output pin and all JTAG input pins are powered by the V_{CCIO} power supply of I/O Banks (I/O Bank 9 for Cyclone IV GX devices and I/O Bank 1 for Cyclone IV E devices).

A JTAG chain can contain several different devices. However, you must use caution if the chain contains devices that have different V_{CCIO} levels. The output voltage level of the `TDO` pin must meet the specification of the `TDI` pin it drives. For example, a device with a 3.3-V `TDO` pin can drive a device with a 5.0-V `TDI` pin because 3.3 V meets the minimum TTL-level V_{IH} for the 5.0-V `TDI` pin.

☞ For multiple devices in a JTAG chain with the 3.0-V/3.3-V I/O standard, you must connect a 25-Ω series resistor on a `TDO` pin driving a `TDI` pin.

You can also interface the `TDI` and `TDO` lines of the devices that have different V_{CCIO} levels by inserting a level shifter between the devices. If possible, the JTAG chain should have a device with a higher V_{CCIO} level driving a device with an equal or lower V_{CCIO} level. This way, a level shifter may be required only to shift the `TDO` level to a level acceptable to the JTAG tester.

Figure 10–3 shows the JTAG chain of mixed voltages and how a level shifter is inserted in the chain.

Figure 10–3. JTAG Chain of Mixed Voltages
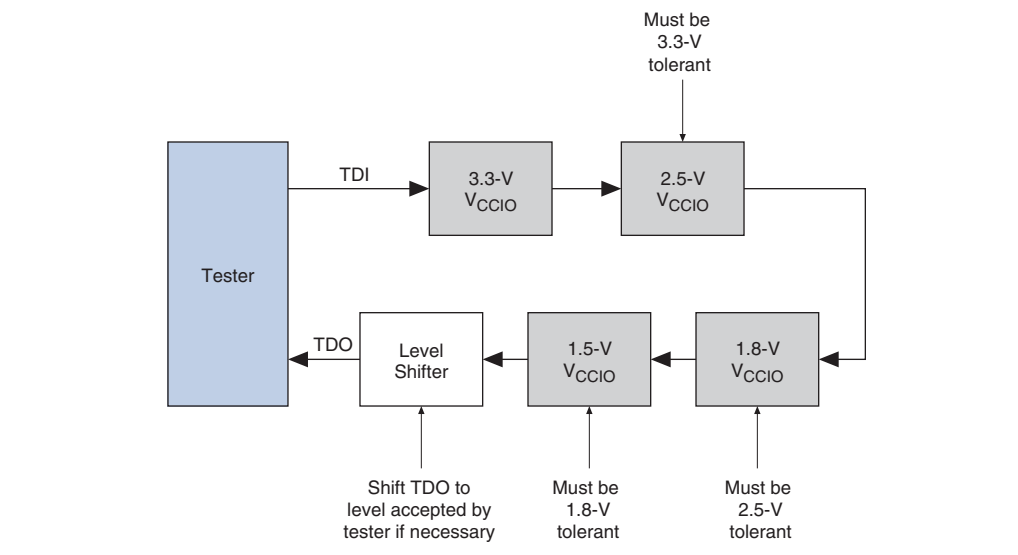
Boundary-Scan Description Language Support

The boundary-scan description language (BSDL), a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.1/IEEE Std. 1149.6 BST-capable device that can be tested.

👣 For more information about how to download BSDL files for IEEE Std. 1149.1-compliant Cyclone IV E devices, refer to *IEEE Std. 1149.1 BSDL Files*.

👣 For more information about how to download BSDL files for IEEE Std. 1149.6-compliant Cyclone IV GX devices, refer to *IEEE Std. 1149.6 BSDL Files*.

👣 You can also generate BSDL files (pre-configuration and post-configuration) for IEEE Std. 1149.1/IEEE Std. 1149.6-compliant Cyclone IV devices with the Quartus® II software version 9.1 SP1 and later. For more information about the procedure to generate BSDL files using the Quartus II software, refer to *BSDL Files Generation in Quartus II*.

Document Revision History

Table 10–3 lists the revision history for this chapter.

Table 10–3. Document Revision History

Date	Version	Changes
December 2013	1.3	■ Updated the "EXTEST_PULSE" section.
November 2011	1.2	■ Updated the "BST Operation Control" section. ■ Updated Table 10–2.
February 2010	1.1	■ Added Cyclone IV E devices in Table 10–1 and Table 10–2 for the Quartus II software version 9.1 SP1 release. ■ Updated Figure 10–1 and Figure 10–2. ■ Minor text edits.
November 2009	1.0	Initial release.

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

This chapter includes the following sections:

- "External Power Supply Requirements" on page 11–1
- "Hot-Socketing Specifications" on page 11–2
- "Hot-socketing Feature Implementation" on page 11–3
- "Power-On Reset Circuitry" on page 11–3

External Power Supply Requirements

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

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

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

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

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

Feedback   Subscribe

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

Power Supply Pin	Nominal Voltage Level (V)	Description
VCCL_GXB	1.2	Transceiver PMA and auxiliary power supply

Notes to Table 11–1:

(1) You must power up VCCA even if the phase-locked loop (PLL) is not used.

(2) I/O banks 3, 8, and 9 contain configuration pins. You can only power up the V$_{CCIO}$ level of I/O banks 3 and 9 to 1.5 V, 1.8 V, 2.5 V, 3.0 V, or 3.3 V. For Fast Passive Parallel (FPP) configuration mode, you must power up the V$_{CCIO}$ level of I/O bank 8 to 1.5 V, 1.8 V, 2.5 V, 3.0 V, or 3.3 V.

(3) All device packages of EP4CGX15, EP4CGX22, and device package F169 and F324 of EP4CGX30 devices have two VCC_CLKIN dedicated clock input I/O located at Banks 3A and 8A. Device package F484 of EP4CGX30, all device packages of EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150 devices have four VCC_CLKIN dedicated clock input I/O bank located at banks 3A, 3B, 8A, and 8B.

(4) You must set VCC_CLKIN to 2.5V if the CLKIN is used as a high-speed serial interface (HSSI) transceiver refclk. When not used as a transceiver refclk, VCC_CLKIN supports 1.2 V/ 1.5 V/ 1.8 V/ 2.5 V/ 3.0 V/ 3.3V voltages.

Table 11–2. Power Supply Descriptions for the Cyclone IV E Devices

Power Supply Pin	Nominal Voltage Level (V)	Description
VCCINT	1.0, 1.2	Core voltage power supply
VCCA [1]	2.5	PLL analog power supply
VCCD_PLL	1.0, 1.2	PLL digital power supply
VCCIO [2]	1.2, 1.5, 1.8, 2.5, 3.0, 3.3	I/O banks power supply

Notes to Table 11–2:

(1) You must power up VCCA even if the PLL is not used.

(2) I/O banks 1, 6, 7, and 8 contain configuration pins.

Hot-Socketing Specifications

Cyclone IV devices are hot-socketing compliant without the need for any external components or special design requirements. Hot-socketing support in Cyclone IV devices has the following advantages:

■ You can drive the device before power up without damaging the device.

■ I/O pins remain tri-stated during power up. The device does not drive out before or during power-up. Therefore, it does not affect other buses in operation.

Devices Driven Before Power-Up

You can drive signals into regular Cyclone IV E I/O pins and transceiver Cyclone IV GX I/O pins before or during power up or power down without damaging the device. Cyclone IV devices support any power-up or power-down sequence to simplify system-level designs.

I/O Pins Remain Tri-stated During Power-Up

The output buffers of Cyclone IV devices are turned off during system power up or power down. Cyclone IV devices do not drive out until the device is configured and working in recommended operating conditions. The I/O pins are tri-stated until the device enters user mode.

☞ The user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are always enabled (after POR) before and during configuration. The weak pull up resistors are not enabled prior to POR.

A possible concern for semiconductor devices in general regarding hot socketing is the potential for latch up. Latch up can occur when electrical subsystems are hot socketed into an active system. During hot socketing, the signal pins may be connected and driven by the active system before the power supply can provide current to the V_{CC} of the device and ground planes. This condition can lead to latch up and cause a low-impedance path from V_{CC} to GND in the device. As a result, the device extends a large amount of current, possibly causing electrical damage.

The design of the I/O buffers and hot-socketing circuitry ensures that Cyclone IV devices are immune to latch up during hot-socketing.

👣 For more information about the hot-socketing specification, refer to the *Cyclone IV Device Datasheet* chapter and the *Hot-Socketing and Power-Sequencing Feature and Testing for Altera Devices* white paper.

Hot-socketing Feature Implementation

The hot-socketing circuit does not include the CONF_DONE, nCEO, and nSTATUS pins to ensure that they are able to operate during configuration. The expected behavior for these pins is to drive out during power-up and power-down sequences.

☞ Altera uses GND as reference for hot-socketing operation and I/O buffer designs. To ensure proper operation, Altera recommends connecting the GND between boards before connecting the power supplies. This prevents the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled up GND can otherwise cause an out-of-specification I/O voltage or current condition with the Altera device.

Power-On Reset Circuitry

Cyclone IV devices contain POR circuitry to keep the device in a reset state until the power supply voltage levels have stabilized during power up. During POR, all user I/O pins are tri-stated until the power supplies reach the recommended operating levels. In addition, the POR circuitry also ensures the V_{CCIO} level of I/O banks that contain configuration pins reach an acceptable level before configuration is triggered.

The POR circuit of the Cyclone IV device monitors the V_{CCINT}, V_{CCA}, and V_{CCIO} that contain configuration pins during power-on. You can power up or power down the V_{CCINT}, V_{CCA}, and V_{CCIO} pins in any sequence. The V_{CCINT}, V_{CCA}, and V_{CCIO} must have a monotonic rise to their steady state levels. All V_{CCA} pins must be powered to 2.5V (even when PLLs are not used), and must be powered up and powered down at the same time.

After the Cyclone IV device enters the user mode, the POR circuit continues to monitor the V_{CCINT} and V_{CCA} pins so that a brown-out condition during user mode is detected. If the V_{CCINT} or V_{CCA} voltage sags below the POR trip point during user mode, the POR circuit resets the device. If the V_{CCIO} voltage sags during user mode, the POR circuit does not reset the device.

In some applications, it is necessary for a device to wake up very quickly to begin operation. Cyclone IV devices offer the Fast-On feature to support fast wake-up time applications. The MSEL pin settings determine the POR time (t_{POR}) of the device.

For more information about the MSEL pin settings, refer to the *Configuration and Remote System Upgrades in Cyclone IV Devices* chapter.

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

Document Revision History

Table 11–3 lists the revision history for this chapter.

Table 11–3. Document Revision History

Date	Version	Changes
May 2013	1.3	Updated Note (4) in Table 11–1.
July 2010	1.2	■ Updated for the Quartus II software version 10.0 release. ■ Updated "I/O Pins Remain Tri-stated During Power-Up" section. ■ Updated Table 11–1.
February 2010	1.1	Updated Table 11–1 and Table 11–2 for the Quartus II software version 9.1 SP1 release.
November 2009	1.0	Initial release.